



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei dieser PDF-Datei handelt es sich um die Texte - Problem, Lösung und VBA-Code - aus über 2.300 Excel-Beispielarbeitsmappen. In der Vollversion können zum einen aus den einzelnen Code-Sequenzen Hilfedateien zu Excel/VBA-Schlüsselwörtern aufgerufen werden. Zudem wird zu jeder Seite auch eine Beispielarbeitsmappe mitgeliefert, bei der die Excel-Funktionen und VBA-Codes offenliegen und frei weiterverwendet werden können.

In der Vollversion finden Sie:

- [Die Excel-Beispiele wie vorstehend beschrieben](#)
- [Eine Reihe von Excel-Anwendungen](#)
- [Einführung in Microsoft Excel](#)
- [Einführung in die VBA-Programmierung](#)
- [Einen Workshop mit Schritt-für-Schritt-Anleitungen](#)

Weitere Excel-Lehr- und Lernmaterialien zur Abrundung der vorstehenden Vollversion der Excel-Beispiele:

- [Das Excel-Tutorial](#)
- [Das Excel-Forumsarchiv](#)

Alle Excel-Materialien können auch als Gesamtpaket bezogen werden.

[Hier kommen Sie zu Bestellseite.](#)



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

AddIn-Namen auslesen

Problem:	Die Namen aller geöffneten AddIns sollen ausgelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Auslesen() Dim oVBP As Object On Error GoTo ERRORHANDLER For Each oVBP In Application.VBE.VBProjects If Right(oVBP.Filename, 3) = "xla" Then MsgBox oVBP.Filename End If ERRORHANDLER: Next On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

AddIn aus Makro aufrufen

Problem:	Wie kann ich aus einem Makro heraus eine AddIn-Prozedur aufrufen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AddInRufen() On Error GoTo ERRORHANDLER Application.Run "vbeplusde.xla!callcomment" Exit Sub ERRORHANDLER: Beep MsgBox prompt:="Bitte AddIn-Namen anpassen!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung ob ein AddIn installiert ist

Problem:	Wie kann ich über VBA abfragen, ob ein bestimmtes AddIn installiert ist?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AddInTest() Dim oAddIn As AddIn Dim sAddIn As String sAddIn = Range("B1").Value For Each oAddIn In Application.AddIns If UCase(oAddIn.Name) = UCase(sAddIn) Then MsgBox UCase(sAddIn) & " ist installiert!" Exit Sub End If Next MsgBox UCase(sAddIn) & " ist nicht installiert!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Den sichtbaren Bereich ermitteln

Problem:	Der im Blatt sichtbare Zell-Bereich wird ermittelt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub FensterBereich() Dim rng As Range Dim iRow As Integer, iCol As Integer, iStartR As Integer, iStartC As Integer Set rng = ActiveCell iStartC = ActiveWindow.ScrollColumn iCol = iStartC Do Until ActiveWindow.ScrollColumn <> iStartC ActiveCell.Offset(0, 1).Activate Loop iStartC = ActiveCell.Column - 1 iStartR = ActiveWindow.ScrollRow iRow = iStartR Do Until ActiveWindow.ScrollRow <> iStartR ActiveCell.Offset(1, 0).Select Loop iStartR = ActiveCell.Row - 1 rng.Select MsgBox "Sichtbarer Bereich: " & Range(Cells(iRow, iCol), Cells(iStartR, iStartC)).Address End Sub StandardModule: Modul2 Sub Makro1() Makro1 Makro Makro am 13.03.2004 von Hans W. Herber aufgezeichnet Range("P1").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob die Zwischenablage leer ist

Problem:	Es soll überprüft werden, ob die Zwischenablage leer ist.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub TestClipboard() Dim bln As Boolean If bln Then MsgBox "Die Zwischenablage ist leer!" Else MsgBox "Die Zwischenablage ist nicht leer!" End If End Sub Private Function ClipboardEmpty() As Boolean ClipboardEmpty = (Application.ClipboardFormats(1) = -1) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblätter ausblenden

Problem:	Alle Arbeitsblätter ausser dem aktiven Blatt sollen aus- bzw. eingeblendet werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SheetsHidden() Dim wks As Worksheet For Each wks In ThisWorkbook.Worksheets If wks.Name <> ActiveSheet.Name Then wks.Visible = xlVeryHidden End If Next wks End Sub Sub SheetsVisible() Dim wks As Worksheet For Each wks In ThisWorkbook.Worksheets wks.Visible = True Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spaltenbreite gemäß den in den Zellen stehenden Zahlen festlegen

Problem:	Die Spaltenbreite im Bereich A:J soll gemäß den in Zeile 1 stehenden Prozentzahlen aufgeteilt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SpaltenBreite() Dim dWidth As Double Dim iCounter As Integer For iCounter = 1 To 10 dWidth = dWidth + Columns(iCounter).ColumnWidth Next iCounter For iCounter = 1 To 10 Columns(iCounter).ColumnWidth = _ dWidth * Cells(1, iCounter) / 100 Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Markierung von Wochenenden

Problem:	Alle Wochenenden sollen mit einem grauen Hintergrund versehen werden.
Lösung:	Formel in bedingte Formatierung: =WOCHENTAG(A1;2)>5
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verankerung von Schaltflächen am oberen Tabellenblattrand

Problem:	Wie können Schaltflächen am oberen Tabellenblattrand fixiert werden?
Lösung:	Fügen Sie die Schaltflächen am oberen Tabellenblattrand ein, wählen Sie eine Zelle in der Zeile unterhalb aus und wählen Sie Menü FENSTER / FIXIEREN.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenzeilen in Abhängigkeit einer zweiten Tabelle einblenden

Problem:	Die ersten 25 Zellen in Spalte A sollen auf Inhalt geprüft werden. Ist die jeweilige Zelle nicht leer, soll die entsprechende Zeile in Tabelle2 eingeblendet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub PruefenEinblenden() Dim iRow As Integer For iRow = 1 To 25 If Not IsEmpty(Cells(iRow, 1)) Then Worksheets("Tabelle2").Rows(iRow).Hidden = False End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bereich aus anderer Tabelle für 2 Sekunden anzeigen

Problem:	Wie kann ich einen vorgegebenen Bereich aus einer zweiten Tabelle im aktiven Blatt für 2 Sekunden anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Anzeige() Worksheets("Rang").Range("B2:M16").CopyPicture _ Appearance:=xlScreen, Format:=xlBitmap With Worksheets("Tabelle1") .Paste Destination:=.Range("B2") End With Application.Wait Now + TimeSerial(0, 0, 2) ActiveSheet.Shapes(ActiveSheet.Shapes.Count).Delete End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenumbrüche im aktiven Blatt entfernen

Problem:	Wie kann ich alle Zeilenumbrüche im aktiven Blatt entfernen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub UmbruchRaus() Dim rng As Range For Each rng In ActiveSheet.UsedRange.Cells If InStr(rng.Text, Chr(10)) Then rng = Application.Substitute(rng, Chr(10), " ") End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schriftart in Zeilen wechselweise fett/nicht-fett formatieren

Problem:	Wie kann ich in zwei Zeilen wechselweise den Schrifttyp "fett" wechseln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ToggleBold() With Worksheets(1) With Rows(1) .Font.Bold = Not .Font.Bold .Rows(2).Font.Bold = Not .Font.Bold End With End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Scrollbereich begrenzen

Problem:	Wie kann ich den Bewegungsraum des Cursors einschränken?
Lösung:	Den nachstehenden Code in die "Personl.xls" einfügen und einer Symbolleistenschaltfläche zuweisen.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdBegrenzung_Click() If ActiveSheet.ScrollArea = "" Then Cells.Interior.ColorIndex = 15 Selection.Interior.ColorIndex = xlColorIndexNone ActiveSheet.ScrollArea = Selection.Address Else ActiveSheet.ScrollArea = "" Cells.Interior.ColorIndex = xlColorIndexNone End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzeige der Formel mit Quellblattangabe

Problem: Im Kommentar soll das mit dem Zellinhalt verknüpfte Tabellenblatt genannt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)
    Dim cmt As Comment
    Dim sTxt As String
    If Target.Cells.Count > 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Application.DisplayCommentIndicator = xlCommentIndicatorOnly
    If Not Target.Comment Is Nothing Then
        Target.Comment.Delete
    End If
    If Target.HasFormula Then
        If InStr(Target.Formula, "!") > 0 Then
            sTxt = Left(Target.Formula, InStr(Target.Formula, "!") - 1)
            Set cmt = Target.AddComment(sTxt)
            cmt.Shape.TextFrame.AutoSize = True
        End If
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen mit Zellen mit rotem Hintergrund ausblenden

Problem:	Wie kann ich alle Zeilen mit Zellen mit einem roten Hintergrund ausblenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Ausblenden() Dim rng As Range For Each rng In Range("A1:C100").Cells If rng.Interior.ColorIndex = 3 Then Rows(rng.Row).Hidden = True End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Laufschrift erzeugen

Problem:	Wie kann ich in Excel eine Laufschrift erzeugen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long) Sub Rotieren() Dim rng As Range Dim iCounter As Integer Application.EnableCancelKey = xlErrorHandler Set rng = Range("E10") On Error GoTo ERRORHANDLER For iCounter = 1 To 10000 rng.Value = Right(rng.Value, Len(rng.Value) - 1) + _ Left(rng.Value, 1) Sleep 100 Next iCounter ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle leeren Spalten ausblenden

Problem:	Wie kann ich alle leeren Spalten eines Tabellenblattes ausblenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub LeereSpalteAus() Dim iCounter As Integer Application.ScreenUpdating = False For iCounter = 1 To 256 If WorksheetFunction.CountA(Columns(iCounter)) = 0 Then Columns(iCounter).Hidden = True End If Next iCounter Application.ScreenUpdating = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Startansicht wiederherstellen

Problem:	Wie kann ich die vor dem Ablauf eines VBA-Programms angezeigte Startansicht am Ende des Programms wiederherstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZurAnsichtZurueck() ActiveWorkbook.CustomViews.Add _ ViewName:="Test", _ PrintSettings:=False, _ RowColSettings:=True Columns("B:C").Hidden = True Rows("3:12").Hidden = True Range("E14").Select MsgBox "Weiter?" With ActiveWorkbook.CustomViews("Test") .Show .Delete End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sanduhr als Mauszeiger

Problem:	Gibt es eine Möglichkeit, das Erscheinungsbild des Mauszeigers in Excel auf Sanduhr umzustellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ChangeCursor() If Application.Cursor = xlWait Then Application.Cursor = xlNormal ActiveSheet.Buttons(1).Caption = "Zeiger auf Sanduhr" Else Application.Cursor = xlWait ActiveSheet.Buttons(1).Caption = "Zeiger auf Normal" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenhöhe in cm festlegen

Problem:	Wie lege ich die Zeilenhöhe statt in Punkten in Zentimetern fest?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZeilenHoehe() Dim varCM As Variant varCM = InputBox(_ prompt:="Bitte die gewünschte Zeilenhöhe in cm angeben:") If varCM = "" Then Exit Sub Selection.RowHeight = CDbl(varCM) / 0.035 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Rechnungsnummer in Registry eintragen, auslesen, löschen

Problem:	Wie kann ich eine Rechnungsnummer in die Registry eintragen, später auslesen, zurücksetzen und den Registryeintrag wieder löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AnlegenUndHochzaehlen() Dim var As Variant var = GetSetting(_ appname:="Rechnungswesen", _ section:="Rechnungen", _ key:="Nummer") If var = "" Then SaveSetting _ appname:="Rechnungswesen", _ section:="Rechnungen", _ key:="Nummer", _ Setting:=1 Range("B1") = 1 Else SaveSetting _ appname:="Rechnungswesen", _ section:="Rechnungen", _ key:="Nummer", _ Setting:=var + 1 Range("B1") = var + 1 End If End Sub Sub RegistryLoeschen() On Error Resume Next DeleteSetting "Rechnungswesen" Range("B1").ClearContents End Sub Sub RegistryReset() Dim var As Variant var = InputBox(_ prompt:="Bitte numerische Rechnungsnummer eingeben:") If var = "" Then Exit Sub SaveSetting _ appname:="Rechnungswesen", _ section:="Rechnungen", _ key:="Nummer", _ Setting:=var Range("B1") = var End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Assistent zum Dialog aufrufen und Eingaben auswerten

Problem:	Wie kann ich den Assistenten mit einer CheckListe aufrufen und diese Checkliste später auswerten?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AssistentEinblenden() Dim intCounter As Integer On Error GoTo ERRORHANDLER Assistant.Visible = True With Assistant.NewBalloon .Heading = "Dämlicher Karneval" .Text = "Beantworten Sie bitte folgende Fragen:" .CheckBoxes(1).Text = "Nur für Rheinländer geeignet" .CheckBoxes(2).Text = "Keine Kohle für so'n Quatsch" .CheckBoxes(3).Text = "Soeben erst zurückgekommen" .Button = msoButtonSetOkCancel If .Show = msoBalloonButtonOK Then If .CheckBoxes(1).Checked And Not .CheckBoxes(2).Checked _ And Not .CheckBoxes(3).Checked Then MsgBox "Nicht für alle Rheinländer" ElseIf .CheckBoxes(2).Checked And .CheckBoxes(3).Checked Then MsgBox "Dafür hast Du ziemlich lange ausgehalten" ElseIf Not .CheckBoxes(1).Checked And Not .CheckBoxes(2).Checked _ And .CheckBoxes(3).Checked Then MsgBox "Willkommen zu Hause" ElseIf .CheckBoxes(2).Checked Then MsgBox "Wer hat die schon" End If End If End With Exit Sub ERRORHANDLER: Beep MsgBox prompt:="Assistent ist nicht installiert!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Funktionstasten aus- und einschalten

Problem: Wie kann ich auf Schaltflächenbetätigung die Funktionen aller Funktionstasten aus- und wieder einschalten?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub cmdStart()
    With ActiveSheet.Buttons(1)
        If .Caption = "Funktionstasten aus" Then
            Call TastenBelegungNeu
            .Caption = "Funktionstasten ein"
        Else
            Call TastenBelegungAus
            .Caption = "Funktionstasten aus"
        End If
    End With
End Sub

Sub TastenBelegungNeu()
    Dim iCounter As Integer
    For iCounter = 1 To 12
        Application.OnKey "{F" & iCounter & "}", ""
    Next iCounter
    Application.OnKey "{tab}", "Eingabe"
End Sub

Sub TastenBelegungAus()
    Dim iCounter As Integer
    For iCounter = 1 To 12
        Application.OnKey "{F" & iCounter & "}"
    Next iCounter
    Application.OnKey "{tab}"
End Sub

Sub Eingabe()
    Application.SendKeys "{enter}"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abfrage, ob 16- oder 32-Bit-System

Problem:	Wie kann ich das Betriebssystem abfragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub GetSystem() MsgBox prompt:=Application.OperatingSystem End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spracheinstellung der Excel-Anwendung feststellen

Problem:	Wie kann ich über VBA die über die Excel-Ländereinstellung feststellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Sprache() MsgBox Application.International(1) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makro in Abhängigkeit der Excel-Version verzweigen

Problem: Wie kann ich ein Makro in Abhängigkeit der abzufragenden Excel-Version verzweigen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub VersionsTest()
    Select Case Left(Application.Version, 1)
        Case Is >= 8: Call Meldung1
        Case 5, 7: Call Meldung2
    End Select
End Sub

Private Sub Meldung1()
    MsgBox prompt:="Version >= 8"
End Sub

Private Sub Meldung2()
    MsgBox prompt:="Version 5/7"
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte auf Blätter verteilen

Problem:	Für jede Gruppe ist im Blatt BlattIndex ein Tabellenblattname hinterlegt. Die einzelnen Datensätze sollen nach diesem Index auf die Tabellenblätter verteilt werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub BlaetterAnlegen() Dim iWks As Integer For iWks = 1 To 6 Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Name = Worksheets("BlattIndex").Cells(iWks, 2).Value Next iWks Worksheets(1).Select End Sub Sub Eintragen() Dim vRow As Variant Dim iRow As Integer, iRowT As Integer iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) vRow = Application.Match(Cells(iRow, 1).Value, Worksheets("BlattIndex").Columns(1), 0) If Not IsError(vRow) Then With Worksheets(Worksheets("BlattIndex").Cells(vRow, 2).Value) iRowT = .Cells(Rows.Count, 1).End(xlUp).Row + 1 .Rows(iRowT).Value = Rows(iRow).Value End With End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen mit Formelfehlern leeren

Problem:	Alle Zellen mit Formelfehlern sollen geleert werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub DeleteErrors() Dim rng As Range Set rng = Tabelle1.UsedRange.SpecialCells(xlCellTypeFormulas, xlErrors) rng.ClearContents End Sub ClassModule: Tabelle2 Private Sub cmdHyperlink_Click() ActiveWorkbook.FollowHyperlink "http://www.excel-cd.de/i_bestellung.html", NewWindow:=True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Jedes zweite Arbeitsblatt auswählen

Problem:	Jedes zweite Arbeitsblatt soll ausgewählt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SelectWks() Dim iWks As Integer For iWks = ActiveSheet.Index To Worksheets.Count Step 2 Worksheets(iWks).Select False Next iWks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zielwertsuche über VBA

Problem:	In den Spalten A:D wird über VBA eine Zielwertsuche durchgeführt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetGoalSeek() Dim iCol As Integer For iCol = 1 To 4 Cells(13, iCol).GoalSeek Goal:=555, ChangingCell:=Cells(1, iCol) Next iCol End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattauswahl über Schaltflächen

Problem:	Über vier Schaltflächen sollen Tabellenblätter gem. Schaltflächenaufschrift ausgewählt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn den Schaltflächen zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SelectCover() Application.Goto Worksheets("Cover").Range("A1"), True End Sub Sub SelectSheets() Dim swks As String swks = ActiveSheet.Buttons(Application.Caller).Caption Application.Goto Worksheets(swks).Range("A1"), True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spaltenbreiten der nicht ausgeblendeten Spalten anpassen

Problem:	Es sollen nur die Breiten der nicht ausgeblendeten Spalten automatisch angepaßt werden. Es sollen nur die ausgewählten Spalten berücksichtigt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub OnlyNoHiddens() Dim iCol As Integer For iCol = Selection.Column To Selection.Column + Selection.Columns.Count - 1 If Columns(iCol).Hidden = False Then Columns(iCol).AutoFit End If Next iCol End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit ShortCut zwischen zwei Tabellen wechseln

Problem:	Ist Tabell1 aktiviert, soll mit dem ShortCut Strg+n in Tabelle3, ist Tabelle3 aktiviert, in Tabelle1 gewechselt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einem ShortCut zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Changewks() If ActiveSheet.Name = "Tabelle1" Then Worksheets("Tabelle3").Activate ElseIf ActiveSheet.Name = "Tabelle3" Then Worksheets("Tabelle1").Activate End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auswahl des jeweils nächsten Blattes

Problem:	Es soll das jeweils nächste Blatt unter Berücksichtigung ausgeblendeter Blätter ausgewählt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SelectSheet() Dim iWks As Integer iWks = ActiveSheet.Index + 1 If iWks > Worksheets.Count Then iWks = 1 Do Until Worksheets(iWks).Visible iWks = iWks + 1 If iWks > Worksheets.Count Then iWks = 1 Loop Worksheets(iWks).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Umwandlung von Zeichenfolgen in Werte

Problem:	Die Zeichenfolgen aus dem Bereich B4:B10 sollen unter Berücksichtigung der Minuszeichen in Werte umgewandelt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetMinus() Dim wks As Worksheet Dim rng As Range For Each wks In Worksheets Cells.Replace ".", "µ" Cells.Replace ",", "." Cells.Replace "µ", "," For Each rng In wks.UsedRange If Not IsEmpty(rng) Then If IsNumeric(Left(rng.Value, Len(rng.Value) - 1)) And _ Right(rng.Value, 1) = "-" Then rng.Formula = "-" & _ WorksheetFunction.Substitute(rng.Value, "-", "") End If End If Next rng Next wks MsgBox "Job erledigt!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattnamen vor Änderung schützen

Problem:	Die Tabellenblattnamen sollen vor Änderungen geschützt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn Schaltflächen zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SchutzEin() ActiveWorkbook.Protect _ Password:="Passwort", _ structure:=True, _ Windows:=False End Sub Sub SchutzAus() ActiveWorkbook.Unprotect "Passwort" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in Formel und vice versa

Problem: Die Formel in Zelle M21 soll bei Schaltflächenklick in den Wert 0 oder der Wert Wert 0 soll in eine Formel umgewandelt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub EditM21()
    With Range("M21")
        If .HasFormula Then
            .Value = 0
        Else
            .Formula = "=M61"
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Artikel im Lagerbestand austragen

Problem:	Die Werte aus Blatt Ausgang sollen im Blatt Lagerbestand ausgetragen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Loeschen() Dim wks As Worksheet Dim vRow As Variant, vCol As Variant Dim iRow As Integer Set wks = Worksheets("Lagerbestand") iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) vRow = Application.Match(Cells(iRow, 1).Value, wks.Columns(1), 0) If Not IsError(vRow) Then vCol = Application.Match(_ Cells(iRow, 2).Value, _ wks.Range(wks.Cells(vRow, 2), _ wks.Cells(vRow, 256)), 0) If Not IsError(vCol) Then wks.Cells(vRow, vCol + 1).Delete shift:=xlShiftToLeft End If End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Umwandeln von Zeichenfolgen in Währung

Problem:	Die Zeichenfolgen in Spalte A sollen in Währung umgewandelt werden. Der zu bearbeitende Bereich ist auszuwählen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Umwandeln() Dim rng As Range For Each rng In Selection.Cells With rng .NumberFormat = "€ #,##0.00" .Value = .Value End With Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle sichtbaren Zeilen zählen

Problem:	Es solle die sichtbare Zeilen gezählt und die Anzahl als Meldung ausgegeben werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub VisibleRowCount() Dim lRows As Long Dim rng As Range Set rng = Cells.SpecialCells(xlCellTypeVisible) lRows = rng.Cells.Count / 256 MsgBox "Sichtbare Zeilen: " & lRows End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wechseln zwischen Arbeitsblättern anhand mehrerer Schaltflächen

Problem: Über mehrere Schaltflächen soll von Blatt zu Blatt gesprungen werden, wobei die übrigen Arbeitsblätter jeweils xlVeryHidden auszublenden sind.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn den Schaltflächen zu.

VBA-Code:

```
StandardModule: Modul1

Sub EinAusblenden()
    Dim wks As Worksheet
    Set wks = ActiveSheet
    Worksheets(AActiveSheet.Buttons(Application.Caller).Caption).Visible = xlSheetVisible
    wks.Visible = xlSheetVeryHidden
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattwechsel zwischen xlVeryHidden ausgeblendeten Blättern

Problem:	Über eine Schaltfläche soll von Blatt zu Blatt gesprungen werden, wobei die übrigen Arbeitsblätter jeweils xlVeryHidden auszublenden sind.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub EinAusblenden() Dim wks As Worksheet Set wks = ActiveSheet Worksheets(AActiveSheet.Buttons(Application.Caller).Caption).Visible = xlSheetVisible wks.Visible = xlSheetVeryHidden End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Cursor nur in Eingabefeldern bewegen

Problem: Der Cursor soll über die TabTaste nur in den Eingabefeldern bewegt werden können. Nur die Eingabezellen sind ungeschützt; der Blattschutz ist gesetzt.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Set rng = ActiveCell
    iColorA = 49
    iColorB = 6
End Sub

ClassModule: Tabelle2

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    ActiveSheet.Protect userinterfaceonly:=True
    If Not rng Is Nothing Then
        rng.Interior.ColorIndex = iColorA
        rng.Font.ColorIndex = iColorB
    End If
    Set rng = Target
    iColorA = Target.Interior.ColorIndex
    iColorB = Target.Font.ColorIndex
    Target.Interior.ColorIndex = 6
    Target.Font.ColorIndex = 49
End Sub

StandardModule: Modul1

Public rng As Range
Public iColorA As Integer, iColorB As Integer
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kassenbuch mit Zu- und Abrechnung von Ein- / Ausgängen

Problem:	Es soll ein einfaches Kassenbuch erstellt werden, das Ein- und Auszahlungen automatisch dem Kassenbestand zu- oder abrechnet.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über InputBox zu ermittelnde Spaltenanzahl einfügen

Problem:	Es soll eine über InputBox abzufragende Spaltenzahl eingefügt werden. Die Startspalte wird im Code über eine Konstante festgelegt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub InsertColumns() Const ciStart As Integer = 4 Dim vCount As Variant vCount = Application.InputBox("Spaltenanzahl:", Type:=1) If vCount = False Then Exit Sub Range(Cells(1, ciStart), Cells(1, ciStart + vCount - 1)).EntireColumn.Insert End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingeblendete Zeilen mit Werten zählen

Problem:	Es sollen die Zeilen gezählt werden, die Werte enthalten und nicht ausgeblendet sind.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub VisibleRowCount() Dim iRow As Integer, iRowL As Integer, iColL As Integer, iCounter As Integer iColL = Cells(1, 256).End(xlToLeft).Column iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL If Rows(iRow).Hidden = False Then If WorksheetFunction.CountA(Rows(iRow)) > 0 Then iCounter = iCounter + 1 End If End If Next iRow MsgBox iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schaltflächen gem. Tabelleninhalten anlegen und benennen.

Problem: Mit der ersten Schaltfläche sollen in Tabelle2 Buttons angelegt, mit der zweiten benannt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub CreateButtons()
    Dim btn As Button
    Dim rng As Range
    Dim iCounter As Integer
    With Worksheets("Tabelle2")
        .Buttons.Delete
        For iCounter = 1 To 40
            Set rng = .Cells(iCounter, 1)
            Set btn = Worksheets("Tabelle2").Buttons.Add( _
                rng.Left, rng.Top, rng.Width, rng.Height)
        Next iCounter
    End With
End Sub

Sub SetBtnNames()
    Dim btn As Button
    Dim iRow As Integer
    For iRow = 1 To 40
        Worksheets("Tabelle2").Buttons(iRow).Caption = _
            Cells(iRow, 1).Value
    Next iRow
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblatt aus einer anderen Arbeitsmappe importieren

Problem:	Aus der in Zelle B1 genannten Arbeitsmappe soll das 1. Tabellenblatt in diese Arbeitsmappe importiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub TabImport() Dim wkb As Workbook Dim sFile As String Application.ScreenUpdating = False sFile = Range("B1").Value If Dir(sFile) = "" Then Beep MsgBox "Datei wurde nicht gefunden!" Exit Sub End If On Error GoTo ERRORHANDLER Application.EnableEvents = False Set wkb = Workbooks.Open(sFile, False) With ThisWorkbook Worksheets(1).Copy after:=.Worksheets(.Worksheets.Count) End With wkb.Close savechanges:=False ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblätter einschließlich Link-Schaltflächen anlegen

Problem:	Es sollen neue Tabellenblätter mit den in Spalte A genannten Namen erstellt werden. Jedem Tabellenblatt ist eine Schaltfläche hinzuzufügen und in diesem Blatt sind Hyperlinks zu den neuen Blättern zu setzen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub NewSheets() Dim wks As Worksheet Dim btn As Button Dim iWks As Integer Application.ScreenUpdating = False Set wks = ActiveSheet iWks = 1 Do Until IsEmpty(wks.Cells(iWks, 1)) Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Name = wks.Cells(iWks, 1).Value Set btn = ActiveSheet.Buttons.Add(200, 100, 120, 20) btn.Caption = "Rufe Makro " & iWks btn.OnAction = "Makro" & iWks wks.Hyperlinks.Add _ anchor:=wks.Cells(iWks, 1), _ Address:="", _ SubAddress:=ActiveSheet.Name & "!A1" iWks = iWks + 1 Loop Worksheets(1).Select Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblatt nach Zellinhalt benennen

Problem:	Das aktive Blatt soll mit der Zeichenfolge aus A1 benannt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub WksName() ActiveSheet.Name = Range("A1").Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen der selektierten Tabellenblätter auslesen

Problem:	Die Namen der selektierten Tabellen sollen genannt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GroupSheets() Dim iCounter As Integer With ActiveWindow.SelectedSheets For iCounter = 1 To .Count MsgBox .Item(iCounter).Name Next iCounter End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

TextBox mit 12.000 Zeichen unterhalb einer Tabelle einfügen

Problem: Zwei Zeilen unter der letzten genutzten Zelle in Spalte C soll eine TextBox mit einem 12.000 Zeichen langen Text eingefügt werden.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code: StandardModule: Modul1

```
Sub InsertTextBoxA()  
    Dim txtBox As Shape  
    Dim rng As Range  
    Dim iCounter As Integer  
    Set rng = Cells(Rows.Count, 3).End(xlUp).Offset(2, 0)  
    Set txtBox = Tabelle1.Shapes.AddTextbox( _  
        msoTextOrientationHorizontal, rng.Left, rng.Top, 400, 200)  
    For iCounter = 1 To 5  
        With txtBox.TextFrame.Characters  
            .Text = .Text & String(200, CStr(iCounter))  
        End With  
    Next iCounter  
End Sub  
  
Sub InsertTextBox()  
    Dim txtBox As OLEObject  
    Dim rng As Range  
    Dim iCounter As Integer  
    Set rng = Cells(Rows.Count, 3).End(xlUp).Offset(2, 0)  
    Set txtBox = Tabelle1.OLEObjects.Add( _  
        link:=False, _  
        displayasicon:=False, _  
        ClassType:="Forms.TextBox.1", _  
        Left:=rng.Left, _  
        Top:=rng.Top, _  
        Width:=400, _  
        Height:=200)  
    txtBox.Object.MultiLine = True  
    txtBox.Object.Text = String(12000, "X")  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einlesen von Blattnamen und Zellwerten

Problem:	In das aktive Blatt sollen die Zellinhalte der jeweiligen A1-Zellen der Folgeblätter und deren Blattnamen eingelesen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Einlesen() Dim iCounter As Integer, iRow As Integer For iCounter = ActiveSheet.Index + 1 To Worksheets.Count iRow = iRow + 1 Cells(iRow, 1) = Worksheets(iCounter).Name Cells(iRow, 2) = Worksheets(iCounter).Range("A1").Value Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anlegen von Tabellenblättern gem. Namensliste

Problem: Für jeden Namen soll ein Tabellenblatt angelegt werden. Die den Namen zugehörigen Daten sind in die entsprechenden Tabellenblätter zu kopieren.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub NachNamenKopieren()
    Dim wks As Worksheet, wksTarget As Worksheet
    Dim iRow As Integer, iRowT As Integer
    Application.ScreenUpdating = False
    Set wks = ActiveSheet
    iRow = 1
    Do Until IsEmpty(wks.Cells(iRow, 1))
        If Left(wks.Cells(iRow, 1).Value, 4) = "name" Then
            On Error Resume Next
            Set wksTarget = Worksheets(wks.Cells(iRow, 1).Value)
            If Err > 0 Or wksTarget Is Nothing Then
                Err.Clear
                Worksheets.Add after:=Worksheets(Worksheets.Count)
                ActiveSheet.Name = wks.Cells(iRow, 1).Value
                Set wksTarget = ActiveSheet
                iRowT = 0
            Else
                iRowT = wksTarget.Cells(Rows.Count, 1).End(xlUp).Row
            End If
            On Error GoTo 0
            iRow = iRow + 1
            Do Until Left(wks.Cells(iRow, 1).Value, 4) = "name"
                iRowT = iRowT + 1
                wksTarget.Cells(iRowT, 1).Value = wks.Cells(iRow, 1).Value
                iRow = iRow + 1
                If IsEmpty(wks.Cells(iRow, 1)) Then Exit Do
            Loop
        End If
    Loop
    wks.Select
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellen von selektierten Arbeitsblattnamen löschen

Problem: Wie kann ich die Arbeitsblätter, deren Namen ich in einer Tabelle ausgewählt habe, ohne Rückfrage löschen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub DeleteTabs()
    Dim rng As Range
    Application.DisplayAlerts = False
    On Error Resume Next
    For Each rng In Range("A1").CurrentRegion.Cells
        Worksheets(rng.Value).Delete
    Next rng
    On Error GoTo 0
    Application.DisplayAlerts = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblatt kopieren und dreistellig fortlaufend numerieren

Problem:	Wie kann ich auf Schaltflächendruck ein Tabellenblatt kopieren und für dieses eine dreistellige fortlaufende Numerierung vergeben? Bereits vergebene Namen dürfen sich auch dann nicht wiederholen, wenn die Blätter inzwischen gelöscht wurden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NeuesBlatt() Dim wks As Worksheet Dim nme As Name Dim iNme As Integer Dim sNme As String For Each nme In ThisWorkbook.Names sNme = Right(nme.Name, 3) If Len(sNme) = 3 And IsNumeric(sNme) Then If CInt(sNme) > iNme Then iNme = CInt(sNme) End If End If Next nme With ThisWorkbook .Worksheets("Muster").Copy after:=.Worksheets(.Worksheets.Count) End With ActiveSheet.Name = "A" & Format(iNme + 1, "000") Set nme = ActiveWorkbook.Names.Add(_ Name:="WKS" & Format(iNme + 1, "000"), _ RefersTo:=Range("A1"), _ Visible:=False) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Monatsblätter mit Gültigkeitsliste anlegen

Problem:	Wie kann ich über VBA 12 Monatsblätter anlegen und die Gültigkeit der Werte in Spalte A jeden Blattes auf einen bestimmten Bereich des ersten Tabellenblattes begrenzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub GueltigkeitFestlegen() Dim iCounter As Integer Columns("A").Name = "nListe" For iCounter = 1 To 12 Worksheets.Add after:=Worksheets(Worksheets.Count) With ActiveSheet .Name = Format(DateSerial(1, iCounter, 1), "mmmm") .Columns("A").Validation.Add Type:=xlValidateList, _ AlertStyle:=xlValidAlertStop, Operator:= _ xlBetween, Formula1:="=nListe" End With Next iCounter Worksheets(1).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über Schaltfläche neues Blatt mit aktuellem Datum einfügen

Problem: Wie kann ich über eine Schaltfläche ein neues Arbeitsblatt mit dem aktuellen Datum als Blattnamen einfügen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub NeuesBlattErstellen()
    Worksheets.Add after:=Worksheets(Worksheets.Count)
    ActiveSheet.Name = Format(Date, "yyddmm")
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausgeblendetes Tabellenblatt bei Paßworteingabe anzeigen

Problem:	Wie kann ich ein ausgeblendetes Tabellenblatt nach Eingabe eines richtigen Paßwortes anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub cmdBlattAufruf_Click() Dim sPassWord As String sPassWord = InputBox("Paßwort eingeben:", , "MeinPaßwort") Select Case sPassWord Case "MeinPaßwort" With Worksheets("Geheim") .Visible = True .Select End With Case "" Case Else Beep MsgBox "War wohl nix!" End Select End Sub Sub Verbergen() Worksheets("Geheim").Visible = xlVeryHidden End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eine Serie von bestehenden Arbeitsblättern auswechseln

Problem: Wie kann ich eine Serie von bestehenden Arbeitsblättern gegen neue austauschen, wobei von den alten jeweils nur die Werte, keine Formeln, übernommen werden sollen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub WerteKopieren()
    Dim wks As Worksheet
    Dim iWks As Integer, iCounter As Integer
    Dim sWks As String
    Application.ScreenUpdating = False
    iWks = Worksheets.Count
    For iCounter = 2 To iWks
        Worksheets(2).Copy after:=Worksheets(Worksheets.Count)
        sWks = Worksheets(2).Name
        Worksheets(2).Name = "Fix&Foxi"
        ActiveSheet.Name = sWks
        Cells.Copy
        Cells.PasteSpecial xlPasteValues
        Application.CutCopyMode = False
        Range("A1").Select
        Application.DisplayAlerts = False
        Worksheets(2).Delete
        Application.DisplayAlerts = True
    Next iCounter
    Worksheets(2).Select
    Application.StatusBar = False
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Numerisch benanntes Arbeitsblatt 1+X aufrufen

Problem:	Ich habe Arbeitsblätter numerisch benannt. Wie kann ich die Zelle A1 im abgefragten Arbeitsblatt aufrufen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub WertAusFeld() Dim swks As String swks = InputBox(_ prompt:="Variable eingeben:", _ Default:="2") MsgBox Worksheets(swks).Range("A1").Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellen hinzufügen und mit Namen aus einer Liste belegen

Problem:	Wie kann ich Tabellenblätter hinzufügen und sie mit den in Spalte A stehenden Namen belegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub TabNamenFestlegen() Dim wks As Worksheet Dim iRow As Integer Set wks = ActiveSheet iRow = 1 Do Until IsEmpty(wks.Cells(iRow, 1)) Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Name = wks.Cells(iRow, 1).Value iRow = iRow + 1 Loop Worksheets(1).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eigenschaftsfenster schließen

Problem:	Wie kann ich über VBA-Code das Eigenschaftsfenster schließen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdClose_Click() On Error Resume Next AppActivate "Eigenschaften" If Err > 0 Then Exit Sub Application.SendKeys "%{F4}" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe durchblättern

Problem:	Wie kann ich die Tabellenblätter einer Arbeitsmappe nacheinander auswählen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub VorigesBlatt() If ActiveSheet.Index = Worksheets.Count Then Worksheets(1).Select Else Worksheets(ActiveSheet.Index + 1).Select End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausgewählte Arbeitsblätter auslesen

Problem:	Wie kann ich feststellen lassen, welche Arbeitsblätter der Anwender ausgewählt hat?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Selektiert() Dim wks As Worksheet For Each wks In ActiveWindow.SelectedSheets MsgBox wks.Name Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zum nächsten Blatt wechseln

Problem:	Wie kann ich über einen VBA-Befehl zum jeweils nächsten Arbeitsblatt wechseln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NaechstesBlatt() If ActiveSheet.Index = Worksheets.Count Then Worksheets(1).Select Else ActiveSheet.Next.Select End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

40 neue Arbeitsblätter hinzufügen

Problem:	Wie kann ich der aktiven Arbeitsmappe 40 neue Arbeitsblätter hinzufügen und diese von 1 bis 40 benennen?
Lösung:	Nachfolgendes Makro bei XL5/7 in ein englischsprachiges, bei XL8 ein allgemeines Modul einfügen und starten.
VBA-Code:	<pre>StandardModule: basMain Sub NeueTabellen() Dim iCounter As Integer For iCounter = 1 To 40 Worksheets.Add.Move after:=Worksheets(Worksheets.Count) ActiveSheet.Name = "Tab" & iCounter Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblatt 40 mal kopieren

Problem:	Wie kann ich das aktive Arbeitsblatt 40 mal in die gleiche Arbeitsmappe kopieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Copy40() Dim wks As Worksheet Dim iCounter As Integer Application.ScreenUpdating = False Set wks = ActiveSheet For iCounter = 1 To 40 wks.Copy after:=Worksheets(Worksheets.Count) ActiveSheet.Name = iCounter Next iCounter Worksheets(1).Select Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattnamen nach Datum

Problem:	Wie kann ich für den aktuellen Monat Tages-Tabellenblätter mit dem jeweiligen Datum als Blattname erstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CreateWks() Dim iCounter As Integer, iDays As Integer Application.ScreenUpdating = False iDays = Day(DateSerial(Year(Date), Month(Date) + 1, 0)) For iCounter = 1 To iDays Worksheets.Add.Move after:=Worksheets(Worksheets.Count) ActiveSheet.Name = Format(_ DateSerial(Year(Date), Month(Date), iCounter), _ Format:="dd.mm.yy") Next iCounter Worksheets(1).Select Application.ScreenUpdating = True End Sub</pre>



1000 Arbeitsblätter aus 1000 Arbeitsmappen einlesen

Problem:	Wie kann ich Daten aus allen Arbeitsmappen eines Verzeichnisses in diese Arbeitsmappe kopieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Sub DatenSammeln() Dim wks As Worksheet Dim fs As FileSearch Dim rng As Range Dim iCounter As Integer, iRow As Integer Dim sMsg As String, sDir As String Application.ScreenUpdating = False Application.EnableEvents = False On Error GoTo ERRORHANDLER sMsg = "Wählen Sie bitte einen Ordner aus:" sDir = GetDirectory(sMsg) If sDir = "" Then Exit Sub Set wks = ActiveSheet iRow = 3 Set fs = Application.FileSearch With fs .LookIn = sDir .FileType = msoFileTypeExcelWorkbooks .Execute For iCounter = 1 To .FoundFiles.Count Workbooks.Open _ FileName:=.FoundFiles(iCounter), _ updateLinks:=False wks.Cells(iRow - 2, 1).Value = ActiveWorkbook.Name & ":" Set rng = RealLastCell(ActiveSheet) Set rng = Range(Cells(1, 1), rng) rng.Copy wks.Cells(iRow, 1) iRow = iRow + rng.Rows.Count + 3 Application.CutCopyMode = False ActiveWorkbook.Close saveChanges:=False Next iCounter End With ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = True End Sub StandardModule: basFunctions Public Type BROWSEINFO hOwner As Long pidlRoot As Long pszDisplayName As String lpszTitle As String ulFlags As Long lpfn As Long lParam As Long iImage As Long End Type Declare Function SHGetPathFromIDList Lib "shell32.dll" _ Alias "SHGetPathFromIDListA" (ByVal pidl As Long, _ ByVal pszPath As String) As Long Declare Function SHBrowseForFolder Lib "shell32.dll" _ Alias "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long Function GetDirectory(Optional msg) As String Dim bInfo As BROWSEINFO Dim Path As String Dim r As Long, x As Long, pos As Integer bInfo.pidlRoot = 0& If IsMissing(msg) Then bInfo.lpszTitle = "Wählen Sie bitte einen Ordner aus." Else bInfo.lpszTitle = msg End If bInfo.ulFlags = &H1 x = SHBrowseForFolder(bInfo) Path = Space\$(512) r = SHGetPathFromIDList(ByVal x, ByVal Path) If r Then pos = InStr(Path, Chr\$(0)) GetDirectory = Left(Path, pos - 1) Else GetDirectory = "" End If End Function Function RealLastCell(TheSheet As Worksheet) As Range Dim ExcelLastCell As Range Dim Row%, Col%, LastRowWithData%, LastColWithData% Application.ScreenUpdating = False Set ExcelLastCell = TheSheet.Cells.SpecialCells(xlLastCell) LastRowWithData = ExcelLastCell.Row Row = ExcelLastCell.Row Do While Application.CountA(TheSheet.Rows(Row)) = 0 And Row <> 1 Row = Row - 1 Loop LastRowWithData = Row LastColWithData = ExcelLastCell.Column Col = ExcelLastCell.Column Do While Application.CountA(TheSheet.Columns(Col)) = 0 And Col <> 1 Col = Col - 1 Loop LastColWithData = Col Set RealLastCell = TheSheet.Cells(Row, Col) End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblatt auswählen

Problem: Ich möchte den Teil eines Tabellenblattnamens in eine InputBox eingeben und das erste Tabellenblatt, daß diesem Kriterium entspricht, auswählen lassen. Wie mache ich das?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub TabSelect()
    Dim wks As Worksheet
    Dim strWks As String
    strWks = InputBox("Bitte Tabellenname angeben:")
    If strWks = "" Then Exit Sub
    For Each wks In Worksheets
        If InStr(UCase(wks.Name), UCase(strWks)) > 0 Then
            wks.Select
            Exit Sub
        End If
    Next wks
    Beep
    MsgBox "Kein Blatt gefunden!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblätter benennen

Problem:	Wie kann ich ein neues Tabellenblatt hinzufügen und es mit der nächsten fortlaufenden Nummer benennen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NewSheet() Dim sName As String sName = Format(Right(Worksheets(2).Name, 3) + 1, "000") sName = Left(Worksheets(2).Name, 3) & sName Worksheets.Add.Move after:=Worksheets(Worksheets.Count) ActiveSheet.Name = sName End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattname ermitteln

Problem:	Wie kann ich ermitteln, ob eine Blattname bereits vorhanden ist?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AbfrageTabellenblatt() Dim swks As String swks = InputBox(prompt:="Gesuchtes Blatt:") If swks = "" Then Exit Sub On Error GoTo ErrorHandler Worksheets(swks).Select End ErrorHandler: MsgBox "Blatt " & swks & " nicht vorhanden!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benennen von Tabellenblättern mit Monatsnamen

Problem:	Wie kann ich eine neue Arbeitsmappe mit 12 Monatsblättern erstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MonthSheets() Dim iCounter As Integer Application.ScreenUpdating = False Workbooks.Add 1 Worksheets.Add Count:=11 For iCounter = 1 To 12 Worksheets(iCounter).Name = _ Format(DateSerial(1, iCounter, 1), "mmmm") Next iCounter Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die "personl"-Arbeitsmappe ausgeblendet speichern

Problem: Wie kann ich die sich im XLStart-Verzeichnis befindende "personl"-Arbeitsmappe so speichern, daß sie beim nächsten Excel-Start ausgeblendet ist?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Speichern()
    With Workbooks("Personl.xls")
        .Windows(1).Visible = False
        .Save
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus anderen Arbeitsmappen importieren

Problem:	Über einen Dialog wird eine Quelldatei ausgewählt. In das Import-Blatt werden die Werte der in Spalte A genannten Zellen aus dem Blatt Tabelle1 der Quelldatei importiert.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CopyValues() Dim vFile As Variant Dim iRow As Integer Dim sPath As String, sFormula As String, sMem As String, sRange As String sPath = "c:\temp" sMem = CurDir ChDrive Left(sPath, 1) ChDir sPath vFile = Application.GetOpenFilename("Excel-Arbeitsmappen (*.xls), *.xls") If vFile = False Then Exit Sub sFormula = "=" & sPath & "\" & Dir(vFile) & "]Tabelle1'!" & Range("A1").Value With Worksheets("Import") .Range(Range("A1").Value).FormulaLocal = sFormula iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) If iRow = 2 Then sRange = Cells(iRow, 1).Value Else sRange = sRange & "," & Cells(iRow, 1).Value End If iRow = iRow + 1 Loop .Range(Range("A1").Value).Copy .Range(sRange) .UsedRange.Value = .UsedRange.Value Application.CutCopyMode = False End With ChDrive Left(sMem, 1) ChDir sMem MsgBox "Job erledigt!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neue Arbeitsmappen mit Optionsfeldern erstellen

Problem: Es sollen 3 neue Arbeitsmappen mit jeweils 3 Optionsfeldern zur Zell-Farbfestlegung erstellt werden.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub NewWkbWithBtns()
    Dim oOption As OptionButton
    Dim oVbc As Object
    Dim iWkb As Integer, iOption As Integer, iLeft As Integer
    Dim sCode As String
    Application.VBE.MainWindow.Visible = False
    Application.ScreenUpdating = False
    sCode = "Sub Farbfestlegung()" & vbCrLf
    'sCode = sCode & "    Selection Case ActiveSheet.OptionButtons(Application.Caller).Caption" &
vbLf
    'sCode = sCode & "        Case ""rot"": range(""A1"").Interior.Colorindex = vbRed" & vbCrLf
    sCode = sCode & "    Dim col as New Collection" & vbCrLf
    sCode = sCode & "    Dim sCaller as String" & vbCrLf
    sCode = sCode & "    sCaller = ActiveSheet.OptionButtons(Application.Caller).Caption" & vbCrLf
    sCode = sCode & "    col.Add ""3"", ""rot""" & vbCrLf
    sCode = sCode & "    col.Add ""6"", ""gelb""" & vbCrLf
    sCode = sCode & "    col.Add ""4"", ""grün""" & vbCrLf
    sCode = sCode & "    Range(""A1"").Interior.Colorindex = col(sCaller)" & vbCrLf
    sCode = sCode & "End Sub" & vbCrLf
    For iWkb = 1 To 3
        Workbooks.Add 1
        Rows(1).RowHeight = 50
        Columns(1).ColumnWidth = 50
        iLeft = 0
        For iOption = 1 To 3
            Set oOption = ActiveSheet.OptionButtons.Add(iLeft, 0, 60, 15)
            With oOption
                Select Case iOption
                    Case 1: .Caption = "rot"
                    Case 2: .Caption = "gelb"
                    Case 3: .Caption = "grün"
                End Select
                .OnAction = ActiveWorkbook.Name & "!Farbfestlegung"
            End With
            iLeft = iLeft + 65
        Next iOption
        Set oVbc = ActiveWorkbook.VBProject.VBComponents.Add(1)
        oVbc.CodeModule.AddFromString sCode
    Next iWkb
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei auswählen und ohne Rückfrage löschen

Problem:	Wie kann ich eine Datei über Dialog auswählen und ohne Rückfrage löschen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DateiLoeschen() Dim vFile As Variant vFile = Application.GetSaveAsFilename _ (fileFilter:="Exceldateien (*.xls), *.xls") If vFile = False Then Exit Sub Kill vFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe mit 12 Monatsblättern anlegen

Problem:	Wie kann ich eine Arbeitsmappe mit 12 Monatsblättern anlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub JahrAnlegen() Dim iMonth As Integer Application.ScreenUpdating = False Workbooks.Add 1 For iMonth = 1 To 12 Worksheets.Add.Move after:=Worksheets(Worksheets.Count) ActiveSheet.Name = Format(DateSerial(1, iMonth, 1), "mmm") Next iMonth Application.DisplayAlerts = False Worksheets(1).Delete Worksheets(1).Select Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Paßwortdatei anlegen

Problem:	Wie kann ich einem Benutzer unabhängig von den Office- und Excelpaßwörtern einen eigenen Paßwortschutz ermöglichen? Das Paßwort soll jederzeit abruf- und Eingaben sollen nachprüfbar sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim sWord As String, sPass As String sWord = GetSetting("MeinProgramm", "Einstellungen", "Kennwort") If sWord = "" Then Exit Sub sPass = InputBox(prompt:="Ihr Passwort:") If sPass = sWord Then MsgBox prompt:="Alles paletti" Else MsgBox "Nix geht mehr!" End If End Sub StandardModule: Modul1 Sub SetPassword() Dim sWord As String sWord = InputBox(prompt:="Ihr Passwort:") If sWord = "" Then Exit Sub SaveSetting _ appname:="MeinProgramm", _ section:="Einstellungen", _ key:="Kennwort", _ setting:=sWord End Sub Sub DeletePassword() DeleteSetting "MeinProgramm" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Arbeitsmappen schließen bzw. Excel beenden

Problem:	Wie kann ich alle geöffneten Arbeitsmappen ungespeichert schließen bzw. Excel beenden, ohne daß die geöffneten Arbeitsmappen gespeichert werden
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AllesBeenden1() Application.DisplayAlerts = False Workbooks.Close Application.DisplayAlerts = True End Sub Sub AllesBeenden2() Application.DisplayAlerts = False Application.Quit Application.DisplayAlerts = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auslesen der Arbeitsmappeneigenschaften

Problem:	Wie kann ich die eingebauten Dokumenten-Eigenschaften der Arbeitsmappe über VBA auslesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Eigenschaften() Dim iCounter As Integer On Error Resume Next For iCounter = 1 To 30 Cells(iCounter, 1).Value = _ ThisWorkbook.BuiltinDocumentProperties(iCounter).Name Cells(iCounter, 2).Value = _ ThisWorkbook.BuiltinDocumentProperties(iCounter) If Err > 0 Then Err.Clear Cells(iCounter, 2).Value = "k.A." End If Next iCounter On Error GoTo 0 Columns.AutoFit Columns("A:B").HorizontalAlignment = xlLeft End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

AutoFilter über mehrere Arbeitsblätter setzen

Problem:	Wie kann ich den AutoFilter in einem Zug über mehrere Arbeitsblätter setzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub AlleFiltern() Dim iWks As Integer For iWks = 2 To Worksheets.Count Worksheets(iWks).Range("A1").AutoFilter _ Field:=1, _ Criteria1:="=Zeile1*", _ Operator:=xlAnd Next iWks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellenwert als Autofilter-Kriterienwert

Problem:	Wie kann ich den Inhalt einer Zelle zum Kriterienwert des Autofilters machen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZelleAlsFilter() Columns("A:F").AutoFilter _ Field:=Range("I2").Value, _ Criteria1:=Range("I1").Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Autofilter auch mit Ziffern von Zahlen

Problem:	Der AutoFilter akzeptiert nur Teile von Zeichenfolgen, nicht von Zahlen. Kann man statt dem AutoFilter VBA einsetzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NumberFilter() Dim iRow As Integer Dim sFilter As Integer sFilter = Range("D1").Text iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) If InStr(Cells(iRow, 1).Text, sFilter) Then Rows(iRow).Hidden = True End If iRow = iRow + 1 Loop End Sub Sub ResetRows() Rows.Hidden = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textinhalte zusammenfügen

Problem:	Mit dem Shortcut Strg+a sollen die in Spalte A ausgewählten Zelleninhalte in Zelle B1 durch ein Komma getrennt verkettet werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einem ShortCut zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Verbinden() Dim rng As Range Dim sTxt As String If ActiveCell.Column <> 1 Then Exit Sub For Each rng In Selection If sTxt = "" Then sTxt = rng.Value Else sTxt = sTxt & "," & rng.Value End If Next rng Range("B1").Value = sTxt End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schlüssel-Aus- und Rückgaben protokollieren

Problem:	Schlüssel-Aus- und Rückgaben sollen auf Grundlage einer Schlüssel- und Namensdatenbank protokolliert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Eintragen() Dim iRow As Integer If WorksheetFunction.CountA(wksVerwaltung.Rows(2)) < 4 Then Beep MsgBox "Die Felder ""Schlüssel"", ""Empfänger"" und " & _ ""Ausgabe"" oder ""Rückgabe"" müssen ausgefüllt sein!" End Sub End If ActiveSheet.Calculate iRow = wksAusgaben.Cells(Rows.Count, 1).End(xlUp).Row + 1 wksAusgaben.Rows(iRow).Value = wksVerwaltung.Rows(2).Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Zeilen und Spalten außerhalb des benutzten Bereiches löschen

Problem: Alle Zeilen und Spalten unterhalb bzw. rechts von der letzten Zelle mit Wert sollen gelöscht werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub DeleteRange()  
    Dim wks As Worksheet  
    Dim rng As Range  
    Dim iCounter As Integer  
    For iCounter = ActiveSheet.Index + 1 To Worksheets.Count  
        Set wks = Worksheets(iCounter)  
        Set rng = RealLastCell(wks)  
        wks.Rows(rng.Row + 1 & ":" & Rows.Count).Delete  
        wks.Range(wks.Cells(1, rng.Column + 1), _  
            wks.Cells(1, 256)).EntireColumn.Delete  
    Next iCounter  
End Sub  
  
Function RealLastCell(TheSheet As Worksheet) As Range  
    Dim ExcelLastCell As Range  
    Dim Row%, Col%, LastRowWithData%, LastColWithData%  
    Application.ScreenUpdating = False  
    Set ExcelLastCell = TheSheet.Cells.SpecialCells(xlLastCell)  
    LastRowWithData = ExcelLastCell.Row  
    Row = ExcelLastCell.Row  
    Do While Application.CountA(TheSheet.Rows(Row)) = 0 And Row <> 1  
        Row = Row - 1  
    Loop  
    LastRowWithData = Row  
    LastColWithData = ExcelLastCell.Column  
    Col = ExcelLastCell.Column  
    Do While Application.CountA(TheSheet.Columns(Col)) = 0 And Col <> 1  
        Col = Col - 1  
    Loop  
    LastColWithData = Col  
    Set RealLastCell = TheSheet.Cells(Row, Col)  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Entfernen der letzten beiden Zeichen in Listenelementen

Problem:	Von den Werten in Spalte A sollen die letzten beiden Stellen entfernt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub FirstThree() Dim iRow As Integer iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) Cells(iRow, 1).Value = Left(Cells(iRow, 1).Value, 3) iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalten mit gleichen Spaltenköpfen zusammenfassen

Problem:	Spalten mit gleichen Kopfzeilen sollen zusammengefaßt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Zusammenfassen() Dim iRow As Integer, iCol As Integer, iAct As Integer Dim iCountC As Integer, iRowLT As Integer, iRowLS As Integer iCol = 1 Do Until IsEmpty(Cells(1, iCol)) For iAct = iCol + 1 To WorksheetFunction.CountA(Rows(1)) If Cells(1, iCol).Value = Cells(1, iAct).Value And _ WorksheetFunction.CountA(Columns(iCol)) > 1 Then iRowLS = Cells(Rows.Count, iAct).End(xlUp).Row iRowLT = Cells(Rows.Count, iCol).End(xlUp).Row + 1 Range(Cells(2, iAct), Cells(iRowLS, iAct)).Cut _ Cells(iRowLT, iCol) End If Next iAct iCol = iCol + 1 Loop Application.CutCopyMode = False For iCol = Range("IV1").End(xlToLeft).Column To 1 Step -1 If WorksheetFunction.CountA(Columns(iCol)) = 1 Then Columns(iCol).Delete End If Next iCol End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Überschriften gemäß Indizes in Tabelle einfügen

Problem: Die Überschriften aus dem Blatt "Titel" sollen gemäß der Index-Spalte vor jedem Zeilenblock eingefügt werden.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code: StandardModule: Modul1

```
Sub InsertTitel()  
    Dim rng As Range  
    Dim var As Variant  
    Dim iRow As Integer, iRowL As Integer  
    Set rng = Worksheets("Titel").Range("A1").CurrentRegion  
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row  
    For iRow = iRowL To 2 Step -1  
        If Cells(iRow, 1).Value <> Cells(iRow - 1, 1).Value Then  
            Rows(iRow).Insert  
            Rows(iRow).Font.Bold = True  
            var = Application.Match(Cells(iRow + 1, 1).Value, rng.Columns(1), 0)  
            Cells(iRow, 1).Value = Cells(iRow + 1, 1).Value  
            If Not IsError(var) Then  
                Cells(iRow, 2).Value = rng.Cells(var, 2).Value  
            Else  
                Cells(iRow, 2).Value = "Kein Titel"  
            End If  
        End If  
    Next iRow  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Rechtschreibprüfung und Fehlerausgabe

Problem:	Die Einträge auf Spalte A sollen auf ihre Rechtschreibkorrektheit geprüft werden. Fehler sollen in Spalte B angezeigt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub StartSpelling() Dim iRow As Integer On Error GoTo ERRORHANDLER For iRow = 1 To WorksheetFunction.CountA(Columns(1)) If Application.CheckSpelling(_ Cells(iRow, 1).Value, , True) = False Then Cells(iRow, 2).Value = False Else Cells(iRow, 2).Value = True End If Next iRow Exit Sub ERRORHANDLER: Beep MsgBox _ prompt:="Die Rechtschreibprüfung ist nicht installiert!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Begrenzung der Einträge pro Zeile

Problem:	Die Anzahl der Einträge pro Zeile ist bis zu Zeile 6 auf maximal 1 Eintrag zu begrenzen.
Lösung:	Lösung über Menü DATEN / GÜLTIGKEIT / BENUTZERDEFINIERT. Formel: =ANZAHL2(1:1)=1
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBA-Aktion über "Rückgängig"-Schaltfläche zurücksetzen

Problem:	Auf Schaltflächendruck sollen die Werte im ausgewählten Bereich auf zwei Nachkommastellen gerundet werden. Die Aktion ist bei Betätigung der "Rückgängig"-Schaltfläche wieder rückgängig zu machen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub HinUndZurueck() Dim rng As Range For Each rng In Selection Worksheets("Dummy").Range(rng.Address).Value = rng.Value If IsNumeric(rng.Value) And Not IsEmpty(rng) Then rng.Value = WorksheetFunction.Round(rng.Value, 2) End If Next rng Application.OnUndo "Werte wiederherstellen", "Zurueck" End Sub Sub Zurueck() Dim rng As Range For Each rng In Selection rng.Value = Worksheets("Dummy").Range(rng.Address).Value Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle nach Gruppen aufteilen

Problem: Die Tabelle soll gruppenweise nach Kriterium Spalte A in neue Tabellen aufgeteilt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Kopieren()
    Dim wks As Worksheet
    Dim iRow As Integer, iStart As Integer
    Application.ScreenUpdating = False
    Set wks = ActiveSheet
    iRow = 2
    iStart = iRow
    Do Until IsEmpty(wks.Cells(iRow, 1))
        If wks.Cells(iRow, 1) <> wks.Cells(iRow + 1, 1) Then
            Worksheets.Add after:=Worksheets(Worksheets.Count)
            ActiveSheet.Name = wks.Cells(iRow, 1)
            wks.Rows(iStart & ":" & iRow).Copy Range("A1")
            iStart = iRow + 1
        End If
        iRow = iRow + 1
    Loop
    Worksheets(1).Select
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makroaktion über die UnDo-Schaltfläche zurücknehmen

Problem: Auf Schaltflächendruck sollen Werte in Spalte A eingetragen werden. Diese Aktion soll über Menü BEARBEITEN / RÜCKGÄNGIG wieder aufgehoben werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub UndoTest()
    Application.OnUndo _
        Text:="Eintragung löschen", _
        Procedure:=ThisWorkbook.Name & "!UndoEin"
End Sub

Sub UndoEin()
    Range("A1:A10").ClearContents
End Sub

Sub Eintragen()
    Dim iCounter As Integer
    For iCounter = 1 To 10
        Cells(iCounter, 1).Value = iCounter
    Next iCounter
    Call UndoTest
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten aus Eingabemaske in Monatstabelle eintragen

Problem: Die Daten aus dem Bereich B6:C6 sollen in der Monatstabelle gem. Datum in Zelle F16 eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Eintragen()
    Dim iRow As Integer
    With Worksheets(Format(Range("C7").Value, "mmm"))
        iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
        .Cells(iRow, 1) = Range("B6").Value
        .Cells(iRow, 2) = Range("C6").Value
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makro mit Escape-Taste ohne Fehlermeldung beenden

Problem:	Ein Makro soll mit der Escape-Taste abgebrochen werden können. Danach soll eine Bestätigungsmeldung erfolgen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MakroUnterbrechen() Dim lRow As Long On Error GoTo ErrorHandler Application.EnableCancelKey = xlErrorHandler For lRow = 1 To Rows.Count Cells(lRow, 1).Value = lRow Next lRow End ErrorHandler: MsgBox "Alles klar, mache Schluß!" Columns(1).ClearContents End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten aus einer Adressmaske in Adressdatenbank

Problem: Die Daten aus dem Bereich C3:C6 sollen in das Blatt Adressen in die nächste freie Zeile eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub InsertData()
    Dim iRow As Integer, iCounter As Integer
    If WorksheetFunction.CountA(Range("C2:C6")) = 0 Then
        MsgBox "Keine Daten vorhanden!"
        Exit Sub
    End If
    With Worksheets("Adressen")
        iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
        For iCounter = 1 To 5
            .Cells(iRow, iCounter) = Cells(iCounter + 1, 3)
        Next iCounter
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit ShortCut STRG+END zur tatsächlich letzten Zelle

Problem: Mit dem Excel-ShortCut STRG+ENDE springt der Cursor nicht zuverlässig zur letztenZelle mit Inhalt des Blattes. Wie kann ich das ändern?

Lösung: Geben Sie den nachfolgenden Code in die genannten Module ein

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Application.OnKey "^{end}"
End Sub

Private Sub Workbook_Open()
    Application.OnKey "^{end}", "LastCell"
End Sub

StandardModule: basMain

Sub LastCell()
    RealLastCell(ActiveSheet).Select
End Sub

Function RealLastCell(TheSheet As Worksheet) As Range
    Dim ExcelLastCell As Range
    Dim Row%, Col%, LastRowWithData%, LastColWithData%
    Application.ScreenUpdating = False
    Set ExcelLastCell = TheSheet.Cells.SpecialCells(xlLastCell)
    LastRowWithData = ExcelLastCell.Row
    Row = ExcelLastCell.Row
    Do While Application.CountA(TheSheet.Rows(Row)) = 0 And Row <> 1
        Row = Row - 1
    Loop
    LastRowWithData = Row
    LastColWithData = ExcelLastCell.Column
    Col = ExcelLastCell.Column
    Do While Application.CountA(TheSheet.Columns(Col)) = 0 And Col <> 1
        Col = Col - 1
    Loop
    LastColWithData = Col
    Set RealLastCell = TheSheet.Cells(Row, Col)
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In mehrere Spalten doppelte Inhalte entfernen

Problem:	Wie kann ich mehrspaltig jeden Eintrag entfernen, der dem in der oberen Zelle entspricht?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DoppelteLoeschen() Dim iRow As Integer, iCol As Integer For iRow = Range("A1").CurrentRegion.Rows.Count To 2 Step -1 For iCol = 1 To Range("A1").CurrentRegion.Columns.Count If Cells(iRow, iCol) = Cells(iRow - 1, iCol) Then Cells(iRow, iCol).ClearContents End If Next iCol Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus A1 in mehrere Zellen übernehmen

Problem:	Wie kann ich einen in Zelle A1 stehenden Wert per VBA in mehrere andere Zellen übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZellenFuellen() Range("B2,B4,C9,F13") = Range("A1") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle #DIV/0!-Fehlerwerte in Tabelle durch 0 ersetzen

Problem:	Wie kann ich alle Fehlerwerte in einer Tabelle über VBA durch den Wert 0 ersetzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FormelnKorrigieren() Range("A1").CurrentRegion.SpecialCells(_ Type:=xlCellTypeFormulas, Value:=xlErrors).Value = 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bearbeitungszeit feststellen

Problem:	Wie kann ich die Dauer der Laufzeit eines Makros feststellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub TimerTest() Dim dteStart As Date, dteEnde As Date Dim iCounter As Single Application.ScreenUpdating = False dteStart = Timer For iCounter = 1 To 5000 If iCounter Mod 1000 = 0 Then Application.StatusBar = _ "Besuche Zeile " & iCounter & "..." End If Cells(iCounter, 1).Select Next iCounter Range("A1").Select Application.StatusBar = False dteEnde = Timer MsgBox "Bearbeitungszeit: " & Format _ (dteEnde - dteStart, "0.00") & " Sekunden..." End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In einem zweiten Tabellenblatt vorgegebene Zeilen beschriften

Problem: Wie kann ich über InputBoxes Von-/Bis-Zeilennummern festlegen und die entsprechenden Zeilen in einem zweiten Arbeitsblatt beschriften?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub cmdStart_Click()
    Dim varStart As Variant, varEnde As Variant
    Dim iCounter As Integer
    varStart = InputBox( _
        prompt:="Startwert:", _
        Default:="5")
    varEnde = InputBox( _
        prompt:="Endwert:", _
        Default:="15")
    For iCounter = varStart To varEnde
        Worksheets("Tabelle2").Cells(iCounter, 1) = _
            iCounter & ". Zeile"
    Next iCounter
    Worksheets("Tabelle2").Select
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in Spalte A in Abhängigkeit des Werte in Spalte B eintragen

Problem: Wie kann ich in Spalte A Werte in Abhängigkeit der Werte in Spalte B eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Eintragen()
    Dim rng As Range
    Set rng = Range("B1")
    Do Until IsEmpty(rng)
        If rng = 2 Then rng.Offset(0, -1) = "Ok"
        Set rng = rng.Offset(1, 0)
    Loop
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Von Eingabe- zu Eingabefeld springen

Problem:	Wie kann ich in einem Tabellenblattformular nur von Eingabe- zu Eingabefeld springen, ohne die übrigen Zellen zu berühren?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus einem nicht zusammenhängendem Bereich auslesen

Problem: Wie kann ich die Werte aller Zellen eines nicht zusammenhängenden Bereiches über VBA auslesen und in einer MsgBox anzeigen lassen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub cmdWerteAuslesen_Click()
    Dim rng As Range, rngAct As Range
    Set rng = Range("A1:A4, C1:C4, E1:E4")
    For Each rngAct In rng.Cells
        If IsEmpty(rngAct) Then
            MsgBox "Zelle " & rngAct.Address(False, False) & _
                " ist leer!"
        Else
            MsgBox "Zelle " & rngAct.Address(False, False) & _
                " hat den Wert " & rngAct
        End If
    Next rngAct
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Zeilen außer der mit dem höchsten Wert löschen

Problem: Wie kann ich alle Zeilen außer der, die den höchsten Wert der Tabelle beinhaltet, löschen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Filtern()
    Dim rng As Range, rngMax As Range
    Set rng = ActiveSheet.UsedRange
    Set rngMax = rng.Find(WorksheetFunction.Max(rng))
    Rows(rngMax.Row + 1 & ":" & Rows.Count).Delete
    If rngMax.Row > 2 Then Rows("2:" & rngMax.Row - 1).Delete
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelte Datensätze und Leerzeilen löschen

Problem: Wie kann ich aus einer Tabelle alle doppelt vorkommenden Datensätze und leere Zeilen löschen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Filtern()  
    With ActiveSheet.UsedRange  
        .Sort Key1:=Range("A2"), Order1:=xlDescending, _  
            Header:=xlGuess, OrderCustom:=1, _  
            MatchCase:=False, Orientation:=xlTopToBottom  
        .AdvancedFilter Action:=xlFilterCopy, _  
            CopyToRange:=Cells(1, .Columns.Count + 1), _  
            Unique:=True  
        .EntireColumn.Delete  
    End With  
    Columns.AutoFit  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Von der Fundstelle eines Zellwertes abhängige Daten eintragen

Problem:	Wie kann ich den Wert einer Zelle in einer Tabelle suchen und die Kopfzeile und den Wert der Vorgängerzelle in einem zweiten Tabellenblatt eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BegriffSuchen() Dim rng As Range Dim sBegriff As String sBegriff = Range("B20") Set rng = Range("A1").CurrentRegion.Find(sBegriff, lookat:=xlWhole) If rng Is Nothing Then Beep MsgBox "Nein, Begriff wurde nicht gefunden!" Exit Sub End If With Worksheets("Tabelle2") .Range("A1") = "Kopfzeile:" .Range("A2") = Cells(1, rng.Column) .Range("B1") = "Vorgängerzelle:" If rng.Row = 2 Then If rng.Column = 1 Then Beep MsgBox "Es gibt keine Vorgängerzelle!" Exit Sub End If .Range("B2") = rng.Offset(0, -1) ElseIf rng.Column = 1 Then .Range("B2") = rng.Offset(-1, Range("A1"). _ .CurrentRegion.Columns.Count - 1) Else .Range("B2") = rng.Offset(0, -1) End If .Columns.AutoFit End With Worksheets("Tabelle2").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle leeren Zeilen und Spalten entfernen

Problem:	Wie kann ich alle leeren Zeilen und Spalten eines Arbeitsblattes löschen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub EmptysDelete() Cells.SpecialCells(xlCellTypeBlanks).Delete xlUp End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Leere Zeilen und Spalten in ausgewähltem Bereich löschen

Problem:	Wie kann ich alle leeren Zeilen und Spalten in einem ausgewählten Bereich über VBA-Code löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CommandButton1_Click() Dim iCounter As Integer If Selection.Cells.Count = 1 Then Beep MsgBox "Sie müssen einen Bereich auswählen!" Exit Sub End If For iCounter = Selection.Columns.Count To 1 Step -1 If WorksheetFunction.CountA(Columns(iCounter)) = 0 Then Selection.Columns(iCounter).Delete End If Next iCounter For iCounter = Selection.Rows.Count To 1 Step -1 If WorksheetFunction.CountA(Rows(iCounter)) = 0 Then Selection.Rows(iCounter).Delete End If Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Formeln und Links in Werte wandeln

Problem:	Wie kann ich alle Formeln und Links aller Tabellenblätter einer Arbeitsmappe in Werte umwandeln und die Arbeitsmappe dann unter einem neuen Namen speichern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SaveValues() Dim wks As Worksheet Dim sFile As String Application.ScreenUpdating = False sFile = Application.DefaultFilePath & "\Test.xls" sFile = InputBox(_ prompt:="Name der neuen Datei:", _ Default:=sFile) If sFile = "" Then Exit Sub For Each wks In Worksheets wks.UsedRange.Cells.Copy wks.UsedRange.PasteSpecial Paste:=xlValues Next wks ActiveWorkbook.SaveAs sFile Application.CutCopyMode = False Range("A1").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wassergeldabrechnung automatisieren

Problem:	Wie kann ich für eine Gruppe von Verbrauchern die monatliche Wassergeldabrechnung ohne VBA-Einsatz automatisieren?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summenbezüge korrigieren

Problem: Beim Löschen von Zeilen kommt es zu Fehlermeldungen bei Summenbezügen. Kann ich diese Fehler korrigieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub NeuSummieren()
    Dim iRowL As Integer, iRow As Integer, iStart As Integer
    Dim sFormula As String
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row - 1
    iStart = 1
    For iRow = 1 To iRowL
        If IsError(Cells(iRow, 2)) Then
            Cells(iRow, 2).Formula = _
                "=SUM(B" & iStart & ":B" & iRow - 1 & ")"
            iStart = iRow + 1
            sFormula = sFormula & Cells(iRow, 2).Address & ","
        End If
    Next iRow
    sFormula = Left(sFormula, Len(sFormula) - 1)
    Cells(iRowL + 1, 2).Formula = "=SUM(" & sFormula & ")"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle überflüssigen Leerzeichen im Arbeitsblatt löschen

Problem: Wie kann ich auf einen Schlag alle überflüssigen Leerzeichen in einem Arbeitsblatt löschen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Glaetten()
    With ActiveSheet.UsedRange
        .Cells.Value = Application.Trim(.Cells.Value)
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile löschen, wenn ein bestimmter Wert vorkommt

Problem: Wie kann ich alle Zeilen löschen lassen, in denen der Wert "g" vorkommt?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub DeleteIfG()
    Dim var As Variant
    Dim iRow As Integer, iRowL As Integer
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row
    For iRow = iRowL To 1 Step -1
        var = Application.Match("*g*", Rows(iRow), 0)
        If Not IsError(var) Then
            Rows(iRow).Delete
        End If
    Next iRow
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle nichtnumerischen Werte auf Null setzen

Problem:	Wie kann ich in einem ausgewählten Bereich für alle nichtnumerischen Werte eine Null eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Pruefen() Dim rng As Range For Each rng In Range("A1").CurrentRegion.Cells If Not IsNumeric(rng.Value) Then rng.Value = 0 End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spielpaarungen zusammenstellen

Problem:	Wie kann ich für 6 Tennisspieler die Paarungen über VBA nach dem Zufallsprinzip zusammenstellen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Zufall() Dim rngA As Range, rngB As Range, rngC As Range Dim iZufall As Integer Randomize Set rngA = Range("B1:C3") rngA.ClearContents For Each rngB In rngA.Cells Set rngC = Range("A1") While Not rngC Is Nothing iZufall = Int((6 * Rnd) + 1) Set rngC = rngA.Find(Cells(iZufall, 1)) Wend rngB.Value = Cells(iZufall, 1) Next rngB End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fehler entfernen

Problem:	Wie kann ich alle Formeln, die eine Fehlermeldung ergeben, aus einem Tabellenblatt entfernen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DelErrors() Cells.SpecialCells(xlFormulas, 16).ClearContents End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spaltennamen bestimmen

Problem:	Wie kann ich über eine VBA-Prozedur den Namen der Spalte (A, B, C) bestimmen, in der sich die aktive Zelle befindet?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ConvertColumn() Dim sAddress As String, sColumn As String sAddress = ActiveCell.Address sAddress = Mid(sAddress, InStr(sAddress, "\$") + 1, _ InStr(2, sAddress, "\$") - 2) MsgBox sAddress End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelte Datensätze löschen

Problem:	Wie kann ich alle in Spalte A doppelt vorkommenden Datensätze löschen?
Lösung:	über Menü DATEN / FILTER / Spezialfilter / Kästchen "Keine Duplikate" aktivieren / Ausgabebereich angeben / Quellspalten löschen. te" aktivieren / Ausgabebereich angeben / Quellspalten löschen, oder über VBA mit nachfolgendem Code.
VBA-Code:	<pre>StandardModule: basMain Sub SetFilter() Application.ScreenUpdating = False Range("A1").CurrentRegion.AdvancedFilter _ Action:=xlFilterCopy, CopyToRange:= _ Range("B1"), Unique:=True Columns(1).EntireColumn.Delete Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Primzahlen 1 bis 10.000.000 eintragen

Problem:	In die Tabelle werden die Primzahlen zwischen 1 und 10.000.000 eingetragen. Die Routine kann mit ESCAPE abgebrochen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Option Base 1 Sub PrimeNumbers() Dim VNum As Long Dim VMaxNum As Long Dim i As Long Dim VMaxInd As Long, VMinNum As Long, lRow As Long Dim Prime() As Variant Dim VCtrl As Boolean Application.EnableCancelKey = xlErrorHandler On Error GoTo ERRORHANDLER Cells.ClearContents Application.ScreenUpdating = False VMaxInd = 1 ReDim Prime(VMaxInd) Prime(1) = 2 VMinNum = 1 VMaxNum = 10000000 For VNum = 3 To VMaxNum Step 2 VCtrl = True For i = 1 To VMaxInd If VNum Mod Prime(i) = 0 Then VCtrl = False Exit For ElseIf (Prime(i)) ^ 2 > VNum Then ' <--- this is the trick! Exit For End If Next i If VCtrl = True Then VMaxInd = VMaxInd + 1 ReDim Preserve Prime(VMaxInd) Prime(VMaxInd) = VNum lRow = lRow + 1 Cells(lRow) = VNum If lRow Mod 1000 = 0 Then Application.StatusBar = "Trage die " & lRow & ". Primzahl ein (abbrechen mit ESC)..." End If End If Next VNum ERRORHANDLER: Application.StatusBar = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In einer Arbeitsmappe mit Monatsblättern zum aktuellen Datum

Problem:	Auf Schaltflächenklick soll der aktuelle Tag in der entsprechenden Monatstabelle ausgewählt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CreateMonths() Dim lDay As Long Dim iWks As Integer, iDay As Integer For iWks = 1 To 12 Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Name = Format(DateSerial(1, iWks, 1), "mmm") For lDay = DateSerial(Year(Date), iWks, 1) To DateSerial(Year(Date), iWks + 1, 0) iDay = iDay + 1 Cells(iDay, 1).Value = DateSerial(Year(Date), iWks, iDay) Next lDay iDay = 0 Next iWks Worksheets(1).Select End Sub Sub GotoToday() Dim iRow As Integer Worksheets(Month(Date) + 1).Select iRow = WorksheetFunction.Match(CDbl(Date), Columns(1), 0) Cells(iRow, 1).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ergebnisse in Fussballtabelle eintragen und diese aktualisieren

Problem:	Ergebnisse eines Spieltags sollen in eine auf einem anderen Blatt befindliche Tabelle eingetragen werden, wobei die Tabell neu zu sortieren ist.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>ClassModule: Tabelle5 StandardModule: Modul1 Sub Tabelle() Dim E As Range, T As Range, aCell As Range, bCell As Range Dim A As Range Dim i%, y%, n% With Worksheets("Spieltag") y = .Cells(Rows.Count, 1).End(xlUp).Row Set E = .Range(Cells(2, 1), Cells(y, 4)) End With With Worksheets("Tabelle") y = .Cells(Rows.Count, 1).End(xlUp).Row Set T = .Range(.Cells(1, 2), .Cells(y, 2)) Set A = .Range(.Cells(3, 1), .Cells(y, 16)) End With For i = 1 To E.Rows.Count Set aCell = T.Find(E.Cells(i, 1)) aCell.Offset(0, 1) = aCell.Offset(0, 1) + 1 Set bCell = T.Find(E.Cells(i, 2)) bCell.Offset(0, 1) = bCell.Offset(0, 1) + 1 If E.Cells(i, 3) > E.Cells(i, 4) Then aCell.Offset(0, 2) = aCell.Offset(0, 2) + 3 aCell.Offset(0, 6) = aCell.Offset(0, 6) + 1 aCell.Offset(0, 9) = aCell.Offset(0, 9) + 1 bCell.Offset(0, 14) = bCell.Offset(0, 14) + 1 ElseIf E.Cells(i, 3) < E.Cells(i, 4) Then bCell.Offset(0, 2) = bCell.Offset(0, 2) + 3 bCell.Offset(0, 6) = bCell.Offset(0, 6) + 1 bCell.Offset(0, 12) = bCell.Offset(0, 12) + 1 aCell.Offset(0, 11) = aCell.Offset(0, 11) + 1 Else aCell.Offset(0, 2) = aCell.Offset(0, 2) + 1 bCell.Offset(0, 2) = bCell.Offset(0, 2) + 1 aCell.Offset(0, 7) = aCell.Offset(0, 7) + 1 aCell.Offset(0, 10) = aCell.Offset(0, 10) + 1 bCell.Offset(0, 7) = bCell.Offset(0, 7) + 1 bCell.Offset(0, 13) = bCell.Offset(0, 13) + 1 End If aCell.Offset(0, 3) = aCell.Offset(0, 3) + E.Cells(i, 3) aCell.Offset(0, 4) = aCell.Offset(0, 4) + E.Cells(i, 4) bCell.Offset(0, 3) = bCell.Offset(0, 3) + E.Cells(i, 4) bCell.Offset(0, 4) = bCell.Offset(0, 4) + E.Cells(i, 3) Next i A.Sort Key1:=A.Range("D1"), Order1:=xlDescending, Key2:= _ A.Range("G1"), Order2:=xlDescending, Key3:=A.Range("E1"), Order3 _ :=xlDescending, Header:=xlGuess, OrderCustom:=1, MatchCase:= _ False, Orientation:=xlTopToBottom Worksheets("Tabelle").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zu- und Abrechnung von Mehrwertsteuer

Problem: Aus ausgewählten Werten soll die Mehrwertsteuer wahlweise heraus- oder hinzugerechnet werden.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code: StandardModule: Modul1

```
Sub ZuNetto()  
    Const cd As Double = 1.16  
    Dim rng As Range  
    For Each rng In Selection.Cells  
        If IsEmpty(rng) = False Then  
            If IsNumeric(rng) Then  
                rng.Value = rng.Value / cd  
            End If  
        End If  
    Next rng  
End Sub  
  
Sub ZuBrutto()  
    Const cd As Double = 1.16  
    Dim rng As Range  
    For Each rng In Selection.Cells  
        If IsEmpty(rng) = False Then  
            If IsNumeric(rng) Then  
                rng.Value = rng.Value * cd  
            End If  
        End If  
    Next rng  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktivierung von Berechnungen bei gültiger Passworteingabe

Problem:	Die Berechnungen des Arbeitsblattes werden nur dann aktiviert, wenn ein gültiges Passwort eingegeben wird.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean) With ThisWorkbook .Names.Add Name:="COUNTRY", RefersToR1C1:="=3" .Names("Country").Visible = False End With End Sub Private Sub Workbook_Open() Application.EnableCancelKey = xlDisabled Dim sTxt As String sTxt = InputBox("Bitte Passwort eingeben:", , "MeinPasswort") If sTxt <> "MeinPasswort" Then Exit Sub With ThisWorkbook .Names.Add Name:="COUNTRY", RefersToR1C1:="=Tabelle1!COUNTRY" .Names("Country").Visible = False End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus Betriebseintrittsdaten Jubiläen ermitteln

Problem: Aus den Betriebseintrittsdaten in Spalte E sollen die 25- und 40jährigen Jubiläen mit einer Karenzzeit von 5 Tagen ausgelesen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Jubis()  
    Dim wks As Worksheet  
    Dim iRow As Integer, iRowT As Integer  
    Dim i40 As Integer, i25 As Integer  
    Application.ScreenUpdating = False  
    Set wks = ActiveSheet  
    iRow = 1  
    i40 = DateSerial(Year(Date) - 40, Month(Date), Day(Date))  
    i25 = DateSerial(Year(Date) - 25, Month(Date), Day(Date))  
    Worksheets.Add after:=Worksheets(Worksheets.Count)  
    With wks  
        Do Until IsEmpty(.Cells(iRow, 1))  
            If CDate(i40) - .Cells(iRow, 5) < 5 And _  
                i40 - .Cells(iRow, 5) > -5 Then  
                iRowT = iRowT + 1  
                Rows(iRowT).Value = .Rows(iRow).Value  
                Cells(iRowT, 7) = 40  
            End If  
            If i25 - .Cells(iRow, 5) < 5 And _  
                i25 - .Cells(iRow, 5) > -5 Then  
                iRowT = iRowT + 1  
                Rows(iRowT).Value = .Rows(iRow).Value  
                Cells(iRowT, 7) = 25  
            End If  
            iRow = iRow + 1  
        Loop  
    End With  
    Columns.AutoFit  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Errechneten Wert durch nächsthöheren ersetzen

Problem: Ein errechneter Wert soll durch den nächsthöheren Wert aus Spalte A ersetzt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub WertWandeln()  
    Dim dValue As Double, dStart As Double  
    Dim iRow As Integer  
    dValue = Range("A1").Value  
    MsgBox _  
        prompt:="Wert aus Zelle A1: " & dValue & "..."  
    dValue = dValue * 5  
    dStart = dValue  
    MsgBox _  
        prompt:="Wert aus Zelle A1 * 5: " & dValue & "..."  
    iRow = 1  
    Do Until IsEmpty(Cells(iRow, 1))  
        If Cells(iRow, 1).Value > dStart Then  
            If Cells(iRow, 1).Value < dValue Or dValue = dStart Then  
                dValue = Cells(iRow, 1).Value  
            End If  
        End If  
        iRow = iRow + 1  
    Loop  
    MsgBox _  
        prompt:="... ersetzt durch den nächsthöheren" & vbLf & _  
        "Wert aus der Tabelle: " & dValue  
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte jeder zweiten Spalte in zwei Stufen dividieren

Problem:	Die Werte in Spalte A:I sollen durch 100 und 1,5 dividiert und mit zwei Nachkommastellen dargestellt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NewCalc() Dim lRow As Long, lRowL As Long Dim dValue As Double Dim iCol As Integer Application.ScreenUpdating = False For iCol = 1 To 9 Step 2 Columns(iCol).NumberFormat = "#,##0.00" lRowL = Cells(Rows.Count, iCol).End(xlUp).Row For lRow = 2 To lRowL If Not IsEmpty(Cells(lRow, iCol)) Then If IsNumeric(Cells(lRow, iCol)) Then dValue = Cells(lRow, iCol).Value dValue = dValue / 100 dValue = dValue / 1.5 Cells(lRow, iCol).Value = dValue End If End If Next lRow Next iCol Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausziffern von Werten

Problem:	Die Werte in den Spalten A und B sollen in Spalte C ausgedruckt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Ausziffern() Dim iRowA As Integer, iRowB As Integer Dim iRowLA As Integer, iRowLB As Integer Dim iCounter As Integer iRowLA = Cells(Rows.Count, 1).End(xlUp).Row iRowLB = Cells(Rows.Count, 2).End(xlUp).Row For iRowA = 1 To iRowLA If Not IsEmpty(Cells(iRowA, 1)) Then For iRowB = 1 To iRowLB If Not IsEmpty(Cells(iRowB, 2)) Then If Cells(iRowB, 2) = Cells(iRowA, 1) And _ IsEmpty(Cells(iRowB, 3)) Then iCounter = iCounter + 1 Cells(iRowA, 3) = iCounter Cells(iRowB, 3) = iCounter End If End If Next iRowB End If Next iRowA End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Änderung der Kostenstelle Werte summieren

Problem:	Wenn der Wert der jeweiligen Zelle in Spalte A von dem der darüberliegenden Zelle abweicht, soll eine Addition eingefügt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Summieren() Dim rng As Range Dim dSum As Double Dim iRow As Integer Set rng = Cells(Rows.Count, 1).End(xlUp) iRow = 3 Do While rng.Row > iRow If Cells(iRow - 1, 1) <> Cells(iRow, 1) Then Rows(iRow).Insert Cells(iRow, 2) = dSum Rows(iRow + 1).Insert iRow = iRow + 2 dSum = 0 End If dSum = dSum + Cells(iRow, 2) iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte summieren, in erste freie Zeile eintragen, Differenz errechnen

Problem:	Wie kann ich zwei Zahlenkolonnen summieren, die Summen in die erste freie Zeile unterhalb der Kolonnen eintragen und in die linke Zelle darunter die Differenz der Summen anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Summieren() Dim iRow As Integer iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 Cells(iRow, 1).FormulaR1C1 = "=sum(R6C:R[-1]C)" Cells(iRow, 2).FormulaR1C1 = "=sum(R6C:R[-1]C)" Cells(iRow + 1, 1).FormulaR1C1 = "=R[-1]C-R[-1]C[1]" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus Spalte A mit Werten aus Spalte B multiplizieren

Problem:	Wie kann ich alle Werte eines variablen Bereiches der Spalten A und B in Spalte C multiplizieren? Es sollen keine festen Werte, sondern Formeln eingetragen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Berechnen() Dim iRowL As Integer, iRow As Integer iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL If Application.Count(Range(Cells(iRow, 1), _ Cells(iRow, 2))) > 0 Then Cells(iRow, 3).FormulaR1C1 = "=RC[-2]*RC[-1]" End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahlen in Abhängigkeit von ihrem Wert multiplizieren

Problem:	Wie kann ich Zahlen in einem vorgegebenen Bereich in Abhängigkeit ihres Wertes mit anderen Zahlen multiplizieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Berechnung() Dim rngAct As Range For Each rngAct In ActiveSheet.UsedRange Select Case CStr(rngAct.Value) Case "1" rngAct = rngAct * 12 Case "3" rngAct = rngAct * 8 Case "4" rngAct = rngAct * 6 End Select Next rngAct End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert auf Schaltfläche und in Zelle um eins erhöhen

Problem:	Wie kann ich auf Schaltflächendruck den Wert in der Schaltflächenaufschrift und in Zelle A1 um jeweils 1 erhöhen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdZaehlen_Click() cmdZaehlen.Caption = "Aktueller Wert: " & CStr(Wert + 1) Range("A1").Value = Wert End Sub Private Function Wert() As Integer With cmdZaehlen Wert = Val(Right(.Caption, Len(.Caption) - _ InStr(15, .Caption, " ")) End With End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen über Tabellenblatt-Schaltfläche multiplizieren

Problem: Wie kann ich die Werte zweier Zellen durch Drücken einer Tabellenblattschaltfläche in einer dritten Zelle multiplizieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Berechnen()
    Range("C1").Value = _
        Range("A1").Value * Range("B1").Value
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Maximalwert aus einem Bereich ermitteln und in Spalte A eintragen

Problem:	Wie kann ich den Maximalwert einer Tabelle ermitteln und in der Fundzeile in Spalte A eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub GetMaxValue() Dim rng As Range Dim dMax As Double dMax = WorksheetFunction.Max(Range("A1:F10")) Set rng = Range("A1:F10").Find(_ what:=dMax, LookIn:=xlValues, lookat:=xlWhole) Cells(rng.Row, 7).Value = dMax End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Markierten Bereich mit jeweiligem Mwst-Satz multiplizieren

Problem:	Wie kann ich aus einer Liste von Mehrwertsteuersätzen einen Satz auswählen und dann einem zu markierenden Bereich diese Mwst. hinzufügen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) On Error Resume Next Application.CommandBars("Standard").Controls("Multiplizieren").Delete On Error GoTo 0 End Sub Private Sub Workbook_Open() Dim objButton As CommandBarButton On Error Resume Next Application.CommandBars("Standard").Controls("Multiplizieren").Delete On Error GoTo 0 Set objButton = Application.CommandBars("Standard").Controls.Add With objButton .Caption = "Multiplizieren" .Style = msoButtonIcon .FaceId = 1088 .OnAction = "Multiplizieren" End With End Sub ClassModule: Tabelle3 Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range) Dim rngAct As Range If Application.CommandBars("Standard") _ .Controls("Multiplizieren").State = msoButtonDown Then If Target.Column <> 3 Then Beep MsgBox "Sie müssen Zellen in Spalte C markieren!" Exit Sub End If For Each rngAct In Selection.Cells rngAct = rngAct.Offset(0, -1) * dMultiplier rngAct.Offset(0, 1) = rngAct.Offset(0, -1) + rngAct Next rngAct End If Application.CommandBars("Standard") _ .Controls("Multiplizieren").State = msoButtonUp End Sub StandardModule: basMain Public dMultiplier As Double Sub Multiplizieren() If Intersect(ActiveCell, Range("A2:A6")) Is Nothing Then Beep MsgBox "Sie müssen sich im Bereich A2:A6 befinden!" Exit Sub End If If Selection.Cells.Count > 1 Then Beep MsgBox "Es darf nur eine Zelle markiert sein!" Exit Sub End If CommandBars("Standard").Controls("Multiplizieren").State = msoButtonDown dMultiplier = ActiveCell End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summenformel per VBA eintragen

Problem:	Wie kann ich die Summenformel in eine variable Anzahl von Zellen untereinander eintragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub cmdBerechnen_Click() Dim iRow% iRow = Cells(Rows.Count, 1).End(xlUp).Row Range("F2") = "=sum(A2:E2)" With Range("F2:F" & iRow) .FillDown End With Range("A1").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zwischensummen eintragen

Problem: Wie kann ich beim Wechsel zu einer neuen Artikelgruppe in einer Liste - abhängig vom 1. Buchstaben - eine Summenzeile eingefügen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub InsertSums()  
    Dim iRow As Integer, iRowL As Integer, iStart As Integer  
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row + 1  
    For iRow = iRowL To 2 Step -1  
        If Left(Cells(iRow, 1).Value, 1) <> _  
            Left(Cells(iRow - 1, 1).Value, 1) Then  
            Rows(iRow).Insert  
            iStart = iRow - 1  
            Do Until Left(Cells(iStart, 1).Value, 1) <> _  
                Left(Cells(iStart - 1, 1).Value, 1)  
                iStart = iStart - 1  
            If iStart = 1 Then Exit Do  
        Loop  
        Cells(iRow, 2).Value = _  
            WorksheetFunction.Sum( _  
                Range(Cells(iStart, 2), Cells(iRow - 1, 2)))  
        Cells(iRow, 1).Value = "Summe:"  
    End If  
Next iRow  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summe einzelner Zellwerte ermitteln

Problem:	Wie kann ich über VBA die Summe einzelner Zellwerte ermitteln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SummeBereich() MsgBox prompt:=Application.Sum(Range("A1, F5, C3")) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte um einen vorgegebenen Faktor erhöhen und verschieden runden

Problem:	Wie kann ich die Werte in einem vorgegebenen Bereich um einen in Zelle G1 stehenden Faktor erhöhen und je nach Höhe unterschiedlich runden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ErhoehenUndFormatieren() Dim rng As Range Dim dValue As Double Dim iCounter As Integer, iRow As Integer Application.ScreenUpdating = False iRow = Cells(Rows.Count, 1).End(xlUp).Row Range("C1").Copy Range("A1:A" & iRow).PasteSpecial _ Paste:=xlValues, _ Operation:=xlMultiply, _ SkipBlanks:=False, _ Transpose:=False For iCounter = 1 To iRow dValue = Cells(iRow, 1).Value Select Case dValue Case Is <= 20 Cells(iRow, 1) = WorksheetFunction.Round(dValue, 2) Case Is <= 50 Cells(iRow, 1) = WorksheetFunction.Round(dValue / 5, 2) * 5 Case Is <= 100 Cells(iRow, 1) = WorksheetFunction.Round(dValue / 10, 2) * 10 Case Is <= 500 Cells(iRow, 1) = WorksheetFunction.Round(dValue / 50, 2) * 50 Case Else Cells(iRow, 1) = WorksheetFunction.Round(dValue / 100, 2) * 100 End Select Next iCounter Application.CutCopyMode = False Application.ScreenUpdating = True End Sub</pre>



Zellbereiche löschen

Problem:	Eine Reihe von Beispielen zum Löschen oder Leeren von Zellbereichen nach verschiedenen Kriterien.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen die einzelnen Prozeduren Schaltflächen zu.
VBA-Code:	<pre>StandardModule: basMain ' Löschen aller leeren Zellen einer Spalte Sub DeleteEmptyCells() Dim intLastRow As Integer Dim intRow As Integer intLastRow = Cells.SpecialCells(xlCellTypeLastCell).Row For intRow = intLastRow To 1 Step -1 If Application.CountA(Rows(intRow)) = 0 Then intLastRow = intLastRow - 1 Else Exit For End If Next intRow For intRow = intLastRow To 1 Step -1 If IsEmpty(Cells(intRow, 1)) Then Cells(intRow, 1).Delete xlShiftUp End If Next intRow End Sub ' Löschen der Zeile, wenn Zelle in Spalte A leer ist Sub DeleteRowIfEmptyCell() Dim intRow As Integer, intLastRow As Integer intLastRow = Cells.SpecialCells(xlCellTypeLastCell).Row For intRow = intLastRow To 1 Step -1 If Application.CountA(Rows(intRow)) = 0 Then intLastRow = intLastRow - 1 Else Exit For End If Next intRow For intRow = intLastRow To 1 Step -1 If IsEmpty(Cells(intRow, 1)) Then Rows(intRow).Delete End If Next intRow End Sub ' Löschen aller leeren Zeilen Sub DeleteEmptyRows() Dim intRow As Integer, intLastRow As Integer intLastRow = Cells.SpecialCells(xlCellTypeLastCell).Row For intRow = intLastRow To 1 Step -1 If Application.CountA(Rows(intRow)) = 0 Then Rows(intRow).Delete End If Next intRow End Sub ' FehlerZellen leeren Sub ClearContentsErrorCells() On Error GoTo ERRORHANDLER Cells.SpecialCells(xlCellTypeFormulas, 16).ClearContents ERRORHANDLER: End Sub ' FehlerZellen löschen Sub ClearErrorCells() On Error GoTo ERRORHANDLER Cells.SpecialCells(xlCellTypeFormulas, 16).Delete xlShiftUp ERRORHANDLER: End Sub ' Löschen aller Zellen in Spalte A mit "hallo" im Text Sub DeleteQueryCells() Dim var As Variant Do While Not IsError(var) var = Application.Match("hallo", Columns(1), 0) If Not IsError(var) Then Cells(var, 1).Delete xlShiftUp Loop End Sub ' Leeren aller Zelle mit gelbem Hintergrund Sub ClearYellowCells() Dim rng As Range For Each rng In ActiveSheet.UsedRange If rng.Interior.ColorIndex = 6 Then rng.ClearContents End If Next rng End Sub ' Alle leeren Zellen löschen Sub DeleteEmptys() Dim rng As Range Application.ScreenUpdating = False For Each rng In ActiveSheet.UsedRange If IsEmpty(rng) Then rng.Delete xlShiftUp Next rng Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen mit Postleitzahlen separieren

Problem:	Aus einer Tabelle sollen die Zellen mit Postleitzahlen separiert werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetPLZ() Dim rng As Range, rngSource As Range Set rngSource = Range("A1").CurrentRegion Workbooks.Add 1 For Each rng In rngSource If rng.Value Like "##### [A-Z]*" Then Cells(rng.Row, 6).Value = Left(rng.Value, 5) Cells(rng.Row, 7).Value = Right(rng.Value, Len(rng.Value) - 6) Else Cells(rng.Row, rng.Column).Value = rng.Value End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Leerzellen mit Dummy-Werten füllen

Problem: In einem auszuwählenden Bereich sollen die Leerzellen mit Dummy-Werten gefüllt werden.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Emptys2Dummys()
    Dim rng As Range, rngEmptys As Range
    Dim vRng As Variant, vDummy As Variant
    On Error GoTo ERRORHANDLER
    Set vRng = Application.InputBox( _
        prompt:="Bitte Bereich auswählen:", _
        Title:="Bereichsauswahl", _
        Type:=8)
    If vRng Is Nothing Then Exit Sub
    vDummy = Application.InputBox( _
        prompt:="Bitte Dummy-Text eingeben:", _
        Title:="Dummy-Text", _
        Type:=2)
    If vDummy = False Then GoTo ERRORHANDLER
    Set rngEmptys = vRng.SpecialCells(xlCellTypeBlanks)
    rngEmptys.Value = vDummy
ERRORHANDLER:
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Länge und Höhe eines Textes auslesen

Problem: Die Länge und Höhe des Textes (abhängig von Schriftart und Größe) soll ausgelesen werden.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub GetLengthHeight()
    Dim shp As Shape
    Dim dWidth As Double, dHeight As Double
    Application.ScreenUpdating = False
    dWidth = Columns(1).ColumnWidth
    dHeight = Rows(1).RowHeight
    Columns(1).AutoFit
    Rows(1).AutoFit
    Range("A1").CopyPicture Appearance:=xlScreen, Format:=xlBitmap
    ActiveSheet.Paste
    Set shp = ActiveSheet.Shapes(2)
    MsgBox "Breite: " & shp.Width & vbCrLf & "Höhe: " & shp.Height
    shp.Delete
    Columns(1).ColumnWidth = dWidth
    Rows(1).RowHeight = dHeight
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen zeilenweise verbinden

Problem:	Ausgewählte Zellen sollen Zeile für Zeile horizontal verbunden werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Verbinden() Dim iRow As Integer, iCol As Integer, iColL As Integer If Selection.Columns.Count = 1 Then Beep MsgBox "Bitte mehr als eine Spalte auswählen!" Exit Sub End If iCol = Selection.Column iColL = Selection.Column + Selection.Columns.Count - 1 For iRow = Selection.Row To Selection.Row + Selection.Rows.Count - 1 Range(Cells(iRow, iCol), Cells(iRow, iColL)).Merge Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur Summenformel zur längsten Spalte eintragen

Problem:	In der per VBA in Zelle G1 einzutragenden Summenformel soll die Spalte variabel gehalten werden. Es ist immer die Spalte zugrunde zu legen, die die meisten Einträge enthält.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub VariableSum() Dim iCol As Integer, iAct As Integer, iCount As Integer For iCol = 1 To 3 If WorksheetFunction.CountA(Columns(iCol)) > iCount Then iCount = WorksheetFunction.CountA(Columns(iCol)) iAct = iCol End If Next iCol Range("G1").Formula = "=SUM(R1C" & iAct & ":R65536C" & iAct & ")" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einfügen von Zeilen gem. Zellwerten

Problem:	In der Spalte "Anzahl" (D) sollen unterhalb der Zellen, in denen der Wert > 0 ist, soviel Zeilen eingefügt werden, wie der Wert beträgt. Unmittelbar unter dem Wert ist eine Null einzufügen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Einfuegen() Dim rng As Range Dim iRow As Integer With Range("Anzahl") Set rng = .Cells(Rows.Count, 1).End(xlUp) iRow = 1 Do Until iRow > rng.Row If WorksheetFunction.IsText(.Cells(iRow)) = False Then If Not IsEmpty(.Cells(iRow)) Then If .Cells(iRow).Value > 0 Then iRow = iRow + 1 Rows(iRow & ":" & _ iRow + .Cells(iRow - 1).Value - 1).Insert .Cells(iRow) = 0 iRow = iRow + .Cells(iRow - 1).Value - 1 End If End If End If iRow = iRow + 1 Loop End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellauswahl in Abhängigkeit des Spaltenkopfes

Problem:	Wie kann ich die Kopfzeile nach einem Suchbegriff durchsuchen lassen und den Zellzeiger entsprechend der Fundstelle positionieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Zellauswahl() Dim iHeader As Integer Dim sHeader As String sHeader = "Spalte 5" iHeader = WorksheetFunction.Match(sHeader, Rows(1), 0) Cells(12, iHeader).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachauswahl über Schaltfläche steuern

Problem: Wie kann ich durch das Drücken einer Schaltfläche mehrere, nicht zusammenhängende Zellen auswählen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub MehrbereichsAuswahl()  
    Dim rngA As Range, rngB As Range, rngUnion As Range  
    Set rngA = ActiveCell  
    Set rngB = Cells(ActiveCell.Row, 7)  
    Set rngUnion = Application.Union(rngA, rngB)  
    rngUnion.Select  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Feststellen, ob eine bestimmte Zelle markiert ist

Problem:	Wie kann ich feststellen, ob eine bestimmte Zelle in der aktuellen Markierung enthalten ist?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Markiert() Dim rng As Range Set rng = Range("B5") If Intersect(rng, Selection) Is Nothing Then MsgBox "Zelle " & rng.Address & " ist nicht markiert!" Else MsgBox "Zelle " & rng.Address & " ist markiert!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellauswahl auf den ausgewählten Bereich beschränken

Problem: Wie kann ich das Auswählen von Zellen auf den ausgewählten Bereich beschränken und diese Beschränkung auf Kommando wieder rückgängig machen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdBegrenzung_Click()
    With cmdBegrenzung
        If .Caption = "Auswahl begrenzen" Then
            .Caption = "Begrenzung aufheben"
        Else
            .Caption = "Auswahl begrenzen"
        End If
    End With
    Call Begrenzung
End Sub

StandardModule: basMain

Sub Begrenzung()
    If ActiveSheet.ScrollArea = "" Then
        Cells.Interior.ColorIndex = 51
        Selection.Interior.ColorIndex = xlColorIndexNone
        ActiveSheet.ScrollArea = Selection.Address
    Else
        ActiveSheet.ScrollArea = ""
        Cells.Interior.ColorIndex = xlColorIndexNone
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bereiche bei Mehrfachauswahl prüfen

Problem:	Wie kann ich bei einer Mehrfachauswahl von Bereichen prüfen, ob einzelne Bereich in oder außerhalb eines vorgegebenen Bereiches liegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub RangeProof() Dim rngAll As Range, rngAct As Range, rngLast As Range Set rngAll = Range("A1:F6") For Each rngAct In Selection.Areas Set rngLast = rngAct.Cells(rngAct.Cells.Count) If Intersect(rngAll, rngAct) Is Nothing Then MsgBox "Bereich " & rngAct.Address _ & " liegt außerhalb der Vorgabe!" ElseIf Intersect(rngAll, rngLast) Is Nothing Then MsgBox "Bereich " & rngAct.Address _ & " liegt teilweise außerhalb der Vorgabe!" End If Next rngAct End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl der einzufügenden Zeilen abfragen

Problem:	Wie kann ich die Anzahl der einzufügenden Zeilen über VBA abfragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZeilenEinfuegen() Dim vRow As Variant vRow = InputBox("Anzahl der Zeilen:") If vRow = "" Then Exit Sub Rows(ActiveCell.Row & ":" & _ ActiveCell.Row + vRow - 1).Insert End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen oberhalb der markierten Zellen einfügen

Problem:	Wie kann ich unterhalb der markierten Zeilen soviel Zeilen einfügen, wie ich markiert habe?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZeilenEinfuegen() Dim iCount As Integer With Selection iCount = .Rows.Count Rows(.Row + iCount & ":" & _ .Row + (iCount * 2) - 1).Rows.Insert End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abfrage der markierten Spalten

Problem:	Wie kann ich über VBA abfragen, welche Spalten in einer Mehrbereichsauswahl markiert sind?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SelectionColumns() Dim iCol As Integer Dim sMsg As String For iCol = 1 To 256 If Not Intersect(Selection, Columns(iCol)) Is Nothing Then sMsg = sMsg & iCol & "," End If Next iCol MsgBox "Markiert: " & Left(sMsg, Len(sMsg) - 1) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benannten Bereich erweitern

Problem:	Wie kann ich einen benannten Zellbereich jeweils um eine Zeile erweitern und die letzte Zelle im Bereich aktivieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Auswaehlen() Dim rng As Range Dim iRow As Integer, iCol As Integer Set rng = Selection iRow = rng.Rows.Count iCol = rng.Columns.Count rng.Resize(iRow + 1, iCol).Name = "Datenbereich" With Range("Datenbereich") .Select .Cells(.Cells.Count).Activate End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalten/Zeilen aus- und einblenden

Problem:	Wie kann ich die Zeilen unterhalb des Cursors und die Spalten rechts des Cursors aus- bzw. einblenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub HideUnHide() If Rows(ActiveCell.Row + 1).Hidden = False Then Rows(ActiveCell.Row + 1 & ":" & Rows.Count).Hidden = True Range(ActiveCell.Offset(0, 1), _ Range("IV" & ActiveCell.Row)).EntireColumn.Hidden = True Else Columns.Hidden = False Rows.Hidden = False End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalten einer Mehrbereichsauswahl ausblenden.

Problem:	Wie kann ich die Spalten einer Mehrbereichsauswahl ausblenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MultiSelectHidden() Const csHidden As String = "Spalten ausblenden" Const csVisible As String = "Spalten einblenden" With ActiveSheet.Buttons(1) If .Caption = csHidden Then Selection.EntireColumn.Hidden = True .Caption = csVisible Else Columns.Hidden = False .Caption = csHidden End If End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Leere Zeilen löschen

Problem:	Wie kann ich alle leeren Zeilen im benutzten Bereich eines Tabellenblattes löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CellsDelete() ActiveSheet.UsedRange.SpecialCells(_ xlCellTypeBlanks).Select Selection.Delete Shift:=xlUp End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus Vorblatt

Problem: Wie kann ich einen Wert aus dem Tabellenblatt vor dem aktiven Blatt ermitteln, ohne dessen Namen zu wissen und unter der Berücksichtigung, daß sich der Name dieses Blattes ändern kann?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function Before(str As String) As Variant
    If Application.Caller.Parent.Index = 1 Then
        Before = "Es existiert kein Vorblatt!"
    Else
        Before = Worksheets( _
            Application.Caller.Parent.Index - 1) _
            .Range(str).Value
    End If
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelte Datensätze löschen

Problem:	Wie kann ich doppelte Datensätze finden und löschen?
Lösung:	Den betreffenden Bereich markieren, Menü DATEN / FILTER / SPEZIALFILTER aufrufen, "An eine andere Stelle kopieren" und "Keine Duplikate" aktivieren und Ausgabebereich angeben.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Befindet sich die aktive Zelle in einem bestimmten Bereich?

Problem: Wie kann ich feststellen, ob sich die aktive Zelle in einem bestimmten Bereich befindet?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Pruefen()  
    If Intersect(Range("A1:D4"), ActiveCell) Is Nothing Then  
        MsgBox "Outside"  
    Else  
        MsgBox "Inside"  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Leerzeichen aus einem Bereich löschen

Problem:	Wie kann ich bei einer importierten Datei alle Leerzeichen löschen?
Lösung:	Den Bereich mit den zu entfernenden Leerzeichen markieren (Mehrfachauswahl ist möglich), dann Menü BEARBEITEN / ERSETZEN - Suchen nach -> Leerzeichen eingeben - Ersetzen durch -> nichts eingeben
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattschutz für alle Arbeitsblätter setzen oder aufheben

Problem:	Über die Schaltflächen soll der Blattschutz für alle Tabellenblätter gesetzt bzw. aufgehoben werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn zwei Schaltflächen zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub YesProtect() Dim wks As Worksheet For Each wks In Worksheets wks.Protect "MeinPasswort" Next wks End Sub Sub NoProtect() Dim wks As Worksheet For Each wks In Worksheets wks.Unprotect "MeinPasswort" Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Masterlistsystem mit Passwortabfrage bei Blattauswahl

Problem: Über Passwörter können anwenderdefinierte Blätter ausgewählt werden, die nur die für den speziellen Anwender gedachten Daten aus der MasterList enthalten.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn den Schaltflächen zu.

VBA-Code:

```
StandardModule: Modul1

Sub SelectWks()
    Dim sPW As String
    sPW = ActiveSheet.Buttons(Application.Caller).Caption
    If InputBox("Passwort:", "Blattwechsel", sPW) <> sPW Then
        Beep
        MsgBox "Falsches Passwort"
        Exit Sub
    End If
    Call EinAusblenden
    With Worksheets(sPW)
        If sPW <> "MasterList" Then
            .Cells.ClearContents
            Worksheets("MasterList").Range(sPW).Copy .Range("A2")
            .Columns.AutoFit
        End If
        .Visible = xlSheetVisible
        .Select
    End With
    Application.CutCopyMode = False
End Sub

Sub GoHome()
    Call EinAusblenden
    Worksheets("Cover").Select
End Sub

Sub EinAusblenden()
    Dim iWks As Integer
    For iWks = Worksheets("Cover").Index + 1 To Worksheets.Count
        Worksheets(iWks).Visible = xlVeryHidden
    Next iWks
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur nichtfarbige Zellen schützen

Problem: Alle Zellen mit farbigem Hintergrund sollen als Eingabezellen nicht geschützt, alle anderen geschützt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SetProtect()
    Dim rng As Range
    Cells.Locked = True
    For Each rng In ActiveSheet.UsedRange.Cells
        If Not rng.Interior.ColorIndex = 6 Then
            rng.Locked = False
        End If
    Next rng
    ActiveSheet.Protect
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattschutz mit und ohne Passwort aufheben und setzen

Problem:	Der Blattschutz für einzelne oder alle Arbeitsblätter soll mit oder ohne Passwort gesetzt und aufgehoben werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub ProtectA() ActiveSheet.Protect End Sub Sub ProtectB() ActiveSheet.Unprotect End Sub Sub ProtectC() Dim wks As Worksheet For Each wks In Worksheets wks.Protect Next wks End Sub Sub ProtectD() Dim wks As Worksheet For Each wks In Worksheets ActiveSheet.Unprotect Next wks End Sub Sub ProtectE() ActiveSheet.Protect "HWH" End Sub Sub ProtectF() ActiveSheet.Unprotect "HWH" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattschutz für VBA-Einträge ausschließen

Problem:	Das Arbeitsblatt soll so geschützt werden, dass dennoch VBA-Einträge erfolgen dürfen.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub InsertData() Range("A1").Value = "Hallo" End Sub Sub SheetProtect() ActiveSheet.Protect userinterfaceonly:=True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattschutz aufheben, Daten eintragen, Blattschutz setzen

Problem:	Vor dem Dateneintrag soll geprüft werden, ob ein Blattschutz besteht. Wenn ja, soll er aufgehoben, die Daten eingetragen und der Schutz wieder gesetzt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub ProtectUnprotect() Dim bln As Boolean bln = Tabelle1.ProtectContents ActiveSheet.Unprotect Range("A1").Value = "Hallo" If bln Then ActiveSheet.Protect End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattschutz setzen und dennoch mit AutoFiltern

Problem:	Der Blattschutz soll gesetzt und das Arbeiten mit dem AutoFilter weiterhin ermöglicht werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BlattSchutz() ActiveSheet.EnableAutoFilter = True Tabelle1.Protect userinterfaceonly:=True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattschutz für eine Serie von Tabellen setzen und aufheben

Problem: Mit der ersten Schaltfläche soll für Tabelle2 bis Tabelle6 ein Blattschutz gesetzt, mit der zweiten soll dieser aufgehoben werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub BlattSchutzSetzen()
    Dim iWks As Integer
    For iWks = 3 To Worksheets.Count
        Worksheets(iWks).Protect "MeinPasswort"
    Next iWks
End Sub

Sub BlattSchutzAufheben()
    Dim iWks As Integer
    For iWks = 3 To Worksheets.Count
        Worksheets(iWks).Unprotect "MeinPasswort"
    Next iWks
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle vor der Bearbeitung schützen

Problem: Wie kann ich eine Tabelle von der Bearbeitung und dem Kopieren von Zellen schützen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub ProtectSheet()
    Range("A1").Select
    Range("A1:O20").CopyPicture _
        Appearance:=xlScreen, _
        Format:=xlBitmap
    ActiveSheet.Paste
    ActiveSheet.Protect
End Sub

Sub UnprotectSheet()
    ActiveSheet.Unprotect
    ActiveSheet.Pictures(1).Delete
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattschutz aufheben

Problem: Wie kann ich den Blattschutz aller Tabellenblätter in allen geöffneten Arbeitsmappen über eine VBA-Routine aufheben?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub BlattschutzAufheben()
    Dim wkb As Workbook
    Dim wks As Worksheet
    For Each wkb In Workbooks
        For Each wks In wkb.Worksheets
            wks.Unprotect
        Next wks
    Next wkb
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kennwort für Blattschutz

Problem:	Wie kann ich den Blattschutz aufheben, wenn ich das Kennwort vergessen habe?
Lösung:	Kopieren Sie den Inhalt des geschützten Blattes in ein neues, nicht geschütztes Blatt. Bitte beachten: Wenn beim Zellformat unter Schutz die Checkbox "Formel ausblenden" aktiviert ist, kann die Formel nicht wiederhergestellt werden.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorgegebene Shortcuts ausschalten

Problem:	Wie kann ich die eingebauten und benutzerdefinierten Shortcuts aus- und wieder einschalten?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ShortysOff() Dim iCounter As Integer On Error Resume Next For iCounter = 1 To 255 Application.OnKey "^" & Chr(iCounter), "" Next iCounter On Error GoTo 0 End Sub Sub ShortysOn() Dim iCounter As Integer On Error Resume Next For iCounter = 1 To 255 Application.OnKey "^" & Chr(iCounter) Next iCounter On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Artikel mit den in einem Farbindex hinterlegten Farben markieren

Problem:	Auf Schaltflächenklick sollen die Artikel in Spalte A mit den in Tabelle Farbindex hinterlegten Hintergrundfarben versehen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetColors() Dim wks As Worksheet Dim vColor As Variant Dim iRow As Integer, iRowL As Integer Set wks = Worksheets("FarbIndex") iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL vColor = Application.Match(Cells(iRow, 1).Value, wks.Columns(1), 0) If Not IsError(vColor) Then Cells(iRow, 1).Interior.ColorIndex = wks.Cells(vColor, 2).Value End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Festlegung von Farkombinationen durch ScrollBars

Problem:	Die Farben von Shapes sollen durch ScrollBars festgelegt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle3 Private Sub scrRed_Change() With Range("C5:D5") .Interior.Color = RGB(scrRed.Value, 0, 0) .Font.Color = 16581375 - (_ scrRed.Value * scrGreen.Value * scrBlue.Value) End With Call SetCellColor End Sub Private Sub scrGreen_Change() Range("C7:D7").Interior.Color = _ RGB(0, scrGreen.Value, 0) Call SetCellColor End Sub Private Sub scrBlue_Change() Range("C9:D9").Interior.Color = _ RGB(0, 0, scrBlue.Value) Call SetCellColor End Sub Private Sub SetCellColor() Range("C4").Interior.Color = _ RGB(scrRed.Value, scrGreen.Value, scrBlue.Value) Shapes(1).Fill.ForeColor.RGB = RGB(_ scrRed.Value, scrGreen.Value, scrBlue.Value) DoEvents End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Label und Zellen farbig markieren

Problem:	In Spalte A befinden sich 35 Labels aus der Steuerelementtoolbox, die mit verschiedenen Gelb-Schattierungen belegt werden können. In Spalte B werden die möglichen Zellfarben eingetragen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetLabelColors() Dim oLabel As OLEObject Dim iCounter As Integer For iCounter = 1 To 25 ActiveSheet.OLEObjects(iCounter).Object.BackColor = RGB(255, 255, 255 - iCounter * 10) 'ActiveSheet.OLEObjects(iCounter).Object.BackColor = RGB(255, 255, 255) Next iCounter End Sub Sub SetCellColors() Dim iCounter As Integer For iCounter = 1 To 56 Cells(iCounter, 2).Interior.ColorIndex = iCounter Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Legende eines Datenblattes einschließlich der Farbmarkierung

Problem:	Wie kann ich die Legende eines Datenblattes einschliesslich der zugehörigen Farbmarkierung auslesen und übertragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub LegendeAuslesen() Dim iCol As Integer, iRow As Integer iCol = 1 With Worksheets("Daten") Do Until IsEmpty(.Cells(1, iCol)) iRow = iRow + 1 Cells(iRow, 1).Interior.Color = _ .Cells(1, iCol).Interior.Color Cells(iRow, 2).Value = .Cells(2, iCol).Value iCol = iCol + 1 Loop End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Farbindex von 1 bis 56 auflisten

Problem:	Wie kann ich alle Farbindex-Nummern von 1 bis 56 mit der zugehörigen Farbe in einer Tabelle auflisten lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FarbIndices() Dim iCounter As Integer For iCounter = 1 To 56 Cells(iCounter, 1).Interior.ColorIndex = iCounter Cells(iCounter, 2) = iCounter Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Farbpalette auf Graustufen setzen

Problem:	Wie kann ich die Farbpalette für die aktive Arbeitsmappe so ändern, daß ausschließlich abgestufte Graustufen zur Verfügung stehen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SetColors() Dim iCounter As Integer, iColor As Integer Application.ScreenUpdating = False For iCounter = 1 To 56 iColor = 256 / 56 ActiveWorkbook.Colors(iCounter) = _ RGB(iColor * iCounter, iColor * _ iCounter, iColor * iCounter) Next iCounter End Sub Sub ResetColors() ActiveWorkbook.ResetColors End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausgeblendetes Fenster ein- und aktives Fenster ausblenden

Problem:	Wie kann ich über eine Schaltfläche das aktive Fenster aus- und ein ausgeblendetes einblenden lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: modMain Sub DateiEinAusblenden() Dim wkb As Workbook Dim sWindow As String, sTest As String sTest = "Test.xls" On Error Resume Next Set wkb = Workbooks(sTest) If Err > 0 Or wkb Is Nothing Then Beep MsgBox "Keine Arbeitsmappe " & sTest & " geladen!" Exit Sub End If On Error GoTo 0 sWindow = ActiveWindow.Caption Windows("Test.xls").Visible = True Windows(sWindow).Visible = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fenstername ändern

Problem:	Wie kann ich dem Namen des aktiven Fensters den Anwendernamen hinzufügen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FensterName() With ActiveWindow .Caption = .Caption & " - " & _ Application.UserName End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalten mit Zellen ohne Eintrag oder 0-Werten ausblenden

Problem:	Alle Spalten ohne Wert oder mit dem Wert 0 sollen ausgeblendet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn zwei Schaltflächen zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub HideColumns() Dim iCol As Integer Application.ScreenUpdating = False For iCol = 1 To 256 If Cells(5, iCol).Value = 0 Then Columns(iCol).Hidden = True End If Next iCol Application.ScreenUpdating = True End Sub Sub UnHideColumns() Columns.Hidden = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Filtern über VBA nach mehreren Kriterien

Problem:	Die Daten in Spalte A sollen nach den in den Zellen D1 und F1 über ListBoxes auszuwählende Kriterien gefiltert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Filtern() Range("A1").CurrentRegion.AutoFilter _ Field:=1, _ Criteria1:="*" & Range("D1").Value & "*", _ Operator:=xlOr, _ Criteria2:="*" & Range("F1").Value & "*" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Autofilterkriterium abfragen und filtern

Problem:	Der Autofilter soll mit einem über InputBox abzufragenden Kriterium aufgerufen werden. Vorschlagswert in der InputBox soll der Wert in Zelle D1 sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetFilter() Dim sFilter As String sFilter = InputBox("Filter:", , Range("D1").Value) If sFilter = "" Then Exit Sub Range("A1").CurrentRegion.AutoFilter _ Field:=1, Criteria1:=sFilter, Operator:=xlAnd End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Liste nach Kriterien Filtern und Ergebnisse in ListBox eintragen

Problem: Bei Eingaben in den Zellen F1, F2 oder F3 sollen die Werte in der Tabelle gefiltert und die sichtbaren Werte in die ListBox eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim iCol As Integer
    Dim sFilter As String
    If Target.Cells.Count > 1 Then Exit Sub
    iCol = Target.Column - 5
    sFilter = Target.Value
    If Intersect(Range("F2:H2"), Target) Is Nothing Then
        Exit Sub
    Else
        Call GetFilter(iCol, sFilter)
        Calculate
    End If
End Sub

StandardModule: Modul1

Sub GetFilter(iColT As Integer, sFilterT As String)
    Dim wks As Worksheet
    Dim rng As Range, rngFilter As Range
    Dim iCol As Integer
    On Error GoTo ERRORHANDLER
    Set wks = ActiveSheet
    Set rng = Range("A1").CurrentRegion
    rng.AutoFilter field:=iColT, Criteria1:="=" & sFilterT
    Set rngFilter = rng.SpecialCells(xlCellTypeVisible)
    Workbooks.Add 1
    rngFilter.Copy Range("A1")
    Rows(1).Delete
    With Tabelle1.lstFilter
        .Clear
        .List = Range("A1").CurrentRegion.Value
    End With
    ActiveWorkbook.Close savechanges:=False
ERRORHANDLER:
    wks.AutoFilterMode = False
    Application.EnableEvents = True
    If Err > 0 Then
        Err.Clear
        MsgBox "Es wurden keine Werte gefunden!"
        If ActiveWorkbook.Name <> ThisWorkbook.Name Then
            ActiveWorkbook.Close savechanges:=False
            ActiveSheet.AutoFilterMode = False
        End If
    End If
End Sub

Sub a()
    Application.EnableEvents
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenweise filtern

Problem:	Eine Tabelle soll zeilenweise gefiltert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub RowFilter() Dim rng As Range Worksheets("ColumnFilter").Cells.Clear Columns.Hidden = False Range("A1").CurrentRegion.Copy Worksheets("ColumnFilter").Select Range("A1").PasteSpecial _ Paste:=xlAll, Operation:=xlNone, _ SkipBlanks:=False, Transpose:=True Range("A1").Select Cells(1, Worksheets("Data").Range("A1").CurrentRegion.Rows.Count + 2).Formula = _ "=COUNTA(A:A)" Worksheets("ColumnFilter").Range("A1").AutoFilter End Sub Sub ShowColumns() Columns.Hidden = False End Sub ClassModule: Tabelle2 Private Sub Worksheet_Calculate() Dim iCol As Integer, iColL As Integer iColL = Range("A1").CurrentRegion.Columns.Count For iCol = 1 To iColL Worksheets("Data").Columns(iCol).Hidden = Rows(iCol).Hidden Next iCol Worksheets("Data").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Zeilen mit dem Wahrheitswert WAHR ausblenden

Problem:	Alle Zeilen, deren Wert in Spalte A WAHR ist, sollen ausgeblendet werden. Das Ausblenden soll nicht zeilenweise erfolgen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Hidewahr() Dim rng As Range Dim iRow As Integer, iRowL As Integer iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL If Cells(iRow, 1).Value = True Then If rng Is Nothing Then Set rng = Cells(iRow, 1) Else Set rng = Application.Union(rng, Cells(iRow, 1)) End If End If Next iRow rng.EntireRow.Hidden = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gefilterte Daten in Array ein und aus Array auslesen

Problem:	Die sichtbaren Zellen der gefilterten Daten sollen in ein Array ein- und aus diesem ausgelesen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub FilterIndex() Dim rng As Range Dim arr As Variant Dim iRow As Integer, iCol As Integer Dim iRowC As Integer, iColC As Integer Application.ScreenUpdating = False Set rng = Range("A1").CurrentRegion _ .SpecialCells(xlCellTypeVisible) Workbooks.Add rng.Copy Range("A1") Rows(1).Delete arr = Range("A1").CurrentRegion With Range("A1").CurrentRegion iRowC = .Rows.Count iColC = .Columns.Count End With ActiveWorkbook.Close savechanges:=False For iRow = 1 To iRowC For iCol = 1 To iColC MsgBox arr(iRow, iCol) Next iCol Next iRow Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus Adressangabe den Ort in eine andere Spalte separieren

Problem:	Aus den Daten in Spalte A soll der Ort in Spalte B separiert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub PLZTrennen() Dim iRow As Integer, iCounter As Integer Dim sOrt As String, sTmp As String For iRow = 1 To WorksheetFunction.CountA(Columns(1)) sOrt = Cells(iRow, 1) sTmp = sOrt iCounter = 0 Do Until sTmp Like "#####" iCounter = iCounter + 1 sTmp = Mid(sOrt, iCounter, 5) Loop Cells(iRow, 2) = Right(sOrt, Len(sOrt) - iCounter - 5) Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Datum filtern

Problem:	Es soll nach Datum in Spalte 2 gefiltert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub DateFilter() Dim datA As Long, datB As Long datA = CLng(DateValue("01.02.2001")) datB = CLng(DateValue("01.03.2001")) Selection.AutoFilter _ field:=2, _ Criteria1:=">=" & datA, _ Operator:=xlAnd, _ Criteria2:="<" & datB End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Autofilter-Kriterien nach dem Filtern anzeigen

Problem:	Nach dem Autofiltern sollen in einer zweiten Tabelle die Überschriften der gefilterten Daten mit den zugehörigen Kriterien angezeigt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FilterCriteria() Dim iRow As Integer, iCol As Integer iRow = Range("A1").CurrentRegion.Rows.Count + 2 iCol = 1 Do Until IsEmpty(Cells(1, iCol)) With ActiveSheet.AutoFilter.Filters(iCol) If .On Then Cells(iRow, iCol).Value = .Criteria1 End If End With iCol = iCol + 1 Loop Range("A1").CurrentRegion.SpecialCells _ (xlCellTypeVisible).Copy Cells(iRow + 1, 1) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit VBA nach Datum filtern

Problem: Aus den Tabellendaten sollen die Zeilen bis zum 05.01.2001 gefiltert werden. Das Filtern soll über VBA vorgenommen werden.

Lösung: eben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub DatumFiltern()
    Range("A1").CurrentRegion.AutoFilter field:=1, _
        Criteria1:="<=" & CDb1(DateSerial(2001, 1, 5))
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus einer Adressliste je 10 Adressen pro PLZ filtern

Problem: Wie kann ich aus einer Adressliste die Adressen einer Liste vorgegebener PLZ filtern und vom Ergebnis maximal 10 Stück pro PLZ auflisten lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub FilternPLZ()  
    Dim iRow As Integer, iCounter As Integer, iRowT As Integer  
    Application.ScreenUpdating = False  
    Columns("A:C").AdvancedFilter _  
        Action:=xlFilterCopy, _  
        CriteriaRange:=Range("E1").CurrentRegion, _  
        CopyToRange:=Range("I1"), _  
        Unique:=False  
    Columns("I:K").Sort _  
        Key1:=Range("K2"), Order1:=xlAscending, Header:=xlGuess, _  
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom  
    iRow = Cells(Rows.Count, 11).End(xlUp).Row  
    Columns("A:H").Delete  
    Range("A:C").Copy Range("D:F")  
    iRow = 2  
    iRowT = 1  
    Do Until IsEmpty(Cells(iRow, 3))  
        Do Until Cells(iRow, 3).Value <> Cells(iRow + 1, 3).Value  
            iCounter = iCounter + 1  
            If iCounter <= 10 Then  
                iRowT = iRowT + 1  
                Range(Cells(iRowT, 4), Cells(iRowT, 6)).Value = _  
                    Range(Cells(iRow, 1), Cells(iRow, 3)).Value  
            End If  
            iRow = iRow + 1  
        Loop  
        iCounter = 0  
        iRow = iRow + 1  
    Loop  
    Columns("A:C").Delete  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Modalwert einer gefilterten Reihe berechnen

Problem:	Wie kann ich den Modalwert einer mit Autofilter gefilterten Zahlenreihe ermitteln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub TeilMod() Dim rng As Range Dim dModal As Double Application.ScreenUpdating = False Set rng = Range("A2:A20").SpecialCells(xlCellTypeVisible) Workbooks.Add rng.Copy Range("A1") dModal = WorksheetFunction.Mode(Range("A1").CurrentRegion) ActiveWorkbook.Close savechanges:=False MsgBox "Modalwert: " & dModal End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle nach kleinstem und größtem Datum filtern

Problem: Wie kann ich auf Schaltflächendruck eine Tabelle nach Min-/MaxDatum filtern? Start- und Enddatum stehen in zwei Zellen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Filtern()
    Range("A1").Select
    Selection.AutoFilter Field:=8, Criteria1:=">" & _
        CDb1(DateValue(Range("Datum1").Text)), Operator:=xlAnd, _
        Criteria2:="<" & CDb1(DateValue(Range("Datum2").Text))
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Filtern und Ergebnisse in Tabellen übertragen

Problem:	Wie kann ich eine Tabelle nach 3 Kriterien Filtern und die Filterergebnisse in 3 Tabellenblätter übertragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FilternUndKopieren() Application.ScreenUpdating = False With Range("A1") .AutoFilter Field:=3, Criteria1:="<10" .CurrentRegion.SpecialCells(xlCellTypeVisible).Copy _ Worksheets("Kleiner10").Range("A1") .AutoFilter Field:=3, Criteria1:="<50" .CurrentRegion.SpecialCells(xlCellTypeVisible).Copy _ Worksheets("Kleiner50").Range("A1") .AutoFilter Field:=3, Criteria1:=">=50" .CurrentRegion.SpecialCells(xlCellTypeVisible).Copy _ Worksheets("Andere").Range("A1") End With ActiveSheet.AutoFilterMode = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Filtern und Kopieren

Problem:	Wie kann ich eine Tabelle nach einem in einer Zelle stehenden Kriterium filtern und die gefundenen Datensätze in eine zweite Tabelle übertragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FilternKopieren() Dim rngAct As Range With Worksheets("Tabelle2") .Cells.Clear Range("A1").AutoFilter _ Field:=1, Criteria1:=Range("D2").Value Set rngAct = Range("A2").CurrentRegion.SpecialCells(xlCellTypeVisible) rngAct.Copy .Range("A1") .Rows(1).Delete End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle nacheinander filtern und drucken

Problem:	Wie kann ich eine Tabelle nacheinander nach verschiedenen Kriterien filtern und das jeweilige Ergebnis drucken lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FilternUndDrucken() Dim arr() Dim iRow As Integer Range("A:A").AdvancedFilter Action:=xlFilterCopy, _ CopyToRange:=Range("D1"), Unique:=True iRow = 2 Do Until IsEmpty(Cells(iRow, 4)) ReDim Preserve arr(iRow - 1) arr(iRow - 1) = Cells(iRow, 4) iRow = iRow + 1 Loop Columns(4).ClearContents For iRow = 1 To UBound(arr) Columns(1).AutoFilter Field:=1, Criteria1:=arr(iRow) ActiveSheet.PrintPreview Next iRow Range("A1").AutoFilter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Unter Einsatz von VBA nach Datum filtern

Problem:	Wie kann ich über VBA nach Datum filtern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Filtern() Dim vDat As Variant vDat = InputBox(_ prompt:="Datum:", _ Default:=Format(Date + 6, "dd.mm.yy")) If vDat = "" Then Exit Sub vDat = CDate(vDat) Range("A1").CurrentRegion.AutoFilter _ field:=3, Criteria1:=">=" & CDb1(vDat) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gefilterte Daten in neues Tabellenblatt übertragen

Problem:	Wie kann ich eine bestimmte Warengruppe einer Tabelle per VBA filtern, in ein neu zu erstellendes Arbeitsblatt übertragen und dieses nach der Warengruppe benennen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub GruppenNummer() Dim rng As Range Dim sFilter As String Application.ScreenUpdating = False sFilter = InputBox(_ prompt:="Gruppe eingeben:", _ Default:="Ware 3") If sFilter = "" Then Exit Sub Range("A1").AutoFilter Field:=1, Criteria1:=sFilter Set rng = Range("A1").CurrentRegion.SpecialCells(xlCellTypeVisible) Worksheets.Add.Move After:=Worksheets(Worksheets.Count) rng.Copy Range("A1") Columns("A:B").AutoFit ActiveSheet.Name = sFilter rng.Parent.Select ActiveSheet.AutoFilterMode = False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlinks setzen, wenn Zieldatei gefunden wurde

Problem:	In den ausgewählten Zellen werden, wenn die Zieladresse gefunden wurde, Hyperlinks zu Zieladresse gesetzt.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetHyperlinks() Dim rng As Range For Each rng In Selection.Cells If Not IsEmpty(rng) Then If Dir(rng.Value) <> "" Then rng.Hyperlinks.Add _ anchor:=rng, _ Address:=rng.Value End If End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adressen grau markierter Zellen als Hyperlink eintragen

Problem:	In Spalte A sollen die Adressen der in den Folgeblättern grau markierten Zellen als Hyperlinks eingetragen werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub InsertHyperlinks() Dim rng As Range Dim iWks As Integer, iRow As Integer For iWks = Worksheets("Cover").Index To Worksheets.Count For Each rng In Worksheets(iWks).UsedRange.Cells If rng.Interior.ColorIndex = 15 Then iRow = iRow + 1 ActiveSheet.Hyperlinks.Add _ anchor:=Cells(iRow, 1), _ Address:="", _ SubAddress:=Worksheets(iWks).Name & "!" & rng.Address End If Next rng Next iWks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Hyperlink-Klick Mitnahme des Zellwertes

Problem:	Bei Anklicken eines Hyperlinks soll der Cursor unter Mitnahme des angeklickten Wertes in die Zelle Ziel in Tabelle 2 springen. Die Zelle Ziel soll vor sonstigen Eingaben geschützt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_FollowHyperlink(ByVal Target As Hyperlink) With Worksheets("Tabelle2") .Unprotect .Range("Ziel").Value = Target.Range.Value .Protect End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Hyperlinks eines Arbeitsblattes entfernen

Problem:	Auf Schaltflächenklick sollen alle Hyperlinks im Blatt Text entfernt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub DeleteHyperlinks() Worksheets("Text").Hyperlinks.Delete End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlink zu einem Diagrammblatt.

Problem: Zu Diagrammen können keine Hyperlinks verweisen. Dieses Workaround simuliert einen Hyperlink. Über den Zelleninhalt in Zelle A11 wurde ein Zeichenelement gelegt, dem ein Makro zugewiesen wurde.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einem Zeichnungselement zu.

VBA-Code: StandardModule: Modul1

```
Sub ToChart()  
    Diagramm1.Activate  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Email-Adressen in Hyperlinks umwandeln

Problem: Die Email-Adressen in Spalte A wurden aus einer Text-Datei importiert und sollen in Email-Hyperlinks umgewandelt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SetHyperlinks()
    Dim iRow As Integer
    ActiveSheet.Hyperlinks.Delete
    iRow = 2
    Do Until IsEmpty(Cells(iRow, 1))
        Cells(iRow, 1).Hyperlinks.Add Cells(iRow, 1), _
            "mailto:" & Cells(iRow, 1).Value
        iRow = iRow + 1
    Loop
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlinks umadressieren

Problem:	Die Hyperlinks in Spalte F sollen auf Schaltflächenklick umadressiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ChangeHypes() Dim iRow As Integer, iRowL As Integer, iChar As Integer Dim sFile As String iRowL = Cells(Rows.Count, 6).End(xlUp).Row For iRow = 1 To iRowL sFile = Cells(iRow, 6).Hyperlinks(1).Address For iChar = Len(sFile) To 1 Step -1 If Mid(sFile, iChar, 1) = "/" Then Exit For Next iChar sFile = Right(sFile, Len(sFile) - iChar) Cells(iRow, 6).Hyperlinks(1).Address = "c:\test\" & sFile Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlinks sollen mit Quell- und Zieladresse gelistet werden

Problem:	Die Hyperlinks in diesem Blatt sollen in einer neuen Arbeitsmappe unter Angabe der Hyperlinkzelle und des Hyperlinkziels gelistet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetHyperlinks() Dim wks As Worksheet Dim hyp As Hyperlink Dim iRow As Integer Set wks = ActiveSheet Workbooks.Add 1 For Each hyp In wks.Hyperlinks iRow = iRow + 1 Cells(iRow, 1).Value = hyp.Range.Address Cells(iRow, 2).Value = hyp.SubAddress Next hyp End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob es sich um eine gültige Internetseite handelt

Problem:	Die Hyperlinks in A sollen auf ihre Gültigkeit geprüft werden. Es muss eine Internetverbindung bestehen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CheckHyperlinks() Dim itc As Inet Dim ws As Worksheet Dim hyp As Hyperlink Set itc = New Inet With itc .Protocol = icHTTP .RequestTimeout = 5 End With On Error Resume Next For Each ws In ActiveWorkbook.Worksheets For Each hyp In ws.Hyperlinks If Len(itc.OpenURL(hyp.Address, icString)) Then hyp.Parent.Interior.ColorIndex = _ xlColorIndexNone Else hyp.Parent.Interior.ColorIndex = 3 End If Next hyp Next ws On Error GoTo 0 Set itc = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlinks zu einer Tabellenblattliste erstellen

Problem:	In Spalte B sollen Hyperlinks zu den in Spalte A genannten Tabellen erstellt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CreateHyperlinks() Dim iRow As Integer iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) Cells(iRow, 2).Hyperlinks.Add _ Anchor:=Cells(iRow, 2), _ Address:="", _ SubAddress:=Cells(iRow, 1).Value & "!A1" iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zum aufrufenden Blatt zurück

Problem: Die Hyplinks in den Zellen A1 der Blätter Tabelle1 und Tabelle2 führen zu Tabelle3. Die Zurück-Schaltfläche in Tabelle3 soll in das jeweilige aufrufende Blatt zurückführen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Activate()
    ThisWorkbook.Worksheets("Dummy").Range("A1").Value = Me.Name
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    ThisWorkbook.Worksheets("Dummy").Range("B1").Value = Target.Address
End Sub

ClassModule: Tabelle2

Private Sub Worksheet_Activate()
    ThisWorkbook.Worksheets("Dummy").Range("A1").Value = Me.Name
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    ThisWorkbook.Worksheets("Dummy").Range("B1").Value = Target.Address
End Sub

StandardModule: Modul1

Sub GoBack()
    With ThisWorkbook.Worksheets("Dummy")
        Worksheets(.Range("A1").Value).Select
        Range(.Range("B1").Value).Select
    End With
End Sub

Sub Ausblenden()
    Worksheets("Dummy").Visible = xlVeryHidden
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelle auswählen, Link erstellen und Format übernehmen

Problem:	Wie kann ich über eine InputBox eine Zelle auswählen, das Format zur aktiven Zellen übertragen und dort einen Link zur ausgewählten Zelle erstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BezugMitFormat() Dim rng As Range Set rng = Application.InputBox _ (prompt:="Quellbereich angeben:", Type:=8) rng.Copy ActiveCell.PasteSpecial xlPasteFormats ActiveCell.Formula = "=" & rng.Parent.Name & _ "!" & rng.Address End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Leere und Zellen mit 0 aus- / einblenden

Problem: Wie kann ich über eine Schaltfläche alle Zeilen, deren Zelle in Spalte A leer ist oder eine 0 beinhaltet über eine Schaltfläche ein- und ausblenden?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Ausblenden()  
    Dim iRow As Integer, iRowL As Integer  
    If ActiveSheet.Buttons(Application.Caller).Caption = "Ausblenden" Then  
        iRowL = Cells(Rows.Count, 1).End(xlUp).Row  
        For iRow = 1 To iRowL  
            If IsEmpty(Cells(iRow, 1)) Then  
                Rows(iRow).Hidden = True  
            ElseIf WorksheetFunction.IsText(Cells(iRow, 1)) Then  
            ElseIf Cells(iRow, 1).Value = 0 Then  
                Rows(iRow).Hidden = True  
            End If  
        Next iRow  
        ActiveSheet.Buttons(Application.Caller).Caption = "Einblenden"  
    Else  
        Rows.Hidden = False  
        ActiveSheet.Buttons(Application.Caller).Caption = "Ausblenden"  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auswahl in andere Arbeitsmappe kopieren, speichern, schließen

Problem: Ausgewählte Zellwerte sollen in eine geöffnete Arbeitsmappe in ein vorgegebenes Blatt ab Zelle C12 kopiert werden. Danach soll die Zielarbeitsmappe gespeichert und geschlossen werden, ohne dass eine Rückfrage irgendwelcher Art erfolgt.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code:

```
StandardModule: Modul1

Sub Copy2Wkb()
    Dim wkb As Workbook
    Dim wks As Worksheet
    Dim rng As Range
    On Error Resume Next
    Set wkb = Workbooks("Test.xls")
    If Err > 0 Or wkb Is Nothing Then
        Beep
        MsgBox _
            prompt:="Es ist keine Testdatei geöffnet!"
        Exit Sub
    End If
    Set wks = wkb.Worksheets("GSM_Daten")
    If Err > 0 Or wks Is Nothing Then
        Beep
        MsgBox _
            prompt:="Die Testdatei enthält das Zielblatt nicht!"
        Exit Sub
    End If
    Set rng = wkb.Worksheets("GSM_Daten").Range("C12")
    Selection.Copy rng
    Application.CutCopyMode = False
    wkb.Close savechanges:=True
End Sub
```



Aus einer Serie von Arbeitsmappen einen Bereich sammeln

Problem:	Aus einem Quellverzeichnis, das aus einem Dialog auszuwählen ist soll aus allen dort vorhandenen Arbeitsmappen der Bereich "A1:C10" untereinander in dieses Arbeitsblatt kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basFunctions Public Type BROWSEINFO hOwner As Long pidlRoot As Long pszDisplayName As String lpszTitle As String ulFlags As Long lpfn As Long lParam As Long iImage As Long End Type Declare Function SHGetPathFromIDList Lib "shell32.dll" _ Alias "SHGetPathFromIDListA" _ (ByVal pidl As Long, ByVal pszPath As String) As Long Declare Function SHBrowseForFolder Lib "shell32.dll" _ Alias "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long Function GetDirectory(Optional Msg As String) As String Dim bInfo As BROWSEINFO Dim Path As String Dim r As Long, x As Long, pos As Integer bInfo.pidlRoot = 0& If IsMissing(Msg) Then bInfo.lpszTitle = "Wählen Sie bitte einen Ordner aus." Else bInfo.lpszTitle = Msg End If bInfo.ulFlags = &H1 x = SHBrowseForFolder(bInfo) Path = Space\$(512) r = SHGetPathFromIDList(ByVal x, ByVal Path) If r Then pos = InStr(Path, Chr\$(0)) GetDirectory = Left(Path, pos - 1) Else GetDirectory = "" End If End Function Function FileArray(strPath As String, strPattern As String) Dim arrDateien() Dim intCounter As Integer Dim strDatei As String If Right(strPath, 1) <> "\" Then strPath = strPath & "\" strDatei = Dir(strPath & strPattern) Do While strDatei <> "" intCounter = intCounter + 1 ReDim Preserve arrDateien(1 To intCounter) arrDateien(intCounter) = strDatei strDatei = Dir() Loop FileArray = arrDateien End Function StandardModule: basMain Sub DatenImport() Dim arr As Variant Dim iCounter As Integer, iRow As Integer Dim sPath As String Application.ScreenUpdating = False sPath = GetDirectory(_ "Bitte Pfad der Quelldateien auswählen:") If sPath = "" Then Exit Sub arr = FileArray(sPath, "*.xls") iRow = 1 For iCounter = 1 To UBound(arr) Workbooks.Open sPath & "\" & arr(iCounter) Range("A1:C10").Copy _ ThisWorkbook.Worksheets("Tabelle1").Cells(iRow, 1) ActiveWorkbook.Close savechanges:=False iRow = iRow + 10 Next iCounter Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufteilen der Arbeitsblätter in neue Arbeitsmappen

Problem:	Die letzten 2 Blätter einer bestehende Arbeitsmappe soll in 2 Einzeldateien aufgeteilt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Verteilen() Dim iCounter As Integer Dim sPath As String Application.ScreenUpdating = False sPath = Application.DefaultFilePath For iCounter = 3 To Worksheets.Count Worksheets(3).Copy ActiveWorkbook.SaveAs _ sPath & "\Test" & iCounter - 2 & ".xls" ActiveWorkbook.Close savechanges:=False Application.DisplayAlerts = False Worksheets(3).Delete Application.DisplayAlerts = True Next iCounter MsgBox "Die Arbeitsmappen wurden im Verzeichnis " & _ sPath & " gespeichert!" Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten aus allen Dateien eines Verzeichnisses zusammenführen

Problem: Wie kann ich die Daten der aktiven Tabelle aller Arbeitsmappen eines Verzeichnisses unter Ausschluß der Zeilen 1:6 zusammenführen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Zusammenfuehren()  
    Dim wksTarget As Worksheet  
    Dim arr As Variant  
    Dim iCounter As Integer, iRowS As Integer, iRowT As Integer  
    Dim sPath As String, sPattern As String  
    Application.ScreenUpdating = False  
    Workbooks.Add  
    Set wksTarget = Worksheets(1)  
    Range("A1") = "Datenimport"  
    Range("A1").Font.Bold = True  
    sPath = ThisWorkbook.Path  
    sPattern = "Test*.xls"  
    arr = arrAll(sPath, sPattern)  
    For iCounter = 1 To UBound(arr)  
        Workbooks.Open arr(iCounter)  
        iRowS = Cells(Rows.Count, 3).End(xlUp).Row  
        iRowT = wksTarget.Cells(Rows.Count, 1).End(xlUp).Row + 2  
        With wksTarget.Cells(iRowT, 1)  
            .Value = arr(iCounter) & ":"  
            .Font.Bold = True  
        End With  
        iRowT = iRowT + 2  
        Rows("7:" & iRowS).Copy wksTarget.Cells(iRowT, 1)  
        Application.DisplayAlerts = False  
        ActiveWorkbook.Close savechanges:=False  
        Application.DisplayAlerts = True  
    Next iCounter  
    Columns.AutoFit  
End Sub  
  
Function arrAll(sPath As String, sPattern As String) As Variant  
    Dim arr()  
    Dim iCounter As Integer  
    Dim sFile As String  
    If Right(sPath, 1) <> "\" Then sPath = sPath & "\"  
    sFile = Dir(sPath & sPattern)  
    Do While sFile <> ""  
        iCounter = iCounter + 1  
        ReDim Preserve arr(1 To iCounter)  
        arr(iCounter) = sFile  
        sFile = Dir()  
    Loop  
    arrAll = arr  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einen vorgegebenen Tabellenblattbereich in neue Arbeitsmappe

Problem:	Wie kann ich einen vorgegebenen Tabellenblattbereich mittels VBA in eine neue Arbeitsmappe kopieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NeueMappe() Workbooks.Add Range("A1:E10").Value = _ ThisWorkbook.Worksheets(2).Range("A1:E10").Value Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdatei in Tabellenblatt importieren

Problem:	Wie kann ich eine Textdatei in ein Tabellenblatt der aktiven Arbeitsmappe importieren? Das Semikolon ist Feldtrenner und der Inhalt des aktiven Blattes muß vorher gelöscht werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub TextImport() Dim wks As Worksheet Dim vFile As Variant Application.ScreenUpdating = False Set wks = ActiveSheet fFile = Application.GetOpenFilename("Textdateien (*.txt), *.txt") If vFile = False Then Exit Sub ActiveSheet.Cells.Delete Workbooks.OpenText FileName:=vFile, Tab:=False, semicolon:=True, _ comma:=False, Space:=False, other:=False ActiveSheet.UsedRange.Copy wks.Range("A1") ActiveWorkbook.Close savechanges:=False Range("A1").Select Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattinhalt von einer zur anderen Arbeitsmappe kopieren

Problem:	Wie kann ich den Inhalt eines Arbeitsblattes der einen Arbeitsmappe in ein Blatt einer zweiten Mappe so kopieren, daß das Programm von sich aus erkennt, wo die Daten zugefügt werden sollen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub OpenTest() Dim sPath As String sPath = ThisWorkbook.Path & "\" If Dir(sPath & "Test1.xls") = "" Or _ Dir(sPath & "Test2.xls") = "" Then Beep MsgBox _ prompt:="Testarbeitsmappen wurden nicht gefunden!" Exit Sub End If Application.ScreenUpdating = False Workbooks.Open sPath & "Test2.xls" Workbooks.Open sPath & "Test3.xls" ThisWorkbook.Activate Worksheets("Tabelle1").Select MsgBox "Die Testarbeitsmappen wurden geöffnet -" & vbCrLf & _ "Sie können den Test durchführen!" Application.ScreenUpdating = True End Sub Sub Kombinieren() Dim wksSource As Worksheet, wksTarget As Worksheet Dim iRow As Integer Dim sWkb As String Call TestWkb("Test2.xls") Call TestWkb("Test3.xls") Set wksSource = Workbooks("Test2.xls").Worksheets(1) Set wksTarget = Workbooks("Test3.xls").Worksheets(1) iRow = wksTarget.Cells(Rows.Count, 1).End(xlUp).Row + 2 wksTarget.Cells(iRow, 1).Value = "Daten aus Test1.xls:" wksSource.UsedRange.Copy wksTarget.Cells(iRow + 2, 1) Application.CutCopyMode = False Workbooks("Test3.xls").Activate MsgBox "Daten wurden eingefügt!" End Sub Private Sub TestWkb(sWkb As String) Dim wkb As Workbook On Error Resume Next Set wkb = Workbooks(sWkb) If Err > 0 Or wkb Is Nothing Then Beep MsgBox "Die Testarbeitsmappe " & sWkb & " ist nicht geöffnet!" End End If On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nummer aus zweiter Datei abrufen

Problem:	Wie kann ich beim Öffnen einer Arbeitsmappe eine Nummer aus einer zweiten Arbeitsmappe abrufen, diese um 1 erhöhen, eintragen und die 2. Arbeitsmappe wieder speichern und schließen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim rng As Range Dim sFile As String Application.ScreenUpdating = False Set rng = Worksheets("Tabelle1").Range("B1") sFile = ThisWorkbook.Path & "\test1.xls" If Dir(sFile) = "" Then Beep MsgBox prompt:="Testarbeitsmappe " & _ sFile & " ist nicht vorhanden!" End If Exit Sub Worksheets.Open sFile Range("B2").Value = Range("B2").Value + 1 rng.Value = Range("B2").Value ActiveWorkbook.Close savechanges:=True Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Variablenübergabe in andere Arbeitsmappen

Problem:	Wie kann ich eine Funktion in einer anderen Arbeitsmappe mit Parameter aufrufen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CallFunction() Dim wkb As Workbook Dim iStart As Integer, iEnd As Integer On Error Resume Next Set wkb = Workbooks("Test.xls") If Err > 0 Or wkb Is Nothing Then Beep MsgBox _ prompt:="Die Testdatei wurde nicht gefunden!" Exit Sub End If On Error GoTo 0 iStart = 5 iEnd = Run("Test1.xls!Plus5", iStart) MsgBox prompt:="Startwert: " & iStart & _ vbCrLf & "Endwert: " & iEnd End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ergebnis einer WebAbfrage in neue Arbeitsmappe

Problem: Zu dem in Zelle B1 genannten Abfragestring soll das Ergebnis in eine neue Arbeitsmappe importiert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub ImportHTML()
    Dim sPath As String, sHTML As String
    Application.ScreenUpdating = False
    sHTML = Range("B1").Value
    Workbooks.Add
    With ActiveSheet.QueryTables.Add(Connection:= _
        "URL;" & sHTML, Destination:=Range("A1"))
        .Name = "Waehrungen"
        If Left(Application.Version, 1) <> "8" Then
            .WebSelectionType = xlEntirePage
            .WebFormatting = True
        End If
        .Refresh BackgroundQuery:=False
    End With
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wertpapiernotierung abrufen

Problem:	Ein Musterdepot der ComDirekt-Bank soll ausgelesen werden. Bitte beachten, dass sich der Seitenaufbau bei ComDirekt ändern kann und somit Fehler bei der Datenerfassung auftreten können.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Aufruf() Call URL_Load("https://isht.comdirect.de/html/musterdepot/main.html?sPage=overview&iDepotId=1") End Sub Private Sub URL_Load(ByVal sURL As String) Dim appIE As Object Dim iCounter As Integer Dim iRow As Integer Dim sTxt As String Dim bln As Boolean Columns(1).ClearContents Set appIE = CreateObject("InternetExplorer.Application") appIE.navigate sURL Do: Loop Until appIE.Busy = False Do: Loop Until appIE.Busy = False appIE.navigate sURL Do: Loop Until appIE.Busy = False Do: Loop Until appIE.Busy = False sTxt = appIE.document.documentElement.innerText Set appIE = Nothing Close Open Application.Path & "\test.txt" For Output As #1 Print #1, sTxt Close Open Application.Path & "\test.txt" For Input As #1 Do Until EOF(1) iCounter = iCounter + 1 Line Input #1, sTxt If iCounter > 41 And iCounter < 52 Then iRow = iRow + 1 Cells(iRow, 1).Value = sTxt End If Loop Close Kill Application.Path & "\test.txt" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Webseite öffnen und eine Schaltfläche mit Makro hinzufügen

Problem: Es soll eine Webseite geöffnet und dieser ein Makro mit Schaltfläche hinzugefügt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub MacroToWeb()
    Dim mdl As Object
    Dim oBtn As Button
    Dim sFile As String, sSource As String, sTarget As String
    Dim sCode As String
    On Error GoTo ERRORHANDLER
    Application.VBE.MainWindow.Visible = False
    Application.ScreenUpdating = False
    sCode = "Sub Message()" & vbLf
    sCode = sCode & "    MsgBox ""Hallo!"" & vbLf
    sCode = sCode & "End Sub" & vbLf
    sSource = "http://www.herber.de"
    sTarget = ThisWorkbook.Path
    sFile = "flash"
    Workbooks.Open sSource & "/" & sFile & ".htm"
    With ActiveWorkbook.VBProject
        Set mdl = .VBComponents.Add(1)
        mdl.codemodule.addfromstring sCode
    End With
    Set oBtn = ActiveSheet.Buttons.Add(100, 150, 80, 20)
    Application.DisplayAlerts = False
    ActiveWorkbook.SaveAs sTarget & "\" & sFile & ".xls"
    Application.DisplayAlerts = True
    With oBtn
        .Caption = "Meldung"
        .OnAction = sFile & ".xls!Message"
    End With
ERRORHANDLER:
    Application.ScreenUpdating = True
End Sub
```



Fußballergebnisse in Tabelle eintragen und zu HTML konvertieren

Problem:	Wie kann ich Fußballergebnisse eines Spieltages in eine Tabelle eintragen, die Tabelle aktualisieren und nach HTML konvertieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basTable Sub Tabelle() Dim rngA As Range, rngB As Range, rngC As Range Dim rngD As Range, rngE As Range Dim iCounter As Integer, iRowL As Integer iRowL = Cells(Rows.Count, 13).End(xlUp).Row Set rngC = Range(Cells(1, 13), Cells(iRowL, 17)) iRowL = Cells(Rows.Count, 1).End(xlUp).Row Set rngD = Range(Cells(4, 2), Cells(iRowL, 2)) Set rngE = Range(Cells(4, 1), Cells(iRowL, 11)) Worksheets("Backup").Range("A1:K15").Value = Range("A1:K15").Value For iCounter = 1 To rngC.Rows.Count Set rngA = rngD.Find(rngC.Cells(iCounter, 1)) rngA.Offset(0, 1) = rngA.Offset(0, 1) + 1 Set rngB = rngD.Find(rngC.Cells(iCounter, 2)) rngB.Offset(0, 1) = rngB.Offset(0, 1) + 1 If rngC.Cells(iCounter, 3) > rngC.Cells(iCounter, 5) Then rngA.Offset(0, 2) = rngA.Offset(0, 2) + 1 rngB.Offset(0, 4) = rngB.Offset(0, 4) + 1 rngA.Offset(0, 9) = rngA.Offset(0, 9) + 3 ElseIf rngC.Cells(iCounter, 3) < rngC.Cells(iCounter, 4) Then rngB.Offset(0, 2) = rngB.Offset(0, 2) + 1 rngA.Offset(0, 4) = rngA.Offset(0, 4) + 1 rngB.Offset(0, 9) = rngB.Offset(0, 9) + 3 Else rngA.Offset(0, 3) = rngA.Offset(0, 3) + 1 rngB.Offset(0, 3) = rngB.Offset(0, 3) + 1 rngA.Offset(0, 9) = rngA.Offset(0, 9) + 1 rngB.Offset(0, 9) = rngB.Offset(0, 9) + 1 End If rngA.Offset(0, 5) = rngA.Offset(0, 5) + rngC.Cells(iCounter, 3) rngA.Offset(0, 7) = rngA.Offset(0, 7) + rngC.Cells(iCounter, 5) rngB.Offset(0, 5) = rngB.Offset(0, 5) + rngC.Cells(iCounter, 5) rngB.Offset(0, 7) = rngB.Offset(0, 7) + rngC.Cells(iCounter, 3) Next iCounter rngE.Sort _ Key1:=Range("K1"), Order1:=xlDescending, _ Key2:=Range("F1"), Order2:=xlDescending, _ Key3:=Range("D1"), Order3:=xlDescending, _ Header:=xlGuess, OrderCustom:=1, _ MatchCase:=False, Orientation:=xlTopToBottom For iCounter = 4 To 15 Cells(iCounter, 1) = iCounter - 3 Cells(iCounter, 10) = Cells(iCounter, 7) - Cells(iCounter, 9) Next iCounter End Sub Sub Zurueck() Worksheets("Tabelle").Range("A1:K15").Value = _ Worksheets("Backup").Range("A1:K15").Value End Sub StandardModule: basHTML Sub Brig2HTML() Dim iRow As Integer, iCol As Integer Dim sFile As String, sTitle As String sFile = Application.DefaultFilePath & "\brig.htm" sTitle = "Tabelle Saison " & Year(Date) & "/" & Year(Date) + 1 Close Open sFile For Output As #1 Print #1, "<html>" Print #1, "<head>" Print #1, "<title>" & sTitle & "</title>" Print #1, "<style type=text/css>" Print #1, "td { font-size: 8pt; font-family: tahoma,verdana,sans-serif }" Print #1, "th { font-size: 8pt; font-family: tahoma,verdana,sans-serif }" Print #1, "</style>" Print #1, "</head>" Print #1, "<body bgcolor=#00e0ff>" Print #1, "<center>" Print #1, "<table border=2 cellpadding=3 cellspacing=3 bgcolor=#ffffe0>" Print #1, " <tr bgcolor=#00ffe0><th colspan=9 align=center>" Print #1, " " & sTitle Print #1, " </th></tr>" Print #1, " <tr bgcolor=#00ffe0>" Print #1, " <th>Pl.</th><th>Verein</th><th>Spiele</th><th>g</th><th>u</th>" Print #1, " <th>v</th><th>Tore</th><th>Diff</th><th>Punkte</th>" Print #1, " </tr>" iRow = 4 Do Until IsEmpty(Cells(iRow, 1)) Print #1, " <tr>" For iCol = 1 To 6 If iCol <> 2 Then Print #1, " <td align=center>" & Cells(iRow, iCol).Value & "</td>" Else Print #1, " <td>" & Cells(iRow, iCol).Value & "</td>" End If Next iCol Print #1, " <td align=center>" Print #1, Cells(iRow, 7).Value & Cells(iRow, 8).Value & Cells(iRow, 9).Value Print #1, " </td>" For iCol = 10 To 11 Print #1, " <td align=center>" & Cells(iRow, iCol).Value & "</td>" Next iCol Print #1, " </tr>" iRow = iRow + 1 Loop Print #1, "</table>" Print #1, "</center>" Print #1, "<hr color=ff0000>" Print #1, "hans@herber.de" Print #1, "</body>" Print #1, "</html>" Close MsgBox "Die Tabelle wurde gespeichert unter:" & vbLf & sFile End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Web-Aktienkursabfrage bei YAHOO

Problem:	Wie kann ich eine Web-Aktienkursabfrage bei YAHOO vornehmen und die Ergebnisse gezielt in ein Datenblatt übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) On Error Resume Next Application.CommandBars("Data").Controls("Yahoo").Delete On Error GoTo 0 End Sub Private Sub Workbook_Open() Dim oBtn As CommandBarButton With Application.CommandBars("Data") On Error Resume Next .Controls("Yahoo").Delete On Error GoTo 0 Set oBtn = .Controls.Add End With With oBtn .Caption = "Yahoo" .FaceId = 610 .BeginGroup = True .OnAction = "Abruf" End With End Sub StandardModule: Modul1 Sub Abruf() Dim wks As Worksheet Dim var As Variant Dim sWkn As String, sQuery As String Application.ScreenUpdating = False Set wks = ActiveSheet sWkn = InputBox(_ prompt:="WKN-Nr.:", _ Title:="Web-Abfrage", _ Default:="840400") If sWkn = "" Then Exit Sub sQuery = "http://de.finance.yahoo.com/q?m=*&s=" & sWkn & "&d=v1" Worksheets.Add after:=Worksheets(Worksheets.Count) With ActiveSheet.QueryTables.Add(Connection:= _ "URL;" & sQuery, _ Destination:=Range("A1")) .Refresh BackgroundQuery:=False End With var = Application.Match(sWkn & "*", Columns(1), 0) wks.Range("A2").Value = Cells(var, 2).Value wks.Range("B2").Value = Cells(var, 1).Value wks.Range("C2").Value = Cells(var, 3).Value wks.Range("D2").Value = Cells(var, 5).Value wks.Range("E2").Value = Cells(var, 6).Value wks.Range("F2").Value = Cells(var, 7).Value wks.Range("G2").Value = Cells(var, 8).Value wks.Range("H2").Value = Now Application.DisplayAlerts = False ActiveSheet.Delete Application.DisplayAlerts = True Worksheets("Daten").Select Range("A1").Select Columns("A:G").AutoFit Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kommentarvorspanne löschen

Problem:	Die Kommentarvorspanne der Kommentare sollen gelöscht werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub EditComment() Dim cmt As Comment Dim sCmt As String, sWhat As String, sReplace As String sWhat = "Hans W. Herber:" & vbCrLf sReplace = "" For Each cmt In ActiveSheet.Comments sCmt = cmt.Text sCmt = WorksheetFunction.Substitute(sCmt, sWhat, sReplace) With cmt.Shape.TextFrame.Characters .Text = sCmt .Font.Bold = False End With Next cmt End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlinkadressen zu Kommentaren konvertieren

Problem: Die Hyperlink-Adressen aus Spalte A sollen in einen Kommentar in Zelle B1 eingefügt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Hype2Comment()
    Dim cmt As Comment
    Dim iRow As Integer
    Dim sTxt As String
    iRow = 1
    Do Until IsEmpty(Cells(iRow, 1))
        sTxt = sTxt & Cells(iRow, 1).Hyperlinks(1).Address & vbCrLf
        iRow = iRow + 1
    Loop
    With Range("B1")
        If Not .Comment Is Nothing Then
            .Comment.Delete
        End If
        Set cmt = .AddComment(sTxt)
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Farbänderung beim Kommentar-Indikator

Problem:	Die Farbe des Kommentar-Indikators soll geändert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub test_fncCreateCommentIndicator() fncCreateCommentIndicator vbBlue, "pat" End Sub Private Function fncCreateCommentIndicator(_ CommentIndicatorColor As Long, _ CommentIndicatorName As String) As Boolean Dim IDnumber As Long Dim aCell As Range Dim aComment As Comment Dim aShape As Shape Dim aWorksheet As Worksheet Dim aWorkbook As Workbook fncCreateCommentIndicator = False If CommentIndicatorName = vbNullString Then GoTo ExitFunction On Error GoTo ExitFunction Set aWorkbook = ActiveWorkbook IDnumber = 0 For Each aWorksheet In aWorkbook.Worksheets For Each aShape In aWorksheet.Shapes If Left(aShape.Name, Len(CommentIndicatorName)) = _ CommentIndicatorName Then aShape.Delete End If Next aShape For Each aComment In aWorksheet.Comments Set aCell = aComment.Parent If InStr(1, aComment.Shape.TextFrame.Characters.Text, ":") > 0 Then If Left(aComment.Shape.TextFrame.Characters.Text, _ InStr(1, aComment.Shape.TextFrame.Characters.Text, ":") - 1) = _ Application.UserName Then GoSub CreateCommentIndicator End If End If Next aComment Next aWorksheet fncCreateCommentIndicator = True ExitFunction: On Error GoTo 0 Set aCell = Nothing Set aComment = Nothing Set aShape = Nothing Set aWorksheet = Nothing Set aWorkbook = Nothing Exit Function CreateCommentIndicator: Set aShape = aWorksheet.Shapes.AddShape(Type:=msoShapeRightTriangle, _ Left:=aCell.Left + aCell.Width - 5, _ Top:=aCell.Top, _ Width:=5, _ Height:=5) IDnumber = IDnumber + 1 With aShape .Name = CommentIndicatorName & CStr(IDnumber) .IncrementRotation -180# .Fill.Visible = msoTrue .Fill.Solid .Fill.ForeColor.RGB = CommentIndicatorColor .Line.Visible = msoTrue .Line.Weight = 1 .Line.Style = msoLineSingle .Line.DashStyle = msoLineSolid .Line.ForeColor.RGB = CommentIndicatorColor .Placement = xlMove End With Return End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kommentare löschen und deren Texte in Zellen übernehmen

Problem:	Die Kommentare in Spalte C sollen gelöscht und als Texte in Spalte D übernommen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SplitComments() Dim cmt As Comment Dim iRow As Integer For iRow = 1 To WorksheetFunction.CountA(Columns(3)) Set cmt = Cells(iRow, 3).Comment If Not cmt Is Nothing Then Cells(iRow, 4) = Cells(iRow, 3).Comment.Text Cells(iRow, 3).Comment.Delete End If Next iRow End Sub Sub a() Range("A1").comm End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kommentarfenster in bestimmter Größe erstellen

Problem:	Wie kann ich einen Kommentar in einer bestimmten Größe erstellen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetComment() Dim cmt As Comment Application.DisplayCommentIndicator = xlCommentAndIndicator Set cmt = ActiveCell.AddComment With cmt .Text Text:=Application.UserName & ":" _ & Chr(10) & "" With .Shape .Height = 80 .Width = 50 End With End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schriftart für Kommentar ändern

Problem: Wie kann ich auf Buttondruck ein Kommentarfeld für die aktive Zelle in einer bestimmt Schriftgröße erstellen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle3

Private Sub CommandButton1_Click()
    Dim cmt As Comment
    ActiveCell.Select
    Application.DisplayCommentIndicator = xlCommentAndIndicator
    Set cmt = ActiveCell.AddComment
    cmt.Shape.Select
    With cmt.Shape.TextFrame.Characters.Font
        .Name = "Arial"
        .Size = 14
    End With
End Sub

Private Sub CommandButton2_Click()
    Application.DisplayCommentIndicator = xlCommentIndicatorOnly
End Sub

StandardModule: Modul1

Sub SetComment()
    Dim cmt As Comment
    Application.DisplayCommentIndicator = xlCommentIndicatorOnly
    If Not ActiveCell.Comment Is Nothing Then
        ActiveCell.Comment.Delete
    End If
    Set cmt = ActiveCell.AddComment("Mein Kommentar")
    With cmt.Shape.TextFrame
        With .Characters.Font
            .Name = "Arial"
            .Size = 14
            .Bold = False
        End With
        .AutoSize = True
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kommentare in geschützte Tabellen einfügen

Problem: Wie kann ich bei Blattschutz Kommentare nur in ungeschützte Zellen eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub SetComment()
    Dim oCmt As Comment
    Application.DisplayCommentIndicator = xlCommentIndicatorOnly
    If ActiveCell.Locked Then
        MsgBox _
            prompt:="Eingabe eines Kommentars nicht möglich -" & _
            vbLf & "dieser Bereich ist geschuetzt. ", _
            Buttons:=vbCritical
    Else
        ActiveSheet.Unprotect
        If Not ActiveCell.Comment Is Nothing Then
            ActiveCell.Comment.Delete
        End If
        ActiveCell.AddComment "Mein Kommentar"
        ActiveSheet.Protect
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen auf Kommentar überprüfen

Problem:	Wie kann ich die Zellen eines ausgewählten Bereiches überprüfen, ob sie einen Kommentar enthalten und wenn ja, diesen ändern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdComment_Click() Dim rng As Range For Each rng In Selection If Not rng.Comment Is Nothing Then rng.NoteText "Hallo!" End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anwendername im Kommentar nicht fettgedruckt

Problem: Wie kann ich erreichen, daß der Anwendername in einem Kommentar nicht fett gedruckt wird?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub NewComment()
    Dim sComment As String
    sComment = InputBox( _
        prompt:="Bitte Kommentar eingeben:")
    If sComment = "" Then Exit Sub
    With ActiveCell
        .AddComment
        .Comment.Visible = False
        .Comment.Text _
            Text:=Application.UserName & _
            ":" & vbCrLf & sComment
    End With
    Application.DisplayCommentIndicator = _
        xlCommentIndicatorOnly
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fußzeile ein- und auslesen

Problem:	Aus der Fußzeile soll der Name hinter dem Doppelpunkt in der letzten Zeile in die obige TextBox eingetragen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub PartLeftFooter() Dim intCounter As Integer, intChr As Integer Dim sTxt As String sTxt = ActiveSheet.PageSetup.LeftFooter Do Until InStr(sTxt, vbCrLf) = 0 For intCounter = Len(sTxt) To 1 Step -1 If Asc(Mid(sTxt, intCounter, 1)) = 10 Then Exit Do intChr = Asc(Mid(sTxt, intCounter, 1)) If intCounter = 1 Then Exit Sub Next intCounter Loop sTxt = Right(sTxt, Len(sTxt) - intCounter) sTxt = Trim(Right(sTxt, Len(sTxt) - InStr(sTxt, ":"))) ActiveSheet.TextBox1.Text = sTxt End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datum in Fußzeile einfügen

Problem:	Wie kann ich das gestrige Datum in die Fußzeile einfügen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FooterYesterday() With ActiveSheet .PageSetup.LeftFooter = Format(Date - 1, "dd.mm.yy") .PrintPreview End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachauswahl an auszuwählende Positionen kopieren

Problem:	Eine Mehrfachauswahl soll in das Blatt Ziel an auszuwählende Stellen kopiert werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub MultipleRanges() Dim colRng As New Collection Dim rng As Range, rngTarget As Range Dim iCounter As Integer Dim sTarget As String For Each rng In Selection.Areas colRng.Add rng Next rng Worksheets("Ziel").Select For iCounter = 1 To colRng.Count Set rngTarget = Application.InputBox("Bitte Zielbereich auswählen", Type:=8) colRng(iCounter).Copy rngTarget Next iCounter Application.CutCopyMode = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachauswahl in ein zweites Blatt kopieren

Problem:	Eine Mehrfachauswahl soll in das Blatt Ziel an identische Stelle kopiert werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub MultipleRanges() Dim rng As Range For Each rng In Selection.Areas Worksheets("Ziel").Range(rng.Address).Value = rng.Value Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile mit letzter Fundstelle an letzte Position der Liste

Problem: Die Zeile mit der letzten Fundstelle eines Suchbegriffs soll an letzte Position der Tabelle kopiert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Auswahl()
    Dim rng As Range, rngLast As Range
    Dim lRow As Long
    Dim sBegriff As String, sAddress As String
    sBegriff = InputBox( _
        prompt:="Bitte Suchbegriff eingeben:", _
        Default:="Hallo")
    If sBegriff = "" Then Exit Sub
    Set rng = Cells.Find( _
        what:=sBegriff, _
        lookat:=xlWhole, _
        LookIn:=xlValues, _
        MatchCase:=False, _
        after:=ActiveCell)
    If rng Is Nothing Then
        Beep
        MsgBox "Suchbegriff nicht gefunden!", , _
            Application.UserName
        Exit Sub
    End If
    sAddress = rng.Address
    rng.Select
    rng.Offset(1).Select
    Do
        Set rngLast = ActiveCell
        Cells.FindNext(after:=ActiveCell).Activate
        If ActiveCell.Address = sAddress Then Exit Do
    Loop
    lRow = Range("A1").CurrentRegion.Rows.Count + 1
    Rows(lRow).Value = rngLast.EntireRow.Value
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachauswahl in zweite Tabelle kopieren

Problem:	Über Merfachauswahl selektierte Zeilen sollen in Tabelle1 kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CopyRowValues() Dim wks As Worksheet Dim rng As Range Dim iRowT As Integer Set wks = Worksheets("Tabelle2") For Each rng In Selection.Areas If IsEmpty(wks.Cells(1, 1)) Then iRowT = 1 Else iRowT = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1 End If wks.Rows(iRowT & ":" & iRowT + rng.Rows.Count - 1).Value = rng.Value Next rng wks.Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in andere Tabelle umorganisieren und kopieren

Problem:	Die Werte aus Spalte A:B sollen in Tabelle2 komprimiert angeordnet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Transformieren() Dim vRow As Variant Dim iRow As Integer, iRowL As Integer, iCol As Integer iRowL = Cells(Rows.Count, 1).End(xlUp).Row iCol = 1 With Worksheets("Tabelle2") For iRow = 1 To iRowL If Not IsNumeric(Cells(iRow, 1)) Then iCol = iCol + 1 .Cells(1, iCol).Value = Cells(iRow, 2).Value Else vRow = Application.Match(Cells(iRow, 1).Value, .Columns(1), 0) If Not IsError(vRow) Then .Cells(vRow, iCol).Value = Cells(iRow, 2).Value End If End If Next iRow End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile kopieren, wenn Wert in Spalte Q

Problem:	Es wird geprüft, ob in Spalte Q ein Wert enthalten ist. Wenn ja, wird die komplette Zeile in das Blatt Target übernommen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CopyValues() Dim lRow As Long, lRowL As Long, lRowT As Long lRowL = Cells(Rows.Count, 17).End(xlUp).Row lRowT = 1 For lRow = 2 To lRowL If Not IsEmpty(Cells(lRow, 17)) Then lRowT = lRowT + 1 Worksheets("Target").Rows(lRowT).Value = Rows(lRow).Value End If Next lRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte als Positiv-Werte kopieren

Problem:	Die Werte aus der letzten Spalte sollen als Positivwerte in die nächste Spalte kopiert werden..
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub PositivKopieren() Dim iRow As Integer, iCol As Integer iCol = Range("IV1").End(xlToLeft).Column iRow = 1 Do Until IsEmpty(Cells(iRow, iCol)) If IsNumeric(Cells(iRow, iCol).Value) Then Cells(iRow, iCol + 1).Value = Abs(Cells(iRow, iCol).Value) End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen gem. vorgegebener Anzahl kopieren

Problem:	Die Tabellenzeilen sollen in das Blatt "Resultat" gem. der in Spalte D angegebenen Anzahl kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Eintragen() Dim iRow As Integer, iRowT As Integer, iCounter As Integer Dim iCol As Integer iCol = Cells(1, 256).End(xlToLeft).Column iRow = 2 iRowT = 1 Worksheets("Resultat").Cells.ClearContents Worksheets("Resultat").Rows(1).Value = Rows(1).Value Do Until IsEmpty(Cells(iRow, 1)) For iCounter = 1 To Cells(iRow, iCol).Value iRowT = iRowT + 1 Worksheets("Resultat").Rows(iRowT).Value = Rows(iRow).Value Next iCounter iRow = iRow + 1 Loop MsgBox "Job erledigt!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte solange kopieren, wie Ganzzahl in Spalte A

Problem:	Die Werte aus Spalte G sollen dann in Spalte H kopiert werden, wie in Spalte A eine Ganzzahl vorkommt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CopyValues() Dim iRow As Integer iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) If IsNumeric(Cells(iRow, 1).Value) Then If Cells(iRow, 1).Value - Fix(Cells(iRow, 1).Value) = 0 Then Cells(iRow, 8).Value = Cells(iRow, 7).Value End If End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Solange kopieren, wie ein Wert in einer Spalte vorkommt

Problem:	Die Werte aus Spalte G sollen solange in Spalte H kopiert werden, wie in Spalte A der Wert 11 vorkommt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CopyValues() Dim dValue As Double Dim iRow As Integer dValue = 11 iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) If Cells(iRow, 1).Value = dValue Then Do While Cells(iRow, 1).Value = dValue Cells(iRow, 8).Value = Cells(iRow, 7).Value iRow = iRow + 1 Loop End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Zeilen mit einem Suchbegriff in nächstes Blatt kopieren

Problem:	Alle Zeilen, in denen ein "WES" vorkommt, sollen in das Blatt "Target" kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub TestCopy() Dim wks As Worksheet Dim iRow As Integer, iRowL As Integer, iRowT As Integer Set wks = Worksheets("Target") iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL If InStr(Cells(iRow, 9).Value, "WES") Then iRowT = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1 Rows(iRow).Copy wks.Rows(iRowT) End If Next iRow wks.Columns.AutoFit Application.CutCopyMode = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Den jeweils darüberliegenden Wert in Leerzellen kopieren

Problem: Die Leerzellen in Spalte A sollen mit dem jeweils darüberstehenden Wert gefüllt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub CopyCells()
    Dim iRow As Integer
    iRow = 2
    Do Until Cells(iRow, 1).Value = "Stop"
        If IsEmpty(Cells(iRow, 1)) Then
            Cells(iRow, 1).Value = Cells(iRow - 1, 1).Value
        End If
        iRow = iRow + 1
    Loop
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktive Zeile kopieren und löschen

Problem: Die Daten der Zeile, in der sich der Cursor befindet, werden in Blatt B übertragen und die Zeile mit der aktiven Zelle wird gelöscht.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Uebertragen()
    Dim iRow As Integer
    With Worksheets("B")
        iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
        ActiveCell.EntireRow.Copy Worksheets("B").Rows(iRow)
        ActiveCell.EntireRow.Delete
        .Columns.AutoFit
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benannte Bereich aus mehreren Arbeitsmappen importieren

Problem: Aus den Dateien des in Zelle B1 genannten Verzeichnisses soll jeweils ein Bereich mit dem Namen "others" importiert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub ImportOthers()
    Dim fs As FileSearch
    Dim lRow As Long
    Dim iCounter As Integer
    Application.ScreenUpdating = False
    Set fs = Application.FileSearch
    lRow = 1
    With fs
        .Filename = "*.xls"
        .LookIn = Range("B1").Value
        .Execute
        For iCounter = 1 To .FoundFiles.Count
            Cells(lRow, 1).Value = .FoundFiles(iCounter)
            lRow = lRow + 1
            Workbooks.Open .FoundFiles(iCounter)
            Range("other").Copy ThisWorkbook.Worksheets(1).Cells(lRow, 1)
            lRow = lRow + Range("other").Rows.Count + 2
            ActiveWorkbook.Close savechanges:=False
        Next iCounter
    End With
    Application.ScreenUpdating = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gruppenüberschriften in Nebenspalten übertragen

Problem:	Die Zeilen mit Datum und nebenstehendem Text sollen jeweils als Vorgängerspalten zu den darunterstehenden Gruppen angeordnet werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub UmOrdnen() Dim iRowL As Integer, iRow As Integer Columns("A:B").Insert iRowL = Cells(Rows.Count, 3).End(xlUp).Row For iRow = iRowL To 1 Step -1 If IsEmpty(Cells(iRow, 5)) Then Range(Cells(iRow + 1, 1), Cells(iRow + 1 + _ WorksheetFunction.CountA(Cells(iRow, 3) _ .CurrentRegion.Columns(1)) - 2, 2)).Value = _ Range(Cells(iRow, 3), Cells(iRow, 4)).Value Rows(iRow).Delete iRow = iRow - 1 End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert von einer Zelle zur anderen in mehreren Tabellen übernehmen

Problem:	In allen Tabellenblättern ab dem 2. Blatt soll der Wert in Zelle A5 gegen den in Zelle A1 getauscht werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetValue() Dim wks As Worksheet For Each wks In Worksheets If wks.Index <> 1 Then wks.Range("A1").Value = wks.Range("A5").Value End If Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten einer Tabelle an eine 2. Tabelle übergeben

Problem:	Die Daten des Bereiches A5:E8 sollen der Tabelle im Blatt Daten angehängen werden. In der Spalte rechts neben den Werten in der Zieltabelle ist das aktuelle Datum einzutragen.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Uebertragen() Dim lRow As Long With Worksheets("Daten") lRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1 Range("A5:E8").Copy .Cells(lRow, 1) .Range(.Cells(lRow, 6), .Cells(lRow + 3, 6)).Value = Date End With Application.CutCopyMode = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahlen eines Bereiches jeweils 4 mal in eine Spalte schreiben.

Problem:	Die Zahlen im Bereich B1:Y3 sollen jeweils 4 mal in Spalte AA geschrieben werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Aufteilen() Dim iRow As Integer, iCol As Integer Dim iCounter As Integer, iAct As Integer For iRow = 1 To 3 For iCol = 2 To 25 For iAct = 1 To 4 iCounter = iCounter + 1 Cells(iCounter, 27).Value = Cells(iRow, iCol).Value Next iAct Next iCol Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die letzte Zeile des aktiven Blattes in andere Arbeitsmappe

Problem:	Die letzte Zeile des aktiven Blattes soll in die erste freie Zeile einer zu öffnenden Arbeitsmappe kopiert werden. Diese wird dann geschlossen und gespeichert.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub DatenKopie() Dim wks Dim iRow As Integer, iRowS As Integer Dim sfile As String Application.ScreenUpdating = False sfile = ThisWorkbook.Path & "\test1.xls" If Dir(sfile) = "" Then Beep MsgBox "Testdatei wurde nicht gefunden!" Exit Sub End If Set wks = ActiveSheet Workbooks.Open Filename:=sfile Worksheets("175802").Select iRowS = wks.Cells(Rows.Count, 1).End(xlUp).Row If IsEmpty(Range("A1")) Then iRow = 1 Else iRow = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1 End If wks.Rows(iRowS).Copy Rows(iRow) ActiveWorkbook.Close savechanges:=True Application.ScreenUpdating = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle in neue Blätter nach Gruppen aufteilen.

Problem:	Eine Tabelle mit Gruppennamen in Spalte A soll in neu anzulegende Tabellenblätter entsprechend den Gruppennamen aufgeteilt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SortSplit() Dim wks As Worksheet Dim iRow As Integer, iRowT As Integer Application.ScreenUpdating = False Set wks = ActiveSheet Range("A1").Sort key1:=Range("A2"), order1:=xlAscending, header:=xlYes iRow = 2 Do Until IsEmpty(wks.Cells(iRow, 1)) If Left(wks.Cells(iRow, 1), 1) <> Left(wks.Cells(iRow - 1, 1), 1) Then Worksheets.Add after:=Worksheets(Worksheets.Count) Rows(1).Value = wks.Rows(1).Value Rows(1).Font.Bold = True Columns(1).Font.Bold = True ActiveSheet.Name = Left(wks.Cells(iRow, 1), 1) iRowT = 1 End If iRowT = iRowT + 1 Rows(iRowT).Value = wks.Rows(iRow).Value iRow = iRow + 1 Loop Worksheets(1).Select Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Zeilen ohne Wert in vorgegebener Spalte kopieren

Problem:	Alle Zeilen, die in Spalte G keinen Wert enthalten, sollen in die zweite Tabelle kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CopyEmptys() Dim iRow As Integer, iRowL As Integer, iRowT As Integer iRowL = Cells(Rows.Count, 2).End(xlUp).Row For iRow = 2 To iRowL If IsEmpty(Cells(iRow, 7)) Then With Worksheets("02.01.02") iRowT = .Cells(.Rows.Count, 2).End(xlUp).Row + 1 .Range(.Cells(iRowT, 2), .Cells(iRowT, 7)).Value = _ Range(Cells(iRow, 2), Cells(iRow, 7)).Value End With End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur Zeilen mit bestimmtem Inhalt kopieren

Problem: Es sollen alle Zeilen in Tabelle2 kopiert werden, in denen die Zeichenfolgen "vb0" vorkommt.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Filtern()  
    Dim wks As Worksheet  
    Dim rng As Range  
    Dim iRow As Integer, iRowT As Integer  
    Application.ScreenUpdating = False  
    Set wks = Worksheets("Tabelle2")  
    Set rng = Range("A1").CurrentRegion  
    rng.Rows(1).Copy wks.Range("A1")  
    iRow = 2  
    iRowT = 1  
    Do Until IsEmpty(Cells(iRow, 1))  
        If WorksheetFunction.CountIf(rng.Rows(iRow), "vb0") > 0 Then  
            iRowT = iRowT + 1  
            rng.Rows(iRow).Copy wks.Cells(iRowT, 1)  
        End If  
        iRow = iRow + 1  
    Loop  
    wks.Columns.AutoFit  
    Application.CutCopyMode = False  
    Application.ScreenUpdating = False  
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopieren und Umbenennen eines Blattes

Problem: Das Blatt Tab_Ges soll kopiert und umbenannt werden. Im Zielblatt ist der Bereich "Bereich" zu löschen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub WksCopy()  
    Const csWks As String = "Tab_Ges"  
    Worksheets(csWks).Copy _  
        after:=Worksheets(Worksheets.Count)  
    Application.DisplayAlerts = False  
    Worksheets(csWks).Delete  
    Application.DisplayAlerts = True  
    ActiveSheet.Name = csWks  
    Range("Bereich").ClearContents  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufteilen einer Tabelle auf mehrere Arbeitsblätter

Problem: Die Daten aus obiger Tabelle sollen gemäß den Nummern in Spalte A auf neue Arbeitsblätter verteilt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub EgaleSpeichern()  
    Dim rng As Range, rngCur As Range  
    Dim lngRow As Long  
    Application.ScreenUpdating = False  
    Set rngCur = Range("A1").CurrentRegion  
    rngCur.Sort _  
        key1:=Range("A2"), _  
        order1:=xlAscending, _  
        header:=xlYes  
    lngRow = 2  
    Do Until IsEmpty(rngCur.Cells(lngRow, 1))  
        If rngCur.Cells(lngRow, 1) <> rngCur.Cells(lngRow - 1, 1) Then  
            rngCur.AutoFilter _  
                field:=1, _  
                Criteria1:=rngCur.Cells(lngRow, 1)  
            Set rng = rngCur.SpecialCells(xlCellTypeVisible)  
            Worksheets.Add.Move after:=Worksheets(Worksheets.Count)  
            ActiveSheet.Name = rngCur.Cells(lngRow, 1)  
            rng.Copy Range("A1")  
        End If  
        lngRow = lngRow + 1  
    Loop  
    Worksheets(1).Select  
    ActiveSheet.AutoFilterMode = False  
    Application.ScreenUpdating = False  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattinhalte dreier Blätter in neues Blatt kopieren

Problem:	Die Daten in den Tabellen 1 bis 3 sollen in eine neue Tabelle kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Kopieren() Dim iCounter As Integer, iRow As Integer Application.ScreenUpdating = False Worksheets.Add after:=Worksheets(Worksheets.Count) iRow = 1 For iCounter = 2 To 4 Worksheets(iCounter).Range("A1").CurrentRegion.Copy _ Cells(iRow, 1) iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 Next iCounter Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten kopieren wenn kein Datum oder Datum > Heute

Problem:	Aus dem Datenbereich dieser Tabelle sollen alle Zeilen, in denen Spalte C leer oder das Datum größer als das aktuelle ist, in Tabelle2 kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Makro1() Makro1 Makro Makro am 06.01.2001 von Hans W. Herber aufgezeichnet Selection.AutoFilter Selection.AutoFilter Field:=3, Criteria1:="", Operator:=xlAnd, _ Criteria2:=">06.01.2001" Selection.AutoFilter Field:=3, Criteria1:="=""", Operator:=xlAnd, _ Criteria2:=">06.01.2001" Selection.AutoFilter Field:=3, Criteria1:="", Operator:=xlOr, Criteria2 _ :=">06.01.2001" End Sub StandardModule: basMain Sub FilternKopieren() Range("A1").AutoFilter _ Field:=3, Criteria1:="", Operator:=xlOr, _ Criteria2:=">" & CDb1(Date) Range("A1").CurrentRegion _ .SpecialCells(xlCellTypeVisible).Copy _ Worksheets("Tabelle2").Range("A1") ActiveSheet.AutoFilterMode = False End Sub</pre>



Werte nach Kriterien prüfen und auf Blätter verteilen

Problem:	Die Werte aus den Spalten A:B sollen in neue Tabellenblätter kopiert und nach Gruppen bis 100, > 100 bis 1000, > 1000 bis 10000, > 10000 bis 100000 und über 100000 aufgeteilt werden. Für jede Gruppe soll ein einzelnes Tabellenblatt angelegt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Sub NachWertKopieren() Dim wksSource As Worksheet, wksTarget As Worksheet Dim rng As Range Dim arr(1 To 4) As Double Dim iCounter As Integer, iRow As Integer, iCount As Integer Application.ScreenUpdating = False Set wksSource = ActiveSheet arr(1) = 100 arr(2) = 1000 arr(3) = 10000 arr(4) = 100000 For iCounter = 1 To 5 If iCounter = 1 Then wksSource.Range("A1").AutoFilter Field:=2, _ Criteria1:="<=" & CStr(arr(iCounter)) ElseIf iCounter < 5 Then wksSource.Range("A1").AutoFilter Field:=2, _ Criteria1:=">" & CStr(arr(iCounter - 1)), _ Operator:=xlAnd, Criteria2:="<=" & _ CStr(arr(iCounter)) Else wksSource.Range("A1").AutoFilter Field:=2, _ Criteria1:=">" & CStr(arr(iCounter - 1)) End If iCount = WorksheetFunction.Subtotal(2, _ wksSource.Range("A1").CurrentRegion) If iCount > 0 Then Set rng = wksSource.Range("A1").CurrentRegion _ .SpecialCells(xlCellTypeVisible) On Error Resume Next Set wksTarget = Worksheets(wksSource _ .AutoFilter.Filters(2).Criteria1) If Err > 0 Or wksTarget Is Nothing Then Err.Clear Worksheets.Add.Move after:=Worksheets(Worksheets.Count) ActiveSheet.Name = _ wksSource.AutoFilter.Filters(2).Criteria1 rng.Copy Range("A3") Rows(3).Delete Else iRow = wksTarget.Cells(Rows.Count, 1).End(xlUp).Row + 1 rng.Copy wksTarget.Cells(iRow, 1) wksTarget.Rows(iRow).Delete End If On Error GoTo 0 End If Next iCounter wksSource.Select ActiveSheet.AutoFilterMode = False Application.ScreenUpdating = True End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte prüfen und kopieren

Problem: Ist eine Artikelnummer in Spalte A noch nicht vorhanden, soll die nächsthöhere Zahl als Artikelnummer eingesetzt und der Wertebereich der Zeile in das Blatt "Ziel" kopiert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub MatchCopy()  
    Dim var As Variant  
    Dim lRowL As Integer, lRow As Integer, lRowD, lRowS As Long  
    Dim bln As Boolean  
    lRowL = Cells(Rows.Count, 2).End(xlUp).Row  
    For lRow = 1 To lRowL  
        bln = False  
        If Not IsEmpty(Cells(lRow, 1)) Then  
            var = Application.Match(Cells(lRow, 1), _  
                Worksheets("Ziel").Columns("A"), 0)  
        Else  
            Cells(lRow, 1).Value = WorksheetFunction.Max(Columns(1)) + 1  
            bln = True  
        End If  
        If IsError(var) Or bln = True Then  
            With Worksheets("Ziel")  
                If IsEmpty(.Cells(1, 1)) Then  
                    lRowS = 1  
                Else  
                    lRowS = .Cells(Rows.Count, 1).End(xlUp).Row + 1  
                End If  
                .Range(.Cells(lRowS, 1), .Cells(lRowS, 4)).Value = _  
                    Range(Cells(lRow, 1), Cells(lRow, 4)).Value  
            End With  
        Else  
            With Worksheets("Vorhanden")  
                If IsEmpty(.Cells(1, 1)) Then  
                    lRowD = 1  
                Else  
                    lRowD = .Cells(Rows.Count, 1).End(xlUp).Row + 1  
                End If  
                .Range(.Cells(lRowD, 1), .Cells(lRowD, 4)).Value = _  
                    Range(Cells(lRow, 1), Cells(lRow, 4)).Value  
            End With  
        End If  
    Next lRow  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten übertragen oder addieren

Problem:	Die Daten aus Tabelle1 sollen in Tabelle2 übertragen werden. Nach der Übertragung ist in Spalte B der jeweiligen Zeile ein X einzutragen. Dieser Datensatz ist bei der nächsten Übertragung nicht mehr zu berücksichtigen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub PruefenUebertragen() Dim var As Variant Dim lRow As Long, lRowL As Long lRow = 1 Do Until IsEmpty(Cells(lRow, 1)) With Worksheets("Tabelle2") If Cells(lRow, 2).Value > 0 And Cells(lRow, 2) <> "X" Then var = Application.Match(Cells(lRow, 1).Value, .Columns(1), 0) If IsError(var) Then lRowL = .Cells(Rows.Count, 1).End(xlUp).Row + 1 .Cells(lRowL, 1) = Cells(lRow, 1) .Cells(lRowL, 2) = Cells(lRow, 2) .Cells(lRowL, 3) = Cells(lRow, 3) Else .Cells(var, 2) = .Cells(var, 2) + Cells(lRow, 2) End If Cells(lRow, 2) = "X" End If lRow = lRow + 1 End With Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur sichtbare Zellen in anderes Arbeitsblatt kopieren

Problem:	Im Bereich A1:B6 befinden sich ausgeblendete Zeilen und Spalten. Es sollen nur die sichtbaren Werte in ein zweites Tabellenblatt kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MinCopy() Range("A1").CurrentRegion _ .SpecialCells(xlCellTypeVisible).Copy _ Worksheets("Tabelle2").Range("A1") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert der aktiven Spalte an nächste freie Position einfügen

Problem: Der Wert aus Spalte B der Zeile, in der sich der Cursor befindet, soll an die nächste freie Position im Blatt Spesen2 in Spalte B eingefügt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Einfuegen()
    With Worksheets("Spesen2")
        .Cells(.Cells(Rows.Count, 2) _
            .End(xlUp).Row + 1, 2).Value = _
            Cells(ActiveCell.Row, 2).Value
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nicht zusammenhängenden Bereich in nächste Zeile kopieren

Problem:	Wie kann ich einen nicht zusammenhängenden Bereich in die erste freie Zeile eines andere Tabellenblattes kopieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MehrFachAuswahl() Dim rngAct As Range Dim intRow As Integer, intCol As Integer With Worksheets("Tabelle2") If IsEmpty(.Cells(1, 1)) Then intRow = 1 Else intRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1 End If For Each rngAct In Range("A1,C2,E5,A6").Cells intCol = intCol + 1 rngAct.Copy .Cells(intRow, intCol) Next rngAct End With Application.CutCopyMode = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Jede zweite Zeile kopieren

Problem:	Wie kann ich alle Zeilen ab einer durch InputBox-Eingabe festgelegten in ein zweites Arbeitsblatt kopieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZeilenKopieren() Dim iEnd As Integer, iRow As Integer Dim sStart As String Application.ScreenUpdating = False sStart = InputBox("Startzeile:", , 3) iEnd = Cells(Rows.Count, 1).End(xlUp).Row For iRow = CInt(sStart) To iEnd Step 2 Rows(iRow).Hidden = True Next iRow Rows(sStart & ":" & iEnd).SpecialCells(xlCellTypeVisible).Copy _ Worksheets("Tabelle2").Range("A1") Worksheets("Tabelle2").Columns.AutoFit Rows.Hidden = False Application.CutCopyMode = False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen einfügen, Werte und Bezüge zu den Werten eintragen

Problem:	Wie kann ich am Tabellenanfang von 2 Tabellen eine durch InputBox ermittelte Anzahl von Zeilen einfügen, in der einen Tabelle Werte und in der anderen Bezüge zu den Werten eintragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZeilenFormelnEinfuegen() Dim var As Variant Dim iRow As Integer Application.ScreenUpdating = False var = InputBox("Anzahl der Zeilen:", , 3) If var = "" Then Exit Sub Rows("1:" & var).Insert Worksheets("Tabelle2").Rows("1:" & var).Insert For iRow = 1 To var Worksheets("Tabelle2").Cells(iRow, 1).Value = iRow Cells(iRow, 1).Formula = "=Tabelle2!" & _ Cells(iRow, 1).Address Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über InputBox festzulegenden Bereich kopieren

Problem:	Wie kann ich einen Kopierbereich über InputBox auswählen und kopieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Kopieren() Dim rngSource As Range, rngTarget As Range Set rngSource = Application.InputBox _ ("Kopierbereich auswählen:", Type:=8) Worksheets("Ziel").Select Set rngTarget = Application.InputBox _ ("Zielbereich eintragen:", Type:=8) If rngTarget.Cells.Count > 1 Then If rngTarget.Rows.Count <> rngSource.Rows.Count Or _ rngTarget.Columns.Count <> _ rngSource.Columns.Count Then Beep MsgBox "Quell- und Zielbereich " & _ "müssen die gleiche Grösse haben!" Exit Sub End If End If rngSource.Copy rngTarget Application.CutCopyMode = False Range("A1").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorgegebene Anzahl von Einträgen in 2. Blatt kopieren

Problem:	Wie kann ich aus dem aktiven Blatt eine bestimmte Anzahl von letzten Zellen in ein zweites Blatt übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FindenUndKopieren() Dim rng As Range Set rng = Cells(Rows.Count, 1).End(xlUp) If rng.Row < 5 Then Exit Sub Range(rng, rng.Offset(-4)).Copy _ Worksheets("Tabelle2").Range("A1") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen in Abhängigkeit des Wertes in Spalte A kopieren

Problem:	Wie kann ich veranlassen, dass nur die Zeilen in eine neue Arbeitsmappe kopiert werden, der Wert in Spalte A einer Bedingung entspricht?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub WerteFiltern() Dim wks As Worksheet Dim iRow As Integer, iRowL As Integer, iRowT As Integer Application.ScreenUpdating = False Set wks = ActiveSheet iRowL = Cells(Rows.Count, 1).End(xlUp).Row Workbooks.Add For iRow = 1 To iRowL With wks If IsNumeric(Left(.Cells(iRow, 1), 6)) Then If CDbl(Left(.Cells(iRow, 1), 6)) > 990000 And _ CDbl(Left(.Cells(iRow, 1), 6)) < 1000000 Then iRowT = iRowT + 1 .Range(.Cells(iRow, 2), .Cells(iRow, 7)).Copy _ Cells(iRowT, 1) End If End If End With Next iRow Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile mit aktiver Zelle in anderes Blatt kopieren

Problem: Wie kann ich die Zeile, in dem sich der Cursor befindet, in eine zweite Tabelle kopieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub ZeilenKopieren()
    Dim wks As Worksheet
    Dim iRow As Integer
    Set wks = Worksheets("Tabelle2")
    iRow = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1
    Rows(ActiveCell.Row).Copy wks.Rows(iRow)
    Application.CutCopyMode = False
    wks.Columns.AutoFit
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus Zelle in Tabellenblatt-Textbox eintragen

Problem: Wie kann ich den Wert aus einer Zelle per VBA in eine Tabellenblatt-TextBox eintragen lassen? Erzwungene Zeilenumbrüche sollen erhalten bleiben.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub InTextBox()
    Worksheets("Tabelle2").TextBoxes(1).Text = _
        Range("A1").Text
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen der Leute mit dem höchsten Umsatz übertragen

Problem:	Wie kann ich aus einer Tabelle, in der Namen und Umsätze stehen, den oder die Namen mit dem höchsten Umsatz in einer Spalte listen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MaxUmsatz() Dim iRow As Integer, iTarget As Integer iRow = 1 iTarget = 2 Do Until IsEmpty(Cells(iRow, 1)) If Cells(iRow, 2) = Application.Max(Columns(2)) Then Cells(iTarget, 3) = "Max. Umsatz hat Herr " & _ Cells(iRow, 1) & ": " & _ Format(Cells(iRow, 2), "#,##0.00") iTarget = iTarget + 1 End If iRow = iRow + 1 Loop Columns(3).AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus Tabelle1 mit zusätzlichen Leerzeilen in Tabelle2

Problem:	Wie kann ich Zeilen aus Tabelle 1 so in Tabelle 2 einfügen, dass jeweils eine Leerzeile hinzugefügt wird?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Kopieren() Dim wksSource As Worksheet, wksTarget As Worksheet Set wksSource = Worksheets("Tabelle1") Set wksTarget = Worksheets("Tabelle2") Dim iRowT As Integer, iRowS As Integer iRowT = 1 iRowS = 1 Do Until IsEmpty(wksSource.Cells(iRowS, 1)) wksSource.Rows(iRowS).Copy wksTarget.Rows(iRowT) iRowS = iRowS + 1 iRowT = iRowT + 2 Loop wksTarget.Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenwerte nach Kriterien auf Arbeitsblätter verteilen

Problem:	Wie kann ich in einer Tabelle enthaltene Werte nach ebenfalls in dieser Tabelle enthaltenen Kriterien auf verschiedene Arbeitsblätter verteilen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: modMain Sub Verteilen() Dim shAct As Worksheet Dim intRow As Integer, intRowT As Integer intRow = 2 Do Until IsEmpty(Cells(intRow, 1)) If Rows(intRow).Hidden = False Then Set shAct = Worksheets(Cells(intRow, 36).Value) intRowT = shAct.Cells(Rows.Count, 1).End(xlUp).Row + 1 Rows(intRow).Copy shAct.Rows(intRowT) End If intRow = intRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Geschriebene Rechnung in Journal eintragen

Problem: Wie kann ich eine geschriebene Rechnung in ein Journal eintragen und das Rechnungsbatt für eine neue Rechnung initialisieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub NeueRechnung()  
    Dim wksJournal As Worksheet  
    Dim iRowT As Integer, iRowS As Integer, iCounter As Integer  
    Set wksJournal = Worksheets("Journal")  
    With Worksheets("Journal")  
        iRowT = .Cells(.Rows.Count, 1).End(xlUp).Row + 1  
        iRowS = 28  
        For iCounter = 6 To 8  
            .Cells(iRowT, iCounter) = Cells(46, iCounter)  
        Next iCounter  
        Do Until IsEmpty(Cells(iRowS, 1))  
            .Cells(iRowT, 1) = Range("H19")  
            .Cells(iRowT, 2) = Range("H21")  
            .Cells(iRowT, 3) = Range("H20")  
            .Cells(iRowT, 4) = Cells(iRowS, 1)  
            .Cells(iRowT, 5) = Cells(iRowS, 4)  
            iRowS = iRowS + 1  
            iRowT = iRowT + 1  
        Loop  
    End With  
    ActiveSheet.Copy  
    ThisWorkbook.Activate  
    Range("A8:A13, H20,A28:H45").ClearContents  
    Range("H19") = Range("H19") + 1  
    Range("H21") = Date  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Überschriften aus Spalte C in Spalte D in den relevanten Bereich

Problem:	In Spalte C habe ich in unregelmäßigen Abständen Überschriften angebracht. Wie kann ich diese in Spalte D in den jeweils relevanten Bereich übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Bearbeiten() Dim iRowL As Integer, iCounter As Integer, iWeek As Integer iRowL = Cells(Rows.Count, 3).End(xlUp).Row For iCounter = 1 To iRowL If Cells(iCounter, 2) = "Woche" Then iWeek = Cells(iCounter, 3) iCounter = iCounter + 1 End If If Not IsEmpty(Cells(iCounter, 3)) Then Cells(iCounter, 4) = iWeek End If Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalte aus Einzelblättern in Deckblatt übertragen

Problem: Wie kann ich Werte aus Einzelblättern in ein Deckblatt übertragen und in eine zusätzliche Spalte den jeweiligen Quellblattnamen eintragen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub cmdAktualisierung_Click()  
    Dim rng As Range  
    Dim iRow As Integer, iWks As Integer  
    Dim iCount As Integer, iCol As Integer  
    For iWks = 1 To 4  
        Set rng = Worksheets("Abteilung" & iWks).Range("A1").CurrentRegion  
        iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1  
        iCount = rng.Rows.Count  
        iCol = rng.Columns.Count  
        Range(Cells(iRow, 1), Cells(iRow + iCount - 2, iCol)).Value = _  
            rng.Rows("2:" & iCount).Value  
        Range(Cells(iRow, iCol + 1), _  
            Cells(iRow + iCount - 2, iCol + 1)).Value = rng.Parent.Name  
    Next iWks  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Teilergebnisse einer Serie von Tabellen in anderes Blatt kopieren

Problem:	Wie kann ich aus einer Serie von Tabellenblättern alle Teilergebnisse in ein gesondertes Blatt kopieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub TeilErgebnisseKopieren() Dim wksSource As Worksheet, wksTarget As Worksheet Dim rng As Range Dim iCounter As Integer, iCounterC As Integer Dim iRowS As Integer, iRowT As Integer Set wksTarget = Worksheets(Worksheets.Count) wksTarget.Cells.ClearContents For iCounter = 1 To Worksheets.Count - 1 Set wksSource = Worksheets(iCounter) For iCounterC = 1 To wksSource.Range("A5"). _ CurrentRegion.Columns.Count iRowS = 7 Do Until IsEmpty(wksSource.Cells(iRowS, iCounterC)) Set rng = wksSource.Cells(iRowS, iCounterC) If InStr(rng.FormulaLocal, "TEILERGEBNIS") Then iRowT = iRowT + 1 wksTarget.Cells(iRowT, iCounterC) = rng End If iRowS = iRowS + 1 Loop iRowT = 0 Next iCounterC Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte nur mit Zahlenformaten übertragen

Problem:	Wie kann ich Werte einer Zelle in eine andere Zelle übernehmen, ohne die Hintergrund- und Schriftfarbe mit zu übertragen? Die Zahlenformate sollen erhalten bleiben.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Uebertragen() Dim rngSource, rngTarget Set rngSource = Application.InputBox(_ prompt:="Quellbereich auswählen:", Type:=8) Set rngTarget = Application.InputBox(_ prompt:="Zielbereich auswählen:", Type:=8) rngSource.Copy rngTarget.PasteSpecial Paste:=xlValues rngTarget.NumberFormat = rngSource.NumberFormat rngSource.ClearContents End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte zu Zieladressen übertragen

Problem:	Wie kann ich Werte aus Spalte A an die in Spalte B stehenden Zieladressen übertragen und dabei berücksichtigen, daß - wenn sich an der Zieladresse bereits ein Wert befindet, dieser zu dem neuen Wert hinzuaddiert wird.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub UebertragenUndAddieren() Dim iCounter As Integer iCounter = 1 Do Until IsEmpty(Cells(iCounter, 1)) If IsEmpty(Range(Cells(iCounter, 2).Value)) Then Range(Cells(iCounter, 2).Value).Value = _ Cells(iCounter, 1).Value Else Range(Cells(iCounter, 2).Value).Value = _ Range(Cells(iCounter, 2).Value).Value + _ Cells(iCounter, 1).Value End If iCounter = iCounter + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Formel bis zur letzten Zeile der Nebenspalte kopieren

Problem:	Wie kann ich eine Formel in Spalte B bis zur letzten Zelle mit Inhalt in Spalte A kopieren? In Spalte A sind die Zellen nicht durchgehend belegt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BisZurLetztenZeile() Dim iRow As Integer iRow = Cells(Rows.Count, 1).End(xlUp).Row Range("B2").Formula = "=sum(\$A\$1:\$A2)" Range("B2:B" & iRow).FillDown End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensätze nach zwei Kriterien in neues Blatt übernehmen

Problem:	Wie kann ich Datensätze, die 2 Kriterien erfüllen, in ein zweites Tabellenblatt übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DatenFiltern() Dim iRow As Integer, iRowT As Integer iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) If Cells(iRow, 1).Value >= 1000 And _ Cells(iRow, 1).Value <= 2000 And _ Cells(iRow, 2).Value = "Hose" Then iRowT = iRowT + 1 Worksheets("Ziel").Rows(iRowT).Value = _ Rows(iRow).Value End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Übernahme von Werten mit "x" in eine zweite Tabelle

Problem: Wie kann ich alle Werte, die ein "x" oder "X" beinhalten, in eine zweite Tabelle übernehmen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Uebernehmen()  
    Dim iRow As Integer, iRowT As Integer  
    iRow = 1  
    Do Until IsEmpty(Cells(iRow, 1))  
        If InStr(UCase(Cells(iRow, 1).Value), "X") Then  
            iRowT = iRowT + 1  
            Worksheets("Ziel").Cells(iRowT, 1).Value = _  
                Cells(iRow, 1).Value  
        End If  
        iRow = iRow + 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Kopieren auch die Zeilenhöhe und Spaltenbreite übernehmen

Problem:	Wie kann ich beim Kopieren eines Bereiches auch die jeweiligen Zeilenhöhen und Spaltenbreiten mit übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SuperKopieren() Dim rngSource As Range, rngTarget As Range Dim iCounter As Integer Set rngSource = Worksheets("Tabelle1").Range("A1:C5") Set rngTarget = Worksheets("Tabelle2").Range("A1:C5") rngSource.Copy rngTarget For iCounter = 1 To rngSource.Rows.Count rngTarget.Rows(iCounter).RowHeight = _ rngSource.Rows(iCounter).RowHeight Next iCounter For iCounter = 1 To rngSource.Columns.Count rngTarget.Columns(iCounter).ColumnWidth = _ rngSource.Columns(iCounter).ColumnWidth Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in alle Arbeitsblätter eintragen

Problem: Wie kann ich über VBA Werte in einer Zeile in die nachfolgenden Arbeitsblättern eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub AlleBlaetter()
    Dim arr As Variant
    arr = Array("Tabelle1", "Tabelle2", "Tabelle3")
    Worksheets(arr).FillAcrossSheets Rows(1)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrmals vorkommende Werte in neues Tabellenblatt

Problem:	Wie kann ich in einer Spalte mehrfach vorkommende Werte in ein neues Arbeitsblatt per VBA übertragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub GleicheUebertragen() Dim wks As Worksheet Dim var As Variant, vValue As Variant Dim iRow As Integer, iRowT As Integer Set wks = ActiveSheet Worksheets.Add.Move after:=Worksheets(Worksheets.Count) iRow = 1 Do Until IsEmpty(wks.Cells(iRow, 1)) vValue = wks.Cells(iRow, 1).Value If WorksheetFunction.CountIf(wks.Columns(1), vValue) > 1 Then var = Application.Match(vValue, Columns(1), 0) If IsError(var) Then iRowT = iRowT + 1 Cells(iRowT, 1).Value = vValue End If End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte ohne Zwischenspeicher übertragen

Problem: Wie kann ich Werte von einem zum anderen Bereich übertragen, ohne die Zwischenablage nutzen zu müssen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Kopieren()
    With Range("A1:B2")
        Range("F1:G2").Value = .Value
        Range("F1:G2").NumberFormat = .NumberFormat
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gefilterte Daten ohne Zwischenablage übertragen

Problem:	Wie kann ich über AutoFilter gefilterte Daten in ein zweites Tabellenblatt übertragen, ohne sie in die Zwischenablage zu übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: mHWH Sub Uebertragen() Dim iCounter As Integer, iRow As Integer For iCounter = 2 To WorksheetFunction.CountA(Columns(1)) If Rows(iCounter).Hidden = False Then iRow = iRow + 1 Worksheets("Tabelle2").Cells(iRow, 1).Value = _ Cells(iCounter, 1).Value End If Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blätter in andere Arbeitsmappen kopieren

Problem: Wie kann ich makrogesteuert Blätter aus einer in eine andere Arbeitsmappe kopieren und dort als letztes Blatt einordnen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub SheetCopy()
    Dim wkb As Workbook
    Set wkb = Workbooks.Add
    ThisWorkbook.Worksheets(1).Copy _
        after:=wkb.Worksheets(wkb.Worksheets.Count)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Terminplananpassung bei Änderung des Kalenderjahres

Problem:	Bei Änderung des Jahres in Zelle F1 werden die nach Wochentagen festgelegten Aufgaben neu zugewiesen.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	

Spielplan generieren	
Problem:	Es wird ein Spielplan generiert. In einer neuen Arbeitsmappe werden die Spiele eingetragen.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: Modul1 <pre>Sub Main() Dim rngTeams As Range, rngPairs As Range Dim iRow As Integer, iCount As Integer Application.ScreenUpdating = False Range("C:E").ClearContents iRow = Cells(Rows.Count, 1).End(xlUp).Row Set rngTeams = Range(Cells(2, 1), Cells(iRow, 1)) iCount = rngTeams.Cells.Count Call Paarungen(rngTeams, iCount) Set rngPairs = Range("C1").CurrentRegion Call PlanAnlegen(rngPairs, iCount) Application.ScreenUpdating = True End Sub Sub Paarungen(rng As Range, iCount As Integer) Dim iRow As Integer, iA As Integer, iB As Integer, iC As Integer, iD As Integer Dim bln As Boolean iRow = 1 For iA = 1 To iCount For iB = iA To iCount If iA <> iB Then bln = Not bln If bln Then Cells(iRow, 3).Value = rng(iA) Cells(iRow, 4).Value = rng(iB) Else Cells(iRow, 4).Value = rng(iA) Cells(iRow, 3).Value = rng(iB) End If Cells(iRow, 5).Value = 0 iRow = iRow + 1 End If Next iB Next iA End Sub Sub PlanAnlegen(rng As Range, iCount As Integer) Dim rngCellA As Range, rngCellB As Range, rngAct As Range Dim iRow As Integer, iCounter As Integer, iAct As Integer Dim bln As Boolean Workbooks.Add 1 Range("A1").Value = "Spielplan" For iRow = 3 To (iCount + 1) * (iCount + 1) / 2 Step (iCount / 2) + 2 iCounter = iCounter + 1 Cells(iRow, 1).Value = iCounter & ". Spieltag" Range(Cells(iRow + 1, 1), Cells(iRow + (iCount - 1) / 2, 2)).Name = "Tag" & iCounter Next iRow For iCounter = 1 To iCount - 1 For Each rngAct In rng.Columns(1).Cells If rngAct.Offset(0, 2).Value = 0 Then Set rngCellA = Range("Tag" & iCounter).Find(rngAct.Value, LookIn:=xlValues, lookat:=xlWhole) Set rngCellB = Range("Tag" & iCounter).Find(rngAct.Offset(0, 1).Value, LookIn:=xlValues, lookat:=xlWhole) If rngCellA Is Nothing And rngCellB Is Nothing Then iAct = 1 Do Until IsEmpty(Range("Tag" & iCounter).Columns(1).Cells(iAct)) iAct = iAct + 1 Loop Range("Tag" & iCounter).Cells(iAct, 1).Value = rngAct.Value Range("Tag" & iCounter).Cells(iAct, 2).Value = rngAct.Offset(0, 1).Value rngAct.Offset(0, 2).Value = 1 End If End If Next rngAct Next iCounter Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppel aus Liste entfernen

Problem: Über eine benutzerdefinierte Funktion werden aus einem selektierten Bereich alle Doppel entfernt. Das Ergebnis wird in Spalte B geschrieben. Die Funktion ist universell in Office-Anwendungen einsetzbar.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub Onlys()  
    Dim arr() As Variant  
    Dim arrOnlys As Variant  
    Dim iRow As Integer, iCount As Integer  
    iCount = Selection.Cells.Count  
    If iCount = 1 Then  
        Beep  
        MsgBox "Sie müssen mindestens 2 Zellen auswählen!"  
        Exit Sub  
    End If  
    ReDim arr(1 To iCount)  
    For iRow = 1 To iCount  
        arr(iRow) = Selection.Cells(iRow).Value  
    Next iRow  
    arrOnlys = GetOnlys(arr)  
    Columns(2).ClearContents  
    For iRow = 1 To UBound(arrOnlys)  
        Cells(iRow, 2).Value = arrOnlys(iRow)  
    Next iRow  
End Sub  
  
Private Function GetOnlys(arr As Variant) As Variant  
    Dim col As New Collection  
    Dim arrNew()  
    Dim iRow As Integer  
    On Error Resume Next  
    For iRow = LBound(arr) To UBound(arr)  
        col.Add Item:=arr(iRow), key:=arr(iRow)  
    Next iRow  
    On Error GoTo 0  
    ReDim Preserve arrNew(1 To col.Count)  
    For iRow = 1 To col.Count  
        arrNew(iRow) = col(iRow)  
    Next iRow  
    GetOnlys = arrNew  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile einfügen, wenn 100er-Sprung

Problem:	Wenn in der Zahlenreihe eine neuer 1000er Abschnitt beginnt, wird ein Text eingefügt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetHeaders() Dim lRow As Long, lRowL As Long Application.ScreenUpdating = False lRowL = Cells(Rows.Count, 1).End(xlUp).Row For lRow = lRowL To 2 Step -1 If lRow = 2 Then Rows(lRow).Insert Cells(lRow, 1).Value = "Start " & Fix(Cells(lRow + 1, 1).Value / 10000) * 10000 ElseIf Fix(Cells(lRow, 1).Value / 10000) <> _ Fix(Cells(lRow - 1, 1).Value / 10000) Then Rows(lRow).Insert Cells(lRow, 1).Value = "Start " & Fix(Cells(lRow + 1, 1).Value / 10000) * 10000 End If Next lRow Range("A2").Font.Bold = False Columns.AutoFit Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In Monatsblättern gelistete Werte nach Produkten addieren

Problem:	Die in den Monatsblättern gelisteten Werte sind in diesem Blatt nach Produkten addiert aufzulisten.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetValues() Dim vProduct As Variant Dim iMonth As Integer, iRow As Integer, iWks As Integer, iRowT As Integer Dim iCol As Integer Rows(1).Font.Bold = True Columns(1).Font.Bold = True Range("A1").Value = "Products" For iMonth = 1 To 12 Cells(1, iMonth + 1).Value = Format(DateSerial(1, iMonth, 2), "mmm") Next iMonth For iWks = 3 To Worksheets.Count For iCol = 1 To 12 vProduct = Application.Match(_ Worksheets(iWks).Cells(1, iCol).Value, Columns(1), 0) If IsError(vProduct) Then iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 Cells(iRow, 1).Value = Worksheets(iWks).Cells(1, iCol).Value Cells(iRow, iWks - 1).Value = Worksheets(iWks).Cells(16, iCol).Value Else Cells(vProduct, iWks - 1).Value = Cells(vProduct, iWks - 1).Value + _ Worksheets(iWks - 1).Cells(16, iCol).Value End If Next iCol Next iWks Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Artikelbenennungen entsprechend ihrer Stückzahl listen

Problem:	Die Artikel aus einer Tabelle sollen in Tabelle2 entsprechend ihrer Stückzahlen gelistet werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub JeStueck() Dim iRow As Integer, iRowT As Integer, iCounter As Integer iRow = 2 iRowT = 2 Do Until IsEmpty(Cells(iRow, 1)) For iCounter = 1 To Cells(iRow, 1).Value With Worksheets("Tabelle2") .Cells(iRowT, 1).Value = Cells(iRow, 2).Value .Cells(iRowT, 2).Value = Cells(iRow, 3).Value End With iRowT = iRowT + 1 Next iCounter iRow = iRow + 1 Loop End Sub</pre>

	Herbers Excel-FAQ - Die Texte Bestandteil der Excel-Lehr- und Lernmaterialien
Sammeln von Monatsblatt-Werten in einer Zusammenfassung	
Problem:	Die Werte aus Spalte A aus Monatsblättern sollen in das Tabellenblatt "Gesamt" in die entsprechende Monatsspalte eingetragen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) With Application.CommandBars("Worksheet Menu Bar") On Error Resume Next .Controls("&Monate").Delete On Error GoTo 0 End With End Sub Private Sub Workbook_Open() Dim objPopUp As CommandBarPopup Dim objBtn As CommandBarButton With Application.CommandBars("Worksheet Menu Bar") On Error Resume Next .Controls("&Monate").Delete On Error GoTo 0 Set objPopUp = .Controls.Add(_ Type:=msoControlPopup, _ before:=.Controls.Count, _ temporary:=True) End With objPopUp.Caption = "&Monate" Set objBtn = objPopUp.Controls.Add With objBtn .Caption = "&In Sammelblatt" .OnAction = "InSammelblatt" .Style = msoButtonCaption End With Set objBtn = objPopUp.Controls.Add With objBtn .Caption = "&Beenden" .OnAction = "EndeMonate" .Style = msoButtonCaption End With End Sub StandardModule: Modul1 Sub MonatAnlegen() Dim iMonth As Integer For iMonth = 1 To 12 Worksheets.Add after:=Worksheets(iMonth) ActiveSheet.Name = Format(DateSerial(1, iMonth, 1), "mmmm") Next iMonth Worksheets(1).Select End Sub Sub EndeMonate() ThisWorkbook.Close End Sub Sub InSammelblatt() Dim rng As Range Dim iCol As Integer If ActiveSheet.Index > 12 Then Beep MsgBox "Sie müssen sich in einem Monatsblatt befinden!" Exit Sub End If Set rng = Range("A2:A" & WorksheetFunction.CountA(Columns(1))) With Worksheets(13) iCol = WorksheetFunction.Match(ActiveSheet.Name, .Rows(1), 0) .Range(.Cells(2, iCol), .Cells(Rows.Count, iCol)).ClearContents .Range(.Cells(2, iCol), .Cells(rng.Rows.Count + 1, iCol)).Value = rng.Value End With End Sub



Auflistung aller Kombinationsmöglichkeiten

Problem:	100 Kugeln sind von 1 bis 100 beschriftet. 3 Kugeln werden gezogen. Alle Kombinationsmöglichkeiten werden aufgelistet.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	StandardModule: Modul1 Sub CombiTest() Dim arr(1 To 100) As Integer Dim lAll As Long, lCount As Long Dim iCounter As Integer Dim iCol As Integer Dim iA As Integer, iB As Integer, iC As Integer Dim sTxt As String, sPath As String Dim bln As Boolean Application.ScreenUpdating = False bln = Application.DisplayStatusBar Application.DisplayStatusBar = True sPath = Application.DefaultFilePath & "\" lCount = WorksheetFunction.Combin(100, 3) For iCounter = 1 To 100 arr(iCounter) = iCounter Next iCounter Close Open sPath & "\combin.txt" For Output As #1 For iA = 1 To 98 For iB = 2 To 99 For iC = 3 To 100 lAll = lAll + 1 If lAll Mod 1000 = 0 Then Application.StatusBar = _ "Bearbeite Kombination " & Format(lAll, "#,##0") & _ " von " & Format(lCount, "#,##0") & "..." If iA >= iB Then iB = 1 + iA If iB >= iC Then iC = iB + 1 If iCol < 3 Then iCol = iCol + 1 sTxt = sTxt & arr(iA) & " " & arr(iB) & " " & arr(iC) & "," Else iCol = 1 Print #1, Left(sTxt, Len(sTxt) - 1) sTxt = arr(iA) & " " & arr(iB) & " " & arr(iC) & "," End If Next Next Next Print #1, sTxt Close Application.StatusBar = "Lade die Daten..." Workbooks.OpenText Filename:=sPath & "combin.txt", _ Origin:=xlWindows, _ DataType:=xlDelimited, _ ConsecutiveDelimiter:=False, _ Tab:=False, _ Semicolon:=False, _ Comma:=True, _ Space:=False, _ Other:=False Application.StatusBar = False Application.DisplayStatusBar = bln Application.ScreenUpdating = True End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Liste in Abhängigkeit einer Gültigkeitsauswahl aktualisieren

Problem:	Die in der Kopfzeile von Tabelle2 stehenden Team-Namen sollen in einem Gültigkeitsfeld in Zelle A1 zur Auswahl gestellt werden. Bei Auswahl eines Teams, soll die Crew dieses Teams in Spalte A angezeigt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datenbereich in zweiter Tabelle ohne Duplikate listen

Problem: Wie kann ich den Datenbereich aus Tabelle1 in Spalte A von Tabelle2 ohne Berücksichtigung von Duplikaten untereinander auflisten lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub KopierenOhneDoppelte()  
    Dim wks As Worksheet  
    Dim iRow As Integer, iCol As Integer, iRowT As Integer  
    Set wks = Worksheets("Tabelle2")  
    iRow = 1  
    iCol = 1  
    wks.Columns("A").ClearContents  
    Do Until IsEmpty(Cells(iRow, iCol))  
        Do Until IsEmpty(Cells(iRow, iCol))  
            If WorksheetFunction.CountIf( _  
                wks.Columns(1), Cells(iRow, iCol).Value) = 0 Then  
                iRowT = iRowT + 1  
                wks.Cells(iRowT, 1).Value = Cells(iRow, iCol).Value  
            End If  
            iCol = iCol + 1  
        Loop  
        iRow = iRow + 1  
        iCol = 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Rechnungsformular mit Zugriff auf Kunden- und Artikelstamm

Problem:	Wie kann ich ein Rechnungsformular mit direkten Zugriff auf einen Kunden- und Artikelstamm entwickeln?
Lösung:	Nur anhand einer Beispieldarstellungsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Listenzusammenstellung anhand von Schlüsselnummern

Problem:	Gegeben ist eine Auflistung von Schlüsselnummern von Firmen, an die Faxe gesandt werden sollen. Die Firmendaten stehen im Blatt Faxnummer. In diesem Blatt soll anhand der Schlüsselnummern der Verteiler ermittelt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FaxNummern() Dim wksAS As Worksheet, wksFax As Worksheet Dim varA As Variant, varB As Variant Set wksAS = Worksheets("AS400") Set wksFax = Worksheets("Faxnummern") Dim iRow As Integer, iRowL As Integer iRow = 2 With wksAS Do Until IsEmpty(.Cells(iRow, 1)) varA = Application.VLookup(.Cells(iRow, 1), _ wksFax.Columns("A:C"), 3, 0) If Not IsError(varA) Then varB = Application.Match(varA, Columns("B"), 0) If IsError(varB) Then iRowL = Cells(Rows.Count, 1).End(xlUp).Row + 1 Cells(iRowL, 1).Value = _ Application.VLookup(.Cells(iRow, 1), _ wksFax.Columns("A:C"), 2, 0) Cells(iRowL, 2).Value = varA Cells(iRowL, 3).Value = 1 Else Cells(varB, 3).Value = Cells(varB, 3).Value + 1 End If End If iRow = iRow + 1 Loop End With Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In Array einlesen, sortieren, auslesen

Problem:	Die Werte aus diesem Tabellenblatt sollen in ein Array eingelesen, von dort in eine temporäre Arbeitsmappe aus-, dort sortiert, dann wieder in das Array eingelesen und letztendlich in Tabelle2 ausgelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub InArrOutArr() Dim rng As Range Dim arr As Variant Application.ScreenUpdating = False Set rng = Range("A1").CurrentRegion arr = rng.Value Workbooks.Add Range("A1").Range(rng.Address).Value = arr Range("A1").Sort key1:=Range("A1"), _ order1:=xlAscending, header:=xlNo arr = Range("A1").CurrentRegion.Value ActiveWorkbook.Close savechanges:=False With Worksheets("Tabelle2") .Range(rng.Address).Value = arr .Select End With Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte bei Übereinstimmung in 2 Spalten sammeln

Problem: Wenn die Werte in Spalte A und B übereinstimmen, sollen die Daten spaltenweise gesammelt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Sammeln()  
    Dim iRow As Integer, iRowL As Integer  
    Dim iCol As Integer, iAct As Integer  
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row  
    For iRow = iRowL To 1 Step -1  
        iCol = 3  
        For iAct = iRow - 1 To 1 Step -1  
            If Cells(iAct, 1) = Cells(iRow, 1) And _  
                Cells(iAct, 2) = Cells(iRow, 2) Then  
                Do Until IsEmpty(Cells(iRow, iCol))  
                    Cells(iAct, _  
                        WorksheetFunction.CountA _  
                        (Rows(iAct & ":" & iAct)) + 1) = _  
                        Cells(iRow, iCol)  
                    iCol = iCol + 1  
                Loop  
                If iCol > 3 Then Rows(iRow).Delete  
            Exit For  
        End If  
    Next iAct  
Next iRow  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einlesen der Autokorrekturliste in eine Tabelle

Problem:	Wie kann ich mir die Autokorrekturliste in eine Tabelle einlesen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub AutoKorrektur() Dim arr As Variant Dim iCounter As Integer arr = Application.AutoCorrect.ReplacementList For iCounter = 1 To UBound(arr) Cells(iCounter, 1) = arr(iCounter, 1) Cells(iCounter, 2) = arr(iCounter, 2) Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen mit Hintergrundfarbe in neuer Arbeitsmappe auflisten

Problem:	Wie kann ich alle Zellen mit Hintergrundfarbe einer Tabelle in einem Arbeitsblatt einer neuen Arbeitsmappe mit Angaben über Wert, Adresse, Hintergrundfarbe und Farbindes auflisten lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MarkierungFinden() Dim rngAct As Range, rngSource As Range Dim iRow As Integer Application.ScreenUpdating = False Set rngSource = ActiveSheet.Range("A1:P100") Workbooks.Add For Each rngAct In rngSource If rngAct.Interior.ColorIndex <> xlNone Then iRow = iRow + 1 Cells(iRow, 1) = rngAct.Address(False, False) Cells(iRow, 2) = rngAct.Value Cells(iRow, 3).Interior.ColorIndex = _ rngAct.Interior.ColorIndex Cells(iRow, 4) = Cells(iRow, 3).Interior.ColorIndex End If Next rngAct Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, welche Tabellenblätter aus einer Liste vorhanden sind

Problem:	Wie kann ich prüfen, welche Tabellenblätter aus einer Liste in Spalte A in der aktiven Arbeitsmappe enthalten sind?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: modMain Sub TabellenPruefen() Dim wks As Worksheet Dim rng As Range For Each wks In ThisWorkbook.Worksheets On Error Resume Next Set rng = Worksheets("Tabelle1").Columns("A").Find(_ wks.Name, lookat:=xlWhole, LookIn:=xlValues) If Err > 0 Or rng Is Nothing Then Err.Clear Else rng.Offset(0, 1) = "X" End If On Error GoTo 0 Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelt vorkommende Werte mit Zelleadresse auflisten

Problem: Wie kann ich alle in einem Tabellenblatt doppelt vorkommenden Werte in einem neuen Arbeitsblatt unter Angabe der Zelladressen auflisten lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub ListDoubles()
    Dim wks As Worksheet
    Dim rng As Range
    Dim iRow As Integer
    Set wks = ActiveSheet
    Worksheets.Add after:=Worksheets(Worksheets.Count)
    For Each rng In wks.UsedRange
        If WorksheetFunction.CountIf(wks.UsedRange, rng.Value) > 1 Then
            iRow = iRow + 1
            Cells(iRow, 1).Value = rng.Value
            Cells(iRow, 2).Value = rng.Address(False, False)
        End If
    Next rng
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Lücken aus Zahlenreihe listen

Problem: Wie kann ich die Lücken in einer in Spalte A stehenden Zahlenreihe in Spalte B auflisten?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub LueckenListen()
    Dim iRow As Integer, iRowT As Integer
    Dim lTmp As Long
    Columns(2).ClearContents
    iRow = 1
    iRowT = 1
    Do Until IsEmpty(Cells(iRow + 1, 1))
        lTmp = Cells(iRow, 1)
        Do Until lTmp = Cells(iRow + 1, 1) - 1
            lTmp = lTmp + 1
            Cells(iRowT, 2) = lTmp
            iRowT = iRowT + 1
        Loop
        iRow = iRow + 1
    Loop
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten in umgekehrter Reihenfolge übertragen

Problem:	Wie kann ich Werte aus einer Tabelle in einem zweiten Arbeitsblatt suchen und die Werte der Nebenspalte dorthin übertragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub cmdStart_Click() Dim wks As Worksheet Dim var As Variant Dim iRow As Integer Set wks = Worksheets("Ziel") iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) var = Application.Match(Cells(iRow, 1).Value, _ wks.Columns(1), 0) If Not IsError(var) Then wks.Cells(var, 2) = Cells(iRow, 2).Value End If iRow = iRow + 1 Loop wks.Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Buchungsliste in Arbeitsmappe mit anzulegenden Konten verteilen

Problem: Wie kann ich Buchungssätze aus einer Buchungsliste auf Konten in einer neuen Arbeitsmappe verteilen? Die Konten müssen neu angelegt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Aufteilen()
    Dim wks As Worksheet, wksTarget As Worksheet
    Dim iRow As Integer, iCol As Integer, iRowL As Integer
    Set wks = ActiveSheet
    iRow = 2
    Workbooks.Add 1
    Do Until IsEmpty(wks.Cells(iRow, 1))
        On Error Resume Next
        Set wksTarget = Worksheets(wks.Cells(iRow, 1).Text)
        If Err > 0 Or wksTarget Is Nothing Then
            Err.Clear
            Worksheets.Add after:=Worksheets(Worksheets.Count)
            ActiveSheet.Name = wks.Cells(iRow, 1).Text
            Set wksTarget = ActiveSheet
            iRowL = 1
        Else
            iRowL = wksTarget.Cells(Rows.Count, 1).End(xlUp).Row + 1
        End If
        For iCol = 1 To 3
            wksTarget.Cells(iRowL, iCol).Value = wks.Cells(iRow, iCol + 1).Value
        Next iCol
        iRow = iRow + 1
    Loop
    Application.DisplayAlerts = False
    Worksheets(1).Delete
    Application.DisplayAlerts = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verstreute Daten in verschiedenen Tabellen zusammenfassen

Problem: Wie kann ich auf mehrere Tabellen verstreute Daten in einer zusätzlichen Tabelle zeilenweise zusammenfassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Auflisten()
    Dim iCounter As Integer, iRow As Integer
    For iCounter = 3 To Worksheets.Count
        iRow = iRow + 1
        Cells(iRow, 1) = Worksheets(iCounter).Range("A1")
        Cells(iRow, 2) = Worksheets(iCounter).Range("B3")
        Cells(iRow, 3) = Worksheets(iCounter).Range("D4")
    Next iCounter
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle mehrfach vorkommenden Datensätze in zweiter Tabelle listen

Problem: Wie kann ich alle mehrfach vorkommenden Datensätze einer Tabelle in einer zweiten Tabelle auflisten lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Vergleich()
    Dim lRow As Long
    Dim iRow As Integer
    Application.ScreenUpdating = False
    lRow = 2
    iRow = 1
    Range("A1").CurrentRegion.AdvancedFilter _
        Action:=xlFilterInPlace, Unique:=True
    Do Until IsEmpty(Cells(lRow, 1))
        If Rows(lRow).Hidden Then
            iRow = iRow + 1
            Worksheets("Tabelle2").Rows(iRow).Value = Rows(lRow).Value
        End If
        lRow = lRow + 1
    Loop
    ActiveSheet.ShowAllData
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gültigkeitsprüfung nach Liste aus zweiter Tabelle

Problem: Wie kann ich als Grundlage einer Gültigkeitsprüfung eine Liste aus einer zweiten Tabelle zugrundelegen? Immer dann, wenn sich die Liste ändert, sollen diese Änderungen auch bei der Gültigkeitsprüfung berücksichtigt werden.

Lösung: Nur anhand einer Beispielarbeitsmappe darstellbar.

VBA-Code:

```
ClassModule: Tabelle3

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    If Target.Column <> 1 Then Exit Sub
    Range("A1").CurrentRegion.Columns(1).Name = "Quelle"
End Sub

StandardModule: modMain

Sub Gueltigkeit()
    With Selection.Validation
        .Add _
            Type:=xlValidateList, _
            AlertStyle:=xlValidAlertStop, _
            Operator:=xlBetween, _
            Formula1:="=Quelle"
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus einer Liste von Mannschaften mögliche Paarungen zusammenstellen

Problem:	Wie kann ich aus einer Liste von Mannschaften die möglichen Spielpaarungen zusammenstellen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Schaltfläche1_BeiKlick() Dim iSpiel As Integer, iStart As Integer Dim iAktuell As Integer, iAnzahl As Integer Columns("A:B").ClearContents iAnzahl = Application.CountA(Columns(4)) iStart = 1 iAktuell = 2 Do Until iStart > iAnzahl - 1 iSpiel = iSpiel + 1 Cells(iSpiel, 1) = Cells(iStart, 4) Cells(iSpiel, 2) = Cells(iAktuell, 4) If iAktuell = iAnzahl Then iStart = iStart + 1 iAktuell = iStart + 1 Else iAktuell = iAktuell + 1 End If Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl in einer Liste vorkommender Namen zählen

Problem:	Wie kann ich die Anzahl in einer Tabellenspalte vorkommender Namen ermitteln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub nNamen() Dim rng As Range Dim iNames As Integer Application.ScreenUpdating = False Set rng = Range("A1").CurrentRegion Workbooks.Add rng.AdvancedFilter Action:=xlFilterCopy, _ CopyToRange:=Range("B1"), Unique:=True iNames = Application.CountA(Columns(2)) ActiveWorkbook.Close savechanges:=False MsgBox "Anzahl Namen: " & iNames Range("B2") = iNames Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

99er Umsatzliste die 98er gegenüberstellen

Problem: Wie kann ich der nach Kunden gegliederten Jahres Umsatzliste die Kundenumsätze eines anderen Jahres gegenüberstellen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Zusammenfassen()
    Dim wks As Worksheet
    Dim var As Variant
    Dim iRow As Integer, iRowT As Integer, iCounter As Integer
    Dim iCol As Integer, iPlus As Integer
    iRowT = 2
    For iCol = 2 To 1 Step -1
        Set wks = Worksheets("200" & iCol)
        iRow = 3
        Do Until IsEmpty(wks.Cells(iRow, 1))
            var = Application.Match(wks.Cells(iRow, 1), Columns(1), 0)
            If IsError(var) Then
                iRowT = iRowT + 1
                var = iRowT
            End If
            Cells(var, 1).Value = wks.Cells(iRow, 1).Value
            Cells(var, 2).Value = wks.Cells(iRow, 2).Value
            Cells(var, 3 + iPlus).Value = wks.Cells(iRow, 3).Value
            iRow = iRow + 1
        Loop
        iPlus = iPlus + 1
    Next iCol
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Listung der mit der Formel KOMBINATIONEN() ermittelten Elemente

Problem:	Wie kann ich die einzelnen Kombinationsmöglichkeiten auflisten, die ich mit der Excel-Funktion KOMBINATIONEN() bei 4 Elementen ermittle?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetKombi() Dim vCount As Variant Dim a As Integer, b As Integer, c As Integer Dim d As Integer, n As Integer, m As Integer Dim s As String, sTxt As String Columns("C:F").ClearContents vCount = Application.InputBox(_ prompt:="Bitte Anzahl der Elemente angeben:", _ Type:=1) If vCount = "" Then Exit Sub vCount = CInt(vCount) s = "A" m = 2 Range("A2").Value = WorksheetFunction.Combin(vCount, 4) For a = 1 To vCount - 3 For b = 2 To vCount - 2 For c = 3 To vCount - 1 For d = 4 To vCount - (c - 3) - (b - 2) - (a - 1) If n >= Cells.Rows.Count Then m = m + 1 n = 1 Else n = n + 1 End If sTxt = s & a sTxt = sTxt & "*" & s sTxt = sTxt & b + (a - 1) sTxt = sTxt & "*" & s sTxt = sTxt & c + ((b - 2) + (a - 1)) sTxt = sTxt & "*" & s sTxt = sTxt & d + ((c - 3) + (b - 2) + (a - 1)) Cells(n, m) = sTxt Next d Next c Next b Next a End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fußballtabelle aktualisieren

Problem:	Wie kann ich Fußballergebnisse in eine Tabelle übernehmen und diese aktualisieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Tabelle() Dim iRow As Integer, iRowH As Integer, iRowA As Integer iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) If Not IsEmpty(Cells(iRow, 10)) And _ Not IsEmpty(Cells(iRow, 11)) Then iRowH = WorksheetFunction.Match(Cells(iRow, 8), Columns(2), 0) iRowA = WorksheetFunction.Match(Cells(iRow, 9), Columns(2), 0) Cells(iRowH, 3).Value = Cells(iRowH, 3).Value + 1 Cells(iRowA, 3).Value = Cells(iRowA, 3).Value + 1 If Cells(iRow, 10).Value > Cells(iRow, 11).Value Then Cells(iRowH, 4).Value = Cells(iRowH, 4).Value + 3 ElseIf Cells(iRow, 10).Value < Cells(iRow, 11).Value Then Cells(iRowA, 4).Value = Cells(iRowH, 4).Value + 3 Else Cells(iRowH, 4).Value = Cells(iRowH, 4).Value + 1 Cells(iRowA, 4).Value = Cells(iRowA, 4).Value + 1 End If Cells(iRowH, 5).Value = Cells(iRowH, 5).Value + Cells(iRow, 10).Value Cells(iRowH, 6).Value = Cells(iRowH, 6).Value + Cells(iRow, 11).Value Cells(iRowH, 7).Value = Cells(iRowH, 7).Value + Cells(iRow, 10).Value - Cells(iRow, 11). Value Cells(iRowA, 5).Value = Cells(iRowA, 5).Value + Cells(iRow, 11).Value Cells(iRowA, 6).Value = Cells(iRowA, 6).Value + Cells(iRow, 10).Value Cells(iRowA, 7).Value = Cells(iRowA, 7).Value + Cells(iRow, 11).Value - Cells(iRow, 10). Value End If iRow = iRow + 1 Loop With Range("A1").CurrentRegion.Columns("B:G") .Sort _ key1:=Range("D2"), order1:=xlDescending, _ key2:=Range("G2"), order2:=xlDescending, _ key3:=Range("E2"), order3:=xlDescending, _ header:=True End With End Sub Sub ResetTab() Range("A1").CurrentRegion.Columns("C:G").Value = 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelte Zelleinträge listen

Problem:	Wie kann ich alle doppelten Zelleinträge eines Tabellenblattes in einen neuen Blatt mit Adressangabe auflisten lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ListDoubles() Dim rng As Range, rngCell As Range Dim fct As WorksheetFunction Dim var As Variant Dim iRow As Integer Set rng = ActiveSheet.UsedRange Set fct = WorksheetFunction Worksheets.Add.Move after:=Worksheets(Worksheets.Count) For Each rngCell In rng.Cells If fct.CountIf(rng, rngCell.Value) > 1 Then var = Application.Match(rngCell.Value, Columns(1), 0) If IsError(var) Then iRow = iRow + 1 Cells(iRow, 1).Value = rngCell.Value Else iRow = var End If Cells(iRow, fct.CountA(Rows(iRow)) + 1).Value = _ rngCell.Address(False, False) End If Next rngCell End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hinzufügen von Begriffen zur AutoKorrektur-Liste

Problem:	Wie kann ich eine Tabelle mit Begriffen zur AutoKorrekturliste hinzufügen bzw. daraus entfernen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AddAutocorrect() Dim iRow As Integer Dim sWas As String, sWomit As String iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) sWas = Cells(iRow, 1) sWomit = Cells(iRow, 2) Application.AutoCorrect.AddReplacement _ what:=sWas, Replacement:=sWomit iRow = iRow + 1 Loop End Sub Sub ResetAutocorrect() Dim iRow As Integer Dim sWas As String On Error GoTo ERRORHANDLER iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) sWas = Cells(iRow, 1) Application.AutoCorrect.DeleteReplacement _ what:=sWas iRow = iRow + 1 Loop Exit Sub ERRORHANDLER: MsgBox prompt:="Listeneintrag existiert nicht!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Markieren von festgelegten Bereichen

Problem:	Über die Markieren-Schaltfläche sollen die im Data-Blatt festgelegten Bereiche in den jeweiligen Spalten markiert oder demarkiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Markieren() Dim iCol As Integer, iRow As Integer Dim rng As Range iCol = ActiveSheet.Buttons(Application.Caller).TopLeftCell.Column iRow = 1 With Worksheets("Data").Columns(iCol) Do Until IsEmpty(.Cells(iRow)) With Range(.Cells(iRow).Value) If .Interior.ColorIndex = 15 Then .Interior.ColorIndex = xlColorIndexNone .ClearContents Else .Interior.ColorIndex = 15 .Value = Worksheets("Grundlagen").Range("B4").Value End If End With iRow = iRow + 1 Loop End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Markieren von Auswahlbereichen

Problem: Über die Markieren-Schaltfläche sollen Einzel- oder Mehrfachauswahlbereiche in den jeweiligen Spalten markiert oder demarkiert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn Schaltflächen zu.

VBA-Code:

```
StandardModule: Modul1

Sub Markieren()
    Dim iCol As Integer
    Dim rng As Range
    iCol = ActiveSheet.Buttons(Application.Caller).TopLeftCell.Column
    For Each rng In Selection.Areas
        If rng.Column = iCol Then
            With rng.Interior
                If .ColorIndex = 15 Then
                    .ColorIndex = xlColorIndexNone
                Else
                    .ColorIndex = 15
                End If
            End With
        End If
    Next rng
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Von einer Zeilenauswahl jede zweite markieren

Problem:	Von den ausgewählten Zeilen soll nur jede zweite markiert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SecondRows() Dim rng As Range Dim lRow As Long Dim bln As Boolean For lRow = Selection.Row To Selection.Row + _ Selection.Rows.Count - 1 bln = Not bln If bln Then If Not rng Is Nothing Then Set rng = Application.Union(rng, Rows(lRow)) Else Set rng = Rows(lRow) End If End If Next lRow rng.Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In Folgeblättern nicht vorkommende Werte markieren

Problem:	Alle Werte einer Datenreihe, die in Folgeblättern nicht vorkommen, sollen fett markiert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Markieren() Dim var As Variant Dim iSheet As Integer, iRow As Integer, iRowL As Integer Dim bln As Boolean iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL If Not IsEmpty(Cells(iRow, 1)) Then For iSheet = ActiveSheet.Index + 1 To Worksheets.Count bln = False var = Application.Match(Cells(iRow, 1).Value, _ Worksheets(iSheet).Columns(1), 0) If Not IsError(var) Then bln = True Exit For End If Next iSheet End If If bln = False Then Cells(iRow, 1).Font.Bold = True End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelte Datensätze farblich markieren und auflisten

Problem: Wie kann ich doppelte Datensätze in Tabelle1 farblich markieren und in Tabelle2 übertragen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Vergleich()  
    Dim iRowA As Integer, iRowB As Integer  
    Dim iCol As Integer, iColor As Integer  
    Dim iRowC As Integer  
    Dim bln As Boolean, blnColor As Boolean  
    iRowA = 2  
    iColor = 2  
    Do Until IsEmpty(Cells(iRowA, 1))  
        iRowB = iRowA + 1  
        Do Until IsEmpty(Cells(iRowB, 1))  
            For iCol = 1 To 3  
                If Cells(iRowA, iCol) <> Cells(iRowB, iCol) Then  
                    bln = True  
                    Exit For  
                End If  
            Next iCol  
            If bln = False Then  
                If blnColor = False Then  
                    iColor = iColor + 1  
                End If  
                If Cells(iRowB, 1).Interior.ColorIndex = _  
                    xlColorIndexNone Then  
                    If Cells(iRowA, 1).Interior.ColorIndex = _  
                        xlColorIndexNone Then  
                        With Worksheets(3)  
                            iRowC = .Cells(Rows.Count, 1).End(xlUp).Row + 1  
                            .Range(.Cells(iRowC, 1), .Cells(iRowC, 3)).Value = _  
                                Range(Cells(iRowB, 1), Cells(iRowB, 3)).Value  
                        End With  
                    End If  
                    Range(Cells(iRowA, 1), Cells(iRowA, 3)). _  
                        Interior.ColorIndex = iColor  
                    Range(Cells(iRowB, 1), Cells(iRowB, 3)). _  
                        Interior.ColorIndex = iColor  
                    blnColor = True  
                End If  
            End If  
            iRowB = iRowB + 1  
            bln = False  
        Loop  
        blnColor = False  
        iRowA = iRowA + 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Verlassen von Spalte B Spalte C markieren

Problem:	Wie kann ich erreichen, daß nach Verlassen einer Zelle in Spalte B die komplette Spalte C markiert wird?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Dim bln As Boolean Private Sub Worksheet_SelectionChange(_ ByVal Target As Excel.Range) If bln Then Columns(3).Select bln = False Else If Target.Column = 2 Then bln = True End If End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle nichtleeren Zeilen markieren

Problem:	Wie kann ich alle nichtleeren Zeilen über VBA auswählen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZeilenMarkieren() Dim rng As Range Dim iRow As Integer, iRowL As Integer iRowL = Cells.Find(what:="*", after:=Range("A1"), _ searchorder:=xlByRows, searchdirection:=xlPrevious).Row For iRow = 1 To iRowL If WorksheetFunction.CountA(Rows(iRow)) > 0 Then If rng Is Nothing Then Set rng = Rows(iRow) Else Set rng = Union(rng, Rows(iRow)) End If End If Next iRow rng.Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Drei Zellen rechts der aktiven Zelle markieren

Problem:	Wie kann ich die 3 Zellen rechts neben der aktiven Zelle markieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub RechtsMarkieren() With ActiveCell Range(.Offset(0, 1), .Offset(0, 3)).Select End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen vergleichen und markieren

Problem:	Wie kann ich zwei Spalten verschiedener Arbeitsblätter vergleichen und Übereinstimmungen in beiden Blättern kennzeichnen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Vergleich() Dim wks As Worksheet Dim var As Variant Dim iRow As Integer Set wks = Worksheets("Tabelle2") For iRow = 1 To WorksheetFunction.CountA(Columns(1)) var = Application.Match(Cells(iRow, 1).Value, wks.Columns(1), 0) If Not IsError(var) Then Cells(iRow, 1).Interior.ColorIndex = 3 wks.Cells(var, 1).Interior.ColorIndex = 3 End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen markieren

Problem:	Wie kann ich zu markierende Zeilen mit Start-, Intervall- und Endwerten über InputBoxes abfragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZeilenMarkieren() Dim rng As Range Dim iStart As Integer, iIntervall As Integer Dim iCount As Integer, iCounter As Integer iStart = InputBox(prompt:="Erste Zeile:", Default:=1) iIntervall = InputBox(prompt:="Intervall:", Default:=2) iCount = InputBox(prompt:="Anzahl:", Default:=20) Set rng = Rows(iStart) For iCounter = iStart To iStart + iCount * iIntervall - 1 Step iIntervall Set rng = Union(rng, Rows(iCounter)) Next iCounter rng.Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bereich mit Maximalwert markieren

Problem: Wie kann ich die Zelle bzw. die Spalte, in dem sich der Maximalwert eines Arbeitsblattes befindet, farbig markieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub CellsSelect()
    Dim rng As Range
    Dim dMax As Double
    dMax = Application.Max(Cells)
    Set rng = Cells.Find( _
        What:=dMax, _
        LookAt:=xlWhole, _
        LookIn:=xlFormulas)
    If Application.Caller = "Zelle markieren" Then
        rng.Interior.ColorIndex = 3
    Else
        Columns(rng.Column).Interior.ColorIndex = 3
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die Zelle mit dem höchsten Wert Auswählen

Problem:	Wie kann ich die Maximalwerte einer Matrix mit gelbem Hintergrund belegen?
Lösung:	Folgende Formel im Menü FORMAT / BEDINGTE FORMATIERUNG: =A1=MAX(A1:C3)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-TextBox-Wert formatieren

Problem:	Nach Verlassen einer TextBox soll die dort eingetragene Zahl im Format "0.00" angezeigt werden.
Lösung:	Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub DialogAufruf() frmFormat.Show End Sub ClassModule: frmFormat Private Sub cmdOK_Click() Unload Me End Sub Private Sub txtNumber_Exit(ByVal Cancel As MSForms.ReturnBoolean) txtNumber.Text = Format(txtNumber.Text, "0.00") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datenmaske mit einem bestimmten Datensatz aufrufen

Problem:	Die Datenmaske zu Tabelle1 soll mit dem Startdatensatz gem. Zelle A1 aufgerufen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DatenMaske() Dim sShortcut As String Select Case Left(Application.Version, 1) Case 5, 7, 8 sShortcut = "%s" Case Else sShortcut = "%k" End Select Application.Goto _ Reference:=Worksheets("Tabelle1").Range("A1"), _ Scroll:=True SendKeys sShortcut & Worksheets("Start").Range("A1").Value SendKeys "%t" ActiveSheet.ShowDataForm End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit InputBoxes Zeilennummern abfragen und Summe bilden

Problem: Wie kann ich durch 2 InputBoxes Start- und Endzeile abfragen und dann in Zelle A1 die Summe der Werte von der Start bis zur Endzeile in Spalte A bilden lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub VariableSumme()
    Dim sVon As String
    Dim sBis As String
    sVon = InputBox("Von:", , 2)
    If sVon = "" Then Exit Sub
    sBis = InputBox( _
        prompt:="Bis:", _
        Default:="5000")
    If sBis = "" Then Exit Sub
    Range("A1").FormulaR1C1 = "=sum(R" & sVon & "C:R" & sBis & "C)"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über InputBoxes Zeile und Spalte auswählen und Ergebnis nennen

Problem:	Wie kann ich über InputBoxes die Zeilen- und Spaltennummer durch Klicken im Tabellenblatt festlegen und das Ergebnis in einer MsgBox anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Auswaehlen() Dim iRow As Integer, iCol As Integer Dim rng As Range Set rng = Application.InputBox _ ("Zeile auswählen:", Type:=8) iRow = rng.Row Set rng = Application.InputBox _ ("Spalte auswählen:", Type:=8) iCol = rng.Column MsgBox Cells(iRow, iCol).Address(False, False) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sicherheitsabfrage, bevor Befehl ausgeführt wird

Problem:	Ich habe ein Makro erstellt, das die Inhalte eines Blattbereichs löschen soll. Wie kann ich vor dem Löschen eine Sicherheitsabfrage aufrufen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Abfrage() If MsgBox(_ prompt:="Löschen?", _ Buttons:=vbQuestion + vbYesNo _) = vbNo Then Exit Sub Range("A1:E16").ClearContents End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bestätigung beim Speichern unterdrücken.

Problem:	Wie kann ich die Rückfrage beim Speichern, ob die bereits vorhandene Datei ersetzt werden soll, verhindern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SaveBook() Dim sName As String On Error GoTo ERRORHANDLER sName = ActiveWorkbook.FullName sName = InputBox(_ prompt:="Bitte Dateiname angeben:", _ Default:=sName) If sName = "" Then Exit Sub Application.DisplayAlerts = False ActiveWorkbook.SaveAs sName Application.DisplayAlerts = True ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe-Schließen-Abfrage abfangen

Problem:	Wie kann ich die Werte der Excelabfrage beim Schließen der Arbeitsmappe über VBA abfangen, um je nach Entscheidung das Makro zu verzweigen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AbfrageBeiSchliessen() Dim sMsg As String sMsg = "Soll die Arbeitsmappe geschlossen und gespeichert werden?" Select Case MsgBox(sMsg, vbInformation + vbYesNoCancel) Case vbYes: ActiveWorkbook.Close savechanges:=True Case vbNo: Call WeiterBeiNein Case vbCancel: Call WeiterBeiAbbruch End Select End Sub Private Sub WeiterBeiNein() MsgBox "Sie haben die Frage nach dem " & Chr(13) _ & "speichern verneint!" End Sub Private Sub WeiterBeiAbbruch() MsgBox "Sie haben den Speichern-Dialog" & Chr(13) _ & "abgebrochen!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fehlermeldung abfangen

Problem:	Wenn Excel innerhalb eines Makros einen Suchbegriff in der Tabelle nicht findet, kommt es zu einer Fehlermeldung. Wie kann ich das verhindern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FindMsg() Dim rng As Range Dim sFind As String sFind = InputBox("Suchbegriff:", , "Hallo!") If sFind = "" Then Exit Sub Set rng = Columns(1).Find(_ what:=sFind, lookat:=xlWhole, LookIn:=xlValues) If rng Is Nothing Then MsgBox "Suchbegriff nicht gefunden!" Else MsgBox rng.Address(False, False) End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Unterdrückung von Sicherheitsabfragen

Problem:	Wie bringe ich Excel dazu, keine Sicherheitsabfragen durchzuführen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ArbeitsblattLoeschen() Const csCpt1 As String = "Blatt anlegen" Const csCpt2 As String = "Blatt löschen" Dim oBtn As Button Set oBtn = ActiveSheet.Buttons(Application.Caller) If oBtn.Caption = csCpt1 Then Worksheets.Add.Move after:=Worksheets(Worksheets.Count) ActiveSheet.Name = "Dummy" oBtn.Caption = csCpt2 oBtn.Parent.Select Else Application.DisplayAlerts = False Worksheets("Dummy").Delete Application.DisplayAlerts = True oBtn.Caption = csCpt1 End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beispiele für den Einsatz von Namen

Problem:	In 7 Tabellenblättern wird gezeigt, wie mit benannten Bereichen und benannten Funktionen gearbeitet werden kann
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim txt As String On Error Resume Next ThisWorkbook.Names("Extern").Delete On Error GoTo 0 txt = "" & ThisWorkbook.Path & "[TestXL4.xls]Tabelle1'!\$A:\$B" txt = "=VLOOKUP(Extern!IV1," & txt & ",2,0)" ThisWorkbook.Names.Add "Extern", txt End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bereichsnamen in Namenn ohne Umlaute umwandeln

Problem:	Bereichsnamen sollen gelesen und in Namen ohne Umlaute umgewandelt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ReadNames() Dim nme As Name For Each nme In ThisWorkbook.Names MsgBox nme.Name Next nme End Sub Sub ChangeNames() Dim nme As Name For Each nme In ThisWorkbook.Names With WorksheetFunction If InStr(nme.Name, "ä") Then nme.Name = .Substitute(nme.Name, "ä", "ae") ElseIf InStr(nme.Name, "ö") Then nme.Name = .Substitute(nme.Name, "ö", "oe") ElseIf InStr(nme.Name, "ue") Then nme.Name = .Substitute(nme.Name, "ü", "ue") End If End With Next nme End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelleinträgen über die Formatierung Spaltennamen hinzufügen

Problem:	Der Bereich A1:C2 soll mit dem Spaltennamen, gefolgt von einem Doppelpunkt formatiert werden. Auf erneuten Schaltflächendruck soll die Formatierung wieder zurückgesetzt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub WithColumnChar() Dim rng As Range For Each rng In Range("A1").CurrentRegion.Cells If rng.NumberFormat = "0" Then rng.NumberFormat = "#####" & _ Chr(rng.Column + 64) & "": "0" Else rng.NumberFormat = "0" End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit VBA-Routine Bereichsnamen festlegen und ändern

Problem: Wie kann ich über eine VBA-Routine eine Serie von vergebenen Bereichsnamen ändern?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub AlteInNeueNamen()  
    Dim nme As Name  
    Dim arr() As String  
    Dim iCounter As Integer, iName As Integer  
    Dim sName As String, swks As String, sAdr As String  
    ReDim Preserve arr(1 To 3, 1 To ThisWorkbook.Names.Count)  
    For Each nme In ThisWorkbook.Names  
        sAdr = nme.RefersTo  
        swks = Left(sAdr, InStr(sAdr, "!") - 1)  
        swks = Right(swks, Len(swks) - 1)  
        sName = Right(sAdr, Len(sAdr) - InStr(sAdr, "!"))  
        iCounter = iCounter + 1  
        arr(1, iCounter) = swks  
        arr(2, iCounter) = sName  
        arr(3, iCounter) = nme.Name  
        nme.Delete  
    Next nme  
    For iName = 1 To iCounter  
        Worksheets(arr(1, iName)).Range(arr(2, iName)).Name = _  
            "New" & arr(3, iName)  
    Next iName  
End Sub  
  
Sub NamenAendern()  
    Dim nme As Name  
    For Each nme In ThisWorkbook.Names  
        nme.Name = "New_" & nme.Name  
    Next nme  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen eines ausgewählten Bereiches auslesen

Problem:	Wie kann ich alle Bereichsnamen ermitteln, die in einem ausgewählten Bereich vorhanden sind?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NamenLesen() Dim nme As Name Dim iCounter As Integer Dim srange As String Range("A1:E16").Select For Each nme In ThisWorkbook.Names srange = nme.RefersTo srange = Right(srange, Len(srange) - InStr(srange, "!")) If Not Intersect(Range(srange), Selection) Is Nothing Then iCounter = iCounter + 1 MsgBox "Name " & iCounter & ":" & nme.Name & " - " & srange End If Next nme End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Von Namen abhängige Nummern vergeben

Problem: Wie kann ich einer Spalte mit Namen eine zweite Spalte mit von den Anfangsbuchstaben abhängige Nummern hinzufügen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub BedingtSchreiben()  
    Dim iRow As Integer  
    iRow = 1  
    Do Until IsEmpty(Cells(iRow, 1))  
        Select Case Left(Cells(iRow, 1), 1)  
            Case "P": Cells(iRow, 2) = 1  
            Case "S": Cells(iRow, 2) = 2  
            Case "c": Cells(iRow, 2) = 3  
        End Select  
        iRow = iRow + 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Namen in der aktiven Arbeitsmappe löschen

Problem:	Wie kann ich alle Namen in der aktiven Arbeitsmappe löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NamenLoeschen() Dim nme As Name For Each nme In ThisWorkbook.Names nme.Delete Next nme End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bereichsnamen ermitteln und als Kommentare anzeigen

Problem: Wie kann ich die in einem Tabellenblatt festgelegten Bereichsnamen ermitteln und durch Kommentare anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Bereiche()  
    Dim cmt As Comment  
    Dim oName As Name  
    Dim sName As String  
    Application.DisplayCommentIndicator = xlCommentIndicatorOnly  
    For Each oName In ActiveWorkbook.Names  
        sName = Right(oName, Len(oName) - InStr(oName, "!"))  
        sName = "Name: " & oName.Name & vbLf & "Bereich: " & sName  
        With Range(oName).Cells(1)  
            If Not .Comment Is Nothing Then  
                .Comment.Delete  
            End If  
            Set cmt = .AddComment(sName)  
        End With  
        cmt.Shape.TextFrame.AutoSize = True  
    Next oName  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Serie von Bereichsnamenteilen einer Arbeitsmappe ändern

Problem:	Wie kann ich Bereichsnamen einer Arbeitsmappe in der Form Weber_A1 oder Müller_C1 in Schorsch_A1 bzw. Wilhelm_C1 ändern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NamenWechseln() Dim oName As Object Dim sName As String For Each oName In ThisWorkbook.Names sName = oName.Name If InStr(sName, "Müller") > 0 Then sName = Application.Substitute(sName, _ "Müller", "Schulze") oName.Name = sName End If Next oName End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Den selben Bereichsnamen in mehreren Tabellenblättern

Problem:	Wie kann ich den selben Bereichsnamen in mehreren Tabellenblättern vergeben?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NamenVergeben() Dim wks As Worksheet Dim sAdr As String sAdr = Selection.Address For Each wks In Worksheets wks.Range(sAdr).Name = wks.Name & "!Druck" Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bezüge auf Namen in anderen Mappen aufheben

Problem:	Wie kann ich mir Namensverknüpfungen auf anderen Arbeitsmappen anzeigen lassen und diese auf Wunsch löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DeleteLinkNames() Dim nmeLink As Object Dim intCounter As Integer, intStart As Integer Dim strMsg As String, strBook As String, strWks As String Dim strRng As String Dim bln As Boolean bln = True For Each nmeLink In ActiveWorkbook.Names If InStr(nmeLink.RefersTo, "[") > 0 Then bln = False intCounter = InStr(nmeLink.RefersTo, "[") + 1 intStart = InStr(nmeLink.RefersTo, "]") - 1 strBook = Mid(nmeLink.RefersTo, intCounter, intStart - intCounter + 1) intCounter = InStr(nmeLink.RefersTo, "!") strWks = Mid(nmeLink.RefersTo, intStart + 2, intCounter - intStart - 2) If Right(strWks, 1) = "'" Then strWks = Left(strWks, Len(strWks) - 1) End If strRng = Mid(nmeLink.RefersTo, intCounter + 1, Len(nmeLink.RefersTo)) strMsg = "Name: " & UCase(nmeLink.Name) & Chr(13) & _ "Arbeitsmappe: " & UCase(strBook) & Chr(13) & _ "Tabellenblatt: " & UCase(strWks) & Chr(13) & _ "Bezug: " & strRng & Chr(13) & Chr(13) & _ "Name löschen?" If MsgBox(strMsg, 292) = vbYes Then nmeLink.Delete End If End If Next nmeLink If bln = True Then MsgBox "Keine Namen gefunden, die auf eine" & Chr(13) & _ "andere Arbeitsmappe verweisen!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verschnittproblem mit Solver lösen

Problem:	Wie kann ich ein Verschnittproblem mit dem Solver lösen? Es stehen Rohrstücke verschiedener Länge zur Verfügung und eine bestimmte Menge soll geliefert werden. Wie kann die günstigste Kombination ermittelt werden?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung

Beispiele zum Thema Sortieren

Problem: In 9 Tabellenblättern werden Beispiele zum Thema Sortieren gezeigt.**Lösung:** Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn Schaltflächen zu.**VBA-Code:** StandardModule: basMain

```
Sub SortBasedOnCustomList()  
    Application.AddCustomList ListArray:=Range("B2:B14")  
    Range("A16:B36").Sort _  
        key1:=Range("B17"), _  
        order1:=xlAscending, _  
        header:=xlYes, _  
        OrderCustom:=Application.CustomListCount + 1  
    Application.DeleteCustomList Application.CustomListCount  
End Sub  
  
Sub CallQuickSortDate()  
    Dim arr(1 To 31) As Date  
    Dim intRow As Integer  
    For intRow = 2 To 32  
        arr(intRow - 1) = Cells(intRow, 1)  
    Next intRow  
    Call QuickSort(arr)  
    For intRow = 2 To 32  
        Cells(intRow, 1).Value = arr(intRow - 1)  
    Next intRow  
End Sub  
  
Sub CallQuickSortFilesA()  
    Dim arr As Variant  
    Dim intCounter As Integer  
    Dim strPath As String  
    strPath = GetDirectory("Bitte Verzeichnis auswählen:")  
    If strPath = "" Then Exit Sub  
    arr = FileArray(strPath, "*.*)")  
    If arr(1) = False Then  
        Beep  
        MsgBox "Keine Dateien gefunden!"  
        Exit Sub  
    End If  
    QuickSort arr  
    Columns("A:B").ClearContents  
    For intCounter = 1 To UBound(arr)  
        Cells(intCounter, 1) = arr(intCounter)  
    Next intCounter  
    Columns(1).AutoFit  
End Sub  
  
Sub CallQuickSortFilesB()  
    Dim arrDate() As Variant  
    Dim arr As Variant  
    Dim intCounter As Integer  
    Dim strPath As String  
    strPath = GetDirectory("Bitte Verzeichnis auswählen:")  
    If strPath = "" Then Exit Sub  
    arr = FileArray(strPath, "*.*)")  
    If arr(1) = False Then  
        Beep  
        MsgBox "Keine Dateien gefunden!"  
        Exit Sub  
    End If  
    Columns("A:B").ClearContents  
    ReDim arrDate(1 To 2, 1 To UBound(arr))  
    For intCounter = 1 To UBound(arr)  
        arrDate(1, intCounter) = arr(intCounter)  
        arrDate(2, intCounter) = FileDateTime(strPath & arr(intCounter))  
    Next intCounter  
    Columns(1).ClearContents  
    For intCounter = 1 To UBound(arr)  
        Cells(intCounter, 1) = arrDate(1, intCounter)  
        Cells(intCounter, 2) = arrDate(2, intCounter)  
    Next intCounter  
    Range("A1").CurrentRegion.Sort key1:=Range("B1"), order1:=xlAscending, header:=xlNo  
    Columns("A:B").AutoFit  
End Sub  
  
Sub CallQuickSortWks()  
    Dim arr() As String  
    Dim intCounter As Integer  
    ReDim arr(1 To Worksheets.Count)  
    For intCounter = 1 To Worksheets.Count  
        arr(intCounter) = Worksheets(intCounter).Name  
    Next intCounter  
    QuickSort arr  
    For intCounter = UBound(arr) To 1 Step -1  
        Worksheets(arr(intCounter)).Move before:=Worksheets(1)  
    Next intCounter  
End Sub  
  
Sub SortSixColumns()  
    Dim intCounter As Integer  
    For intCounter = 2 To 1 Step -1  
        Range("A1").CurrentRegion.Sort _  
            key1:=Cells(1, intCounter * 3 - 2), _  
            order1:=xlAscending, _  
            key2:=Cells(1, intCounter * 3 - 1), _  
            order2:=xlAscending, _  
            key3:=Cells(1, intCounter * 3), _  
            order3:=xlAscending, _  
            header:=xlNo  
    Next intCounter  
End Sub  
  
Sub SpecialSort()  
    With Columns("A")  
        .Replace What:="Ä", Replacement:="Ae", LookAt:=xlPart, SearchOrder _  
            :=xlByRows, MatchCase:=True  
        .Replace What:="Sch", Replacement:="Rzz", LookAt:=xlPart, _  
            SearchOrder:=xlByRows, MatchCase:=True  
        .Sort key1:=Range("A2"), order1:=xlAscending, header:=xlGuess, _  
            OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom  
        .Replace What:="Rzz", Replacement:="Sch", LookAt:=xlPart, _  
            SearchOrder:=xlByRows, MatchCase:=True  
        .Replace What:="Ae", Replacement:="Ä", LookAt:=xlPart, SearchOrder _  
            :=xlByRows, MatchCase:=True  
    End With  
End Sub  
  
Sub CountIfSort()  
    Dim intRow As Integer, intCounter As Integer, intArr As Integer  
    Range("A1").Sort key1:=Range("A1"), order1:=xlAscending, header:=xlNo  
    Range("B1").Formula = "=countif(A:A,A1)"  
    Range("B1:B" & Range("A1").CurrentRegion.Rows.Count).FillDown  
    Range("A1").Sort key1:=Range("B1"), order1:=xlDescending, header:=xlNo  
    Columns("B").ClearContents  
End Sub  
  
Sub SortAll()  
    Dim rngHidden As Range  
    Dim lngLastRow As Long, lngRow As Long  
    Application.ScreenUpdating = False  
    Set rngHidden = Rows(1)  
    lngLastRow = Cells(Rows.Count, 1).End(xlUp).Row  
    For lngRow = 1 To lngLastRow  
        If Rows(lngRow).Hidden = True Then  
            Set rngHidden = Union(rngHidden, Rows(lngRow))  
        End If  
    Next lngRow  
    rngHidden.EntireRow.Hidden = False  
    Range("A1").CurrentRegion.Sort key1:=Range("A2"), _  
        order1:=xlAscending, header:=xlYes  
    rngHidden.EntireRow.Hidden = True  
    Rows(1).Hidden = False  
    Application.ScreenUpdating = True  
End Sub  
  
Sub MultiSort()  
    Dim intRow As Integer  
    For intRow = 1 To 19 Step 6  
        Range(Cells(intRow, 1), Cells(intRow + 4, 8)).Sort _  
            key1:=Cells(intRow + 1, 7), _  
            order1:=xlAscending, header:=xlYes  
    Next intRow  
End Sub  
  
Sub CallSortDialogA()  
    Application.Dialogs(xlDialogSort).Show  
End Sub  
  
Sub CallSortDialogB()  
    Range("A1").Select  
    CommandBars.FindControl(ID:=928).Execute  
End Sub  
  
#####  
# Prozedur zum Sortieren einer Matrix #  
# Autor: John Green #  
#####  
  
Sub QuickSort(ByRef VA_array, Optional V_Low1, Optional V_high1)  
    On Error Resume Next  
    Dim V_Low2, V_high2, V_loop As Integer  
    Dim V_val1, V_val2 As Variant  
    If IsMissing(V_Low1) Then  
        V_Low1 = LBound(VA_array, 1)  
    End If  
    If IsMissing(V_high1) Then  
        V_high1 = UBound(VA_array, 1)  
    End If  
    V_Low2 = V_Low1  
    V_high2 = V_high1  
    V_val1 = VA_array((V_Low1 + V_high1) / 2)  
    While (V_Low2 <= V_high2)  
        While (VA_array(V_Low2) < V_val1 And _  
            V_Low2 < V_high2)  
            V_Low2 = V_Low2 + 1  
        Wend  
        While (VA_array(V_high2) > V_val1 And _  
            V_high2 > V_Low1)  
            V_high2 = V_high2 - 1  
        Wend  
        If (V_Low2 <= V_high2) Then  
            V_val2 = VA_array(V_Low2)  
            VA_array(V_Low2) = VA_array(V_high2)  
            VA_array(V_high2) = V_val2  
            V_Low2 = V_Low2 + 1  
            V_high2 = V_high2 - 1  
        End If  
    Wend  
    If (V_high2 > V_Low1) Then Call _  
        QuickSort(VA_array, V_Low1, V_high2)  
    If (V_Low2 < V_high1) Then Call _  
        QuickSort(VA_array, V_Low2, V_high1)  
End Sub  
  
StandardModule: basFunctions  
  
Public Type BROWSEINFO  
    hOwner As Long  
    pidlRoot As Long  
    pszDisplayName As String  
    lpszTitle As String  
    ulFlags As Long  
    lpfn As Long  
    lParam As Long  
    iImage As Long  
End Type  
  
Declare Function SHGetPathFromIDList Lib "shell32.dll" _  
    Alias "SHGetPathFromIDListA" (ByVal pidl As Long, _  
    ByVal pszPath As String) As Long  
  
Declare Function SHBrowseForFolder Lib "shell32.dll" _  
    Alias "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long  
  
Sub DirAuswahl()  
    Dim msg As String  
    msg = "Wählen Sie bitte einen Ordner aus:"  
    MsgBox GetDirectory(msg)  
End Sub  
  
Function GetDirectory(Optional msg) As String  
    Dim bInfo As BROWSEINFO  
    Dim Path As String  
    Dim r As Long, x As Long, pos As Integer  
    bInfo.pidlRoot = 0&  
    If IsMissing(msg) Then  
        bInfo.lpszTitle = "Wählen Sie bitte einen Ordner aus."  
    Else  
        bInfo.lpszTitle = msg  
    End If  
    bInfo.ulFlags = &H1  
    x = SHBrowseForFolder(bInfo)  
    Path = Space$(512)  
    r = SHGetPathFromIDList(ByVal x, ByVal Path)  
    If r Then  
        pos = InStr(Path, Chr$(0))  
        GetDirectory = Left(Path, pos - 1)  
    Else  
        GetDirectory = ""  
    End If  
End Function  
  
Function FileArray(strPath As String, strPattern As String)  
    Dim arrDateien()  
    Dim intCounter As Integer  
    Dim strDatei As String  
    If Right(strPath, 1) <> "\" Then strPath = strPath & "\"  
    strDatei = Dir(strPath & strPattern)  
    Do While strDatei <> ""  
        intCounter = intCounter + 1  
        ReDim Preserve arrDateien(1 To intCounter)  
        arrDateien(intCounter) = strDatei  
        strDatei = Dir()  
    Loop  
    If intCounter = 0 Then  
        ReDim arrDateien(1)  
        arrDateien(1) = False  
    End If  
    FileArray = arrDateien  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sortieren der Arbeitsblätter nach Datums-Blattnamen

Problem:	Die Tabellenblätter sollen nach dem Datumsnamen sortiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SortWks() Dim wkb As Workbook Dim wks As Worksheet, wksAct As Worksheet Dim iCounter As Integer Application.ScreenUpdating = False ReDim arr(1 To Worksheets.Count) Set wkb = ThisWorkbook Workbooks.Add 1 Set wksAct = ActiveSheet For Each wks In wkb.Worksheets iCounter = iCounter + 1 wksAct.Cells(iCounter, 1).Value = CDate(wks.Name) Range("A1").CurrentRegion.Sort key1:=Range("A1"), order1:=xlAscending Next wks wkb.Worksheets(Format(wksAct.Cells(1, 1).Value, "mmm yy")).Move _ before:=wkb.Worksheets(2) For iCounter = 2 To wkb.Worksheets.Count wkb.Worksheets(Format(wksAct.Cells(iCounter, 1).Value, "mmm yy")).Move _ after:=wkb.Worksheets(iCounter) Next iCounter wksAct.Parent.Close savechanges:=False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Festlegung der Sortierspalten nach dem Feature Gültigkeit

Problem:	Die Sortierspalten sind über die Gültigkeitsfelder G1 und I1 festzulegen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Sortieren() Range("A1").CurrentRegion.Sort _ key1:=Cells(_ 2, _ WorksheetFunction.Match(Range("G1").Value, _ Range("Header"), 0)), order1:=xlAscending, _ key2:=Cells(_ 2, _ WorksheetFunction.Match(Range("I1").Value, _ Range("Header"), 0)), order2:=xlAscending, _ header:=xlYes End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sortieren und nur Uniques einlesen

Problem:	Die Zahlen aus der Tabelle sollen rechts daneben so sortiert werden, dass sie jeweils nur in einer Zeile erscheinen und dort in der Spalte, in der sie in der Quelltable vorhanden sind.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub MySort() Dim rngSource As Range, rngTarget As Range, rng As Range, rngFind As Range Dim iCol As Integer, iRow As Integer Set rngSource = Range("A1").CurrentRegion iCol = rngSource.Columns.Count Set rngTarget = Range(Cells(1, iCol + 2), Cells(65536, iCol * 2 + 1)) For Each rng In rngSource.Cells Set rngFind = rngTarget.Find(rng.Value, LookIn:=xlValues, lookat:=xlWhole) If rngFind Is Nothing Then iRow = iRow + 1 rngTarget(iRow, rng.Column).Value = rng.Value Else rngTarget(rngFind.Row, rng.Column).Value = rng.Value End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle nach Farbindizes sortieren

Problem:	Spalte A soll nach den Farbindizes der Hintergrundfarben sortiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SortColors() Dim iRow As Integer Application.ScreenUpdating = False iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) Cells(iRow, 2).Value = Cells(iRow, 1).Interior.ColorIndex iRow = iRow + 1 Loop Range("A1").CurrentRegion.Sort _ key1:=Range("B1"), _ order1:=xlAscending, _ header:=xlNo Columns("B").ClearContents Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle nach Farben sortieren

Problem:	Die Tabelle soll nach den Farben in Spalte A sortiert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Farben() Dim iCounter As Integer Application.ScreenUpdating = False For iCounter = 2 To 55 Cells(iCounter, 3) = _ Cells(iCounter, 1).Interior.ColorIndex Next iCounter Range("C1") = "Index" Columns("A:C").Sort key1:=Range("C2"), _ order1:=xlAscending, header:=xlYes Columns(3).ClearContents Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adressblöcke umformen und in neue Tabelle ausgeben

Problem:	Die Adressblöcke in Spalte A sollen in Zeilen einer neuen Tabelle aufgelistet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Aufteilen() Dim wks As Worksheet Dim iRowL As Integer, iRow As Integer, iRowT As Integer Application.ScreenUpdating = False Set wks = ActiveSheet Worksheets.Add after:=Worksheets(Worksheets.Count) iRowL = wks.Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL If iRow Mod 7 = 1 Then iRowT = iRowT + 1 If Not IsEmpty(wks.Cells(iRow, 1)) Then Cells(iRowT, iRow Mod 7) = wks.Cells(iRow, 1) End If Next iRow Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Geburtstag sortieren

Problem:	Wie kann ich eine Namenstabelle nach Geburtstagen, also ohne Berücksichtigung des Jahres sortieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Sortieren() Dim iRow As Integer Application.DisplayAlerts = False iRow = Cells(Rows.Count, 1).End(xlUp).Row Range("C1").FormulaR1C1 = "=TEXT(RC[-1],""MM.TT"")" Range("C1:C" & iRow).FillDown Range("A1").CurrentRegion.Sort _ key1:=Range("C1"), order1:=xlAscending, _ key2:=Range("A1"), order2:=xlAscending, _ header:=xlNo Columns("C").Delete Application.DisplayAlerts = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sortierung nach Sortierfolge in zweitem Blatt

Problem:	Spalte A soll nach der in einem zweiten Blatt festgelegten Sortierfolge sortiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SpecialSort() Dim iCount As Integer iCount = Application.CustomListCount Application.AddCustomList _ Worksheets("Tabelle2").Range("A1").CurrentRegion Range("A1").Sort key1:=Range("A1"), _ order1:=xlAscending, ordercustom:=iCount + 2 Application.DeleteCustomList iCount + 1 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neue Anordnung einer Tabelle in zweitem Tabellenblatt

Problem: Die Daten aus der einen Tabelle sollen in einer neuen Tabelle neu angeordnet werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub TabUmwandeln()  
    Dim wks As Worksheet  
    Dim var As Variant  
    Dim iRowL As Integer, iRow As Integer, _  
        iCol As Integer, iRowT As Integer  
    Dim iColT As Integer  
    Dim sLager As String  
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row  
    Set wks = ActiveSheet  
    Worksheets.Add after:=Worksheets(wks.Index)  
    iRowT = 1  
    For iRow = 1 To iRowL  
        If IsEmpty(wks.Cells(iRow, 2)) Then  
            If IsEmpty(Range("B1")) Then  
                iCol = 2  
            Else  
                iCol = Cells(1, Columns.Count).End(xlToLeft).Column + 1  
            End If  
            Cells(1, iCol).Value = wks.Cells(iRow, 1).Value  
            sLager = wks.Cells(iRow, 1).Value  
        Else  
            var = Application.Match(wks.Cells(iRow, 1).Value, Columns(1), 0)  
            If IsError(var) Then  
                iRowT = iRowT + 1  
                Cells(iRowT, 1).Value = wks.Cells(iRow, 1).Value  
                Cells(iRowT, 2).Value = wks.Cells(iRow, 2).Value  
            Else  
                iColT = WorksheetFunction.Match(sLager, Rows(1), 0)  
                Cells(var, iColT).Value = _  
                    Cells(iRowT, iColT).Value + wks.Cells(iRow, 2).Value  
            End If  
        End If  
    Next iRow  
    Cells(1, iCol + 1).Value = "Gesamt"  
    iRow = 2  
    Do Until IsEmpty(Cells(iRow, 1))  
        Cells(iRow, iCol + 1).Value = _  
            WorksheetFunction.Sum(Range(Cells(iRow, 2), Cells(iRow, iCol)))  
        iRow = iRow + 1  
    Loop  
    Rows(1).Font.Bold = True  
    Rows(1).HorizontalAlignment = xlRight  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Sortieren Ä als Ae und Sch vor S

Problem: Spalte A soll so sortiert werden, dass Ä als Ae angesehen wird und Sch vor S kommt.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Sortieren()
    With Columns("A")
        .Replace What:="Ä", _
            Replacement:="Ae", _
            LookAt:=xlPart, _
            SearchOrder:=xlByRows, _
            MatchCase:=True
        .Replace _
            What:="Sch", _
            Replacement:="Rxx", _
            LookAt:=xlPart, _
            SearchOrder:=xlByRows, _
            MatchCase:=True
        .Sort _
            Key1:=Range("A2"), _
            Order1:=xlAscending, _
            Header:=xlGuess, _
            OrderCustom:=1, _
            MatchCase:=False, _
            Orientation:=xlTopToBottom
        .Replace _
            What:="Rxx", _
            Replacement:="Sch", _
            LookAt:=xlPart, _
            SearchOrder:=xlByRows, _
            MatchCase:=True
        .Replace _
            What:="Ae", _
            Replacement:="Ä", _
            LookAt:=xlPart, _
            SearchOrder:=xlByRows, _
            MatchCase:=True
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adressen mit vorgegebenen Anfangsnamen sortieren

Problem:	Die Namen aus Spalte A sollen so sortiert werden, dass bei der Ausgabe die Namen WERNER, DANIELA, ANGELA, JOACHIM, ROSI, HARALD immer am Anfang stehen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub TeilSort() Dim arrFirst(1 To 6) As String Dim arrSecond() As String Dim arrTmp As Variant Dim varTest As Variant Dim iRow As Integer, iCounter As Integer Columns("B").ClearContents arrFirst(1) = "WERNER" arrFirst(2) = "DANIELA" arrFirst(3) = "ANGELA" arrFirst(4) = "JOACHIM" arrFirst(5) = "ROSI" arrFirst(6) = "HARALD" iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) varTest = Application.Match(Cells(iRow, 1).Value, arrFirst, 0) If IsError(varTest) Then iCounter = iCounter + 1 ReDim Preserve arrSecond(1 To iCounter) arrSecond(iCounter) = Cells(iRow, 1).Value End If iRow = iRow + 1 Loop arrTmp = QuickSort(arrSecond) For iRow = 1 To 6 Cells(iRow, 2).Value = arrFirst(iRow) Next iRow For iCounter = iRow To UBound(arrTmp) + iRow - 1 Cells(iCounter, 2).Value = arrTmp(iCounter - iRow + 1) Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte nach Häufigkeit sortieren

Problem:	Die Werte in Spalte A sollen nach Häufigkeit des Vorkommens sortiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Sortieren() Range("A1").Sort _ key1:=Range("A1"), _ order1:=xlAscending, _ header:=xlNo Range("B1").Formula = "=countif(A:A,A1)" Range("B1:B" & Range("A1") _ .CurrentRegion.Rows.Count).FillDown Range("A1").Sort _ key1:=Range("B1"), _ order1:=xlDescending, _ header:=xlNo Columns("B").ClearContents End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sortieren ein- und ausgeblendeter Zeilen

Problem:	Im Normalmodus werden beim Sortieren ausgeblendete Zeilen nicht mitsortiert. Wie kann ich das umgehen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SortAll() Dim rng As Range Dim lRowL As Long, lRow As Long Application.ScreenUpdating = False Set rng = Rows(1) lRowL = Cells(Rows.Count, 1).End(xlUp).Row For lRow = 1 To lRowL If Rows(lRow).Hidden = True Then Set rng = Union(rng, Rows(lRow)) End If Next lRow rng.EntireRow.Hidden = False Range("A1").CurrentRegion.Sort key1:=Range("A2"), _ order1:=xlAscending, header:=xlYes rng.EntireRow.Hidden = True Rows(1).Hidden = False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Durchgehende Sortierung mehrerer nebeneinanderliegender Spalten

Problem:	Wie kann ich mehrere nebeneinanderliegende Spalten sortieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ColumnSort() Dim wks As Worksheet Dim iCol As Integer, iRow As Integer Application.ScreenUpdating = False Set wks = ActiveSheet Workbooks.Add iRow = 1 For iCol = 1 To 9 Step 2 Range(Cells(iRow, 1), Cells(iRow + 39, 1)).Value = _ wks.Range(wks.Cells(3, iCol), wks.Cells(42, iCol)).Value iRow = iRow + 40 Next iCol Range("A1").Sort _ key1:=Range("A1"), order1:=xlAscending, header:=xlNo iRow = 1 For iCol = 1 To 9 Step 2 wks.Range(wks.Cells(3, iCol), wks.Cells(42, iCol)).Value = _ Range(Cells(iRow, 1), Cells(iRow + 40, 1)).Value iRow = iRow + 40 Next iCol ActiveWorkbook.Close savechanges:=False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adressliste mit jeweils Name, Straße, Ort sortieren

Problem: Wie kann ich eine Adressliste, die jeweils in der ersten Zeile den Namen, in der Zweiten die Straße und in der Dritten den Ort beinhaltet, alphabetisch nach den Namen sortieren lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Sortieren()  
    Dim lRow As Long  
    Dim iCol As Integer, iRow As Integer  
    Application.ScreenUpdating = False  
    lRow = 1  
    iCol = 1  
    Do Until IsEmpty(Cells(lRow, 1))  
        iCol = iCol + 1  
        If iCol = 5 Then iCol = 2  
        Cells(CInt((lRow + 1) / 3), iCol) = Cells(lRow, 1)  
        lRow = lRow + 1  
    Loop  
    Columns("B:D").Sort key1:=Range("B1"), order1:=xlAscending, header:=xlNo  
    iRow = 1  
    lRow = 1  
    Do Until IsEmpty(Cells(iRow, 2))  
        For iCol = 2 To 4  
            Cells(lRow, 1) = Cells(iRow, iCol)  
            lRow = lRow + 1  
        Next iCol  
        iRow = iRow + 1  
    Loop  
    Columns("B:D").ClearContents  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrere Tabellenblattbereich unabhängig voneinander sortieren

Problem:	Wie kann ich mehrere, nicht zusammenhängende Tabellenblattbereiche über Schaltflächendruck absteigend sortieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MultiSort() Dim iRow As Integer For iRow = 1 To 19 Step 6 Range(Cells(iRow, 1), Cells(iRow + 4, 8)).Sort _ key1:=Cells(iRow + 1, 7), _ order1:=xlAscending, header:=xlYes Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Viererblock-Datensätze sortieren

Problem:	Wie kann ich zwei Tabellen gleichzeitig sortieren lassen? Zu berücksichtigen ist, dass sich in der zweiten Tabelle auf ViererBlöcke erweiterte Datensätze aus Tabelle1 befinden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Sortieren() Dim rng As Range Dim iRow As Integer, iTarget As Integer iRow = 1 Range("A1").CurrentRegion.Sort key1:=Range("A1"), _ order1:=xlAscending, header:=xlNo With Worksheets("Tabelle2") Do Until IsEmpty(Cells(iRow, 1)) Set rng = .Columns("A").Find(Cells(iRow, 1), _ lookat:=xlWhole, LookIn:=xlValues) iTarget = .Cells(Rows.Count, 5).End(xlUp).Row + 1 If iTarget = 2 Then iTarget = 1 .Range(rng, rng.Offset(3, 3)).Cut .Range _ (.Cells(iTarget, 5), .Cells(iTarget + 3, 8)) iRow = iRow + 1 Loop .Columns("A:D").Delete Application.Goto .Range("A1"), True .Columns.AutoFit End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über VBA den Excel-Sortierdialog aufrufen

Problem:	Wie kann ich über VBA den Excel-Sortierdialog aufrufen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: modMain Sub Sortieren() Range("A1").Select Application.Dialogs(xlDialogSort).Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sortierung mit Formel-Leerstrings am Ende

Problem:	Eine Tabelle soll so sortiert werden, dass Formelergebnisse, die einen Leerstring ergeben, am Ende der Tabelle erscheinen sollen.
Lösung:	Nur anhand einer Beispieldarstellungsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensätze in zweitem Blatt anders anordnen

Problem:	In Tabellenform angeordnete Datensätze sollen in einem zweiten Blatt in Adressform angeordnet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Anordnen() Dim iRow As Integer, iRowT As Integer, iCol As Integer iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) For iCol = 1 To 4 With Worksheets("Data2") iRowT = iRowT + 1 .Cells(iRowT, 1).Value = Cells(iRow, iCol).Value End With Next iCol iRow = iRow + 1 iRowT = iRowT + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenweise sortieren

Problem:	Wie kann ich alle Zeilen eines Arbeitsblattes so sortieren lassen, daß die Werte in jeder Zeile für sich aufsteigend erscheinen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZeilenSortieren() Dim iRow As Integer iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) Rows(iRow).Sort _ Key1:=Cells(iRow, 1), _ Order1:=xlAscending, _ Header:=xlGuess, _ OrderCustom:=1, _ MatchCase:=False, _ Orientation:=xlLeftToRight iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

IP-Nummern nach Gruppen sortieren

Problem:	Wie kann ich IP-Nummern nach Gruppen sortieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Sortieren() Dim iRow As Integer Application.ScreenUpdating = False Range("A1").CurrentRegion.TextToColumns _ Destination:=Range("A1"), _ DataType:=xlDelimited, _ ConsecutiveDelimiter:=False, _ Tab:=False, _ Semicolon:=False, _ Comma:=False, _ Space:=False, _ Other:=True, _ OtherChar:="." Range("A1").CurrentRegion.Sort _ key1:=Range("A1"), order1:=xlAscending, _ key2:=Range("B1"), order2:=xlAscending, _ key3:=Range("C1"), order3:=xlAscending, _ header:=xlNo Range("A1").CurrentRegion.Sort _ key1:=Range("A1"), order1:=xlAscending, _ header:=xlNo iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) Cells(iRow, 1).Value = _ Cells(iRow, 1).Value & _ "." & Cells(iRow, 2).Value & _ "." & Cells(iRow, 3).Value & _ "." & Cells(iRow, 4).Value iRow = iRow + 1 Loop Columns("B:D").ClearContents Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in mehreren Spalten sortieren

Problem:	Wie kann ich die Werte in mehreren Spalten sortieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MultiSort() Dim iCol As Integer, iRow As Integer, iRowT As Integer For iCol = 1 To 2 iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) iRowT = iRowT + 1 Cells(iRowT, 27).Value = Cells(iRow, iCol).Value iRow = iRow + 1 Loop Next iCol Range("AA1").CurrentRegion.Sort _ key1:=Range("AA1"), order1:=xlAscending iRowT = 0 For iCol = 1 To 2 iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) iRowT = iRowT + 1 Cells(iRow, iCol).Value = Cells(iRowT, 27).Value iRow = iRow + 1 Loop Next iCol Columns("AA").ClearContents End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alphanumerisch sortieren

Problem:	Wie kann ich eine Spalte mit alphanumerischen Daten sortieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub NumerischSortieren() Dim rng As Range Application.ScreenUpdating = False For Each rng In Range("A1").CurrentRegion.Cells With rng .Offset(0, 1) = Val(.Value) .Offset(0, 2) = Right(.Value, Len(.Value) - _ Len(.Offset(0, 1).Value)) End With Next rng Range("A1").CurrentRegion.Sort _ Key1:=Range(Cells(1, 2), Cells(1, 2)), _ Order1:=xlAscending, _ Key2:=Range(Cells(1, 3), Cells(1, 3)), _ Order2:=xlAscending, _ Header:=xlGuess, _ OrderCustom:=1, _ MatchCase:=False, _ Orientation:=xlTopToBottom Columns("B:C").Delete Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellen hinzufügen und sortieren

Problem: Wie kann ich einer Arbeitsmappe Arbeitsblätter, deren Namen in Zellen stehen, hinzufügen und die Blätter dann sortieren lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub Blattauswahl()  
    Dim wksList As Worksheet, wks As Worksheet  
    Dim iRow As Integer  
    Set wksList = ActiveSheet  
    iRow = 1  
    On Error Resume Next  
    Do Until IsEmpty(wksList.Cells(iRow, 1))  
        Set wks = Worksheets(wksList.Cells(iRow, 1).Value)  
        If Err > 0 Or wks Is Nothing Then  
            Err.Clear  
            Worksheets.Add  
            ActiveSheet.Name = wksList.Cells(iRow, 1).Value  
        End If  
        iRow = iRow + 1  
    Loop  
    On Error GoTo 0  
    Call SortWorksheets  
    Worksheets("Text").Move before:=Worksheets(1)  
    Worksheets(1).Select  
End Sub  
  
Public Sub SortWorksheets()  
    Dim iCount As Integer, iFirst As Integer, iSecond As Integer  
    iCount = ActiveWorkbook.Worksheets.Count  
    For iFirst = 1 To iCount  
        For iSecond = iFirst To iCount  
            If Worksheets(iSecond).Name < Worksheets(iFirst).Name Then  
                Worksheets(iSecond).Move before:=Worksheets(iFirst)  
            End If  
        Next iSecond  
    Next iFirst  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalte zeilenweise sortieren

Problem:	Wie kann ich Zeilen in ähnlicher Weise wie Spalten sortieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: mHWH Sub Sortieren() Range("A1").CurrentRegion.Sort _ Key1:=Range("A1"), _ Order1:=xlAscending, _ Header:=xlNo, _ OrderCustom:=1, _ MatchCase:=False, _ Orientation:=xlLeftToRight End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Felder über VBA sortieren

Problem:	Wie kann ich über eine VBA-Routine Datenfelder sortieren lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SortArray() Dim arr(1 To 4) Dim iCounter As Integer, iCount As Integer, iTmp As Integer arr(1) = 9: arr(2) = 7: arr(3) = 15: arr(4) = 1 For iCounter = 1 To 4 For iCount = iCounter + 1 To 4 If arr(iCounter) > arr(iCount) Then iTmp = arr(iCounter) arr(iCounter) = arr(iCount) arr(iCount) = iTmp End If Next iCount Next iCounter For iCounter = 1 To 4 MsgBox prompt:=arr(iCounter) Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattdaten als Textdatei mit vorgegebenem Namen speichern

Problem:	Die Werte aus Spalte A von Tabelle2 sollen in einer Textdatei im Verzeichnis "c:\temp" unter dem in Zelle A1 genannten Dateinamen gespeichert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SaveAsText() Dim lRow As Long, lRowL As Long Dim sPath As String sPath = Application.DefaultFilePath & "\" Close Open sPath & Range("A1").Value For Output As #1 With Worksheets("Tabelle2") lRowL = .Cells(Rows.Count, 1).End(xlUp).Row For lRow = 1 To lRowL Print #1, .Cells(lRow, 1).Text Next lRow End With Close MsgBox "Datei gespeichert unter " & _ sPath & Range("A1").Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Laufschrift in Stautsleiste

Problem: In der Statusleiste soll über die in Zelle B1 genannte Anzahl von Durchläufen eine Laufschrift in dem in Zelle B2 genannten Tempo angezeigt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Sub Blinken()
    TextBlinkenLassen (Application.UserName & " sagt guten Morgen!")
End Sub

Sub TextBlinkenLassen(sText As String)
    Dim iChar As Integer, iCounter As Integer
    Dim sTxt As String
    Dim bln As Boolean
    bln = Application.DisplayStatusBar
    Application.DisplayStatusBar = True
    For iCounter = 1 To Range("B1").Value
        For iChar = 1 To Len(sText)
            Sleep Range("B2").Value
            sTxt = sTxt & Mid(sText, iChar, 1)
            Application.StatusBar = sTxt
            DoEvents
        Next iChar
        sTxt = ""
    Next iCounter
    Application.StatusBar = False
    Application.DisplayStatusBar = bln
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Countdown in Statusleiste

Problem:	In der Statusleiste soll ein Countdown von 30 bis 0 Sekunden gezeigt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Countdown() Dim intCounter As Integer Dim bln As Boolean bln = Application.DisplayStatusBar Application.DisplayStatusBar = True For intCounter = 30 To 1 Step -1 Application.StatusBar = "Noch " & _ intCounter & " Sekunden..." Application.Wait Now + TimeSerial(0, 0, 1) Next intCounter Application.StatusBar = False Application.DisplayStatusBar = bln End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sortierspaltenkopf vorgeben und Meldung in Statusleiste für 2 sec.

Problem:	Wie kann ich eine Tabelle mit Spaltenköpfen nach einem in einer Zelle genannten Spaltenkopftext sortieren und danach in der Statusleiste für 2 Sekunden eine Vollzugsmeldung anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Sortieren() Dim rng As Range Dim bln As Boolean bln = Application.DisplayStatusBar Application.DisplayStatusBar = True Set rng = Rows(1).Find(Range("E2").Value, _ lookat:=xlWhole, LookIn:=xlValues) If rng Is Nothing Then Beep MsgBox "Spaltenüberschrift wurde nicht gefunden!" Exit Sub End If Range("A1").Sort key1:=rng.Offset(1, 0), _ order1:=xlAscending, header:=xlYes Application.StatusBar = "Sortiert nach " & _ Range("E2").Value & " - Zelle " & rng.Address(False, False) Application.Wait Now + TimeSerial(0, 0, 2) Application.StatusBar = False Application.DisplayStatusBar = bln End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fortschrittsanzeige in Statusleiste

Problem:	Wie kann ich in der Statusleiste während des Abarbeitens einer VBA-Routine eine Fortschrittsanzeige anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Application.DisplayStatusBar = bln End Sub Private Sub Workbook_Open() bln = Application.DisplayStatusBar Application.DisplayStatusBar = True End Sub StandardModule: Modul1 Public bln As Boolean Sub Fortschritt() Dim iCounter As Integer Application.DisplayStatusBar = True For iCounter = 1 To 10 StatusLED "Bisher abgearbeitet: ", iCounter / 10 Application.Wait Now + TimeSerial(0, 0, 1) Next iCounter Application.StatusBar = False End Sub Private Function StatusLED(sMsg As String, sPct As Single) Dim iPct As Integer, iReps As Integer With WorksheetFunction iPct = .Round(sPct, 2) * 100 iReps = Int(iPct / 10) Application.StatusBar = sMsg & .Rept(Chr(14), iReps) & _ .Rept(" ", 10 - iReps) & " " & iPct & "%" End With End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in Zwischenablage suchen

Problem:	Es wird geprüft, ob der Suchbegriff aus Zelle B1 in dem in der Zwischenablage gespeicherten Text vorkommt. Wenn ja, wird der Wert der Zwischenablage in Zelle B2 eingetragen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ReadClipboard() Dim wks As Worksheet Dim rng As Range Dim vaFormats As Variant, vFormat As Variant Dim sSearch As String Set wks = ActiveSheet sSearch = Range("B1").Value Application.ScreenUpdating = False Range("B2").Select vaFormats = Application.ClipboardFormats Workbooks.Add 1 For Each vFormat In vaFormats If vFormat = xlClipboardFormatText Then ActiveSheet.Paste Set rng = Cells.Find(what:=sSearch, LookIn:=xlValues, lookat:=xlPart) End If Next vFormat If Not rng Is Nothing Then wks.Paste Else Beep MsgBox "Der Suchbegriff wurde in der Zwischenablage nicht gefunden!" End If ActiveWorkbook.Close savechanges:=False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert bei Übereinstimmung von 3 Spalten eintragen

Problem:	Die Werte aus C2:C4 sollen im Datenblatt spaltenweise gesucht und der Wert aus Zelle C5 neben den Fundwerten eingetragen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Eintragen() Dim iRow As Integer Dim bln As Boolean With Worksheets("Daten") iRow = 2 Do Until IsEmpty(.Cells(iRow, 1)) If .Cells(iRow, 1).Value = Range("MaterialNummer").Value And _ .Cells(iRow, 2).Value = Range("NummerDispo").Value And _ .Cells(iRow, 3).Value = Range("NameDispo").Value Then .Cells(iRow, 4).Value = Range("NeuerWert").Value bln = True End If iRow = iRow + 1 Loop End With If bln = False Then Beep MsgBox "Die Eingabezeile wurde nicht gefunden!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchdialog über VBA aufrufen

Problem:	Der Suchdialog soll über VBA-Befehl aufgerufen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SuchDialog() Dim oBtn As CommandBarControl Set oBtn = Application.CommandBars.FindControl(ID:=1849) oBtn.Execute End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Artikel bei Nichtvorhandensein eintragen, sonst addieren

Problem:	Es soll überprüft werden, ob die jeweiligen Artikel aus Spalte A in Tabelle 2 vorhanden sind. Wenn ja, ist die Stückzahl zu addieren, wenn nein, ist der Artikel mit der Stückzahl einzutragen. Die Reihenfolge ist ohne Relevanz.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Addieren() Dim wks As Worksheet Dim vRow As Variant Dim iRow As Integer Set wks = Worksheets("Tabelle2") iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) vRow = Application.Match(Cells(iRow, 1).Value, wks.Columns(1), 0) If IsError(vRow) Then vRow = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1 wks.Cells(vRow, 1).Value = Cells(iRow, 1).Value End If wks.Cells(vRow, 2).Value = wks.Cells(vRow, 2).Value + Cells(iRow, 2).Value iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahl suchen und an den Fundstellen Zeile einfügen

Problem: Eine über InputBox abzufragende Zahl soll in allen Blättern gesucht werden. Unterhalb der Fundstellen ist eine Zeile einzufügen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub FindenEinfuegen()
    Dim wks As Worksheet
    Dim rng As Range
    Dim dInput As Double
    Dim sRng As String
    dInput = Application.InputBox( _
        prompt:="Geben Sie eine Zahl ein:", _
        Title:="Zahleneingabe", _
        Default:="7", _
        Type:=1)
    For Each wks In Worksheets
        If wks.Index > 1 Then
            Set rng = wks.Cells.Find( _
                what:=dInput, _
                lookat:=xlWhole, _
                LookIn:=xlValues)
            If Not rng Is Nothing Then
                sRng = rng.Address
                rng.Offset(1, 0).EntireRow.Insert
                Do
                    Set rng = wks.Cells.FindNext(rng)
                    If rng.Address <> sRng Then
                        rng.Offset(1, 0).EntireRow.Insert
                    Else
                        Exit Do
                    End If
                Loop
            End If
        End If
    Next wks
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Begriff suchen und Fundzeilen in anderes Blatt kopieren

Problem: Ein über InputBox abzufragender Name ist zu suchen und die Fundzeilen sind in ein zweites Blatt zu übertragen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SearchNames()
    Dim rng As Range, rngSource As Range, rngStart As Range
    Dim varInput As Variant
    Dim iRow As Integer
    varInput = Application.InputBox( _
        prompt:="Geben Sie bitte den Namen ein:", _
        Title:="Namen-Zeilen kopieren", _
        Default:="Name 4", _
        Left:=263, _
        Top:=169, _
        Type:=2)
    If varInput = False Then Exit Sub
    Set rng = ActiveSheet.Columns("A:F").Find( _
        what:=varInput, lookat:=xlWhole, LookIn:=xlValues)
    If rng Is Nothing Then
        Beep
        MsgBox "Suchbegriff nicht gefunden!"
        Exit Sub
    End If
    Set rngStart = rng
    Set rngSource = rng.EntireRow
    Do
        Set rng = Cells.FindNext(After:=rng)
        If rng.Address = rngStart.Address Then Exit Do
        Set rngSource = Application.Union(rngSource, rng.EntireRow)
    Loop
    With Worksheets("Target")
        iRow = .Cells(Rows.Count, 1).End(xlUp).Row
        If iRow = 1 Then iRow = 2 Else iRow = iRow + 3
        rngSource.Copy .Cells(iRow, 1)
        .Columns.AutoFit
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wenn Wert nicht gefunden wird, eintragen

Problem:	Jeder Wert in Spalte A soll mit den Werten in Spalte B verglichen werden. Wird der jeweilige Wert nicht gefunden, ist er in Spalte C einzutragen.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub VergleichenKopieren() Dim v As Variant Dim lRowL As Long, lRow As Long lRowL = Cells(Rows.Count, 1).End(xlUp).Row For lRow = 1 To lRowL v = Application.Match(Cells(lRow, 1).Value, Columns(2), 0) If IsError(v) Then Cells(lRow, 3).Value = Cells(lRow, 1).Value End If Next lRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Artikelnummern nach Kriterien suchen und Datensätze kopieren

Problem:	Die Artikelnummern aus dem Blatt "Kriterien" sollen im Blatt "Daten" in Spalte A gesucht werden. Beim Auffinden sollen die Daten der Fundzeile in das Blatt "Gefunden" kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SuchenFinden() Dim wksCriteria As Worksheet, WksData As Worksheet Dim wksTrue As Worksheet, wksFalse As Worksheet Dim var As Variant Dim iRow As Integer, iRowL As Integer Set wksCriteria = Worksheets("Kriterien") Set WksData = Worksheets("Daten") Set wksTrue = Worksheets("Gefunden") Set wksFalse = Worksheets("NichtGefunden") iRow = 2 Do Until IsEmpty(wksCriteria.Cells(iRow, 1)) var = Application.Match(_ wksCriteria.Cells(iRow, 1), WksData.Columns(1), 0) If IsError(var) Then iRowL = wksFalse.Cells(Rows.Count, 1).End(xlUp).Row + 1 wksFalse.Rows(iRowL).Value = wksCriteria.Rows(iRow).Value Else iRowL = wksTrue.Cells(Rows.Count, 1).End(xlUp).Row + 1 wksTrue.Rows(iRowL).Value = WksData.Rows(iRow).Value End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile 1 nach Wert der aktiven Zelle durchsuchen

Problem:	In Zeile 1 soll der Bereich rechts von der aktiven Zelle nach dem Wert in der aktiven Zelle durchsucht und eine Ergebnismeldung angezeigt werden
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Durchsuchen() Dim rng As Range Set rng = Range(ActiveCell.Offset(0, 1), _ ActiveCell.Offset(0, 256 - ActiveCell.Column)).Find(_ what:=ActiveCell.Value, _ lookat:=xlWhole, LookIn:=xlValues) If Not rng Is Nothing Then MsgBox "Fundstelle: " & rng.Address Else MsgBox "Wert wurde nicht gefunden!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte suchen, eintragen und löschen

Problem:	Der Wert in der aktiven Zelle in Spalte A wird in den übrigen Zellen der Spalte gesucht. Wird er gefunden, wird der Wert aus Spalte C der Fundstelle dem Wert der Zelle in der aktiven Zeile in Spalte C hinzugefügt. Die Fundzeile wird gelöscht.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub PruefenLoeschen() Dim rng As Range Dim lRow As Long, lRoll As Long Application.ScreenUpdating = False Set rng = ActiveCell lRoll = Cells(Rows.Count, 1).End(xlUp).Row For lRow = lRoll To rng.Row + 1 Step -1 If Cells(lRow, 1).Value = rng.Value Then rng.Offset(0, 2).Value = _ rng.Offset(0, 2).Value + Cells(lRow, 3).Value Rows(lRow).Delete End If Next lRow Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriff abfragen und Fundzeile markieren

Problem:	Wie kann ich über InputBox einen Suchbegriff abfragen lassen und dann die Fundzeile selektieren? Wird der Suchbegriff nicht gefunden, soll eine Meldung angezeigt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Suchen() Dim rng As Range Dim sFind As String sFind = InputBox(_ prompt:="Suchbegriff:", _ Default:="def456") If sFind = "" Then Exit Sub Set rng = Columns(1).Find(_ what:=sFind, lookat:=xlWhole, LookIn:=xlValues) If rng Is Nothing Then Beep MsgBox "Suchbegriff wurde nicht gefunden!" Exit Sub End If Rows(rng.Row).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriff in Tabelle suchen und Fundzeilen auswählen

Problem:	Wie kann ich einen durch InputBox abzufragenden Suchbegriff in einer Tabelle suchen und alle Fundzeilen auswählen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MultiSelect() Dim wks As Worksheet Dim rngFind As Range, rngRows As Range Dim sFind As String, sSearch As String sSearch = InputBox("Suchbegriff:", , "test") Set rngFind = Cells.Find(sSearch) If rngRows Is Nothing Then Set rngRows = rngFind End If If Not rngFind Is Nothing Then sFind = rngFind.Address Do Set rngRows = Application.Union(rngRows, rngFind.EntireRow) Set rngFind = Cells.FindNext(After:=rngFind) If rngFind.Address = sFind Then Exit Do Loop End If rngRows.Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in Tabelle 2 suchen und Fundstellen übertragen

Problem: Die Wert aus Spalte A sollen in Tabelle2 an beliebiger Stelle gesucht und die Werte aus den Spalten B:C der Fundzeile in dieses Blatt übernommen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Uebetragen()
    Dim rng As Range
    Dim iRowL As Integer, iRow As Integer
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row
    For iRow = 1 To iRowL
        If Not IsEmpty(Cells(iRow, 1)) Then
            With Worksheets("Tabelle2")
                Set rng = .Cells.Find(Cells(iRow, 1), _
                    lookat:=xlWhole, LookIn:=xlValues)
                If Not rng Is Nothing Then
                    Cells(iRow, 2) = .Cells(rng.Row, 2)
                    Cells(iRow, 3) = .Cells(rng.Row, 3)
                End If
            End With
        End If
    Next iRow
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchen in 2. Tabelle und Ersetzen der Quelldaten

Problem:	Die Werte aus Spalte A sollen in Tabelle2 in Spalte A gesucht werden. Wird der jeweilige Wert gefunden, ist die Zeile im aktiven Blatt durch die Fundzeile zu ersetzen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SuchenErsetzen() Dim wksSource As Worksheet, wksTarget As Worksheet Dim var As Variant Dim lRow As Long Application.ScreenUpdating = False Set wksSource = Worksheets("Tabelle2") Set wksTarget = Worksheets("Tabelle1") lRow = 1 Do Until IsEmpty(wksTarget.Cells(lRow, 1)) var = Application.Match(wksTarget.Cells(lRow, 1), _ wksSource.Columns(1), 0) If Not IsError(var) Then wksTarget.Rows(lRow).Value = wksSource.Rows(var).Value End If lRow = lRow + 1 Loop Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Höchstes Datum einer Reihe ermitteln und eintragen

Problem: Aus Spalte A soll das höchste Datum ausgelesen und alle Datumseintragungen in Spalte B geschrieben werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub LastDate()
    Dim iRow As Integer, iRowL As Integer
    Dim iTarget As Integer
    Dim dat As Date
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row
    For iRow = 1 To iRowL
        If IsDate(Cells(iRow, 1)) Then
            If Cells(iRow, 1).Value > dat Then
                dat = Cells(iRow, 1).Value
            End If
            iTarget = iTarget + 1
            Cells(iTarget, 2) = Cells(iRow, 1)
        End If
    Next iRow
    MsgBox Format(dat, "dd.mm.yy")
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Artikelnummern suchen und jeweils letzte Fundstelle listen

Problem: Wie kann ich mehrere Tabellenblätter nach einer Liste von Artikelnummern durchsuchen lassen und die jeweils letzte Fundstelle neben den Artikelnummern eintragen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub LetztePosition()  
    Dim iRowT As Integer, iRowS As Integer  
    Dim iRowL As Integer, iCounter As Integer  
    iRowT = 1  
    Do Until IsEmpty(Cells(iRowT, 1))  
        For iCounter = Worksheets.Count - 1 To 1 Step -1  
            With Worksheets(iCounter)  
                iRowL = .Cells(Rows.Count, 1).End(xlUp).Row  
                For iRowS = iRowL To 1 Step -1  
                    If .Cells(iRowS, 1) = Cells(iRowT, 1) Then  
                        If .Cells(iRowS, 2) <> "liegt nicht vor" Then  
                            Cells(iRowT, 2) = .Name & "!" & _  
                                .Cells(iRowS, 1).Address(False, False)  
                            Exit For  
                        End If  
                    End If  
                Next iRowS  
            End With  
        Next iCounter  
        iRowT = iRowT + 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriff-Fundstelle mit anderer Zelle multiplizieren

Problem: Wie kann ich die Zelle neben einer Suchbegriff-Fundstelle mit einem Wert in einem zweiten Arbeitsblatt multiplizieren und das Ergebnis in einer MessageBox anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub SeekAndMulti()  
    Dim rng As Range  
    Dim sSearch As String  
    sSearch = InputBox( _  
        prompt:="Suchbegriff:", _  
        Default:="Zeile 5 - Spalte 1")  
    If sSearch = "" Then Exit Sub  
    Set rng = Columns("A").Find(sSearch, _  
        lookat:=xlWhole, LookIn:=xlValues)  
    If rng Is Nothing Then  
        Beep  
        MsgBox "Suchbegriff wurde nicht gefunden!"  
    Else  
        With Worksheets("Tabelle2")  
            MsgBox "Die Multiplikation ergibt:" & vbLf & _  
                rng.Offset(0, 1) & " * " & _  
                .Cells(rng.Row, 2) & " = " & rng.Offset(0, 1) * _  
                .Cells(rng.Row, 2)  
        End With  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriff in vorhergehenden Tabellen suchen

Problem:	Wie kann ich die vor dem aktiven Arbeitsblatt liegenden Tabellen nach einem Suchbegriff durchsuchen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Suchen() Dim rng As Range Dim iCounter As Integer Dim sSearch As String, sAddress As String, sFirst As String Dim bln As Boolean sSearch = InputBox("Bitte Suchbegriff eingeben:", , "Martin") If sSearch = "" Then Exit Sub For iCounter = ActiveSheet.Index - 1 To 1 Step -1 Set rng = Worksheets(iCounter).Columns("A").Find _ (sSearch, lookat:=xlPart, LookIn:=xlValues) If Not rng Is Nothing Then sFirst = rng.Address(False, False) MsgBox Worksheets(iCounter).Name & "!" & sFirst bln = True Do Until sAddress = sFirst Set rng = Worksheets(iCounter).Columns("A").FindNext(after:=rng) sAddress = rng.Address(False, False) If sFirst <> sAddress Then MsgBox Worksheets(iCounter).Name & "!" & sAddress End If Loop sAddress = "" sFirst = "" End If Next iCounter If bln = False Then MsgBox "Der Suchbegriff wurde nicht gefunden!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Begriffe in zweiter Tabelle suchen und Werte anfügen

Problem:	Wie kann ich die Zeichenfolgen aus Spalte A der aktiven Tabelle in Spalte A einer zweiten Tabelle suchen und rechts neben den Fundstellen die Werte aus dem aktiven Blatt eintragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub VergleichenKopieren() Dim rng As Range Dim iRow As Integer iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) Set rng = Worksheets("Tabelle2").Columns(1).Find(_ what:=Cells(iRow, 1), lookat:=xlWhole, LookIn:=xlValues) If Not rng Is Nothing Then Range(rng.Offset(0, 1), rng.Offset(0, 3)).Value = _ Range(Cells(iRow, 2), Cells(iRow, 4)).Value End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datum suchen und Adresse der Fundzelle ausgeben

Problem:	Wie kann ich in Spalte A ein Datum suchen und die Adresse der Fundzelle in einer MsgBox ausgeben lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SuchDatum() Dim rng As Range Dim dat As Date Dim var As Variant dat = DateSerial(2000, 5, 5) var = Application.Match(CDbl(dat), Columns(1), 0) If IsError(var) Then Beep MsgBox prompt:="Datum wurde nicht gefunden!" Else MsgBox prompt:="Fundzeile: " & var End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriff über InputBox abfragen und Fundzeile löschen

Problem: Wie kann ich einen Suchbegriff über eine InputBox abfragen, diesen in der aktiven Tabelle suchen und die Fundzeile löschen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Loeschen()  
    Dim rngFind As Range  
    Dim var As Variant  
    Dim swkn As String  
    swkn = Application.InputBox( _  
        prompt:="Wie lautet die WKN des zu löschenden Papiers?", _  
        Title:="Löschung von Papieren", _  
        Default:="123461")  
    If swkn = "" Then Exit Sub  
    var = Application.Match(CLng(swkn), Columns(2), 0)  
    If IsError(var) Then  
        Beep  
        MsgBox "WKN-Nummer wurde nicht gefunden!"  
    Else  
        If MsgBox( _  
            prompt:="Soll dieses Papier gelöscht werden?", _  
            Buttons:=vbQuestion + vbYesNo _  
            ) = vbNo Then Exit Sub  
        Rows(var).Delete  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Begriffe in ausgeblendeter Spalte suchen und Fundzeile markieren

Problem: Wie kann ich in der ausgeblendeten Spalte A durch eine InputBox ermittelte Begriffe suchen und die Fundzeile markieren lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub SuchenUndMarkieren()
    Dim rng As Range
    Dim sBegriff As String
    sBegriff = InputBox( _
        prompt:="Suchbegriff:", _
        Default:="drei")
    If sBegriff = "" Then Exit Sub
    Set rng = Columns(1).Find( _
        what:=sBegriff, lookat:=xlWhole, LookIn:=xlFormulas)
    If rng Is Nothing Then
        Beep
        MsgBox "Suchbegriff wurde nicht gefunden!"
        Exit Sub
    End If
    Rows(rng.Row).Select
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in Tabellenblatt suchen und alle Fundwerte in zweite Tabelle

Problem:	Wie kann ich ein Tabellenblatt nach einem bestimmten Begriff durchsuchen und die gefundenen Werte in eine zweite Tabelle eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FindenUndKopieren() Dim iRowS As Integer, iRowT As Integer Dim sWord As String sWord = InputBox(_ prompt:="Suchbegriff:", _ Default:="Zeile 3 - Spalte 1") If sWord = "" Then Exit Sub iRowS = 1 iRowT = 1 With Worksheets("Tabelle2") Do Until IsEmpty(Cells(iRowS, 1)) If Cells(iRowS, 1) = sWord Then Rows(iRowS).Copy .Rows(iRowT) iRowT = iRowT + 1 End If iRowS = iRowS + 1 Loop .Columns.AutoFit .Select End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateisuche unter Einbeziehung der untergeordneten Ordner

Problem: Wie kann ich eine Datei in einem Verzeichnis unter Einbeziehung der Unterverzeichnisse suchen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Suchen()  
    Dim fs As FileSearch  
    Dim sDir As String, sFile As String  
    sDir = InputBox( _  
        prompt:="Verzeichnis:", _  
        Default:=Application.DefaultFilePath)  
    If sDir = "" Then Exit Sub  
    sFile = InputBox( _  
        prompt:="Dateiname:", _  
        Default:="test1.xls")  
    If sFile = "" Then Exit Sub  
    Set fs = Application.FileSearch  
    With fs  
        .LookIn = sDir  
        .SearchSubFolders = True  
        .FileName = sFile  
        If .Execute(SortBy:=msoSortByFileName, _  
            SortOrder:=msoSortOrderAscending) > 0 Then  
            MsgBox "Gefunden: " & .FoundFiles(1)  
        Else  
            MsgBox "Datei wurde nicht gefunden!"  
        End If  
    End With  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Formeln und Zahlen löschen

Problem:	Wie kann ich in einem Arbeitsblatt alle enthaltenen Formeln und Zahlen löschen, die Texte aber stehenlassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub WerteLoeschen() Dim rng As Range For Each rng In ActiveSheet.UsedRange.Cells If rng.HasFormula Or IsNumeric(rng) Then rng.ClearContents End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Letzte Zelle mit Inhalt suchen

Problem: Mit der Tastenkombination ENDE+Pos1 wird nicht immer die real letzte Zelle markiert. Wie kann ich das Problem beheben?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub LetzteZelle()  
    RealLastCell(Worksheets("Tabelle1")).Select  
End Sub  
  
'Funktion von John Walkenbach  
Function RealLastCell(TheSheet As Worksheet) As Range  
    Dim ExcelLastCell As Range  
    Dim Row%, Col%, LastRowWithData%, LastColWithData%  
    Application.ScreenUpdating = False  
    Set ExcelLastCell = TheSheet.Cells.SpecialCells(xlLastCell)  
    LastRowWithData = ExcelLastCell.Row  
    Row = ExcelLastCell.Row  
    Do While Application.CountA(TheSheet.Rows(Row)) = 0 And Row <> 1  
        Row = Row - 1  
    Loop  
    LastRowWithData = Row  
    LastColWithData = ExcelLastCell.Column  
    Col = ExcelLastCell.Column  
    Do While Application.CountA(TheSheet.Columns(Col)) = 0 And Col <> 1  
        Col = Col - 1  
    Loop  
    LastColWithData = Col  
    Set RealLastCell = TheSheet.Cells(Row, Col)  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Zellen mit der #BEZUG!-Fehlermeldung auswählen

Problem:	Wie kann ich in einem Arbeitsblatt alle die Zellen auswählen, die einen #BEZUG!-Fehler enthalten?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BezugFehlerSuchen() Dim rng As Range, rngErrors As Range, rngSelect As Range Set rngErrors = Cells.SpecialCells(xlFormulas, 16) For Each rng In rngErrors.Cells If rng.Text = "#BEZUG!" Then If rngSelect Is Nothing Then Set rngSelect = rng Else Set rngSelect = Union(rng, rngSelect) End If End If Next rng rngSelect.Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Füllen aller leeren Zellen mit einem Platzhalter

Problem:	Wie kann ich alle leeren Zellen eines ausgewählten Bereiches mit einem Platzhalter füllen und die Hintergrundfarbe festlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Fuellen() With ActiveSheet.Cells.SpecialCells(xlCellTypeBlanks) .Value = "Hallo" .Interior.ColorIndex = 6 End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fortlaufende Suche unter Vermeidung eines Laufzeitfehlers

Problem: Wie kann ich einen Tabellenbereich nach einem bestimmten Begriff durchsuchen? Wird der Begriff nicht gefunden, soll ein Laufzeitfehler vermieden, wird er gefunden, soll die Suche fortgesetzt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Auswahl()  
    Dim rng As Range  
    Dim sSearch As String, sAddress As String  
    Range("A1").Select  
    sSearch = InputBox(prompt:="Bitte Suchbegriff eingeben:")  
    If sSearch = "" Then Exit Sub  
    Set rng = ActiveSheet.Columns("A:F").Find( _  
        what:=sSearch, lookin:=xlValues, lookat:=xlWhole, _  
        searchorder:=xlByRows)  
    If rng Is Nothing Then  
        Beep  
        MsgBox prompt:="Suchbegriff nicht gefunden!"  
        Exit Sub  
    End If  
    sAddress = rng.Address  
    MsgBox rng.Address(False, False)  
    While ActiveCell.Address <> sAddress  
        Set rng = Cells.FindNext(After:=rng)  
        If rng.Address = sAddress Then Exit Sub  
        MsgBox rng.Address(False, False)  
    Wend  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen löschen, wenn in bestimmten Spalten keine Werte stehen

Problem: Wie kann ich alle Zeilen eines Arbeitsblattes löschen, in denen in den Spalten B, D und J keine Werte stehen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub ZeilenLoeschen()  
    Dim iRow As Integer, iRowL As Integer  
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row  
    For iRow = iRowL To 1 Step -1  
        If IsEmpty(Cells(iRow, 2)) And _  
            IsEmpty(Cells(iRow, 4)) And _  
            IsEmpty(Cells(iRow, 10)) Then  
            Rows(iRow).Delete  
        End If  
    Next iRow  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahl +/- 1 suchen

Problem:	Wie kann ich eine Zahl +/- 1 in einer Spalte suchen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZahlSuchen() Dim rng As Range Dim var As Variant var = Application.InputBox(_ prompt:="Bitte Zahl eingeben:", _ Default:=7, _ Type:=1) If var = "" Then Exit Sub var = CLng(var) For Each rng In Selection.Cells If rng.Value >= var - 1 And rng.Value <= var + 1 Then rng.Select Exit Sub End If Next rng Beep MsgBox prompt:="Zahl nicht gefunden!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchen und weitersuchen

Problem: Wie erreiche ich, dass ein Suchbegriff im aktiven Blatt gesucht und jede Fundstelle über MsgBox gemeldet wird?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Auswahl()
    Dim rng As Range
    Dim sBegriff As String, sAddress As String
    sBegriff = InputBox( _
        prompt:="Bitte Suchbegriff eingeben:", _
        Default:="Hallo")
    If sBegriff = "" Then Exit Sub
    Set rng = Cells.Find( _
        what:=sBegriff, _
        lookat:=xlWhole, _
        LookIn:=xlValues, _
        MatchCase:=False, _
        after:=ActiveCell)
    If rng Is Nothing Then
        Beep
        MsgBox "Suchbegriff nicht gefunden!", , _
            Application.UserName
        Exit Sub
    End If
    sAddress = rng.Address
    rng.Select
    MsgBox rng.Address(False, False)
    rng.Offset(1).Select
    Do
        Cells.FindNext(after:=ActiveCell).Activate
        If ActiveCell.Address = sAddress Then Exit Sub
        MsgBox ActiveCell.Address(False, False)
    Loop
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriff über mehrere Tabellenblätter suchen.

Problem:	Wie kann ich eine ganze Arbeitsmappe nach einem über eine Inputbox einzugebenden Suchbegriff durchsuchen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MultiSeek() Dim wks As Worksheet Dim rng As Range Dim sAddress As String, sFind As String sFind = InputBox("Bitte Suchbegriff eingeben:") For Each wks In Worksheets Set rng = wks.Cells.Find(_ what:=sFind, _ lookat:=xlWhole, _ LookIn:=xlFormulas) If Not rng Is Nothing Then sAddress = rng.Address Do Application.Goto rng, True If MsgBox(_ prompt:="Weiter", _ Buttons:=vbYesNo + vbQuestion _) = vbNo Then Exit Sub Set rng = Cells.FindNext(after:=ActiveCell) If rng.Address = sAddress Then Exit Do Loop End If Next wks MsgBox prompt:="Keine neue Fundstelle!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suche über mehrere Tabellen

Problem:	Wie kann ich das Suchen auf mehrere oder alle Tabellenblätter erweitern?
Lösung:	Alle Tabs der zu durchsuchenden Blätter markieren und den Suchvorgang starten.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelte Einträge finden und löschen

Problem:	Ich möchte in einem Tabellenblatt in einer Spalte nach doppelten Einträgen suchen. Ist ein Wert doppelt vorhanden, soll die Zeile gelöscht werden. Spalten der Zeilen mit den
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DblFind() Dim iRow As Integer, iRowL As Integer iRowL = Cells(Cells.Rows.Count, 1).End(xlUp).Row For iRow = iRowL To 1 Step -1 If WorksheetFunction.CountIf(Columns(1), Cells(iRow, 1)) > 1 Then Rows(iRow).Delete End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalt suchen und Zelle auswählen

Problem: Wie kann ich einen vorzugebenden Begriff in einer Anzahl von Tabellenblättern suchen und die Fundstellen melden lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Lagernummer()
    Dim rng As Range
    Dim vNumber As Variant
    Dim iCounter As Integer
    Dim sFirst As String
    Dim bln As Boolean
    vNumber = InputBox( _
        prompt:="Bitte Lagernummer eingeben:", _
        Default:="Nummer 12")
    If vNumber = "" Then Exit Sub
    For iCounter = 2 To Worksheets.Count
        Set rng = Worksheets(iCounter).Cells.Find(vNumber)
        If rng Is Nothing = False Then
            sFirst = rng.Address
            Do
                Set rng = Worksheets(iCounter).Cells.FindNext(rng)
                MsgBox "Gefunden in Blatt " & rng.Parent.Name _
                    & " - Zelle " & rng.Address(False, False)
                Loop While Not rng Is Nothing And rng.Address <> sFirst
            bln = True
        End If
    Next iCounter
    If bln = False Then
        Beep
        MsgBox prompt:="Lagernummer nicht gefunden!"
    End
End If
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten in zweidimensionales Array ein- und in Tabelle auslesen

Problem:	Die Wochentage und Datumswerte eines Monats sollen zuerst in ein zweidimensionales Array eingelesen und dann in die Tabelle eingetragen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DayArray() Dim arr(1 To 31, 1 To 2) As Variant Dim iCounter As Integer For iCounter = 1 To 31 arr(iCounter, 2) = DateSerial(Year(Date), 12, iCounter) arr(iCounter, 1) = Format(arr(iCounter, 2), "dddd") Next iCounter For iCounter = 1 To 31 Cells(iCounter, 1) = arr(iCounter, 1) Cells(iCounter, 2) = arr(iCounter, 2) Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten in benutzerdefinierten Datentyp ein- und auslesen

Problem: Die Daten aus einer Tabelle sollen in Felder eines benutzerdefinierten Datentyps ein- und von dort ausgelesen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Type Adresse
    Vorname As String
    Nachname As String
    Strasse As String
    PLZ As String
    Ort As String
End Type

Sub AdresseEintragen()
    Dim iRow As Integer
    Dim sAdr As String
    Dim MyAddresses(1 To 5) As Adresse
    For iRow = 2 To 6
        With MyAddresses(iRow - 1)
            .Vorname = Cells(iRow, 1)
            .Nachname = Cells(iRow, 2)
            .Strasse = Cells(iRow, 3)
            .PLZ = Cells(iRow, 4)
            .Ort = Cells(iRow, 5)
        End With
    Next iRow
    For iRow = 2 To 6
        With MyAddresses(iRow - 1)
            sAdr = .Vorname & " " & _
                .Nachname & vbLf & _
                .Strasse & ", " & _
                .PLZ & " " & _
                .Ort
        End With
        MsgBox sAdr
        sAdr = ""
    Next iRow
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Feldvariable auf Inhalt testen

Problem:	Wie kann ich feststellen, ob ein bestimmter Wert in einer Matrix enthalten ist, ohne daß die einzelnen Felder durchlaufen werden müssen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ArrayTest() Dim arrTest(1 To 12) Dim var As Variant Dim iCounter As Integer For iCounter = 1 To 12 arrTest(iCounter) = iCounter Next iCounter var = Application.Match(7, arrTest, 0) If IsError(var) Then MsgBox "Der Wert wurde nicht gefunden!" Else MsgBox "Position: " & var End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Variable in Zwischenablage übernehmen

Problem:	Wie kann ich den in einer Variablen gespeicherten Wert in die Zwischenablage übernehmen, so daß ich ihn mit STRG+V an beliebige Stellen des Arbeitsblattes einfügen kann?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Var2Clipboard() Dim ClipAbLage As DataObject Dim sTxt As String Set ClipAbLage = New DataObject Range("A1").Select sTxt = InputBox("Variablenwert:", , "Hallo!") ClipAbLage.SetText sTxt ClipAbLage.PutInClipboard ActiveSheet.Paste End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einlesen von Zellinhalten in Variablen

Problem:	Wie kann ich Zellinhalte in Variablen einlesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: mHWH Sub WerteZuweisen() Dim arr As Variant Dim iCounter As Integer arr = Range("A1").CurrentRegion.Value For iCounter = 1 To UBound(arr) MsgBox arr(iCounter, 1) Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblätter kopieren, wenn Email-Adresse vorhanden

Problem: Alle Tabellenblätter, die in Zelle C5 eine abzufragende Email-Adresse haben, sollen in eine neue Arbeitsmappe kopiert werden.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub CopyWks()
    Dim wkb As Workbook
    Dim wks As Worksheet
    Dim sEmail
    Application.ScreenUpdating = False
    sEmail = InputBox("Email-Adresse:", , "hans@herber.de")
    If sEmail = "" Then Exit Sub
    Set wkb = ThisWorkbook
    For Each wks In wkb.Worksheets
        If wks.Range("C5").Value = sEmail Then
            If wkb.Name = ActiveWorkbook.Name Then
                wks.Copy
            Else
                wks.Copy after:=Worksheets(Worksheets.Count)
            End If
        End If
    Next wks
    Worksheets(1).Select
    ActiveSheet.Buttons.Delete
    Rows("7:16").Delete
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten aus zwei Blättern in einem Zielblatt zusammenfassen

Problem:	Die Daten aus zwei Blättern sollen im Blatt Daten zusammengefaßt werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: Modul1

```
Sub Zusammenfuehren()  
    Dim iRow As Integer, iRowS As Integer, iRowT As Integer  
    Dim wksA As Worksheet, wksB As Worksheet  
    Set wksA = Worksheets("Artikel")  
    Set wksB = Worksheets("Daten")  
    iRowT = 1  
    iRow = 2  
    Do Until IsEmpty(Cells(iRow, 1))  
        iRowS = 2  
        Do Until IsEmpty(wksA.Cells(iRowS, 1))  
            If Cells(iRow, 1).Value = wksA.Cells(iRowS, 1).Value Then  
                iRowT = iRowT + 1  
                wksB.Cells(iRowT, 1).Value = Cells(iRow, 1).Value  
                wksB.Cells(iRowT, 2).Value = Cells(iRow, 2).Value  
                wksB.Cells(iRowT, 3).Value = Cells(iRow, 3).Value  
                wksB.Cells(iRowT, 4).Value = wksA.Cells(iRowS, 2).Value  
                wksB.Cells(iRowT, 5).Value = wksA.Cells(iRowS, 3).Value  
            End If  
            iRowS = iRowS + 1  
        Loop  
        iRow = iRow + 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Liste nach Namen verdichten

Problem:	Die Liste soll so verdichtet werden, dass jeder Name nur noch einmal vorkommt und die Stückahlen jeweils addiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Sammeln() Dim vRow As Variant Dim iRow As Integer, iRowL As Integer iRowL = WorksheetFunction.CountA(Columns(1)) For iRow = iRowL To 2 Step -1 vRow = Application.Match(Cells(iRow, 1).Value, Range(Cells(1, 1), Cells(iRow - 1, 1)), 0) If Not IsError(vRow) Then Cells(iRow, 4).Value = Cells(iRow, 4).Value + Cells(vRow, 4).Value Rows(vRow).Delete End If Next iRow Range("A1").CurrentRegion.Sort _ key1:=Range("A2"), order1:=xlAscending End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensätze mit Minimumwert übernehmen

Problem:	In Spalte D:E sollen die Datensätze eingetragen werden, die in Spalte B dem Minimum entsprechen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetMin() Dim iRow As Integer, iRowT As Integer iRow = 2 iRowT = 1 Do Until IsEmpty(Cells(iRow, 1)) If Cells(iRow, 2).Value = WorksheetFunction.Min(Columns(2)) Then iRowT = iRowT + 1 Cells(iRowT, 4).Value = Cells(iRow, 1).Value Cells(iRowT, 5).Value = Cells(iRow, 2).Value End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textmuster vergleichen

Problem: Eine Zeichenfolge wird daraufhin überprüft, ob sie ein Datum im Format ##.##.#### enthält.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub GetAnswer()
    Dim sPattern As String
    sPattern = "##.##.####"
    MsgBox FindPattern(Range("A1").Value, sPattern)
End Sub

Function FindPattern(sString As String, sPattern As String) As Boolean
    Dim iChar As Integer
    For iChar = 1 To Len(sString)
        If Mid(sString, iChar, Len(sPattern)) Like sPattern Then
            FindPattern = True
            Exit Function
        End If
    Next iChar
    FindPattern = False
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensätze wechselseitig bei Nichtauffinden markieren

Problem: Die Preise in Tabelle1 sind durch Preise in Tabelle2 zu ersetzen. Wird ein Artikel in Tabelle2 nicht gefunden, ist der Hintergrund zu markieren und umgekehrt.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub PreisVergleich()  
    Dim wks As Worksheet  
    Dim vRow As Variant  
    Dim iRow As Integer, iRowL As Integer  
    Set wks = Worksheets("Tabelle2")  
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row  
    For iRow = 1 To iRowL  
        vRow = Application.Match(Cells(iRow, 1).Value, wks.Columns(1), 0)  
        If IsError(vRow) Then  
            Cells(iRow, 1).Interior.ColorIndex = 36  
        Else  
            Cells(iRow, 2).Value = wks.Cells(vRow, 2).Value  
        End If  
    Next iRow  
    iRowL = wks.Cells(Rows.Count, 1).End(xlUp).Row  
    For iRow = 1 To iRowL  
        vRow = Application.Match(wks.Cells(iRow, 1).Value, Columns(1), 0)  
        If IsError(vRow) Then  
            wks.Cells(iRow, 1).Interior.ColorIndex = 36  
        End If  
    Next iRow  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur SVERWEIS eintragen, wenn Wert vorhanden

Problem:	Es soll geprüft werden, ob die Werte aus Spalte E in Spalte A vorhanden sind. Wenn ja, soll in Spalte F eine SVERWEIS-Formel eingesetzt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetVLookup() Dim vRow As Variant Dim iRow As Integer iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) vRow = Application.Match(Cells(iRow, 5).Value, Columns(1), 0) If Not IsError(vRow) Then Um die Formel einzutragen Cells(iRow, 6).Formula = "=VLOOKUP(" & Cells(iRow, 5).Address & ",A:A,0)" Um Werte einzutragen: 'Cells(iRow, 6).Value = _ WorksheetFunction.VLookup(Cells(iRow, 5).Value, Columns(1), 0) End If iRow = iRow + 1 Loop End Sub</pre>

Teilstrings vergleichen und markieren

Problem:	Aus der ersten Spalte sollen die Teilstrings Position 2 bis 5 und 7 bis 10 der Werte mit den Teilstrings aller Werte aus Spalte B verglichen werden. Gibt es keine Übereinstimmung, wird die Zelle in Spalte A gekennzeichnet.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Vergleich() a = 1 Do While 1 zahl1 = Cells(a, 1) zahl1pos2bis5 = Mid(zahl1, 2, 4) ' bei 123456789100000 ... 2345 zahl1pos7bis10 = Mid(zahl1, 7, 4) ' ... 7891 b = 1 fehler = 0 Do While 1 zahl2 = Cells(b, 2) zahl2pos2bis5 = Mid(zahl2, 2, 4) ' bei 123456789100000 ... 2345 zahl2pos7bis10 = Mid(zahl2, 7, 4) ' ... 7891 If zahl2pos2bis5 <> zahl1pos2bis5 Or zahl1pos2bis5 <> zahl2pos7bis10 Then fehler = 1 End If b = b + 1 If Cells(b, 2) = "" Then Exit Do ' Leere Zelle = Ende Spalte 1 Loop If fehler <> 0 Then MsgBox "Error! Spalte A Reihe " & a & " nicht in Spalte B gefunden!" End If a = a + 1 If Cells(a, 1) = "" Then Exit Do ' Leere Zelle = Ende Spalte 1 Loop End Sub StandardModule: basMain Sub Vergleichen() Dim iRowA As Integer, iCounter As Integer, iFirst As Integer, iSecond As Integer Dim iRowB As Integer Dim sTxtA As String, sTxtB As String Dim bln As Boolean iRowA = 1 iFirst = 1 iSecond = 2 For iCounter = 1 To 2 Do Until IsEmpty(Cells(iRowA, iFirst)) bln = False iRowB = 1 sTxtA = Cells(iRowA, iFirst).Value Do Until IsEmpty(Cells(iRowB, iSecond)) sTxtB = Cells(iRowB, iSecond).Value If Mid(sTxtA, 2, 4) = Mid(sTxtB, 2, 4) And _ Mid(sTxtA, 7, 4) = Mid(sTxtB, 7, 4) Then bln = True Exit Do End If iRowB = iRowB + 1 Loop If bln = False Then Cells(iRowA, iFirst).Interior.ColorIndex = 6 iRowA = iRowA + 1 Loop iFirst = 2 iSecond = 1 iRowA = 1 Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte vergleichen und übertragen

Problem: Die Werte in Spalte A von Tabelle1 sollen mit Spalte A von Tabelle2 verglichen werden. Bei Übereinstimmung sind die Werte aus den den Spalten A und D aus Tabelle2 einzutragen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Vergleichen()
    Dim vRow As Variant
    Dim lRow As Long, lRowT As Long
    Dim bln As Boolean
    bln = Application.DisplayStatusBar
    Application.DisplayStatusBar = True
    lRow = 1
    Do Until IsEmpty(Cells(lRow, 1))
        vRow = Application.Match(Cells(lRow, 1).Value, Worksheets("Tabelle2").Columns(1), 0)
        If Not IsError(vRow) Then
            lRowT = lRowT + 1
            Worksheets("Tabelle3").Cells(lRowT, 1).Value = Cells(lRow, 1).Value
            Worksheets("Tabelle3").Cells(lRowT, 2).Value = Worksheets("Tabelle2").Cells(vRow, 4).Value
        End If
        lRow = lRow + 1
    Loop
    Application.DisplayStatusBar = bln
    Application.StatusBar = False
    Worksheets("Tabelle3").Activate
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelte Vorkommen in einer Zeile löschen

Problem:	Bei Mehrfachvorkommen eines Gruppennamens in einer Zeile sollen die nachfolgenden den Doppel gelöscht werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub PruefenLoeschen() Dim iRow As Integer, iRowL As Integer, iCol As Integer, iColL As Integer Application.ScreenUpdating = False iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 2 To iRowL iColL = Cells(iRow, 256).End(xlToLeft).Column For iCol = iColL To 2 Step -1 If WorksheetFunction.CountIf(Rows(iRow), Cells(iRow, iCol).Value) > 1 Then Cells(iRow, iCol).Delete xlShiftToLeft End If Next iCol Next iRow Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gleiche Einträge markieren

Problem:	Gleiche Einträge - in sortierter Reihenfolge sollen farblich markiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Vergleich() Dim iRow As Integer, iColor As Integer iColor = 41 Cells(1, 1).Interior.ColorIndex = iColor iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) If Cells(iRow, 1).Value <> Cells(iRow - 1, 1).Value Then If iColor = 41 Then iColor = 42 Else iColor = 41 End If Cells(iRow, 1).Interior.ColorIndex = iColor iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vergleich von Zellwerten mit unterschiedlichen Methoden

Problem:	Es gibt verschiedene Methoden, Zellinhalte miteinander zu vergleichen. Die Methoden können zu unterschiedlichen Ergebnissen führen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Vergleich() Dim blnA As Boolean, blnB As Boolean, blnC As Boolean blnA = Range("A1") = Range("B1") blnB = Range("A1").Value = Range("B1").Value blnC = Range("A1").Formula = Range("B1").Formula MsgBox "Ohne Eigenschaft: " & blnA MsgBox "Als Value: " & blnB MsgBox "Als Formula: " & blnC End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Name und Vorname suchen, Werte und Fundstellen importieren

Problem:	Die Werte aus Spalte C der in den Zellen F1 und F2 genannten Person sollen aus den Folgeblättern ausgelesen und mit Fundstellenangabe in A:C gelistet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetValues() Dim iWks As Integer, iRow As Integer, iRowT As Integer Dim sName As String, sSName As String sName = Range("F1").Value sSName = Range("F2").Value For iWks = ActiveSheet.Index + 1 To Worksheets.Count iRow = 1 With Worksheets(iWks) Do Until IsEmpty(.Cells(iRow, 1)) If .Cells(iRow, 1).Value = sName And .Cells(iRow, 2).Value = sSName Then iRowT = iRowT + 1 Cells(iRowT, 1).Value = .Cells(iRow, 3).Value Cells(iRowT, 2).Value = .Name Cells(iRowT, 3).Value = .Cells(iRow, 1).Address(False, False) End If iRow = iRow + 1 Loop End With Next iWks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensätze, die öfter als 3 mal vorkommen, kopieren

Problem: Alle Datensätze, deren Datum in Spalte A öfter als 3 mal vorkommen, sollen in eine neue Arbeitsmappe kopiert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub MatchCopy()
    Dim rng As Range, rCell As Range
    Dim iRow As Integer
    Set rng = Range("A1").CurrentRegion
    Workbooks.Add 1
    For Each rCell In rng.Columns(1).Cells
        If WorksheetFunction.CountIf(rng.Columns(1), rCell.Value) > 3 Then
            iRow = iRow + 1
            rng.Rows(rCell.Row).Copy Cells(iRow, 1)
        End If
    Next rCell
    If IsEmpty(Range("A1")) Then
        ActiveWorkbook.Close savechanges:=False
        Beep
        MsgBox "Keine Datensätze gefunden!"
    Else
        Columns.AutoFit
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In einer Namensliste Großbuchstaben einsetzen

Problem: Die Namen aus Spalte A sollen so umgewandelt werden, dass an den richtigen Positionen Klein- durch Großbuchstaben ersetzt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub ZuGross()
    Dim iRow As Integer, iRowL As Integer, iChr As Integer
    Dim sTxt As String
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row
    For iRow = 1 To iRowL
        If Not IsEmpty(Cells(iRow, 1)) Then
            Cells(iRow, 1).Value = _
                WorksheetFunction.Proper(Cells(iRow, 1).Value)
        End If
    Next iRow
    With Worksheets("NegativListe")
        iRowL = .Cells(Rows.Count, 1).End(xlUp).Row
        For iRow = 1 To iRowL
            Columns(1).Replace _
                WorksheetFunction.Proper(.Cells(iRow, 1).Value), _
                .Cells(iRow, 1)
        Next iRow
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolge auf Inhalt überprüfen

Problem:	Über die VBA-Like-Funktion soll geprüft werden, ob eine bestimmte Zeichenfolge im Wert von Zelle A1 enthalten ist, wobei die Zeichenfolge als Variable zu übergeben ist.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub IsLike() Dim dValue As Double Dim sTxt As String, sEuro As String, sLike As String dValue = 23.21 sEuro = Format(dValue, "0.00 €") sLike = "*" & sEuro & "*" sTxt = Range("A1").Value If sTxt Like sLike Then MsgBox "Im String enthalten!" Else MsgBox "Nicht im String enthalten!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelte aus Array löschen und sortieren

Problem: Aus einem Array sollen die doppelten Einträge gelöscht und das Array soll neu dimensioniert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub DeleteDoubles()  
    Dim col As New Collection  
    Dim arr() As Integer  
    Dim iCounter As Integer, iCount As Integer  
    ReDim arr(1 To 13)  
    For iCounter = 1 To 10  
        arr(iCounter) = iCounter  
    Next iCounter  
    arr(11) = 3  
    arr(12) = 5  
    arr(13) = 6  
    On Error Resume Next  
    For iCounter = 1 To UBound(arr)  
        col.Add arr(iCounter), CStr(arr(iCounter))  
    Next iCounter  
    On Error GoTo 0  
    ReDim arr(1 To col.Count)  
    For iCounter = 1 To col.Count  
        arr(iCounter) = col(iCounter)  
    Next iCounter  
    For iCounter = 1 To UBound(arr)  
        MsgBox arr(iCounter)  
    Next iCounter  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Telefonnummern mit Liste vergleichen und durch Namen ersetzen

Problem:	Die Telefonnummern aus Spalte A sollen mit denen im Blatt "Data" in Spalte B verglichen bei Übereinstimmung durch den jeweilig danebenstehenden Namen ersetzt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Nos2Names() Dim var As Variant Dim iRow As Integer iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) var = Application.Match(Cells(iRow, 1).Value, _ Worksheets("Data").Columns(2), 0) If Not IsError(var) Then Cells(iRow, 1).Value = _ Worksheets("Data").Cells(var, 1).Value End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalte A mit B vergleichen und Doppel in C eintragen

Problem:	Spalte A soll mit Spalte B verglichen und die in beiden Spalten vorkommenden Elemente sollen in Spalte C gelistet werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Vergleichen() Dim var As Variant Dim iRow As Integer, iRowT As Integer iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) var = Application.Match(Cells(iRow, 1), Columns(2), 0) If Not IsError(var) Then iRowT = iRowT + 1 Cells(iRowT, 3) = Cells(iRow, 1) End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In zwei Tabellen identische Zeilen löschen

Problem:	Die Zeilen, deren Werte mit denen aus Spalte A von Tabelle2 identisch sind, sollen gelöscht werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Loeschen() Dim var As Variant Dim iRow As Integer iRow = 1 With Worksheets("Tabelle2") Do Until IsEmpty(.Cells(iRow, 1)) var = Application.Match(.Cells(iRow, 1).Value, Columns(1), 0) If Not IsError(var) Then Rows(var).Delete iRow = iRow + 1 Loop End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Übernahme in 2. Tabelle, wenn Zelle in Spalte D nicht leer ist

Problem:	Wenn die jeweilige Zelle in Spalte D nicht leer ist, sollen die Zeilenwerte in Tabelle2 übernommen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Kopieren() Dim iRow As Integer, iRowT For iRow = 1 To 20 If Not IsEmpty(Cells(iRow, 4)) Then iRowT = iRowT + 1 Worksheets("Tabelle2").Rows(iRowT).Value = _ Rows(iRow).Value End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spaltentexte zusammenfassen und neu belegen

Problem:	Spaltentexte aus C:D sollen zusammengefaßt und Spalte D neu belegt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Editieren() Dim iRow As Integer, iRowL As Integer Application.ScreenUpdating = False iRowL = Cells(Rows.Count, 3).End(xlUp).Row For iRow = 2 To iRowL Cells(iRow, 3) = Cells(iRow, 3) & " - " & Cells(iRow, 4) Next iRow Range(Cells(2, 4), Cells(iRowL, 4)).Value = "489-610" Columns.AutoFit Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung von Werten bis zum Zeit-Endpunkt

Problem: In Tabelle 2 sollen die Messwerte im Zeitbereich von A1 dieser Tabelle + 3 Minuten addiert und zusammen mit der Benennung in B1 ausgegeben werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub MessWert()
    Dim datStart As Date, datEnd As Date
    Dim dValue As Double
    Dim iRow As Integer
    iRow = 1
    With Worksheets("Tabelle2")
        Do Until .Cells(iRow, 1) >= Range("A1").Value
            iRow = iRow + 1
        Loop
        iRow = iRow - 1
        datStart = .Cells(iRow, 1)
        Do Until .Cells(iRow, 1) >= Range("A1").Value + _
            TimeSerial(0, 30, 0) Or IsEmpty(.Cells(iRow, 1))
            dValue = dValue + .Cells(iRow, 35).Value
            iRow = iRow + 1
        Loop
        If datStart = 0 Then
            MsgBox "Keine Werte gefunden!"
        Else
            MsgBox "Messwerte für Stoffklasse " & _
                Range("B1").Value & ":" & vbLf & dValue
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte übertragen, wenn numerisch

Problem: Die Werte der Zeilen in Spalte sollen in ein zweites Tabellenblatt übertragen werden, wenn der Wert in Spalte A numerisch ist.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub PruefenKopieren()  
    Dim wksTarget As Worksheet, wksSource As Worksheet  
    Dim lRowL As Long, lRow As Integer, lRowT As Long  
    Application.ScreenUpdating = False  
    Set wksSource = ActiveSheet  
    Worksheets.Add.Move after:=Worksheets(wksSource.Index)  
    Set wksTarget = ActiveSheet  
    wksSource.Select  
    lRowL = Cells(Rows.Count, 1).End(xlUp).Row  
    For lRow = 1 To lRowL  
        If IsNumeric(Cells(lRow, 1)) And _  
            Not IsEmpty(Cells(lRow, 1)) Then  
            lRowT = lRowT + 1  
            wksTarget.Rows(lRowT).Value = Rows(lRow).Value  
        End If  
    Next lRow  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellen vergleichen und bei Nichtübereinstimmung markieren

Problem: Tabelle1 soll mit Tabelle2 verglichen werden. Bei Nichtübereinstimmung eines Datensatzes (Zeile) ist die Zeile im relevanten Bereich farbig zu hinterlegen.

Lösung: Die Tabellen haben eine unterschiedliche Anzahl von Datensätzen.

VBA-Code: StandardModule: basMain

```
Sub Vergleich()  
    Dim rng As Range  
    Dim lRow As Long, lRowT As Long  
    Dim iCol As Integer  
    Dim bln As Boolean  
    Set rng = Worksheets("Tabelle2").Range("A1").CurrentRegion  
    For lRow = 1 To Range("A1").CurrentRegion.Rows.Count  
        bln = True  
        For lRowT = 1 To rng.Rows.Count  
            For iCol = 1 To 4  
                If Cells(lRow, iCol) <> rng(lRowT, iCol) Then  
                    bln = False  
                    Exit For  
                End If  
            Next iCol  
            If bln = True Then  
                Exit For  
            ElseIf lRowT < rng.Rows.Count Then  
                bln = True  
            End If  
        Next lRowT  
        If bln = False Then  
            Range(Cells(lRow, 1), Cells(lRow, 4)).Interior.ColorIndex = 6  
        End If  
    Next lRow  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellen vergleichen und Zeilen ersetzen

Problem: Die Spalte A der Listen in Tabelle2 und Tabelle3 sollen mit der Liste in Tabelle1 verglichen werden. Bei Übereinstimmung ist die jeweilige Zeile in Tabelle1 zu ersetzen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub SuchenErsetzen()  
    Dim var As Variant  
    Dim lRow As Long  
    Dim iWks As Integer  
    For iWks = 3 To Worksheets.Count  
        With Worksheets(iWks).Range("A1").CurrentRegion  
            For lRow = 1 To .Rows.Count  
                var = Application.Match(.Cells(lRow, 1), Columns(1), 0)  
                If Not IsError(var) Then  
                    Range(Cells(var, 1), Cells(var, 3)).Value = _  
                        .Range(.Cells(lRow, 1), .Cells(lRow, 3)).Value  
                End If  
            Next lRow  
        End With  
    Next iWks  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte vergleichen und Ergebnis in Tabelle eintragen

Problem:	Die Werte-Bereiche der Spalten D:G sollen nacheinander mit Spalte A verglichen werden. Das Ergebnis wird in den Bereich E12:E16 eingetragen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MultiMatch() Dim iCol As Integer Dim rngFirst As Range, rngSecond As Range Set rngFirst = Range("A1:A10") For iCol = 3 To 7 Set rngSecond = Range(Cells(1, iCol), Cells(10, iCol)) Cells(iCol + 9, 5) = BereichVergleich(rngFirst, rngSecond) Next iCol End Sub Private Function BereichVergleich(_ rngEins As Range, rngZwei As Range _) As Boolean Dim rng As Range Dim iCount As Integer Dim Schalter As Boolean For Each rng In rngEins.Cells iCount = iCount + 1 If rng <> rngZwei.Cells(iCount) Then Schalter = True Exit For End If Next rng If Schalter = False Then BereichVergleich = True End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, welche Zellen nach Öffnen verändert wurden

Problem: Wie kann ich feststellen, welche Zellen eines bestimmten Bereiches seit dem Öffnen der Arbeitsmappe verändert wurden?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Vergleich()  
    Dim wks As Worksheet  
    Dim rngChange As Range, rngTest As Range, rngSource As Range  
    Set rngSource = Worksheets("Daten").Range("C5:K39")  
    Set wks = Worksheets("Dummy")  
    For Each rngTest In rngSource.Cells  
        If rngTest.Value <> wks.Range(rngTest.Address).Value Then  
            If rngChange Is Nothing Then  
                Set rngChange = rngTest  
            Else  
                Set rngChange = Union(rngChange, rngTest)  
            End If  
        End If  
    Next rngTest  
    If rngChange Is Nothing Then  
        MsgBox "Keine Veränderungen!"  
    Else  
        rngChange.Select  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktivitäten und Stunden eines vorgegebenen Zeitraums zusammenfassen

Problem:	Wie kann ich aus einer Stundenliste die Aktivitäten und Zeiten zusammenfassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: modMain Sub Zusammenfassen() Dim rng As Range Dim iRow As Integer, iCounter As Integer iRow = 1 iCounter = 1 With Worksheets("Tabelle2") .Range("A2:A" & Rows.Count).ClearContents Do Until IsEmpty(Cells(iRow, 1)) If Cells(iRow, 2) >= .Range("F1") And _ Cells(iRow, 2) <= .Range("F2") Then Set rng = .Columns(1).Find(Cells(iRow, 1), _ lookat:=xlWhole, LookIn:=xlValues) If rng Is Nothing Then iCounter = iCounter + 1 .Cells(iCounter, 1) = Cells(iRow, 1) .Cells(iCounter, 2) = Cells(iRow, 3) Else rng.Offset(0, 1) = _ rng.Offset(0, 1) + Cells(iRow, 3) End If End If iRow = iRow + 1 Loop .Select End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zwei Tabellen ohne Duplikate in einer Dritten zusammenfassen

Problem:	Wie kann ich zwei Tabellen ohne Berücksichtigung von doppelten Datensätzen in einer dritten Tabelle zusammenfassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZusammenFuehren() Dim wks As Worksheet Dim rngA As Range, rngB As Range Dim iRow As Integer, iCounter As Integer, iCol As Integer Application.ScreenUpdating = False Set rngA = Worksheets("Tabelle1").Range("A1").CurrentRegion Set rngB = Worksheets("Tabelle2").Range("A1").CurrentRegion Set wks = Worksheets("Tabelle3") iCol = rngA.Columns.Count If rngB.Columns.Count > iCol Then iCol = rngB.Columns.Count End If For iCounter = 1 To iCol wks.Cells(1, iCounter) = "Spalte" & iCounter Next iCounter wks.Rows(1).Font.Bold = True rngA.Range("A1").CurrentRegion.Copy wks.Range("A2") iRow = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1 rngB.Range("A1").CurrentRegion.Copy wks.Cells(iRow, 1) wks.Range("A1").CurrentRegion.AdvancedFilter Action:=xlFilterCopy, _ CopyToRange:=wks.Cells(1, wks.UsedRange.Columns.Count + 1), _ Unique:=True wks.Range(wks.Cells(1, 1), wks.Cells(1, iCol)). _ EntireColumn.Delete wks.Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Überprüfung auf Übereinstimmung und Anzeige in MsgBox

Problem: Wie kann ich die Werte in Spalte A vergleichen und bei Übereinstimmung in einer MsgBox anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub ControlsTest()  
    Dim arr(1 To 18)  
    Dim iCounter As Integer, iScnd As Integer  
    For iCounter = 1 To 18  
        arr(iCounter) = Cells(iCounter, 1)  
    Next iCounter  
    For iCounter = 1 To 18  
        iScnd = iCounter + 1  
        Do Until iScnd >= 18  
            If arr(iCounter) = arr(iScnd) Then  
                MsgBox _  
                    prompt:="Wert " & iCounter & _  
                        " ist gleich Wert " & iScnd  
            End If  
            iScnd = iScnd + 1  
        Loop  
    Next iCounter  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl Artikelnummern auf Deckblatt nach Blättern zählen

Problem: Wie kann ich Artikelnummern, die in einer Reihe von Arbeitsblättern mehrfach vorkommen, auf einem Deckblatt zusammenzählen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub ArtikelnummernListen()  
    Dim rng As Range  
    Dim iCounter As Integer, iRow As Integer, _  
        iCol As Integer, iRowL As Integer  
    Cells.ClearContents  
    iCol = 2  
    For iCounter = 3 To Worksheets.Count  
        With Worksheets(iCounter)  
            Cells(1, iCol) = .Name  
            iRow = 1  
            Do Until IsEmpty(.Cells(iRow, 1))  
                Set rng = Columns(1).Find(.Cells(iRow, 1), _  
                    lookat:=xlWhole, LookIn:=xlValues)  
                If rng Is Nothing Then  
                    iRowL = Cells(Rows.Count, 1).End(xlUp).Row + 1  
                    Cells(iRowL, 1) = .Cells(iRow, 1)  
                    Cells(iRowL, iCol) = 1  
                Else  
                    rng.Offset(0, iCol - 1) = _  
                        rng.Offset(0, iCol - 1) + 1  
                End If  
                iRow = iRow + 1  
            Loop  
        End With  
        iRow = 1  
        iCol = iCol + 1  
    Next iCounter  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob Parallelzellen übereinstimmen

Problem:	Wie kann ich mir durch eine MessageBox anzeigen lassen, welche Werte in den Spalten A und B übereinstimmen und welche nicht?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NamensVergleich() Dim iRow As Integer iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) If Cells(iRow, 1).Value = Cells(iRow, 25).Value Then MsgBox "Namen in Zeile " & iRow & " stimmen überein!" Else MsgBox "Namen in Zeile " & iRow & " stimmen nicht überein!" End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten in jeweils einer Spalte zweier Tabellen vergleichen

Problem: Wie kann ich die Daten in jeweils einer Spalte von zwei Tabellen vergleichen und doppelte Datensätze löschen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub DoppelteRaus()  
    Dim wksA As Worksheet, wksB As Worksheet  
    Dim rng As Range  
    Dim iRow As Integer  
    Set wksA = Worksheets("Tabelle1")  
    Set wksB = Worksheets("Tabelle2")  
    iRow = 1  
    Do Until IsEmpty(wksA.Cells(iRow, 1))  
        Set rng = wksB.Columns(1).Find(wksA.Cells(iRow, 1))  
        If Not rng Is Nothing Then  
            wksA.Rows(iRow).Delete  
            wksB.Rows(rng.Row).Delete  
        End If  
        iRow = iRow + 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte vergleichen

Problem:	Wie kann ich prüfen, ob ein Zellwert in einer Matrix enthalten ist?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Vergleich() Dim arr As Variant, var As Variant arr = Array(52, 54, 58, 72, 30) var = Application.Match(Range("A1").Value, arr, 0) If IsError(var) Then MsgBox "Wert ist nicht in der Matrix enthalten!" Else MsgBox "Wert ist in der Matrix enthalten!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalten A und B vergleichen und Doppel in C listen

Problem:	Wie kann ich die Spalten A und B nach doppelten Zellinhalten durchsuchen lassen und die gefundenen doppelten Werte in Spalte C auflisten lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub VergleichKopieren() Dim rng As Range, rngCell As Range Dim iRow As Integer Set rng = Range("A1").CurrentRegion For Each rngCell In rng.Cells If WorksheetFunction.CountIf(_ rng, rngCell.Value) > 1 Then If WorksheetFunction.CountIf(_ Columns(3), rngCell.Value) < 1 Then iRow = iRow + 1 Cells(iRow, 3).Value = rngCell.Value End If End If Next rngCell End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelt vorkommende Werte sammeln

Problem:	Wie kann ich die Datensätze in zwei Tabellen vergleichen und die doppelt vorkommenden Werte in einem dritten Arbeitsblatt sammeln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Vergleich() Dim wks As Worksheet, wksSec As Worksheet Dim iRow As Integer, iAct As Integer, iRowC As Integer Dim iRowT As Integer, iCol As Integer, iColC As Integer Dim iRowS As Integer Dim bln As Boolean Set wks = ActiveSheet Set wksSec = Worksheets("Spieler2") iRowC = WorksheetFunction.CountA(Columns(1)) iRowS = WorksheetFunction.CountA(wksSec.Columns(1)) iColC = WorksheetFunction.CountA(Rows(1)) iRow = 1 Worksheets.Add after:=Worksheets(Worksheets.Count) For iRow = 1 To iRowC For iAct = 1 To iRowC If iRow <> iAct Then bln = False For iCol = 1 To iColC If wks.Cells(iRow, iCol).Value <> wksSec.Cells(iAct, iCol).Value Then bln = True Exit For End If Next iCol If bln = False Then iRowT = iRowT + 1 Range(Cells(iRowT, 1), Cells(iRowT, iRowC)).Value = _ wks.Range(wks.Cells(iRow, 1), wks.Cells(iRow, iRowC)).Value End If End If Next iAct Next iRow Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenvergleich und Übernahme der Doppel

Problem:	Wie kann ich die Datensätze in einer Tabelle vergleichen und alle doppelten Datensätze in eine zweite Tabelle übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Vergleich() Dim wks As Worksheet Dim iRow As Integer, iAct As Integer, iRowC As Integer Dim iRowT As Integer, iCol As Integer, iColC As Integer Dim bln As Boolean Set wks = ActiveSheet iRowC = WorksheetFunction.CountA(Columns(1)) iColC = WorksheetFunction.CountA(Rows(1)) iRow = 1 Worksheets.Add after:=Worksheets(Worksheets.Count) For iRow = 1 To iRowC For iAct = 1 To iRowC If iRow <> iAct Then bln = False For iCol = 1 To iColC If wks.Cells(iRow, iCol).Value <> wks.Cells(iAct, iCol).Value Then bln = True Exit For End If Next iCol If bln = False Then iRowT = iRowT + 1 Range(Cells(iRowT, 1), Cells(iRowT, iRowC)).Value = _ wks.Range(wks.Cells(iRow, 1), wks.Cells(iRow, iRowC)).Value End If End If Next iAct Next iRow Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellen vergleichen

Problem: Ich möchte 2 Tabellen in 2 Arbeitsmappen miteinander vergleichen und die Datensätze, die nicht doppelt vorkommen, in einer neuen Arbeitsmappe sammeln. Wie kann ich das Problem über VBA lösen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Vergleichen()  
    Dim wkb As Workbook  
    Dim wksA As Worksheet, wksB As Worksheet, wksC As Worksheet  
    Dim iWks As Integer, iRow As Integer, iRowT As Integer  
    On Error Resume Next  
    For iWks = 1 To 3  
        Set wkb = Workbooks("Mappe" & iWks)  
    Next iWks  
    If Err > 0 Or wkb Is Nothing Then  
        Beep  
        Err.Clear  
        MsgBox prompt:="Die 3 Arbeitsmappen sind nicht vorhanden!"  
        Exit Sub  
    End If  
    On Error GoTo 0  
    Set wksA = Workbooks("Mappe1").Worksheets(1)  
    Set wksB = Workbooks("Mappe2").Worksheets(1)  
    Set wksC = Workbooks("Mappe3").Worksheets(1)  
    iRow = 1  
    Do Until IsEmpty(wksA.Cells(iRow, 1))  
        If WorksheetFunction.CountIf( _  
            wksB.Columns(1), _  
            wksA.Cells(iRow, 1).Value) = 0 Then  
            iRowT = iRowT + 1  
            wksC.Cells(iRowT, 1).Value = wksA.Cells(iRow, 1).Value  
            wksC.Cells(iRowT, 2).Value = wksA.Parent.Name  
        End If  
        iRow = iRow + 1  
    Loop  
    iRow = 1  
    Do Until IsEmpty(wksB.Cells(iRow, 1))  
        If WorksheetFunction.CountIf( _  
            wksA.Columns(1), _  
            wksB.Cells(iRow, 1).Value) = 0 Then  
            iRowT = iRowT + 1  
            wksC.Cells(iRowT, 1).Value = wksB.Cells(iRow, 1).Value  
            wksC.Cells(iRowT, 2).Value = wksB.Parent.Name  
        End If  
        iRow = iRow + 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Formulardaten aus Zellen an Webformular und Ergebnis eintragen

Problem:	Formulardaten aus den Zellen B1 und B2 werden an ein Webformular übergeben und das Ergebnis wird als Webabfrage abgerufen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetValue() On Error Resume Next ActiveSheet.QueryTables(1).Delete On Error GoTo 0 With ActiveSheet.QueryTables.Add(_ Connection:= _ "URL;http://www.herber.de/cgi-bin/samples/myform.pl?value=" & _ Range("B1").Value & "&factor=" & Range("B2").Value & "", _ Destination:=Range("B3")) .Refresh BackgroundQuery:=False End With End Sub StandardModule: Modul2 Sub Makro1() Makro1 Makro Makro am 26.04.2004 von Hans W. Herber aufgezeichnet With Selection.QueryTable .Connection = _ "URL;http://www.herber.de/cgi-bin/samples/myform.pl?value=12&factor=24" .WebSelectionType = xlEntirePage .WebFormatting = xlWebFormattingNone .WebPreFormattedTextToColumns = True .WebConsecutiveDelimitersAsOne = True .WebSingleBlockTextImport = False .WebDisableDateRecognition = False .Refresh BackgroundQuery:=False End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Webabfrage in formatierter Tabelle

Problem: In eine formatierte Tabelle werden per Web-Abfrage Daten importiert.

Lösung: Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Webabfrage über ein Perl-Script

Problem:	Die Webabfrage in Zelle A1 ruft über ein Perl-Script die Anzahl der aktuellen Tagesbeiträge aus Herbers Excel-Forum ab.
Lösung:	Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolgen mit InputTexten ergänzen

Problem:	Den Zellinhalten in Spalte A soll ein über InputBox abzufragender Text hinzugefügt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Hinzufuegen() Dim iRow As Integer Dim sNew As String sNew = InputBox("Neuen Text eingeben:") If sNew = "" Then Exit Sub iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) Cells(iRow, 1).Value = Cells(iRow, 1).Value & " " & sNew iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolge nach Zeilenumbrüchen aufteilen

Problem:	Der Wert in Zelle A1 soll nach Zeilenumbrüchen aufgeteilt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SplitText() Dim vText As Variant Dim iCounter As Integer vText = Split(Range("A1").Value, vbCrLf) If Left(Application.Version, 1) <> "8" Then For iCounter = 0 To UBound(vText) Cells(iCounter + 1, 1).Value = vText(iCounter) Next iCounter Else vText = Range("A1").Value Do While InStr(vText, vbCrLf) iCounter = iCounter + 1 Cells(iCounter, 1).Value = Left(vText, InStr(vText, vbCrLf) - 1) vText = Right(vText, Len(vText) - InStr(vText, vbCrLf)) Loop Cells(iCounter + 1, 1).Value = vText End If End Sub</pre>



String mit Prüfziffer über Dialogeingabe erstellen

Problem:	Nach Auswahl eines Landes aus einer ComboBox und der Eingabe einer 11-stelligen RinderCode-Nummer soll der komplette Code mit Länderkurzbezeichnung und Prüfziffer in einem LabelFeld angezeigt werden.
Lösung:	Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmCode Private Sub cboLaender_Change() If txtCode.TextLength = 11 Then Call txtCode_Change End If End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() ActiveCell.Value = lblCode.Caption Unload Me End Sub Private Sub txtCode_Change() Dim wks As Worksheet Dim iChar As Integer, iCode As Integer Dim sCode As String, sTmp As String If txtCode.TextLength = 0 Then lblCode.Caption = "" Exit Sub End If If Right(txtCode.Text, 1) Like "[0-9]" = False Then txtCode.Text = Left(txtCode.Text, Len(txtCode.Text) - 1) End If Set wks = ThisWorkbook.Worksheets("Länder") If txtCode.TextLength = 11 Then sCode = wks.Cells(cboLaender.ListIndex + 1, 3).Value sCode = sCode & txtCode.Text For iChar = 14 To 1 Step -2 iCode = iCode + CInt(Mid(sCode, iChar, 1)) Next iChar iCode = iCode * 3 For iChar = 13 To 1 Step -2 iCode = iCode + CInt(Mid(sCode, iChar, 1)) Next iChar iCode = Fix(iCode / 10) * 10 + 10 - iCode sTmp = wks.Cells(cboLaender.ListIndex + 1, 2).Value & _ " " & Mid(sCode, 4, 3) & "." & _ Mid(sCode, 7, 4) & "." & _ Mid(sCode, 11, 4) & "." & iCode If Right(sTmp, 2) = "10" Then sTmp = Left(sTmp, Len(sTmp) - 2) & "0" End If iCode = InStr(sTmp, " ") + 1 Do Until Mid(sTmp, iCode, 1) <> "0" Mid(sTmp, iCode, 1) = "µ" iCode = iCode + 1 Loop sTmp = WorksheetFunction.Substitute(sTmp, "µ", "") lblCode.Caption = sTmp cmdOK.SetFocus End If End Sub Private Sub UserForm_Initialize() cboLaender.List = ThisWorkbook.Worksheets("Länder") _ .Range("A1").CurrentRegion.Columns(1).Value cboLaender.ListIndex = 0 End Sub StandardModule: Modul1 Sub DialogAufruf() frmCode.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bundesländernamen durch Abkürzungen ersetzen

Problem:	Die Ländernamen sollen durch die in Tabelle2 hinterlegten Abkürzungen ersetzt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SuchenErsetzen() Dim iRow As Integer With Worksheets("Tabelle2") For iRow = 1 To WorksheetFunction.CountA(.Columns(1)) Cells.Replace _ What:=.Cells(iRow, 1), _ Replacement:=.Cells(iRow, 2), _ LookAt:=xlPart, _ SearchOrder:=xlByRows, _ MatchCase:=True Next iRow End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Überflüssige Leerzeichen in einem vorgegebenen Bereich löschen

Problem:	Die überflüssigen Leerzeichen in einem vorgegebenen Bereich sollen gelöscht werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub KillSpaces() Dim rng As Range For Each rng In ActiveSheet.UsedRange rng.Value = WorksheetFunction.Trim(rng.Value) Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Strings, die wie Zahlen aussehen, in Zahlen umwandeln

Problem:	Nach einer Formatänderung von Zellen von Zahlenformaten zu Text werden die Zahlen zwar linksbündig dargestellt, verlieren ihre Zahleneigenschaft aber erst, wenn sie editiert werden. Die Routine wandelt den ausgewählten Bereich in Zahlen um.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Val2Text() Dim rng As Range For Each rng In Selection.Cells If WorksheetFunction.IsNumber(rng.Value) Then rng.Value = "'" & rng.Value End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text gemäß Zeilenumbrüchen in Einzelzellen aufteilen

Problem:	Ein mit Zeilenumbrüchen eingegebener Zelltext in Zellen aufgeteilt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Aufteilen() Selection.TextToColumns Destination:=Range("A1"), DataType:=xlDelimited, _ TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=False, _ Semicolon:=False, Comma:=False, Space:=False, Other:=True, OtherChar _ :=Chr(10), FieldInfo:=Array(1, 1) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Leerzeichen zwischen Zeichen- und Zahlenfolge entfernen

Problem:	Aus einem String aus einer Buchstaben- und Zahlenfolge soll das Leerzeichen zwischen dem letzten Buchstaben und der ersten Zahl entfernt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub DeleteBars() Dim rng As Range Dim iCounter As Integer Dim sTxt As String For Each rng In Range("A1").CurrentRegion.Columns(1).Cells sTxt = rng.Text On Error Resume Next iCounter = 1 Do While Not IsNumeric(Mid(sTxt, iCounter, 1)) iCounter = iCounter + 1 Loop sTxt = Left(sTxt, iCounter - 2) & _ Right(sTxt, Len(sTxt) - iCounter + 1) rng.Value = sTxt On Error GoTo 0 Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufhebung der Suchstringbegrenzung von 255 Zeichen

Problem:	Excel findet über das Suchen-Feature keine Texte, die länger als 255 Zeichen sind. Gesucht wird eine VBA-Lösung als Behebung des Problems.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SuchenFinden() Dim sTxt As String Dim iRow As Integer sTxt = Worksheets("Tabelle2").Range("A1").Value For iRow = 1 To WorksheetFunction.CountA(Columns(1)) If Cells(iRow, 1).Value = sTxt Then MsgBox Cells(iRow, 1).Address End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufteilen eines Strings mit Zeilenumbrüchen

Problem:	Eine Zeichenfolge soll gemäß den enthaltenen Zeilenumbrüchen in mehrere Zellen aufgeteilt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Aufteilen() Dim iRow As Integer Dim sTxt As String Range("A1").Select sTxt = ActiveCell.Value Do While InStr(sTxt, vbLf) ActiveCell.Offset(iRow, 1).Value = _ Left(sTxt, InStr(sTxt, vbLf) - 1) sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, vbLf)) iRow = iRow + 1 Loop ActiveCell.Offset(iRow, 1).Value = sTxt End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle "u" bis zum aktuellen Datum in "U" umwandeln

Problem:	Alle "u" bis zum aktuellen Datum sollen in "U" umgewandelt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub UCaseU() Dim iRow As Integer, iRowL As Integer, iCol As Integer iCol = 1 Do While Cells(1, iCol).Value <= Date iRowL = Cells(Rows.Count, iCol).End(xlUp).Row For iRow = 2 To iRowL If Cells(iRow, iCol).Value = "u" Then Cells(iRow, iCol).Value = "U" End If Next iRow iCol = iCol + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit ShortCut Textbaustein einfügen

Problem: Mit dem Tastaturkürzel Strg+t soll in die aktive Zelle ein Textbaustein eingefügt werden.

Lösung: Das nachfolgende Makro in ein Standardmodul einfügen. Dem Makro wurde über Menü EXTRAS / MAKRO / MAKROS / OPTIONEN ein ShortCut zugewiesen.

VBA-Code:

```
StandardModule: basMain

Sub TextBaustein()
    ActiveCell.Value = "Dies ist mein Text"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einer Zeichenfolge eine zweite in anderer Schriftart hinzufügen

Problem:	Wie kann ich einem Zelleintrag eine Zeichenfolge mit anderer Schriftfarbe und mit anderen Schriftattributen hinzufügen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub TextKorrigieren() Dim sOld As String Dim sNew As String sOld = Range("A1").Text sNew = "ZweiterText" With Range("A1") .Value = sOld & sNew With .Characters(Start:=1, Length:=Len(sOld)).Font .Name = "Arial" .Size = 10 .ColorIndex = xlAutomatic End With With .Characters(Start:=Len(sOld) + 1, Length:=Len(sNew)).Font .Name = "Arial" .Size = 6 .ColorIndex = 3 End With End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolge analysieren und Zeichen auflisten

Problem:	Wie kann ich eine lange Zeichenfolge analysieren, indem ich in einer zweiten Tabelle ein Liste aller enthaltenen Zeichen mit der Anzahl ihres Vorkommens auflisten lasse?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Analyse() Dim iRow As Integer, iCounter As Integer, iRowT As Integer Dim sTxt As String Dim bln As Boolean sTxt = Range("A1").Value Workbooks.Add Columns("A:B").NumberFormat = "@" Columns("A:B").HorizontalAlignment = xlCenter Range("A1") = "REPORT" Range("A1").Font.Bold = True iRow = 1 iRowT = 2 For iCounter = 1 To Len(sTxt) Do Until IsEmpty(Cells(iRowT, 1)) If Cells(iRowT, 1) = Mid(sTxt, iCounter, 1) Then bln = True Exit Do End If iRowT = iRowT + 1 Loop If bln = True Then Cells(iRowT, 2) = Val(Cells(iRowT, 2)) + 1 & " mal" Else iRow = iRow + 1 Cells(iRow, 1) = Mid(sTxt, iCounter, 1) Cells(iRow, 2) = "1 mal" End If bln = False iRowT = 2 Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Positionen einzelner Zeichen in langer Zeichenfolge auslesen

Problem:	Wie kann ich die Position einzelner Zeichen in einem fast 10.000 Zeichen enthaltender Zeichenfolgen auslesen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Lesen() Dim arr() Dim iCounter As Integer, iA As Integer, iB As Integer Dim iC As Integer, id As Integer Dim sTxt As String Columns("B:E").ClearContents Application.ScreenUpdating = False sTxt = Range("A1").Value iA = 1: iB = 1: iC = 1: id = 1 For iCounter = 1 To Len(sTxt) Select Case Mid(sTxt, iCounter, 1) Case "A" iA = iA + 1 Cells(iA, 2) = iCounter Case "B" iB = iB + 1 Cells(iB, 3) = iCounter Case "C" iC = iC + 1 Cells(iC, 4) = iCounter Case "d" id = id + 1 Cells(id, 5) = iCounter End Select Next iCounter Range("B1") = "A" Range("C1") = "B" Range("D1") = "C" Range("E1") = "d" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Werte mit Punkt aus Spalte A in Spalte B

Problem:	Wie kann ich alle Werte aus Spalte A, die einen Punkt beinhalten, in Spalte B verschieben?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Verschieben() Dim iRowL As Integer, iRow As Integer iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = iRowL To 1 Step -1 If InStr(Cells(iRow, 1), ".") Then Cells(iRow - 1, 2).Value = Cells(iRow, 1).Value Rows(iRow).Delete End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchen von Nachnamen in Zeichenfolgen

Problem:	Wie kann ich in einem Zellbereich einen bestimmten Nachnamen ermitteln? Vor- und Nachnamen stehen gemeinsam in einer Zelle. Es muss zwischen Vor- und Nachnamen unterschieden werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NachName() Dim rng As Range Dim iCounter As Integer Dim sName As String sName = "hansen" For Each rng In Range("A1:Z1").Cells If Not IsEmpty(rng) Then For iCounter = Len(rng.Value) To 1 Step -1 If Mid(rng.Value, iCounter, 1) = " " Then Exit For Next iCounter If iCounter = 0 Then If LCase(rng.Value) = sName Then rng.Select Exit Sub End If Else If LCase(Mid(rng.Value, iCounter + 1, _ Len(rng.Value) - iCounter)) = sName Then rng.Select Exit Sub End If End If End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichen in Zeichenfolgen umwandeln und Leerzeichen entfernen

Problem:	Wie kann ich in allen Zeichenfolgen in Spalte A die Punkte durch Leerzeichen ersetzen und die überflüssigen Leerzeichen entfernen. Das Ergebnis soll aus Kleinbuchstaben bestehen und in Spalte B ausgegeben werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Umwandeln() Dim lRow As Long Application.ScreenUpdating = False lRow = Cells(Rows.Count, 1).End(xlUp).Row Range("B1").Formula = "=lower(trim(substitute(A1,"". """, "" "")))" Range("B1:B" & lRow).FillDown Columns("B").Copy Columns("B").PasteSpecial Paste:=xlValues Application.CutCopyMode = False Range("A1").Select Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalte vor dem Verbinden der Zellen zusammenfassen

Problem: Wenn ich Zellen verbinde, gehen die Zellinhalte der Folgezellen verloren. Wie kann ich das verhindern?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Verbinden()  
    Dim rng As Range  
    Dim var As Variant  
    Dim iCounter As Integer  
    Select Case ActiveSheet.Buttons(Application.Caller).Caption  
        Case "Verbinden mit Zeilenumbruch"  
            var = vbLf  
        Case "Verbinden mit Leerzeichen"  
            var = " "  
        Case "Verbinden mit Komma"  
            var = ","  
    End Select  
    For Each rng In Range("A1").CurrentRegion.Columns(1).Cells  
        rng.Value = rng.Value & var & _  
            rng.Offset(0, 1).Value & var & _  
            rng.Offset(0, 2).Value  
        rng.WrapText = True  
        Application.DisplayAlerts = False  
        Range(rng, rng.Offset(0, 2)).Merge  
        Application.DisplayAlerts = True  
    Next rng  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Als Text vorhandene Werte in Zahlen umwandeln

Problem:	Wie kann ich als Text vorhandene Werte in Zahlen umwandeln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub InZahlen1() With ActiveSheet.UsedRange .Value = .Value End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolgen ändern, wenn ein sie einen bestimmten Text enthalten

Problem: Wie kann ich alle Zeichenfolgen in Spalte A ändern, die einen bestimmten Text enthalten?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub StringEdit()
    Dim rng As Range
    Dim iCounter As Integer
    For Each rng In Range("A1").CurrentRegion.Cells
        If Left(rng, 4) = "Test" Then
            iCounter = iCounter + 1
            rng = "Testzeile" & iCounter
        End If
    Next rng
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahlenstring mit Formel auslesen und Fehlermeldung verhindern

Problem:	Wie kann ich aus einem Zahlenstring bestimmte, sich in der Kopfzeile befindliche Begriffe auslesen und beim Nichtvorhandensein eines dieser Begriffe einen Fehlerwert verhindern?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Worte in einem vorgegebenen Bereich zählen

Problem:	Wie kann ich die Anzahl der Worte in einem mit der Zelle A1 zusammenhängenden Bereich zählen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub WorteZaehlen() Dim rng As Range Dim lAdd As Long For Each rng In Range("A1").CurrentRegion Range("IV1").FormulaArray = "=len(" & rng.Address & _ ")-len(substitute(substitute(" & rng.Address & _ ", "" "" , """"), "" "" , """"))+1" lAdd = lAdd + Range("IV1").Value Next rng Range("IV1").ClearContents MsgBox lAdd End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolge AAAA bis DDDD listen

Problem: Wie kann ich alle Kombinationsmöglichkeiten einer Zeichenfolge von AAAA bis DDDD in einer Tabelle auflisten?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Kombinationen()
Dim sArr As Variant
Dim i As Integer, y As Integer, x As Integer
Dim z As Integer, R As Integer
    sArr = Array("A", "B", "C", "D")
    For i = 0 To UBound(sArr)
        For y = 0 To UBound(sArr)
            For x = 0 To UBound(sArr)
                For z = 0 To UBound(sArr)
                    R = R + 1
                    Cells(R, 1) = _
                        sArr(i) & sArr(0 + y) & sArr(0 + x) & sArr(0 + z)
                Next z
            Next x
        Next y
    Next i
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einzelnes Zeichen aus einer InputBox-Eingabe auslesen

Problem:	Wie kann ich ein einzelnes Zeichen aus der Eingabe in eine InputBox auslesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub EingabeBox() Dim sInput As String sInput = InputBox("Zeichenfolge:") If sInput = "" Then Exit Sub sInput = Mid(sInput, 3, 1) MsgBox sInput End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text kürzen

Problem:	Wie kann ich den Text in einer Zelle per VBA-Makro auf 15 Zeichen kürzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Cutten() Dim rng As Range For Each rng In Range("A1:A20").Cells If WorksheetFunction.IsText(rng) Then rng.Value = Left(rng.Value, 15) End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Automatisches Kompletieren einer Zelleingabe

Problem:	Das Autoausfüllen von Zellen wird durch leere Zellen unterbrochen. Wie kann ich das ändern?
Lösung:	Geben Sie in die leeren Zellen ein Apostroph (Hochkomma) ein.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In eine InputBox eingegebene Daten in Elemente aufteilen

Problem:	Wie kann ich in eine InputBox eingegebene Daten in Variablen aufteilen? Als Trennzeichen ist ein Semikolon vorgesehen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Aufteilen() Dim col As New Collection Dim iCounter As Integer Dim sTxt As String, sPart As String sTxt = InputBox(_ prompt:="Zeichenfolge:" & vbCrLf & _ "(Semikoli als Feldtrenner)", _ Default:="ab;cd;ef;gh") Do col.Add Left(sTxt, InStr(sTxt, ";") - 1) sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, ";")) If InStr(sTxt, ";") = False Then col.Add sTxt Exit Do End If Loop For iCounter = 1 To col.Count MsgBox col(iCounter) Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abkürzungen umwandeln

Problem:	In einer Liste enthaltene Kürzel sollen durch Komplettbegriffe ersetzt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CopyShortys() Dim wks As Worksheet Dim rngAll As Range, rngAct As Range Dim rngShortys As Range Set wks = Worksheets("Shortys") Set rngShortys = wks.Columns("A:A") For Each rngAct In Selection If Not IsError(Application.Match(rngAct, rngShortys, 0)) Then rngAct = wks.Cells(Application.Match(rngAct, rngShortys, 0), 2) End If Next rngAct End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichen separieren

Problem:	Wie kann ich eine Zeichenfolge in die darin enthaltenen Einzelzeichen zerlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SplitWord() Dim iCounter As Integer, iLen As Integer Dim sText As String sText = Range("A1").Value iLen = Len(sText) If Len(sText) > 255 Then Beep MsgBox prompt:="Zeichenfolge wird abgeschnitten" iLen = 255 End If For iCounter = 1 To iLen Cells(1, iCounter + 1) = Mid(sText, iCounter, 1) Next iCounter Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolge aufteilen

Problem:	Wie teile ich einen Namen in die einzelnen Elemente bzw. einen Satz in Wörter auf, ohne das Feature Menü DATEN / TEXT IN SPALTEN zu nutzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: mHWH Sub Splitten() Dim iCol As Integer Dim sTxt As String sTxt = Range("A1").Value Do While InStr(sTxt, " ") iCol = iCol + 1 Cells(1, iCol).Value = Left(sTxt, InStr(sTxt, " ") - 1) sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, " ")) Loop Cells(1, iCol + 1).Value = sTxt End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolge korrigieren

Problem: Die Gesellschaftszusätze eines Firmennamens sollen geprüft und ggbf. Korrigiert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub CoReplace()
    Dim arr As Variant
    Dim iCounter As Integer, iStart As Integer
    Dim sCompany As String
    arr = Array("KG", "AG", "KG a.A.", "GmbH", "Co.")
    sCompany = Range("A1").Value
    For iCounter = 0 To 4
        iStart = InStr(1, sCompany, " " & arr(iCounter), 1)
        If iStart > 0 Then
            Mid(sCompany, iStart + 1) = arr(iCounter)
        End If
    Next iCounter
    If MsgBox( _
        prompt:=sCompany, _
        Buttons:=vbQuestion + vbOKCancel _
    ) = vbOK Then
        Range("A1").Value = sCompany
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolgen in Zahlen nach Textimport

Problem:	Nachdem ich Texte importiert habe, erkennt Excel Zahlen nicht als Zahlen und behandelt sie als Zeichenfolgen. Wie kann ich das ändern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub TextInWert() With ActiveSheet.UsedRange .Value = .Value End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolge aufteilen

Problem:	Wie kann ich eine Kommaseparierte Zeichenfolge in Spalten aufteilen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Aufteilen() Columns(1).TextToColumns _ Destination:=Range("A1"), _ DataType:=xlDelimited, _ TextQualifier:=xlDoubleQuote, _ ConsecutiveDelimiter:=False, _ Tab:=False, _ Semicolon:=False, _ Comma:=True, _ Space:=False, _ Other:=False Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sprung zu einer Zellkoordinate und Aufruf des Zellkontextmenüs

Problem:	Die Zelle gem. den Koordinaten in B1:B2 soll ausgewählt und das Zellkontextmenü aufgerufen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub GoToCoordinate() Cells(Range("B1").Value, Range("B2").Value).Select Application.CommandBars("Cell").ShowPopup End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabe von Werten verhindern, wenn Wert in Nebenspalte

Problem:	Die Eingabe von Werten in den Spalten A bzw. B soll verhindert werden, wenn in der jeweiligen Korrespondenzzelle der anderen Spalte bereits ein Wert vorhanden ist.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Markierung des jeweils aktuellen Datums

Problem:	Das jeweils aktuelle Datum soll markiert werden.
Lösung:	Formel in derbedingten Formatierung: =A1=HEUTE()
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte von jeweils zwei Zeilen verbinden

Problem:	Wie kann ich die Werte von zwei aufeinanderfolgenden Zeile zu einem neuen, gemeinsamen Wert verbinden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZellenVerbinden() Dim iRow As Integer, iCounter As Integer, iRowL As Integer Application.ScreenUpdating = False iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iCounter = 1 To iRowL Step 2 iRow = iRow + 1 Cells(iRow, 2) = Cells(iCounter, 1) & _ " - " & Cells(iCounter + 1, 1) Next iCounter Columns(1).Delete Columns(1).AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalte über VBA verbinden

Problem:	Wie kann ich in Spalte A markierte Zellinhalte mit den jeweils rechts davon liegenden Zellinhalten vereinen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: modMain Sub Vereinen() Dim Rng As Range For Each Rng In Selection Rng.Offset(0, 2) = Rng & ";" & Rng.Offset(0, 1) Next Rng Columns("A:B").Delete Columns("A").AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen verbinden, mehrzeiligen Text eingeben und Höhe anpassen

Problem:	Wie kann ich in einem Tabellenblatt mehrere Zeilen verbinden, einen mehrzeiligen Text eingeben und die Zeilenhöhe an den Text anpassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub cmdStart_Click() Dim intRow As Integer Dim strText As String Range("A1:D1").MergeCells = True strText = "Zeile 1" For intRow = 2 To 10 strText = strText & vbCrLf strText = strText & "Zeile " & intRow Next intRow Range("A1:A1") = strText Rows(1).RowHeight = 10 * 13 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adresse der ausgewählten Zelle merken

Problem:	Wie kann ich später feststellen, welches die aktive Zelle vor einer bestimmten Aktion war und diese wieder auswählen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Merken() Dim rngOld As Range Set rngOld = ActiveCell Range("AA1").Select MsgBox "Bin gewandert!" rngOld.Select MsgBox "Bin zurück!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die tatsächliche Anzeige eines Zellinhaltes per VBA abrufen

Problem:	Wie kann ich über VBA prüfen, ob eine Spalte zu schmal konzipiert wurde, also Gitterzäune angezeigt werden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CommandButton1_Click() Call Pruefen End Sub StandardModule: basMain Sub Pruefen() Dim sTxt As String sTxt = Range("A1").Text If InStr(sTxt, "#") And Not IsError(Range("A1")) Then MsgBox prompt:="Spalte zu schmal" Else MsgBox prompt:="Spaltenbreite ok" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen verbinden und trennen

Problem:	Wie kann ich markierte Zellen verbinden bzw. verbundene Zellen trennen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Verbinden() With Selection .MergeCells = Not .MergeCells End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zählen formatierter Zellen

Problem:	Wie kann ich die Anzahl aller mit einer vorgegebenen Zellinnenfarbe markierter Zellen eines vorgegebenen Bereiches zählen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: mHWH Function ColorCount(_ rng As Range, _ iColor As Integer _) As Integer Dim rngAct As Range Dim iCount As Integer For Each rngAct In rng.Cells If rngAct.Interior.ColorIndex = iColor Then iCount = iCount + 1 End If Next rngAct ColorCount = iCount End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in erste leere Zelle eintragen

Problem:	Wie kann ich Eingabelücken mit Werten auffüllen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CopyInFirstEmptyCell() Dim rng As Range On Error GoTo ERRORHANDLER Set rng = Cells.SpecialCells(xlCellTypeBlanks) rng(1) = Range("C1").Value Exit Sub ERRORHANDLER: Beep MsgBox prompt:="Keine frei Zelle gefunden!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellindex eines gefundenen Wertes

Problem:	Wie kann ich einen Wert suchen und statt des gesuchten Wertes nur den Zellenindex ausgeben lassen
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZelleSuchen() Dim rng As Range Dim sSearch As String sSearch = InputBox(prompt:="Suchbegriff", _ Default:="Hallo!") Set rng = Columns(1).Find(_ what:=sSearch, _ lookat:=xlWhole, _ LookIn:=xlValues) If Not rng Is Nothing Then MsgBox prompt:=rng.Address(ReferenceStyle:=xlR1C1) Else MsgBox prompt:="Suchbegriff wurde nicht gefunden!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adressdaten der aktiven Zelle über VBA feststellen

Problem:	Wie kann ich die Zelladresse bzw. die Zeilen-/Spaltennummer der aktiven Zelle über Makro feststellen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: mHWH Sub Zelladresse() Dim sAdr As String With ActiveCell sAdr = "Adresse A: " & .Address & vbCrLf sAdr = sAdr & "Adresse B: " & .Address(_ rowabsolute:=False) & vbCrLf sAdr = sAdr & "Adresse C: " & .Address(_ columnabsolute:=False) & vbCrLf sAdr = sAdr & "Adresse D: " & .Address(_ rowabsolute:=False, columnabsolute:=False) & vbCrLf sAdr = sAdr & "Zeile: " & .Row & vbCrLf sAdr = sAdr & "Spalte: " & .Column End With MsgBox prompt:=sAdr End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelltausch

Problem:	Wie kann ich die Werte und Zahlenformate zweier ausgewählter Zellen tauschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Zelltausch() Dim vValue As Variant Dim sFormat As String If Selection.Cells.Count <> 2 Then Beep MsgBox "Sie müssen 2 Zellen auswählen!" Exit Sub End If With Selection vValue = .Cells(1).Value sFormat = .Cells(1).NumberFormat .Cells(1).Value = Selection.Cells(2).Value .Cells(1).NumberFormat = Selection.Cells(2).NumberFormat .Cells(2).Value = vValue .Cells(2).NumberFormat = sFormat End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenumbruch

Problem:	Wie kann ich per VBA einen Zeilenumbruch in einer Zelle erzeugen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub WrapRow() With ActiveSheet.Range("A1") .WrapText = True .Value = "Hallo" & vbLf & "mein Lieber!" End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Letzte nichtleere Zelle ermitteln

Problem:	Ich möchte die letzte nichtleere Zelle in einer anderen Spalte ermitteln. Wie mache ich das?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub LastCell() Dim rng As Range Dim vCol As Variant vCol = Application.InputBox(_ prompt:="Bitte Spalte als Zahl angeben:", _ Default:=2, _ Type:=1) If vCol = "" Then Exit Sub Set rng = Cells(Rows.Count, CInt(vCol)).End(xlUp) If IsEmpty(rng) Then MsgBox "Keine Zelle mit Inhalt in Spalte " & vCol & "!" Else Range("A1") = rng.Address(False, False) End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Lottozahlen generieren und sortieren

Problem:	In den Bereich A1:E5 sollen nach dem Zufallsprinzip Lottozahlen generiert werden. Die Sortierung erfolgt über Formeln in Spalte B.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Zufall() Dim rng As Range, rngAll As Range Dim iRandomize As Integer Set rngAll = Range("A2:A9") Randomize rngAll.ClearContents For Each rng In rngAll.Cells iRandomize = Int((49 * Rnd) + 1) Do Until WorksheetFunction.CountIf(rngAll, iRandomize) = 0 iRandomize = Int((49 * Rnd) + 1) Loop rng.Value = iRandomize Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auswahl von Werten nach dem Zufallsprinzip

Problem: Aus dem Bereich A1:A10 sollen nach dem Zufallsprinzip 5 Werte ausgewählt werden, ohne dass sich diese wiederholen. Die Werte sollen in ein Array eingelesen und gemeldet werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub Zufall()  
    Dim rng As Range, rngAll As Range  
    Dim arr(1 To 5) As String  
    Dim iRandomize As Integer  
    Dim sMsg As String  
    Application.ScreenUpdating = False  
    Set rngAll = Range("B1:B5")  
    Randomize  
    rngAll.ClearContents  
    For Each rng In rngAll.Cells  
        iRandomize = Int((5 * Rnd) + 1)  
        Do Until WorksheetFunction.CountIf(rngAll, iRandomize) = 0  
            iRandomize = Int((5 * Rnd) + 1)  
        Loop  
        rng.Value = iRandomize  
    Next rng  
    For iRandomize = 1 To 5  
        arr(iRandomize) = Cells(iRandomize, 2).Value  
    Next iRandomize  
    Columns(2).ClearContents  
    For iRandomize = 1 To 5  
        sMsg = sMsg & Cells(arr(iRandomize), 1).Value & vbCrLf  
    Next iRandomize  
    MsgBox sMsg  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorgegebene Anzahl Artikel nach dem Zufallsprinzip auswählen

Problem:	Nach dem Zufallsprinzip sollen aus den Warenlisten im Blatt Listen eine Anzahl Artikel gem. Zelle G1 ausgewählt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ArtikelSchreiben() Dim rngA As Range, rngB As Range Dim iRowCount As Integer, iColCount As Integer Dim iCol As Integer, iRow As Integer, iRandomize As Integer Randomize With Worksheets("Listen") iRowCount = .Range("A1").CurrentRegion.Rows.Count iColCount = .Range("A1").CurrentRegion.Columns.Count Set rngA = .Range(.Cells(2, 1), .Cells(iRowCount, iColCount)) End With Set rngB = Range(Cells(2, 1), Cells(Range("Count").Value + 1, iColCount)) Range(Cells(2, 1), Cells(65536, iColCount)).ClearContents rngB.ClearContents For iCol = 1 To iColCount For iRow = 1 To Range("Count").Value iRandomize = Int((iRowCount * Rnd) + 1) Do Until WorksheetFunction.CountIf(rngB.Columns(iCol), rngB.Cells(iRow, iCol).Value) = 1 rngB.Cells(iRow, iCol).Value = rngA.Cells(iRandomize, iCol).Value iRandomize = Int((iRowCount * Rnd) + 1) Loop Next iRow Next iCol End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Lottozahlen generieren und Ergebnisse auswerten

Problem: Auf Schaltflächenclick sollen im Bereich B2:IV7 Lottozahlen generiert werden, die im Bereich B14:D25 gem. dem Ergebnis in Spalte auszuwerten sind.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code: StandardModule: Modul1

```
Sub SetNumbers()  
    Dim rng As Range, rngAll As Range  
    Dim iCol As Integer, iRow As Integer, iRandomize As Integer  
    Range("B2:IV7").ClearContents  
    For iCol = 2 To 256  
        Set rngAll = Range(Cells(2, iCol), Cells(7, iCol))  
        Randomize  
        For Each rng In rngAll  
            iRandomize = Int((49 * Rnd) + 1)  
            Do Until WorksheetFunction.CountIf(rngAll, iRandomize) = 0  
                iRandomize = Int((49 * Rnd) + 1)  
            Loop  
            rng.Value = iRandomize  
        Next rng  
    Next iCol  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen nach dem Zufallsprinzip in variable Gruppen aufteilen

Problem: Die in Spalte A stehenden Namen sollen nach dem Zufallsprinzip in Gruppen aufgeteilt werden, wobei die Gruppenanzahl variabel ist und sich über eine InputBox abfragen lassen soll.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub ZufallsNamen()  
    Dim rng As Range  
    Dim var As Variant  
    Dim iRowL As Integer, iCell As Integer, iCol As Integer  
    Dim iRow As Integer, iAct As Integer  
    Dim sName As String  
    Columns("B:IV").ClearContents  
    var = Application.InputBox( _  
        prompt:="Anzahl Gruppen:", _  
        Default:=6, Type:=1)  
    If var = "" Then Exit Sub  
    If Not IsNumeric(var) Then Exit Sub  
    For iCol = 1 To CInt(var)  
        Cells(2, iCol + 2) = "Gruppe" & CStr(iCol)  
    Next iCol  
    Randomize  
    iRowL = Range("A1").CurrentRegion.Rows.Count  
    iRow = 3  
    iCol = 3  
    For iCell = 1 To iRowL  
        iAct = Int((iRowL * Rnd) + 1)  
        sName = Cells(iAct, 1).Value  
        Set rng = Range("C2").CurrentRegion.Find( _  
            what:=sName, lookat:=xlWhole, LookIn:=xlValues)  
        Do While Not rng Is Nothing  
            iAct = Int((iRowL * Rnd) + 1)  
            sName = Cells(iAct, 1).Value  
            Set rng = Range("C2").CurrentRegion.Find( _  
                what:=sName, lookat:=xlWhole, LookIn:=xlValues)  
        Loop  
        Cells(iRow, iCol) = sName  
        iCol = iCol + 1  
        If IsEmpty(Cells(2, iCol)) Then  
            iRow = iRow + 1  
            iCol = 3  
        End If  
    Next iCell  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Reihenfolge einer Zahlenreihe nach dem Zufallsprinzip variieren

Problem: Wie kann ich die Reihenfolge der Werte eines ausgewählten Bereiches in den Nebenspalten nach dem Zufallsprinzip 150 mal variieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Zufall()  
    Dim iStart As Integer, iEnd As Integer  
    Dim iCol As Integer, iRow As Integer  
    Dim iZufall As Integer, iCount As Integer  
    Randomize  
    Range("A1").CurrentRegion.Columns(1).Select  
    iCount = Selection.Cells(Selection.Rows.Count)  
    iStart = Selection.Row  
    iEnd = iStart + iCount - 1  
    iZufall = Int((Selection.Rows.Count * Rnd) + 1)  
    For iCol = Selection.Column To Selection.Column + 150  
        For iRow = iStart To iEnd  
            Do Until IsEmpty(Cells _  
                (Selection.Row + iZufall - 1, iCol + 1))  
                iZufall = Int((iCount * Rnd) + 1)  
            Loop  
            Cells(Selection.Row + iZufall - 1, iCol + 1) = _  
                Cells(iRow, iCol)  
        Next iRow  
    Next iCol  
    Range("A1").Select  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte mit Zufallszahl zwischen 1 und 2 multiplizieren

Problem:	Wie kann ich die Werte eines ausgewählten Bereiches mit einer Zufallszahl in der Größe zwischen 1 und 2 multiplizieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZufallMulti() Dim rng As Range Randomize For Each rng In Range("A1:C2").Cells rng.Value = rng.Value * ((1 * Rnd) + 1) Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zufallszahlen bis zum 4-fachen Wert generieren

Problem:	Wie kann ich in einem Tabellenblattbereich Zufallszahlen so generieren, dass maximal der vierfache Wert der jeweiligen Zelle erreicht werden kann?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: modMain Sub Zufall() Dim rng As Range Randomize For Each rng In Range("A1:E5").Cells rng = Int(Rnd * rng.Value * 4) + 1 Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gleiche Zufallszahlen zählen und in Diagramm anzeigen

Problem:	Wie kann ich in Spalte A1:A100 Zufallszahlen von 1 bis 10 eintragen und die jeweilige Anzahl in einem Diagramm ausgeben lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Zufall() Dim intRow As Integer Randomize For intRow = 1 To 100 Cells(intRow, 1) = Int((10 * Rnd) + 1) Next intRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sätze nach dem Zufallsprinzip bilden

Problem: Wie kann ich Sätze nach dem Zufallsprinzip bilden und über eine MsgBox ausgeben lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub ZufallsSatz()  
    Dim arrS As Variant  
    Dim arrP As Variant  
    Dim arrA As Variant  
    Dim arrG As Variant  
    Dim sA As String, sB As String, sC As String, sD As String  
    arrS = Array("Gerhard Schröder ", "Edmund Stoiber ", _  
        "Guido Westerwelle ", "Otto Schily ")  
    arrP = Array("schätzt ", "nutzt ", _  
        "verurteilt ", "genießt ")  
    arrA = Array("den Zauber ", "die Freuden ", _  
        "das Prinzip ", "den Rausch ")  
    arrG = Array("der Macht.", "des Bundestages.", _  
        "der Bundeshauptstadt.", "der Sinneslust.")  
    Randomize  
    sA = arrS(Int((4 * Rnd) + 0))  
    sB = arrP(Int((4 * Rnd) + 0))  
    sC = arrA(Int((4 * Rnd) + 0))  
    sD = arrG(Int((4 * Rnd) + 0))  
    MsgBox sA & sB & sC & sD  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen nach dem Zufallsprinzip auswählen und eintragen

Problem: Aus den Spalten A:B sollen nach dem Zufallsprinzip Namen ausgewählt und in Spalten C:D eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Zufall()  
    Dim rngAct As Range, rngPart As Range, rngFind As Range  
    Dim iCol As Integer, iTmp As Integer  
    Dim iRow As Integer, iCount As Integer  
    Application.ScreenUpdating = False  
    Randomize  
    If IsEmpty(Cells(1, 3)) Then  
        iRow = 1  
    Else  
        iRow = Cells(Rows.Count, 3).End(xlUp).Row + 1  
    End If  
    iCount = WorksheetFunction.CountA(Columns(1))  
    For iCol = 1 To 2  
        Set rngPart = Range(Cells(1, iCol + 2), Cells(iCount, iCol + 2))  
        iTmp = Int((iCount * Rnd) + 1)  
        Set rngFind = rngPart.Find(Cells(iTmp, iCol), _  
            lookat:=xlWhole, LookIn:=xlValues)  
        While Not rngFind Is Nothing  
            iTmp = Int((iCount * Rnd) + 1)  
            Set rngFind = rngPart.Find(Cells(iTmp, iCol), _  
                lookat:=xlWhole, LookIn:=xlValues)  
        Wend  
        Cells(iRow, iCol + 2) = Cells(iTmp, iCol)  
    Next iCol  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Lottozahlen-Generator

Problem:	Wie kann ich Lottozahlen nach dem Zufallsprinzip ermitteln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Lotto() Dim wks As Worksheet Dim rng As Range Dim iLotto As Integer, iCount As Integer Randomize Set wks = Worksheets("Nummern") For iLotto = 1 To 49 wks.Cells(iLotto, 1).Value = iLotto Next iLotto For Each rng In Range("A1:G7").Cells iCount = WorksheetFunction.CountA(wks.Columns(1)) iLotto = Int((iCount * Rnd) + 1) rng.Value = wks.Cells(iLotto, 1).Value wks.Rows(iLotto).Delete Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wertereihe nach dem Zufallsprinzip auf 4 Spalten verteilen

Problem: Wie kann ich die Werte in Spalte A in einem zweiten Tabellenblatt nach dem Zufallsprinzip auf 4 Spalten verteilen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Zufall()  
    Dim wks As Worksheet  
    Dim iCounter As Integer, iRow As Integer  
    Dim iCol As Integer, iAct As Integer  
    Application.ScreenUpdating = False  
    Set wks = Worksheets("Tabelle2")  
    Randomize  
    For iRow = 1 To 64  
        For iCol = 1 To 4  
            iAct = Int((Range("A1").Value * Rnd) + 2)  
            wks.Cells(iRow, iCol).Value = Cells(iAct, 1).Value  
            Rows(iAct).Delete  
        Next iCol  
    Next iRow  
    wks.Columns.AutoFit  
    wks.Select  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Festlegung einer von drei Maßnahmen durch den Zufallsgenerator

Problem: Wie kann ich über einen Zufallsgenerator über 50 Durchläufe hinweg ermitteln lassen, ob Maßnahme 1, 2 oder 3 durchgeführt werden soll?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub ZuFaelle()  
    Dim iRow As Integer  
    Randomize  
    For iRow = 2 To 51  
        Cells(iRow, 2) = Int((3 * Rnd) + 1)  
        Cells(iRow, 3) = Cells(iRow, 2)  
        Do While Cells(iRow, 3) = Cells(iRow, 2)  
            Cells(iRow, 3) = Int((3 * Rnd) + 1)  
        Loop  
        If Cells(iRow, 2) = 1 And Cells(iRow, 3) = 2 Then  
            Cells(iRow, 4) = 3  
        ElseIf Cells(iRow, 2) = 1 And Cells(iRow, 3) = 3 Then  
            Cells(iRow, 4) = 2  
        ElseIf Cells(iRow, 2) = 2 And Cells(iRow, 3) = 1 Then  
            Cells(iRow, 4) = 3  
        ElseIf Cells(iRow, 2) = 2 And Cells(iRow, 3) = 3 Then  
            Cells(iRow, 4) = 1  
        ElseIf Cells(iRow, 2) = 3 And Cells(iRow, 3) = 1 Then  
            Cells(iRow, 4) = 2  
        ElseIf Cells(iRow, 2) = 3 And Cells(iRow, 3) = 2 Then  
            Cells(iRow, 4) = 1  
        End If  
    Next iRow  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zufallszahlen generieren, die sich nicht wiederholen

Problem:	Wie kann ich im Bereich A1:E5 Zufallszahlen von 1 bis 25 eintragen lassen, ohne daß diese sich wiederholen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Zufall() Dim rng As Range, rngAll As Range Dim iRandomize As Integer Set rngAll = Range("A1:E5") Randomize rngAll.ClearContents For Each rng In rngAll.Cells iRandomize = Int((25 * Rnd) + 1) Do Until WorksheetFunction.CountIf(rngAll, iRandomize) = 0 iRandomize = Int((25 * Rnd) + 1) Loop rng.Value = iRandomize Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach dem Zufallsprinzip umsortieren

Problem:	Wie kann ich eine Liste mit Werten nach dem Zufallsprinzip umsortieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Zufall() Dim rngA As Range Dim iRow As Integer, iZufall As Integer Randomize iRow = Selection.Rows.Count Columns(2).ClearContents For Each rngA In Selection.Cells iZufall = Int((iRow * Rnd) + 1) Do While Not IsEmpty(Cells(iZufall, 2)) iZufall = Int((iRow * Rnd) + 1) Loop Cells(iZufall, 2).Value = rngA.Value Next rngA End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte 1 und 0 in einem bestimmten Verhältnis zufällig anordnen

Problem:	Wie kann ich die Werte 0 und 1 in 100 Zellen mit einer vorgegebenen Gewichtung zufällig anordnen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Zufall() Dim iCounter As Integer, iZufall As Integer Range("A1:A100").ClearContents For iCounter = 1 To 100 iZufall = Int((100 * Rnd) + 1) While Not IsEmpty(Cells(iZufall, 1)) iZufall = Int((100 * Rnd) + 1) Wend If iCounter <= Range("C1").Value Then Cells(iZufall, 1) = 0 Else Cells(iZufall, 1) = 1 End If Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Excel-Fenster blinken lassen

Problem:	Das Excel-Fenster blinkt kurz auf.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Private Declare Function FlashWindow Lib "user32.dll" (ByVal hwnd As Long, _ bInvert As Long) As Long Private Declare Function FindWindow _ Lib "user32" _ Alias "FindWindowA" _ (_ ByVal lpClassName As String, _ ByVal lpWindowName As String _) _ As Long Sub Timer1_Timer() Static Retval As Long Retval = FlashWindow(FindWindow("XLMAIN", Application.Caption), Not Retval) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anwendungspfad ermitteln und Anwendung starten

Problem:	Der Kommandozeilen-Pfad von Excel wird ermittelt und Excel wird gestartet.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Declare Function GetCommandLine Lib "kernel32" _ Alias "GetCommandLineA" () As Long Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" _ (lpvDest As Any, lpvSource As Any, ByVal NumBytes As Long) Sub NewExcelInstanz() Dim sShell As String sShell = VBACommand If sShell <> "....." Then Shell VBACommand, vbMaximizedFocus End If End Sub Function VBACommand() As String Dim Buffer() As Byte Dim Addr As Long Dim sTemp As String Dim i As Long Const BufferLen = 256 ReDim Buffer(0 To BufferLen - 1) Addr = GetCommandLine() CopyMemory Buffer(0), ByVal Addr, BufferLen sTemp = StrConv(Buffer(), vbUnicode) i = InStr(sTemp, Chr\$(0)) If i > 0 Then VBACommand = Left\$(sTemp, i - 1) Else VBACommand = sTemp & "....." End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fenstername einer Drittanwendung auslesen

Problem: In Zelle B2 soll der Fenstername der in Zelle B1 genannten Anwendung genannt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Private Declare Function GetForegroundWindow Lib "user32" () As Long

Private Declare Function GetWindowText Lib "user32" _
    Alias "GetWindowTextA" ( _
        ByVal hwnd As Long, _
        ByVal lpString As String, _
        ByVal cch As Long) As Long

Public WndText As String

Public Function GetActiveWindowTitle() As String
    Dim nHwnd As Long
    Dim sTitle As String
    Dim nResult As Long

    nHwnd = GetForegroundWindow()

    sTitle = Space$(255)
    nResult = GetWindowText(nHwnd, sTitle, Len(sTitle))

    GetActiveWindowTitle = Left$(sTitle, nResult)
End Function

Sub Aufruf()
    On Error GoTo ERRORHANDLER
    Shell Range("B1").Value, vbNormalFocus
    Range("B2").Value = GetActiveWindowTitle
    Exit Sub
ERRORHANDLER:
    MsgBox "Das Programm wurde nicht gefunden!"
    Range("B2").ClearContents
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Notepad aufrufen und Routine nach Beenden fortsetzen

Problem:	Notepad soll aufgerufen und das VBA-Programm fortgesetzt werden, wenn Notepad beendet wird.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Private Type STARTUPINFO cb As Long lpReserved As String lpDesktop As String lpTitle As String dwX As Long dwY As Long dwXSize As Long dwYSize As Long dwXCountChars As Long dwYCountChars As Long dwFillAttribute As Long dwFlags As Long wShowWindow As Integer cbReserved2 As Integer lpReserved2 As Long hStdInput As Long hStdOutput As Long hStdError As Long End Type Private Type PROCESS_INFORMATION hProcess As Long hThread As Long dwProcessID As Long dwThreadID As Long End Type Private Declare Function WaitForSingleObject Lib "kernel32" (ByVal _ hHandle As Long, ByVal dwMilliseconds As Long) As Long Private Declare Function CreateProcessA Lib "kernel32" (ByVal _ lpApplicationName As String, ByVal lpCommandLine As String, ByVal _ lpProcessAttributes As Long, ByVal lpThreadAttributes As Long, _ ByVal bInheritHandles As Long, ByVal dwCreationFlags As Long, _ ByVal lpEnvironment As Long, ByVal lpCurrentDirectory As String, _ lpStartupInfo As STARTUPINFO, lpProcessInformation As _ PROCESS_INFORMATION) As Long Private Declare Function CloseHandle Lib "kernel32" _ (ByVal hObject As Long) As Long Private Declare Function GetExitCodeProcess Lib "kernel32" _ (ByVal hProcess As Long, lpExitCode As Long) As Long Private Const NORMAL_PRIORITY_CLASS = &H20& Private Const INFINITE = -1& Public Function ExecCmd(cmdline\$) Dim proc As PROCESS_INFORMATION Dim start As STARTUPINFO ' Initialize the STARTUPINFO structure: start.cb = Len(start) ' Start the shelled application: ret& = CreateProcessA(vbNullString, cmdline\$, 0&, 0&, 1&, _ NORMAL_PRIORITY_CLASS, 0&, vbNullString, start, proc) ' Wait for the shelled application to finish: ret& = WaitForSingleObject(proc.hProcess, INFINITE) Call GetExitCodeProcess(proc.hProcess, ret&) Call CloseHandle(proc.hThread) Call CloseHandle(proc.hProcess) ExecCmd = ret& End Function Sub Form_Click() Dim retval As Long retval = ExecCmd("notepad.exe") MsgBox "Process Finished, Exit Code " & retval End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Windows beenden

Problem:	Wie kann ich Windows auf Schaltflächendruck beenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function ExitWindowsEx Lib "user32" _ (ByVal Flags As Long, _ ByVal Reserved As Long) As Long Sub Beenden() ExitWindowsEx 1, 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anwendung aufrufen und Makroausführung unterbrechen

Problem:	Wie kann ich über ein VBA-Programm eine andere Anwendung aufrufen und die Makroausführung solange unterbrechen, bis die andere Anwendung beendet wird?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Public Const PROCESS_QUERY_INFORMATION = &H400 Public Const WAIT_TIMEOUT = &H102& Declare Function OpenProcess Lib "kernel32" (ByVal dwDesiredAccess As Long, _ ByVal bInheritHandle As Long, ByVal dwProcessId As Long) As Long Declare Function WaitForSingleObject Lib "kernel32" (ByVal hHandle As Long, _ ByVal dwMilliseconds As Long) As Long Sub Aufruf() Call Win32WaitTilFinished("notepad.exe") MsgBox "Notepad ist beendet!" End Sub Sub Win32WaitTilFinished(ProgEXE As String) Dim ProcessID As Long Dim hProcess As Long Dim RetVal As Long ProcessID = Shell(ProgEXE, vbNormalFocus) hProcess = OpenProcess(PROCESS_QUERY_INFORMATION, False, ProcessID) Do DoEvents RetVal = WaitForSingleObject(hProcess, 50) Loop Until RetVal <> WAIT_TIMEOUT End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Import einer Grafikdatei aus dem Internet

Problem:	Eine Grafik-Datei wird aus dem Internet importiert. Das Zielverzeichnis wird abgefragt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Private Declare Function DoFileDownload Lib "shdocvw.dll" _ (ByVal lpszFile As String) As Long Sub GifImport() Dim sPath As String Dim UniCodeURL As String sPath = Application.DefaultFilePath ChDrive Left(sPath, 1) ChDir sPath UniCodeURL = StrConv("http://www.herber.de/images/start1.gif", vbUnicode) DoFileDownload UniCodeURL End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung des Windows-System-Verzeichnisses

Problem:	Das Windows-Systemverzeichnis soll ermittelt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Declare Function GetSystemDirectory Lib "kernel32.dll" _ Alias "GetSystemDirectoryA" (_ ByVal lpBuffer As String, _ ByVal nSize As Long) As Long Sub SystemFolder() Dim lLength As Long Dim sSysDir As String sSysDir = Space(255) lLength = GetSystemDirectory(sSysDir, 255) sSysDir = Left(sSysDir, lLength) MsgBox "Das Systemverzeichnis: " & sSysDir End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung des Windows-Verzeichnisses

Problem:	Das Windows-Verzeichnis soll ermittelt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Declare Function GetWindowsDirectory Lib "kernel32.dll" _ Alias "GetWindowsDirectoryA" (_ ByVal lpBuffer As String, ByVal nSize As Long) As Long Sub WindowsFolder() Dim lLength As Long Dim sWinDir As String sWinDir = Space(255) lLength = GetWindowsDirectory(sWinDir, 255) sWinDir = Left(sWinDir, lLength) MsgBox "Das Windows-Verzeichnis: " & sWinDir End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung des Windows-Temp-Verzeichnisses

Problem:	Das Windows-Temp-Verzeichnis soll ermittelt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function GetTempPath _ Lib "kernel32" Alias "GetTempPathA" _ (ByVal nBufferLength As Long, _ ByVal lpBuffer As String) As Long Public Function fncGetTempPath() As String Dim PathLen As Long Dim WinTempDir As String Dim BufferLength As Long BufferLength = 260 WinTempDir = Space(BufferLength) PathLen = GetTempPath(BufferLength, WinTempDir) If Not PathLen = 0 Then fncGetTempPath = Left(WinTempDir, PathLen) Else fncGetTempPath = CurDir() End If End Function Sub Test() MsgBox fncGetTempPath End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verzeichnisbaum aufrufen und Verzeichnis auswählen

Problem:	Wie kann ich den Verzeichnisbaum anzeigen lassen und ein Verzeichnis auswählen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Option Private Module Public Type BROWSEINFO hOwner As Long pidlRoot As Long pszDisplayName As String lpszTitle As String ulFlags As Long lpfn As Long lParam As Long iImage As Long End Type Declare Function SHGetPathFromIDList Lib "shell32.dll" _ Alias "SHGetPathFromIDListA" (ByVal pidl As Long, ByVal pszPath As String) As Long Declare Function SHBrowseForFolder Lib "shell32.dll" _ Alias "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long Sub DirAuswahl() Dim sMsg As String, sPath As String sMsg = "Wählen Sie bitte einen Ordner aus:" sPath = getdirectory(sMsg) If sPath <> "" Then MsgBox sPath End Sub Function getdirectory(Optional msg) As String Dim bInfo As BROWSEINFO Dim Path As String Dim r As Long, x As Long, pos As Integer bInfo.pidlRoot = 0& If IsMissing(msg) Then bInfo.lpszTitle = "Wählen Sie bitte einen Ordner aus." Else bInfo.lpszTitle = msg End If bInfo.ulFlags = &H1 x = SHBrowseForFolder(bInfo) Path = Space\$(512) r = SHGetPathFromIDList(ByVal x, ByVal Path) If r Then pos = InStr(Path, Chr\$(0)) getdirectory = Left(Path, pos - 1) Else getdirectory = "" End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Unterverzeichnisse mit Inhalt löschen

Problem:	Wie kann ich über VBA-Code einen bestimmten Vereichnisbaum einschließlich der Dateien löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Von Jim Rech Private Const FO_DELETE = &H3& Private Const FOF_ALLOWUNDO = &H40& Private Const FOF_NOCONFIRMATION = &H10& Private Type SHFILEOPSTRUCT hwnd As Long wFunc As Long pFrom As String pTo As String fFlags As Integer fAnyOperationsAborted As Long hNameMappings As Long lpszProgressTitle As String End Type Private Declare Function SHFileOperation Lib "Shell32.dll" Alias _ "SHFileOperationA" (lpFileOp As Any) As Long Sub ShellDelete(SrcFile As String) Dim FileOp As SHFILEOPSTRUCT With FileOp .hwnd = 0 .wFunc = FO_DELETE .pFrom = SrcFile .fFlags = FOF_NOCONFIRMATION + FOF_ALLOWUNDO .lpszProgressTitle = "" End With SHFileOperation FileOp End Sub Sub DeleteFolder() Dim sFolder As String sFolder = InputBox(_ prompt:="Zu löschender Ordner:", _ Default:="c:\MyFolder") If sFolder = "" Then Exit Sub If MsgBox(_ prompt:="Sind Sie sicher?", _ Buttons:=vbQuestion + vbYesNo) = vbYes Then ShellDelete "c:\MyDir" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm ohne Schließkreuz

Problem:	Eine UserForm ohne Schließkreuz soll erstellt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallForm() frmCross.Show End Sub ClassModule: frmCross Private Const GWL_STYLE = (-16) Private Const WS_SYSMENU = &H80000 Private Declare Function _ FindWindow Lib "User32" Alias "FindWindowA" (_ ByVal lpClassName As String, _ ByVal lpWindowName As String) As Long Private Declare Function _ GetWindowLong Lib "User32" Alias "GetWindowLongA" (_ ByVal hWnd As Long, ByVal nIndex As Long) As Long Private Declare Function _ SetWindowLong Lib "User32" Alias "SetWindowLongA" (_ ByVal hWnd As Long, ByVal nIndex As Long, _ ByVal dwNewLong As Long) As Long Private Declare Function _ DrawMenuBar Lib "User32" (ByVal hWnd As Long) As Long Private Sub cmdCancel_Click() Unload Me End Sub Private Sub UserForm_Activate() 'Code von Nepumuk Dim xl_hwnd, lStyle xl_hwnd = FindWindow(vbNullString, Me.Caption) If xl_hwnd <> 0 Then lStyle = GetWindowLong(xl_hwnd, GWL_STYLE) lStyle = SetWindowLong(xl_hwnd, GWL_STYLE, lStyle And Not WS_SYSMENU) DrawMenuBar xl_hwnd End If End Sub</pre>

	Herbers Excel-FAQ - Die Texte Bestandteil der Excel-Lehr- und Lernmaterialien
ComDlg-Dialog aufrufen	
Problem:	Statt des Standard-Öffnendialogs soll der ComDlg-Dialog aufgerufen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Private Declare Function GetOpenFileName Lib "comdlg32.dll" _ Alias "GetOpenFileNameA" (pOpenfilename As OPENFILENAME) As Long Private Type OPENFILENAME lStructSize As Long hwndOwner As Long hInstance As Long lpstrFilter As String lpstrCustomFilter As String nMaxCustFilter As Long nFilterIndex As Long lpstrFile As String nMaxFile As Long lpstrFileName As String nMaxFileName As Long lpstrInitialDir As String lpstrTitle As String flags As Long nFileOffset As Integer nFileExtension As Integer lpstrDefExt As String lCustData As Long lpfnHook As Long lpTemplateName As String End Type Sub CallOpenDialog() Dim OpenFile As OPENFILENAME Dim lReturn As Long Dim sFilter As String OpenFile.lStructSize = Len(OpenFile) sFilter = "All Files (*.*)" & Chr(0) & "*.*" & Chr(0) With OpenFile .lpstrFilter = sFilter .nFilterIndex = 1 .lpstrFile = String(257, 0) .nMaxFile = Len(OpenFile.lpstrFile) - 1 .lpstrFileName = OpenFile.lpstrFile .nMaxFileName = OpenFile.nMaxFile .lpstrInitialDir = Application.DefaultFilePath .lpstrTitle = "ComDlg API statt OCX nutzen" .flags = 0 End With lReturn = GetOpenFileName(OpenFile) If lReturn = 0 Then MsgBox "Suche abgebrochen!", vbInformation Else MsgBox Trim(Left(OpenFile.lpstrFile, _ InStr(1, OpenFile.lpstrFile, Chr(0)) - 1)) End If End Sub Sub CopyText() Dim ClipAbLage As DataObject Dim iRow As Integer Dim sTxt As String Set ClipAbLage = New DataObject iRow = ActiveCell.Row Do Until IsEmpty(Cells(iRow, ActiveCell.Column)) sTxt = sTxt & " " & Cells(iRow, ActiveCell.Column).Value iRow = iRow + 1 Loop sTxt = Right(sTxt, Len(sTxt) - 1) ClipAbLage.SetText sTxt ClipAbLage.PutInClipboard End Sub</pre>

	Herbers Excel-FAQ - Die Texte Bestandteil der Excel-Lehr- und Lernmaterialien
Kopfleiste einer UserForm ein- und ausblenden	
Problem:	Die Titelleiste einer UserForm soll ausgeblendet werden, damit diese nicht verschoben werden kann.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: frmCaption Private Type RECT Left As Long Top As Long Right As Long Bottom As Long End Type Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (_ ByVal lpClassName As String, _ ByVal lpWindowName As String) As Long Private Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA" _ (ByVal hWnd As Long, ByVal nIndex As Long) As Long Private Declare Function SetWindowLong _ Lib "user32" Alias "SetWindowLongA" (ByVal hWnd As Long, _ ByVal nIndex As Long, ByVal dwNewLong As Long) As Long Private Declare Function DrawMenuBar Lib "user32" (_ ByVal hWnd As Long) As Long Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOn_Click() Call fncHasUserformCaption(True) End Sub Private Sub cmdOff_Click() Call fncHasUserformCaption(False) End Sub Private Sub UserForm_Initialize() Call fncHasUserformCaption(False) End Sub Private Function fncHasUserformCaption(bState As Boolean) Dim Userform_hWnd As Long Dim Userform_Style As Long Dim Userform_Rect As RECT Const GWL_STYLE = (-16) Const WS_CAPTION = &HC000000 Userform_hWnd = FindWindow(_ lpClassName:=IIf(Val(Application.Version) > 8, _ "ThunderDFrame", "ThunderXFrame"), _ lpWindowName:=Me.Caption) Userform_Style = GetWindowLong(hWnd:=Userform_hWnd, _ nIndex:=GWL_STYLE) If bState = True Then Userform_Style = Userform_Style Or WS_CAPTION Else Userform_Style = Userform_Style And Not WS_CAPTION End If Call SetWindowLong(hWnd:=Userform_hWnd, nIndex:=GWL_STYLE, _ dwNewLong:=Userform_Style) Call DrawMenuBar(hWnd:=Userform_hWnd) End Function StandardModule: Modul1 Sub DialogAufruf() frmCaption.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Drucken der UserForm im Querformat mit Kopfzeile

Problem: Eine UserForm soll im Querformat mit Einstellungen für die Kopfzeile gedruckt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: frmPrint

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdPrint_Click()
    Application.ScreenUpdating = False
    keybd_event VK_SNAPSHOT, 1, 0, 0
    Workbooks.Add 1
    ActiveSheet.Range("A1").Select
    Application.Wait Now + TimeValue("00:00:01")
    ActiveSheet.PasteSpecial _
        Format:="Bitmap", _
        Link:=False, _
        DisplayAsIcon:=False
    With ActiveSheet
        .PageSetup.Orientation = xlLandscape
        .PageSetup.LeftHeader = "Meine UserForm"
        .PrintOut
    End With
    ActiveWorkbook.Close False
    Application.ScreenUpdating = True
End Sub

StandardModule: Modul1

Declare Sub keybd_event Lib "user32" (ByVal bVk As Byte, _
    ByVal bScan As Byte, ByVal dwFlags As Long, ByVal dwExtraInfo As Long)

Public Const VK_SNAPSHOT = &H2C

Sub CallForm()
    frmPrint.Show
End Sub
```



Installierte Drucker auslesen

Problem:	Die installierten Drucker sollen ausgelesen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	ClassModule: DieseArbeitsmappe StandardModule: Modul1 Type FILETIME dwLowDateTime As Long dwHighDateTime As Long End Type Declare Function RegOpenKeyEx _ Lib "advapi32.dll" _ Alias "RegOpenKeyExA" _ (ByVal hKey As Long, _ ByVal lpSubKey As String, _ ByVal ulOptions As Long, _ ByVal samDesired As Long, _ phkResult As Long _) _ As Long Declare Function RegEnumKeyEx _ Lib "advapi32.dll" _ Alias "RegEnumKeyExA" _ (ByVal hKey As Long, _ ByVal dwIndex As Long, _ ByVal lpName As String, _ lpcbName As Long, ByVal _ lpReserved As Long, _ ByVal lpClass As String, _ lpcbClass As Long, _ lpftLastWriteTime As FILETIME _) _ As Long Declare Function RegCloseKey _ Lib "advapi32.dll" _ (ByVal hKey As Long _) _ As Long Public Function fncEnumInstalledPrintersReg() As Collection Dim tmpFunctionResult As Boolean Dim aFileTimeStruc As FILETIME Dim AddressofOpenKey As Long, aPrinterName As String Dim aPrinterIndex As Integer, aPrinterNameLen As Long Const KEY_ENUMERATE_SUB_KEYS = &H8 Const HKEY_LOCAL_MACHINE = &H80000002 Set fncEnumInstalledPrintersReg = New Collection aPrinterIndex = 0 tmpFunctionResult = Not CBool _ (RegOpenKeyEx _ (hKey:=HKEY_LOCAL_MACHINE, _ lpSubKey:="SYSTEM\CURRENTCONTROLSET\CONTROL\PRINT\PRINTERS", _ ulOptions:=0, _ samDesired:=KEY_ENUMERATE_SUB_KEYS, _ phkResult:=AddressofOpenKey _) _) If tmpFunctionResult = False Then GoTo ExitFunction Do aPrinterNameLen = 255 aPrinterName = String(aPrinterNameLen, CStr(0)) tmpFunctionResult = Not CBool _ (RegEnumKeyEx _ (hKey:=AddressofOpenKey, _ dwIndex:=aPrinterIndex, _ lpName:=aPrinterName, _ lpcbName:=aPrinterNameLen, _ lpReserved:=0, _ lpClass:=vbNullString, _ lpcbClass:=0, _ lpftLastWriteTime:=aFileTimeStruc _) _) aPrinterIndex = aPrinterIndex + 1 If tmpFunctionResult = False Then Exit Do aPrinterName = Left(aPrinterName, aPrinterNameLen) On Error Resume Next fncEnumInstalledPrintersReg.Add aPrinterName On Error GoTo 0 Loop Call RegCloseKey(AddressofOpenKey) Exit Function ExitFunction: If Not AddressofOpenKey = 0 Then _ Call RegCloseKey(AddressofOpenKey) Set fncEnumInstalledPrintersReg = Nothing End Function Sub DruckerAuslesen() Dim aPrinter As Variant Dim iRow As Integer For Each aPrinter In fncEnumInstalledPrintersReg iRow = iRow + 1 Cells(iRow, 1) = aPrinter Next aPrinter End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Outlook-Maildialog aus UserForm aufrufen

Problem: Aus einer UserForm soll auf SchaltflächenKlick Outlook mit dem Mail-Dialog aufgerufen werden.

Lösung: Geben Sie den nachfolgenden Code in die genannten Module.

VBA-Code:

```
ClassModule: frmMail

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdMail_Click()
    Call CallMail("hans@herber.de")
End Sub

StandardModule: Modul1

Private Declare Function ShellExecute Lib "Shell32.dll" _
    Alias "ShellExecuteA" (ByVal hWnd As Long, _
    ByVal lpOperation As String, ByVal lpFile As String, _
    ByVal lpParameters As String, ByVal lpDirectory As String, _
    ByVal nShowCmd As Long) As Long

Sub CallMail( _
    eMail As String)
    Call ShellExecute(0&, "Open", "mailto:" + eMail, "", "", 1)
End Sub

Sub CallForm()
    frmMail.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Inhalt einer Arbeitsmappe als Mail senden (ohne die Mappe selbst)

Problem: Wie kann man den Inhalt einer Excel-Tabelle per Mail versenden, ohne daß die Arbeitsmappe selbst versandt wird?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Private Declare Function ShellExecute Lib "Shell32.dll" _
    Alias "ShellExecuteA" (ByVal hWnd As Long, _
    ByVal lpOperation As String, ByVal lpFile As String, _
    ByVal lpParameters As String, ByVal lpDirectory As String, _
    ByVal nShowCmd As Long) As Long

Private Sub Mail( _
    eMail As String, _
    Optional Subject As String, _
    Optional Body As String)
    Call ShellExecute(0&, "Open", "mailto:" + eMail + _
        "?Subject=" + Subject + "&Body=" + Body, "", "", 1)
End Sub

Sub MailVersenden()
    Dim rng As Range
    Dim sMail As String, sSubject As String
    Dim sBody As String
    Dim iRow As Integer, iCol As Integer
    sMail = "hans@herber.de"
    sSubject = "Excel-Daten"
    Set rng = Range("A1").CurrentRegion
    For iCol = 1 To rng.Columns.Count
        For iRow = 1 To rng.Rows.Count
            sBody = sBody & rng.Cells(iRow, iCol) & " "
        Next iRow
    Next iCol
    Call Mail(sMail, sSubject, sBody)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob Internet-Verbindung besteht

Problem:	Es soll geprüft werden, ob eine Internetverbindung besteht.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Public Declare Function InternetGetConnectedState _ Lib "wininet.dll" (lpdwFlags As Long, _ ByVal dwReserved As Long) As Boolean Function IsConnected() As Boolean Dim lStat As Long IsConnected = (InternetGetConnectedState(lStat, 0&) <> 0) End Function Sub Meldung() Dim bln As Boolean bln = IsConnected If bln Then MsgBox "Internetverbindung besteht!" Else MsgBox "Internetverbindung besteht nicht!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Internetverbindung trennen

Problem:	Eine evtl. bestehende Internetverbindung soll getrennt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 von Fritz Grath Private Type RASCONN dwSize As Long hRasConn As Long szEntryName(256) As Byte szDeviceType(16) As Byte szDeviceName(128) As Byte End Type Private Declare Function RasEnumConnections Lib "RasApi32.DLL" _ Alias "RasEnumConnectionsA" (lprasconn As Any, lpcb As Long, _ lpcConnections As Long) As Long Private Declare Function RasHangUp Lib "RasApi32.DLL" Alias _ "RasHangUpA" (ByVal hRasConn As Long) As Long Sub Verbindung_trennen() Dim lprasconn(255) As RASCONN lprasconn(0).dwSize = 412 lpcb& = 256 * lprasconn(0).dwSize RasEnumConnections lprasconn(0), lpcb&, lpcConnections& If lpcConnections& > 0 Then RasHangUp lprasconn(0).hRasConn End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

DNS und IP abfragen

Problem:	Bei Vorgabe der DNS soll die IP-Adresse, bei Vorgabe der IP-Adresse die zugehörige DNS ermittelt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Private Declare Function WSAGetLastError Lib "WSOCK32.DLL" () _ As Long Private Declare Function WSAStartup Lib "WSOCK32.DLL" (ByVal _ wVersionRequired&, lpWSAData As WinSocketDataType) _ As Long Private Declare Function WSACleanup Lib "WSOCK32.DLL" () _ As Long Private Declare Function gethostname Lib "WSOCK32.DLL" (ByVal _ HostName\$, ByVal HostLen%) As Long Private Declare Function gethostbyname Lib "WSOCK32.DLL" _ (ByVal HostName\$) As Long Private Declare Function gethostbyaddr Lib "WSOCK32.DLL" _ (ByVal addr\$, ByVal laenge%, ByVal typ%) As Long Private Declare Sub RtlMoveMemory Lib "kernel32" (hvpDest As _ Any, ByVal hvpSource&, ByVal cbCopy&) Const WS_VERSION_REQD = &H101 Const SOCKET_ERROR = -1 Const WSADESCRIPTION_LEN = 256 Const WSASYS_STATUS_LEN = 128 Private Type HostDeType hName As Long hAliases As Long hAddrType As Integer hLength As Integer hAddrList As Long End Type Private Type WinSocketDataType wVersion As Integer wHighVersion As Integer szDescription(0 To WSADESCRIPTION_LEN) As Byte szSystemStatus(0 To WSASYS_STATUS_LEN) As Byte iMaxSockets As Integer iMaxUdpDg As Integer lpzVendorInfo As Long End Type Private Declare Function RasEnumConnections Lib "RasApi32.DLL" _ Alias "RasEnumConnectionsA" (lpRasCon As Any, lpcb As _ Long, lpcConnections As Long) As Long Private Declare Function RasGetConnectStatus Lib "RasApi32.DLL" _ Alias "RasGetConnectStatusA" (ByVal hRasCon As Long, _ lpStatus As Any) As Long Private Declare Function RasEnumEntries Lib "RasApi32.DLL" _ Alias "RasEnumEntriesA" (ByVal reserved\$, ByVal _ lpzPhonebook\$, lpasentryname As Any, lpcb As Long, _ lpcEntries As Long) As Long Private Declare Function RasHangUp Lib "RasApi32.DLL" _ Alias "RasHangUpA" (ByVal hRasConn As Long) As Long Const RAS_MaxEntryName = 256 Const RAS_MaxDeviceType = 16 Const RAS_MaxDeviceName = 32 Private Type RASType dwSize As Long hRasCon As Long szEntryName(RAS_MaxEntryName) As Byte szDeviceType(RAS_MaxDeviceType) As Byte szDeviceName(RAS_MaxDeviceName) As Byte End Type Private Type RASStatusType dwSize As Long RasConnState As Long dwError As Long szDeviceType(RAS_MaxDeviceType) As Byte szDeviceName(RAS_MaxDeviceName) As Byte End Type Private Type RASENTRYNAME95 dwSize As Long szEntryName(RAS_MaxEntryName) As Byte End Type Dim DFÜName\$ Sub AufrufDNS() Dim sIP As String sIP = InputBox("IP-Adresse:", , "212.227.119.87") If sIP = "" Then Exit Sub Call WWWAddress(sIP) End Sub Sub AufrufIP() Dim sDNS As String sDNS = InputBox("DNS-Adresse:", , "www.herber.de") Call WWWAddress(sDNS) End Sub Sub WWWAddress(strAddress As String) Dim X% Dim IP\$, DNS\$, HOST\$ If Not Online Then Exit Sub Call InitSockets If IsNumeric(Left(strAddress, 1)) Then DNS = HostByAddress(strAddress) IP = strAddress Else IP = HostByName(strAddress) DNS = strAddress End If ActiveCell.Value = "DNS: " & DNS & " " & "IP: " & IP Call CleanSockets End Sub Private Sub InitSockets() Dim Result% Dim LoBy%, HiBy% Dim SocketData As WinSocketDataType Result = WSAStartup(WS_VERSION_REQD, SocketData) If Result <> 0 Then MsgBox ("'winsock.dll' antwortet nicht !") End End If End Sub Private Function HostByName(Name\$, Optional X% = 0) As String Dim MemIp() As Byte Dim Y% Dim HostDeAddress&, HostIp& Dim IPAddress\$ Dim HOST As HostDeType HostDeAddress = gethostbyname(Name) If HostDeAddress = 0 Then HostByName = "" Exit Function End If Call RtlMoveMemory(HOST, HostDeAddress, LenB(HOST)) For Y = 0 To X Call RtlMoveMemory(HostIp, HOST.hAddrList + 4 * Y, 4) If HostIp = 0 Then HostByName = "" Exit Function End If Next Y ReDim MemIp(1 To HOST.hLength) Call RtlMoveMemory(MemIp(1), HostIp, HOST.hLength) IPAddress = "" For Y = 1 To HOST.hLength IPAddress = IPAddress & MemIp(Y) & "." Next Y IPAddress = Left\$(IPAddress, Len(IPAddress) - 1) HostByName = IPAddress End Function Private Function DFÜStatus() As Boolean Dim RAS(255) As RASType, RASStatus As RASStatusType Dim lg&, lpcon&, Result& RAS(0).dwSize = 412 lg = 256 * RAS(0).dwSize Result = RasEnumConnections(RAS(0), lg, lpcon) If lpcon = 0 Then DFÜStatus = False Else RASStatus.dwSize = 160 Result = RasGetConnectStatus(RAS(0).hRasCon, RASStatus) If RASStatus.RasConnState = &H2000 Then DFÜStatus = True Else DFÜStatus = False End If End If End Function Private Function Online() As Boolean Dim Test As Boolean Test = DFÜStatus If Test = False Then MsgBox _ ("Keine Online Verbindung vorhanden ! Bitte einwählen !") Online = Test End Function Private Function HostByAddress(ByVal Adresse\$) As String Dim X% Dim HostDeAddress& Dim aa\$, BB As String * 5 Dim HOST As HostDeType aa = Chr\$(Val(NextChar(Adresse, "."))) aa = aa + Chr\$(Val(NextChar(Adresse, "."))) aa = aa + Chr\$(Val(NextChar(Adresse, "."))) aa = aa + Chr\$(Val(Adresse)) HostDeAddress = gethostbyaddr(aa, Len(aa), 2) If HostDeAddress = 0 Then HostByAddress = "" Exit Function End If Call RtlMoveMemory(HOST, HostDeAddress, LenB(HOST)) aa = "" X = 0 Do Call RtlMoveMemory(ByVal BB, HOST.hName + X, 1) If Left\$(BB, 1) = Chr\$(0) Then Exit Do aa = aa + Left\$(BB, 1) X = X + 1 Loop HostByAddress = aa End Function Private Function MyHostName() As String Dim HostName As String * 256 If gethostname(HostName, 256) = SOCKET_ERROR Then MsgBox "Windows Sockets error " & Str(WSAGetLastError()) Exit Function Else MyHostName = NextChar(Trim\$(HostName), Chr\$(0)) End If End Function Private Sub CleanSockets() Dim Result& Result = WSACleanup() If Result <> 0 Then MsgBox ("Socket Error " & Trim\$(Str\$(Result)) & _ " in Prozedur 'CleanSockets' aufgetreten !") End End If End Sub Private Function NextChar(Text\$, Char\$) As String Dim POS% POS = InStr(1, Text, Char) If POS = 0 Then NextChar = Text Text = "" Else NextChar = Left\$(Text, POS - 1) Text = Mid\$(Text, POS + Len(Char)) End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Überprüfung, ob eine RAS/Internet-Verbindung besteht

Problem:	Das Bestehen einer RAS/Internetverbindung soll geprüft werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Private Const NETWORK_ALIVE_LAN = &H1 'net card connection Private Const NETWORK_ALIVE_WAN = &H2 'RAS connection Private Const NETWORK_ALIVE_AOL = &H4 'AOL Private Declare Function IsNetworkAlive Lib "Sensapi" _ (lpdwFlags As Long) As Long Private Function IsNetConnectionAlive() As Boolean Dim lngAlive As Long IsNetConnectionAlive = IsNetworkAlive(lngAlive) = 1 End Function Private Function IsNetConnectionLAN() As Boolean Dim lngLAN As Long If IsNetworkAlive(lngLAN) = 1 Then IsNetConnectionLAN = lngLAN = NETWORK_ALIVE_LAN End If End Function Private Function IsNetConnectionRAS() As Boolean Dim lngRAS As Long If IsNetworkAlive(lngRAS) = 1 Then IsNetConnectionRAS = lngRAS = NETWORK_ALIVE_WAN End If End Function Private Function IsNetConnectionAOL() As Boolean Dim lngAOL As Long If IsNetworkAlive(tmp) = 1 Then IsNetConnectionAOL = lngAOL = NETWORK_ALIVE_AOL End If End Function Private Function GetNetConnectionType() As String Dim lngAlive As Long If IsNetworkAlive(lngAlive) = 1 Then Select Case lngAlive Case NETWORK_ALIVE_LAN: GetNetConnectionType = _ "Es bestehen eine oder mehr aktive LAN-Verbindungen." Case NETWORK_ALIVE_WAN: GetNetConnectionType = _ "Es bestehen eine oder mehr aktive RAS-Verbindungen." Case NETWORK_ALIVE_AOL: GetNetConnectionType = _ "Es besteht eine Verbindung über AOL." Case Else: End Select Else GetNetConnectionType = _ "Ich kann keine Verbindung feststellen." End If End Function Sub IsConnection() MsgBox GetNetConnectionType End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auslesen der aktuellen Mausposition

Problem:	Die aktuelle Mausposition soll ausgelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Private Declare Function GetCursorPos Lib "user32" (lpPoint As POINTAPI) As Long Private Type POINTAPI x As Long y As Long End Type Sub WhereAmI() Dim pTargetPoint As POINTAPI Dim lRetVal As Long lRetVal = GetCursorPos(pTargetPoint) MsgBox "Meine Position:" & vbCrLf & _ pTargetPoint.x & "," & pTargetPoint.y End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auslesen der Mausposition

Problem:	Wie kann ich die Mausposition nach Points auslesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function GetCursorPos& Lib "user32" (lpPoint As POINTAPI) Type POINTAPI x As Long y As Long End Type Sub Orientieren() Dim lPos As Long Dim PT As POINTAPI If Application.DisplayFullScreen = False Then Beep MsgBox "Zuerst Schaltfläche 'Start' drücken!" Exit Sub End If lPos = GetCursorPos(PT) If PT.x >= 3 And PT.x <= 30 And PT.y >= 198 And PT.y <= 240 Then MsgBox "Bad Honnef" ElseIf PT.x >= 68 And PT.x <= 116 And PT.y >= 328 And PT.y <= 337 Then MsgBox "Linz am Rhein" ElseIf PT.x >= 214 And PT.x <= 239 And PT.y >= 238 And PT.y <= 264 Then MsgBox "Neustadt/Wied" ElseIf PT.x >= 207 And PT.x <= 240 And PT.y >= 145 And PT.y <= 182 Then Beep MsgBox "Asbach" ElseIf PT.x >= 222 And PT.x <= 237 And PT.y >= 278 And PT.y <= 293 Then MsgBox "Autobahnabfahrt Neustadt/Wied" ElseIf PT.x >= 236 And PT.x <= 251 And PT.y >= 288 And PT.y <= 309 Then MsgBox "Autobahntankstelle Fernthal" ElseIf PT.x >= 112 And PT.x <= 127 And PT.y >= 194 And PT.y <= 209 Then MsgBox "Autobahnabfahrt Bad Honnef/Linz" ElseIf PT.x >= 52 And PT.x <= 67 And PT.y >= 131 And PT.y <= 146 Then MsgBox "Autobahnabfahrt Siebengebirge" ElseIf PT.x >= 182 And PT.x <= 201 And PT.y >= 126 And PT.y <= 144 Then MsgBox "Buchholz" ElseIf PT.x >= 141 And PT.x <= 159 And PT.y >= 201 And PT.y <= 217 Then MsgBox "Windhagen" ElseIf PT.x >= 42 And PT.x <= 70 And PT.y >= 90 And PT.y <= 111 Then MsgBox "Oberpleis" ElseIf PT.x >= 8 And PT.x <= 33 And PT.y >= 105 And PT.y <= 135 Then MsgBox "Heisterbacherrott" ElseIf PT.x >= 33 And PT.x <= 58 And PT.y >= 142 And PT.y <= 163 Then MsgBox "Ittenbach" ElseIf PT.x >= 121 And PT.x <= 143 And PT.y >= 250 And PT.y <= 274 Then MsgBox "Vettelschoß" ElseIf PT.x >= 130 And PT.x <= 149 And PT.y >= 294 And PT.y <= 310 Then MsgBox "St. Katharinen" Else MsgBox "Bitte einen Ort oder eine Autobahnabfahrt anklicken!" End If End Sub Sub Zurueck() With Application .DisplayFullScreen = Not .DisplayFullScreen End With If [Button].Caption = "Zurück" Then [Button].Caption = "Start" Else [Button].Caption = "Zurück" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Uhr in Menüleiste

Problem:	In der Excle-Menüleiste soll eine Uhr integriert und auf Wunsch wieder gelöscht werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub StartClock() StartClockinMenu End Sub Sub StopClock() StopClockinMenu End Sub StandardModule: Modul2 Private Declare Function FindWindow _ Lib "user32" _ Alias "FindWindowA" _ (_ ByVal lpClassName As String, _ ByVal lpWindowName As String _) _ As Long Private Declare Function SetTimer _ Lib "user32" _ (_ ByVal hWnd As Long, _ ByVal nIDEvent As Long, _ ByVal uElapse As Long, _ ByVal lpTimerFunc As Long _) _ As Long Private Declare Function KillTimer _ Lib "user32" _ (_ ByVal hWnd As Long, _ ByVal nIDEvent As Long _) _ As Long Private Declare Function GetCurrentVbaProject _ Lib "vba332.dll" _ Alias "EbGetExecutingProj" _ (_ hProject As Long _) _ As Long Private Declare Function GetFuncID _ Lib "vba332.dll" _ Alias "TipGetFunctionId" _ (_ ByVal hProject As Long, _ ByVal strFunctionName As String, _ ByRef strFunctionID As String _) _ As Long Private Declare Function GetAddr _ Lib "vba332.dll" _ Alias "TipGetLpfnOfFunctionId" _ (_ ByVal hProject As Long, _ ByVal strFunctionID As String, _ ByRef lpfnAddressOf As Long _) _ As Long Private WindowsTimer As Long Private ClockCBControl As CommandBarButton Sub StartClockinMenu() Set ClockCBControl = _ Application.CommandBars(1).Controls.Add(_ Type:=msoControlButton, Temporary:=True) ClockCBControl.Style = msoButtonCaption ClockCBControl.Caption = Format(Now, "Long Time") fncWindowsTimer 1000 End Sub Sub StopClockinMenu() fncStopWindowsTimer ClockCBControl.Delete End Sub Private Function fncWindowsTimer(_ TimeInterval As Long _) As Boolean Dim WindowsTimer As Long WindowsTimer = 0 If Val(Application.Version) > 8 Then WindowsTimer = SetTimer _ (_ hWnd:=FindWindow("XLMAIN", Application.Caption), _ nIDEvent:=0, _ uElapse:=TimeInterval, _ lpTimerFunc:=AddrOf_cbkCustomTimer _) Else WindowsTimer = SetTimer _ (_ hWnd:=FindWindow("XLMAIN", Application.Caption), _ nIDEvent:=0, _ uElapse:=TimeInterval, _ lpTimerFunc:=AddrOf("cbkCustomTimer") _) End If fncWindowsTimer = CBool(WindowsTimer) End Function Private Function fncStopWindowsTimer() KillTimer _ hWnd:=FindWindow("XLMAIN", Application.Caption), _ nIDEvent:=WindowsTimer End Function Private Function cbkCustomTimer _ (_ ByVal Window_hWnd As Long, _ ByVal WindowsMessage As Long, _ ByVal EventID As Long, _ ByVal SystemTime As Long _) _ As Long Dim CurrentTime As String On Error Resume Next ClockCBControl.Caption = Format(Now, "Long Time") End Function Private Function AddrOf _ (_ CallbackFunctionName As String _) _ As Long Dim aResult As Long Dim CurrentVBProject As Long Dim strFunctionID As String Dim AddressOfFunction As Long Dim UnicodeFunctionName As String UnicodeFunctionName = StrConv(CallbackFunctionName, vbUnicode) If Not GetCurrentVbaProject(CurrentVBProject) = 0 Then aResult = GetFuncID _ (_ hProject:=CurrentVBProject, _ strFunctionName:=UnicodeFunctionName, _ strFunctionID:=strFunctionID _) If aResult = 0 Then aResult = GetAddr _ (_ hProject:=CurrentVBProject, _ strFunctionID:=strFunctionID, _ lpfnAddressOf:=AddressOfFunction _) If aResult = 0 Then AddrOf = AddressOfFunction End If End If End If End Function Private Function AddrOf_cbkCustomTimer() As Long AddrOf_cbkCustomTimer = vbaPass(AddressOf cbkCustomTimer) End Function Private Function vbaPass(AddressOfFunction As Long) As Long vbaPass = AddressOfFunction End Function</pre>



Auswechseln der Excel- und Arbeitsmappen-Icons

Problem:	Auswechseln der Anwendungs- und Arbeitsmappen-Icons gegen eigene Kreationen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Code von Stratos Malasiotis ----- Win32 API Function Declarations ----- Declare Function FindWindow _ Lib "user32" _ Alias "FindWindowA" _ (ByVal lpClassName As String, _ ByVal lpWindowName As String _) As Long Declare Function FindWindowEx _ Lib "user32" _ Alias "FindWindowExA" _ (ByVal hWnd1 As Long, _ ByVal hWnd2 As Long, _ ByVal lpsz1 As String, _ ByVal lpsz2 As String _) As Long Declare Function ExtractIcon _ Lib "shell32.dll" _ Alias "ExtractIconA" _ (ByVal hInst As Long, _ ByVal lpszExeFileName As String, _ ByVal nIconIndex As Long _) As Long Declare Function SendMessage _ Lib "user32" _ Alias "SendMessageA" _ (ByVal hWnd As Long, _ ByVal wParam As Long, _ ByVal lparam As Long _) As Long ----- Win32 API Constant Declarations ----- Const WM_SETICON As Long = &H80 ----- Custom function for changing Excel's windows icons ----- Public Function fncSetXLWindowIcon _ (Optional IconFile As String = vbNullString, _ Optional WorkbookName As String = vbNullString _) As Boolean '----- 'Variable Declarations '----- Dim XLMAINhWnd As Long, XLDESKhWnd As Long, _ EXCEL7hWnd As Long, TargetWindowhWnd As Long, _ VirtualIcon As Long 'initialise the result of the function to false; assume failure fncSetXLWindowIcon = True '----- 'STEP 1. Identify the target window '----- 'get the caption from the first window of the specified workbook; if any On Error Resume Next If CBool(Len((Workbooks(WorkbookName).Name))) Then WorkbookName = Workbooks(WorkbookName).Windows(1).Caption End If On Error GoTo ExitFunction 'if a caption has been extracted get a hendle to that workbook window; 'else get a handle to Excel's main window If Not WorkbookName = vbNullString Then XLMAINhWnd = FindWindow("XLMAIN", Application.Caption) XLDESKhWnd = FindWindowEx(XLMAINhWnd, 0, "XLDESK", vbNullString) TargetWindowhWnd = FindWindowEx(XLDESKhWnd, 0, "EXCEL7", WorkbookName) Else XLMAINhWnd = FindWindow("XLMAIN", Application.Caption) TargetWindowhWnd = XLMAINhWnd End If 'if we couldn't get a handle, exit the function If TargetWindowhWnd = 0 Then Exit Function '----- 'STEP 2. Extract the icon from the respective file '----- If IconFile = vbNullString Then 'assume that the user asked to restore the original icon VirtualIcon = 0 Else 'try to extract the icon from the specified file VirtualIcon = ExtractIcon(0, IconFile, 0) 'If the file could not be found (1), or if the no icon could be 'found in the file (0), exit the function If VirtualIcon <= 1 Then Exit Function End If '----- 'STEP 3. Send a Windows message to the specified window to change ' its icon '----- 'in most cases only the second (False) message is adequate SendMessage TargetWindowhWnd, WM_SETICON, True, VirtualIcon SendMessage TargetWindowhWnd, WM_SETICON, False, VirtualIcon ' 'the functio has been completed succesfully fncSetXLWindowIcon = True ExitFunction: End Function Sub test1_fncSetXLWindowIcon() 'set XL's main window icon fncSetXLWindowIcon Range("A1").Value End Sub Sub test2_fncSetXLWindowIcon() 'restore XL's main window icon fncSetXLWindowIcon End Sub Sub test3_fncSetXLWindowIcon() 'set active workbook's window icon fncSetXLWindowIcon Range("A1").Value, _ ActiveWorkbook.Name End Sub Sub test4_fncSetXLWindowIcon() 'restore active workbook's window icon fncSetXLWindowIcon , ActiveWorkbook.Name End Sub ()



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Excel-Menü oben links am Fenster ausblenden

Problem: Wie kann ich das Excel-Menü oben links im Fenster ausblenden und im DATEI-Menü die BEENDEN-Schaltfläche deaktivieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Declare Function FindWindowA Lib "User32" _
    (ByVal lpClassName As Any, _
    ByVal lpWindowName As String) As Long
Declare Function GetWindowLongA Lib "User32" _
    (ByVal hwnd As Long, ByVal nIndex As Integer) As Long
Declare Function SetWindowLongA Lib "User32" _
    (ByVal hwnd As Long, ByVal nIndex As Integer, _
    ByVal dwNewLong As Long) As Long

Global Const GWL_STYLE = (-16)
Global Const WS_SYSMENU = &H80000

Sub RemoveControlMenuExcel32()
    Dim WindowStyle As Long
    Dim hwnd As Long
    Dim WindowName As String
    Dim result
    WindowName = Application.Caption
    hwnd = FindWindowA(0&, WindowName)
    WindowStyle = GetWindowLongA(hwnd, GWL_STYLE)
    WindowStyle = WindowStyle And (Not WS_SYSMENU)
    result = SetWindowLongA(hwnd, GWL_STYLE, WindowStyle)
End Sub

Sub RestoreControlMenuExcel32()
    Dim WindowStyle As Long
    Dim hwnd As Long
    Dim WindowName As String
    Dim result
    WindowName = Application.Caption
    hwnd = FindWindowA(0&, ByVal WindowName)
    WindowStyle = GetWindowLongA(hwnd, GWL_STYLE)
    WindowStyle = WindowStyle Or WS_SYSMENU
    result = SetWindowLongA(hwnd, GWL_STYLE, WindowStyle)
End Sub

Sub SetEnabledTrue()
    Application.CommandBars("Worksheet Menu Bar"). _
        Controls("Datei").Controls("Beenden").Enabled = True
    Call RestoreControlMenuExcel32
End Sub

Sub SetEnabledFalse()
    Application.CommandBars("Worksheet Menu Bar"). _
        Controls("Datei").Controls("Beenden").Enabled = False
    Call RemoveControlMenuExcel32
End Sub
```



Schriftart der Statusbar verändern

Problem:	Wie kann ich die Schriftart in der StatusBar verändern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Option Base 1 Declare Function FindWindow32 Lib "user32" Alias "FindWindowA" _ (ByVal szClass\$, ByVal szTitle\$) As Long Declare Function GetWindow32 Lib "user32" Alias "GetWindow" _ (ByVal hwnd As Long, ByVal wCmd As Long) As Long Declare Function GetClassName32 Lib "user32" Alias "GetClassNameA" _ (ByVal hwnd As Long, ByVal lpClassName As String, _ ByVal nmaxCount As Long) As Long Declare Function GetDC32 Lib "user32" Alias "GetDC" _ (ByVal hwnd As Long) As Long Declare Function CreateFont32 Lib "gdi32" Alias "CreateFontA" _ (ByVal H As Long, ByVal w As Long, ByVal E As Long, _ ByVal O As Long, ByVal w As Long, ByVal i As Long, _ ByVal u As Long, ByVal s As Long, ByVal c As Long, _ ByVal OP As Long, ByVal CP As Long, ByVal Q As Long, _ ByVal PAF As Long, ByVal f As String) As Long Declare Function SelectObject32 Lib "gdi32" Alias "SelectObject" _ (ByVal hdc As Long, ByVal hObject As Long) As Long Declare Function SetTextColor32 Lib "gdi32" Alias "SetTextColor" _ (ByVal hdc As Long, ByVal crColor As Long) As Long Declare Function DeleteObject32 Lib "gdi32" Alias "DeleteObject" _ (ByVal hObject As Long) As Long Declare Function ReleaseDC32 Lib "user32" Alias "ReleaseDC" _ (ByVal hwnd As Long, ByVal hdc As Long) As Long Sub MsgInBold() Dim hWndNext Dim hWndWks32 As Long Dim hDcWks32 As Long Dim hFont32 As Long, hFontOld32 As Long Dim sBuff As String * 255 Dim a As Long Const GW_CHILD As Integer = 5 Const GW_HWNDFIRST As Integer = 0 Const GW_HWNDNEXT As Integer = 2 Application.DisplayStatusBar = True 'Get the handle of the Excel window hWndWks32 = FindWindow32("XLMAIN", Application.Caption) 'Get a child window hWndNext = GetWindow32(hWndWks32, GW_CHILD) 'Get the first child window hWndNext = GetWindow32(hWndNext, GW_HWNDFIRST) a = GetClassName32(hWndNext, sBuff, 255) hDcWks32 = GetDC32(hWndNext) 'Create the font to use for the text 'CreateFont(Height, Width, Escapement, Orientation, Weight, _ ' Italic, Underline, Strikeout, CharSet, Precision, _ ' Clipping, Quality, Pitch & Family, Font Name) ' 'Times New Roman, 32pt, Bold: ' CreateFont32(-32, 0, 0, 0, 700, 0, 0, 0, 0, 0, 0, 0, 0, _ ' "Times NewRoman ") ' 'MS Sans Serif, 12pt, Normal, Italic, Underline & Strikeout: ' CreateFont32(-12, 0, 0, 0, 400, 1, 1, 1, 0, 0, 0, 0, 0, _ ' "MS SansSerif ") hFont32 = CreateFont32(-32, 0, 0, 0, 700, 0, 0, 0, 0, 0, 0, 0, 0, _ ' "Times New Roman") hFontOld32 = SelectObject32(hDcWks32, hFont32) 'Display the message in the correct font Application.StatusBar = "Please press the OK button" 'Display a message box MsgBox "How's that!" Application.StatusBar = False 'Restore the original font a = SelectObject32(hDcWks32, hFontOld32) 'Release the device context to the status bar a = ReleaseDC32(hWndNext, hDcWks32) End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit einer Datei-Suffix verknüpfte Anwendung auslesen

Problem:	Die mit einer Datei-Suffix verknüpfte Anwendung wird incl. Pfad ausgelesen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Private Declare Function FindExecutable Lib "shell32.dll" Alias "FindExecutableA" _ (ByVal lpFile As String, ByVal lpDirectory As String, ByVal lpResult As String) As Long Private Declare Function lstrlen Lib "kernel32.dll" Alias "lstrlenA" _ (ByVal lpString As Any) As Long FindExecutable Konstanen Private Const ERROR_FILE_NOT_FOUND = 2& Private Const ERROR_PATH_NOT_FOUND = 3& Private Const ERROR_BAD_FORMAT = 11& Sub GetApplication() Dim Retval As Long, Puffer As String Puffer = Space(256) Retval = FindExecutable(Range("A1").Value, "", Puffer) Select Case Retval Case 0 MsgBox "Es ist nicht genügend Speicher vorhanden, " & _ "um diese Funktion durchzuführen." Case 31 MsgBox "Für diese Datei existiert keine verknüpfte Anwendung!" Case ERROR_FILE_NOT_FOUND MsgBox "Die angegebene Datei wurde nicht gefunden." Case ERROR_PATH_NOT_FOUND MsgBox "Der angegebene Pfad wurde nicht gefunden" Case ERROR_BAD_FORMAT MsgBox "Die verknüpfte Anwendung ist ungültig oder keine Win32 Anwendung" Case Else MsgBox "Verknüpfte Anwendung: " & left\$(Puffer, lstrlen(Puffer)) End Select End Sub</pre>

Speicherinformationen ermitteln

Problem:	Die Speicherinformationen auszuwählender Laufwerke werden ermittelt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: Modul1 Private Declare Function GetDiskFreeSpace Lib "kernel32.dll" Alias _ "GetDiskFreeSpaceA" (ByVal lpRootPathName As String, _ lpSectorsPerCluster As Long, lpBytesPerSector As Long, _ lpNumberOfFreeClusters As Long, lpTotalNumberOfClusters As Long) As Long Private Declare Function GetDiskFreeSpaceEx Lib "kernel32.dll" Alias _ "GetDiskFreeSpaceExA" (ByVal lpDirectoryName As String, _ lpFreeBytesAvailableToCaller As ULARGE_INTEGER, _ lpTotalNumberOfBytes As ULARGE_INTEGER, _ lpTotalNumberOfFreeBytes As ULARGE_INTEGER) As Long Private Declare Function GetVersionEx Lib "kernel32.dll" Alias "GetVersionExA" _ (lpVersionInformation As OSVERSIONINFO) As Long Private Declare Function GetLogicalDriveStrings Lib "kernel32.dll" Alias _ "GetLogicalDriveStringsA" (ByVal nBufferLength As Long, _ ByVal lpBuffer As String) As Long Private Declare Sub CopyMemory Lib "kernel32.dll" Alias "RtlMoveMemory" _ (Destination As Any, Source As Any, ByVal Length As Long) Private Type ULARGE_INTEGER LowPart As Long HighPart As Long End Type Private Type OSVERSIONINFO dwOSVersionInfoSize As Long dwMajorVersion As Long dwMinorVersion As Long dwBuildNumber As Long dwPlatformId As Long szCSDVersion As String * 128 End Type Private Const VER_PLATFORM_WIN32s = 0 ' Win 3.1 mit 32 Bit-Erweiterung Private Const VER_PLATFORM_WIN32_WINDOWS = 1 ' Win NT/2000 Private Const VER_PLATFORM_WIN32_NT = 2 ' Win 9x Sub GetInfos() Dim Retval As Long, OS As OSVERSIONINFO, InputRet As String, Buffer As String Buffer = Space(256) Retval = GetLogicalDriveStrings(Len(Buffer), Buffer) Buffer = Left\$(Buffer, Retval) Buffer = Replace(Buffer, "\" & vbNullChar, ", ") InputRet = InputBox("Bitte geben sie einen Laufwerksbuchstaben ein (" & _ Buffer & ").", "Speicher Information ermitteln", "C:\") If InputRet = "" Or Len(InputRet) > 3 Then MsgBox "Ungültige eingabe, Bitte geben sie nur Laufwerksbuchstaben, Doppelpunkt und Backslash ein" Exit Sub End If OS.dwOSVersionInfoSize = Len(OS) Retval = GetVersionEx(OS) If Retval = 0 Then MsgBox "Betriebssystem Version konnte nicht ermittelt werden." Exit Sub End If With OS Select Case .dwPlatformId Case VER_PLATFORM_WIN32_WINDOWS If InStr(1, OS.szCSDVersion, "B") <> 0 Or .dwMinorVersion > 0 Then Call GetNewFreespace(InputRet) Else Call GetOldFreespace(InputRet) End If Case VER_PLATFORM_WIN32_NT If .dwMajorVersion >= 4 Then Call GetNewFreespace(InputRet) Else Call GetOldFreespace(InputRet) End If Case Else MsgBox "Windows 3.x Version kann nicht ermittelt werden." End Select End With End Sub Private Function GetOldFreespace(ByVal Root As String) Dim Retval As Long Dim SC As Long, BC As Long, FC As Long, TC As Long Dim TSpace As Long, FSpace As Long, USpace As Long Retval = GetDiskFreeSpace(Root, SC, BC, FC, TC) TSpace = TC * SC * BC / 1024 / 1024 FSpace = FC * SC * BC / 1024 / 1024 USpace = TSpace - FSpace MsgBox "Gesamt: " & Format\$(TSpace, "##.00 MB") & vbCrLf & _ "Belegt: " & Format\$(USpace, "##.00 MB") & vbCrLf & _ "Frei: " & Format\$(FSpace, "##.00 MB") & vbCrLf & _ , , "Festplattenspeicher Laufwerk "" & Root & """ End Function Private Function GetNewFreespace(ByVal Root As String) Dim Retval As Long Dim CBytes As ULARGE_INTEGER, TBytes As ULARGE_INTEGER, FBytes As ULARGE_INTEGER Dim CB As Currency, TB As Currency, FB As Currency, UB As Currency Retval = GetDiskFreeSpaceEx(Root, CBytes, TBytes, FBytes) CopyMemory CB, CBytes, 8 CB = CB * 10000 / 1024 / 1024 CopyMemory TB, TBytes, 8 TB = TB * 10000 / 1024 / 1024 CopyMemory FB, FBytes, 8 FB = FB * 10000 / 1024 / 1024 UB = TB - FB MsgBox "Gesamt: " & Format\$(TB, "##.00 MB") & vbCrLf & _ "Belegt: " & Format\$(UB, "##.00 MB") & vbCrLf & _ "Frei: " & Format\$(FB, "##.00 MB") & vbCrLf & _ , , "Festplattenspeicher Laufwerk "" & Root & """ End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anwenderverzeichnis auslesen

Problem:	Das Anwenderverzeichnis im Ordner Dokumente und Einstellungen soll ausgelesen werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Private Const S_OK = &H0 Private Const S_FALSE = &H1 Private Const E_INVALIDARG = &H80070057 Private Const CSIDL_FLAG_CREATE = &H8000& Private Const CSIDL_PERSONAL = &H5& Private Const SHGFP_TYPE_CURRENT = 0 Private Const SHGFP_TYPE_DEFAULT = 1 Private Const MAX_PATH = 260 Private Declare Function SHGetFolderPath Lib "shfolder" _ Alias "SHGetFolderPathA" _ (ByVal hwndOwner As Long, ByVal nFolder As Long, _ ByVal hToken As Long, ByVal dwFlags As Long, _ ByVal pszPath As String) As Long Function GetUserDirectory() Dim RetVal As Long Dim iChar As Integer Dim sPath As String sPath = "" sPath = String(MAX_PATH, 0) RetVal = SHGetFolderPath(0, CSIDL_PERSONAL Or CSIDL_FLAG_CREATE, _ 0, SHGFP_TYPE_CURRENT, sPath) Select Case RetVal Case S_OK sPath = Left(sPath, InStr(1, sPath, Chr(0)) - 1) Case S_FALSE, E_INVALIDARG sPath = "Nicht gefunden!" End Select For iChar = Len(sPath) To 1 Step -1 If Mid(sPath, iChar, 1) = "\" Then Exit For iChar = iChar - 1 Next iChar sPath = Left(sPath, iChar - 1) GetUserDirectory = sPath End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Windows-Wallpaper festlegen

Problem: Das Windows-Desktop-Hintergrundbild soll festgelegt werden. Anzugeben ist der komplette Pfad- und Dateiname der *.bmp-Datei.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code:

```
StandardModule: Modul1

Public Declare Function _
    SystemParametersInfo Lib "user32" Alias _
    "SystemParametersInfoA" ( _
        ByVal uAction As Long, _
        ByVal uParam As Long, _
        ByVal lpvParam As String, _
        ByVal fuWinIni As Long) As Long

Sub ChangeWallpaper()
    Dim lng As Long
    Dim strWP As String
    strWP = InputBox("Pfad der Hintergrunddatei:")
    If strWP = "" Then Exit Sub
    lng = SystemParametersInfo(20, 0, strWP, 1)
    If lng = 0 Then
        MsgBox "Hintergrunddatei konnte nicht gesetzt werden!"
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

CD-Laufwerk öffnen und schließen

Problem:	Wie kann ich über VBA das CD-Laufwerk öffnen und schließen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Private Declare Function mciExecute Lib "winmm.dll" _ (ByVal lpstrCommand As String) As Long Sub CD_oeffnen() Call mciExecute("Set CDAudio door open") End Sub Sub CD_schliessen() Call mciExecute("Set CDAudio door closed") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Erstes CD-ROM-Laufwerk ermitteln

Problem:	Wie kann ich das erste CD-ROM-Laufwerk ermitteln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Declare Function GetDriveType Lib "kernel32" Alias "GetDriveTypeA" _ (ByVal nDrive As String) As Long Declare Function GetLogicalDriveStrings Lib "kernel32" Alias _ "GetLogicalDriveStringsA" (ByVal nBufferLength As Long, _ ByVal lpBuffer As String) As Long Public Const DRIVE_CDROM As Long = 5 Function GetFirstCdRomDriveLetter() As String Dim lDriveType As Long Dim strDrive As String Dim lStart As Long: lStart = 1 Dim strDrives As String strDrives = Space(150) Dim lRetVal As Long lRetVal = GetLogicalDriveStrings(150, strDrives) If lRetVal = 0 Then GetFirstCdRomDriveLetter = vbNullString Exit Function End If strDrive = Mid(strDrives, lStart, 3) Do lDriveType = GetDriveType(strDrive) If lDriveType = DRIVE_CDROM Then GetFirstCdRomDriveLetter = strDrive Exit Function End If lStart = lStart + 4 strDrive = Mid(strDrives, lStart, 3) Loop While (Mid(strDrives, lStart, 1) <> vbNullChar) End Function Sub Main() Dim sDriveLetter As String sDriveLetter = GetFirstCdRomDriveLetter() MsgBox sDriveLetter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Plattenspeicher auslesen

Problem:	Wie kann ich den Plattenspeicher auslesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function GetDiskFreeSpace Lib "kernel32" _ Alias "GetDiskFreeSpaceA" (ByVal lpRootPathName As String, _ lpSectorsPerCluster As Long, lpBytesPerSector As Long, _ lpNumberOfFreeClusters As Long, lpTtoalNumberOfClusters As Long) _ As Long Sub Aufruf() Dim sLW As String sLW = InputBox("Laufwerk:", , "c") If sLW = "" Then Exit Sub MsgBox Plattenspeicher(sLW) End Sub Function Plattenspeicher(Laufwerksbuchstabe As String) As Long Dim lRet As Long Dim SectorsPerCluster As Long Dim BytesPerSector As Long Dim NumberofFreeClusters As Long Dim TotalClusters As Long Dim DLetter As String DLetter = Left(Laufwerksbuchstabe, 1) & ":\\" lRet = GetDiskFreeSpace(DLetter, SectorsPerCluster, BytesPerSector, _ NumberofFreeClusters, TotalClusters) If lRet = 0 Then 'Error occurred Plattenspeicher = -99 Exit Function End If Plattenspeicher = (SectorsPerCluster * BytesPerSector * TotalClusters) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bildschirmauflösung auslesen

Problem:	Wie kann ich die aktuelle Bildschirmauflösung ermitteln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function GetDeviceCaps Lib "gdi32" (ByVal hdc As Long, _ ByVal nIndex As Long) As Long Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long Declare Function ReleaseDC Lib "user32" (ByVal hwnd As Long, _ ByVal hdc As Long) As Long Const HORZRES = 8 Const VERTRES = 10 Function ScreenResolution() Dim lRval As Long Dim lDc As Long Dim lHSize As Long Dim lVSize As Long lDc = GetDC(0&) lHSize = GetDeviceCaps(lDc, HORZRES) lVSize = GetDeviceCaps(lDc, VERTRES) lRval = ReleaseDC(0, lDc) ScreenResolution = lHSize & "x" & lVSize End Function Sub GetScreenSize() MsgBox ScreenResolution() End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Computer ausschalten

Problem:	Wie kann ich den Computer auf Schaltflächendruck ausschalten?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function ExitWindowsEx& Lib "user32" (ByVal uFlags&, ByVal wReserved&) Global Const EWX_FORCE = 8 Global Const EWX_LOGOFF = 0 Global Const EWX_REBOOT = 2 Global Const EWX_SHUTDOWN = 1 Sub Tschuess() Dim LResult LResult = ExitWindowsEx(EWX_SHUTDOWN, 0&) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Umgebungsvariable aus der autoexec.bat auslesen

Problem:	Wie kann ich aus der Autoexec.bat eine Umgebungsvariable auslesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function GetEnvironmentVariableA Lib "Kernel32" _ (ByVal lpName As String, ByVal lpBuffer As String, _ ByVal nsize As Long) As Long Declare Function GetDoseEnvironment Lib "Kernel" () As Long Declare Function lstrcpy Lib "Kernel" Alias "lStrCpy" _ (ByVal lpString1 As Any, ByVal lpString2 As Any) As Long Sub Test() MsgBox GetDoseEnvironmentVariable("temp") End Sub Function GetDoseEnvironmentVariable(sEnvVar As String) As String Dim lpszOrigEnv As Long Dim lpszEnv As Long Dim szDoseEnv As String If b32Bit() Then szDoseEnv = Space(4096) lpszEnv = GetEnvironmentVariableA(sEnvVar, szDoseEnv, 4096) GetDoseEnvironmentVariable = Left(szDoseEnv, lpszEnv) Else lpszOrigEnv = GetDoseEnvironment() szDoseEnv = Space\$(4096) lpszEnv = lstrcpy(szDoseEnv, lpszOrigEnv) szDoseEnv = Trim(szDoseEnv) szDoseEnv = Left(szDoseEnv, Len(szDoseEnv) - 1) Do While szDoseEnv <> "" If Left(szDoseEnv, Len(sEnvVar)) = sEnvVar Then GetDoseEnvironmentVariable = Mid(szDoseEnv, _ Len(sEnvVar) + 2, Len(szDoseEnv)) Exit Function End If lpszOrigEnv = lpszOrigEnv + Len(szDoseEnv) + 1 szDoseEnv = Space\$(4096) lpszEnv = lstrcpy(szDoseEnv, lpszOrigEnv) szDoseEnv = Trim(szDoseEnv) szDoseEnv = Left(szDoseEnv, Len(szDoseEnv) - 1) Loop End If End Function Function b32Bit() As Boolean Select Case Left(Application.Version, 1) Case 5 b32Bit = False Case 7, 8, 9 b32Bit = True End Select End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Computernamen auslesen

Problem:	Wie kann ich den Computernamen ermitteln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function GetComputerName& Lib "kernel32" Alias _ "GetComputerNameA" (ByVal lbuffer As String, nsize As Long) Sub CptName() Dim sTxt As String * 64 Call GetComputerName(sTxt, 64) MsgBox sTxt End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Festplatten-Seriennummer auslesen

Problem:	Wie kann ich die Seriennummer der Festplatte auslesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function GetVolumeInformationA Lib "kernel32" _ (ByVal lpRootPathName As String, _ ByVal lpVolumeNameBuffer As String, _ ByVal nVolumeNameSize As Long, _ lpVolumeSerialNumber As Long, _ lpMaximumComponentLength As Long, _ lpFileSystemFlags As Long, _ ByVal lpFileSystemNameBuffer As String, _ ByVal nFileSystemNameSize As Long) As Long Sub SerienNummer() Dim SerialNumber As Long GetVolumeInformationA "C:\", vbNullString, 0, SerialNumber, _ 0, 0, vbNullString, 0 MsgBox SerialNumber End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eintrag aus der Win.ini auslesen

Problem:	Wie kann ich den Wert "Strasse" aus der Abteilung "Data2" der Win.ini auslesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function GetProfileString Lib "kernel32" Alias _ "GetProfileStringA" (ByVal lpApplicationName As String, _ ByVal lpKeyName As String, ByVal lpDefault As String, _ ByVal lpReturnedString As String, ByVal nSize As Long) As Long Sub Test() MsgBox GetString("Desktop", "Wallpaper") End Sub Private Function GetString(_ Section As String, Key As String) As String Dim lChr As Long Dim sKey As String sKey = String(256, 0) lChr = GetProfileString(Section, Key, "", sKey, 256) If lChr > 1 Then sKey = Left(sKey, lChr) End If GetString = sKey End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserName aus der Windows-Registry auslesen

Problem:	Wie kann ich den in der Windows-Registry eingetragenen User-Namen auslesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function GetUserName Lib "advapi32.dll" _ Alias "GetUserNameA" (ByVal lpBuffer As String, _ nSize As Long) As Long Sub ShowUserName() Dim Buffer As String * 100 Dim BuffLen As Long BuffLen = 100 GetUserName Buffer, BuffLen MsgBox Left(Buffer, BuffLen - 1) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus- und Einlesen der NumLock-, CapsLock und ScrollLock-Tasten

Problem:	Wie kann ich den Status der NumLock-, CapsLock und ScrollLock-Tasten über UserForm-CheckBoxes auslesen und setzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: frmLock Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim var As Variant Dim intNum As Integer, intCap As Integer, intScroll As Integer If chkNum.Value = True Then intNum = 1 Else intNum = 0 If chkCap.Value = True Then intCap = 1 Else intCap = 0 If chkScroll.Value = True Then intScroll = 1 Else intScroll = 0 var = SetKeys(intNum, intCap, intScroll) Unload Me End Sub Private Sub UserForm_Click() End Sub Private Sub UserForm_Initialize() Dim intNum As Integer, intCap As Integer, intScroll As Integer intNum = GetKeyState(VK_NUMLOCK) intCap = GetKeyState(VK_CAPITAL) intScroll = GetKeyState(VK_SCROLL) If intNum = 1 Then chkNum.Value = True Else chkNum = False If intCap = 1 Then chkCap.Value = True Else chkCap = False If intScroll = 1 Then chkScroll.Value = True Else chkScroll = False End Sub StandardModule: basMain Public Const VK_NUMLOCK = &H90 Public Const VK_CAPITAL = &H14 Public Const VK_SCROLL = &H91 Private Type KeyboardBytes kbByte(0 To 255) As Byte End Type Public kbArray As KeyboardBytes Public Declare Function GetKeyState Lib "user32" _ (ByVal nVirtKey As Long) As Integer Public Declare Function SetKeyboardState Lib "user32" _ (kbArray As KeyboardBytes) As Long Sub CallSetKeys() frmLock.Show End Sub Function SetKeys(_ intNumLockKey As Integer, _ intCapLockKey As Integer, _ intScrollLockKey As Integer) With kbArray .kbByte(VK_NUMLOCK) = intNumLockKey .kbByte(VK_CAPITAL) = intCapLockKey .kbByte(VK_SCROLL) = intScrollLockKey End With SetKeyboardState kbArray End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wav-Datei abspielen und mit ESC abbrechen

Problem:	Wie kann ich eine WAV-Datei 10 mal hintereinander abspielen lassen und die Schleife mit der ESC-Taste unterbrechen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function sndPlaySound32 Lib "winmm.dll" _ Alias "sndPlaySoundA" (ByVal lpszSoundName _ As String, ByVal uFlags As Long) As Long Sub SoundStart() Dim iCounter As Integer Application.EnableCancelKey = xlErrorHandler On Error GoTo ERRORHANDLER For iCounter = 1 To 10 Call sndPlaySound32(Range("A1").Value, 1) Application.Wait Now + TimeSerial(0, 0, 2) Next iCounter ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

WAV-Datei abspielen

Problem:	Wie kann ich über einen aufgefangenen Fehler eine WAV-Datei abspielen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function sndPlaySound32 Lib "winmm.dll" _ Alias "sndPlaySoundA" (ByVal lpszSoundName _ As String, ByVal uFlags As Long) As Long Sub LaufwerkARufen() On Error Resume Next ChDir "irgendwas" If Err = 76 Then GoTo ERRORHANDLER End ERRORHANDLER: Call sndPlaySound32(Range("A1").Value, 1) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Userform in der Entwicklungsumgebung anzeigen

Problem:	Wie kann ich mit dem Start einer UserForm in die Entwicklungsumgebung wechseln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub DialogAufruf() frmVBIDE.Show End Sub ClassModule: frmVBIDE Private Declare Function FindWindowA Lib "user32" _ (ByVal lpClassName As String, _ ByVal lpWindowName As String) As Long Private Declare Function SetParent Lib "user32" _ (ByVal hWndChild As Long, _ ByVal hWndNewParent As Long) As Long Private Sub CommandButton1_Click() Unload Me End Sub Sub UserForm_Initialize() Dim lReturn As Long Dim lHwnd As Long Dim lVBEHwnd As Long lVBEHwnd = Application.VBE.MainWindow.hwnd lHwnd = FindWindowA("ThunderXFrame", vbNullString) lReturn = SetParent(lHwnd, lVBEHwnd) Application.VBE.MainWindow.SetFocus End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Laufzeit von Windows seit letzten Start ermitteln

Problem:	Wie kann ich ermitteln, wie lange Windows seit dem letzten Start läuft?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Function GetTickCount Lib "kernel32" () As Long Sub WindowsTime() MsgBox Format(GetTickCount / 24 / 60 / 60 / 1000, _ "d" Tage, " hh:mm:ss") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bild für einen Zeitraum von weniger als 1 Sekunde zeigen

Problem:	Wie kann ich in einem Tabellenblatt ein Bild für einen Zeitraum < 1 Sekunde anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long) Sub Warten() Dim pct As Picture Dim sPct As String sPct = ThisWorkbook.Path & "\tileexc.gif" If Dir(sPct) = "" Then Beep MsgBox "Testdatei wurde nicht gefunden!" Exit Sub End If Set pct = ActiveSheet.Pictures.Insert(sPct) Sleep 200 pct.Delete End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufhebung der Workbook_Open-Befehle verhindern

Problem: In einem Workbook_Open-Befehl frage ich ein Paßwort ab. Wie kann ich verhindern, daß jemand durch gehaltene Umschalttaste die Ausführung des Workbook_Open-Befehls verhindert und somit die Arbeitsblätter sichtbar werden?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Dim sWort As String
    sWort = InputBox( _
        prompt:="Passwort:", _
        Title:="Sicherheitsabfrage", _
        Default:="Passwort")
    If sWort <> "Passwort" Then
        ThisWorkbook.Close savechanges:=False
    Else
        ThisWorkbook.IsAddin = False
    End If
End Sub

Sub Schliessen()
    ThisWorkbook.IsAddin = True
    ThisWorkbook.Close savechanges:=True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anwendung auf Festplatte suchen

Problem:	Eine in Zelle B1 stehende Anwendung soll auf dem aktiven Laufwerk gesucht werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Public Const PROCESS_QUERY_INFORMATION = &H400 Public Const WAIT_TIMEOUT = &H102& Declare Function OpenProcess Lib "kernel32" (ByVal dwDesiredAccess As Long, _ ByVal bInheritHandle As Long, ByVal dwProcessId As Long) As Long Declare Function WaitForSingleObject Lib "kernel32" (ByVal hHandle As Long, _ ByVal dwMilliseconds As Long) As Long Sub DOSShell() Application.ScreenUpdating = False On Error Resume Next ChDrive "c" ChDir "c:\" On Error GoTo 0 ShellAndWait ("command.com /c dir/s/b " & _ Range("B1").Value & " >" & _ Application.DefaultFilePath & "\dirliste.txt") Workbooks.Open Application.DefaultFilePath & "\dirliste.txt" If IsEmpty(Range("A1")) Then Beep MsgBox "AcrobatReader wurde nicht gefunden!" Else MsgBox "Pfad: " & Range("A1").Value End If ActiveWorkbook.Close savechanges:=False End Sub Sub ShellAndWait(strEXE As String) Dim ProcessID As Long Dim hProcess As Long Dim RetVal As Long ProcessID = Shell(strEXE, vbHide) hProcess = OpenProcess(PROCESS_QUERY_INFORMATION, False, ProcessID) Do DoEvents RetVal = WaitForSingleObject(hProcess, 50) Loop Until RetVal <> WAIT_TIMEOUT End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neue Excel-Sitzung mit Arbeitsmappe starten

Problem:	Es soll eine neue Excel-Instanz mit einer neuen Arbeitsmappe mit einem Tabellenblatt geöffnet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NeueSitzung() Dim appXL As Excel.Application Dim wkb As Workbook Set appXL = New Excel.Application Set wkb = appXL.Workbooks.Add(1) appXL.Visible = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappen öffnen und Bereiche kopieren

Problem: Alle Arbeitsmappen in dem in Zelle B1 genannten Verzeichnis sollen geöffnet und der mit Zelle A1 zusammenhängende Bereich des 3. Tabellenblattes in dieses Blatt kopiert werden.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Zusammenfassen()
    Dim wkb As Workbook
    Dim wks As Worksheet
    Dim arr As Variant
    Dim lRow As Long
    Dim iCounter As Integer
    Dim sFile As String
    Application.ScreenUpdating = False
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    Set wks = ActiveSheet
    sFile = Range("B1").Value
    arr = FileArray(sFile, "*.xls")
    For iCounter = 1 To UBound(arr)
        Set wkb = Workbooks.Open(sFile & "\" & _
            arr(iCounter))
        lRow = wks.Cells(Rows.Count, 1).End(xlUp).Row + 2
        wks.Cells(lRow, 1).Value = wkb.Name
        lRow = lRow + 2
        wkb.Worksheets(3).Range("A1").CurrentRegion.Copy wks.Cells(lRow, 1)
        wkb.Close savechanges:=False
    Next iCounter
    Worksheets(1).Select
ERRORHANDLER:
    Application.EnableEvents = True
    Application.ScreenUpdating = True
End Sub

Function FileArray(ByVal strPath As String, _
    sPattern As String)
    Dim arr()
    Dim iCounter As Integer
    Dim sFile As String
    If Right(strPath, 1) <> "\" Then strPath = strPath & "\"
    sFile = Dir(strPath & sPattern)
    If sFile = "" Then
        Beep
        MsgBox "Keine Dateien gefunden!"
    End If
    Do While sFile <> ""
        iCounter = iCounter + 1
        ReDim Preserve arr(1 To iCounter)
        arr(iCounter) = sFile
        sFile = Dir()
    Loop
    FileArray = arr
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aller A1-Zellen aus allen Blättern Tabelle1

Problem:	In Spalte A sollen die Werte aus A1 aus allen Tabellenblättern "Tabelle1" aller Arbeitsmappen aus dem in Zelle B1 genannten Verzeichnis eingelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Einlesen() Dim iCounter As Integer, iRow As Integer Dim sfile As String, sPath As String iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 With Application.FileSearch .LookIn = Range("B1").Value .FileType = msoFileTypeExcelWorkbooks .Execute For iCounter = 1 To .FoundFiles.Count sfile = Dir(.FoundFiles(iCounter)) sPath = WorksheetFunction.Substitute(.FoundFiles(iCounter), sfile, "") With Cells(iRow, 1) .Formula = "=" & sPath & "[" & sfile & "]"Tabelle1'!A1" .Value = .Value End With iRow = iRow + 1 Next iCounter End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe ohne Neuberechnung und Linkaktualisierung öffnen

Problem: Die in Zelle B1 genannte Arbeitsmappe soll ohne Neuberechnung und ohne Aktualisierung evtl. vorhandener Links geöffnet werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Oeffnen()
    Dim vStatus As Variant
    vStatus = Application.Calculation
    Application.Calculation = xlCalculationManual
    Workbooks.Open _
        Filename:=Range("B1").Value, _
        updatelinks:=False
    Application.Calculation = vStatus
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Dateien löschen, die älter als 30 Tage sind

Problem: Aus dem in Zelle B1 genannten Verzeichnis sollen alle Excel-Dateien gelöscht werden, deren letztes Bearbeitungsdatum mehr als 30 Tage zurückliegt.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub DateiLöschen()
    Dim sPath As String
    Dim iCount As Integer, iCounter As Integer
    sPath = Range("B1").Value
    If MsgBox("Sollen die Dateien tatsächlich gelöscht werden?", _
        vbCritical + vbYesNo) = vbNo Then Exit Sub
    iCount = 30
    With Application.FileSearch
        .LookIn = sPath
        .Filename = "*.xls"
        .Execute
        For iCounter = 1 To .FoundFiles.Count
            If FileDateTime(.FoundFiles(iCounter)) + 30 < Date Then
                Kill .FoundFiles(iCounter)
            End If
        Next iCounter
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Erste Blätter aus allen Dateien eines Verzeichnisses sammeln

Problem:	Alle Arbeitsmappen eines vorgegebenen Verzeichnisses öffnen und das dort einzig vorhandene Tabellenblatt in diese Arbeitsmappe kopieren.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Zusammenfassen() Dim wkb As Workbook Dim arr As Variant Dim iCounter As Integer Dim sFile As String Application.ScreenUpdating = False Application.EnableEvents = False On Error GoTo ERRORHANDLER sFile = Range("B1").Value arr = FileArray(sFile, "*.xls") For iCounter = 1 To UBound(arr) Set wkb = Workbooks.Open(sFile & "\" & _ arr(iCounter)) wkb.Worksheets.Add wkb.Worksheets(2).Move _ after:=ThisWorkbook.Worksheets(_ ThisWorkbook.Worksheets.Count) wkb.Close savechanges:=False Next iCounter Worksheets(1).Select ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = True End Sub Function FileArray(ByVal strPath As String, _ sPattern As String) Dim arr() Dim iCounter As Integer Dim sFile As String If Right(strPath, 1) <> "\" Then strPath = strPath & "\" sFile = Dir(strPath & sPattern) If sFile = "" Then Beep MsgBox "Keine Dateien gefunden!" End End If Do While sFile <> "" iCounter = iCounter + 1 ReDim Preserve arr(1 To iCounter) arr(iCounter) = sFile sFile = Dir() Loop FileArray = arr End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktive Arbeitsmappe löschen

Problem:	Wie kann ich die aktive Arbeitsmappe von der Festplatte löschen und schließen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub KillMe() Dim strFile As String ActiveWorkbook.ChangeFileAccess xlReadOnly Kill ActiveWorkbook.FullName ThisWorkbook.Close False End Sub</pre>



Daten aus geschlossenen Arbeitsmappen listen

Problem:	Aus dem Verzeichnis c:\temp soll aus allen Excel-Tabellen der erste und letzte Wert in Spalte F ausgelesen werden, ohne dass diese Arbeitsmappen geöffnet werden. Die Werte sind durchgehend eingetragen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Sub DateienAuslesen() Dim rng As Range Dim arr As Variant Dim iCounter As Integer, iRow As Integer, iAct As Integer Dim sPath As String, sFormula As String, sTmp As String Application.ScreenUpdating = False sPath = Range("B1").Value arr = FileArray(sPath, "*.xls") For iCounter = 1 To UBound(arr) If FileDateTime(sPath & arr(iCounter)) <= Date + 1 Then With Worksheets("Import") If IsEmpty(.Cells(1, 1)) Then Set rng = .Range("A1") Else Set rng = .Cells(Rows.Count, 1).End(xlUp).Offset(1, 0) End If End With rng.Value = sPath & arr(iCounter) sFormula = "=" sFormula = sFormula & sPath & "[" sFormula = sFormula & arr(iCounter) & "]" sFormula = sFormula & "Tabelle1'!" sTmp = Right(sFormula, Len(sFormula) - 1) sFormula = sFormula & "F1" rng.Offset(0, 1).Formula = sFormula rng.Offset(0, 2).Formula = "=counta(" & sTmp & "F:F)" iRow = rng.Offset(0, 2).Value If iRow > 0 Then rng.Offset(0, 2).Formula = "=" & sTmp & "F" & iRow Else rng.Offset(0, 2).ClearContents End If End If Next iCounter With Worksheets("Import").Range("A1").CurrentRegion .Value = .Value End With End Sub Function FileArray(sPath As String, sPattern As String) Dim arr() Dim iCounter As Integer Dim sFile As String If Right(sPath, 1) <> "\" Then sPath = sPath & "\" sFile = Dir(sPath & sPattern) Do While sFile <> "" iCounter = iCounter + 1 ReDim Preserve arr(1 To iCounter) arr(iCounter) = sFile sFile = Dir() Loop FileArray = arr End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die Eigenschaft "Stichwort" in allen Dateien eines Ordners ändern

Problem:	Wie kann ich in einem vorgegebenen Verzeichnis bei allen ExcelArbeitsmappen die Stichwort-Eigenschaft festlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SetProperties() Dim iCounter As Integer Application.ScreenUpdating = False Application.EnableEvents = False On Error GoTo ERRORHANDLER With Application.FileSearch .LookIn = Range("B1").Value .FileType = msoFileTypeExcelWorkbooks .FileName = "*.xls" .Execute For iCounter = 1 To .FoundFiles.Count Workbooks.Open _ FileName:=.FoundFiles(iCounter), _ updatelinks:=False ActiveWorkbook.BuiltinDocumentProperties(4) = _ "Excel, VBA" ActiveWorkbook.Close Savechanges:=True Next iCounter End With ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe nach Ausfüllen der Eingabezellen auf ReadOnly setzen

Problem:	Wie kann ich eine Arbeitsmappe, nachdem 3 Eingabezellen ausgefüllt wurden, unverzüglich auf "ReadOnly" setzen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Intersect(Target, Range("B3:B5")) Is Nothing Then Exit Sub If IsEmpty(Target) Then Exit Sub Application.DisplayAlerts = False If WorksheetFunction.CountA(Range("B3:B5")) = 3 Then ActiveWorkbook.SaveAs _ FileName:=Range("B1").Value, _ writeresPassword:="HWH" ActiveWorkbook.Close End If Application.DisplayAlerts = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Passwortgeschützte Mappe öffnen, Daten kopieren, schliessen

Problem: Wie kann ich eine passwortgeschützte Arbeitsmappe öffnen, Werte in diese Arbeitsmappe einlesen und die Mappe wieder schießen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub DatenImport()
    Dim rng As Range
    Dim sFile As String
    Application.ScreenUpdating = False
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    sFile = Range("B1").Value
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Testdatei wurde nicht gefunden!"
        GoTo ERRORHANDLER
    End If
    Set rng = Worksheets("Data").Range("F1:I10")
    Workbooks.Open _
        FileName:=sFile, _
        password:="Password"
    rng.Value = Range("A1:D10").Value
    ActiveWorkbook.Close savechanges:=False
    Worksheets("Data").Select
ERRORHANDLER:
    Application.EnableEvents = True
    Application.ScreenUpdating = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Automatisch Sicherungsdatei erstellen

Problem:	Wie kann ich beim Speichern jeder Arbeitsmappe automatisch eine Sicherungsdatei erstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SaveAndBackup() Dim sName As String, sMsg As String On Error GoTo ERRORHANDLER sMsg = "Bitte Pfad und Dateinamen angeben:" sName = ActiveWorkbook.FullName sName = InputBox(_ prompt:=sMsg, Default:=sName) If sName = "" Then Exit Sub ActiveWorkbook.SaveAs FileName:=sName, FileFormat:= _ xlNormal, Password:="", WriteResPassword:="", _ ReadOnlyRecommended:=False, CreateBackup:=True Exit Sub ERRORHANDLER: MsgBox "Datei konnte nicht gespeichert werden!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Dateien im Verzeichnis öffnen

Problem:	Wie kann ich alle Excel-Dateien eines Verzeichnisses über VBA öffnen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub OpenWkb() Dim sFile As String, sPath As String Application.ScreenUpdating = False sPath = Range("B1").Value If Right(sPath, 1) <> "/" Then sPath = sPath & "\" End If sFile = Dir(sPath & "*.xls") Do While sFile <> "" Workbooks.Open sPath & sFile sFile = Dir() Loop Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob eine Arbeitsmappe geöffnet ist

Problem: Wie kann ich vor dem Öffnenbefehl feststellen, ob eine bestimmte Arbeitsmappe bereits geöffnet ist.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function WkbExists(sFile As String) As Boolean
    Dim wkb As Object
    On Error Resume Next
    Set wkb = Workbooks(sFile)
    If Err = 0 And Not wkb Is Nothing Then
        WkbExists = True
    End If
    On Error GoTo 0
End Function

Sub Aufruf()
    Dim sFile As String
    sFile = InputBox( _
        prompt:="Dateiname:", _
        Default:="Personl.xls")
    If WkbExists(sFile) Then
        MsgBox "Mappe ist offen!"
    Else
        MsgBox "Mappe ist geschlossen!"
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verzeichnisdialog mit vorgegebenem Verzeichnis aufrufen

Problem:	Es soll ein Ordnerdialog mit dem in Zelle B1 genannten Ordner als Startverzeichnis aufgerufen und das Anlegen eines neuen Ordners ermöglicht werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub OrdnerDialog() Const WINDOW_HANDLE = 0 Const NO_OPTIONS = 0 Dim Wscript As Application Dim objShell As Object Dim objFolder As Object Dim objFolderItem As Object Dim objFile As Object Dim colFiles As Object Dim objWMIService As Object Dim strComputer As String, strPath As String Set objShell = CreateObject("Shell.Application") Set objFolder = objShell.BrowseForFolder _ (WINDOW_HANDLE, "Wählen Sie einen Ordner aus:", NO_OPTIONS, Range("B1").Value) If objFolder Is Nothing Then Exit Sub Set objFolderItem = objFolder.Self strPath = objFolderItem.Path strPath = Replace(strPath, "\", "\\") strComputer = "." Set objWMIService = GetObject _ ("winmgmts:" & "!\\" & strComputer & "\root\cimv2") Set colFiles = objWMIService.ExecQuery _ ("Select * from Win32_Directory where name = '" & strPath & "'") On Error GoTo ERRORHANDLER For Each objFile In colFiles Wscript.Echo "Readable: " & objFile.Readable Next ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus GetOpenFilename-Dialog eine Mehrfachauswahl auslesen

Problem:	Der GetOpenFileName-Dialog soll mit dem Parameter True für MultiSelect aufgerufen und die ausgewählten Dateien sollen genannt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub FileSelection() Dim var As Variant Dim iCounter As Integer var = Application.GetOpenFilename(_ FileFilter:="Excel-Dateien (*.xls), *.xls", _ MultiSelect:=True) On Error GoTo ERRORHANDLER For iCounter = 1 To UBound(var) MsgBox iCounter & ". Datei von " & UBound(var) & ": " & var(iCounter) Next iCounter Exit Sub ERRORHANDLER: Beep MsgBox "Abbruch!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Speichern ausschließlich über UserForm-Schaltfläche

Problem: Die Arbeitsmappe darf ausschließlich über eine UserForm-Schaltfläche gespeichert werden.

Lösung: Geben Sie den nachfolgenden Code in die genannten Module ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeSave( _
    ByVal SaveAsUI As Boolean, Cancel As Boolean)
    If gbln = False Then
        Beep
        MsgBox "Ich darf nicht speichern!"
        Cancel = True
    End If
    gbln = False
End Sub

StandardModule: Modul1

Public gbln As Boolean

Sub DialogAufruf()
    frmSave.Show
End Sub

ClassModule: frmSave

Private Sub cmdSave_Click()
    If MsgBox( _
        prompt:="Speichern?", _
        Buttons:=vbQuestion + vbYesNo _
    ) = vbYes Then
        gbln = True
        Worksheets(1).Select
        ActiveWorkbook.Save
    Else
        gbln = False
    End If
    Unload Me
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus Dialog Verzeichnis auswählen, dann neuen Ordner erstellen

Problem:	Wie kann ich aus einem Dialog heraus ein Verzeichnis auswählen, dann ein neues Verzeichnis festlegen und erstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Public Type BROWSEINFO hOwner As Long pidlRoot As Long pszDisplayName As String lpszTitle As String ulFlags As Long lpfn As Long lParam As Long iImage As Long End Type Declare Function SHGetPathFromIDList Lib "shell32.dll" _ Alias "SHGetPathFromIDListA" (ByVal pidl As Long, ByVal pszPath As String) As Long Declare Function SHBrowseForFolder Lib "shell32.dll" _ Alias "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long Function GetDirectory(Optional Msg As String) As String Dim bInfo As BROWSEINFO Dim Path As String Dim r As Long, x As Long, pos As Integer bInfo.pidlRoot = 0& If IsMissing(Msg) Then bInfo.lpszTitle = "Wählen Sie bitte einen Ordner aus." Else bInfo.lpszTitle = Msg End If bInfo.ulFlags = &H1 x = SHBrowseForFolder(bInfo) Path = Space\$(512) r = SHGetPathFromIDList(ByVal x, ByVal Path) If r Then pos = InStr(Path, Chr\$(0)) GetDirectory = Left(Path, pos - 1) Else GetDirectory = "" End If End Function Sub NeuesVerzeichnis() Dim sDir As String sDir = GetDirectory If sDir = "" Then Exit Sub If Right(sDir, 1) <> "\" Then sDir = sDir & "\" SendKeys "{end}" sDir = InputBox("Neuen Verzeichnisnamen eingeben:", , sDir) If sDir = "" Then Exit Sub On Error GoTo ERRORHANDLER MkDir sDir Exit Sub ERRORHANDLER: MsgBox "Das Verzeichnis konnte nicht erstellt werden!" End Sub</pre>

Ordnerdialog mit nachfolgendem Dateidialog aufrufen

Problem:	Wie kann ich zuerst einen Dialog zur Auswahl eines Verzeichnisses und danach zur Auswahl einer Excel-Datei aus diesem Verzeichnis aufrufen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: frmDateiListe Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim iCounter As Integer Dim sOrdner As String sOrdner = GetDirectory("Ein Verzeichnis auswählen!") If sOrdner = "" Then Unload Me Else Me.Caption = "Verzeichnis: " & sOrdner With Application.FileSearch .LookIn = sOrdner .FileType = msoFileTypeExcelWorkbooks .Execute For iCounter = 1 To .FoundFiles.Count lstFiles.AddItem .FoundFiles(iCounter) Next iCounter End With End If End Sub StandardModule: basMain Public Type BROWSEINFO hOwner As Long pidlRoot As Long pszDisplayName As String lpszTitle As String ulFlags As Long lpfn As Long lParam As Long iImage As Long End Type Declare Function SHGetPathFromIDList Lib "shell32.dll" _ Alias "SHGetPathFromIDListA" (ByVal pidl As Long, ByVal pszPath As String) As Long Declare Function SHBrowseForFolder Lib "shell32.dll" _ Alias "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long Function GetDirectory(Optional Msg As String) As String Dim bInfo As BROWSEINFO Dim Path As String Dim r As Long, x As Long, pos As Integer bInfo.pidlRoot = 0& If IsMissing(Msg) Then bInfo.lpszTitle = "Wählen Sie bitte einen Ordner aus." Else bInfo.lpszTitle = Msg End If bInfo.ulFlags = &H1 x = SHBrowseForFolder(bInfo) Path = Space\$(512) r = SHGetPathFromIDList(ByVal x, ByVal Path) If r Then pos = InStr(Path, Chr\$(0)) GetDirectory = Left(Path, pos - 1) Else GetDirectory = "" End If End Function Sub DialogAufruf() frmDateiListe.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateien suchen

Problem: Im Verzeichnis C:\ wird nach allen Dateien mit den Extensions *.xls, *.gif und *.jpg gesucht.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub DateienSuchen()
    Dim arr As Variant
    Dim iCounter As Integer, iFile As Integer
    arr = Array("*.xls", "*.gif", "*.jpg")
    For iCounter = LBound(arr) To UBound(arr)
        With Application.FileSearch
            .NewSearch
            .LookIn = "c:"
            .Filename = arr(iCounter)
            .MatchTextExactly = True
            .SearchSubFolders = False
            .Execute
            For iFile = 1 To .FoundFiles.Count
                Cells(iFile, 1).Value = .FoundFiles(iFile)
            Next iFile
        End With
    Next iCounter
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateinamen filtern

Problem:	Wie kann ich beim VBA-Aufruf des Öffnen-Dialogs die angezeigten Dateien auf Typen, beispielsweise Textdateien, begrenzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FileOpen() Dim vFile As Variant vFile = Application.GetOpenFilename(_ "Textdateien (*.txt),*.txt") If vFile = False Then Exit Sub Workbooks.Open vFile End Sub</pre>



Dateien listen und Blätter in neue Arbeitsmappe kopieren

Problem:	Über die erste Schaltfläche sollen alle Excel-Arbeitsmappen eines auszuwählenden Verzeichnisses in Spalte A gelistet werden. Blätter mit vorgegebenem Index sollen kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn zweir Schaltflächen zu.
VBA-Code:	StandardModule: basMain Sub ListFiles() Dim iCounter As Integer Dim sPath As String Range("A2:B65536").ClearContents sPath = GetDirectory("Bitte ein Verzeichnis auswählen:") If sPath = "" Then Exit Sub With Application.FileSearch .FileType = msoFileTypeExcelWorkbooks .LookIn = sPath .Execute For iCounter = 1 To .FoundFiles.Count Cells(iCounter + 1, 1).Value = .FoundFiles(iCounter) Next iCounter End With Columns("A:B").AutoFit End Sub Sub CopySheets() Dim wkb As Workbook, wkbSource As Workbook Dim wks As Worksheet Dim iCounter As Integer Application.ScreenUpdating = False Set wks = ActiveSheet On Error GoTo ERRORHANDLER Application.EnableEvents = False Set wkb = Workbooks.Add(1) iCounter = 2 Do Until IsEmpty(wks.Cells(iCounter, 1)) If Not IsEmpty(wks.Cells(iCounter, 2)) Then Set wkbSource = Workbooks.Open(_ wks.Cells(iCounter, 1).Value, updatelinks:=False) Worksheets(wks.Cells(iCounter, 2).Value).Copy _ after:=wkb.Worksheets(wkb.Worksheets.Count) wkbSource.Close savechanges:=False End If iCounter = iCounter + 1 Loop ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = True End Sub StandardModule: basFunctions Public Type BROWSEINFO hOwner As Long pidlRoot As Long pszDisplayName As String lpszTitle As String ulFlags As Long lpfn As Long lParam As Long iImage As Long End Type Declare Function SHGetPathFromIDList Lib "shell32.dll" _ Alias "SHGetPathFromIDListA" (ByVal pidl As Long, _ ByVal pszPath As String) As Long Declare Function SHBrowseForFolder Lib "shell32.dll" _ Alias "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long Function GetDirectory(Optional msg) As String Dim bInfo As BROWSEINFO Dim Path As String Dim r As Long, x As Long, pos As Integer bInfo.pidlRoot = 0& If IsMissing(msg) Then bInfo.lpszTitle = "Wählen Sie bitte einen Ordner aus." Else bInfo.lpszTitle = msg End If bInfo.ulFlags = &H1 x = SHBrowseForFolder(bInfo) Path = Space\$(512) r = SHGetPathFromIDList(ByVal x, ByVal Path) If r Then pos = InStr(Path, Chr\$(0)) GetDirectory = Left(Path, pos - 1) Else GetDirectory = "" End If End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sicherungskopie in einem anderen Verzeichnis erstellen

Problem:	Wenn ich beim Speichern die CheckBox "Sicherungsdatei erstellen" aktiviere, wird die BackUp-Datei im gleichen Verzeichnis wie die Arbeitsmappe erstellt. Kann ich hier ein anderes Verzeichnis vorgeben?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SpeichernMitKopie() Dim sPath As String, sFile As String, sName As String sPath = Range("B1").Value If Right(sPath, 1) <> "\" Then sPath = sPath & "\" End If sName = ThisWorkbook.FullName sFile = ThisWorkbook.Name sFile = Left(sFile, Len(sFile) - 1) & "bak" ThisWorkbook.SaveAs sPath & sFile ActiveWorkbook.SaveAs sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei kopieren, wenn noch nicht vorhanden

Problem:	Wie kann ich vor dem Kopieren einer Datei in ein anderes Verzeichnis prüfen, ob diese dort schon vorhanden ist?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DateiKopieren() Dim sQuelle As String, sZiel As String sQuelle = Range("B1").Value sZiel = Range("B2").Value If Dir(sZiel) = "" Then FileCopy sQuelle, sZiel MsgBox "Datei wurde kopiert!" Else MsgBox "Datei war schon vorhanden!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe blitzschnell kopieren

Problem:	Die aktive Arbeitsmappe soll mit einer schnellen Methode 100 mal in das Standardverzeichnis kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Kopieren() Dim iCounter As Integer Dim sFile As String, sPath As String sPath = Application.DefaultFilePath & "\" Close Open "speichern.bat" For Output As #1 For iCounter = 1 To 100 sFile = ThisWorkbook.FullName & " " sFile = sFile & sPath & "test" & iCounter & ".xls" Print #1, "copy " & sFile Next iCounter Close Call Win32WaitTilFinished("command.com /c speichern.bat") Kill "speichern.bat" MsgBox "Die Dateien wurde angelegt!" End Sub StandardModule: basFunctions Public Const PROCESS_QUERY_INFORMATION = &H400 Public Const WAIT_TIMEOUT = &H102& Declare Function OpenProcess Lib "kernel32" (_ ByVal dwDesiredAccess As Long, _ ByVal bInheritHandle As Long, _ ByVal dwProcessId As Long) As Long Declare Function WaitForSingleObject Lib "kernel32" (_ ByVal hHandle As Long, _ ByVal dwMilliseconds As Long) As Long Sub Win32WaitTilFinished(ProgEXE As String) Dim ProcessID As Long Dim hProcess As Long Dim RetVal As Long ProcessID = Shell(ProgEXE, vbHide) hProcess = OpenProcess(PROCESS_QUERY_INFORMATION, False, ProcessID) Do DoEvents RetVal = WaitForSingleObject(hProcess, 50) Loop Until RetVal <> WAIT_TIMEOUT End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Laufwerk über InputBox auswählen

Problem:	Die zur Verfügung stehenden Laufwerke (ohne CD-ROM) sollen in einer InputBox zur Auswahl angegeben werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub LaufwerkAuswahl() Dim iCounter As Integer Dim sDrive As String On Error Resume Next For iCounter = 97 To 122 ChDrive Chr(iCounter) If Err > 0 Then Err.Clear Else sDrive = sDrive & Chr(iCounter) & ", " End If Next iCounter On Error GoTo 0 sDrive = Left(sDrive, Len(sDrive) - 1) sDrive = InputBox(_ prompt:="Die Laufwerke: " & sDrive, _ Title:="Bitte ein Laufwerk auswählen") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Diskette in Laufwerk A?

Problem:	Wie kann ich prüfen, ob in Laufwerk A eine Diskette eingelegt ist?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub InsertDisk() Dim sFile As String On Error GoTo ERRORHANDLER sFile = Dir("a:\", vbVolume) MsgBox prompt:="Alles klar!" Exit Sub ERRORHANDLER: MsgBox prompt:="Bitte eine Diskette einlegen!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob eine Datei im Netz bereits geöffnet ist

Problem:	Es soll festgestellt werden, ob auf eine Netzwerkdatei von einem anderen Anwender zugegriffen wird.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub TestFileOpen() Dim iOpen As Integer Dim sFile As String sFile = InputBox("Path and Filename:", , "c:\test\test.xls") If sFile = "" Then Exit Sub iOpen = TestOpen(sFile) Select Case iOpen Case 0: MsgBox "Datei " & sFile & " ist frei" Case 1: MsgBox "Datei " & sFile & " ist geöffnet" Case 2: MsgBox "Datei " & sFile & " wurde nicht gefunden" End Select End Sub Private Function TestOpen(sPath As String) As Integer If Dir(sPath) = "" Then TestOpen = 2 Else On Error GoTo ERRORHANDLER Open sPath For Random Access Read Lock Read Write As #1 Close #1 End If ERRORHANDLER: If Err = 70 Then TestOpen = 1 End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei entsprechend der Eingabe des Dateinamens öffnen

Problem:	Bei Eingabe eines Dateinamens in Zelle B2 soll die Datei in dem in Zelle B1 genannten Verzeichnis gesucht und geöffnet werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim sFile As String If Target.Address <> "\$B\$2" Then Exit Sub sFile = Dir(Range("B1") & "\" & Target) If sFile <> "" Then Workbooks.Open Range("B1") & "\" & sFile Else MsgBox "Arbeitsmappe wurde nicht gefunden!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob Datei existiert und bei nein Meldung

Problem: Wie kann ich abfragen, ob eine Arbeitsmappe schon geöffnet ist bzw. überhaupt existiert? existiert. Wenn sie existiert soll sie geöffnet, wenn sie nicht

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub MappeOeffnen()  
    Dim wkb As Workbook  
    Dim sFile As String, sPath As String  
    sFile = Range("B1").Value  
    sPath = Range("B2").Value  
    If Right(sPath, 1) <> "\" Then  
        sPath = sPath & "\"  
    End If  
    On Error Resume Next  
    Set wkb = Workbooks(sFile)  
    If Err > 0 Or wkb Is Nothing Then  
        Err.Clear  
        If Dir(sPath & sFile) = "" Then  
            MsgBox "Die Datei wurde nicht gefunden!"  
        Else  
            Workbooks.Open sPath & sFile  
        End If  
    Else  
        wkb.Activate  
    End If  
End Sub
```



Dateiliste mit DOS- und Windowsnamen erstellen

Problem:	Wie kann ich im aktuellen Blatt alle Dateien des Rechners listen? Ich benötige Angaben über DOS- und Windows-Namen, Erstellungs- und Änderungsdatum.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Michael Schwimmer Declare Function FindClose Lib "kernel32" _ (ByVal hFindFile As Long) As Long Declare Function FindFirstFile Lib "kernel32" Alias "FindFirstFileA" _ (ByVal lpFileName As String, lpFindFileData As WIN32_FIND_DATA) As Long Declare Function FindNextFile Lib "kernel32" Alias "FindNextFileA" _ (ByVal hFindFile As Long, lpFindFileData As WIN32_FIND_DATA) As Long Public Const FILE_ATTRIBUTE_DIRECTORY = &H10 Public Const MAX_PATH = 260 Declare Function FileTimeToSystemTime Lib "kernel32" _ (lpFileTime As FILETIME, lpSystemTime As SYSTEMTIME) As Long Type SYSTEMTIME wYear As Integer wMonth As Integer wDayOfWeek As Integer wDay As Integer wHour As Integer wMinute As Integer wSecond As Integer wMilliseconds As Integer End Type Type FILETIME dwLowDateTime As Long dwHighDateTime As Long End Type Type WIN32_FIND_DATA dwFileAttributes As Long ftCreationTime As FILETIME ftLastAccessTime As FILETIME ftLastWriteTime As FILETIME nFileSizeHigh As Long nFileSizeLow As Long dwReserved0 As Long dwReserved1 As Long cFileName As String * MAX_PATH cAlternate As String * 14 End Type Type Dateistruktur Datei As String MSDOSName As String Pfad As String KompletterPfad As String Erstellungsdatum As String LetzterZugriff As String LetzteÄnderung As String Größe As Long End Type Dim Dateiliste() As Dateistruktur Sub DateilisteErstellen() ReDim Dateiliste(1000) Dim Zähler As Long Dim Pfad As String Dim AnzahlDateien As Long Dim Datenblock(1 To 7) Pfad = InputBox("Bitte Pfad eingeben:", , "c:") If Right(Pfad, 1) = "\" Then Pfad = Left(Pfad, Len(Pfad) - 1) AnzahlDateien = DurchlaufePfad(Pfad, 1) For Zähler = 1 To AnzahlDateien Datenblock(1) = Dateiliste(Zähler).Datei Datenblock(2) = Dateiliste(Zähler).MSDOSName Datenblock(3) = Dateiliste(Zähler).KompletterPfad Datenblock(4) = Dateiliste(Zähler).Erstellungsdatum Datenblock(5) = Dateiliste(Zähler).LetzteÄnderung Datenblock(6) = Dateiliste(Zähler).LetzterZugriff Datenblock(7) = Dateiliste(Zähler).Größe Range(Cells(Zähler + 1, 1), Cells(Zähler + 1, 7)) = _ Datenblock Next Columns.AutoFit End Sub Function DurchlaufePfad(ByVal Pfadname As String, _ ByVal Dateiindex As Long) As Long Dim Suchhandle As Long Dim Rückgabewert1 As Long Dim dummy Dim Suchkriterium As String Dim Zeile As Long Dim Erstellungsdatum As Date Dim Datumszwischenspeicher As SYSTEMTIME Dim Filedaten As WIN32_FIND_DATA Pfadname = Trim(Pfadname) If Right\$(Pfadname, 1) = "\" Then Suchkriterium = Pfadname & "*"' Else Suchkriterium = Pfadname & "*" End If Zeile = Dateiindex Filedaten.cAlternate = String(14, Chr(0)) Filedaten.cFileName = String(260, Chr(0)) Suchhandle = FindFirstFile(Suchkriterium, Filedaten) Rückgabewert1 = Suchhandle Do While Rückgabewert1 <> 0 Filedaten.cFileName = Left(Filedaten.cFileName, InStr _ (1, Filedaten.cFileName, Chr(0)) - 1) Filedaten.cAlternate = Left(Filedaten.cAlternate, InStr _ (1, Filedaten.cAlternate, Chr(0)) - 1) If Trim(Filedaten.cFileName) <> "." And Trim _ (Filedaten.cFileName) <> ".." Then If Filedaten.dwFileAttributes = FILE_ATTRIBUTE_DIRECTORY Then Zeile = DurchlaufePfad((Pfadname & "\" & _ Trim(Filedaten.cFileName)), Zeile) End If If Zeile = UBound(Dateiliste) Then ReDim _ Preserve Dateiliste(Zeile + 100) Dateiliste(Zeile).Datei = Trim(Filedaten.cFileName) Dateiliste(Zeile).MSDOSName = Trim(Filedaten.cAlternate) Dateiliste(Zeile).Pfad = Pfadname Dateiliste(Zeile).KompletterPfad = Pfadname & "\" & _ Dateiliste(Zeile).Datei dummy = FileTimeToSystemTime(Filedaten.ftCreationTime, _ Datumszwischenspeicher) Dateiliste(Zeile).Erstellungsdatum = DateSerial _ (Datumszwischenspeicher.wYear, Datumszwischenspeicher.wMonth, _ Datumszwischenspeicher.wDay) _ + TimeSerial(Datumszwischenspeicher.wHour, _ Datumszwischenspeicher.wMinute, Datumszwischenspeicher.wSecond) dummy = FileTimeToSystemTime(Filedaten.ftLastAccessTime, _ Datumszwischenspeicher) Dateiliste(Zeile).LetzterZugriff = DateSerial(_ Datumszwischenspeicher.wYear, Datumszwischenspeicher.wMonth, _ Datumszwischenspeicher.wDay) _ + TimeSerial(Datumszwischenspeicher.wHour, _ Datumszwischenspeicher.wMinute, Datumszwischenspeicher.wSecond) dummy = FileTimeToSystemTime(Filedaten.ftLastWriteTime, _ Datumszwischenspeicher) Dateiliste(Zeile).LetzteÄnderung = _ DateSerial(Datumszwischenspeicher.wYear, _ Datumszwischenspeicher.wMonth, Datumszwischenspeicher.wDay) _ + TimeSerial(Datumszwischenspeicher.wHour, _ Datumszwischenspeicher.wMinute, Datumszwischenspeicher.wSecond) Dateiliste(Zeile).Größe = Filedaten.nFileSizeLow Zeile = Zeile + 1 End If Filedaten.cAlternate = String(14, Chr(0)) Filedaten.cFileName = String(260, Chr(0)) Rückgabewert1 = FindNextFile(Suchhandle, Filedaten) Loop DurchlaufePfad = Zeile dummy = FindClose(Suchhandle) End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blatttyp und Arbeitsmappennamen prüfen

Problem:	Wie kann ich den Blatttyp (Arbeitsblatt oder Diagramm) und den kompletten Dateinamen ermitteln?
Lösung:	Den nachstehenden Code bei XL5/7 in ein englischsprachiges, bei XL8 ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: modMain Sub Pruefen() MsgBox "Das aktive Blatt ist ein " & TypeName(ActiveSheet) If InStr(ThisWorkbook.FullName, ":\") Then MsgBox "Die Arbeitsmappe wurde gespeichert unter " & vbLf & _ ThisWorkbook.FullName Else MsgBox "Die Arbeitsmappe ist noch nicht gespeichert!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateinamen in Tabelle einlesen

Problem: Wie kann ich die Dateinamen eines anzugebenden Verzeichnisses und einer anzugebenden Suffix in ein Tabellenblatt einlesen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub DateienEinlesen()  
    Dim sFile As String, sPattern As String, sPath As String  
    Dim iRow As Integer  
    Columns(1).ClearContents  
    sPath = InputBox( _  
        prompt:="Pfad:", Default:="c:\eigene dateien")  
    If Right(sPath, 1) <> "\" Then sPath = sPath & "\"  
    sPattern = InputBox( _  
        prompt:="Dateifilter:", Default:="*.xls")  
    sFile = Dir(sPath & sPattern)  
    Do Until sFile = ""  
        iRow = iRow + 1  
        Cells(iRow, 1).Value = sFile  
        sFile = Dir()  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei umbenennen

Problem:	Wie kann ich eine Datei umbenennen? Der Quellname steht in Zelle A2 und der Zielname in B2. Fehlermeldungen sollen abgefangen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DateiUmbenennen() Const csMsg As String = "Datei existiert, überschreiben?" On Error GoTo ERRORHANDLER Name Range("A2").Value As Range("B2").Value Exit Sub ERRORHANDLER: If Err = 58 Then Beep If MsgBox(csMsg, vbQuestion + vbYesNo) = vbNo Then Exit Sub Kill Range("B2").Value Resume ElseIf Err = 53 Then Beep MsgBox "Datei nicht gefunden!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermitteln von Datei- und Pfadnamen

Problem:	Wie kann ich aus einem mit dem Befehl "GetOpenFilename" ermittelten vollen Dateinamen den eigentlichen Dateinamen und den Pfad ermitteln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FileNamePath() Dim vAdr As Variant vAdr = Application.GetOpenFilename If vAdr = False Then Exit Sub MsgBox vAdr & Chr(13) & Dir(vAdr) & _ Chr(13) & Left(vAdr, Len(vAdr) - Len(Dir(vAdr))) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

400 leere Textdateien anlegen

Problem:	Wie kann ich im aktuellen Verzeichnis 400 Textdateien anlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CreateTextFiles() Const csMsg As String = "Die Dateien können nicht " & _ "auf CD-ROM angelegt werden. Speichern Sie die " & _ "Arbeitsmappe auf Ihrer Festplatte und versuchen " & _ "Sie es erneut." Dim iCounter As Integer Dim sFile As String On Error GoTo ERRORHANDLER For iCounter = 1 To 400 sFile = Format(iCounter, "0000") sFile = CurDir & "\" & sFile & ".txt" Open sFile For Output As #1 Print #1, "Datei Nummer " & iCounter Close Next iCounter MsgBox prompt:="Die Textdateien wurden angelegt" & vbCrLf & _ "im Verzeichnis " & CurDir Exit Sub ERRORHANDLER: Close MsgBox prompt:=csMsg End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateiname aus dem vollen Namen extrahieren

Problem: Mit der GetOpenFileName-Methode kann ich den Datei- mit dem Pfadnamen abrufen. Wie kann ich aus dieser Zeichenfolge den Dateinamen filtern?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub FileFilter()
    Dim vFile As Variant
    vFile = Application.GetOpenFilename
    If vFile = False Then Exit Sub
    vFile = Dir(vFile)
    MsgBox vFile
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateien öffnen, Verknüpfungen aktualisieren, speichern, schießen

Problem:	Die in Spalte A genannten Dateien sollen geöffnet, die Verknüpfungen aktualisiert gespeichert und wieder geschlossen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub OeffnenSchliessen() Dim iRow As Integer Dim sFile As String On Error GoTo ERRORHANDLER Application.ScreenUpdating = False Application.EnableEvents = False iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) sFile = Cells(iRow, 1).Value If Dir(sFile) <> "" Then Workbooks.Open _ Filename:=sFile, _ updatelinks:=True ActiveWorkbook.Close savechanges:=True End If iRow = iRow + 1 Loop ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfen, ob Arbeitsmappe geöffnet und wenn nein, öffnen

Problem:	Es soll geprüft werden, ob eine Datei mit dem Namen Test.xls geöffnet ist. Wenn ja, ist sie zu aktivieren, wenn nein zu öffnen. Existiert die Datei nicht im Verzeichnis dieser Arbeitsmappe, ist eine entsprechende Meldung auszugeben.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub DateiOeffnen() Dim sFile As String, sPath As String sFile = "Test.xls" sPath = ThisWorkbook.Path & "\" & sFile If WkbExists("Test.xls") = False Then If Dir(sPath) = "" Then MsgBox "Datei " & sPath & " wurde nicht gefunden!" Else Workbooks.Open sPath End If Else Workbooks(sFile).Activate End If End Sub Private Function WkbExists(sFile As String) As Boolean Dim wkb As Object On Error Resume Next Set wkb = Workbooks(sFile) If Not wkb Is Nothing Then WkbExists = True End If On Error GoTo 0 End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Zelleintrag Datei öffnen

Problem: Bei Ändern des Pfades in Zelle B1 oder des Dateinamens in Zelle B2 soll die jeweilige Datei geöffnet werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim sFile As String
    If Intersect(Target, Range("B1:B2")) Is Nothing Then Exit Sub
    sFile = Range("B1").Value & "\" & Range("B2").Value
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Datei " & sFile & " wurde nicht gefunden!"
    Else
        Workbooks.Open sFile
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateiauswahl-Dialog mit vorgegebenen Dateinamen aufrufen

Problem: Der Datei-Öffnen-Dialog soll mit einem vorgegebenen Dateinamen aufgerufen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub CallDialog()
    Dim vFile As Variant
    SendKeys "test.xls{home}+{end}"
    Application.FindFile
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe auswählen und öffnen

Problem:	Es soll ein Dateiname ausgewählt und die zugehörige Arbeitsmappe geöffnet werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Sub WkbOpen() Dim vFile As Variant vFile = Application.GetOpenFilename("Excel-Arbeitsmappe (*.xls), *.xls") If vFile = False Then Exit Sub Workbooks.Open vFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei öffnen, wenn sie am aktuellen Tag geändert wurde

Problem:	Die Datei aus Zelle B1 soll auf ihr Änderungsdatum geprüft werden. Wurde sie am aktuellen Tag geändert, ist sie zu öffnen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub IfNew() Dim dat As Date If Dir(Range("B1").Value) = "" Then Beep MsgBox "Datei wurde nicht gefunden!" Exit Sub End If dat = FileDateTime(Range("B1").Value) If Fix(CDbl(dat)) = Date Then Workbooks.Open Range("B1").Value End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateien listen und gekennzeichnete öffnen

Problem: Mit der ersten Schaltfläche sollen die Dateinamen des in Zelle B1 genannten Verzeichnisses gelistet und mit der zweiten die in Spalte B mit einem x gekennzeichneten Arbeitsmappen geöffnet werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub ReadFiles()  
    Dim iCounter As Integer  
    With Application.FileSearch  
        .LookIn = Range("B1").Value  
        .Filename = "*.xls"  
        .Execute  
        For iCounter = 1 To .FoundFiles.Count  
            Cells(iCounter + 1, 1).Value = Dir(.FoundFiles(iCounter))  
        Next iCounter  
    End With  
End Sub  
  
Sub OpenFiles()  
    Dim wks As Worksheet  
    Dim iRow As Integer  
    Dim sPath As String  
    Application.ScreenUpdating = False  
    Application.EnableEvents = False  
    Set wks = ActiveSheet  
    iRow = 1  
    sPath = Range("B1").Value  
    On Error GoTo ERRORHANDLER  
    Do Until IsEmpty(wks.Cells(iRow, 1))  
        iRow = iRow + 1  
        If LCase(wks.Cells(iRow, 2).Value) = "x" Then  
            Workbooks.Open sPath & "\" & wks.Cells(iRow, 1).Value, False  
        End If  
    Loop  
ERRORHANDLER:  
    Application.EnableEvents = True  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Warten, bis eine Datei geschlossen wird, dann öffnen

Problem: Es soll in einer Schleife geprüft werden, ob eine Datei bereits geöffnet ist. Sobald der Anwender, der die Datei in Benutzung hat, diese schließt, soll sie hier geöffnet werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub TestFileOpen()
    Dim iOpen As Integer
    Dim sFile As String
    sFile = InputBox("Path and Filename:", , "c:\test\test.xls")
    If sFile = "" Then Exit Sub
    Do While TestOpen(sFile) = 1
        Loop
    Workbooks.Open sFile
End Sub

Private Function TestOpen(sPath As String) As Integer
    If Dir(sPath) = "" Then
        TestOpen = 2
    Else
        On Error GoTo ERRORHANDLER
        Open sPath For Random Access Read Lock Read Write As #1
        Close #1
    End If
ERRORHANDLER:
    If Err = 70 Then TestOpen = 1
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorlage auswählen und Arbeitsmappe öffnen

Problem:	Wie kann ich aus einem Dialog aus einem festgelegten Vorlagenordner eine Vorlage auswählen und eine darauf basierende Arbeitsmappe laden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub VorlageAuswaehlen() Dim vTemplate As Variant Dim sFolder As String, sAct As String sAct = CurDir sFolder = Range("B1").Value ChDrive Left(sFolder, 1) ChDir sFolder vTemplate = Application.GetOpenFilename(_ "Vorlagen(*.xlt), *.xlt") If vTemplate = False Then Exit Sub Workbooks.Add vTemplate ChDrive Left(sAct, 1) ChDir sAct End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei auswählen, öffnen und Formelsuchdialog anzeigen

Problem:	Wie kann ich einen Dialog zur Dateiauswahl anzeigen lassen, die im Dialog ausgewählte Datei öffnen und unmittelbar nach dem Öffnen den Formelsuchdialog aufrufen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub cmdDateiOeffnen_Click() Dim vFile As Variant vFile = Application.GetOpenFilename If vFile = False Then Exit Sub Workbooks.Open vFile Application.Dialogs(xlDialogFormulaFind).Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Arbeitsmappen eines Verzeichnisses öffnen

Problem: Wie kann ich alle Arbeitsmappen eines Verzeichnisses durch das Drücken einer Schaltfläche öffnen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub DateienOeffnen()  
    Dim arr As Variant  
    Dim iCounter As Integer  
    Dim sPath As String  
    Dim bln As Boolean  
    Application.ScreenUpdating = False  
    bln = Application.DisplayStatusBar  
    Application.DisplayStatusBar = True  
    sPath = Range("B1").Value  
    arr = FileArray(sPath, "*.xls")  
    For iCounter = 1 To UBound(arr)  
        Application.StatusBar = "Öffne Datei " & _  
            arr(iCounter) & "..."  
        Workbooks.Open sPath & arr(iCounter)  
    Next iCounter  
    Application.StatusBar = False  
    Application.DisplayStatusBar = bln  
    Application.ScreenUpdating = True  
End Sub  
  
Private Function FileArray(sPath As String, sPattern As String)  
    Dim arrFiles()  
    Dim iCounter As Integer  
    Dim sFile As String  
    If Right(sPath, 1) <> "\" Then sPath = sPath & "\"  
    sFile = Dir(sPath & sPattern)  
    Do While sFile <> ""  
        iCounter = iCounter + 1  
        ReDim Preserve arrFiles(1 To iCounter)  
        arrFiles(iCounter) = sFile  
        sFile = Dir()  
    Loop  
    FileArray = arrFiles  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Geöffnete Arbeitsmappen mit Arbeitsblättern listen

Problem: Wie kann ich alle geöffneten Arbeitsmappen mit den zugehörigen Arbeitsblättern auflisten?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Listen()
    Dim wkb As Workbook
    Dim wks As Worksheet
    Dim iRow As Integer
    For Each wkb In Workbooks
        For Each wks In wkb.Worksheets
            iRow = iRow + 1
            Cells(iRow, 1).Value = wkb.Name
            Cells(iRow, 2).Value = wks.Name
        Next wks
    Next wkb
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei öffnen, deren Name in einer Zelle steht

Problem:	Wie kann ich eine Arbeitsmappe öffnen, deren Name in einer Zelle vorgegeben ist?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DateiOeffnen() Workbooks.Open Range("A2").Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe ohne Rückfrage nach Aktualisierung öffnen

Problem:	Wie kann ich eine Arbeitsmappe öffnen, ohne daß eine Rückfrage nach der Aktualisierung von Verknüpfungen erfolgt und ohne daß diese aktualisiert werden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub OpenOhneLink() Workbooks.Open FileName:=Range("A2").Value, _ UpdateLinks:=False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Löschen von Dateien beim Schließen einer Arbeitsmappe

Problem:	Beim Schließen der Arbeitsmappe sollen alle *.doc- und *.pdf-Dateien im Verzeichnis c:\temp gelöscht werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Dim arr As Variant Dim iCounter As Integer, iArray As Integer On Error GoTo ERRORHANDLER arr = Array("*.pdf", "*.doc") For iArray = 0 To 1 With Application.FileSearch .NewSearch .LookIn = Range("B1").Value .FileName = arr(iArray) .Execute For iCounter = 1 To .FoundFiles.Count Kill .FoundFiles(iCounter) Next iCounter End With Next iArray Exit Sub ERRORHANDLER: MsgBox "Datei konnte nicht gelöscht werden!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe nur über Schaltfläche schließen

Problem:	Wie kann ich erreichen, dass eine Arbeitsmappe nur über eine Schaltfläche geschlossen werden kann?
Lösung:	Geben Sie den nachfolgenden Code in die genannten Module ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Sub Workbook_BeforeClose(Cancel As Boolean) Dim nme As Name On Error Resume Next Set nme = ThisWorkbook.Names("OK") If Err > 0 Or nme Is Nothing Then Cancel = True End Sub Private Sub Workbook_Open() On Error Resume Next ThisWorkbook.Names("OK").Delete On Error GoTo 0 End Sub StandardModule: basMain Sub Schliessen() ThisWorkbook.Names.Add "OK", _ RefersTo:=Range("A1"), Visible:=False ThisWorkbook.Close End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe ohne Rückfrage schließen

Problem:	Wie kann ich eine Arbeitsmappe ohne Rückfragen, ob die Datei gespeichert werden soll und ob Daten in der Zwischenablage bleiben sollen, schließen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Schliessen1() On Error GoTo ERRORHANDLER Application.CutCopyMode = False ActiveWorkbook.Close True Exit Sub ERRORHANDLER: MsgBox "Speichern nicht möglich!" End Sub Sub Schliessen2() Application.CutCopyMode = False ActiveWorkbook.Close False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Arbeitsmappen außer einer schließen

Problem:	Wie kann ich alle Arbeitsmappen außer der aktiven schließen, ohne sie zu speichern und ohne eine diesbezügliche Abfrage?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Schliessen() Dim wkb As Workbook For Each wkb In Workbooks If wkb.Name <> ThisWorkbook.Name Then wkb.Close savechanges:=False End If Next wkb End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktive Arbeitsmappe ohne VBA-Code und Button speichern

Problem:	Die aktive Arbeitsmappe soll über die eine Schaltfläche ohne VBA-Code und ohne Schaltfläche gespeichert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SaveOnlySheets() Dim iWks As Integer Dim sFile As String iWks = ActiveSheet.Index sFile = Application.GetOpenFilename("Excel-Dateien (*.xls), *.xls") If sFile = "" Then Exit Sub Worksheets.Copy Worksheets(iWks).Buttons.Delete Application.DisplayAlerts = False ActiveWorkbook.SaveAs sFile Application.DisplayAlerts = True ThisWorkbook.Close savechanges:=False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Speichern-Dialog mit Dateinamen-Vorschlag aufrufen

Problem:	Der Speicherndialog soll aufgerufen und der in Zelle B1 genannte Dateiname vorgeschlagen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Speichern() Application.SendKeys Range("B1").Value Application.Dialogs(xlDialogSaveAs).Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei-Speichern-Dialog zusammen mit Paßwortdialog aufrufen

Problem:	Der Datei-Speichern-Dialog soll gemeinsam mit dem Dialog zur Festlegung eines Paßwortes aufgerufen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Speichern() Dim Results As Variant SendKeys "%xg{tab}" Results = Application.Dialogs(xlDialogSaveAs).Show If Results = False Then Exit Sub Else MsgBox "Gespeichert" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte nach Gruppen aufteilen und in gesonderten Dateien speichern

Problem: Die Werte aus den Spalten A:C sollen je Filiale in gesonderten Arbeitsmapen in dem in Zelle G1 genannten Verzeichnis gespeichert werden.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code:

```
StandardModule: Modul1

Sub FilialenSpeichern()
    Dim rng As Range
    Dim col As New Collection
    Dim iRow As Integer
    Dim sFile As String
    Application.ScreenUpdating = False
    iRow = 2
    sFile = Range("G1").Value
    On Error Resume Next
    Do Until IsEmpty(Cells(iRow, 1))
        col.Add Cells(iRow, 1).Value, Cells(iRow, 1).Value
        iRow = iRow + 1
    Loop
    On Error GoTo 0
    Application.DisplayAlerts = False
    For iRow = 1 To col.Count
        Range("A1").CurrentRegion.AutoFilter field:=1, Criteria1:=col(iRow)
        Set rng = Range("A1").CurrentRegion.SpecialCells(xlCellTypeVisible)
        Workbooks.Add
        rng.Copy Range("A1")
        ActiveWorkbook.SaveAs sFile & "\" & col(iRow) & ".xls"
        ActiveWorkbook.Close savechanges:=True
    Next iRow
    Application.DisplayAlerts = True
    Application.ScreenUpdating = True
    ActiveSheet.AutoFilterMode = False
    MsgBox "Job erledigt!"
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Speichern-Befehl abfangen und unter Paßwort speichern

Problem:	Der Speichern-Befehl soll abgefangen werden. Danach soll die Arbeitsmappe mit Schreib-/ Leseschutz und einem dem Anwender nicht bekannten Passwort ("MeinPasswort") gespeichert werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Dim bln As Boolean Private Sub Workbook_BeforeSave(_ ByVal SaveAsUI As Boolean, Cancel As Boolean) Application.DisplayAlerts = False If bln = False Then Cancel = True bln = True ThisWorkbook.SaveAs _ FileName:=Range("B1").Value & "\test.xls", _ Password:="MeinPasswort" End If Application.DisplayAlerts = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einzel-Arbeitsblätter einer Arbeitsmappe speichern.

Problem:	Die drei letzten Blätter der Arbeitsmappe sollen im Verzeichnis der Arbeitsmappe unter ihrem Blattnamen gespeichert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SaveSheets() Dim iSheet As Integer Dim sPath As String Application.ScreenUpdating = False sPath = Range("B1").Value & "\" For iSheet = Worksheets.Count - 2 To Worksheets.Count Worksheets(iSheet).Copy ActiveWorkbook.SaveAs sPath & ActiveSheet.Name ActiveWorkbook.Close savechanges:=False Next iSheet Application.ScreenUpdating = True MsgBox "Job erledigt" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Freien Speicherplatz eines abzufragenden Laufwerks ermitteln.

Problem:	Der freie Speicherplatz eines abzufragenden Laufwerks ist zu ermitteln.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Festplatte() Dim var As Variant Dim sLW As String, sSpace As String, sFile As String sLW = InputBox("Laufwerk:", , "c") If sLW = "" Then Exit Sub sLW = Left(sLW, 1) sFile = sLW & ":\dirliste.txt" var = Shell("command.com /c dir *.xls >" & sFile, vbHide) Application.Wait Now + TimeSerial(0, 0, 1) Close Open sFile For Input As #1 Do Until EOF(1) Line Input #1, sSpace Loop Close sSpace = Trim(sSpace) sSpace = Right(sSpace, Len(sSpace) - InStr(sSpace, " ")) sSpace = Trim(sSpace) sSpace = Left(sSpace, InStr(sSpace, " ") - 1) MsgBox "Freie MB auf Laufwerk " & sLW & ":" & vbCrLf & sSpace Kill sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Speichern unter fortlaufender Nummer

Problem:	Eine Arbeitsmappe soll unter einer sich immer um 1 erhöhenden fortlaufenden Nummer gespeichert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Speichern() Dim sFile As String, sOld As String sOld = ThisWorkbook.Name If IsNumeric(Mid(sOld, 9, 4)) Then sOld = Mid(sOld, 9, 4) Else sOld = 0 End If sOld = Format(CInt(sOld) + 1, "0000") sFile = Range("A1").Text & "_" sFile = sFile & Range("B1").Text & "_" sFile = sFile & sOld & ".xls" ThisWorkbook.SaveAs Range("E1").Value & _ "\" & sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sicherheitskopie einzelner Arbeitsblätter

Problem: Die Blätter Tabelle1 und Tabelle3 sollen als Sicherheitsdatei mit der Datei-Suffix *.bak in einem BackUp-Verzeichnis gespeichert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub SaveAndBackUp()  
    Dim sPath As String, sFile As String  
    Application.ScreenUpdating = False  
    sPath = Range("B1").Value & "\"  
    With ActiveWorkbook  
        sFile = Left(.Name, Len(.Name) - 3) & "bak"  
    End With  
    Worksheets(Array("Tabelle1", "Tabelle3")).Copy  
    ActiveWorkbook.SaveAs sPath & sFile  
    ActiveWorkbook.Close savechanges:=False  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sicherheitskopien von Arbeitsmappen speichern

Problem:	Wie kann ich Sicherheitskopien von Arbeitsmappen anlegen, nachdem ich vorher über InputBox abgefragt habe, welche Arbeitsmappen gespeichert werden sollen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Speichern() Dim wkb As Workbook Dim sSave As String, sFile As String Dim sOld As String, sPath As String sSave = InputBox("Backup für ""alle"" oder " & _ """"aktive"" Arbeitsmappe(n)?"", , "aktive") If sSave = "" Then Exit Sub sPath = Range("B1").Value If sSave = "alle" Then For Each wkb In Workbooks sFile = wkb.Name sPath = wkb.FullName wkb.SaveAs Range("B1").Value & sFile wkb.SaveAs sPath Next wkb Else Set wkb = ActiveWorkbook sPath = wkb.FullName wkb.SaveAs Range("B1").Value & ActiveWorkbook.Name wkb.SaveAs sPath End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe mit absoluten Werten unter neuem Namen speichern

Problem:	Wie kann ich in der aktiven Arbeitsmappe alle Zellebezüge in absolute Werte umwandeln und die Arbeitsmappe unter einem anderen, durch ein InputBox abzufragendem Namen speichern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Speichern() Dim wks As Worksheet Dim sFile As String Application.ScreenUpdating = False sFile = Application.DefaultFilePath & "\test.xls" For Each wks In Worksheets With wks.UsedRange .Value = .Value End With Next wks ActiveWorkbook.SaveAs sFile Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einzelnes Tabellenblatt unter Tagesdatum speichern

Problem:	Wie kann ich ein einzelnes Tabellenblatt aus einer Arbeitsmappe unter dem aktuellen Tagesdatum speichern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Speichern() Dim sPath As String, sFile As String Application.ScreenUpdating = False sPath = Application.DefaultFilePath & "\" sFile = Format(Date, "yyyymmdd") & ".xls" ActiveSheet.Copy ActiveSheet.Buttons(1).Delete ActiveWorkbook.SaveAs sFile ActiveWorkbook.Close savechanges:=False MsgBox "Das Blatt wurde unter " & _ & sPath & sFile & " gespeichert!" Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausgewählte Tabellenblätter als eigene Arbeitsmappen speichern

Problem:	Wie kann ich mehrere, ausgewählte Arbeitsblätter einer Arbeitsmappe als einzelne Arbeitsmappen speichern? Der Dateiname soll abgefragt werden und es soll eine Warnmeldung erscheinen, wenn der Dateiname bereits existiert.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BlaetterSpeichern() Dim sFile As String ActiveWorkbook.Windows(1).SelectedSheets.Copy sFile = InputBox("Dateiname:") If sFile = "" Then Exit Sub If Dir(sFile) <> "" Then Beep If MsgBox("Existiert schon, überschreiben?", _ vbCritical + vbYesNo) = vbNo Then Exit Sub End If Application.DisplayAlerts = False ActiveWorkbook.SaveAs sFile Application.DisplayAlerts = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Programmabbruch bei Überspeichernabfrage verhindern

Problem:	Wenn ich eine Datei unter einem Namen speichern möchte, der schon vorhanden ist, fragt mich Excel, ob die Datei überschrieben werden soll. Beantworte ich diese Frage mit Ja, läuft das Programm anstandslos weiter, antworte ich mit nein, bricht es mit Laufzeitfehler 1004 ab. Wie kann ich das verhindern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Speichern() Dim sFile As String sFile = Range("A2").Value If Dir(sFile) <> "" Then If MsgBox(_ prompt:="Datei existiert, überschreiben?", _ Buttons:=vbQuestion + vbYesNo) = vbYes Then Application.DisplayAlerts = False ActiveWorkbook.SaveAs sFile Application.DisplayAlerts = True End If Else ActiveWorkbook.SaveAs sFile End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe unter Namen speichern

Problem:	In Zelle B3 habe ich den Vor-, in Zelle B4 den Nachnamen stehen. Wie kann ich die Datei unter der Namenssyntax "Bill Gates.xls" speichern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub UnterNamenSpeichern() Dim sFile As String, sPath As String sPath = Application.DefaultFilePath & "\" sFile = Range("B2").Value & "_" & _ Range("B1").Value & ".xls)" ActiveWorkbook.SaveAs sPath & sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe doppelt speichern

Problem:	Wie kann ich bei jedem Speichern einer Arbeitsmappe eine zusätzliche Kopie im Standardverzeichnis anlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Speichern() Dim sFile As String, sPath As String sPath = Application.DefaultFilePath & "\" With ActiveWorkbook sFile = .Name .SaveAs sPath & sFile .SaveAs .FullName End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mappe unter Zelldatum speichern

Problem:	Wie kann ich die aktive Arbeitsmappe unter dem Datum, daß in Zelle A1 des 1. Arbeitsblattes steht speichern? Der Dateiname soll das Format TTMM.XLS, also z.B. 0312.XLS haben.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Speichern() Dim sFile As String, sPath As String sPath = Application.DefaultFilePath & "\" sFile = Range("A1").Value sFile = Format(CDate(sFile), "yyyymmdd") & ".xls" ActiveWorkbook.SaveAs sPath & sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einzelnes Tabellenblatt speichern

Problem:	Wie kann ich ein einzelnes, zu benennendes Tabellenblatt unter einem zu benennenden Dateinamen speichern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BlattSpeichern() Dim sPath As String, swks As String, sFile As String Application.ScreenUpdating = False sPath = Application.DefaultFilePath & "\" swks = InputBox(_ prompt:="Blattname", _ Default:="Tabelle") If swks = "" Then Exit Sub sFile = InputBox(_ prompt:="Dateiname:", _ Default:="c:\test.xls") If sFile = "" Then Exit Sub ActiveSheet.Copy ActiveSheet.Name = swks ActiveWorkbook.SaveAs sPath & sFile Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe unter dem Text eines Zelleintrags speichern

Problem:	Wie kann ich die aktive Arbeitsmappe unter dem Namen eines Zelleintrags speichern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Speichern() ThisWorkbook.SaveAs _ FileName:=Range("A2").Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateien mit bestimmtem Inhalt suchen

Problem: Die in Spalte A genannten Verzeichnisse sollen gem. den Kriterien in Spalte D durchsucht und die gefundenen Dateien in einer neuen Arbeitsmappe mit Hyperlinks gelistet werden.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub GrepFiles()
    Dim wks As Worksheet
    Dim iCounter As Integer, iRow As Integer
    Set wks = ActiveSheet
    iRow = 2
    Workbooks.Add 1
    Do Until IsEmpty(wks.Cells(iRow, 1))
        With Application.FileSearch
            .NewSearch
            .LookIn = wks.Cells(iRow, 1).Value
            .Filename = wks.Range("D1").Value
            .MatchTextExactly = True
            .TextOrProperty = wks.Range("D2").Value
            .SearchSubFolders = wks.Range("D3").Value
            .Execute
            For iCounter = 1 To .FoundFiles.Count
                Cells(iCounter, 6).Value = .FoundFiles(iCounter)
                ActiveSheet.Hyperlinks.Add Cells(iCounter, 6), .FoundFiles(iCounter)
            Next iCounter
        End With
        iRow = iRow + 1
    Loop
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafikdateien suchen

Problem: Die in Spalte B benannten Dateien (die Suffix *.jpg wird automatisch angefügt) sollen in einem vorgegebenen Verzeichnis gesucht werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub DateienFinden()  
    Dim iRow As Integer, iRowL As Integer  
    Dim sPathS As String, sPathT As String  
    Dim sFileS As String, sFileT As String  
    sPathS = Range("B2").Value  
    sPathT = Range("C2").Value  
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row  
    For iRow = 2 To iRowL  
        sFileS = sPathS & "\" & Cells(iRow, 1).Value & ".jpg"  
        sFileT = sPathT & "\" & Cells(iRow, 1).Value & ".jpg"  
        If Dir(sFileS) = "" Then  
            Cells(iRow, 1).Interior.ColorIndex = 6  
        Else  
            Name sFileS As sFileT  
        End If  
    Next iRow  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateien mit Angabe des Dateidatums mit Hyperlink listen

Problem: In Spalte A sollen die Dateien aus dem in Zelle A1 genannten Verzeichnis einschl. aller Unterverzeichnissen mit Hyperlinks gelistet und in Spalte B soll das zugehörige Dateidatum eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub ScanDir()  
    Dim lRow As Long, lCounter As Long  
    Dim iCol As Integer  
    lRow = 1  
    iCol = 1  
    With Application.FileSearch  
        .LookIn = Range("A1").Value  
        .SearchSubFolders = True  
        .Execute  
        For lCounter = 1 To .FoundFiles.Count  
            lRow = lRow + 1  
            If lRow = 65537 Then  
                lRow = 2  
                iCol = iCol + 1  
            End If  
            Cells(lRow, iCol).Value = .FoundFiles(lCounter)  
            Cells(lRow, iCol).Hyperlinks.Add _  
                Cells(lRow, iCol), .FoundFiles(lCounter)  
            Cells(lRow, iCol + 1).Value = FileDateTime(.FoundFiles(lCounter))  
        Next lCounter  
    End With  
    Columns.AutoFit  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlinks zu allen Excel-Dateien eines Verzeichnisses erstellen

Problem:	Wie kann ich in der aktiven Tabelle in Spalte A Hyperlinks zu allen Excel-Dateien eines bestimmten Verzeichnisses erstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SetHyperlinks() Dim arr As Variant Dim iCounter As Integer ActiveSheet.Hyperlinks.Delete Range("A2:A65536").ClearContents With Application.FileSearch .LookIn = Range("B1").Value .FileType = msoFileTypeExcelWorkbooks .Execute For iCounter = 1 To .FoundFiles.Count Cells(iCounter + 1, 1).Value = .FoundFiles(iCounter) ActiveSheet.Hyperlinks.Add _ Anchor:=Cells(iCounter + 1, 1), _ Address:=.FoundFiles(iCounter) Next iCounter End With Columns(1).AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fehler abfangen, wenn eine Arbeitsmappe nicht geöffnet ist

Problem:	Wie kann ich eine Fehlermeldung abfangen, wenn ich eine Arbeitsmappe anspreche, die nicht geöffnet ist? In diesem Fall soll sie geöffnet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FehlerAbfangen() Dim wkb As Workbook Dim sFile As String sFile = ThisWorkbook.Path & "\test1.xls" On Error Resume Next Set wkb = Workbooks("Test.xls") If Err > 0 Or wkb Is Nothing Then If Dir(sFile) = "" Then MsgBox "Die Testdatei " & sFile & " existiert nicht!" Exit Sub Else Set wkb = Workbooks.Open(sFile) End If End If Application.Goto wkb.Worksheets(1).Range("D12") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verweise auf Zellen in geschlossener Arbeitsmappe erstellen

Problem:	Wie kann ich Daten aus einer geschlossenen Arbeitsmappe importieren, ohne sie zu öffnen. Pfad, Dateiname und Importbereich sollen variabel sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Eintragen() Dim sTxt As String Dim rng As Range, rngCell As Range sTxt = "" & Range("B1").Value sTxt = sTxt & "[" & Range("B2").Value & "]" sTxt = sTxt & Range("B3").Value & "!" sTxt = sTxt & Range("B4").Value Set rng = Worksheets("Ziel").Range(Range("B4").Value) rng.Cells(1).Formula = "=if(" & sTxt & "=""", "", " & sTxt & ")" rng.Cells(1).Copy rng rng.Value = rng.Value rng.Columns.AutoFit rng.Parent.Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei verschieben

Problem:	Wie kann ich Dateien - auch Nicht-Excel-Files - von einem Ordner zum anderen verschieben. Fehlermeldungen sollen abgefangen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DateiVerschieben() On Error Resume Next Name Range("B1").Value As Range("B2").Value If Err > 0 Then Select Case Err Case 53: MsgBox "Datei nicht vorhanden!" Case 58: MsgBox "Datei existiert bereits!" Case Else: MsgBox Error() End Select End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Serie von Dateien in das XL5/7-Format konvertieren

Problem:	Wie kann ich eine Serie von Dateien in das Dateiformat XL5.0/7.0 konvertieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Konvertieren() Dim iRow As Integer Application.ScreenUpdating = False Application.DisplayAlerts = False Application.EnableEvents = False On Error GoTo ERRORHANDLER iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) Workbooks.Open _ FileName:=Cells(iRow, 1).Value, _ updatelinks:=False With ActiveWorkbook .SaveAs _ FileName:=ActiveWorkbook.Name, _ FileFormat:=xlExcel5 .Close savechanges:=False End With iRow = iRow + 1 Loop Application.DisplayAlerts = True Application.ScreenUpdating = True ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

XL5/7-Dateien nach XL8 konvertieren

Problem: Wie kann ich alle XL5/7-Dateien eines Verzeichnisses in das XL8Format konvertieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CommandButton1_Click()
    Call Konvertieren
End Sub

StandardModule: basMain

Sub Konvertieren()
    Dim arr()
    Dim sName As String, iCount As Integer
    Dim iCounter As Integer
    sName = Dir("*.xls")
    Do While sName <> ""
        iCount = iCount + 1
        ReDim Preserve arr(1 To iCount)
        arr(iCount) = sName
        sName = Dir()
    Loop
    Application.DisplayAlerts = False
    For iCounter = 1 To iCount
        Workbooks.Open arr(iCounter)
        ActiveWorkbook.SaveAs _
            FileName:=arr(iCounter), _
            FileFormat:=xlNormal
        ActiveWorkbook.Close
    Next iCounter
    Application.DisplayAlerts = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nächsthöhere Ebene des Ordners mit der Arbeitsmappe ermitteln

Problem:	Der Ordner in der nächsthöheren Ebene des Verzeichnisses der Arbeitsmappe soll ermittelt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetFolder() MsgBox EbeneNachOben End Sub Function EbeneNachOben() Dim iChar As Integer Dim sPath As String sPath = ThisWorkbook.Path For iChar = Len(sPath) To 1 Step -1 If Mid(sPath, iChar, 1) = "\" Then Exit For Next iChar EbeneNachOben = "Obere Ebene: " & Left(sPath, iChar - 1) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ordneigenschaften auslesen

Problem:	In Spalte B werden die Eigenschaften des in Zelle B1 genannten Ordners eingetragen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ReadProperties() Dim Wscript As Application Dim objWMIService As Object Dim colFolders As Object Dim objFolder As Object Dim strComputer As String, sPath As String sPath = Range("B1").Value sPath = WorksheetFunction.Substitute(sPath, "\", "\\") strComputer = "." Set objWMIService = GetObject("winmgmts:" _ & "{impersonationLevel=impersonate}!\\\" & strComputer & "\root\cimv2") Set colFolders = objWMIService. _ ExecQuery("Select * from Win32_Directory where name = '" & sPath & "'") For Each objFolder In colFolders With objFolder Cells(3, 2).Value = .Archive Cells(4, 2).Value = .Caption Cells(5, 2).Value = .Compressed Cells(6, 2).Value = .CompressionMethod Cells(7, 2).Value = .CreationDate Cells(8, 2).Value = .Encrypted Cells(9, 2).Value = .EncryptionMethod Cells(10, 2).Value = .Hidden Cells(11, 2).Value = .InUseCount Cells(12, 2).Value = .LastAccessed Cells(13, 2).Value = .LastModified Cells(14, 2).Value = .Name Cells(15, 2).Value = .Path Cells(16, 2).Value = .Readable Cells(17, 2).Value = .System Cells(18, 2).Value = .Writeable End With Next End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateiliste mit allen xl-Dateien und Hyperlinks erstellen

Problem:	Zu dem in Zelle B1 genannten Laufwerk bzw. Pfad soll eine Liste aller Excel-Dateien mit Größe, Hyperlink und Aktualisierungsdatum erstellt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CreateFileList() Dim wks As Worksheet Dim iCounter As Integer, iCount As Integer Dim sSource As String, sTxt As String Application.ScreenUpdating = False Rows(5).Hidden = False sSource = Range("Source").Value If Len(sSource) = 1 Then sSource = sSource & ":\\" End If Workbooks.Add 1 Set wks = ActiveSheet ThisWorkbook.Activate Application.ScreenUpdating = True wks.Columns("A:B").NumberFormat = "@" wks.Columns("C").NumberFormat = "#,##0" wks.Range("A1").Value = "LfdNr." wks.Range("B1").Value = "Dateiname" wks.Range("C1").Value = "Dateigröße" wks.Range("D1").Value = "Dateidatum" With wks.Range("A1:D1") .Font.Bold = True .Interior.ColorIndex = 1 .Font.ColorIndex = 2 End With With Application.FileSearch .NewSearch .LookIn = sSource .Filename = "*.xl*" .SearchSubFolders = True .Execute iCount = .FoundFiles.Count For iCounter = 1 To iCount If iCounter Mod 100 = 0 Then Range("D5").Value = _ "Bearbeite Datei Nr. " & iCounter & "..." wks.Cells(iCounter + 1, 2).Value = .FoundFiles(iCounter) wks.Hyperlinks.Add _ anchor:=wks.Cells(iCounter + 1, 2), _ Address:=.FoundFiles(iCounter) wks.Cells(iCounter + 1, 3).Value = FileLen(.FoundFiles(iCounter)) wks.Cells(iCounter + 1, 4).Value = FileDateTime(.FoundFiles(iCounter)) Next iCounter End With Application.ScreenUpdating = False wks.Range("A1").CurrentRegion.Sort _ key1:=wks.Range("B2"), order1:=xlAscending, header:=xlYes For iCounter = 2 To iCount + 1 wks.Cells(iCounter, 1).Value = Format(iCounter - 1, "0000") Next iCounter Rows(5).Hidden = True wks.Columns.AutoFit wks.Columns("E:IV").Hidden = True wks.Rows(iCounter & ":" & Rows.Count).Hidden = True wks.Name = "Dateiliste" Windows(wks.Parent.Name).DisplayHeadings = False wks.Parent.Activate End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateiheaderinformationen abrufen

Problem: Die Dateiheaderinformationen der sich im in Zelle B1 genannten Verzeichnis befindlichen Dateien sollen aufgelistet werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub ExtendedInfos()
    Dim objShell As Object
    Dim objFolder As Object
    Dim iCounter As Integer, iRow As Integer, iCol As Integer
    Dim strFileName As Variant
    Dim arrHeaders(34)
    Application.ScreenUpdating = False
    Set objShell = CreateObject("Shell.Application")
    Set objFolder = objShell.Namespace(Range("B1").Value)
    iRow = 3
    iCol = 1
    For iCounter = 0 To 33
        arrHeaders(iCounter) = _
            objFolder.GetDetailsOf(objFolder.Items, iCounter)
    Next iCounter
    For Each strFileName In objFolder.Items
        For iCounter = 0 To 33
            Cells(iRow + iCounter, 1).Value = iCounter + 1
            Cells(iRow + iCounter, 2).Value = arrHeaders(iCounter)
            Cells(iRow + iCounter, 3).Value = _
                objFolder.GetDetailsOf(strFileName, iCounter)
        Next iCounter
        iRow = iRow + 35
    Next strFileName
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorhandensein von Verzeichnissen prüfen und Ordner anlegen

Problem: Es soll geprüft werden, ob ein Verzeichnis mit Unterverzeichnissen bereits besteht. Wenn ja, soll eine entsprechende Meldung erfolgen, wenn nein sollen sie erstellt werden.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code:

```
StandardModule: Modul1

Sub Pruefen()
    Dim arrOrdner As Variant
    Dim iOrdner As Integer
    Dim sDrive As String, sOrdner As String, sTmp As String
    sOrdner = InputBox("Zu erstellendes Verzeichnis:", , Range("B1").Value)
    If sOrdner = "" Then Exit Sub
    If Right(sOrdner, 1) = "\" Then
        sOrdner = Left(sOrdner, Len(sOrdner) - 1)
    End If
    arrOrdner = fncFolders(sOrdner)
    For iOrdner = UBound(arrOrdner) To 1 Step -1
        If fncIfFolderExists(CStr(arrOrdner(iOrdner))) Then
            MsgBox "Ordner " & arrOrdner(iOrdner) & " ist bereits vorhanden!"
        Else
            MkDir arrOrdner(iOrdner)
        End If
    Next iOrdner
End Sub

Private Function fncFolders(sFolder As String) As Variant
    Dim arr() As String
    Dim iCounter As Integer, iFolder As Integer
    ReDim arr(1 To 1)
    arr(1) = sFolder
    iFolder = 1
    For iCounter = Len(sFolder) To 4 Step -1
        If Mid(sFolder, iCounter, 1) = "\" Or iCounter = 1 Then
            iFolder = iFolder + 1
            ReDim Preserve arr(1 To iFolder)
            arr(iFolder) = Left(sFolder, iCounter - 1)
        End If
    Next iCounter
    fncFolders = arr
End Function

Private Function fncIfFolderExists(sFolder As String) As Boolean
    Dim sold As String
    sold = CurDir
    On Error Resume Next
    ChDrive Left(sFolder, 1)
    ChDir sFolder
    If Err = 0 Then fncIfFolderExists = True
    On Error GoTo 0
    ChDrive Left(sold, 1)
    ChDir sold
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verzeichnisse gemäß Tabelleneintragen anlegen

Problem:	Es sollen Verzeichnisse gemäß den Eintragungen in der Tabelle angelegt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetDir() Dim iRow As Integer, iCol As Integer Dim sDir As String, sVerz As String iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) iCol = 1 Do Until IsEmpty(Cells(iRow, iCol)) sVerz = Cells(iRow, iCol).Value sVerz = WorksheetFunction.Substitute(sVerz, "\", "") sDir = sDir & "\" & sVerz If Left(sDir, 1) = "\" Then sDir = Right(sDir, Len(sDir) - 1) On Error Resume Next If iCol > 1 Then MkDir sDir On Error GoTo 0 iCol = iCol + 1 Loop sDir = "" iRow = iRow + 1 Loop On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorkommen eines Wortes in Arbeitsmappen eines Verzeichnisse

Problem: Im jeweils ersten Tabellenblatt aller Arbeitsmappen in einem Ordner soll das Vorkommen des Wortes "Apfelsinensaft" gezählt und in diese Tabelle eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub EintraegeZaehlen()
    Dim arr As Variant
    Dim iCounter As Integer, iRow As Integer
    Dim sPath As String
    Application.ScreenUpdating = False
    Application.EnableEvents = True
    On Error GoTo ERRORHANDLER
    sPath = Range("B1").Value
    arr = FileArray(sPath, "*.xls")
    For iCounter = 1 To UBound(arr)
        Workbooks.Open sPath & arr(iCounter), False
        With ThisWorkbook.Worksheets("Import")
            iRow = iRow + 1
            .Cells(iRow, 1) = ActiveWorkbook.Name
            .Cells(iRow, 2) = WorksheetFunction.CountIf(Columns(1), _
                "Apfelsinensaft")
        End With
        ActiveWorkbook.Close savechanges:=False
    Next iCounter
ERRORHANDLER:
    Application.EnableEvents = True
    Application.ScreenUpdating = True
End Sub

Function FileArray(sPath As String, sPattern As String)
    Dim arrFiles()
    Dim iCounter As Integer
    Dim sFile As String
    If Right(sPath, 1) <> "\" Then sPath = sPath & "\"
    sFile = Dir(sPath & sPattern)
    Do While sFile <> ""
        iCounter = iCounter + 1
        ReDim Preserve arrFiles(1 To iCounter)
        arrFiles(iCounter) = sFile
        sFile = Dir()
    Loop
    FileArray = arrFiles
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

XL-Dateien aus Haupt- und Unterverzeichnissen einlesen

Problem:	Wie kann ich alle Excel-Dateien des aktuellen Verzeichnisses einschliesslich aller Unterverzeichnisse in ein Tabellenblatt einlesen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Public Const PROCESS_QUERY_INFORMATION = &H400 Public Const WAIT_TIMEOUT = &H102& Declare Function OpenProcess Lib "kernel32" (ByVal dwDesiredAccess As Long, _ ByVal bInheritHandle As Long, ByVal dwProcessId As Long) As Long Declare Function WaitForSingleObject Lib "kernel32" (ByVal hHandle As Long, _ ByVal dwMilliseconds As Long) As Long Sub DOSShell() Dim sFile As String, sPath As String sPath = Range("B1").Value sFile = Application.DefaultFilePath & "\dirlist.txt" WartenBisFertig ("command.com /c dir/s/b " & sPath & "*.xls >" & sFile) Workbooks.Open sFile Columns.AutoFit End Sub Sub WartenBisFertig(strEXE As String) Dim ProcessID As Long Dim hProcess As Long Dim RetVal As Long ProcessID = Shell(strEXE, vbHide) hProcess = OpenProcess(PROCESS_QUERY_INFORMATION, False, ProcessID) Do DoEvents RetVal = WaitForSingleObject(hProcess, 50) Loop Until RetVal <> WAIT_TIMEOUT End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateien in einem Verzeichnis und Unterverzeichnissen suchen

Problem:	Wie kann ich, ausgehend von einem Startverzeichnis, dieses und alle Unterverzeichnisse nach einem bestimmten Dateinamen durchsuchen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SearchFile() Dim iCounter As Integer With Application.FileSearch .FileName = Range("B1").Value .LookIn = Range("B2").Value .SearchSubFolders = True .Execute If .FoundFiles.Count = 0 Then Beep MsgBox "Datei wurde nicht gefunden!" Else For iCounter = 1 To .FoundFiles.Count MsgBox .FoundFiles(iCounter) Next iCounter End If End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verzeichniswechsel über VBA

Problem: Wie kann ich in das Verzeichnis dieser Datei wechseln, den GetOpenFileName-Dialog aufrufen und wieder in das alte Verzeichnis zurückkehren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub VerzeichnisWechsel()  
    Dim vFile As Variant  
    Dim sDirOld As String, sDirNew As String  
    sDirOld = CurDir()  
    sDirNew = ThisWorkbook.Path  
    If sDirNew = "" Then  
        MsgBox "Die Datei muß zuerst gespeichert werden!"  
        Exit Sub  
    End If  
    MsgBox "Aktuelles Verzeichnis: " & CurDir  
    ChDrive Left(sDirNew, 1)  
    ChDir sDirNew  
    vFile = Application.GetSaveAsFilename(, "Excel-Datei (*.xls),*.xls")  
    If vFile = False Then  
        MsgBox "Keine Datei ausgewählt!"  
    Else  
        MsgBox "Ausgewählte Datei: " & vFile  
    End If  
    MsgBox "Aktuelles Verzeichnis: " & CurDir  
    ChDrive Left(sDirOld, 1)  
    ChDir sDirOld  
    MsgBox "Aktuelles Verzeichnis: " & CurDir  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappen- und Tabellenblattnamen auslesen

Problem: Wie kann ich alle Arbeitsmappen- und die zugehörigen Tabellenblattnamen des aktuellen Verzeichnisses über eine MsgBox anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub TabellenAuslesen()
    Dim wks As Worksheet
    Dim iCounter As Integer
    Application.ScreenUpdating = False
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    With Application.FileSearch
        .LookIn = Range("B1").Value
        .FileType = msoFileTypeExcelWorkbooks
        .Execute
        For iCounter = 1 To .FoundFiles.Count
            Workbooks.Open .FoundFiles(iCounter), False
            For Each wks In Worksheets
                MsgBox _
                    prompt:=wks.Name, _
                    Title:=ActiveWorkbook.Name
            Next wks
            ActiveWorkbook.Close savechanges:=False
        Next iCounter
    End With
ERRORHANDLER:
    Application.EnableEvents = True
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappen eines Verzeichnisses ändern

Problem:	Wie kann ich in allen Arbeitsmappen eines vorgegebenen Verzeichnisses im aktiven Blatt die Zelle A1 ändern, die Mappen speichern und wieder schließen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ChangeCellA1() Dim iCounter As Integer Application.ScreenUpdating = False Application.EnableEvents = False On Error GoTo ERRORHANDLER With Application.FileSearch .LookIn = Range("B1").Value .FileType = msoFileTypeExcelWorkbooks .Execute For iCounter = 1 To .FoundFiles.Count Workbooks.Open .FoundFiles(iCounter), False Range("A1").Value = "Tabelle " & Year(Date) ActiveWorkbook.Close savechanges:=True Next iCounter End With ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateien einer Ordnerliste in einer Tabelle zusammenstellen

Problem: Wie kann ich mir eine Liste aller Dateien erstellen lassen, deren Verzeichnisse in Spalte A einer Tabelle stehen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub DateiListe()  
    Dim wks As Worksheet  
    Dim iRow As Integer, iCounter As Integer, iRowT As Integer  
    Application.ScreenUpdating = False  
    Set wks = ActiveSheet  
    Workbooks.Add 1  
    iRow = 1  
    Do Until IsEmpty(wks.Cells(iRow, 1))  
        Cells(1, iRow).Value = wks.Cells(iRow, 1).Value  
        iRowT = 1  
        With Application.FileSearch  
            .NewSearch  
            .LookIn = wks.Cells(iRow, 1).Value  
            .Execute  
            For iCounter = 1 To .FoundFiles.Count  
                iRowT = iRowT + 1  
                Cells(iRowT, iRow).Value = _  
                    .FoundFiles(iCounter)  
            Next iCounter  
        End With  
        iRow = iRow + 1  
    Loop  
    Rows(1).Font.Bold = True  
    Columns.AutoFit  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

xlSaveAs-Dialog in einem bestimmten Verzeichnis aufrufen

Problem:	Wie kann ich den xlSaveAs-Dialog so aufrufen, daß er die Dateien eines bestimmten Verzeichnisses zeigt? Danach muß das Ursprungsverzeichnis wieder das aktuelle sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Speichern() Dim sDirOld As String, sDirNew As String On Error GoTo ERRORHANDLER sDirOld = CurDir sDirNew = Range("B1").Value ChDrive Left(sDirNew, 1) ChDir sDirNew Application.Dialogs(xlDialogSaveAs).Show ChDrive Left(sDirOld, 1) ChDir sDirOld Exit Sub ERRORHANDLER: MsgBox "Verzeichnis wurde nicht gefunden!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Feststellen, ob eine Datei in einem Verzeichnis existiert

Problem:	Wie kann ich über VBA feststellen, ob eine bestimmte Datei in einem vorgegebenen Verzeichnis existiert?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DateiVorhanden() Dim sFile As String sFile = Range("B1").Value If Dir(sFile) <> "" Then MsgBox sFile & " ist vorhanden!" Else MsgBox sFile & " ist nicht vorhanden!" End If End Sub</pre>



Ordner eines Verzeichnisses in ListBox übernehmen

Problem:	Wie kann ich alle Ordner eines anzugebenden Verzeichnisses in eine UserForm-ComboBox und zugehörige Dateien in eine ListBox übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Sub DialogAufruf() frmFiles.Show End Sub ClassModule: frmFiles Private Sub cboFolders_Change() Dim fs As FileSearch Dim iCounter As Integer lstFiles.Clear Set fs = Application.FileSearch With fs .NewSearch .LookIn = Range("B1").Value & cboFolders.Value .FileType = msoFileTypeExcelWorkbooks .Execute For iCounter = 1 To .FoundFiles.Count lstFiles.AddItem .FoundFiles(iCounter) Next iCounter End With End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() End Sub Private Sub cmdOpen_Click() If lstFiles.Value <> "" Then Workbooks.Open lstFiles.Value End If End Sub Private Sub UserForm_Initialize() Dim iCounter As Integer Dim sDir As String, sFile As String If Right(Range("B1").Value, 1) <> "\" Then Range("B1").Value = Range("B1").Value & "\" End If sDir = Range("B1").Value Me.Caption = "Start: " & sDir sFile = Dir(sDir & ".*", vbDirectory) Do While sFile <> "" If sFile = "." Or sFile = ".." Then ElseIf (GetAttr(sDir & sFile) And _ vbDirectory) = vbDirectory Then cboFolders.AddItem sFile End If sFile = Dir Loop With Application.FileSearch .LookIn = sDir .FileType = msoFileTypeExcelWorkbooks .Execute For iCounter = 1 To .FoundFiles.Count lstFiles.AddItem .FoundFiles(iCounter) Next iCounter End With End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Excel-Dateien eines Ordners listen

Problem:	Wie kann ich alle Excel-Dateien eines bestimmten Unterverzeichnisses in einem Tabellenblatt auflisten?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Einlesen() Dim fs As FileSearch Dim iCounter As Integer Set fs = Application.FileSearch With fs .FileType = msoFileTypeExcelWorkbooks .LookIn = CurDir .Execute For iCounter = 1 To .FoundFiles.Count Cells(iCounter, 1).Value = .FoundFiles(iCounter) Next iCounter End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verzeichnisbaum anlegen

Problem: Über den Mkdir-Befehl kann ich ein Unterverzeichnis anlegen. Wie lege ich aber einen kompletten Verzeichnisbaum an? Das komplette Verzeichnis soll über ein Inputbox angegeben werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub MultiMkdir()  
    Dim iCounter  
    Dim sPath As String, sTxt As String  
    sPath = InputBox("Bitte Pfad angeben:", , "c:\mypath")  
    If Right(sPath, 1) <> "\" Then sPath = sPath & "\"  
    iCounter = 4  
    Do While InStr(iCounter, sPath, "\") > 0  
        sTxt = Left(sPath, InStr(iCounter, sPath, "\") - 1)  
        iCounter = InStr(iCounter, sPath, "\") + 1  
        On Error GoTo ErrorHandler  
        Mkdir sTxt  
    Loop  
    Exit Sub  
ErrorHandler:  
    If Err = 75 Then  
        Beep  
        MsgBox "Verzeichnis besteht schon!"  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateien listen

Problem: Wie kann ich alle Dateien eines Verzeichnisses in ein Listefeld einlesen und bei Doppelklick die jeweilige Datei öffnen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle2

```
Private Sub cmdGetList_Click()  
    Dim fs As FileSearch  
    Dim iCounter As Integer  
    Dim sPath As String  
    sPath = ThisWorkbook.Path  
    Set fs = Application.FileSearch  
    lstFiles.Clear  
    With fs  
        .LookIn = sPath  
        .FileType = msoFileTypeExcelWorkbooks  
        .Execute  
        For iCounter = 1 To .FoundFiles.Count  
            lstFiles.AddItem .FoundFiles(iCounter)  
        Next iCounter  
    End With  
End Sub  
  
Private Sub lstFiles_DblClick( _  
    ByVal Cancel As MSForms.ReturnBoolean)  
    Workbooks.Open lstFiles.Value  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateidatum manipulieren

Problem:	Wie kann ich das Dateidatum auf den in Zelle A1 stehenden Wert setzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FileDatum() ThisWorkbook.BuiltinDocumentProperties(11) = _ Range("A1").Value MsgBox "Dateidatum:" & vbCrLf & _ ThisWorkbook.BuiltinDocumentProperties(11) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neueste Textdatei eines Verzeichnisses suchen und umbenennen

Problem: Wie kann ich in einem vorgegebenen Verzeichnis die neueste Textdatei suchen und diese umbenennen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub RenameTextFile()  
    Dim dat As Date  
    Dim iCounter As Integer  
    Dim sFile As String, sPath As String  
    sPath = Range("B1").Value  
    If Right(sPath, 1) <> "\" Then  
        sPath = sPath & "\"  
    End If  
    With Application.FileSearch  
        .LookIn = Left(sPath, Len(sPath) - 1)  
        .FileName = "*.txt"  
        .SearchSubFolders = False  
        .Execute  
        For iCounter = 1 To .FoundFiles.Count  
            If FileDateTime(.FoundFiles(iCounter)) > dat Then  
                sFile = .FoundFiles(iCounter)  
            End If  
        Next iCounter  
    End With  
    If sFile <> "" Then  
        If Dir(sPath & "letzte.txt") <> "" Then  
            Kill sPath & "letzte.txt"  
        End If  
        Name sFile As sPath & "letzte.txt"  
        MsgBox "Die Datei" & vbCrLf & _  
            sFile & vbCrLf & " wurde umbenannt in " & vbCrLf & _  
            sPath & "letzte.txt"  
    Else  
        MsgBox "Es wurde keine Textdatei gefunden!"  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe 10-minütlich als *.xls und *.bak-Datei speichern

Problem: Wie kann ich eine Arbeitsmappe im Minutenrhythmus als ExcelArbeitsmappe und als *.bak-Datei speichern?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub SpeichernMinuetlich()
    Dim NextTime As Date
    Dim sPathB As String, sPathW As String, sPath As String
    Application.DisplayAlerts = False
    sPath = Range("B1").Value
    If Right(sPath, 1) <> "\" Then
        sPath = sPath & "\"
    End If
    sPathW = sPath & Format(Date, "yymmdd") & ".xls"
    sPathB = sPath & Format(Now, "yymmddhhmm") & ".bak"
    ActiveWorkbook.SaveAs sPathB
    ActiveWorkbook.SaveAs sPathW
    Range("B2").Value = Now + TimeValue("00:01:00")
    Application.OnTime CDate(Range("B2").Value), "SpeichernMinuetlich"
    Application.DisplayAlerts = True
End Sub

Sub StopSpeichern()
    If IsEmpty(Range("B2")) Then Exit Sub
    Application.OnTime _
        earliesttime:=CDate(Range("B2").Value), _
        procedure:="SpeichernMinuetlich", _
        schedule:=False
    Range("B2").ClearContents
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Dateien ab einem bestimmten Datum listen

Problem:	Wie kann ich in einem Tabellenblatt alle Dateien auflisten, deren letztes Änderungsdatum nach einem anzugebenden Datum liegt?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DateiListe() Dim fs As FileSearch Dim iCounter As Integer Dim sPath As String sPath = InputBox(_ prompt:="Verzeichnis:", _ Default:=CurDir) If sPath = "" Then Exit Sub Set fs = Application.FileSearch With fs .LookIn = sPath .FileType = msoFileTypeExcelWorkbooks .LastModified = msoLastModifiedThisMonth .Execute For iCounter = 1 To .FoundFiles.Count Cells(iCounter, 1).Value = .FoundFiles(iCounter) Next iCounter End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappen-History erstellen

Problem:	Wie kann ich Pfad und Namen aller geöffneten Excel-Arbeitsmappe in einer Tabelle auflisten?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DateienListen() Dim wkb As Workbook Dim iRow As Integer For Each wkb In Workbooks iRow = iRow + 1 Cells(iRow, 1).Value = wkb.Path Cells(iRow, 2).Value = wkb.Name Next wkb End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe mit dem Namen des aktuellen Datums öffnen

Problem:	Wie kann ich über ein VBA-Makro eine Arbeitsmappe öffnen, die nach dem aktuellen Datum im Datumsformat "JJJJMMTT", also z.B. "19980512.xls" benannt ist?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DateFileOpen() Dim sName As String sName = Format(Date, "yyyymmdd") & ".xls" If Dir(sName) = "" Then Beep MsgBox "Arbeitsmappe " & sName & " nicht gefunden!" Exit Sub Else Workbooks.Open sName End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Letztes Speicherdatum eintragen

Problem: Wie kann ich das letzte Speicherdatum der aktiven Arbeitsmappe in eine Zelle eintragen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub FileDate()
    ActiveSheet.Range("A1") = _
        ThisWorkbook.BuiltinDocumentProperties(12)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung von Mitarbeiter-Zeiten für einen Zeitraum

Problem:	In Zelle I3 sollen die im Bereich I1:I2 genannten Kontrollzeitraum angefallenen Zeiten ermittelt werden.
Lösung:	Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktuellen Arbeitstag in allen Arbeitsblättern suchen.

Problem:	In allen Arbeitsblättern soll der aktuelle Arbeitstag (ohne Berücksichtigung von Samstagen und Sonntagen) gesucht werden. Ist der aktuelle Tag ein Samstag oder Sonntag, wird der Montag gesucht.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub FindWorkDay() Dim wks As Worksheet Dim dat As Date Dim vRow As Variant dat = Date Call MyWorkDay(dat) For Each wks In Worksheets vRow = Application.Match(CDbl(dat), wks.Columns(1), 0) If Not IsError(vRow) Then Exit For Next wks If Not IsError(vRow) Then Application.Goto wks.Cells(vRow, 1), True Else MsgBox "Datum wurde nicht gefunden!" End If End Sub Sub MyWorkDay(ByRef dat As Date) Dim iDay As Integer iDay = WorksheetFunction.Weekday(dat) Select Case iDay Case 7: dat = dat + 2 Case 1: dat = dat + 1 End Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung von Sonn-/Feiertagsstunden

Problem:	In einem dynamischen Monats-Arbeitsplan soll die Anzahl der Sonn- und Feiertagsstunden ermittelt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolgen in Industriezeiten umwandeln

Problem: Die Zeichenfolgen aus Spalte B sollen in Industriezeiten umgewandelt werden. In Spalte D eine Lösung aufgrund von Excel-Funktionen, in Spalte E aufgrund einer benutzerdefinierten Funktion, die auch Sekunden mit berücksichtigt.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: Modul1

```
Function Txt2No(sTxt As String) As Double
    If InStr(sTxt, "Stunde") Then
        Txt2No = CDb(Left(sTxt, InStr(sTxt, " ") - 1))
    ElseIf InStr(sTxt, "Minute") Then
        Txt2No = CDb(Left(sTxt, InStr(sTxt, " ") - 1)) / 60
    Else
        Txt2No = CDb(Left(sTxt, InStr(sTxt, " ") - 1)) / (60 * 60)
    End If
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wieviele Mitarbeiter sind zu welcher Stunde anwesend?

Problem: Über eine benutzerdefinierte Funktion soll ermittelt werden, zu welcher Stunde wieviel Mitarbeiter anwesend sind.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code: StandardModule: basFunctions

```
Function MA(rngAll As Range, iHour As Integer)
    Dim rng As Range
    Dim iCount As Integer
    For Each rng In rngAll.Cells
        If rng.Interior.ColorIndex = 6 And _
            Hour(Cells(1, rng.Column).Value) = iHour And _
            rng.Offset(0, 1).Interior.ColorIndex = 6 Then
            If Hour(Cells(1, rng.Column - 1).Value) <> iHour Or _
                rng.Offset(0, -1).Interior.ColorIndex <> 6 Then
                iCount = iCount + 1
            End If
        End If
    Next rng
    MA = iCount
End Function

Function MA1(rngAll As Range, iHour As Integer)
    Dim iRow As Integer, iCol As Integer, iCount As Integer
    iCol = WorksheetFunction.CountA(Rows(1)) - 1
    For iRow = 2 To rngAll.Rows.Count
        If iHour / 24 >= Cells(iRow, iCol).Value And _
            iHour / 24 < Cells(iRow, iCol + 1).Value Then
            iCount = iCount + 1
        End If
    Next iRow
    MA1 = iCount
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitszeiberechnung unter Berücksichtigung der Regelarbeitszeit

Problem:	Über die Formel in Spalte D soll unter Berücksichtigung des Arbeitsbeginns, der Pausen- und der Regelarbeitszeit das Arbeitsende bestimmt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Nachtstunden ermitteln und Lohn errechnen

Problem:	Die Arbeitszeit soll gem. den Daten in Spalte J in Normal- und Nachtstunden gesplittet werden. Der Lohn ist auf dieser Grundlage zu ermitteln.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Type PeriodStartStop Start As Double Stop_ As Double End Type Function ShiftHours(InTime As Date, OutTime As Date, _ Optional ShiftStart As Date, _ Optional ShiftEnd As Date) As Double Dim i As Long Dim j As Long Dim N As Double Dim Shift() As PeriodStartStop Dim Worked() As PeriodStartStop 'if worked hours don't span midnight, there's just one period; if do 'span midnight, will break into 2 periods, before and after midnight GetPeriods CDbl(InTime), CDbl(OutTime), Worked() 'ditto for the shift GetPeriods CDbl(ShiftStart), CDbl(ShiftEnd), Shift() 'get overlap of each worked period with each shift period N = 0 For i = 0 To UBound(Worked()) For j = 0 To UBound(Shift()) N = N + GetOverlap(Worked(i), Shift(j)) Next j Next i ShiftHours = Round(N * 24, 5) End Function Private Sub GetPeriods(ByVal t1 As Double, ByVal t2 As Double, _ Period() As PeriodStartStop) 'NB: t1 and t2 are time values 0 <= t < 1, not hours t1 = t1 - Int(t1) t2 = t2 - Int(t2) 'does period span midnight? If t1 <= t2 Then 'no: just one period ReDim Period(0 To 0) Period(0).Start = t1 Period(0).Stop_ = t2 Else 'yes: split into 2 periods: t1 to midnight, midnight to t2 'note: in 1st period midnight = 1, in 2nd period, midnight = 0 ReDim Period(0 To 1) Period(0).Start = t1 Period(0).Stop_ = 1 Period(1).Start = 0 Period(1).Stop_ = t2 End If End Sub Private Function GetOverlap(Period1 As PeriodStartStop, _ Period2 As PeriodStartStop) As Double Dim t1 As Double Dim t2 As Double 'NB: Stop_ must never be less than Start, which is 'ensured by splitting periods that span midnight 'into 2 periods 'overlap, if any, is from later start to earlier end 'get the later of the start times If Period1.Start >= Period2.Start Then t1 = Period1.Start Else t1 = Period2.Start End If 'get the earlier of the Stop_ times If Period1.Stop_ <= Period2.Stop_ Then t2 = Period1.Stop_ Else t2 = Period2.Stop_ End If 'subtract start from stop to get length of interval 'if result is positive, this is length of overlap 'if result = 0, one period starts when the other ends 'if result < 0, the periods don't overlap; result ' is the size of the gap between periods; 'since we are only interested in overlap, 'change negative result to 0 t2 = t2 - t1 If t2 < 0 Then t2 = 0 GetOverlap = t2 End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeitdifferenzen auf Viertelstunden ab- und aufrunden

Problem:	Die Zeitdifferenz aus Zelle C2 soll in D2 bei Restwerten unter einer Viertelstunde auf die nächste halbe Stunde ab-, sonst aufgerundet werden
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nettoarbeitstage unter Urlaubsberücksichtigung bestimmen

Problem:	Unter Berücksichtigung der Im Blattt "Urlaub" befindlichen Eintragungen sollen unter Einsatz der NETTOARBEITSTAGE-Formel die tatsächlichen Arbeitstage bestimmt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Splittung der Arbeitszeit in Normal- und Überstunden

Problem:	Arbeitszeit soll unter Berücksichtigung von Pause und Regelarbeitszeit in Normalarbeitszeit und Überstunden gesplittet werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Lohnberechnung mit Pausenberücksichtigung

Problem:	Eine Arbeitszeitberechnung auf Formelbasis unter Berücksichtigung von Arbeitszonen und Pausen.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Application.EnableEvents = False On Error GoTo ERRORHANDLER If Target.Name.Name = "BEGINN" Then If Target.Value >= Range("P1A").Value And Target.Value <= Range("P1E").Value Then Target.Value = Range("P1E").Value End If If Target.Value >= Range("P2A").Value And Target.Value <= Range("P2E").Value Then Target.Value = Range("P2E").Value End If ElseIf Target.Name.Name = "ENDE" Then If Target.Value >= Range("P1A").Value And Target.Value <= Range("P1E").Value Then Target.Value = Range("P1A").Value End If If Target.Value >= Range("P2A").Value And Target.Value <= Range("P2E").Value Then Target.Value = Range("P2A").Value End If End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Lohnberechnung nach Anfangs- und Ende-Zeit

Problem:	Einfache Lohnberechnung unter Berücksichtigung von Start- und Endezeit sowie dem Stundensatz.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einen Trainingsplan dynamisch gestalten

Problem:	Bei Auswahl eines Jahres oder Monats sollen nur die Trainingstage Dienstag und Donnerstag erscheinen. Lösung über benannte Formeln.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>StandardModule: Modul1 Sub NamesVisible() Dim nme As Name For Each nme In ThisWorkbook.Names nme.Visible = True Next nme End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Darstellung von Minuszeiten ohne Umstellung der 1904-Datumswerte

Problem:	In Spalte D soll die Abweichung im Zeitformat mit Minuszeichen dargestellt werden, ohne dass eine Umschaltung auf 1904-Datumswerte erfolgt. Lösung über die benannte Formel DIFF.

Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gleitzeitberechnung

Problem:	Die Gleitzeit soll unter Berücksichtigung von Anfang, Ende und Standardarbeitszeit ermittelt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Pausenberücksichtigung bei Arbeitszeitberechnungen

Problem:	Wenn A1 kleiner als 06:00 soll A1, wenn größer als 06:30 soll A1 - 00:30, wenn größer als 9 soll A1 - 00:45, sonst soll 06:00 eingetragen werden.
Lösung:	Folgende Formel: =WENN(A1*24<6;A1;WENN(A1*24>9;(A1*24-0,75)/24;WENN(A1*24>6,5;(A1*24-0,5)/24;6/24)))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Urlaub gem. Tabelle in Urlaubsliste eintragen

Problem: In einer Tabelle habe ich Mitarbeiternamen, Urlaubsbeginn und Urlaubsende stehen. Wie kann ich die Urlaubstage in den nachfolgenden Monatsblättern bei den einzelnen Mitarbeitern farblich markieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub UrlaubsEintrag()  
    Dim rng As Range  
    Dim iRow As Integer, iMonth As Integer, iCounter As Integer  
    iRow = 3  
    Do Until IsEmpty(Cells(iRow, 1))  
        For iMonth = Month(Cells(iRow, 2)) To Month(Cells(iRow, 3))  
            Set rng = Worksheets(Format(DateSerial(1, iMonth, 1), "mmm")). _  
                Columns(1).Find _  
                (Cells(iRow, 1), LookIn:=xlValues, lookat:=xlWhole)  
            If iMonth = Month(Cells(iRow, 2)) And iMonth = _  
                Month(Cells(iRow, 3)) Then  
                For iCounter = Day(Cells(iRow, 2)) To Day(Cells(iRow, 3))  
                    rng.Offset(0, iCounter).Interior.ColorIndex = 3  
                Next iCounter  
            ElseIf iMonth = Month(Cells(iRow, 2)) Then  
                For iCounter = Day(Cells(iRow, 2)) To Day(DateSerial _  
                    (1, Month(Cells(iRow, 2)) + 1, 0))  
                    rng.Offset(0, iCounter).Interior.ColorIndex = 3  
                Next iCounter  
            Else  
                For iCounter = 1 To Day(Cells(iRow, 3))  
                    rng.Offset(0, iCounter).Interior.ColorIndex = 3  
                Next iCounter  
            End If  
        Next iMonth  
        iRow = iRow + 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamischer Monatsarbeitsplan

Problem:	Wie kann ich einen dynamischen Monatsarbeitsplan erstellen. Nach Eingabe des Jahres oder des Monats soll sich die Tagesliste automatisch generieren. Wochenende und Feiertage - auch die beweglichen - sollen markiert werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

3-Schicht-Arbeitsplan gem. Vorgabedaten erstellen

Problem: Wie kann ich einen 3-Schicht-Arbeitsplan unter Berücksichtigung variabler freier Tage und Markierung der Wochenenden erstellen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub ArbeitsPlan()  
    Dim datStart As Date  
    Dim datEnd As Date  
    Dim iFrei As Integer, iRow As Integer, iAct As Integer  
    Dim lDay As Long  
    datStart = Range("B1").Value  
    datEnd = Range("B2").Value  
    iFrei = Range("B3").Value  
    Workbooks.Add 1  
    Columns(1).NumberFormat = "dd.mm.yy"  
    Columns(2).NumberFormat = "dddd"  
    For lDay = datStart To datEnd  
        iRow = iRow + 1  
        Cells(iRow, 1) = datStart + iRow - 1  
        Cells(iRow, 2) = datStart + iRow - 1  
        If Application.WeekDay(datStart + iRow - 1, 2) > 5 Then  
            Cells(iRow, 1).Interior.ColorIndex = 6  
            Cells(iRow, 2).Interior.ColorIndex = 6  
        End If  
        iAct = iAct + 1  
        If iAct = 21 + iFrei * 3 Then iAct = 1  
        If iAct < 8 Then  
            Cells(iRow, 3) = "Früh"  
        ElseIf iAct < 8 + iFrei Then  
            Cells(iRow, 3) = "iFrei"  
        ElseIf iAct < 14 + iFrei Then  
            Cells(iRow, 3) = "Mittag"  
        ElseIf iAct < 14 + iFrei * 2 Then  
            Cells(iRow, 3) = "iFrei"  
        ElseIf iAct < 21 + iFrei * 2 Then  
            Cells(iRow, 3) = "Nacht"  
        ElseIf iAct < 21 + iFrei * 3 Then  
            Cells(iRow, 3) = "iFrei"  
        End If  
    Next lDay  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitszeiten einer Personal-Nummer und eines Tages ermitteln

Problem:	Wie kann ich aus einer Monatsarbeitstabelle die Zeiten einer bestimmten Personalnummer an einem bestimmten Tag ermitteln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Berechnung der Wochenarbeitsstunden

Problem:	Wie kann ich aus einer Tabelle die Wochenarbeitsstunden unter Berücksichtigung von Pausen und einer Regelarbeitszeit berechnen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitszeit unter Berücksichtigung von Pause und Regelarbeitszeit

Problem:	Wie kann ich mir in einer Tabelle bei Eingabe der Arbeitsbeginnund Arbeitsendezeit automatisch das Plus oder Minus der Arbeitszeit zur Regelarbeitszeit anzeigen lassen. Die Mittagspause soll automatisch berücksichtigt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verbleibende Nettoarbeitstage des Monats berechnen

Problem:	Wie kann ich die Nettoarbeitstage, die verbleibenden Nettoarbeitstage ab einem bestimmten Tag und das Verhältnis der beiden Werte zueinander berechnen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zuschlagszeit aus einer Gesamtzeit filtern

Problem:	Wie kann ich aus einer durch Start- und Endzeit definierten Gesamtzeit eine zuschlagspflichtige Zeit filtern?
Lösung:	Folgende Formel: =WENN(B2>D2;D2+1;B2+1)-WENN(A2<C2;C2;A2)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Stempelkarten auswerten

Problem: Wie kann ich die Tagesarbeitszeit unter Zugrundelegung einer Stempelkartenauswertung berechnen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Column <> 4 Or Target.Row = 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Target.Cells.Count > 1 Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    If Target.Value < Target.Offset(-1, 0).Value And _
        Target.Row > 2 Then
        Beep
        MsgBox "Die Werte müssen kumuliert sein, der" & vbCrLf & _
            "Wert in der aktuellen Zelle darf also" & vbCrLf & _
            "nicht kleiner als der Wert in er Zelle darüber sein."
        Target.ClearContents
        GoTo ERRORHANDLER
    End If
    If Target.Row = 2 Then
        Target.Offset(0, -1).Value = Target.Value / 24
    Else
        Target.Offset(0, -1).Value = _
            (Target.Value - Target.Offset(-1, 0).Value) / 24
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorgegebenes Datum plus 33 1/3 Jahr errechnen

Problem: Es soll das Datum aus Zelle A2 plus 33 1/3 Jahr errechnet werden.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus einem String mit Datumsangabe das Jahr auslesen

Problem:	Aus dem ein Datum enthaltenden String soll das Jahr ausgefiltert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Function GebJahr(txt As String) As Integer Dim intCounter As Integer txt = txt & " " For intCounter = 1 To Len(txt) - 4 If Mid(txt, intCounter, 5) Like "*#### *" Then GebJahr = Mid(txt, intCounter, 4) Exit Function End If Next intCounter End Function Sub Aufruf() Dim x As Integer x = GebJahr("Freiburg am 22. Juli 1955") MsgBox x End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Zeilen löschen, in denen das Datum < Heute ist

Problem:	Wie kann ich alle Zeilen löschen, wenn das kleiner als Heute ist?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub PruefenLoeschen() Dim iCounter As Integer, iRow As Integer iRow = Cells(Rows.Count, 1).End(xlUp).Row For iCounter = iRow To 1 Step -1 If Not IsEmpty(Cells(iCounter, 1)) And _ CDb1(Cells(iCounter, 1).Value) < CDb1(Date) Then Rows(iCounter).Delete End If Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fixpunkt in einem genannten Zeitraum

Problem:	Ich nehme Zeiteinträge in der Form "11:12 bis 16:15" vor. In einer anderen Zelle soll immer dann, wenn der Fixpunkt 12:00 im angegebenen Zeitraum enthalten ist, ein "X" angezeigt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Monatsabhängige Tagesliste anlegen und Wochenenden markieren

Problem:	Wie kann ich jahres- und monatsabhängig eine Tagesliste anlegen, die sich bei Eingaben des Jahres oder des Monats automatisch aktualisiert und die Samstage und Sonntage mit einer unterschiedlichen Hintergrundfarbe markiert?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Monatstage eines vorgegebenen Zeitraum listen und addieren

Problem: Wie kann ich in einer Tabelle die Monatstage, ausgehend von einem Start- bis zu einem Enddatum, mit der jeweiligen Monatsbezeichnung auflisten und addieren lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Tage()  
    Dim lCounter As Long  
    Dim iRow As Integer, iDays As Integer  
    Columns("D:E").ClearContents  
    lCounter = Range("B1")  
    Do  
        iDays = iDays + 1  
        If Month(lCounter) <> Month(lCounter + 1) Then  
            iRow = iRow + 1  
            Cells(iRow, 4) = Format(lCounter, "mmm")  
            Cells(iRow, 5) = iDays  
            iDays = 0  
        End If  
        lCounter = lCounter + 1  
    Loop Until lCounter > Range("B2").Value  
    iRow = iRow + 1  
    Cells(iRow, 4) = Format(Range("B2"), "mmm")  
    Cells(iRow, 5) = Day(Range("B2"))  
    Cells(iRow + 1, 5) = Application.Sum(Range("E1:E" & iRow))  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Start- und Enddatum abfragen und markieren

Problem:	Wie kann ich ein Start- und Enddatum per InputBox abfragen und die beiden Daten dann farblich markieren lassen?
Lösung:	Formel für die bedingte Formatierung: =UND(A1>=\$D\$1;A1<=\$D\$2)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datum suchen und Wert eintragen

Problem:	Wie kann ich in Spalte 1 von Tabelle2 ein Datum finden, das in Zelle A1 von Tabelle1 hinterlegt ist? In die Zelle rechts neben dem gefundenen Datum soll ein Wert eingetragen werden. Ist die Zelle jedoch nicht leer, soll eine Sicherheitsabfrage erfolgen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DatumSuchen() Dim var As Variant With Worksheets("Tabelle2") var = Application.Match(CDbl(Range("A1").Value), .Columns(1), 0) If Not IsError(var) Then If Not IsEmpty(.Cells(var, 2)) Then If MsgBox("Nicht leer, trotzdem eintragen?", _ vbCritical + vbYesNo) = vbNo Then Exit Sub End If End If .Cells(var, 2).Value = "Hallo!" Application.Goto .Cells(var, 1), True End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Stringeingaben von HHMM und HHMMSS in Zeit wandeln

Problem: Wie kann ich wahlweise Texteingaben in eine Zelle im Format HHMM oder HHMMSS in die entsprechenden Zeiten umwandeln lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim sTxt As String
    If Target.Column < 5 Or Target.Column > 6 Then Exit Sub
    If IsEmpty(Target) Or Selection.Cells.Count > 1 Then Exit Sub
    sTxt = CStr(Target.Value)
    Select Case Len(sTxt)
        Case 3: sTxt = "0" & sTxt & "00"
        Case 4: sTxt = sTxt & "00"
        Case 5: sTxt = "0" & sTxt
    End Select
    sTxt = Left(sTxt, 2) & ":" & Mid(sTxt, 3, 2) & ":" & Right(sTxt, 2)
    On Error GoTo ERRORHANDLER
    Application.EnableEvents = False
    Target.Value = TimeValue(sTxt)
    Application.EnableEvents = True
    Exit Sub
ERRORHANDLER:
    ActiveCell.ClearContents
    Application.EnableEvents = True
End Sub
```



TextBox-Eingaben im Format TTMMJJ in gültiges Datum umwandeln

Problem:	Wie kann ich Eingaben in eine UserForm-TextBox im Format TTMMJJ in ein gültiges Datum umwandeln? Es soll ebenfalls eine Eingabeprüfung erfolgen.
Lösung:	Der Code befindet sich im Klassenmodul der UserForm.
VBA-Code:	<pre>ClassModule: frmDatum Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub txtDatum_Change() Dim dteEingabe As Date Dim iDay As Integer, iMonth As Integer, iYear As Integer If txtDatum.Text = "" Then Exit Sub If Not IsNumeric(Right(txtDatum.Text, 1)) Then Beep txtDatum.SelStart = Len(txtDatum.Text) - 1 txtDatum.SelLength = 1 lblAusgabe.Caption = "Nur Ziffern erlaubt!" Exit Sub Else lblAusgabe.Caption = "" End If iDay = CInt(Left(txtDatum.Text, 2)) Select Case Len(txtDatum.Text) Case 0 Case 1 If iDay > 3 Then Beep txtDatum.SelStart = 0 txtDatum.SelLength = 1 lblAusgabe.Caption = "Maximal 31 Tage" Else lblAusgabe.Caption = "" End If Case 2 If iDay > 31 Then Beep txtDatum.SelStart = 0 txtDatum.SelLength = 2 lblAusgabe.Caption = "Maximal 31 Tage" Else lblAusgabe.Caption = "" End If Case 3 iMonth = CInt(Right(txtDatum.Text, 1)) If iMonth > 1 Then Beep txtDatum.SelStart = 2 txtDatum.SelLength = 1 lblAusgabe.Caption = "Maximal 12 Monate" Else lblAusgabe.Caption = "" End If Case 4 iMonth = CInt(Right(txtDatum.Text, 2)) If iMonth > 12 Then Beep txtDatum.SelStart = 2 txtDatum.SelLength = 2 lblAusgabe.Caption = "Maximal 12 Monate" Else lblAusgabe.Caption = "" End If Select Case iMonth Case 2 If iDay > 29 Then Beep txtDatum.SelStart = 0 txtDatum.SelLength = 4 lblAusgabe.Caption = "Maximal 29 Tage" Else lblAusgabe.Caption = "" End If Case 4, 6, 9, 11 If iDay > 30 Then Beep txtDatum.SelStart = 0 txtDatum.SelLength = 4 lblAusgabe.Caption = "Maximal 30 Tage" Else lblAusgabe.Caption = "" End If End Select Case 6 iMonth = CInt(Mid(txtDatum.Text, 3, 2)) iYear = CInt(Right(txtDatum.Text, 2)) If iYear Mod 4 > 0 And iMonth = 2 And iDay = 29 Then Beep txtDatum.SelStart = 0 txtDatum.SelLength = 6 lblAusgabe.Caption = "Maximal 28 Tage" Else dteEingabe = DateSerial(iYear, iMonth, iDay) lblAusgabe.Caption = dteEingabe & vbLf & Format(dteEingabe, "dddd") txtDatum.SelStart = 0 txtDatum.SelLength = 6 End If End Select End Sub Private Sub txtDatum_Exit(ByVal Cancel As MSForms.ReturnBoolean) If Len(txtDatum.Text) < 6 Then Cancel = True Beep lblAusgabe.Caption = "Datum eingeben!" End If End Sub ClassModule: Tabelle1 Private Sub cmdDialogAufruf_Click() frmDatum.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datumseingabe in Textfeld erzwingen

Problem: Wie kann ich in XL8 die Datumseingabe in ein Textfeld einer UserForm erzwingen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code: StandardModule: basMain

```
Sub CallForm()  
    frmDatum.Show  
End Sub
```

ClassModule: frmDatum

```
Private Sub cmdContinue_Click()  
    Unload Me  
End Sub
```

```
Private Sub txtDate_Change()  
    Dim sTxt As String  
    If txtDate.Text = "" Then Exit Sub  
    sTxt = txtDate.Text  
    If Right(sTxt, 1) Like "[!0-9]" Then  
        txtDate.Text = Left(sTxt, Len(sTxt) - 1)  
        Exit Sub  
    End If  
    If IsNumeric(sTxt) = False Then GoTo ERRORHANDLER  
    If Len(sTxt) = 6 Then  
        sTxt = Left(sTxt, 2) & "." & Mid(sTxt, 3, 2) _  
            & "." & Right(sTxt, 2)  
        If Not IsDate(sTxt) Then  
            GoTo ERRORHANDLER  
        Else  
            txtDate.Text = sTxt  
            Exit Sub  
        End If  
    End If  
    Exit Sub  
ERRORHANDLER:  
    Beep  
    MsgBox "Kein Datum!"  
    txtDate.Text = ""  
    txtDate.SetFocus  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vereinfachte Zeiteingabe

Problem: Ich habe in eine Spalte eine große Anzahl von Uhrzeiten einzugeben. Hierbei hemmt die Eingabe des Doppelpunktes bei z.B. "12:15" den Eingabefluß. Wie kann ich die Eingabe vereinfachen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub InZeit()  
    Dim rng As Range  
    Dim sTxt As String  
    For Each rng In Selection.Cells  
        sTxt = rng.Value  
        rng.NumberFormat = "hh:mm"  
        rng.Value = TimeSerial(Left(sTxt, 2), Right(sTxt, 2), 0)  
    Next rng  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verknüpfte Zelle einer ComboBox mit ListIndex statt Wert

Problem:	Die mit einer Tabellenblatt-ComboBox verknüpfte Zelle soll den ListIndex (bei 0 beginnend), nicht den Wert ausweisen.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub ComboBox1_Change() End Sub Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datumsreihen nach Vorgabe markieren

Problem:	Wie kann ich eine Datumsreihe nach Vorgabe von Tages- und MonatsIndizes markieren und mit Bereichsnamen belegen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DatumSuchen() Dim iRow As Integer, iStart As Integer Dim iEnd As Integer, iMonth As Integer iRow = 2 For iMonth = Range("B2").Value To Range("C2").Value Do Until Month(Cells(iRow, 1)) = iMonth iRow = iRow + 1 Loop iStart = iRow Do While Month(Cells(iRow, 1)) = iMonth iRow = iRow + 1 If IsEmpty(Cells(iRow, 1)) Then Exit Do Loop iEnd = iRow - 1 Range(Cells(iStart, 1), Cells(iEnd, 1)).Name = _ Format(DateSerial(0, iMonth, 1), "mmm") Range(Format(DateSerial(0, iMonth, 1), "mmm")). _ Interior.ColorIndex = iMonth + 2 Next iMonth End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrundfarbe nach Datum festlegen

Problem:	Wie kann ich die Hintergrundfarbe einer festzulegenden Datumsreihe ändern?
Lösung:	Formel für bedingte Formatierung: =UND(C1>=\$A\$2;C1<=\$B\$2)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dezimal- als Zeitformat und umgekehrt

Problem:	Wie kann ich Dezimalzahlen in Uhrzeiten und Uhrzeiten in Dezimalzahlen umwandeln und addieren?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Markieren in Abhängigkeit vom Quartal

Problem:	Wie kann ich bei Eingabe eines Datums in Spalte A die Spalten A:F der entsprechenden Zeile unterschiedlich nch Quartal markieren lassen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 1 Then Exit Sub If Target.Cells.Count > 1 Then Exit Sub If IsEmpty(Target) Then Target.Interior.ColorIndex = xlColorIndexNone Exit Sub Else Select Case Format(Target, "q") Case 1: Target.Interior.ColorIndex = 5 Case 2: Target.Interior.ColorIndex = 6 Case 3: Target.Interior.ColorIndex = 7 Case 4: Target.Interior.ColorIndex = 8 End Select End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Runde Geburtstage aus einer Liste ermitteln

Problem:	In Spalte E sollen die runden Geburtstage gem. den Werten in Spalte C ermittelt werden. Über die Hilfsspalte D wird festgestellt, ob es sich um einen Geburtstag handelt.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Geburtstage eines vorzugebenden Monats in neue Tabelle

Problem: Wie kann ich aus einer Personalliste die Geburtstage eines bestimmten Monats in eine neue Tabelle übertragen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Filtern()  
    Dim wks As Worksheet  
    Dim iRow As Integer, iRowT As Integer  
    Set wks = ActiveSheet  
    iRow = 2  
    iRowT = 2  
    Worksheets.Add after:=Worksheets(Worksheets.Count)  
    Columns(3).NumberFormat = "dd.mm.yy"  
    wks.Range("A1:C1").Copy Range("A1:C1")  
    Do Until IsEmpty(wks.Cells(iRow, 1))  
        If Month(wks.Cells(iRow, 3).Value) = _  
            Month(Date) Then  
            Range(Cells(iRowT, 1), Cells(iRowT, 3)).Value = _  
                wks.Range(wks.Cells(iRow, 1), wks.Cells(iRow, 3)).Value  
            iRowT = iRowT + 1  
        End If  
        iRow = iRow + 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Jahreskalender mit Mitarbeiterliste und Feiertagen anlegen

Problem:	Es soll ein Jahreskalender mit 12 Monatsblättern und einer Mitarbeiterliste angelegt werden. Feiertage und Wochenenden sollen hinzugefügt oder weggelassen werden können.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub MonateAnlegen() Dim wks As Worksheet Dim var As Variant Dim datDay As Date Dim iMonth As Integer, iCol As Integer, iCounter As Integer, iYear As Integer Dim sMonth As String Dim bln As Boolean With Application .ScreenUpdating = False bln = .DisplayStatusBar .DisplayStatusBar = True End With iYear = Cover.SpinButton1.Value Workbooks.Add Application.DisplayAlerts = False For iCounter = 1 To Worksheets.Count - 1 Worksheets(2).Delete Next iCounter Application.DisplayAlerts = True Set wks = ThisWorkbook.Worksheets("Mitarbeiter") For iMonth = 1 To 12 sMonth = Format(DateSerial(1, iMonth, 1), "mmmm") Application.StatusBar = "Lege Monat " & sMonth & " an..." Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Name = sMonth wks.Range(wks.Cells(3, 1), wks.Cells(_ WorksheetFunction.CountA(wks.Columns(1)) + 1, 1)).Copy Range("A2") Range("A1").Value = "" & ActiveSheet.Name & " " & iYear If Cover.OptionButton1.Value And Cover.OptionButton3.Value Then Call WithHW(iMonth) ElseIf Cover.OptionButton1.Value And Cover.OptionButton3.Value = False Then Call WithWsansH(iMonth) ElseIf Cover.OptionButton1.Value = False And Cover.OptionButton3.Value Then Call WithHsansW(iMonth) Else Call SansWH(iMonth) End If Rows(2).Value = Rows(1).Value Rows(2).NumberFormat = "ddd" Range("A2").Value = "Wochentage" Rows("1:2").Font.Bold = True Columns.AutoFit Next iMonth Application.DisplayAlerts = False Worksheets(1).Delete Application.DisplayAlerts = True Worksheets(1).Select ActiveWindow.Caption = "Jahreskalender " & iYear With Application .ScreenUpdating = True .DisplayStatusBar = bln .StatusBar = False End With End Sub Private Sub WithHW(ByVal iMonth As Integer) Dim cmt As Comment Dim rng As Range Dim var As Variant Dim datDay As Date Dim iYear As Integer, iCol As Integer iCol = 1 iYear = Cover.SpinButton1.Value For datDay = DateSerial(iYear, iMonth, 1) To _ DateSerial(iYear, iMonth + 1, 0) iCol = iCol + 1 Set rng = Range(Cells(1, iCol), _ Cells(WorksheetFunction.CountA(Columns(1)), iCol)) var = Application.Match(CDbl(datDay), _ ThisWorkbook.Worksheets("Feiertage").Columns(1), 0) Cells(1, iCol).Value = datDay With rng.Interior Select Case Weekday(datDay) Case 1 .ColorIndex = 35 Case 7 .ColorIndex = 36 End Select If Not IsError(var) Then .ColorIndex = 34 Set cmt = Cells(1, iCol).AddComment(_ ThisWorkbook.Worksheets("Feiertage").Cells(var, 2).Value) cmt.Shape.TextFrame.AutoSize = True End If End With Next datDay End Sub Private Sub WithHsansW(ByVal iMonth As Integer) Dim datDay As Date Dim iYear As Integer, iCol As Integer iCol = 1 iYear = Cover.SpinButton1.Value For datDay = DateSerial(iYear, iMonth, 1) To _ DateSerial(iYear, iMonth + 1, 0) If WorksheetFunction.Weekday(datDay, 2) < 6 Then iCol = iCol + 1 Cells(1, iCol).Value = datDay End If Next datDay End Sub Private Sub WithWsansH(ByVal iMonth As Integer) Dim var As Variant Dim datDay As Date Dim iYear As Integer, iCol As Integer iCol = 1 iYear = Cover.SpinButton1.Value For datDay = DateSerial(iYear, iMonth, 1) To _ DateSerial(iYear, iMonth + 1, 0) var = Application.Match(CDbl(datDay), _ ThisWorkbook.Worksheets("Feiertage").Columns(1), 0) If IsError(var) Then iCol = iCol + 1 Cells(1, iCol).Value = datDay End If Next datDay End Sub Private Sub SansWH(ByVal iMonth As Integer) Dim var As Variant Dim datDay As Date Dim iYear As Integer, iCol As Integer iCol = 1 iYear = Cover.SpinButton1.Value For datDay = DateSerial(iYear, iMonth, 1) To _ DateSerial(iYear, iMonth + 1, 0) If WorksheetFunction.Weekday(datDay, 2) < 6 Then var = Application.Match(CDbl(datDay), _ ThisWorkbook.Worksheets("Feiertage").Columns(1), 0) If IsError(var) Then iCol = iCol + 1 Cells(1, iCol).Value = datDay End If End If Next datDay End Sub Sub FeiertageEinAus() With Worksheets("Feiertage") If .Visible = xlVeryHidden Then .Visible = True .Select Else .Visible = xlVeryHidden Worksheets(1).Select End If End With End Sub Sub MitarbeiterEinAus() With Worksheets("Mitarbeiter") If .Visible = xlVeryHidden Then .Visible = True .Select Else .Visible = xlVeryHidden Worksheets(1).Select End If End With End Sub Sub Zurueck() ActiveSheet.Visible = xlVeryHidden Worksheets(1).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamischer Kalender mit Jahr, Quartal und Monaten

Problem: In Zeile 1 soll das jeweilige Jahr, in Zeile 2 das Quartal und in Zeile 4 der Monat eingetragen werden. Der Kalender soll dynamisch und durch den Zellwert in Zelle A4 bedingt sein.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Function Quartal(dat As Date)
    Quartal = Format(dat, "q") & "Q"
End Function
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Jahreskalender mit KW nach DIN1355 anlegen

Problem:	In Spalte A sollen die Tage des aktuellen Jahres, in Spalte B die zugehörige Kalenderwochen nach DIN 1355 eingetragen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub DatumUndKW() Dim iCount As Integer, iCounter As Integer If Month(DateSerial(Year(Date), 2, 29)) = 2 Then iCount = 366 Else iCount = 365 End If For iCounter = 1 To iCount Cells(iCounter, 1) = DateSerial(Year(Date), 1, iCounter) Cells(iCounter, 2) = DINWeek(Cells(iCounter, 1)) Next iCounter End Sub Private Function DINWeek(dat As Date) As Integer Dim dbl As Double dbl = DateSerial(Year(dat + (8 - Weekday(dat)) Mod 7 - 3), 1, 1) DINWeek = (dat - dbl - 3 + (Weekday(dbl) + 1) Mod 7) \ 7 + 1 End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tagesblätter eines Monats ohne Wochenenden und Feiertage anlegen

Problem: Wie kann ich in der aktuellen Arbeitsmappe für jeden Tag des Monats ein Arbeitsblatt anlegen, wobei Wochenenden und Feiertage ausgespart bleiben sollen? Die Tagesblätter sind nach Datum zu benennen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub MonatAnlegen()
    Dim wks As Worksheet
    Dim datDay As Date
    Dim var As Variant
    Dim lDay As Long
    Dim iYear As Integer, iMonth As Integer
    Application.ScreenUpdating = False
    Set wks = ActiveSheet
    iYear = Year(Range("A1").Value)
    iMonth = Month(Range("A1").Value)
    For lDay = DateSerial(iYear, iMonth, 1) To _
        DateSerial(iYear, iMonth + 1, 0)
        var = Application.Match( _
            lDay, Worksheets("Feiertage").Columns(1), 0)
        If IsError(var) And WorksheetFunction.WeekDay(lDay) < 6 Then
            Worksheets.Add.Move after:=Worksheets(Worksheets.Count)
            ActiveSheet.Name = Format(lDay, "dd.mm.yy")
        End If
    Next lDay
    wks.Select
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus einem Kalender-Steuerelement Datum und Zeit auswählen

Problem: Wie kann ich in einer UserForm durch Auswahl in einer zweiten UserForm Datum und Zeit eintragen lassen?

Lösung: Den nachfolgenden Code in das Klassenmodul der benannten UserForms einfügen

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdDialogAufruf_Click()
    frmAufruf.Show
End Sub

ClassModule: frmAufruf

Private Sub cmdAuswahl_Click()
    frmDateControl.Show
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

ClassModule: frmDateControl

Private Sub cmdWeiter_Click()
    frmAufruf.lblDatum.Caption = _
        Format(calJahr.Value, "dd. mmmm yyyy")
    frmAufruf.lblZeit.Caption = _
        Format(cboZeit.Value, "hh:mm")
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim dTime As Double
    Dim iList As Integer
    For iList = 1 To 24 * 4
        cboZeit.AddItem Format(dTime, "hh:mm")
        dTime = dTime + (1 / 24 / 4)
    Next iList
    cboZeit.ListIndex = 0
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Jahresarbeitszeitkalender

Problem:	Wie kann ich einen Jahresarbeitszeitkalender erstellen, der meine individuellen Anfangs- und Pausenzeiten berücksichtigt?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	<pre>StandardModule: basMain Sub SchutzEin() Dim iWks As Integer For iWks = 2 To 13 Worksheets(iWks).Protect Next iWks End Sub Sub SchutzAus() Dim iWks As Integer For iWks = 2 To 13 Worksheets(iWks).Unprotect Next iWks End Sub</pre>



Jahreskalender mit Monatsblättern anlegen

Problem:	Wie kann ich auf Buttondruck einen kompletten Jahreskalender mit 12 Monatsblättern anlegen? Samstag, Sonntag und Feiertage, einschließlich der beweglichen, sollen markiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Sub Main() Dim wks As Worksheet Dim cmt As Comment Dim vYear As Variant Dim iRow As Integer Dim bln As Boolean Application.ScreenUpdating = False bln = Application.DisplayStatusBar Application.DisplayStatusBar = True Application.DisplayCommentIndicator = xlCommentIndicatorOnly Set wks = ActiveSheet vYear = InputBox(_ prompt:="Gewünschtes Kalenderjahr angeben:", _ Default:=Year(Date)) Range("C1").Value = CInt(vYear) Workbooks.Add 1 Call MonateAnlegen Call TageEintragen iRow = 1 Do Until IsEmpty(wks.Cells(iRow, 1)) With Worksheets(Month(wks.Cells(iRow, 2).Value)) With .Cells(Day(wks.Cells(iRow, 2).Value), 1) .Interior.ColorIndex = 36 Set cmt = .AddComment(wks.Cells(iRow, 1).Value) cmt.Shape.TextFrame.AutoSize = True End With End With iRow = iRow + 1 Loop Application.DisplayStatusBar = bln Application.StatusBar = False Application.ScreenUpdating = True End Sub Private Sub MonateAnlegen() Dim iMonth As Integer For iMonth = 1 To 12 If iMonth > 1 Then Worksheets.Add.Move after:=Worksheets(Worksheets.Count) End If ActiveSheet.Name = Format(_ DateSerial(Range("C1").Value, iMonth, 1), "mmmm") Next iMonth End Sub Private Sub TageEintragen() Dim wks As Worksheet, wksMy As Worksheet Dim lDay As Long Dim iMonth As Integer, iDay As Integer Set wksMy = ThisWorkbook.Worksheets("Feiertage") For iMonth = 1 To 12 Set wks = Worksheets(iMonth) Application.StatusBar = "Bearbeite Monat " & wks.Name wks.Columns(1).NumberFormat = "dd.mm.yy" wks.Columns(2).NumberFormat = "dddd" For lDay = DateSerial(wksMy.Range("C1").Value, iMonth, 1) To _ DateSerial(wksMy.Range("C1").Value, iMonth + 1, 0) iDay = iDay + 1 wks.Cells(iDay, 1) = lDay wks.Cells(iDay, 2) = lDay If WeekDay(lDay) = 7 Then wks.Cells(iDay, 1).Interior.ColorIndex = 34 wks.Cells(iDay, 2).Interior.ColorIndex = 34 ElseIf WeekDay(lDay) = 1 Then wks.Cells(iDay, 1).Interior.ColorIndex = 35 wks.Cells(iDay, 2).Interior.ColorIndex = 35 End If Next lDay iDay = 0 Next iMonth Worksheets(1).Select ActiveWindow.Caption = "Jahreskalender " & wksMy.Range("C1").Value End Sub StandardModule: basFunction Function Ostern(iYear As Integer) Dim iDay As Integer iDay = (((255 - 11 * (iYear Mod 19)) - 21) Mod 30) + 21 Ostern = DateSerial(iYear, 3, 1) + iDay + (iDay > 48) + _ 6 - ((iYear + iYear \ 4 + iDay + (iDay > 48) + 1) Mod 7) End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eintrag der aktuellen Kalenderwoche über ShortCut

Problem:	Mit dem ShortCut Strg+k soll in die aktive Zelle die aktuelle Kalenderwochen nach DIN 1355 eingetragen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn dem ShortCut zu.
VBA-Code:	<pre>StandardModule: Modul1 Function KW() With WorksheetFunction KW = CInt((Date - .Weekday(Date, 2) - DateSerial(Year(Date) + 4 - .Weekday(Date, 2)), 1, -10)) / 7) End With End Function Sub InsertKW() ActiveCell.Value = KW End Sub StandardModule: Modul2 Sub Makro2() Makro2 Makro Makro am 02.10.2003 von Hans W. Herber aufgezeichnet ActiveSheet.Paste End Sub Sub Makro3() Makro3 Makro Makro am 02.10.2003 von Hans W. Herber aufgezeichnet ActiveSheet.Paste End Sub Sub Makro4() Makro4 Makro Makro am 02.10.2003 von Hans W. Herber aufgezeichnet Range("E21").Select ActiveCell.FormulaR1C1 = _ "=TRUNC((R[-19]C[-4]-WEEKDAY(R[-19]C[-4],2)-DATE(YEAR(R[-19]C[-4])+4-WEEKDAY(R[-19]C[-4],2))),1,-10))/7)" End Sub</pre>



Tabellenblätter für Kalenderwochen anlegen und ein / ausblenden

Problem:	Es sollen Tabellenblätter nach DIN 1355 angelegt, ausgeblendet, eingeblendet und gelöscht werden können.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: Modul1 <pre>Sub CreateWks() Dim datStart As Date, datEnd As Date Dim lKW As Long Dim iKW As Integer Dim sKW As String Application.ScreenUpdating = False datStart = DateSerial(Year(Date), 1, 1) datEnd = DateSerial(Year(Date), 12, 31) For lKW = datStart To datEnd Step 7 Worksheets.Add after:=Worksheets(Worksheets.Count) iKW = ISOWeek(CDate(lKW)) If lKW = CLng(datStart) And iKW > 1 Then sKW = "KW" & iKW & "S" ElseIf lKW >= datEnd - 7 And iKW = 1 Then sKW = "KW" & iKW & "E" Else sKW = "KW" & Format(iKW, "00") End If ActiveSheet.Name = sKW Next lKW Application.ScreenUpdating = True Worksheets(1).Select End Sub Private Function ISOWeek(dat As Date) As Integer With WorksheetFunction ISOWeek = Fix((dat - .Weekday(dat, 2) - _ DateSerial(Year(dat + 4 - _ .Weekday(dat, 2)), 1, -10)) / 7) End With End Function Sub WksHidden() Dim iWks As Integer, iAct As Integer Dim iKW As String, sKW As String Application.ScreenUpdating = False iKW = ISOWeek(Date) If Month(Date) = 1 And iKW > 50 Then sKW = "KW" & Format(iKW, "00") & "A" ElseIf Month(Date) = 12 And iKW = 1 Then sKW = "KW01" & "E" Else sKW = "KW" & Format(iKW, "00") End If For iWks = 2 To Worksheets.Count If Worksheets(iWks).Name = sKW Then iAct = iWks Exit For End If Next iWks For iWks = 2 To Worksheets.Count If iWks < iAct - 4 Or iWks > iAct + 4 Then Worksheets(iWks).Visible = False Else Worksheets(iWks).Visible = True End If Next iWks Application.ScreenUpdating = True End Sub Sub WksNoHidden() Dim wks As Worksheet Application.ScreenUpdating = False For Each wks In Worksheets wks.Visible = True Next wks Application.ScreenUpdating = True End Sub Sub WksDelete() Dim wks As Worksheet With Application .ScreenUpdating = False .DisplayAlerts = False For Each wks In Worksheets If wks.Index > 1 Then wks.Delete Next wks .DisplayAlerts = True .ScreenUpdating = True End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle mit der jeweils nächsten Kalenderwoche öffnen

Problem:	Wie kann ich veranlassen, daß eine Tabelle sich beim Öffnen der Arbeitsmappe immer an die auf das aktuelle Datum folgende Kaldenderwoche anpaßt?
Lösung:	Formel für die aktuelle Kalenderwoche: =KÜRZEN((C11-WOCHENTAG(C11;2)-DATUM(JAHR(C11+4-WOCHENTAG(C11;2));1;-10))/7)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Allen Daten löschen, wenn das Jahr nicht der Vorgabe entspricht

Problem: In den Folgeblättern sollen alle Zeilen mit Datumsangaben in Spalte A gelöscht werden, bei denen das Jahr nicht dem vorgegebenen entspricht.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub DeleteAndPrint()
    Dim vYear As Variant
    Dim iWks As Integer, iRow As Integer, iRowL As Integer
    vYear = InputBox("Jahr:", , "2004")
    If vYear = "" Then Exit Sub
    For iWks = ActiveSheet.Index + 1 To Worksheets.Count
        With Worksheets(iWks)
            iRowL = .Cells(Rows.Count, 1).End(xlUp).Row
            For iRow = iRowL To 1 Step -1
                If Year(.Cells(iRow, 1).Value) <> CInt(vYear) Then
                    .Rows(iRow).Delete
                End If
            Next iRow
            .PrintPreview
        End With
    Next iWks
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bestimmte Wochentage ohne Feiertage listen

Problem:	In den Spalten E:F werden alle Dienstage und Donnerstage gelistet, sofern es sich nicht um einen Feiertag handelt.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Anlegen() Dim vDay As Variant Dim iRow As Integer, lDay As Long For lDay = DateSerial(Range("C1").Value, 1, 1) To DateSerial(Range("C1").Value, 12, 31) If Weekday(lDay) = 3 Or Weekday(lDay) = 5 Then If IsError(Application.Match(lDay, Columns(1), 0)) Then iRow = iRow + 1 Cells(iRow, 5).Value = CDate(lDay) Cells(iRow, 6).Value = Format(lDay, "ddd") End If End If Next lDay End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamische Tabelle mit Tages, Wochen- und Monatsbereichen

Problem:	In einer dynamischen Tabelle soll in Abhängigkeit der Datumseingabe in Zelle B2 wird für die aktuelle Woche eine Stundenliste angelegt, für den Restmonat eine Tages und das Restjahr eine Monatsliste.
Lösung:	Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten aus Tagesliste auf Tagesblätter verteilen

Problem:	Wie kann ich Daten aus einer Tagesliste (Datum in Spalte A) auf die einzelnen Tagesblätter verteilen. Die Blattnamen entsprechen dem jeweiligen Datum.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Aufteilen() Dim wks As Worksheet Dim iRowS As Integer, iRowT As Integer iRowS = 2 Do Until IsEmpty(Cells(iRowS, 1)) Set wks = Worksheets(Cells(iRowS, 1).Text) iRowT = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1 Range(Cells(iRowS, 1), Cells(iRowS, 4)).Copy _ wks.Range(wks.Cells(iRowT, 1), wks.Cells(iRowT, 4)) iRowS = iRowS + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl eines bestimmten Wochentages im Monat

Problem:	Wie kann ich feststellen, wie oft ein bestimmter Wochentag in einem vorgegebenen Monat vorkommt?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function nDays(dat, iDay As Integer) Dim datFirst As Date, datLast As Date Dim iCount As Integer Dim lDay As Long datFirst = DateSerial(Year(dat), Month(dat), 1) datLast = DateSerial(Year(dat), Month(dat) + 1, 0) For lDay = datFirst To datLast If WeekDay(lDay) = iDay Then iCount = iCount + 1 End If Next lDay nDays = iCount End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wochentage Mo - Fr in Tabelle eintragen

Problem:	Ich möchte in Spalte A die Wochentage Mo - Fr mit einer nachfolgenden Leerzelle und in Spalte B das jeweil zugehörige Datum für einen vorgegebenen Zeitraum eintragen. Wie kann ich das über VBA lösen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub WochenendeWeg() Dim datStart As Date, datEnd As Date Dim lDay As Long Dim iRow As Integer datStart = Range("D1").Value datEnd = Range("D2").Value For lDay = datStart To datEnd If WeekDay(lDay, 2) < 6 Then iRow = iRow + 1 Cells(iRow, 1) = lDay Cells(iRow, 2) = lDay End If Next lDay End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beispiele für die Zeitberechnung in Excel

Problem:	Beispiel für die Excel'sche Zeitberechnung. Die Einzelheiten gehen aus den Formeln hervor.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Negative Zeitdarstellung ohne Umstellung der 1904-Datumswerte

Problem:	Eine Negativzeit soll ohne Umstellung des 1904-Datumswertes dargestellt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Km / h errechnen

Problem:	Die Geschwindigkeit per Stunde wird errechnet.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zählen von Jahren, Monaten, Tagen, Stunden, Minuten, Sekunden

Problem:	In einer Tabelle wird gezählt, wieviel Zeitabschnitte zum Ausgangspunkt vergangen sind bzw. zum Endpunkt noch zu erwarten sind. Abbruch der Routine mit der ESCAPE-Taste.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>StandardModule: basMain Sub Countdown() On Error GoTo ERRORHANDLER Application.EnableCancelKey = xlErrorHandler Do Application.Wait Now + Range("B3").Value Application.Calculate Loop While Err = 0 ERRORHANDLER: MsgBox "Ablauf wurde unterbrochen!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeit über InputBox abfragen, dividieren und zurückgeben

Problem:	Über InputBox wird eine Zeit abgefragt und durch 4 dividiert zurückgegeben.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub ZeitTeilen() Dim vTime As Variant vTime = InputBox("Zeitangabe:", , Format(Now, "hh:mm")) If vTime = "" Then Exit Sub If IsDate(vTime) = False Then Beep MsgBox "Keine gültige Zeitangabe!" Exit Sub End If vTime = CDate(vTime) / 4 MsgBox "Division durch 4:" & vbCrLf & Format(vTime, "hh:mm") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Rechnen mit Zehntausendstel-Sekunden

Problem:	Es soll mit Werten im Zehntausendstelsekundenbereich gerechnet werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function SpecTime(sTxt As String) Const DIV As Long = 86400 Dim dValue As Double dValue = CInt(Right(sTxt, 4)) / 1000 / DIV dValue = dValue + CInt(Mid(sTxt, 4, 2)) / (DIV / 60) dValue = dValue + CInt(Left(sTxt, 2)) / (DIV / 60 / 60) SpecTime = dValue End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Von einer Gesamtzeit die aktuell vergangene Zeit abziehen

Problem: Wie kann ich von einer vorgegebenen Gesamtzeit die von Start bis Aktuell vergangene Zeit abziehen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call StopClock
End Sub

StandardModule: basMain

Public Const gsMacro As String = "UpdateClock"
Public gdNextTime As Double

Sub StartClock()
    gdNextTime = Now + TimeSerial(0, 0, 5)
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=True
End Sub

Sub StopClock()
    On Error Resume Next
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=False
End Sub

Private Sub UpdateClock()
    If Range("B2").Value >= (Range("B1").Value - _
        TimeSerial(0, 0, 5)) Then
        Beep
        Range("B2").Value = 0
        Call StopClock
    Else
        Range("B2").Value = Range("B2").Value + TimeSerial(0, 0, 5)
        Call StartClock
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Startzeit und auf Schaltflächendruck Zwischenzeiten eintragen

Problem:	Wie kann ich auf Schaltflächendruck in einer Zelle eine Startzeit eingeben? Auf weiteren Schaltflächendruck sollen die Differenzen der aktuellen Zeit sowohl zu der Startzeit wie auch zu der letzten Zwischenzeit angegeben werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub StartClock() Range("A2:C" & Rows.Count).ClearContents Range("A2") = Time End Sub Sub PrintClock() Dim intRow As Integer intRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 Cells(intRow, 1) = Time Cells(intRow, 2) = Cells(intRow, 1) - Range("A2") Cells(intRow, 3) = Cells(intRow, 1) - Cells(intRow - 1, 1) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zusammenstellung der eingebauten Dialoge

Problem: Eine Zusammenstellung aller eingebauten Excel-Dialoge

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_FollowHyperlink(ByVal Target As Hyperlink)
    Dim vDlg As Variant
    On Error Resume Next
    vDlg = Range(Target.SubAddress).Offset(0, 1).Value
    Application.Dialogs(vDlg).Show
    On Error GoTo 0
End Sub
```

StandardModule: Modul1

```
Sub SetHyperlinks()
    Dim iRow As Integer
    ActiveSheet.Hyperlinks.Delete
    For iRow = 2 To WorksheetFunction.CountA(Columns(1))
        ActiveSheet.Hyperlinks.Add _
            anchor:=Cells(iRow, 1), _
            Address:="", _
            SubAddress:=Cells(iRow, 1).Address(False, False)
    Next iRow
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufruf des GetOpenFilename-Dialogs

Problem:	Der GetOpenFilename-Dialog soll mit den Dateifiltern Text, Word und Excel aufgerufen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub TxtAndXl() Dim var As Variant Dim sFiles As String sFiles = "Text-Files (*.txt), *.txt, " sFiles = sFiles & "Word-Files (*.doc), *.doc, " sFiles = sFiles & "Excel-Files(*.xls), *.xls" var = Application.GetOpenFilename(sFiles) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattschutzdialog über VBA aufrufen

Problem:	Wie kann ich den Blattschutzdialog über VBA aufrufen und den Blattschutz mit oder ohne Kennwort setzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BlattSchutzEin() Application.Dialogs(xlDialogProtectDocument).Show End Sub Sub BlattSchutzAus() ActiveSheet.Unprotect End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Office-Assistenten mit vorgegebenen Texte einblenden

Problem:	Auf Schaltflächendruck in einer UserForm soll der Assistent mit einem Hilfstext eingeblendet werden.
Lösung:	Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() frmHelp.Show End Sub Sub AssistentEinblenden() Dim intSelect As Integer Dim strUrl As String Static blnAssi As Boolean On Error GoTo ERRORHANDLER blnAssi = Assistant.Visible Assistant.Visible = True With Assistant.NewBalloon .Heading = "Excel-Lösungen von Hans W. Herber finden Sie ..." .Labels(1).Text = _ "... {cf 252}{ul 1}auf dem Excel-Server{ul 0}{cf 0} ..." .Labels(2).Text = _ "... {cf 252}{ul 1}auf der Excel-CD-ROM{ul 0}{cf 0} ..." .Labels(3).Text = _ "... {cf 252}{ul 1}im Excel-Forum{ul 0}{cf 0} ..." .Button = msoButtonSetCancel .Animation = msoAnimationSearching intSelect = .Show End With Select Case intSelect Case 1: Call ExplorerOeffnen("http://www.herber.de/") Case 2: Call ExplorerOeffnen("http://www.excel-cd.de/") Case 3: Call ExplorerOeffnen("http://www.herber.de/forum/") End Select Assistant.Visible = blnAssi Exit Sub ERRORHANDLER: MsgBox Chr(169) & "2001 Hans W. Herber" & vbLf & _ "hans@herber.de" & vbLf & "http://www.herber.de" End Sub Private Sub ExplorerOeffnen(strUrl As String) Dim objExplorer As Object On Error GoTo ERRORHANDLER Set objExplorer = CreateObject("InternetExplorer.Application") With objExplorer .Navigate strUrl .Visible = True .Offline = False End With Exit Sub ERRORHANDLER: End Sub ClassModule: frmHelp Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdHelp_Click() Call AssistentEinblenden End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm mit Laufschriftfeld

Problem: Es soll eine UserForm mit einem Laufschriftfeld aufgerufen werden. Der Text der Laufschrift ist variabel zu halten.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMarquee

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub cmdText_Click()
    Dim sPath As String
    WebBrowser1.Navigate ""
    sPath = Application.DefaultFilePath & "\marquee.htm"
    Call CreateHTML(sPath)
    WebBrowser1.Navigate sPath
End Sub

StandardModule: basMain

Sub CallForm()
    Dim sFile As String
    sFile = Application.DefaultFilePath & "\marquee.htm"
    Call CreateHTML(sFile)
    With frmMarquee
        .WebBrowser1.Navigate url:=sFile
        .Show
    End With
    Kill sFile
End Sub

Sub CreateHTML(sPath As String)
    Open sPath For Output As #1
    Print #1, "<html>"
    Print #1, "<head>"
    Print #1, "<style>"
    Print #1, "td { "
    Print #1, "    background-color:#000000;"
    Print #1, "    color:#ffff00;"
    Print #1, "    font-family:tahoma,verdana;"
    Print #1, "    font-size:12px"
    Print #1, "}"
    Print #1, "</style>"
    Print #1, "</head>"
    Print #1, "<body bgcolor=#c0c0c0>"
    Print #1, "<table border=1 cellpadding=3 cellspacing=1><tr><td width=300>"
    Print #1, "<marquee>" & Range("A1").Value & "</marquee>"
    Print #1, "</td></tr></table>"
    Print #1, "</body>"
    Print #1, "</html>"
    Close
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kalendersteuerelement an Tabellenblattwerte binden

Problem: Kalendersteuerelementwerte sollen an Tabellenblattwerte gebunden werden.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm mit Kalendersteuerelement immer mit aktuellem Datum

Problem: Das Kalendersteuerelement soll immer mit dem aktuellen Datum starten. Falls das Steuerelement nicht installiert ist, nehmen Sie die Installation über Access vor.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmCal

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Calendar1.Year = Year(Date)
    Calendar1.Month = Month(Date)
    Calendar1.Day = Day(Date)
End Sub

StandardModule: Modul1

Sub Aufruf()
    frmCal.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bereiche gem. CheckBox-Aktivierung auswählen

Problem: Die Bereiche gemäß den CheckBoxAufschriften sollen bei CheckBox-Aktivierung ausgewählt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CheckBox1_Click()
    If CheckBox1.Value Then
        Range(CheckBox2.Caption).Select
        If CheckBox2.Value Then
            Range(CheckBox1.Caption, CheckBox2.Caption).Select
        Else
            Range(CheckBox1.Caption).Select
        End If
    Else
        If CheckBox2.Value Then
            Range(CheckBox2.Caption).Select
        End If
    End If
End Sub

Private Sub CheckBox2_Click()
    If CheckBox2.Value Then
        Range(CheckBox1.Caption).Select
        If CheckBox1.Value Then
            Range(CheckBox2.Caption, CheckBox1.Caption).Select
        Else
            Range(CheckBox2.Caption).Select
        End If
    Else
        If CheckBox1.Value Then
            Range(CheckBox1.Caption).Select
        End If
    End If
End Sub

StandardModule: Modul1

Sub Versenden()
    With Workbooks("11215.xls")
        .SendMail "hans@herber.de", "test"
        .Close savechanges:=False
    End With
    Application.DisplayFullScreen = False
    Application.CommandBars("Worksheet Menu Bar").Enabled = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Name der zuletzt erstellten CheckBox auslesen

Problem:	Der Name der zuletzt erstellten CheckBox wird ausgelesen um gemeldet.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub LastChkBox() Dim oOle As OLEObject Dim arr() As Integer Dim iCounter As Integer Dim sName As String For Each oOle In ActiveSheet.OLEObjects If TypeName(oOle.Object) = "CheckBox" Then iCounter = iCounter + 1 sName = oOle.Name sName = WorksheetFunction.Substitute(sName, "CheckBox", "") ReDim Preserve arr(1 To iCounter) arr(iCounter) = CInt(sName) End If Next oOle MsgBox "Name der zuletzt erstellten CheckBox:" & vbCrLf & _ "CheckBox" & WorksheetFunction.Max(arr) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufschriften und Anzahl der aktivierten ComboBoxes auslesen

Problem: Die Aufschriften und die Anzahl der ausgewählten CheckBoxes sollen ermittelt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmCheckBoxes

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdValues_Click()
    Dim cnt As Control
    Dim arr() As String
    Dim iCounter As Integer
    For Each cnt In Controls
        If UCase(TypeName(cnt)) = "CHECKBOX" Then
            If cnt.Value Then
                iCounter = iCounter + 1
                ReDim Preserve arr(1 To iCounter)
                arr(iCounter) = cnt.Caption
            End If
        End If
    Next cnt
    MsgBox "Ausgewählt: " & iCounter
    For iCounter = 1 To UBound(arr)
        MsgBox iCounter & ". Auswah: " & arr(iCounter)
    Next iCounter
End Sub

StandardModule: Modul1

Sub CallForm()
    frmCheckBoxes.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in Abhängigkeit von CheckBox-Aktivierung summieren

Problem: Bei Klick auf eine CheckBox sollen die Werte aus Spalte B von den Zellen addiert werden, deren zugeordnete CheckBox aktiviert ist.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CheckBox1_Click()
    Call GetSum
End Sub

Private Sub CheckBox2_Click()
    Call GetSum
End Sub

Private Sub CheckBox3_Click()
    Call GetSum
End Sub

Private Sub CheckBox4_Click()
    Call GetSum
End Sub

Private Sub CheckBox5_Click()
    Call GetSum
End Sub

Sub GetSum()
    Dim chb As OLEObject
    Dim dSum As Double
    Dim iChb As Integer
    For iChb = 1 To 5
        Set chb = OLEObjects(iChb)
        If chb.Object.Value = True Then
            dSum = dSum + chb.TopLeftCell.Offset(0, -1).Value
        End If
    Next iChb
    Range("B7").Value = dSum
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benennung und Aktivierung von CheckBoxes gem. Zellinhalten

Problem:	UserForm-CheckBoxes sollen nach den Zellinhalten im Bereich A1:B4 benannt werden. Ist die Zelle leer, soll die CheckBox deaktiviert werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmEnabled Private Sub cmdOK_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim rng As Range Dim iCounter As Integer Set rng = Range("A1:B4") For iCounter = 1 To 8 If IsEmpty(rng(iCounter)) Then Controls("CheckBox" & iCounter).Enabled = False Else Controls("CheckBox" & iCounter).Caption = rng(iCounter) End If Next iCounter End Sub StandardModule: Modul1 Sub CallForm() frmEnabled.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalte bei CheckBox-Klick aus- und einblenden

Problem:	Bei aktivierter CheckBox soll in allen Blättern Spalte H sichtbar, bei deaktivierter CheckBox unsichtbar sein.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CheckBox1_Click() Dim wks As Worksheet For Each wks In Worksheets If wks.Index <> 1 Then wks.Columns("H").Hidden = Not CheckBox1.Value End If Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

CheckBox-Werte eines Blattes in ein anderes Blatt eintragen

Problem: Bei jeder Änderung der CheckBox-Wertes des aktiven Blattes sollen die aktivierten CheckBox-Aufschriften im Bereich A1:A5 der Tabelle2 eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CheckBox1_Click()
    Call Eintragen
End Sub

Private Sub CheckBox2_Click()
    Call Eintragen
End Sub

Private Sub CheckBox3_Click()
    Call Eintragen
End Sub

Private Sub CheckBox4_Click()
    Call Eintragen
End Sub

Private Sub CheckBox5_Click()
    Call Eintragen
End Sub

StandardModule: Modul1

Sub Eintragen()
    Dim iCounter As Integer, iRow As Integer
    With Worksheets("Tabelle2")
        .Range("A1:A5").ClearContents
        For iCounter = 1 To 5
            If Worksheets("Tabelle1") _
                .OLEObjects(iCounter).Object.Value = True Then
                iRow = iRow + 1
                .Cells(iRow, 1).Value = _
                    Worksheets("Tabelle1") _
                    .OLEObjects(iCounter).Object.Caption
            End If
        Next iCounter
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

CheckBox-Aktivierung in Abhängigkeit von anderen CheckBoxes

Problem: CheckBox3 darf sich nur dann aktivieren lassen, wenn CheckBox1 und CheckBox2 ebenfalls aktiviert sind.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CheckBox3_Click()
    If CheckBox1.Value = False Or _
        CheckBox2.Value = False Then _
        CheckBox3.Value = False
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Rückfrage vor CheckBox-Werteänderung

Problem:	Bei CheckBox1 soll bei Verneinung der beim Anklicken gezeigten MsgBox-Frage der alte Wert beibehalten, bei CheckBox2 soll er geändert werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CheckBox1_MouseDown(_ ByVal Button As Integer, _ ByVal Shift As Integer, _ ByVal X As Single, ByVal Y As Single) If MsgBox("Aktion wirklich durchführen?", _ vbQuestion + vbYesNo) = vbYes Then MsgBox "Führe Aktion durch!" CheckBox1.Value = Not CheckBox1.Value End If End Sub Private Sub CheckBox2_Change() If MsgBox("Aktion wirklich durchführen?", _ vbQuestion + vbYesNo) = vbNo Then Exit Sub Else MsgBox "Führe Aktion durch!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

CheckBoxes eines vorgegebenen Bereiches prüfen

Problem:	Die CheckBoxes im Bereich A1:F10 sollen auf ihren Wert hin geprüft werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CheckBoxActivate() Dim arr() As String Dim ole As OLEObject Dim iCounter As Integer For Each ole In ActiveSheet.OLEObjects If Not Intersect(_ ole.BottomRightCell, Range("A1:F10")) Is Nothing And _ TypeName(ole.Object) = "CheckBox" Then If ole.Object.Value = True Then iCounter = iCounter + 1 ReDim Preserve arr(1 To iCounter) arr(iCounter) = ole.Name End If End If Next ole If iCounter = 0 Then Beep MsgBox "Keine aktivierten CheckBoxes gefunden!" Exit Sub End If For iCounter = 1 To UBound(arr) MsgBox arr(iCounter) Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

CheckBox mit Häkchen statt mit Kreuz

Problem:	Wie kann ich in ein Tabellenblatt eine CheckBox mit Häkchen statt mit einem Kreuz integrieren?
Lösung:	Den nachfolgenden Code in ein Standardmodul einfügen und dem selbst erstellten Shape-Objekt zuweisen.
VBA-Code:	<pre>StandardModule: basMain Sub KreuzWeg() ActiveSheet.Shapes(1).Ungroup With ActiveSheet.TextBoxes(1) If .Text = "X" Then .Text = "" Else .Text = "X" End If End With ActiveSheet.Shapes.Range(Array("Kreuz", "CheckBox")).Group End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktivieren mehrerer CheckBoxes unterbinden

Problem: Die Position der angeklickten CheckBox soll festgestellt und mehrfaches Anklicken in einer Zeile verhindert werden. Optionsfelder sollen keine eingesetzt werden.

Lösung: Den nachfolgenden Code in ein Standardmodul eingeben und den CheckBoxes aus der Formular-Symbolleiste zuweisen.

VBA-Code: StandardModule: basMain

```
Sub MsgBox_Frage()  
    Dim rng As Range  
    Set rng = Range(ActiveSheet.CheckBoxes _  
        (Application.Caller).LinkedCell)  
    If WorksheetFunction.Sum(Rows(rng.Row)) > 1 Then  
        MsgBox "Es ist nur eine Bewertung pro Frage zulässig;" & _  
            "bitte entscheiden Sie sich!", vbCritical, "Zu oft geklickt!"  
        ActiveSheet.CheckBoxes(Application.Caller).Value = xlOff  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

10 UserForm-CheckBoxes als Serie aktivieren

Problem: Wie kann ich 20 UserForm-TextBoxes auf Schaltflächendruck aktivieren bzw. deaktivieren?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmCheckBoxes

Private Sub cmdFalse_Click()
    Call Switch(False)
End Sub

Private Sub cmdTrue_Click()
    Call Switch(True)
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 10
        Controls("CheckBox" & iCounter).Caption = "CheckBox" & iCounter
    Next iCounter
End Sub

Sub Switch(bln As Boolean)
    Dim iCounter As Integer
    For iCounter = 1 To 10
        Controls("CheckBox" & iCounter).Value = bln
    Next iCounter
End Sub

StandardModule: Modul1

Sub CallForm()
    frmCheckBoxes.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in Zelle eintragen, wenn CheckBox aktiviert

Problem: Wie kann ich in die Nebenzelle einer CheckBox Werte eintragen lassen, wenn die CheckBox aktiviert wird?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CheckBox1_Click()
    ActiveCell.Select
    With Range("A2")
        If CheckBox1.Value = True Then
            .Value = 2.5
        Else
            .Value = ""
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Aktivierung einer UserForm-CheckBox Häkchen in Zelle eintragen

Problem:	Wie kann ich bei Aktivierung einer UserForm-CheckBox ein Häkchen in eine bestimmte Zelle eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmHaekchen Private Sub chbhaekchen_Click() With Range("A1") On Error Resume Next .Font.Name = "Wingdings" If chbHaekchen.Value = True Then .Value = "ü" Else .Value = "" End If End With End Sub Private Sub cmdWeiter_Click() Unload Me End Sub StandardModule: basMain Sub CallForm() frmHaekchen.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte von CheckBoxes in einer UserForm abfragen

Problem: Wie kann ich über eine Schaltfläche abfragen, welche CheckBoxes einer UserForm aktiviert sind?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmCheckBoxesValue

Private Sub cmdAbfragen_Click()
    Dim cnt As Control
    Dim iValues As Integer
    Dim arr() As Boolean
    For Each cnt In Controls
        If Left(cnt.Name, 8) = "CheckBox" Then
            iValues = iValues + 1
            ReDim Preserve arr(iValues)
            arr(iValues) = cnt.Value
        End If
    Next cnt
    For iValues = 1 To UBound(arr)
        If arr(iValues) = True Then
            MsgBox "CheckBox Nr. " & iValues & " aktiviert!"
        End If
    Next iValues
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmCheckBoxesValue.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl von CheckBoxes in einer UserForm ermitteln

Problem: Wie kann ich ermitteln, wieviel CheckBoxes sich in einer UserForm befinden?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmCheckBoxWerte

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub cmdCount_Click()
    Dim cnt As Control
    Dim iCount As Integer
    For Each cnt In Controls
        If TypeName(cnt) = "CheckBox" Then iCount = iCount + 1
    Next cnt
    MsgBox "Diese UserForm besitzt " & iCount & " CheckBoxes!"
End Sub

StandardModule: basMain

Sub CallForm()
    frmCheckBoxWerte.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

CheckBoxes in UserForms in eine Schleife einbinden

Problem: Wie kann ich über eine VBA-Schleife den ersten 5 CheckBoxes einer Serie von 8 Stück einen Wert zuweisen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmCheckBoxWerte

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 5
        Controls("CheckBox" & iCounter).Value = True
    Next iCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmCheckBoxWerte.Show
End Sub
```

Laufwerke im Dialog zur Auswahl anzeigen lassen

Problem:	Wie kann ich mir alle verfügbaren Festplatten und CD-Rom-Laufwerke im Dialog zur Auswahl anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain <pre>Sub GetDrives() Dim frmNew As Object Dim chb As MSForms.CheckBox Dim cmd As MSForms.CommandButton Dim arr() As String Dim iCounter As Integer, iDrive As Integer Dim iTop As Integer Dim sCode As String On Error Resume Next For iCounter = 65 To 90 ChDrive Chr(iCounter) If Err > 0 Then Err.Clear Else iDrive = iDrive + 1 ReDim Preserve arr(1 To iDrive) arr(iDrive) = Chr(iCounter) End If Next iCounter Application.VBE.MainWindow.Visible = False Set frmNew = ThisWorkbook.VBProject.VBComponents("frmDrives") If Err = 0 Then GoTo ERRORHANDLER On Error GoTo 0 Set frmNew = ThisWorkbook.VBProject.VBComponents.Add(3) iTop = 5 For iCounter = 1 To UBound(arr) Set chb = frmNew.Designer.Controls.Add("forms.CheckBox.1") With chb .Top = iTop .Left = 5 .Width = 100 .Caption = "Laufwerk " & arr(iCounter) End With iTop = iTop + 20 Next iCounter Set cmd = frmNew.Designer.Controls.Add("forms.CommandButton.1") With cmd .Top = iTop .Left = 5 .Width = 100 .Height = 25 .Caption = "OK" .Name = "cmdOK" End With iTop = iTop + 30 Set cmd = frmNew.Designer.Controls.Add("forms.CommandButton.1") With cmd .Top = iTop .Left = 5 .Width = 100 .Height = 25 .Caption = "Abbrechen" .Name = "cmdCancel" End With With frmNew .Properties("Width") = 118 .Properties("Height") = iTop + 50 .Properties("Caption") = "Laufwerke" .Properties("Name") = "frmDrives" End With sCode = "Private Sub cmdCancel_Click" & vbCrLf sCode = sCode & " Unload Me" & vbCrLf sCode = sCode & "End Sub" & vbCrLf & vbCrLf sCode = sCode & "Private Sub cmdOK_Click" & vbCrLf sCode = sCode & " Dim sCode as String" & vbCrLf sCode = sCode & " Dim iCounter As Integer" & vbCrLf sCode = sCode & " sCode = ""Ausgewählte Laufwerke: "" & vbCrLf & vbCrLf sCode = sCode & " For iCounter = 1 to " & UBound(arr) & vbCrLf sCode = sCode & " If Controls(""CheckBox"" & iCounter).Value" sCode = sCode & " = True Then" & vbCrLf sCode = sCode & " sCode = sCode & Controls(""CheckBox"" sCode = sCode & " & iCounter).Caption & vbCrLf" & vbCrLf sCode = sCode & " End If" & vbCrLf sCode = sCode & " Next iCounter" & vbCrLf sCode = sCode & " MsgBox sCode" & vbCrLf sCode = sCode & "End Sub" & vbCrLf ThisWorkbook.VBProject.VBComponents("frmDrives").CodeModule.AddFromString sCode ERRORHANDLER: VBA.UserForms.Add(frmNew.Name).Show ThisWorkbook.VBProject.VBComponents.Remove ThisWorkbook.VBProject.VBComponents(frmNew.Name) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Passwort vergeben und bei CheckBox-Aktivität abfragen

Problem: Wie kann ich ein Paßwort festlegen und bei Anklicken einer CheckBox abfragen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub CheckBox1_Click()
    Dim sWord As String
    If CheckBox1.Value = True Then
        sWord = InputBox("Passwort:")
        If sWord = "" Then Exit Sub
        If sWord = Range("IV1").Value Then
            MsgBox "Passwort ist OK"
        Else
            MsgBox "Falsches Passwort"
        End If
    End If
End Sub

StandardModule: basMain

Sub SetPassword()
    Dim sWord As String
    sWord = InputBox("Passwort:")
    If sWord = "" Then Exit Sub
    With Range("IV1")
        .Value = sWord
        .NumberFormat = ";;;;"
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Aktivieren einer CheckBox Häkchen in Zelle eintragen

Problem:	Wie kann ich beim Aktivieren einer CheckBox in die danebenliegende Zelle ein Häkchen eintragen bzw. löschen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle12 Private Sub CheckBox1_Click() On Error Resume Next With Range("A1") If CheckBox1.Value = True Then .Value = "ü" Else .Value = "" End If End With On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle CheckBoxes einer UserForm ansprechen

Problem: Wie kann ich alle CheckBoxes einer UserForm in eine For/NextSchleife einbinden?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmCheckBox

Private Sub CommandButton1_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim cnt As Control
    Dim iCounter As Integer
    For Each cnt In frmCheckBox.Controls
        If UCase(TypeName(cnt)) = "CHECKBOX" Then
            iCounter = iCounter + 1
            cnt.Caption = "MeineCheckbox " & iCounter
        End If
    Next cnt
End Sub

StandardModule: basMain

Sub CallForm()
    frmCheckBox.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summenbildung bei Checkboxaktivierung

Problem:	Bei aktivierung einer Checkbox in Spalte A soll in Spalte B die Summe der Werte dieser Zeile ab Spalte C eingetragen werden. Wie löse ich das über VBA-Code?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CheckBox1_Change() If CheckBox1.Value = True Then Cells(1, 2).Formula = "=sum(C1:IV1)" Else Cells(1, 2) = "" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Häkchen eintragen

Problem: Wie kann ich bei Anklicken einer Checkbox in einer Zelle der Arbeitsmappe eine Häkchen eintragen bzw. löschen lassen?

Lösung: Die Ereignis-Programmierung befindet sich im Standardmodul

VBA-Code:

```
StandardModule: basMain

Sub CheckBox1_Click()
    With Range("A1")
        .Font.Name = "Wingdings"
        If ActiveSheet.CheckBoxes(1).Value = xlOn Then
            .Value = "ü"
        Else
            .Value = ""
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zustand von Tabellenblatt-Checkboxes ermitteln

Problem: In das Tabellenblatt integrierte CheckBoxes rufe ich im VBA-Code als "OleObjects" auf. "OleObjects" hat aber keine "Value"-Eigenschaft. Wie ermittle ich den Zustand der CheckBoxes?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdChb_Click()
    Dim oOle As OLEObject
    For Each oOle In OLEObjects
        If TypeName(oOle.Object) = "CheckBox" Then
            MsgBox oOle.Name & ": " & oOle.Object.Value
        End If
    Next oOle
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In einem Dialog ausgewählte Spalten drucken

Problem: Ich habe eine Dialogbox mit einer Reihe von CheckBoxes, die Spalten in einem Arbeitsblatt (Name, Vorname, PLZ, Ort etc.) darstellten. Nach Beendigung des Dialogs sollen die Spalten, deren Checkbox aktiviert wurde, gedruckt werden. Wie mache ich das?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmPrint.Show
End Sub

Sub SetPrintPreview(arr As Variant)
    Dim iCounter As Integer
    For iCounter = 1 To 4
        If arr(iCounter) = True Then
            Columns(iCounter).Hidden = False
        Else
            Columns(iCounter).Hidden = True
        End If
    Next iCounter
    ActiveSheet.PrintPreview
    Columns.Hidden = False
    ActiveSheet.DisplayPageBreaks = False
End Sub

ClassModule: frmPrint

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim iCounter As Integer
    Dim arrBln(1 To 4) As Boolean
    For iCounter = 1 To 4
        If Controls("CheckBox" & iCounter).Value = True Then
            arrBln(iCounter) = True
        End If
    Next iCounter
    Unload Me
    Call SetPrintPreview(arrBln)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen gem. UserForm-Auswahl in Tabelle suchen

Problem: Über UserForm-ComboBoxes werden Name und Vorname gesucht. Die Fundzeile wird markiert.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmNames

Private Sub cboNames_Change()
    Call selectname
End Sub

Private Sub cboSurnames_Change()
    Call selectname
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    cboNames.List = Range("A1").CurrentRegion.Columns(1).Value
    cboSurnames.List = Range("A1").CurrentRegion.Columns(2).Value
    cboNames.ListIndex = 0
    cboSurnames.ListIndex = 0
End Sub

Private Sub selectname()
    Dim iRow As Integer
    iRow = 1
    Do Until IsEmpty(Cells(iRow, 1))
        If Cells(iRow, 1).Value = cboNames.Value And _
            Cells(iRow, 2).Value = cboSurnames.Value Then
            Rows(iRow).Select
        End If
        iRow = iRow + 1
    Loop
End Sub

StandardModule: Modul1

Sub CallForm()
    frmNames.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Variable Spalten in UserForm-ComboBox einlesen

Problem: In eine UserForm-ComboBox sollen die in Spalte M genannten Spalten eingelesen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmColumns

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim arr() As Variant
    Dim iRow As Integer, iCol As Integer, iRowL As Integer, iColL As Integer
    iRowL = WorksheetFunction.CountA(Columns(1))
    iColL = WorksheetFunction.CountA(Columns(13))
    ReDim arr(1 To iRowL, 1 To iColL)
    For iRow = 1 To iRowL
        For iCol = 1 To iColL
            arr(iRow, iCol) = Cells(iRow, Cells(iCol, 13).Value)
        Next iCol
    Next iRow
    With cboColumns
        .ColumnCount = iColL
        .List = arr
        .ListIndex = 0
    End With
End Sub

StandardModule: Modul1

Sub CallForm()
    frmColumns.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox mit der Funktion des Autofilters

Problem: Eine ComboBox in einem Tabellenblatt soll die Funktion des Autofilters erfüllen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub ComboBox1_Change()
    Dim iRow As Integer
    If ComboBox1.Value = "" Then Exit Sub
    Rows.Hidden = False
    iRow = 2
    Do Until IsEmpty(Cells(iRow, 1))
        If Cells(iRow, 1).Value <> CInt(ComboBox1.Value) Then
            Rows(iRow).Hidden = True
        End If
        iRow = iRow + 1
    Loop
End Sub

Private Sub ComboBox1_DropButtonClick()
    Dim col As New Collection
    Dim iRow As Integer
    ComboBox1.Clear
    Range("A1").CurrentRegion.Sort _
        key1:=Range("A2"), order1:=xlAscending, header:=xlYes
    iRow = 2
    On Error Resume Next
    Do Until IsEmpty(Cells(iRow, 1))
        col.Add CStr(Cells(iRow, 1).Value), CStr(Cells(iRow, 1).Value)
        iRow = iRow + 1
    Loop
    On Error GoTo 0
    For iRow = 1 To col.Count
        ComboBox1.AddItem col(iRow)
    Next iRow
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Darstellung der unterschiedlichen Auslesemethoden von ComboBoxes

Problem:	Die Unterschiede beim Auslesen von ComboBox- und DropDown-Felder-Werten als Indizes und Werte werden dargestellt.
Lösung:	Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalte 2 ohne Leerzellen und Doppel in ComboBox

Problem: In eine UserForm-ComboBox sollen die Werte aus Spalte 2 ohne Leerzellen und ohne Doppel aufgerufen werden

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmValues

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim wks As Worksheet
    Dim vRow As Variant
    Dim iRow As Integer, iRowT As Integer, iRowL As Integer
    Application.ScreenUpdating = False
    Set wks = ActiveSheet
    iRowL = Cells(Rows.Count, 2).End(xlUp).Row
    Workbooks.Add
    For iRow = 3 To iRowL
        vRow = Application.Match(wks.Cells(iRow, 2).Value, Columns(1), 0)
        If IsError(vRow) Then
            iRowT = iRowT + 1
            Cells(iRowT, 1).Value = wks.Cells(iRow, 2).Value
        End If
    Next iRow
    Range("A1").CurrentRegion.Sort _
        key1:=Range("A1"), order1:=xlAscending, header:=xlNo
    With cboValues
        .List = Range("A1").CurrentRegion.Value
        If .ListCount > 0 Then .ListIndex = 0
    End With
    ActiveWorkbook.Close savechanges:=False
    Application.ScreenUpdating = True
End Sub

StandardModule: Modul1

Sub CallForm()
    frmValues.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in Tabellenblatt-DropDownfeld ein- und auslesen

Problem: Mit der ersten Schaltfläche soll das DropDownfeld gefüllt, mit der zweiten soll der aktuelle Werte ausgelesen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn den Schaltflächen zu.

VBA-Code:

```
StandardModule: Modul1

Sub SetValues()
    Dim iMonth As Integer
    With ActiveSheet.DropDowns(1)
        .RemoveAllItems
        For iMonth = 1 To 12
            .AddItem Format(DateSerial(1, iMonth, 1), "mmm")
        Next iMonth
        .ListIndex = 1
    End With
End Sub

Sub GetValue()
    With ActiveSheet.DropDowns(1)
        If .ListIndex > 0 Then
            MsgBox .List(.ListIndex)
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sprache von ComboBox-Listen nach Sprachauswahl festlegen

Problem:	Die Sprachinhalte einer ComboBox sollen über eine weitere ComboBox festgelegt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmLanguage Private Sub cboLanguage_Change() With cboMonths .List = Range("A1").CurrentRegion.Columns(cboLanguage.ListIndex + 1).Value .ListIndex = 0 End With End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim arr As Variant arr = Array("Deutsch", "Englisch", "Französisch", "Italienisch", "Spanisch") With cboLanguage .List = arr .ListIndex = 0 End With End Sub StandardModule: Modul1 Sub CallForm() frmLanguage.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einlesen des aktuellen Datums und einer Datumsreihe

Problem:	Das Datum aus Zelle A1 soll in eine UserForm-TextBox übernommen werden. Die Datumsreihe wird in eine ComboBox eingelesen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmDate Private Sub cboDate_Change() txtDate.Value = cboDate.Value lblDate.Caption = cboDate.Value End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub UserForm_Initialize() cboDate.List = Range("A1").CurrentRegion.Value cboDate.ListIndex = 0 End Sub StandardModule: Modul1 Sub CallForm() frmDate.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über ComboBox in Datenbestand blättern

Problem:	Über die ComboBox soll in den Artikeln aus dem Datenblatt geblättert werden können.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamisch Blatt-ComboBoxes

Problem: Bei Auswahl des Blattes "Ergebnis" werden die ComboBoxes neu eingelesen. Eine Werteänderung einer ComboBoxes wirkt sich auf die andere aus.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cboHersteller_Change()
    Dim iRow As Integer
    cboProdukt.Clear
    iRow = 2
    With Worksheets("Listen")
        Do Until IsEmpty(.Cells(iRow, 1))
            If .Cells(iRow, 1).Value = cboHersteller.Value Then
                cboProdukt.AddItem .Cells(iRow, 2).Value
            End If
            iRow = iRow + 1
        Loop
    End With
    If cboProdukt.ListCount > 0 Then
        cboProdukt.ListIndex = 0
    End If
End Sub

Private Sub cboProdukt_Change()
    Dim iRow As Integer
    iRow = 2
    With Worksheets("Listen")
        Do Until IsEmpty(.Cells(iRow, 1))
            If .Cells(iRow, 1).Value = cboHersteller.Value And _
                .Cells(iRow, 2).Value = cboProdukt.Value Then
                Cells(2, 3).Value = .Cells(iRow, 3).Value
                Exit Do
            End If
            iRow = iRow + 1
        Loop
    End With
End Sub

Private Sub Worksheet_Activate()
    Dim col As New Collection
    Dim iRow As Integer
    cboHersteller.Clear
    iRow = 2
    On Error Resume Next
    With Worksheets("Listen")
        Do Until IsEmpty(.Cells(iRow, 1))
            col.Add .Cells(iRow, 1).Value, .Cells(iRow, 1).Value
            iRow = iRow + 1
        Loop
    End With
    On Error GoTo 0
    For iRow = 1 To col.Count
        cboHersteller.AddItem col(iRow)
    Next iRow
    If cboHersteller.ListCount > 0 Then
        cboHersteller.ListIndex = 0
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über Schaltflächenklick in ComboBox scrollen

Problem:	Über Schaltflächenklick soll in einer ComboBox gescrollt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdSetCbo_Click() With cboMonths If .ListIndex < .ListCount - 1 Then .ListIndex = .ListIndex + 1 Else .ListIndex = 0 End If End With End Sub</pre>



Schaltfläche anlegen und Daten an ComboBox sortiert übergeben

Problem:	Die Werte aus den durch Schaltflächenklick in Zeile 2 ausgewählten Spalten sollen in der ComboBox in Zeile 1 sortiert angezeigt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Sub CreateButtons() Dim btn As Button Dim iBtn As Integer ActiveSheet.Buttons("NewButtons").Delete For iBtn = 1 To 26 Set btn = ActiveSheet.Buttons.Add(_ Cells(2, iBtn).Left, _ Cells(2, iBtn).Top, _ Cells(2, iBtn).Width, _ Cells(2, iBtn).Height) btn.Caption = Chr(iBtn + 64) btn.OnAction = "FillCbo" Next iBtn End Sub Sub ShowAll() Dim cbo As Object Dim arr() As Variant Dim iRow As Integer, iCol As Integer, iCounter As Integer iRow = 3 iCol = 1 Do Until IsEmpty(Cells(iRow, iCol)) Do Until IsEmpty(Cells(iRow, iCol)) iCounter = iCounter + 1 ReDim Preserve arr(1 To iCounter) arr(iCounter) = Cells(iRow, iCol).Value iRow = iRow + 1 Loop iRow = 3 iCol = iCol + 1 Loop If iCounter > 0 Then Call QuickSort(arr) With wksValues.cboValues .List = arr .ListIndex = 0 End With End If End Sub Sub FillCbo() Dim cbo As Object Dim arr() As Variant Dim iCounter As Integer, iRow As Integer, iCol As Integer Set cbo = wksValues.cboValues iCol = wksValues.Buttons(Application.Caller).TopLeftCell.Column iRow = Cells(Rows.Count, iCol).End(xlUp).Row - 2 cbo.Clear If iRow > 2 Then ReDim arr(1 To iRow) For iCounter = 3 To iRow + 2 arr(iCounter - 2) = Cells(iCounter, iCol).Value Next iCounter Call QuickSort(arr) With cbo .List = arr .ListIndex = 0 End With End If End Sub Sub QuickSort(ByRef VA_array, Optional V_Low1, Optional V_high1) On Error Resume Next Dim V_Low2, V_high2, V_loop As Integer Dim V_val1, V_val2 As Variant If IsMissing(V_Low1) Then V_Low1 = LBound(VA_array, 1) End If If IsMissing(V_high1) Then V_high1 = UBound(VA_array, 1) End If V_Low2 = V_Low1 V_high2 = V_high1 V_val1 = VA_array((V_Low1 + V_high1) / 2) While (V_Low2 <= V_high2) While (VA_array(V_Low2) < V_val1 And _ V_Low2 < V_high1) V_Low2 = V_Low2 + 1 Wend While (VA_array(V_high2) > V_val1 And _ V_high2 > V_Low1) V_high2 = V_high2 - 1 Wend If (V_Low2 <= V_high2) Then V_val2 = VA_array(V_Low2) VA_array(V_Low2) = VA_array(V_high2) VA_array(V_high2) = V_val2 V_Low2 = V_Low2 + 1 V_high2 = V_high2 - 1 End If Wend If (V_high2 > V_Low1) Then Call _ QuickSort(VA_array, V_Low1, V_high2) If (V_Low2 < V_high1) Then Call _ QuickSort(VA_array, V_Low2, V_high1) End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Artikel mit vorgegebenem Wortanfang in ComboBox listen

Problem:	Bei Eingabe einer Zahl in Spalte A ab A3 soll eine UserForm-ComboBox mit den Artikeln erscheinen, deren Nummer im Blatt Material mit dieser Zahl beginnt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 1 Then Exit Sub If Target.Row < 3 Then Exit Sub If IsEmpty(Target) Then Exit Sub Application.EnableEvents = False On Error GoTo ERRORHANDLER frmSelect.Show ERRORHANDLER: Application.EnableEvents = True End Sub ClassModule: frmSelect Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim iRow As Integer With Worksheets("Material") iRow = WorksheetFunction.Match(CInt(cboSelect.Value), .Columns(1), 0) ActiveCell.Value = .Cells(iRow, 1).Value ActiveCell.Offset(0, 1).Value = .Cells(iRow, 2).Value ActiveCell.Offset(0, 2).Value = .Cells(iRow, 3).Value End With Unload Me End Sub Private Sub UserForm_Initialize() Dim iRow As Integer iRow = 4 With Worksheets("Material") Do Until IsEmpty(.Cells(iRow, 1)) If Left(.Cells(iRow, 1).Value, 1) = CStr(ActiveCell.Value) Then cboSelect.AddItem .Cells(iRow, 1).Value End If iRow = iRow + 1 Loop End With If cboSelect.ListCount > 0 Then cboSelect.ListIndex = 0 Else Unload Me End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prozentwerte in UserForm-ComboBox einlesen

Problem: In eine UserForm-ComboBox sollen Prozentwerte eingelesen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmPercent

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iRow As Integer
    iRow = 1
    Do Until IsEmpty(Cells(iRow, 1))
        cboPercent.AddItem Cells(iRow, 1).Text
        iRow = iRow + 1
    Loop
    With cboPercent
        If .ListCount > 0 Then
            .ListIndex = 0
        End If
    End With
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmPercent.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox mit LinkedCell-Eigenschaft in aktivem Blatt erstellen

Problem: Im aktiven Blatt soll eine neue ComboBox mit LinkedCell- und ListFillRange-Zuweisung erstellt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub AddComboBox()
    Dim oObj As OLEObject
    Dim oCbox As MSForms.ComboBox
    Dim cell As Range
    Set oObj = ActiveSheet.OLEObjects.Add( _
        ClassType:="Forms.ComboBox.1", _
        Link:=False, _
        DisplayAsIcon:=False, _
        Left:=20, _
        Top:=50, _
        Width:=150, _
        Height:=20)
    Set oCbox = oObj.Object
    oCbox.LinkedCell = Range("F1")
    oObj.ListFillRange = Range("G1:G12").Address
    oCbox.ListIndex = 0
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox in Arbeitsblatt erstellen und füllen

Problem: Im aktiven Blatt soll eine ComboBox mit den Monatsnamen als Inhalt erstellt werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code: StandardModule: Modul1

```
Sub NewCboBox()  
    Dim cbo As OLEObject  
    Dim iCounter As Integer  
    Range("A1").Select  
    Set cbo = ActiveSheet.OLEObjects.Add( _  
        ClassType:="Forms.ComboBox.1", _  
        Link:=False, _  
        DisplayAsIcon:=False, _  
        Left:=40, _  
        Top:=40, _  
        Width:=170, _  
        Height:=20)  
    For iCounter = 1 To 12  
        cbo.Object.AddItem Format(DateSerial(1, iCounter, 1), "mmm")  
    Next iCounter  
    cbo.Object.ListIndex = 0  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

TextBox-Werte in Abhängigkeit zu ComboBox-Auswahl

Problem:	Die TextBoxes in Spalte C sollen entsprechend der ComboBox-Auswahl in Spalte A auf die entsprechenden Daten im Blatt Monate zugreifen. Lösung über die LinkedCell-Eigenschaften und die Formeln in C1:C3.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Filtern von Daten und Übernahme in UserForm-ComboBox

Problem: Eine Tabelle soll über UserForm gefiltert und die gefilterten Daten aus Spalte E sollen in eine ComboBox übernommen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: Modul1

Sub DialogAufruf()
    frmFilter.Show
End Sub

ClassModule: frmFilter

Private Sub cmdCancel_Click()
    Unload Me
    ActiveSheet.AutoFilterMode = False
End Sub

Private Sub cmdOK_Click()
    Dim iRow As Integer
    With Range("A1")
        .AutoFilter Field:=1, Criteria1:="=*2*"
        .AutoFilter Field:=2, Criteria1:="=*1*"
    End With
    iRow = 2
    Do Until IsEmpty(Cells(iRow, 5))
        If Rows(iRow).Hidden = False Then
            cboFilter.AddItem Cells(iRow, 5).Value
        End If
        iRow = iRow + 1
    Loop
    If cboFilter.ListCount > 0 Then
        cboFilter.ListIndex = 0
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Aktivieren eines Arbeitsblattes ComboBox neu einlesen

Problem: Bei Aktivieren dieses Arbeitsblattes sollen die Orte aus dem Arbeitsblatt "Orte" in die ComboBox neu eingelesen werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Activate()
    ComboBox1.List = _
        Worksheets("Orte").Range("A1") _
        .CurrentRegion.Columns("B:C").Value
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Im TextFeld einer UserForm-ComboBox Wert aus 2 Spalten anzeigen

Problem: Im Textfeld einer UserForm-ComboBox soll immer der Wert aus Spalte 1 zusammen mit dem Wert aus Spalte 2 angezeigt werden, ohne dass das Auslesen der Auswahl beeinträchtigt wird.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmTowColumns

Private Sub cboTwoColumns_Change()
    Dim iCounter As Integer, iRow As Integer
    Dim sSelect As String
    With cboTwoColumns
        If .ListIndex >= 0 Then
            sSelect = .List(.ListIndex, 0) & " - " & .List(.ListIndex, 1)
            iRow = .ListIndex
            cboTwoColumns.Text = sSelect
        End If
        For iCounter = 1 To 3
            Controls("TextBox" & iCounter).Text = .List(iRow, iCounter - 1)
        Next iCounter
    End With
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    cboTwoColumns.List = Range("A1").CurrentRegion.Value
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmTowColumns.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox-Breiten während der Laufzeit festlegen

Problem: Die Breite einer UserForm-ComboBox soll während der Laufzeit in Abhängigkeit vom breitesten Inhalt festgelegt werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: Modul1

Sub DialogAufruf()
    frmListWidth.Show
End Sub

ClassModule: frmListWidth

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdListA_Click()
    Dim iRow As Integer, iMax As Integer
    Dim sTxt As String
    cboList.Clear
    For iRow = 1 To 12
        sTxt = Format(DateSerial(1, iRow, 1), "mmm")
        cboList.AddItem sTxt
        If Len(sTxt) > iMax Then iMax = Len(sTxt)
    Next iRow
    cboList.Width = iMax * 7
    cboList.ListIndex = 0
End Sub

Private Sub cmdListB_Click()
    Dim iRow As Integer, iMax As Integer
    Dim sTxt As String
    cboList.Clear
    For iRow = 1 To 31
        sTxt = "Asbach, den " & _
            Format(DateSerial(Year(Date), 1, iRow), _
                "dd. mmm yyyy")
        cboList.AddItem sTxt
        If Len(sTxt) > iMax Then iMax = Len(sTxt)
    Next iRow
    cboList.Width = iMax * 7
    cboList.ListIndex = 0
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte ohne Doppel in UserForm einlesen und löschen

Problem:	Spalte A soll ohne Duplikate in eine UserForm-ComboBox eingelesen werden.
Lösung:	Von den in der ComboBox ausgewählten Elementen soll die Zeile mit den Duplikaten gelöscht werden.
VBA-Code:	<pre>ClassModule: frmDuplikate Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim iRow As Integer, iRowL As Integer iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = iRowL To 2 Step -1 If Cells(iRow, 1) = cboDuplikate.Value And _ WorksheetFunction.CountIf(Range("A1:A" & iRow - 1), _ Cells(iRow, 1)) > 0 Then Rows(iRow).Delete End If Next iRow End Sub Private Sub UserForm_Initialize() Dim col As New Collection Dim iRow As Integer iRow = 1 On Error Resume Next Do Until IsEmpty(Cells(iRow, 1)) col.Add Cells(iRow, 1), Cells(iRow, 1) iRow = iRow + 1 Loop For iRow = 1 To col.Count cboDuplikate.AddItem col(iRow) Next iRow cboDuplikate.ListIndex = 0 End Sub StandardModule: basMain Sub CallForm() frmDuplikate.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Liste ohne Doppelte Werte in ComboBox einlesen

Problem: Die Liste aus Spalte A soll ohne Doppelte in eine UserForm-ComboBox eingelesen werden. Nach Auswahl eines Namens soll in einem Label die zu diesem Namen gehörige Summe aus Spalte B übernommen werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: frmSumme

Private Sub cboNamen_Change()
    lblSumme.Caption = _
        Format(WorksheetFunction _
            .SumIf(Columns(1), cboNamen.Value, Columns(2)), "#,##0")
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim col As New Collection
    Dim iRow As Integer
    iRow = 1
    On Error Resume Next
    Do Until IsEmpty(Cells(iRow, 1))
        col.Add Cells(iRow, 1), Cells(iRow, 1)
        If Err = 0 Then
            cboNamen.AddItem Cells(iRow, 1)
        Else
            Err.Clear
        End If
        iRow = iRow + 1
    Loop
    On Error GoTo 0
    cboNamen.ListIndex = 0
End Sub

StandardModule: Modul1

Sub CallForm()
    frmSumme.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in ComboBox einlesen, suchen, editieren, ausgeben

Problem: Die Suchbegriffe aus Spalte F sollen in eine ComboBox eingelesen werden. Bei Auswahl des Begriffes in der ComboBox soll dieser in eine TextBox zum Editieren übernommen und bei Betätigung der OK-Schaltfläche in das Blatt zurückgeschrieben werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: frmSuchen

Private Sub cboNamen_Change()
    Dim rng As Range
    Set rng = Columns(1).Find(cboNamen.Value, lookat:=xlPart, LookIn:=xlValues)
    If Not rng Is Nothing Then
        txtEdit.Text = rng.Text
        txtEdit.Tag = rng.Address
    End If
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Range(txtEdit.Tag).Value = txtEdit.Text
End Sub

Private Sub UserForm_Initialize()
    cboNamen.List = Range("F1").CurrentRegion.Value
    cboNamen.ListIndex = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmSuchen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Artikelauswahl in ComboBox Preis ermitteln

Problem:	Aus Tabelle2 sollen die Werte der Spalte A in die ComboBox übernommen werden. Zu dem in der ComboBox ausgewählten Wert soll in Zelle E4 der zugehörige Preis angezeigt werden.
Lösung:	Nur anhand einer Beipspielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Selektieren von über ComboBox-Felder ausgewählten Werten

Problem:	Die DropDownfelder in Zeile 1 sollen sich bei jedem Eintrag aktualisieren und die Werte der jeweiligen Spalte ohne Berücksichtigung von Leerzeilen sortiert anzeigen. Nach Auswahl einer Zahl im DropDown-Feld soll die jeweilige Zeile ausgewählt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Excel.Range) If Target.Column <= 4 Then Call SetCbo End Sub StandardModule: basMain Sub SetCbo() Dim arr() As Variant Dim iRow As Integer, iRowL As Integer Dim iCol As Integer, iArr As Integer iCol = ActiveCell.Column If iCol > 4 Then Exit Sub iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 2 To iRowL If Not IsEmpty(Cells(iRow, iCol)) Then iArr = iArr + 1 ReDim Preserve arr(1 To iArr) arr(iArr) = Cells(iRow, iCol).Value End If Next iRow If iArr > 1 Then QuickSort arr With ActiveSheet.DropDowns(iCol) .RemoveAllItems .AddItem Cells(1, iCol) For iArr = 1 To UBound(arr) .AddItem arr(iArr) Next iArr End With End Sub Sub Auswahl() Dim drpDwn As DropDown Dim iRow As Integer, iCol As Integer iCol = Right(Application.Caller, 1) Set drpDwn = ActiveSheet.DropDowns(iCol) If drpDwn.ListIndex = 1 Then Exit Sub iRow = WorksheetFunction.Match _ (CInt(drpDwn.List(drpDwn.ListIndex)), Columns(iCol), 0) Cells(iRow, iCol).Select drpDwn.ListIndex = 1 End Sub Private Sub QuickSort(ByRef VA_array, Optional V_Low1, Optional V_high1) On Error Resume Next Dim V_Low2, V_high2, V_loop As Integer Dim V_val1, V_val2 As Variant If IsMissing(V_Low1) Then V_Low1 = LBound(VA_array, 1) End If If IsMissing(V_high1) Then V_high1 = UBound(VA_array, 1) End If V_Low2 = V_Low1 V_high2 = V_high1 V_val1 = VA_array((V_Low1 + V_high1) / 2) While (V_Low2 <= V_high2) While (VA_array(V_Low2) < V_val1 And _ V_Low2 < V_high1) V_Low2 = V_Low2 + 1 Wend While (VA_array(V_high2) > V_val1 And _ V_high2 > V_Low1) V_high2 = V_high2 - 1 Wend If (V_Low2 <= V_high2) Then V_val2 = VA_array(V_Low2) VA_array(V_Low2) = VA_array(V_high2) VA_array(V_high2) = V_val2 V_Low2 = V_Low2 + 1 V_high2 = V_high2 - 1 End If Wend If (V_high2 > V_Low1) Then Call _ QuickSort(VA_array, V_Low1, V_high2) If (V_Low2 < V_high1) Then Call _ QuickSort(VA_array, V_Low2, V_high1) End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelt vorkommende Werte in einer UserForm-Combobox listen

Problem: In einer Tabelle doppelt vorkommende Werte sollen in einer UserForm-ComboBox gelistet werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: frmSuchen

Private Sub cmdSuchen_Click()
    Dim rng As Range
    Dim iRowL As Integer, iCol As Integer
    Application.ScreenUpdating = False
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row
    iCol = Cells(1, 256).End(xlToLeft).Column + 1
    Cells(1, iCol).Formula = "=countif(A:A,A1)"
    Range(Range(Cells(1, iCol), Cells(iRowL, iCol)).Address).FillDown
    Range("A1").CurrentRegion.AutoFilter field:=iCol, Criteria1:=">1"
    Set rng = Columns(1).CurrentRegion.SpecialCells(xlCellTypeVisible)
    Workbooks.Add
    rng.Copy Range("A1")
    Rows(1).Delete
    cboDoubles.List = Range("A1").CurrentRegion.Columns(1).Value
    Application.CutCopyMode = False
    ActiveWorkbook.Close savechanges:=False
    ActiveSheet.AutoFilterMode = False
    Columns(iCol).ClearContents
    If cboDoubles.ListCount > 0 Then cboDoubles.ListIndex = 0
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmSuchen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateinamen in UserForm-ComboBox einlesen und Datei öffnen

Problem: Die Werte aus Spalte A sollen in eine UserForm-ComboBox eingelesen werden.

Lösung: Die in der ComboBox ausgewählte, durch einen Hyperlink verbundene Datei soll auf Schaltflächendruck zu öffnen sein.

VBA-Code:

```
StandardModule: Modul1

Sub CallForm()
    frmHype.Show
End Sub

ClassModule: frmHype

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Application.ScreenUpdating = False
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    Cells(cboFiles.ListIndex + 1, 1).Hyperlinks(1).Follow
ERRORHANDLER:
    Application.EnableEvents = True
    Application.ScreenUpdating = True
End Sub

Private Sub UserForm_Initialize()
    Dim iRow As Integer
    iRow = 1
    Do Until IsEmpty(Cells(iRow, 1))
        cboFiles.AddItem Cells(iRow, 1)
        iRow = iRow + 1
    Loop
    cboFiles.ListIndex = 0
End Sub
```



Dateistrukturen in UserForm einlesen und Datei öffnen

Problem:	Über eine UserForm sollen Unterverzeichnisse und Dateien gelistet und geöffnet werden können.
Lösung:	Geben Sie den nachfolgenden Code in die genannten Module ein
VBA-Code:	ClassModule: frmFiles Private Sub cboDirectories_Change() Dim fs As FileSearch Dim intCounter As Integer, iRow As Integer Dim sDir As String Columns("C").Clear cboFiles.Clear Set fs = Application.FileSearch With fs .LookIn = cmdEinlesen.Tag & cboDirectories.Value .FileName = "*.*" .Execute For intCounter = 1 To .FoundFiles.Count cboFiles.AddItem FileName(.FoundFiles(intCounter)) Cells(intCounter, 3).Value = FileName(.FoundFiles(intCounter)) Next intCounter If .FoundFiles.Count > 0 Then cboFiles.ListIndex = 0 End With End Sub Private Function FileName(strName As String) As String Dim intCounter As Integer For intCounter = Len(strName) To 1 Step -1 If Mid(strName, intCounter, 1) = "\" Then Exit For Next intCounter FileName = Right(strName, Len(strName) - intCounter) End Function Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdEinlesen_Click() Dim iRow As Integer Dim sDir As String, sFile As String Columns("A:B").ClearContents sDir = GetDirectory("Bitte Verzeichnis auswählen:") If sDir = "" Then Exit Sub If Right(sDir, 1) <> "\" Then sDir = sDir & "\" Range("A1").Value = sDir cmdEinlesen.Tag = sDir sFile = Dir(sDir & "*.*", vbDirectory) Do While sFile <> "" If sFile = "." Or sFile = ".." Then ElseIf (GetAttr(sDir & sFile) And _ vbDirectory) = vbDirectory Then cboDirectories.AddItem sFile iRow = iRow + 1 Cells(iRow, 2).Value = sFile End If sFile = Dir() Loop If cboDirectories.ListCount > 0 Then cboDirectories.ListIndex = 0 cmdOK.Enabled = True cmdEinlesen.Enabled = False End If End Sub Private Sub cmdOK_Click() Shell "explorer " & cmdEinlesen.Tag & _ cboDirectories.Value & "\" & cboFiles.Value Unload Me End Sub StandardModule: basFunctions Public Type BROWSEINFO hOwner As Long pidlRoot As Long pszDisplayName As String lpszTitle As String ulFlags As Long lpfn As Long lParam As Long iImage As Long End Type Declare Function SHGetPathFromIDList Lib "shell32.dll" _ Alias "SHGetPathFromIDListA" (ByVal pidl As Long, _ ByVal pszPath As String) As Long Declare Function SHBrowseForFolder Lib "shell32.dll" _ Alias "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long Function GetDirectory(Optional Msg) As String Dim bInfo As BROWSEINFO Dim Path As String Dim R As Long, x As Long, pos As Integer bInfo.pidlRoot = 0& If IsMissing(Msg) Then bInfo.lpszTitle = "Wählen Sie bitte einen Ordner aus." Else bInfo.lpszTitle = Msg End If bInfo.ulFlags = &H1 x = SHBrowseForFolder(bInfo) Path = Space\$(512) R = SHGetPathFromIDList(ByVal x, ByVal Path) If R Then pos = InStr(Path, Chr\$(0)) GetDirectory = Left(Path, pos - 1) Else GetDirectory = "" End If End Function StandardModule: basMain Sub DialogAufruf() frmFiles.Show End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattauswahl über eine UserForm-ComboBox

Problem: Die Blattauswahl soll über eine UserForm-ComboBox erfolgen.

Lösung: Aufruf der UserForm über ShortCut STRG+a

VBA-Code:

```
ClassModule: frmSelect

Private Sub cboSheets_Change()
    Worksheets(cboSheets.Value).Select
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim wks As Worksheet
    For Each wks In ThisWorkbook.Worksheets
        cboSheets.AddItem wks.Name
    Next wks
End Sub

StandardModule: basMain

Sub DialogAufruf()
    frmSelect.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle nach ComboBox-Auswahl durchsuchen

Problem:	Die Werte aus den Spalten A:B von Tabelle1 sollen in einer ComboBox erscheinen. Die ausgewählte Maschine und deren Preis sollen in Tabelle2 angezeigt werden.
Lösung:	Nur anhand einer Beispiellarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summe aller Werte gem. ComboBox-Eintrag

Problem:	In Zelle D1 soll die Summe aller Werte aus Spalte B angezeigt werden, die dem Namen in der ComboBox in Spalte an entsprechen.
Lösung:	Nur anhand einer Beispieldarstellungsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus ComboBox in TextBox, dann in Tabelle suchen

Problem: In einer 3-spaltigen ComboBox soll ein Wert ausgewählt werden. Dieser soll in eine TextBox übertragen und in Tabelle2 in Spalte 3 gesucht und markiert werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmSuchen.Show
End Sub

ClassModule: frmSuchen

Private Sub cboSuchen_Change()
    txtSuchen.Text = cboSuchen.Value
End Sub

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdSuchen_Click()
    Dim rng As Range
    If txtSuchen.Text = "" Then
        Beep
        MsgBox "Sie müssen in der ComboBox einen Suchbegriff auswählen!"
        Exit Sub
    End If
    Set rng = Worksheets("Tabelle2").Columns(3) _
        .Find(txtSuchen.Text, lookat:=xlWhole, LookIn:=xlValues)
    If Not rng Is Nothing Then
        Application.Goto rng, Scroll:=True
    Else
        MsgBox "Suchbegriff wurde nicht gefunden!"
    End If
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    cboSuchen.List = Worksheets("Tabelle1") _
        .Range("A1").CurrentRegion.Value
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In Abhängigkeit einer ComboBox-Auswahl eine ListBox füllen

Problem: Wie kann ich in einer UserForm eine ListBox in Abhängigkeit einer ComboBox-Auswahl füllen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmListenFuellen.Show
End Sub

ClassModule: frmListenFuellen

Private Sub cboZeilen_Change()
    lstSpalten.List = _
        WorksheetFunction.Transpose( _
            Range(Cells(cboZeilen.ListIndex + 1, 2), _
                Cells(cboZeilen.ListIndex + 1, 6)).Value)
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    cboZeilen.List = Range("A1") _
        .CurrentRegion.Columns(1).Value
    cboZeilen.ListIndex = 0
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zweidimensionales Array an UserForm-ComboBox übergeben

Problem:	Ein zweidiimensionales Array soll über eine Schleife gefüllt und an eine UserForm-ComboBox übergeben werden.
Lösung:	Den nachfolgenden Code in die benannten Module einfügen.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() frmArray.Show End Sub ClassModule: frmArray Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim iCounter As Integer, iCount As Integer Dim arr() As String iCount = Day(DateSerial(Year(Date), Month(Date) + 1, 0)) ReDim arr(1 To iCount, 1 To 2) For iCounter = 1 To iCount arr(iCounter, 1) = Format(DateSerial(Year(Date), _ Month(Date), iCounter), "dd.mm.yy") arr(iCounter, 2) = Format(DateSerial(Year(Date), _ Month(Date), iCounter), "dddd") Next iCounter cboArray.List = arr cboArray.ListIndex = 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eintrag eines UserForm-ComboBox-Wertes in aktive Zelle

Problem:	Ein in einer UserForm-ComboBox ausgewählter Wert soll in die aktive Zelle eingetragen werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() frmCboToCell.Show End Sub ClassModule: frmCboToCell Private Sub cboTag_Change() ActiveCell.Value = cboTag.Value End Sub Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim intDay As Integer For intDay = 1 To 7 cboTag.AddItem Format(Day(Date) + intDay + 0, "dddd") Next intDay cboTag.ListIndex = 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-ComboBoxes als Collection-Objekte ansprechen

Problem:	Wie kann ich eine Reihe von UserForm-ComboBoxes in ein CollectionObjekt aufnehmen und die Elemente als solche ansprechen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmCollection Private Sub cmdBeenden_Click() Unload Me End Sub Private Sub cmdColA_Click() Dim iCounter As Integer, iRow As Integer For iCounter = 1 To 4 colA(iCounter).Clear For iRow = 1 To 12 colA(iCounter).AddItem Format(_ DateSerial(1, iRow, 1), "mmm") Next iRow colA(iCounter).ListIndex = 0 Next iCounter End Sub Private Sub cmdColALeeren_Click() Dim iCounter As Integer For iCounter = 1 To 4 colA(iCounter).Clear Next iCounter End Sub Private Sub cmdColBLEeren_Click() Dim iCounter As Integer For iCounter = 1 To 4 colB(iCounter).Clear Next iCounter End Sub Private Sub cmdColB_Click() Dim iCounter As Integer, iRow As Integer For iCounter = 1 To 4 colB(iCounter).Clear For iRow = 1 To 7 colB(iCounter).AddItem Format(_ DateSerial(1, 1, iRow), "ddd") Next iRow colB(iCounter).ListIndex = 0 Next iCounter End Sub Private Sub UserForm_Initialize() Dim iCounter As Integer On Error Resume Next For iCounter = 1 To 4 colA.Remove iCounter colB.Remove iCounter Next iCounter On Error GoTo 0 For iCounter = 1 To 8 Step 2 colA.Add Controls("ComboBox" & iCounter) colB.Add Controls("ComboBox" & iCounter + 1) Next iCounter End Sub StandardModule: basMain Public colA As New Collection Public colB As New Collection Sub CallForm() frmCollection.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei einer mehrspaltigen ComboBox Wert aus Spalte 2 als Startwert

Problem:	Bei einer mehrspaltigen UserForm-ComboBox wird standardmäßig der Wert aus der 1. Spalte des ListIndex angezeigt. Wie kann ich mir beim Aufruf den Wert aus der 2. Spalte anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmZweiteSpalte Private Sub cmdWeiter_Click() MsgBox cboWerte.Value Unload Me End Sub Private Sub cboWerte_Change() If cboWerte.ListIndex > -1 Then cboWerte.Value = cboWerte.List(cboWerte.ListIndex, 1) End If End Sub Private Sub UserForm_Initialize() cboWerte.List = Range("A1").CurrentRegion.Value cboWerte.Value = cboWerte.List(0, 1) End Sub StandardModule: basMain Sub CallForm() frmZweiteSpalte.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus UserForm-ComboBox suchen und Fundstelle kopieren

Problem: Wie kann ich die Werte aus Spalte A in eine UserForm-ComboBox übernehmen, den ausgewählten Wert in der Tabelle suchen und die Fundstelle in ein zweites Tabellenblatt kopieren?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSuchen

Private Sub cboSuchen_Change()
    txtSuchen.Text = cboSuchen.Value
End Sub

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdSuchen_Click()
    Dim rng As Range
    Dim iRow As Integer
    Application.ScreenUpdating = False
    Set rng = Columns(1).Find( _
        what:=txtSuchen.Text, _
        lookat:=xlPart, LookIn:=xlValues)
    If Not rng Is Nothing Then
        With Worksheets("Target")
            If IsEmpty(.Range("A1")) Then
                iRow = 1
            Else
                iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
            End If
            Rows(rng.Row).Copy .Rows(iRow)
        End With
    Else
        MsgBox "Suchbegriff wurde nicht gefunden!"
    End If
    Application.CutCopyMode = False
    Range("A1").Select
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    cboSuchen.List = _
        Range("A1").CurrentRegion.Columns(1).Value
    cboSuchen.ListIndex = 0
End Sub

StandardModule: Modul1

Sub CallForm()
    frmSuchen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Tabellenblatt-ComboBox den vorher selektierte Bereich wählen

Problem: Wie kann ich nach der Auswahl aus einer Tabellenblatt-ComboBox wieder den vorher selektierten Tabellenblattbereich auswählen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
StandardModule: basMain

Public grng As Range

ClassModule: Tabelle1

Private Sub ComboBox1_Change()
    grng.Select
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Set grng = Selection
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Artikel und Preise aus UserForm in Rechnung eintragen

Problem: Bei Doppelklick in einem Rechnungsformular soll eine UserForm mit einer mehrspaltigen ListBox mit Daten aus einem Datenblatt gezeigt werden. Nach Auswahl eines Datensatzes soll dieser im Formular eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    Cancel = True
    frmArtikel.Show
End Sub

ClassModule: frmArtikel

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim iRowL As Integer, iRow As Integer, iCounter As Integer
    iRowL = Cells(21, 1).End(xlUp).Row + 1
    iRow = 1
    For iCounter = 0 To lstArtikel.ListCount - 1
        If lstArtikel.Selected(iCounter) Then
            iRow = iRow + 1
            Cells(iRow, 1) = Worksheets("Daten").Cells(iCounter + 2, 1)
            Cells(iRow, 2) = Worksheets("Daten").Cells(iCounter + 2, 2)
            Cells(iRow, 3) = Worksheets("Daten").Cells(iCounter + 2, 3)
            Cells(iRow, 5).Formula = "=RC[-2]*RC[-1]"
        End If
    Next iCounter
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim rng As Range
    Set rng = Worksheets("Daten").Range("A1").CurrentRegion
    lstArtikel.List = rng.Offset(1, 0).Resize(rng.Rows.Count - 1, _
        rng.Columns.Count).Value
End Sub
```




Ein- und ausblenden von Blättern gem. UserForm-ComboBoxes

Problem:	Wie kann ich das Ein- und Ausblenden von Monats-Arbeitsblättern über UserForm-ComboBoxes steuern? Ein Datenblatt soll nur nach Passworтеingabe geöffnet werden können.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Dim oExtras As CommandBarPopup Set oExtras = Application.CommandBars(_ "Worksheet Menu Bar").FindControl(ID:=30007) On Error Resume Next oExtras.Controls("Blattauswahl").Delete On Error GoTo 0 End Sub Private Sub Workbook_Open() Dim oExtras As CommandBarPopup Dim oBtn As CommandBarButton Set oExtras = Application.CommandBars(_ "Worksheet Menu Bar").FindControl(ID:=30007) On Error Resume Next oExtras.Controls("Blattauswahl").Delete On Error GoTo 0 Set oBtn = oExtras.Controls.Add With oBtn .Caption = "Blattauswahl" .OnAction = "CallForm" .BeginGroup = True .Style = msoButtonCaption End With End Sub ClassModule: frmPassword Private Sub cmdOK_Click() If txtPassword.Text = "Password" Then Worksheets("Daten").Visible = True frmAusEinblenden.cmdDaten.Caption = "Daten ausblenden" Else Beep MsgBox "Falsches Passwort!" End If Unload Me End Sub ClassModule: frmAusEinblenden Private Sub cboAusblenden_Change() Dim wksAct As Worksheet Dim intCounter As Integer For Each wksAct In Worksheets If wksAct.Visible = xlSheetVisible Then intCounter = intCounter + 1 If intCounter = 2 Then Exit For End If Next wksAct If intCounter = 1 And cboAusblenden.Value <> "" Then MsgBox "Ein Blatt muss sichtbar bleiben!" Else If cboAusblenden.Value <> "" Then Worksheets(cboAusblenden.Value).Visible = xlHidden Call UserForm_Initialize End If End If End Sub Private Sub cboEinblenden_Change() If cboEinblenden.Value <> "" Then Worksheets(cboEinblenden.Value).Visible = xlSheetVisible Call UserForm_Initialize End If End Sub Private Sub cmdDaten_Click() If cmdDaten.Caption = "Daten einblenden" Then frmPassword.Show Else Worksheets("Daten").Visible = xlVeryHidden cmdDaten.Caption = "Daten einblenden" End If End Sub Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim wksAct As Worksheet cboEinblenden.Clear cboAusblenden.Clear For Each wksAct In ThisWorkbook.Worksheets Select Case wksAct.Visible Case xlSheetVisible If wksAct.Name = "Daten" Then cmdDaten.Caption = "Daten ausblenden" Else cboAusblenden.AddItem wksAct.Name End If Case xlSheetHidden cboEinblenden.AddItem wksAct.Name Case xlSheetVeryHidden cmdDaten.Caption = "Daten einblenden" End Select Next wksAct End Sub StandardModule: basMain Sub CallForm() frmAusEinblenden.Show End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in UserForm-ComboBox einlesen und ausgeben

Problem: Wie kann ich die Werte einer Spalte in eine UserForm-ComboBox einlesen und unmittelbar nach Auswahl in eine Zelle ausgeben?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmComboDemo

Private Sub UserForm_Initialize()
    Dim rng As Range
    For Each rng In Range("A1:A5").Cells
        cboListe.List = Range("A1").CurrentRegion.Columns(1).Value
    Next rng
    cboListe.ListIndex = 0
End Sub

Private Sub cboListe_Change()
    Cells(1, 2) = cboListe.Value
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmComboDemo.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

TextBox in Abhängigkeit einer ComboBox-Auswahl füllen

Problem: Wie kann ich in einer UserForm eine TextBox aus einem Arbeitsblatt in Abhängigkeit einer in einer ComboBox getroffenen Auswahl füllen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmInTextBox

Private Sub cboListe_Change()
    txtAuswahl.Text = Cells(cboListe.ListIndex + 1, 2)
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim intCounter As Integer
    For intCounter = 1 To 12
        cboListe.AddItem Cells(intCounter, 1)
    Next intCounter
    cboListe.ListIndex = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmInTextBox.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adressen aus Adresstabelle übernehmen, bearbeiten, eintragen

Problem: Wie kann ich Daten aus einer Adresstabelle in eine UserFormComboBox übernehmen, dort auswählen und in anderer Form in einer zweiten Tabelle ausgeben lassen? Im Dialog vorgenommene Änderungen sollen berücksichtigt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSuchen

Private Sub ComboBox1_Change()
    Dim wks As Worksheet
    Dim iRow As Integer
    Set wks = Worksheets("Adressen")
    iRow = ComboBox1.ListIndex + 2
    TextBox1.Text = wks.Cells(iRow, 1)
    TextBox2.Text = wks.Cells(iRow, 2)
    TextBox3.Text = Trim(wks.Cells(iRow, 3) & " " & wks.Cells(iRow, 4))
    TextBox4.Text = wks.Cells(iRow, 7)
    TextBox5.Text = Trim(wks.Cells(iRow, 5) & " " & wks.Cells(iRow, 6))
    TextBox6.Text = wks.Cells(iRow, 8)
    TextBox7.Text = wks.Cells(iRow, 9)
    TextBox8.Text = wks.Cells(iRow, 10)
End Sub

Private Sub cmdUebernehmen_Click()
    Dim iCol As Integer, iRowL As Integer
    If IsEmpty(Cells(1, 1)) Then
        iRowL = 1
    Else
        iRowL = Cells(Rows.Count, 1).End(xlUp).Row + 1
    End If
    For iCol = 1 To 8
        Cells(iRowL, iCol) = frmSuchen.Controls("TextBox" & iCol).Value
    Next iCol
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim sAdr As String
    sAdr = ThisWorkbook.Worksheets("Adressen"). _
        Cells(Rows.Count, 1).End(xlUp).Address(False, False)
    ComboBox1.RowSource = "Adressen!A2:" & sAdr
End Sub

StandardModule: basMain

Sub CallForm()
    frmSuchen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adresstabelle in UserForm einlesen und in Adressfeld ausgeben

Problem:	Wie kann ich eine Tabelle mit Adressen in eine UserForm-ComboBox einlesen und von dort in Adressfelder ausgeben lassen? Neue Adressen sollen aufgenommen werden können.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmSuchen Private Sub cmdNeu_Click() Dim iRow As Integer, iCol As Integer With Worksheets("Daten") iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1 For iCol = 1 To 5 .Cells(iRow, iCol) = Controls("TextBox" & iCol).Text Next iCol End With End Sub Private Sub ComboBox1_Change() Dim iCounter As Integer For iCounter = 1 To 5 Controls("TextBox" & iCounter).Text = _ Worksheets("Daten").Cells(ComboBox1.ListIndex + 2, iCounter) Next iCounter End Sub Private Sub CommandButton1_Click() Range("A12") = TextBox2.Text & " " & TextBox1.Text Range("A13") = TextBox3.Text Range("A15") = TextBox4.Text & " " & TextBox5.Text Unload Me End Sub Private Sub CommandButton2_Click() Unload Me End Sub Private Sub UserForm_Initialize() With Worksheets("Daten") ComboBox1.List = _ .Range(.Cells(2, 1), .Cells(_ .UsedRange.Rows.Count, 1)).Value End With End Sub StandardModule: basMain Sub CallForm() frmSuchen.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus ComboBox in Tabelle eintragen

Problem:	Wie kann ich einen in einer Tabellenblatt-ComboBox ausgewählten Wert in der aktiven Tabelle eintragen lassen?
Lösung:	Legen Sie die LinkedCell-Eigenschaft der ComboBox fest.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBoxes per VBA-Code in Tabellenblatt erzeugen

Problem: Wie kann ich über VBA-Code in einem Tabellenblatt ComboBoxes je Tabellenblattspalte erstellen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub DropDownKopieren()  
    Dim oDropDown As DropDown  
    Dim iRow As Integer  
    For iRow = 1 To 10  
        Set oDropDown = ActiveSheet.DropDowns.Add( _  
            Cells(iRow, 1).Left, _  
            Cells(iRow, 1).Top, _  
            Cells(iRow, 1).Width + Cells(iRow, 2).Width, _  
            Cells(iRow, 1).Height)  
        With oDropDown  
            .ListFillRange = Range( _  
                Cells(1, iRow + 3), Cells(10, iRow + 3)).Address  
            .LinkedCell = Cells(iRow, 3).Address  
            .ListIndex = 1  
        End With  
    Next iRow  
    Range("A1").Select  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm mit ComboBoxes durch VBA-Code erstellen

Problem: Wie kann ich über VBA-Code eine UserForm mit je einer ComboBox je Tabellenblattspalte erstellen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub NeueUserForm()
    Dim frmNew
    Dim cboBox As MSForms.ComboBox
    Dim cmdWeiter As MSForms.CommandButton
    Dim intCounter As Integer, intTop As Integer, intRow As Integer
    Dim strCode As String
    Application.VBE.MainWindow.Visible = False
    On Error Resume Next
        Set frmNew = ThisWorkbook.VBProject.VBComponents("frmTageslisten")
    If Err = 0 Then GoTo ErrorHandler
    On Error GoTo 0
    Set frmNew = ThisWorkbook.VBProject.VBComponents.Add(3)
    intTop = 5
    For intCounter = 1 To 31
        If intCounter = 17 Then intTop = 5
        Set cboBox = frmNew.designer.Controls.Add("forms.ComboBox.1")
        With cboBox
            If intCounter < 17 Then .Left = 5 Else .Left = 160
            .Top = intTop
            .Width = 150
            .RowSource = Range( _
                Cells(2, intCounter), Cells(10, intCounter)).Address
            .ControlSource = Cells(intCounter, 32)
            .ListIndex = 0
        End With
        intTop = intTop + 20
    Next intCounter
    Set cmdWeiter = frmNew.designer.Controls.Add("forms.CommandButton.1")
    With cmdWeiter
        .Top = intTop + 30
        .Left = 5
        .Width = 305
        .Height = 30
        .Caption = "Weiter"
        .Name = "cmdWeiter"
    End With
    With frmNew
        .properties("Width") = 320
        .properties("Height") = 17 * 20 + 50
        .properties("Caption") = "Tageslisten"
        .properties("Name") = "frmTageslisten"
    End With
    strCode = "Private Sub cmdWeiter_Click" & vbLf
    strCode = strCode & "    Unload Me" & vbLf
    strCode = strCode & "End Sub"
    ThisWorkbook.VBProject.VBComponents("frmTageslisten") _
        .CodeModule.AddFromString strCode
ErrorHandler:
    VBA.UserForms.Add(frmNew.Name).Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tages-ComboBox einer UserForm an den ausgewählten Monat anpassen

Problem: Wie kann ich eine UserForm-ComboBox mit Tagesangaben an die Auswahl eines Monats in einer Monats-ComboBox anpassen. Die ausgewählten Daten sollen dann in ein Tabellenblatt eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmComboBoxes

Private Sub cboMonate_Change()
    Dim intCounter As Integer
    cboTage.Clear
    For intCounter = 1 To Day( _
        DateSerial(Year(Date), cboMonate.ListIndex + 2, 0))
        cboTage.AddItem intCounter
    Next intCounter
    cboTage.ListIndex = cboTage.ListCount - 1
End Sub

Private Sub cmdEintragen_Click()
    Range("B1").Value = cboMonate.Value
    Range("B2").Value = cboTage.Value
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim intCounter As Integer
    For intCounter = 1 To 12
        cboMonate.AddItem Format(DateSerial(1, intCounter, 1), "mmm")
    Next intCounter
    cboMonate.ListIndex = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmComboBoxes.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adressfeld aus UserForm-ComboBox füllen

Problem: Wie kann ich ein Adressfeld in einem Tabellenblatt durch Auswahl des Firmennamens in einer UserForm-ComboBox füllen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAdressen

Private Sub cbbAdressen_Change()
    Dim wks As Worksheet, shBrief As Worksheet
    Set wks = Worksheets("Adressen")
    With cbbAdressen
        Range("A12") = wks.Cells(.ListIndex + 2, 1)
        Range("A13") = wks.Cells(.ListIndex + 2, 2)
        Range("A15") = wks.Cells(.ListIndex + 2, 3) & _
            " " & wks.Cells(.ListIndex + 2, 4)
    End With
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim lRow As Long
    With Worksheets("Adressen")
        lRow = .Cells(Rows.Count, 1).End(xlUp).Row
        cbbAdressen.List = .Range(.Cells(2, 1), .Cells(lRow, 1)).Value
    End With
    cbbAdressen.ListIndex = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmAdressen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox in Abhängigkeit von einer CheckBox anzeigen

Problem:	Wie kann ich eine ComboBox nur bei aktivierter CheckBox anzeigen, sonst ausblenden lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle12 Private Sub CheckBox1_Click() ComboBox1.Visible = CheckBox1.Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensätze in UserForm auswählen und im Blatt löschen

Problem: Wie kann ich in einer UserForm auszuwählende Datensätze im Tabellenblatt löschen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmLoeschen

Private Sub cmdLoeschen_Click()
    Rows(cboLoeschen.ListIndex + 1).Delete
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmLoeschen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Offene Arbeitsmappen in Dialog listen

Problem: Wie kann ich alle offenen Arbeitsmappen in einem DropDown-Feld eines Dialogs listen und über eine MessageBox den Wert aus Zelle A1 aus Arbeitsblatt "Tabelle1" der ausgewählten Arbeitsmappe anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmWkb.Show
End Sub

ClassModule: frmWkb

Private Sub cbowkb_Change()
    Workbooks(cbowkb.Value).Activate
    Unload Me
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim wkb As Workbook
    For Each wkb In Workbooks
        cbowkb.AddItem wkb.Name
    Next wkb
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In einer ComboBox selektiertes Arbeitsblatt auswählen

Problem:	Wie kann ich über ein sich in einem Tabellenblatt befindenden DropDown-Feld ein Arbeitsblatt auswählen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub cmdNewSheets_Click() Dim iSheet As Integer ComboBox1.Clear For iSheet = 1 To 12 Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Name = Format(DateSerial(1, iSheet, 1), "mmm") ComboBox1.AddItem ActiveSheet.Name Next iSheet ComboBox1.ListIndex = 0 End Sub Private Sub ComboBox1_Change() Worksheets(ComboBox1.Value).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus 2. UserForm-ComboBox-Spalte berechnen

Problem: Wie kann ich bei Drücken einer Schaltfläche den Wert aus der 2. Spalte einer UserForm-ComboBox berechnen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSecondValue

Private Sub cmdCalculate_Click()
    Dim iCounter As Integer
    iCounter = 45
    MsgBox "Ausgewählter Wert * " & _
        iCounter & ": " & cboValues.Value * iCounter
End Sub

Private Sub UserForm_Initialize()
    cboValues.List = Range("A1:B12").Value
    cboValues.ListIndex = 0
End Sub

Private Sub cmdContinue_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmSecondValue.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-ComboBox-Feld nach Tasteneingabe aufklappen

Problem: Wie kann ich erreichen, daß ein UserForm-DropDown-Feld nach einer Tasteneingabe automatisch aufklappt?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDropDown

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub cboA_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    cboA.DropDown
End Sub

Private Sub cboB_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    cboB.DropDown
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 12
        cboA.AddItem Format(DateSerial(1, iCounter, 1), "mmm")
    Next iCounter
    cboA.ListIndex = 0
    For iCounter = 1 To 7
        cboB.AddItem Format(DateSerial(1, 1, iCounter), "ddd")
    Next iCounter
    cboB.ListIndex = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmDropDown.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox-Felder in UserForm aufklappen

Problem: Wie kann ich erreichen, daß DropDown-Felder in UserForms - wenn sie den Focus haben - nach Betätigung der Enter-Taste aufklappen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDropDown

Private Sub cboA_Enter()
    cboA.DropDown
End Sub

Private Sub cboB_Enter()
    cboB.DropDown
End Sub

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 12
        cboA.AddItem Format(DateSerial(1, iCounter, 1), "mmm")
    Next iCounter
    cboA.ListIndex = 0
    For iCounter = 1 To 7
        cboB.AddItem Format(DateSerial(1, 1, iCounter), "ddd")
    Next iCounter
    cboB.ListIndex = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmDropDown.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe aus Dialog schließen

Problem: Wie kann ich aus einem Dialog heraus die aktive oder auch eine andere Arbeitsmappe schließen?

Lösung: Den nachfolgenden Code in die genannten Module eingeben.

VBA-Code:

```
StandardModule: basMain

Sub DialogAufruf()
    frmWkbClose.Show
End Sub

ClassModule: frmWkbClose

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdClose_Click()
    Workbooks(cboWkb.Value).Close
End Sub

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_Initialize()
    Dim wkb As Workbook
    For Each wkb In Workbooks
        cboWkb.AddItem wkb.Name
    Next wkb
    cboWkb.ListIndex = 0
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blätter in ComboBox-Feld einlesen und auswählen

Problem:	Wie kann ich Arbeitsblätter einer bestimmten Serie in ein DropDownFeld eines Dialogblattes einlesen und das ausgewählte selektieren?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() frmWks.Show End Sub ClassModule: frmWks Private Sub cboWks_Change() Worksheets(cboWks.Value).Select Unload Me End Sub Private Sub cmdContinue_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim wks As Worksheet For Each wks In Worksheets cboWks.AddItem wks.Name Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bestellfax über Dialog

Problem: Wie kann ich Bestellartikel über einen Dialog auswählen und in eine Faxbestellung eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmArtikel

Private Sub cboArtikel_Change()
    With Worksheets("Artikelliste")
        TextBox1.Text = _
            .Cells(cboArtikel.ListIndex + 2, 1).Value
        TextBox2.Text = _
            .Cells(cboArtikel.ListIndex + 2, 3).Value
    End With
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim iRow As Integer
    iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
    Cells(iRow, 1).Value = TextBox1.Text
    Cells(iRow, 2).Value = cboArtikel.Value
    Cells(iRow, 3).Value = TextBox3.Text
    Cells(iRow, 4).Value = TextBox2.Text
    Cells(iRow, 5).Value = _
        Cells(iRow, 3).Value * Cells(iRow, 4).Value
End Sub

Private Sub UserForm_Initialize()
    Dim iRow As Integer
    iRow = 2
    With Worksheets("Artikelliste")
        Do Until IsEmpty(.Cells(iRow, 1))
            cboArtikel.AddItem .Cells(iRow, 2).Value
            iRow = iRow + 1
        Loop
    End With
    cboArtikel.ListIndex = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmArtikel.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adressdaten via Dialog in eine Adressdatei eintragen

Problem:	Wie kann ich Adressdaten in einen Dialog eingeben und diese dann in eine Adressdatei eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_BeforeDoubleClick(_ ByVal Target As Range, Cancel As Boolean) If Target.Cells.Count > 1 Then Exit Sub If Intersect(_ Target, Range("A1").CurrentRegion _) Is Nothing Then Exit Sub Cancel = True frmAddress.Show End Sub ClassModule: frmAddress Private Sub cboName_Change() Dim iCol As Integer For iCol = 1 To 7 Controls("TextBox" & iCol).Text = _ Cells(ActiveCell.Row, iCol + 1).Value Next iCol End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim iRow As Integer, iCol As Integer If cboName.ListIndex = -1 Then iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 Else iRow = ActiveCell.Row End If Cells(iRow, 1).Value = cboName.Value For iCol = 1 To 7 Cells(iRow, iCol + 1).Value = _ Controls("TextBox" & iCol).Text Next iCol End Sub Private Sub UserForm_Click() End Sub Private Sub UserForm_Initialize() Dim iRow As Integer iRow = 2 With cboName Do Until IsEmpty(Cells(iRow, 1)) .AddItem Cells(iRow, 1).Value iRow = iRow + 1 Loop .ListIndex = ActiveCell.Row - 2 End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werteübernahme gem. Auswahl in ComboBox - ohne VBA

Problem:	Wie kann ich von der Auswahl in einem DropDown-Feld abhängige Werte in eine Tabelle eintragen?
Lösung:	Nur anhand einer Beispiellarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Bestandsführung für 2 Läger

Problem:	Wie kann ich per Dialog die Bestände von 2 Lägern führen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmLager Private Sub cboArtikel_Change() Dim wks As Worksheet Dim iRow As Integer If optLagerA.Value = True Then Set wks = Worksheets("Lager1") Else Set wks = Worksheets("Lager2") End If If cboArtikel.ListIndex = -1 Then txtStueck.Text = "" Else iRow = WorksheetFunction.Match(_ cboArtikel.Value, wks.Columns(1), 0) txtStueck.Text = wks.Cells(iRow, 2).Value End If End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim wks As Worksheet Dim iRow As Integer If optLagerA.Value = True Then Set wks = Worksheets("Lager1") Else Set wks = Worksheets("Lager2") End If If cboArtikel.ListIndex = -1 Then iRow = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1 Else iRow = WorksheetFunction.Match(_ cboArtikel.Value, wks.Columns(1), 0) End If wks.Cells(iRow, 1).Value = cboArtikel.Value If optEingang.Value = True Then wks.Cells(iRow, 2).Value = _ wks.Cells(iRow, 2).Value + CDbl(txtStueck.Text) Else wks.Cells(iRow, 2).Value = _ wks.Cells(iRow, 2).Value - CDbl(txtStueck.Text) End If txtStueck.Text = wks.Cells(iRow, 2).Value End Sub Private Sub optLagerA_Change() Call optLager_Change End Sub Private Sub optLagerB_Change() Call optLager_Change End Sub Private Sub optLager_Change() Dim wks As Worksheet Dim iRow As Integer cboArtikel.Clear If optLagerA.Value = True Then Set wks = Worksheets("Lager1") Else Set wks = Worksheets("Lager2") End If iRow = 2 With cboArtikel Do Until IsEmpty(wks.Cells(iRow, 1)) .AddItem wks.Cells(iRow, 1).Value iRow = iRow + 1 Loop If .ListCount > 0 Then .ListIndex = 0 End If End With End Sub StandardModule: basMain Sub CallForm() If ActiveSheet.Name = "Lager1" Then frmLager.optLagerA.Value = True Else frmLager.optLagerB.Value = True End If frmLager.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neues Element in ComboBox

Problem: Wie kann ich die Elementliste einer ComboBox um ein Element erweitern?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmStoffdaten

Private Sub cboStoffdaten_Change()
    Dim iRow As Integer
    If cboStoffdaten.ListIndex = 0 Then
        frmStoff.Show
        iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
        Cells(iRow, 1) = frmStoff.txtStoff.Text
        cboStoffdaten.AddItem frmStoff.txtStoff.Text
    Else
        If IsEmpty(Cells(1, 2)) Then
            iRow = 1
        Else
            iRow = Cells(Rows.Count, 2).End(xlUp).Row + 1
        End If
        Cells(iRow, 2) = cboStoffdaten.Value
    End If
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iRow As Integer
    iRow = 1
    With cboStoffdaten
        .Clear
        .AddItem "Neu"
        Do While IsEmpty(Cells(iRow, 1)) = False
            .AddItem Cells(iRow, 1)
            iRow = iRow + 1
        Loop
    End With
End Sub

ClassModule: frmStoff

Private Sub cmdInsert_Click()
    Dim iRow As Integer
    iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
    Cells(iRow, 1).Value = txtStoff.Text
    frmStoffdaten.cboStoffdaten.AddItem txtStoff.Text
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmStoffdaten.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Ortsauswahl neue UserForm mit Ortsteilauswahl

Problem: Wie kann ich mir nach der Auswahl eines Ortes in einer ComboBox einer UserForm in einer neuen UserForm die zugehörigen Ortsteile zur Auswahl anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAddress

Private Sub cboOrt_Change()
    Dim iCol As Integer
    iCol = 2
    cboOrtsteil.Clear
    With Worksheets("Daten")
        Do Until IsEmpty(.Cells(cboOrt.ListIndex + 1, iCol))
            cboOrtsteil.AddItem .Cells(cboOrt.ListIndex + 1, iCol)
            iCol = iCol + 1
        Wend
    End With
    cboOrtsteil.ListIndex = 0
End Sub

Private Sub cmdContinue_Click()
    Dim iRow As Integer
    If cboOrtsteil.ListIndex <> -1 Then
        iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
        Cells(iRow, 1) = cboOrt.Value
        Cells(iRow, 2) = cboOrtsteil.Value
    End If
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    cboOrt.ListIndex = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmAddress.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neue UserForm bedingt nach ComboBox-Auswahl aufrufen

Problem:	Wie kann ich bedingt durch die Auswahl in einer Mehrspalten-ListBox eine zweite UserForm aufrufen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmListe Private Sub cmdContinue_Click() Unload Me End Sub Private Sub lstPersonal_Click() Dim iCounter As Integer For iCounter = 1 To 7 frmPersonal.Controls("TextBox" & iCounter).Text = _ lstpersonal.List(lstpersonal.ListIndex, iCounter - 1) Next iCounter End Sub Private Sub UserForm_Initialize() Dim rng As Range Set rng = Range("A1").CurrentRegion lstpersonal.List = rng.Rows(2, rng.Rows.Count).Value End With End Sub ClassModule: frmPersonal Private Sub cmdContinue_Click() Unload Me End Sub Private Sub cmdInsert_Click() Dim iCounter As Integer gbln = True For iCounter = 1 To 7 With frmListe.lstPersonal Worksheets("Dummy").Cells(1, iCounter).Value = _ .List(.ListIndex, iCounter - 1) .List(.ListIndex, iCounter - 1) = _ Controls("TextBox" & iCounter).Text Cells(.ListIndex + 2, iCounter).Value = _ .List(.ListIndex, iCounter - 1) End With Next iCounter cmdReset.Enabled = True End Sub Private Sub cmdReset_Click() Dim iCounter As Integer For iCounter = 1 To 7 With frmListe.lstPersonal .List(.ListIndex, iCounter - 1) = _ Worksheets("Dummy").Cells(1, iCounter).Value Cells(.ListIndex + 2, iCounter).Value = _ Worksheets("Dummy").Cells(1, iCounter).Value End With Next iCounter cmdReset.Enabled = False gbln = False End Sub StandardModule: basMain Sub CallForm() frmListe.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur Zellen mit Inhalt in ComboBox listen

Problem: Wie kann ich aus einer Tabellenspalte nur die Zellen in eine UserForm übernehmen, die nicht leer sind?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle12

Private Sub CommandButton1_Click()
    UserForm1.Show
End Sub

ClassModule: frmNoEmptys

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iRow As Integer, iRowL As Integer
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row
    For iRow = 1 To iRowL
        If Not IsEmpty(Cells(iRow, 1)) Then
            cbonoemptys.AddItem Cells(iRow, 1).Value
        End If
    Next iRow
End Sub

StandardModule: basMain

Sub CallForm()
    frmNoEmptys.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox2 in Abhängigkeit von ComboBox1 bestücken

Problem: Wie kann ich eine zweite ComboBox in einer UserForm in Abhängigkeit der Auswahl in der ersten ComboBox bestücken lassen? Wenn in der 1. Box A gewählt wird, sollen in der 2. C und D zur Auswahl stehen, bei B in der 1. D und E in der zweiten usw.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmBedingt

Private Sub cboa_Change()
    With cbob
        .Clear
        If cboa.ListIndex < 25 Then
            .AddItem Chr(cboa.ListIndex + 67)
            .AddItem Chr(cboa.ListIndex + 68)
            .ListIndex = 0
        End If
    End With
End Sub

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 26
        cboa.AddItem Chr(iCounter + 64)
    Next iCounter
    cboa.ListIndex = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmBedingt.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrspaltige ComboBox

Problem: Wie kann ich eine mehrspaltige ComboBox mit Werten füllen und nach dem Aufruf die ausgewählte Zeile auslesen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code: StandardModule: basMain

```
Sub GebundeneComboBox()  
    With frmMehrspaltig  
        With .cboColumns  
            .RowSource = _  
                ActiveSheet.Name & "!" & Range("A1") _  
                .CurrentRegion.Address  
            .ListIndex = 0  
        End With  
        .Show  
    End With  
End Sub  
  
Sub FreieComboBox()  
    With frmMehrspaltig  
        With .cboColumns  
            .List = Range("A1").CurrentRegion.Value  
            .ListIndex = 0  
        End With  
        .Show  
    End With  
End Sub  
  
ClassModule: frmMehrspaltig  
  
Private Sub cmdContinue_Click()  
    Unload Me  
End Sub  
  
Private Sub cmdValues_Click()  
    Dim iCounter As Integer  
    Dim sMsg As String  
    For iCounter = 0 To 2  
        With cboColumns  
            sMsg = sMsg & "Spalte " & iCounter + 1 & ": " & _  
                .List(.ListIndex, iCounter) & vbCrLf  
        End With  
    Next iCounter  
    MsgBox sMsg  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach ausgewählten Daten filtern

Problem:	Wie kann ich über ComboBoxes Begriffe auswählen, nach denen eine Tabelle gefiltert werden soll? Die gefilterten Datensätze sollen in ein neues Tabellenblatt übernommen werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmFilter Private Sub cmdFilter_Click() Dim iCounter As Integer For iCounter = 1 To 5 If Controls("ComboBox" & iCounter).ListIndex <> -1 Then Range("A1").AutoFilter _ Field:=iCounter, _ Criteria1:=Controls("ComboBox" & iCounter).Value End If Next iCounter Range("A1").CurrentRegion.Copy Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Paste Worksheets("Tabelle1").AutoFilterMode = False Unload Me End Sub StandardModule: basMain Sub CallForm() frmFilter.Show Application.CutCopyMode = False ActiveSheet.Range("A1").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox mit Monatsnamen füllen

Problem:	Wie kann ich eine ComboBox in einer UserForm mit den Monatsnamen eines Jahres füllen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmMonate Private Sub cmdcontinue_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim iCounter As Integer cbomonths.Clear For iCounter = 1 To 12 cbomonths.AddItem Format(DateSerial(1, iCounter, 1), "mmmm") Next iCounter cbomonths.ListIndex = 0 End Sub StandardModule: basMain Sub CallForm() frmMonate.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox in Abhängigkeit einer 2. ComboBox ein- / ausblenden

Problem:	Wie kann ich eine in einem Tabellenblatt verankerte ComboBox in Abhängigkeit einer zweiten, sich ebenfalls in diesem Tabellenblatt befindlichen ComboBox ein- und ausblenden?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub ComboBox1_Change() If ComboBox1.Value = "Zeile 5" Then ComboBox2.Visible = False Else ComboBox2.Visible = True End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in eine ComboBox einer UserForm übernehmen

Problem: Wie kann ich eine Liste von Werten ohne Berücksichtigung der Leerzellen und ohne doppelte Werte in eine UserForm-ComboBox einlesen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmColumnA

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim rng As Range
    Set rng = Columns(1)
    Application.ScreenUpdating = False
    Workbooks.Add 1
    rng.Copy Columns(1)
    Columns(1).AdvancedFilter _
        Action:=xlFilterCopy, _
        CopyToRange:=Range("B1"), _
        Unique:=True
    Columns(1).Delete
    Columns(1).Sort _
        Key1:=Range("A2"), Order1:=xlAscending, _
        Header:=xlYes, OrderCustom:=1, _
        MatchCase:=False, Orientation:=xlTopToBottom
    Rows(1).Delete
    cboColumnA.List = Range("A1").CurrentRegion.Value
    ActiveWorkbook.Close savechanges:=False
    Range("A1").Select
    Application.ScreenUpdating = True
End Sub

StandardModule: Modul1

Sub CallForm()
    frmColumnA.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle ComboBoxes auf Position 1

Problem: Wie kann ich bei EXCEL 5.0/7.0 in einer Prozedur festlegen, daß bei Aufruf eines Dialogs alle Dropdowns das erste Element der Liste anzeigen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code: StandardModule: basMain

```
Sub CallForm()  
    frmDate.Show  
End Sub
```

ClassModule: frmDate

```
Private Sub cboMonth_Change()  
    Call SetDays  
End Sub
```

```
Private Sub cboYear_Change()  
    Call SetDays  
End Sub
```

```
Private Sub cmdContinue_Click()  
    Unload Me  
End Sub
```

```
Private Sub UserForm_Initialize()  
    Dim lDay As Long  
    Dim iYear As Integer, iMonth As Integer  
    For iYear = Year(Date) - 3 To Year(Date) + 2  
        cboYear.AddItem iYear  
    Next iYear  
    For iMonth = 1 To 12  
        cboMonth.AddItem Format(DateSerial(1, iMonth, 1), "mmm")  
    Next iMonth  
    cboYear.ListIndex = 0  
    cboMonth.ListIndex = 0  
    Call SetDays  
End Sub
```

```
Private Sub SetDays()  
    Dim lDay As Long  
    cboDay.Clear  
    For lDay = DateSerial(cboYear.Value, cboMonth.ListIndex + 1, 1) To _  
        DateSerial(cboYear.Value, cboMonth.ListIndex + 2, 0)  
        cboDay.AddItem Day(lDay)  
    Next lDay  
    cboDay.ListIndex = 0  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox als Symbol

Problem:	Wie kann ich eine Tabellenblatt-ListBox auf Icon-Größe verkleinern?
Lösung:	Geben Sie den nachfolgenden Code in die genannten Module ein und weisen Sie dem im Standardmodul einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub lstBox_DblClick(ByVal Cancel As MSForms.ReturnBoolean) lstBox.Visible = False Shapes("objList").Visible = True End Sub StandardModule: Modul1 Sub objList_Click() With Worksheets("Tabelle1") .lstBox.Visible = True .Shapes("objList").Visible = False End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schaltflächen auf Schaltflächenklick tauschen

Problem:	Über die untere Schaltfläche sollen die beiden oberen Schaltflächen getauscht werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub MessageA() MsgBox "Aufruf von Button A" End Sub Sub messageB() MsgBox "Aufruf von Button B" End Sub Sub ChangeButton() With ActiveSheet .Buttons(1).Visible = Not .Buttons(1).Visible .Buttons(2).Visible = Not .Buttons(2).Visible End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

CommandButtons sollen auf Schaltflächen-Klick deaktiviert werden

Problem:	2 von 4 CommandButtons sollen auf Schaltflächen-Klick deaktiviert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CmdsEnabled() Dim iCounter As Integer For iCounter = 2 To 3 ActiveSheet.OLEObjects("CommandButton" & iCounter) _ .Object.Enabled = False Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Button im selektierten Bereich erstellen und Makro zuweisen

Problem:	Im aktiven Blatt soll eine Schaltfläche an der Position und in der Größe des selektierten Bereiches erstellt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetButton() Dim btn As Button Dim dWidth As Double, dHeight As Double With Selection dWidth = .Cells(.Cells.Count).Left - _ .Cells(1).Left + .Cells(.Cells.Count).Width dHeight = .Cells(.Cells.Count).Top - _ .Cells(1).Top + .Cells(.Cells.Count).RowHeight Set btn = ActiveSheet.Buttons.Add(.Cells(1).Left, _ .Cells(1).Top, dWidth, dHeight) End With btn.Caption = "Aufruf" btn.OnAction = "Message" End Sub Sub Message() MsgBox "Hallo" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In der UserForm aufrufende Schaltfläche erkennen

Problem:	In die Zeile mit der die UserForm aufrufenden Schaltfläche sollen Werte eingetragen werden.
Lösung:	Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmCallerRow Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim iCounter As Integer For iCounter = 2 To 5 Cells(Me.Tag, iCounter).Value = iCounter - 1 Next iCounter End Sub StandardModule: Modul1 Sub DialogAufruf() frmCallerRow.Tag = ActiveSheet.Buttons(Application.Caller).TopLeftCell.Row frmCallerRow.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

CommandButtons in verschiedenen Tabellen mit einem Makro

Problem:	Wie kann ich eine Serie von CommandButtons aus der Steuerelement-Toolbar durch nur ein einziges Makro steuern?
Lösung:	Den nachfolgenden Code in die benannten Module einfügen.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Call UnionCommands End Sub StandardModule: Modul1 Dim Buttons() As New Klasse1 Sub UnionCommands() Dim wks As Worksheet Dim ButtonCount As Integer Dim ctl As Object ButtonCount = 0 For Each wks In Worksheets For Each ctl In wks.OLEObjects If TypeName(ctl.Object) = "CommandButton" Then ButtonCount = ButtonCount + 1 ReDim Preserve Buttons(1 To ButtonCount) Set Buttons(ButtonCount).ButtonGroup = ctl.Object End If Next ctl Next wks End Sub ClassModule: Klasse1 Public WithEvents ButtonGroup As CommandButton Private Sub ButtonGroup_Click() MsgBox "Hallo von " & _ ButtonGroup.Name & " aus " & ActiveSheet.Name End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Position der aufrufenden Schaltfläche ermitteln

Problem:	Wie kann ich die linke obere Ecke unter der aufrufenden Schaltfläche sowie die Aufschrift des Buttons ermitteln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Aufruf() With ActiveSheet.Buttons(1) MsgBox "Adresse: " & .TopLeftCell.Address & vbCrLf & _ "Text: " & .Caption End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in Abhängigkeit der aufrufenden Schaltfläche eintragen

Problem: Wie kann ich einen UserForm-TextBox-Wert in Abhängigkeit zu der die UserForm aufrufenden Schaltfläche in Zellen eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdButtonA_Click()
    frmEintragen.Tag = "A1"
    frmEintragen.Show
End Sub

Private Sub cmdButtonB_Click()
    frmEintragen.Tag = "A2"
    frmEintragen.Show
End Sub

Private Sub cmdButtonC_Click()
    frmEintragen.Tag = "A3"
    frmEintragen.Show
End Sub

ClassModule: frmEintragen

Private Sub cmdEintragen_Click()
    Range(Me.Tag).Value = txtEintragen.Value
    Unload Me
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Dialoganzeige Blattauswahl wechseln

Problem: Wie kann ich auf Schaltflächendruck im Dialog zwischen zwei Arbeitsblättern wechseln? Der Dialog soll sichtbar bleiben.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmChangeWks.Show
End Sub

ClassModule: frmChangeWks

Private Sub cmdChange_Click()
    If ActiveSheet.Name = "Tabelle1" Then
        Worksheets("Tabelle2").Select
    Else
        Worksheets("Tabelle1").Select
    End If
End Sub

Private Sub cmdContinue_Click()
    Unload Me
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Excel und VBE auf Schaltflächendruck ein- und ausblenden

Problem: Wie kann ich Excel und die Entwicklungsumgebung auf Schaltflächendruck ein- und ausblenden?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmEinblenden

Private Sub cmdEinblenden_Click()
    Application.ScreenUpdating = False
    Application.VBE.MainWindow.Visible = True
    Application.Visible = True
    Unload Me
End Sub

StandardModule: basMain

Sub DialogAufruf()
    Application.VBE.MainWindow.Visible = False
    Application.Visible = False
    frmEinblenden.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schaltfläche in einem Tabellenblatt ein- und ausblenden

Problem:	Wie kann ich eine in sich im Tabellenblatt befindende Schaltfläche zur Betätigung einer zweiten Schaltfläche ein- und ausblenden?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdAusschalten_Click() With cmdAusschalten If .Caption = "Schaltfläche ausblenden" Then cmdTemporaer.Visible = False .Caption = "Schaltfläche einblenden" Else cmdTemporaer.Visible = True .Caption = "Schaltfläche ausblenden" End If End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Name der aufrufenden Schaltfläche abfragen

Problem: Wie kann ich in Abhängigkeit der die Prozedur aufrufenden Schaltfläche bestimmte Zellen auswählen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: mHWH

Sub Schaltflaeche()
    Select Case Application.Caller
        Case "Schaltfläche 1"
            Range("A1").Select
        Case "Schaltfläche 2"
            Range("A51").Select
        Case "Schaltfläche 3"
            Range("A101").Select
    End Select
End Sub

StandardModule: Modul1

Sub Schaltflaeche()
    Select Case Application.Caller
        Case "Schaltfläche 1"
            Range("A1").Select
        Case "Schaltfläche 2"
            Range("A51").Select
        Case "Schaltfläche 3"
            Range("A101").Select
    End Select
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrere Schaltflächen für ein Makro

Problem: In einem Blatt habe ich 4 Schaltflächen, die alle mit demselben Makro verknüpft sind. Wie greife ich innerhalb des Makros auf den Namen der aufrufenden Schaltfläche zu?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn den Schaltflächen zu.

VBA-Code: StandardModule: basMain

```
Sub MakroVerteiler()  
    Select Case Right(Application.Caller, 1)  
        Case "1": Call Makro1  
        Case "2": Call Makro2  
        Case "3": Call Makro3  
        Case "4": Call Makro4  
    End Select  
End Sub  
  
Sub Makro1()  
    MsgBox "Ich bin Makro1"  
End Sub  
  
Sub Makro2()  
    MsgBox "Ich bin Makro2"  
End Sub  
  
Sub Makro3()  
    MsgBox "Ich bin Makro3"  
End Sub  
  
Sub Makro4()  
    MsgBox "Ich bin Makro4"  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datenmaske zur Aufnahme des nächsten Satzes aufrufen

Problem:	Die Datenmaske soll so aufgerufen werden, dass sie zur Aufnahme eines neuen Datensatzes bereit ist.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallMaske() Application.SendKeys "%n" ActiveSheet.ShowDataForm End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datenmaske mit landesspezifischen oder US-Formaten aufrufen

Problem:	Wenn ich die Datenmaske über ShowDataForm aufrufe, werden mir US-Datums- und Zeitformate angezeigt. Wie kann ich das verhindern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ShowDataFormLocalFormats() Range("A1").Select CommandBars.FindControl(ID:=860).Execute End Sub Sub ShowDataFormUSFormats() Range("A1").Select ActiveSheet.ShowDataForm End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datenbank für Datenmaske festlegen

Problem:	Wie kann ich beim VBA-Aufruf der Excel-Datenmaske einen Datenbankbereich vorgeben?
Lösung:	Nachfolgendes Makro in bei XL5/7 in ein englischsprachiges, bei XL8 ein allgemeines Modul einfügen, anpassen und starten.
VBA-Code:	<pre>StandardModule: basMain Sub DatenMaskeZeigen() ActiveWorkbook.Names.Add _ Name:="Database", _ RefersTo:="=" & Worksheets(2).Name & "!" & _ Range("A15:F35").Address Worksheets("Tabelle1").ShowDataForm End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datenmaske beim Aufruf mit Nullen füllen

Problem:	Bei einer Datenmaske sind in den meisten Felder Nullen einzugeben. Kann ich diese vorgeben, ohne die Nullen vorher in das Tabellenblatt einzugeben?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basmain Sub ZeroInMask() SendKeys "%n{0}{tab}{0}{tab}{0}{tab 7}" ActiveSheet.ShowDataForm End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Diagramm in Userform

Problem:	Wie kann ich in eine Userform ein Diagramm einfügen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub GrafikInUserForm() Const csMsg As String = "Die Statistikgrafik kann nicht " & _ "auf CD-ROM gespeichert werden. Speichern Sie die " & _ "Arbeitsmappe auf Ihrer Festplatte und versuchen " & _ "Sie es erneut." Dim sFile As String sFile = ThisWorkbook.Path & "\Statistik.gif" On Error GoTo ERRORHANDLER ActiveWorkbook.Charts(1).Export sFile With frmChart .imgChart.Picture = LoadPicture(sFile) .imgChart.PictureSizeMode = fmPictureSizeModeZoom .Show End With Kill sFile Exit Sub ERRORHANDLER: MsgBox prompt:=csMsg End Sub ClassModule: frmChart Private Sub cmdOK_Click() Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Drucken aus einem DialogSheet

Problem:	Aus einem DialogSheet heraus kann man nicht drucken. Dieses Beispiel zeigt eine Möglichkeit, das Problem zu umgehen. Wichtig sind die Eigenschaften der Druckschaltfläche im DialogSheet.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Public bln As Boolean Sub CallDlg() DialogSheets(1).Show If bln Then ActiveSheet.PrintPreview bln = False End If End Sub Sub CallPrint() bln = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Paßwortabfrage beim Öffnen der Arbeitsmappe

Problem:	Beim Öffnen der Arbeitsmappe soll eine Namen/Passwortabfrage erfolgen. Die Code-Zeile zum Schließend er Arbeitsmappe bei falscher Eingabe ist auskommentiert. Die Routine ist auch in XL5/7 lauffähig.
Lösung:	Geben Sie den Code in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub auto_open() With DialogSheets("dlgPW") .Show If .EditBoxes("edtName").Text <> "Name" Or .EditBoxes("edtPW").Text <> "Passwort" Then Beep MsgBox "Falscher Name oder falsches Passwort!" 'ActiveWorkbook.Close savechanges:=false End If End With End Sub Sub AusEinblenden() Sheets("dlgPW").Visible = xlVeryHidden 'Sheets("dlgPW").Visible = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Label-Anzeige bei Editor-Feldeingabe

Problem:	Über die Schaltfläche soll ein XL5-Dialog aufgerufen werden. Bei Eingabe einer Artikel-Nummer in ein Editfeld soll eine Bezeichnungsfeld die Artikelbenennung anzeigen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ArtikelEintrag() Dim oEdt As EditBox Dim vRow As Variant With ActiveDialog Set oEdt = .EditBoxes("edtNumber") If Len(oEdt.Text) = 6 Then vRow = Application.Match(oEdt.Text, Columns(1), 0) If IsError(vRow) Then MsgBox "Artikelnummer wurde nicht gefunden!" oEdt.Text = "" Else .Labels("lblItem").Caption = Cells(vRow, 2).Value End If End If End With End Sub Sub CallForm() Dim iRow As Integer iRow = 2 With DialogSheets("dlgArtikel") .EditBoxes("edtNumber").Text = "" .Labels("lblItem").Caption = "" .DropDowns("dpdNumber").ListIndex = 0 Do Until IsEmpty(Cells(iRow, 1)) .DropDowns("dpdNumber").AddItem Cells(iRow, 2).Value iRow = iRow + 1 Loop .Show End With End Sub Sub ArtikelAuswahl() Dim oDpd As DropDown Dim vRow As Variant With ActiveDialog Set oDpd = .DropDowns("dpdNumber") .Labels("lblItem").Caption = oDpd.List(oDpd.ListIndex) End With End Sub</pre>



Farbauswahldialog generieren, aufrufen und löschen

Problem:	Wie kann ich auf Schaltflächendruck ein Dialogblatt zur Farbauswahl generieren, den Dialog aufrufen, Farben auswöhlen und das Dialogblatt wieder l÷schen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Public clrArt As Boolean Sub ColorPicker() Dim dlgFarben As DialogSheet Dim btnOK As Button, btnCancel As Button Dim optInterior As OptionButton, optFont As OptionButton Dim txtClr As TextBox Dim intRow As Integer, intCol As Integer Dim l As Integer, t As Integer, intClr As Integer Application.ScreenUpdating = False Set dlgFarben = DialogSheets.Add With dlgFarben .Name = "dlgFarben" With .DialogFrame .Top = 0 .Left = 0 .Height = 200 .Width = 170 .Caption = "Color Picker" End With l = 15 t = 15 Set optInterior = .OptionButtons.Add(l, t, 75, 15) optInterior.Caption = "Zellhintergrund" optInterior.Value = xlOn Set optFont = .OptionButtons.Add(l + 75, t, 75, 15) optFont.Caption = "Schriftfarbe" t = t + 20 For intRow = 1 To 7 l = 15 For intCol = 1 To 8 intClr = intClr + 1 Set txtClr = .TextBoxes.Add(l, t, 16, 16) With txtClr .Interior.ColorIndex = intClr .OnAction = "FarbAuswahl" .Name = "clr" & intClr End With l = l + 18 Next intCol t = t + 18 Next intRow Set btnOK = .Buttons(1) Set btnCancel = .Buttons(2) btnOK.Top = 180 btnOK.Left = 25 btnCancel.Top = 180 btnCancel.Left = 100 Worksheets(1).Select .Show Application.DisplayAlerts = False .Delete Application.DisplayAlerts = True End With End Sub Sub FarbAuswahl() Dim strAc As String Worksheets("Tabelle1").Select Application.ScreenUpdating = True strAc = Application.Caller If DialogSheets("dlgFarben").OptionButtons(1).Value = xlOn Then Selection.Interior.ColorIndex = CInt(Right(strAc, Len(strAc) - 3)) Else Selection.Font.ColorIndex = CInt(Right(strAc, Len(strAc) - 3)) End If End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

AVI-Datei in UserForm abspielen

Problem:	In einer UserForm soll eine AVI-Datei abgespielt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmAVI Private Sub cmdCancel_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim sFile As String sFile = ThisWorkbook.Path & "\fishtank.avi" If Dir(sFile) = "" Then Beep MsgBox sFile & " wurde nicht gefunden!" Else MediaPlayer1.FileName = sFile MediaPlayer1.AutoStart = True End If End Sub StandardModule: Modul1 Sub CallForm() frmAVI.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abspielen einer AVI-Datei in einem UserForm-Steuerelement

Problem: In einem UserForm-Steuerelement soll eine AVI-Datei abgespielt werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmHelmut

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    cmdOK.Enabled = False
    Animation1.Open ThisWorkbook.Path & "\ykohl.avi"
End Sub

StandardModule: Modul1

Sub CallHelmut()
    Dim sFile As String
    sFile = ThisWorkbook.Path & "\ykohl.avi"
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Die AVI-Datei fehlt:" & _
            vbNewLine & sFile
        Exit Sub
    End If
    frmHelmut.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Midi-Datei abspielen

Problem: Wie kann ich eine Midi-Datei, deren Pfad und Name in Zelle A1 stehen, über ein UserForm-Steuerelement abspielen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdDialogAufruf_Click()
    frmMidi.Show
End Sub

StandardModule: basMain

Sub CallForm()
    frmMidi.Show
End Sub

ClassModule: frmMidi

Private Sub cmdWeiter_Click()
    mdp.Stop
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    mdp.FileName = Range("A1").Value
    mdp.AutoStart = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Focus nach Klick auf einen Frame nicht dorthin setzen

Problem:	Wie kann ich verhindern, daß der Focus nach Klick auf einen Frame ein Textfeld verläßt und auf den Frame gesetzt wird? Der Frame darf nicht auf ENABLE = FALSE gesetzt werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmFocus Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub Frame1_MouseDown(ByVal Button As Integer, _ ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single) If Frame1.Tag = "" Then txtFirst.SetFocus Else Controls(Frame1.Tag).SetFocus End If End Sub Private Sub Label1_MouseDown(ByVal Button As Integer, _ ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single) If Frame1.Tag = "" Then txtFirst.SetFocus Else Controls(Frame1.Tag).SetFocus End If End Sub Private Sub txtFirst_Exit(ByVal Cancel As MSForms.ReturnBoolean) Frame1.Tag = txtSecond.Name End Sub Private Sub txtSecond_Exit(ByVal Cancel As MSForms.ReturnBoolean) Frame1.Tag = txtThird.Name End Sub Private Sub txtThird_Exit(ByVal Cancel As MSForms.ReturnBoolean) Frame1.Tag = txtFirst.Name End Sub Private Sub txtFirst_MouseUp(ByVal Button As Integer, _ ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single) Frame1.Tag = txtFirst.Name End Sub Private Sub txtSecond_MouseUp(ByVal Button As Integer, _ ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single) Frame1.Tag = txtSecond.Name End Sub Private Sub txtThird_MouseUp(ByVal Button As Integer, _ ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single) Frame1.Tag = txtThird.Name End Sub StandardModule: basMain Sub CallForm() frmFocus.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenumbruch in simuliertem ControlTipText-Element

Problem:	Der ControlTipText-Eigenschaft eines Steuerlementes kann man keinen Zeilenumbruch zuweisen. Dieses Beispiel demonstriert ein simuliertes ControlTipText auf Basis eines Labels.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmHWH Private Sub cmdCancel_Click() Unload Me End Sub Private Sub Image1_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single) If X > 5 And Y > 5 And X < 70 And Y < 70 Then Label1.Visible = 1 Label1.Caption = "Dies ist" & vbCrLf & "Hans W. Herber" Else Label1.Visible = 0 End If End Sub Private Sub UserForm_Click() End Sub StandardModule: Modul1 Sub CallForm() frmHWH.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-Images animieren

Problem:	In einer UserForm werden Grafiken beim Überfahren mit der Maus animiert dargestellt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmAnimation Dim bln As Boolean Private Sub cmdCancel_Click() Unload Me End Sub Private Sub Image1_Click() bln = False Image1.Picture = LoadPicture(ThisWorkbook.Path & "\prev_button_on.gif") Label1.Caption = CLng(Label1.Caption) + 1 Me.Repaint End Sub Private Sub Image2_Click() bln = False Image2.Picture = LoadPicture(ThisWorkbook.Path & "\prev_button_on.gif") Label1.Caption = CLng(Label1.Caption) - 1 Me.Repaint End Sub Private Sub Image2_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single) Call ChangeImage(Image2, X, Y) End Sub Private Sub Image1_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single) Call ChangeImage(Image1, X, Y) End Sub Private Sub ChangeImage(img As Control, X As Single, Y As Single) Dim sPath As String, sPctA As String, sPctB As String sPath = ThisWorkbook.Path & "\" If img.Name = "Image2" Then sPctA = sPath & "prev_button.gif" sPctB = sPath & "prev_button_on.gif" Else sPctA = sPath & "next_button.gif" sPctB = sPath & "next_button_on.gif" End If If bln = False Then If X > 2 And Y > 2 And X < 16 And Y < 16 Then img.Picture = LoadPicture(sPctA) Else img.Picture = LoadPicture(sPctB) bln = False End If Me.Repaint End If End Sub StandardModule: Modul1 Sub CallForm() frmAnimation.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Überfahren einer Grafik mit der Maus Uhrzeit anzeigen

Problem: Beim Überfahren einer Grafik mit der Maus soll die aktuelle Uhrzeit als Kommentar angezeigt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    Application.DisplayCommentIndicator = _
        ThisWorkbook.Worksheets(1).Range("IV1").Value
    On Error GoTo 0
End Sub

Private Sub Workbook_Open()
    ThisWorkbook.Worksheets(1).Range("IV1").Value = _
        Application.DisplayCommentIndicator
    Application.DisplayCommentIndicator = xlCommentAndIndicator
End Sub

ClassModule: Tabelle1

Dim bln As Boolean

Private Sub Image1_MouseMove( _
    ByVal Button As Integer, _
    ByVal Shift As Integer, _
    ByVal X As Single, _
    ByVal Y As Single)
    Dim rng As Range
    Set rng = Image1.TopLeftCell
    If X > 0 And X < 59 And Y > 0 And Y < 24.5 Then
        If rng.Comment Is Nothing And bln = False Then
            rng.AddComment Format(Time, "hh:mm:ss")
            rng.Comment.Shape.TextFrame.AutoSize = True
            bln = True
        End If
    Else
        If Not rng.Comment Is Nothing Then
            rng.Comment.Delete
            bln = False
        End If
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrundfarbe eines UserForm-Image-Objekts festlegen

Problem: Über Schaltfläche soll die Hintergrundfarbe eines Image-Objekts festgelegt werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
StandardModule: Modul1

Sub CallForm()
    frmColor.Show
End Sub

ClassModule: frmColor

Private Sub cmdOK_Click()
    Zeile = False
    If Zeile = True Then
        Image1.BackColor = HC000&
    Else
        Image1.BackColor = &HFF&
    End If
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sich bewegendes GIF-Grafik in UserForm

Problem: Wie kann ich erreichen, dass sich eine GIF-Grafik innerhalb einer UserForm bewegt?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Declare Sub Sleep Lib "kernel32" _
    (ByVal dwMilliseconds As Long)

Sub CallForm()
    frmAnimation.Show
End Sub

ClassModule: frmAnimation

Private Sub cmdStart_Click()
    Dim iCol As Integer
    Dim iRow As Integer
    For iRow = 1 To 4
        For iCol = 6 To 216
            Sleep SpinButton1.Value
            Image1.Left = Image1.Left + 1
            DoEvents
        Next iCol
        Image1.Top = Image1.Top + 12
        For iCol = 216 To 6 Step -1
            Sleep SpinButton1.Value
            Image1.Left = Image1.Left - 1
            DoEvents
        Next iCol
        Image1.Top = Image1.Top + 12
    Next iRow
    Image1.Top = 6
    Image1.Left = 6
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub SpinButton1_Change()
    TextBox1.Text = SpinButton1.Value
End Sub

Private Sub UserForm_Initialize()
    SpinButton1.Value = 5
End Sub
```



Beispiele zum Einsatz der InputBox

Problem:	10 Beispiele zum Einsatz der eingebauten InputBox
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn Schaltflächen zu.
VBA-Code:	StandardModule: Modul1 <pre>Sub CallVBA() Dim sTxt As String sTxt = InputBox("Bitte Eingabe tätigen:") If sTxt = "" Then Exit Sub MsgBox sTxt End Sub Sub CallXL() Dim vTxt As Variant vTxt = Application.InputBox("Bitte Eingabe tätigen:") If vTxt = False Then Exit Sub MsgBox vTxt End Sub Sub SetPrompt() Dim sTxt As String, sPrompt As String sPrompt = "Nutzen Sie bitte folgende Syntax:" & vbCrLf & _ " 'dd.mm.yyyy' oder" & vbCrLf & _ " 'dd/mm/yyyy' oder" & vbCrLf & _ " 'dd-mm-yyyy':" sTxt = InputBox(prompt:=sPrompt) On Error GoTo ERRORHANDLER MsgBox CDate(sTxt) Exit Sub ERRORHANDLER: MsgBox "Kein gültiges Datumsformat!" End Sub Sub SetTitle() Dim sTxt As String, sTitle As String sTitle = Application.UserName & " says:" sTxt = InputBox(prompt:="Eingabe:", Title:=sTitle) If sTxt = "" Then Exit Sub MsgBox sTxt End Sub Sub SetDefault() Dim sTxt As String, sPrompt As String, sDefault As String sPrompt = "Eingangsdatum eingeben:" sDefault = Format(Date - 3, "dd.mm.yyyy") sTxt = InputBox(prompt:=sPrompt, Default:=sDefault) On Error GoTo ERRORHANDLER MsgBox CDate(sTxt) Exit Sub ERRORHANDLER: MsgBox "Kein gültiges Datumsformat!" End Sub Sub SelDefault() Dim sTxt As String, sPrompt As String, sDefault As String sPrompt = "Eingangsdatum eingeben:" sDefault = Format(Date - 3, "dd.mm.yyyy") SendKeys "{home}{right 3}+{right 2}" sTxt = InputBox(prompt:=sPrompt, Default:=sDefault) On Error GoTo ERRORHANDLER MsgBox CDate(sTxt) Exit Sub ERRORHANDLER: MsgBox "Kein gültiges Datumsformat!" End Sub Application.InputBox Sub InputBoxPositionA() Dim sTxt As String, sPrompt As String sPrompt = "Eingabe:" sTxt = InputBox(prompt:="Eingabe:", xpos:=1000, ypos:=1000) End Sub Sub InputBoxPositionB() Dim rng As Range Dim iLeft As Integer, iTop As Integer Dim sTxt As String, sPrompt As String Set rng = Range("B3") iLeft = rng.Left + 23 iTop = rng.Top + 16 sPrompt = "Eingabe:" sTxt = Application.InputBox(_ prompt:="Eingabe:", Left:=iLeft, Top:=iTop) End Sub Sub CallHelpA() Dim sTxt As String, sPrompt As String, sFile As String sFile = "..\..\2001\Beispiele\Kontext.chm" If Dir(sFile) = "" Then Beep MsgBox "Hilfedatei existiert nicht!" Exit Sub End If sPrompt = "Eingabe:" sTxt = InputBox(_ prompt:="Eingabe:", _ HelpFile:=sFile, _ Context:="1001") End Sub Sub CallHelpB() Dim sTxt As String, sPrompt As String, sFile As String sFile = "..\..\2001\Beispiele\Kontext.chm" If Dir(sFile) = "" Then Beep MsgBox "Hilfedatei existiert nicht!" Exit Sub End If sPrompt = "Eingabe:" sTxt = Application.InputBox(_ prompt:="Eingabe:", _ HelpFile:=sFile, _ HelpContextID:="1001") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Inputbox wahlweise zur Zahlen- oder Zeichenfolgeeingabe

Problem: Über Optionsfelder wird eine InputBox zur Zahlen- oder eine zur Zeichenfolgeeingabe aufgerufen.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: frmEingabe

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub optNumber_Click()
    Dim vValue As Variant
    Dim iRow As Integer
    vValue = Application.InputBox( _
        prompt:="Zahleneingabe", _
        Title:="Zahl", _
        Type:=1)
    If vValue = False Then Exit Sub
    If IsEmpty(Cells(1, 1)) Then
        iRow = 1
    Else
        iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
    End If
    Cells(iRow, 1).Value = vValue
End Sub

Private Sub optString_Click()
    Dim vValue As Variant
    Dim iRow As Integer
    vValue = Application.InputBox( _
        prompt:="Zeichenfolgeeingabe", _
        Title:="Zeichenfolge", _
        Type:=2)
    If vValue = False Then Exit Sub
    If IsEmpty(Cells(1, 1)) Then
        iRow = 1
    Else
        iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
    End If
    Cells(iRow, 1).Value = vValue
End Sub

StandardModule: Modul1

Sub CallForm()
    frmEingabe.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahleneingaben in InputBox erzwingen

Problem: Es werden 5 verschiedene Varianten gezeigt, eine Zahleneingabe in eine InputBox zu erzwingen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn fünf Schaltflächen zu.

VBA-Code:

```
StandardModule: Modul1

Sub InputBoxA()
    Dim lValue As Long
    On Error GoTo ERRORHANDLER
    lValue = InputBox("Bitte eine Ganzzahl eingeben:")
    MsgBox lValue
    Exit Sub
ERRORHANDLER:
    MsgBox "Nur Zahlen erlaubt!"
    Resume
End Sub

Sub InputBoxB()
    Dim lValue As Long
    lValue = Application.InputBox( _
        prompt:="Bitte eine Ganzzahl eingeben:", _
        Type:=1)
    MsgBox lValue
End Sub

Sub InputBoxC()
    Dim vValue As Variant
    vValue = "Start"
    Do While Not IsNumeric(vValue)
        vValue = InputBox("Bitte eine Ganzzahl eingeben:")
    Loop
    MsgBox vValue
End Sub

Sub InputBoxD()
    Dim vValue As Variant
    Do
        vValue = InputBox("Bitte eine Ganzzahl eingeben:")
        If IsNumeric(vValue) Then Exit Do
    Loop
    MsgBox vValue
End Sub

Sub InputBoxE()
    Dim lValue As Long
    On Error Resume Next
    Err.Raise 6
    Do While Err
        Err.Clear
        lValue = InputBox("Bitte eine Ganzzahl eingeben:")
    Loop
    MsgBox lValue
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

InputBox-Eingabe auf gültige Zahl prüfen

Problem:	Die Eingabe in eine InputBox soll auf eine gültige Zahl geprüft werden. Alternative: Verwendung einer Application.InputBox mit Typ-Festlegung.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Berechnung() Dim vWert As Variant Do While vWert = "" Or Not IsNumeric(vWert) vWert = InputBox("Wert:") If IsNumeric(vWert) Then Exit Do MsgBox "Einen Wert, zum Teufel!" Loop MsgBox "Danke!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über interaktive InputBox Bereich auswählen und zurückgeben

Problem:	Über eine interaktive InputBox soll ein Bereich ausgewählt und zurückgegeben werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SelectCell() Dim rng As Range Set rng = Application.InputBox("Bitte einen Bereich auswählen:", Type:=8) rng.Select MsgBox "Bedanke mich für die Auswahl von " & rng.Address & "!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Numerische InputBox-Werte fortlaufend in eine Spalte eintragen

Problem:	Wie kann ich durch InputBox-Aufrufe eingelesene Werte nacheinander in eine Spalte einlesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub InputWerte() Dim var As Variant Dim iRow As Integer If IsEmpty(Cells(1, 1)) Then iRow = 1 Else iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 End If var = 1 Do Until var = False var = Application.InputBox(_ prompt:="Wert:", _ Default:=var, _ Type:=1) If var = False Then Exit Sub Cells(iRow, 1).Value = var iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Labelwerte werden beim Verlassen einer TextBox berechnet

Problem:	Der Wert eines UserForm-Labels soll um einen über TextBox-Eingabe ermittelten Prozentsatz vermindert und nach Verlassen der TextBox in ein weiteres Label eingetragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmPercent Private Sub cmdCancel_Click() Unload Me End Sub Private Sub txtPercent_Exit(ByVal Cancel As MSForms.ReturnBoolean) Dim dValue As Double dValue = CDBl(Label1.Caption) dValue = dValue - ((CDBl(txtPercent.Text) / 100) * dValue) Label2.Caption = Format(dValue, "0.00") End Sub StandardModule: Modul1 Sub DialogAufruf() frmPercent.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ein UserForm-Label in einer 3D-Darstellung anzeigen

Problem:	Ein UserForm-Label soll in 3D-Darstellung angezeigt werden, wobei das Anklicken des Labels mitberücksichtigt werden soll.
Lösung:	Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmTogglen Private Sub cmdOK_Click() Unload Me End Sub Private Sub lblClick_MouseDown(_ ByVal Button As Integer, _ ByVal Shift As Integer, _ ByVal X As Single, _ ByVal Y As Single) Label1.Move Left:=lblClick.Left + lblClick.Width - 1 Label2.Move Top:=lblClick.Top + lblClick.Height - 1 lblClick.ForeColor = RGB(120, 120, 120) End Sub Private Sub lblClick_MouseUp(_ ByVal Button As Integer, _ ByVal Shift As Integer, _ ByVal X As Single, _ ByVal Y As Single) Label1.Move _ Left:=lblClick.Left, _ Top:=lblClick.Top, _ Width:=1, _ Height:=lblClick.Height Label2.Move _ Left:=lblClick.Left, _ Top:=lblClick.Top, _ Width:=lblClick.Width, _ Height:=1 lblClick.ForeColor = RGB(0, 0, 0) End Sub Private Sub UserForm_Initialize() Label1.Width = 1 Label2.Height = 1 Label1.Move Top:=lblClick.Top Label2.Move Left:=lblClick.Left lblClick.ZOrder 1 End Sub StandardModule: Modul1 Sub DialogAufruf() frmTogglen.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelltext als Hilfetext in einer UserForm anzeigen

Problem: Der Text in Zelle A1 soll in einer UserForm-Hilfe-TextBox mit vertikaler ScrollBar angezeigt werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmHelp

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub tglHelp_Click()
    On Error Resume Next
    With txtHelp
        .Visible = tglHelp.Value
        .ScrollBars = fmScrollBarsBoth
        .SetFocus
        .SelStart = 0
    End With
End Sub

Private Sub UserForm_Initialize()
    txtHelp.Visible = False
    'txtHelp.Text = WorksheetFunction.Substitute(Range("A1").Value, vbLf, "")
    txtHelp.Text = Range("A1").Value
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmHelp.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktuelle Zeit in UserForm darstellen

Problem: Wie kann ich die aktuelle Zeit mit Aktualisierung in einer UserForm darstellen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: modMain

Public NextTime As Date

Sub UpdateClock()
    frmZeit.lblZeit.Caption = Format(Time, "hh:mm:ss")
    NextTime = Now + TimeValue("00:00:01")
    frmZeit.Repaint
    Application.OnTime NextTime, "UpdateClock"
End Sub

Sub CallForm()
    frmZeit.Show
End Sub

ClassModule: frmZeit

Private Sub cmdWeiter_Click()
    Application.OnTime NextTime, "UpdateClock", , False
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Call UpdateClock
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datum aus Zelle in UserForm-Label eintragen

Problem:	Wie kann ich ein Zelldatum in ein Label einer UserForm eintragen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmDatum Private Sub cmdDatum_Click() Label1.Caption = Range("A1").Text End Sub Private Sub cmdWeiter_Click() Unload Me End Sub StandardModule: basMain Sub CallForm() frmDatum.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-Elemente auslesen

Problem: Wie kann ich die Caption- bzw. Text-Eigenschaften der Elemente einer UserForm auslesen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code: ClassModule: frmAuslesen

```
Private Sub cmdAuslesen_Click()
    Dim cntLabel As Control, cntText As Control
    Dim arr(1 To 2, 1 To 4)
    Dim iCounterA As Integer, iCounterB As Integer
    For iCounterA = 1 To 2
        For iCounterB = 1 To 4
            If iCounterA = 1 Then
                arr(iCounterA, iCounterB) = _
                    Controls("lblLC" & iCounterB).Caption
            Else
                arr(iCounterA, iCounterB) = _
                    Controls("txtLC" & iCounterB).Text
            End If
        Next iCounterB
    Next iCounterA
    For iCounterA = 1 To 2
        For iCounterB = 1 To 4
            If iCounterA = 1 Then
                MsgBox "Das " & iCounterB & ". Label hat den Wert: " & _
                    & arr(iCounterA, iCounterB)
            Else
                MsgBox "Das erste " & iCounterB & _
                    ". Textfeld hat den Wert: " & _
                    & arr(iCounterA, iCounterB)
            End If
        Next iCounterB
    Next iCounterA
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmAuslesen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in Tabelle suchen und in UserForm ausgeben

Problem: Wie kann ich einen in einer UserForm eingegebenen Suchbegriff in einer Tabelle suchen und dann in einem Label in der UserForm ausgeben lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSuchen

Private Sub cmdEintragen_Click()
    Dim rng As Range
    Set rng = Columns(1).Find( _
        what:=txtSearch.Text, _
        lookat:=xlWhole, _
        LookIn:=xlValues)
    If rng Is Nothing Then
        lblfound.Caption = "Nicht gefunden!"
    Else
        lblfound.Caption = rng.Offset(1, 0)
    End If
End Sub

Private Sub cmdContinue_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmSuchen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalt als Label-Aufschrift in UserForm

Problem:	Wie kann ich einen bestimmten Zellinhalt als Aufschrift eines Labelfeldes in einer UserForm vorgeben? Zeigt der Zellinhalt eine Fehlermeldung, soll diese nicht übernommen werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmWertAusZelle Private Sub cmdContinue_Click() Unload Me End Sub Private Sub UserForm_Initialize() If IsError(Range("A1")) = False Then lblTxt.Caption = Range("A1") End If End Sub StandardModule: basMain Sub CallForm() frmWertAusZelle.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In einer Schleife auf Elemente einer UserForm zugreifen

Problem: Wie kann ich in XL8 auf einzelne, vorher durch VBA-Code erzeugte Label zugreifen und deren Caption neu festlegen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

StandardModule: basMain

Sub CallForm()
    frmLabel.Show
End Sub

ClassModule: frmLabel

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim lbl As MSForms.Label
    Dim iCounter As Integer
    For iCounter = 1 To 3
        Set lbl = Controls.Add("Forms.Label.1")
        With lbl
            .Top = iCounter * 15
            .Left = 20
            .Caption = lbl.Name
        End With
    Next iCounter
    For iCounter = 1 To 3
        Controls("Label" & iCounter).Caption = _
            "Testlabel Nr. " & iCounter
    Next iCounter
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattauswahl nach ListBox-Selektion

Problem:	Bei Auswahl eines ListBox-Elementes wird das entsprechende Tabellenblatt ausgewählt. Die Blattnamen werden bei Aktivieren des Blattes neu eingelesen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub lstWks_Click() Worksheets(lstWks.Value).Select End Sub Private Sub Worksheet_Activate() Dim wks As Worksheet lstWks.Clear For Each wks In Worksheets If wks.Name <> "Tabelle1" Then lstWks.AddItem wks.Name End If Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verschieben von ListBox-Eintragen nach oben und unten

Problem: Die Felder einer ListBox sollen nach oben oder unten verschoben werden können.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMove

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdTop_Click()
    Dim sTxt As String
    If lstValues.ListIndex = -1 Then
        Beep
        MsgBox "Sie müssen ein Element auswählen"
        Exit Sub
    End If
    With lstValues
        If .ListIndex > 0 Then
            sTxt = .List(.ListIndex - 1)
            .List(.ListIndex - 1) = .List(.ListIndex)
            .List(.ListIndex) = sTxt
            .Selected(.ListIndex - 1) = True
        End If
    End With
End Sub

Private Sub cmdDown_Click()
    Dim sTxt As String
    If lstValues.ListIndex = -1 Then
        Beep
        MsgBox "Sie müssen ein Element auswählen"
        Exit Sub
    End If
    With lstValues
        If .ListIndex < .ListCount - 1 Then
            sTxt = .List(.ListIndex + 1)
            .List(.ListIndex + 1) = .List(.ListIndex)
            .List(.ListIndex) = sTxt
            .Selected(.ListIndex + 1) = True
        End If
    End With
End Sub

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_Initialize()
    Dim iMonth As Integer
    For iMonth = 1 To 12
        lstValues.AddItem Format(DateSerial(1, iMonth, 1), "mmmm")
    Next iMonth
End Sub

StandardModule: Modul1

Sub CallForm()
    frmMove.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textbox-Werte gem. UserForm-ListBox-Auswahl in Tabelle eintragen

Problem: Neben den in einer UserForm-ListBox ausgewählten Einträgen aus Spalte A sollen TextBox-Werte eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSetValues

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdInsert_Click()
    Dim iRow As Integer
    For iRow = 0 To lstValues.ListCount - 1
        If lstValues.Selected(iRow) Then
            Cells(iRow + 1, 2).Value = txtValue.Text
        End If
    Next iRow
End Sub

Private Sub UserForm_Initialize()
    lstValues.List = Range("A1").CurrentRegion.Value
End Sub

StandardModule: Modul1

Sub InsertValues()
    frmSetValues.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufteilen mehrzeiliger Zelleninhalte in UserForm-ListBox

Problem:	Der Inhalt aus Spalte A soll in einer UserFormListBox in jeweils zwei Zeilen aufgeteilt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallForm() frmSplit.Show End Sub ClassModule: frmSplit Private Sub cmdCancel_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim iRow As Integer Dim sTxt As String iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) sTxt = Cells(iRow, 1).Value If InStr(sTxt, vbLf) Then With lstValues .AddItem Left(sTxt, InStr(sTxt, vbLf) - 1) .AddItem Right(sTxt, Len(sTxt) - InStr(sTxt, vbLf)) End With End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblatteintrag gem. ListBox-Auswahl suchen

Problem:	Bei Auswahl eines Monats im Listenfild soll die entsprechende Tabellenzelle ausgewählt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub lstMonths_Change() Cells(lstMonths.ListIndex + 1, 1).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Darstellung der verschiedenen Erscheinungsformen von ListBoxes

Problem: Die verschiedenen Erscheinungsformen einer UserForm-ListBox werden dargestellt.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmListBox

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdCheck_Click()
    ListBox1.ListStyle = fmListStyleOption
    ListBox1.MultiSelect = fmMultiSelectMulti
End Sub

Private Sub cmdOption_Click()
    ListBox1.ListStyle = fmListStyleOption
    ListBox1.MultiSelect = fmMultiSelectSingle
End Sub

Private Sub cmdPlain_Click()
    ListBox1.ListStyle = fmListStylePlain
End Sub

Private Sub UserForm_Initialize()
    ListBox1.List = Range("A1").CurrentRegion.Value
End Sub

StandardModule: Modul1

Sub CallForm()
    frmListBox.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Begriff in der ersten Spalte einer ListBox suchen

Problem:	In einer UserForm soll ein Suchbegriff in der ersten Spalte einer mehrspaltigen Listbox gesucht und markiert werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmSearch Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdSearch_Click() Dim iRow As Integer For iRow = 0 To lstData.ListCount - 1 If lstData.List(iRow, 0) = txtSearch.Text Then lstData.Selected(iRow) = True End If Next iRow End Sub Private Sub UserForm_Click() End Sub Private Sub UserForm_Initialize() lstData.List = Range("A1").CurrentRegion.Value End Sub StandardModule: Modul1 Sub CallForm() frmSearch.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausgewählte Einträge einer mehrspaltigen ListBox in Tabellenblatt

Problem:	Ausgewählte Einträge aus einer mehrspaltigen UserForm-ListBox sollen in Tabelle2 übernommen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmWrite Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdWrite_Click() Dim iRow As Integer, iCol As Integer, iCounter As Integer For iRow = 0 To lstValues.ListCount - 1 If lstValues.Selected(iRow) Then iCounter = iCounter + 1 For iCol = 1 To 4 Worksheets("Tabelle2").Cells(iCounter, iCol).Value = lstValues.List(iRow, iCol - 1) Next iCol End If Next iRow End Sub Private Sub UserForm_Initialize() lstValues.List = Range("A1").CurrentRegion.Value End Sub StandardModule: Modul1 Sub CallForm() frmWrite.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einer zweidimensionalen UserForm-ListBox Elemente hinzufügen

Problem: Einer zweidimensionalen UserForm-ListBox sollen Textfeld-Inhalte hinzugefügt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAddElements

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim arr(0 To 0, 0 To 1) As String
    arr(0, 0) = txtA.Value
    arr(0, 1) = txtB.Value
    lstValues.AddItem arr(0, 0)
    lstValues.List(lstValues.ListCount - 1, 1) = arr(0, 1)
End Sub

StandardModule: Modul1

Sub CallForm()
    frmAddElements.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In UserForm-ListBox nur Datensätze gleichen Datums auswählen

Problem: In einer UserForm-ListBox dürfen nur Datensätze gleichen Datums ausgewählt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMultiSelect

Dim bln As Boolean

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub lstValues_Change()
    Dim iRow As Integer, iCount As Integer
    Dim sTxt As String
    If bln Then
        bln = False
        Exit Sub
    End If
    sTxt = lstValues.List(lstValues.ListIndex, 0)
    For iRow = 0 To lstValues.ListCount - 1
        If lstValues.Selected(iRow) Then
            iCount = iCount + 1
            If iCount = 2 Then Exit For
        End If
    Next iRow
    If iCount > 1 Then
        For iRow = 0 To lstValues.ListCount - 1
            If iRow <> lstValues.ListIndex Then
                If lstValues.Selected(iRow) And CStr(lstValues.List(iRow, 0)) <> sTxt Then
                    MsgBox "Es dürfen nur Datensätze eines Datums ausgewählt werden!"
                    bln = True
                    lstValues.Selected(lstValues.ListIndex) = False
                    Exit For
                End If
            End If
        Next iRow
    End If
End Sub

Private Sub UserForm_Initialize()
    lstValues.List = Range("A1").CurrentRegion.Value
End Sub

StandardModule: Modul1

Sub CallForm()
    frmMultiSelect.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schaltfläche erst aktivieren, wenn 3. Element ausgewählt

Problem: In einer UserForm soll eine OK-Schaltfläche erst aktiviert werden, wenn in einer ListBox das 3. Element ausgewählt wird.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmThree

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim iCounter As Integer, iCount As Integer
    For iCounter = 0 To lstValues.ListCount - 1
        If lstValues.Selected(iCounter) Then
            iCount = iCount + 1
            Cells(iCount, 2).Value = lstValues.List(iCounter)
        End If
    Next iCounter
    Unload Me
End Sub

Private Sub lstValues_Change()
    Dim iCounter As Integer, iCount As Integer
    For iCounter = 0 To lstValues.ListCount - 1
        If lstValues.Selected(iCounter) Then
            iCount = iCount + 1
        End If
    Next iCounter
    If iCount = 3 Then
        cmdOK.Enabled = True
    Else
        cmdOK.Enabled = False
    End If
End Sub

Private Sub UserForm_Initialize()
    lstValues.List = Range("A1").CurrentRegion.Value
End Sub

StandardModule: basMain

Sub CallForm()
    frmThree.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ListBox-Elemente beim Anklicken in Tabelle übertragen

Problem: In einer UserForm-ListBox sollen Zahlen von 1 bis 7 gelistet und bei Mausklick in eine auszuwählende Spalte eingegeben werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmInsert

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub lstInsert_Click()
    Dim iRow As Integer, iCol As Integer
    iCol = CInt(cboColumn.Value)
    iRow = Cells(Rows.Count, iCol).End(xlUp).Row + 1
    If iRow = 2 And IsEmpty(Cells(1, iCol)) Then
        iRow = 1
    End If
    Cells(iRow, iCol).Value = lstInsert.Value
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 256
        cboColumn.AddItem iCounter
    Next iCounter
    cboColumn.ListIndex = 0
    For iCounter = 1 To 7
        lstInsert.AddItem iCounter
    Next iCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmInsert.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spaltenwerte in UserForm-ListBox übernehmen und ausgeben

Problem: Spalte A soll in eine UserForm-ListBox eingelesen werden. Die in der ListBox angeklickten Werte sind in Spalte C zu übernehmen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmValues

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub lstValues_Click()
    Dim iRow As Integer
    iRow = Cells(Rows.Count, 3).End(xlUp).Row + 1
    Cells(iRow, 3).Value = lstValues.Text
End Sub

Private Sub UserForm_Initialize()
    lstValues.List = Range("A1").CurrentRegion.Value
End Sub

StandardModule: Modul1

Sub CallForm()
    frmValues.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ListBox-Einträge entsprechend TextBox-Eingaben auswählen

Problem: Über eine TextBox sollen ListBox-Einträge je nach Buchstabeneingabe selektiert werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMonths

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub txtMonth_Change()
    Dim iRow As Integer
    For iRow = 0 To lstMonths.ListCount - 1
        If LCase(Left(lstMonths.List(iRow), _
            txtMonth.TextLength)) = LCase(txtMonth.Text) Then
            lstMonths.ListIndex = iRow
            Exit For
        Else
            lstMonths.ListIndex = -1
        End If
    Next iRow
End Sub

Private Sub UserForm_Initialize()
    Dim iRow As Integer
    For iRow = 1 To 12
        lstMonths.AddItem Format(DateSerial(1, iRow, 1), "mmm")
    Next iRow
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmMonths.Show
End Sub
```



UserForm-ListBox in Abhängigkeit von Datumsauswahl füllen

Problem:	Über eine UserForm sollen Start- und Enddatum festgelegt und die passenden Daten in eine mehrspaltige ListBox eingelesen werden.
Lösung:	Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub DialogAufruf() frmNamen.Show End Sub ClassModule: frmNamen Private Sub cboBis_Change() Call FillList End Sub Private Sub cboVon_Change() Dim lCounter As Long cboBis.Clear For lCounter = CDate(cboVon.Value) To DateSerial(2002, 1, 31) cboBis.AddItem Format(lCounter, "dd.mm.yy") Next lCounter cboBis.ListIndex = 0 Call FillList End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim iCounter As Integer, iCol As Integer Unload Me Workbooks.Add Range("A1").Value = "Termine vom " & cboVon.Value & " bis " & cboBis.Value For iCounter = 0 To lstNamen.ListCount - 1 For iCol = 1 To 4 Cells(iCounter + 2, iCol) = lstNamen.List(iCounter, iCol - 1) Next iCol Next iCounter ActiveSheet.PrintPreview ActiveWorkbook.Close savechanges:=False End Sub Private Sub UserForm_Initialize() Dim lCounter As Long For lCounter = DateSerial(2002, 1, 1) To DateSerial(2002, 1, 31) cboVon.AddItem Format(lCounter, "dd.mm.yy") Next lCounter cboVon.ListIndex = 0 End Sub Private Sub FillList() Dim arr() As Variant Dim lCounter As Long, iCounter As Integer, iCol As Integer If cboVon.Value = "" Or cboBis.Value = "" Then Exit Sub lstNamen.Clear lCounter = 1 Do Until IsEmpty(Cells(lCounter, 1)) If CDb1(Cells(lCounter, 4)) >= CDb1(CDate(cboVon.Value)) And _ CDb1(Cells(lCounter, 4)) <= CDb1(CDate(cboBis.Value)) Then ReDim Preserve arr(0 To 3, 0 To iCounter) For iCol = 1 To 4 arr(iCol - 1, iCounter) = Cells(lCounter, iCol) Next iCol iCounter = iCounter + 1 End If lCounter = lCounter + 1 Loop lstNamen.Column = arr End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eintrag eines ListBox-Wertes in die aktive Zelle

Problem: Bei Anklicken eines ListBox-Wertes soll dieser in die aktive Zelle eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub lstData_Click()
    ActiveCell.Select
    ActiveCell.Value = lstData.Value
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dialogaufruf, wenn zwei Zellen Werte enthalten

Problem: Bei Eingaben in Spalte A und B soll jeweils dann, wenn beide Zellen einer Zeile ausgefüllt sind, eine UserForm aufgerufen werden. Die Zellinhalte sind in den in der UserForm auszuwählenden Blättern einzufügen.

Lösung: Den nachfolgenden Code in die benannten Module einfügen.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Column > 2 Then Exit Sub
    If Not IsEmpty(Cells(Target.Row, 1)) And Not _
        IsEmpty(Cells(Target.Row, 2)) Then
        frmSelect.Show
    End If
End Sub

ClassModule: frmSelect

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim intCounter As Integer, intRow As Integer
    For intCounter = 0 To lstSheets.ListCount - 1
        If lstSheets.Selected(intCounter) Then
            With Worksheets(lstSheets.List(intCounter))
                intRow = .Cells(Rows.Count, 1) _
                    .End(xlUp).Row + 1
                .Cells(intRow, 1) = ActiveCell _
                    .Offset(0, -1).Value
                .Cells(intRow, 2) = ActiveCell.Value
            End With
        End If
    Next intCounter
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim intCounter As Integer
    For intCounter = ActiveSheet.Index + 1 To _
        Worksheets.Count
        lstSheets.AddItem Worksheets(intCounter).Name
    Next intCounter
End Sub
```



Spalten über UserForm-Auswahl ein- und ausblenden

Problem:	Über eine UserForm sollen die sichtbar bleibenden Spalten vorgegeben werden.
Lösung:	Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.
VBA-Code:	ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Dim oCtr As CommandBarPopup Set oCtr = Application.CommandBars("Worksheet Menu Bar") _ .FindControl(ID:=30007) On Error GoTo ERRORHANDLER oCtr.Controls("Spalten").Delete ERRORHANDLER: End Sub Private Sub Workbook_Open() Dim oCtr As CommandBarPopup Dim oBtn As CommandBarButton Set oCtr = Application.CommandBars("Worksheet Menu Bar") _ .FindControl(ID:=30007) On Error Resume Next oCtr.Controls("Spalten").Delete On Error GoTo 0 Set oBtn = oCtr.Controls.Add With oBtn .Caption = "Spalten" .OnAction = "DialogAufruf" .Style = msoButtonCaption End With End Sub ClassModule: frmHidden Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() ActiveSheet.Protect Unload Me End Sub Private Sub lstColumns_Change() Dim iCounter As Integer For iCounter = 0 To lstColumns.ListCount - 1 If lstColumns.Selected(iCounter) Then Columns(iCounter + 1).Hidden = False Else Columns(iCounter + 1).Hidden = True End If Next iCounter Range("A1").Select End Sub Private Sub UserForm_Initialize() ActiveSheet.Unprotect lstColumns.Column = Range("A1").CurrentRegion.Rows(1).Value End Sub StandardModule: Modul1 Sub DialogAufruf() frmHidden.Show End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blätter in UserForm-ListBox auswählen und speichern

Problem: Die in einer UserForm-ListBox auszuwählenden Arbeitsblätter sind in einer neuen Arbeitsmappe zu speichern.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: frmSheets

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim arr() As String
    Dim iRow As Integer, iCounter As Integer
    Dim sPath As String
    Application.ScreenUpdating = False
    sPath = Application.DefaultFilePath & "\"
    For iRow = 0 To lstSheets.ListCount - 1
        If lstSheets.Selected(iRow) Then
            iCounter = iCounter + 1
            ReDim Preserve arr(1 To iCounter)
            arr(iCounter) = Worksheets(lstSheets.List(iRow)).Name
        End If
    Next iRow
    Worksheets(arr).Copy
    Application.DisplayAlerts = False
    ActiveWorkbook.SaveAs sPath & "test.xls"
    Application.DisplayAlerts = True
    ActiveWorkbook.Close savechanges:=False
    Application.ScreenUpdating = True
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim wks As Worksheet
    For Each wks In Worksheets
        lstSheets.AddItem wks.Name
    Next wks
End Sub

StandardModule: Modul1

Sub CallForm()
    frmSheets.Show
End Sub
```



Daten in ListBoxes verschieben und sortieren

Problem:	Die Werte aus Spalte A sollen in eine UserForm-ListBox eingelesen werden. In der UserForm sollen die Werte von einer ListBox zur anderen verschoben und jeweils nach dem Sortierkriterium der Anordnung in der Tabelle sortiert werden können.
Lösung:	Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.
VBA-Code:	ClassModule: frmListen Private Sub cmdHer_Click() Dim arr As Variant Dim iRow As Integer, iCount As Integer If lstB.ListIndex = -1 Then Beep MsgBox "Sie müssen ein Element auswählen!" Exit Sub End If arr = ActiveSheet.Range("A1").CurrentRegion lstA.AddItem lstB.Value lstB.RemoveItem lstB.ListIndex Workbooks.Add Range(Cells(1, 1), Cells(lstA.ListCount, 1)).Value = lstA.List iCount = Application.CustomListCount Application.AddCustomList arr Range("A1").CurrentRegion.Sort key1:=Range("A1"), order1:=xlAscending, _ ordercustom:=iCount + 2, header:=xlNo If IsEmpty(Range("A2")) Then lstA.Clear lstA.AddItem Range("A1").Value Else lstA.List = Range("A1").CurrentRegion.Value End If Application.DeleteCustomList iCount + 1 ActiveWorkbook.Close savechanges:=False End Sub Private Sub cmdHin_Click() Dim arr As Variant Dim iRow As Integer, iCount As Integer If lstA.ListIndex = -1 Then Beep MsgBox "Sie müssen ein Element auswählen!" Exit Sub End If arr = ActiveSheet.Range("A1").CurrentRegion lstB.AddItem lstA.Value lstA.RemoveItem lstA.ListIndex Workbooks.Add Range(Cells(1, 1), Cells(lstB.ListCount, 1)).Value = lstB.List iCount = Application.CustomListCount Application.AddCustomList arr Range("A1").CurrentRegion.Sort _ key1:=Range("A1"), order1:=xlAscending, _ ordercustom:=iCount + 2, header:=xlNo If IsEmpty(Range("A2")) Then lstB.Clear lstB.AddItem Range("A1").Value Else lstB.List = Range("A1").CurrentRegion.Value End If Application.DeleteCustomList iCount + 1 ActiveWorkbook.Close savechanges:=False End Sub Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub UserForm_Initialize() lstA.List = Range("A1").CurrentRegion.Value End Sub StandardModule: basMain Sub CallForm() frmListen.Show End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelleauswahl gemäß ListBox

Problem: Wie kann ich eine Reihe von Zelladressen in eine ListBox übernehmen und nach Auswahl in der ListBox ansteuern?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAuswahl

Private Sub cmdAuswahl_Click()
    Dim iCounter As Integer
    Dim sTxt As String
    For iCounter = 0 To lstAuswahl.ListCount - 1
        If lstAuswahl.Selected(iCounter) Then
            sTxt = sTxt & lstAuswahl.List(iCounter) & ","
        End If
    Next iCounter
    sTxt = Left(sTxt, Len(sTxt) - 1)
    Range(sTxt).Select
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    lstAuswahl.List = Range("A1").CurrentRegion.Value
End Sub

StandardModule: basMain

Sub CallForm()
    frmAuswahl.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in zweispaltiger ListBox rechts- und linksbündig darstellen

Problem: Die Spalten A und C sollen in einer UserForm-ListBox linksbündig, Spalte B rechtsbündig dargestellt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAlign

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim arr(1 To 10, 1 To 3)
    Dim iRow As Integer, iCol As Integer
    For iRow = 1 To 10
        For iCol = 1 To 3
            If iCol = 2 Then
                arr(iRow, iCol) = String(8 - _
                    Len(Cells(iRow, iCol).Text), " ") & _
                    Cells(iRow, iCol).Text
            Else
                arr(iRow, iCol) = Cells(iRow, iCol)
            End If
        Next iCol
    Next iRow
    lstAlign.List = arr
End Sub

StandardModule: Modul1

Sub CallForm()
    frmAlign.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zweite ListBox in Abhängigkeit der ersten füllen

Problem: In Abhängigkeit einer Auswahl in der ersten ListBox soll die zweite ListBox gefüllt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmListen.Show
End Sub

ClassModule: frmListen

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub lstA_Click()
    Dim iRow As Integer
    iRow = 2
    lstB.Clear
    Do Until IsEmpty(Cells(iRow, 1))
        lstB.AddItem Cells(iRow, lstA.ListIndex + 1)
        iRow = iRow + 1
    Loop
End Sub

Private Sub UserForm_Initialize()
    lstA.Column = Range("A1").CurrentRegion.Rows(1).Value
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausgewählen Zeilen aus UserForm-ListBox in neue Tabelle

Problem: In einer UserForm-ListBox sollen die Spalten C:D gelistet werden. Die ausgewählten Zeilen sollen komplett in eine zweite Arbeitsmappe übertragen werden. Eine Mehrfachauswahl soll möglich sein.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmAuswahl.Show
End Sub

ClassModule: frmAuswahl

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdEintragen_Click()
    Dim wks As Worksheet
    Dim rng As Range
    Dim iCounter As Integer
    Set wks = ActiveSheet
    Workbooks.Add
    Set rng = ActiveSheet.Range("A1")
    For iCounter = 0 To lstAuswahl.ListCount - 1
        If lstAuswahl.Selected(iCounter) Then
            wks.Rows(iCounter + 1).Copy rng
            Set rng = rng.Offset(1, 0)
        End If
    Next iCounter
End Sub

Private Sub UserForm_Initialize()
    lstAuswahl.List = Range("A1") _
        .CurrentRegion.Columns("C:D").Value
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrspaltige UserForm-ListBox sortieren

Problem: wei Spalten einer ListBox sollen in der Reihenfolge Spalte1, Spalte2 sortiert werden.

Lösung: Den nachfolgenden Code in die benannten Module einfügen.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmSort.Show
End Sub

ClassModule: frmSort

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdSort_Click()
    Application.ScreenUpdating = False
    Workbooks.Add
    Range(Cells(1, 1), Cells _
        (lstSort.ListCount, 2)).Value = lstSort.List
    Range("A1").Sort key1:=Range("A1"), _
        order1:=xlAscending, key2:=Range("B2"), _
        order2:=xlAscending
    lstSort.List = Range("A1").CurrentRegion.Value
    ActiveWorkbook.Close savechanges:=False
End Sub

Private Sub UserForm_Initialize()
    lstSort.List = Range("A1").CurrentRegion.Value
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelte Einträge in UserForm-ListBox löschen

Problem:	Wie kann ich doppelte Einträge in einer UserForm-ListBox löschen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmDoppelte Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdDoppelte_Click() Dim arr() As String Dim var As Variant Dim iCounter As Integer, iItem As Integer ReDim Preserve arr(0) arr(0) = lstDoppelte.List(0) For iCounter = 1 To 11 var = Application.Match(lstDoppelte.List(iCounter), arr, 0) If IsError(var) Then iItem = iItem + 1 ReDim Preserve arr(iItem) arr(iItem) = lstDoppelte.List(iCounter) End If Next iCounter lstDoppelte.List = arr End Sub Private Sub UserForm_Initialize() Dim iCounter As Integer For iCounter = 1 To 12 lstDoppelte.AddItem Format(Date + iCounter, "dddd") Next iCounter End Sub StandardModule: basMain Sub CallForm() frmDoppelte.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Elemente in einer UserForm-ListBox bei Drücken der Entf-Taste lös

Problem: Wie kann ich Elemente in einer UserForm-ListBox bei Drücken der Entf-Taste löschen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDelete

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub lstMonate_KeyDown( _
    ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
    Dim iCounter As Integer
    If KeyCode = 46 Then
        For iCounter = lstMonate.ListCount - 1 To 1 Step -1
            If lstMonate.Selected(iCounter) Then
                lstMonate.RemoveItem iCounter
            End If
        Next
    End If
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 12
        lstMonate.AddItem Format( _
            DateSerial(0, iCounter, 1), "mmm")
    Next iCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmDelete.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-ListBox-Einträge sortieren

Problem:	Wie kann ich Einträge in einer UserForm-ListBox aufsteigend sortieren?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	

```
ClassModule: frmSortieren

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdSortieren_Click()
    Dim iLast As Integer, iNext As Integer
    Dim iTmp As Integer
    With lstSortieren
        For iLast = 0 To .ListCount - 1
            For iNext = iLast + 1 To .ListCount - 1
                If .List(iLast) > .List(iNext) Then
                    iTmp = .List(iLast)
                    .List(iLast) = .List(iNext)
                    .List(iNext) = iTmp
                End If
            Next iNext
        Next iLast
    End With
End Sub

Private Sub UserForm_Initialize()
    With lstSortieren
        .AddItem 1
        .AddItem 3
        .AddItem 9
        .AddItem 2
        .AddItem 5
        .AddItem 7
        .AddItem 2
    End With
End Sub

StandardModule: basMain

Sub CallForm()
    frmSortieren.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Im Abstand von 2 Sekunden durch UserForm-ListBox scrollen

Problem: Wie kann ich mich ohne eigenes Zutun im Zeitabstand von 2 Sekunden zum nächsten UserForm-ListBox-Eintrag scrollen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmZeitgesteuert.Show
End Sub

ClassModule: frmZeitgesteuert

Private Sub UserForm_Activate()
    Dim rng As Range
    Dim iRow As Integer
    For iRow = 1 To 100 Step 10
        Set rng = Range(Cells(iRow, 1), Cells(iRow + 9, 1))
        lblBereich.Caption = "Bereich " & _
            rng.Address(False, False)
        lstWerte.List = rng.Value
        Application.Wait Now + TimeSerial(0, 0, 2)
    Next iRow
    Unload Me
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Element in ListBox in Abhängigkeit einer TextBoxeingabe auswählen

Problem: Wie kann ich bei Eingabe eines Buchstabens in eine TextBox in einer ListBox das erste Element selektiert bekommen, dass diesem Zeichen entspricht? Bei Eingabe weiterer Zeichen soll eine weitere Suche und evtl. Selektion erfolgen.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSuchen

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub txtSuchen_Change()
    Dim iCounter As Integer
    For iCounter = 0 To lstSuchen.ListCount - 1
        If UCase(Left(lstSuchen.List(iCounter), _
            Len(txtSuchen.Text))) = UCase(txtSuchen.Text) Then
            lstSuchen.Selected(iCounter) = True
            Exit Sub
        End If
    Next iCounter
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 12
        lstSuchen.AddItem Format( _
            DateSerial(1, iCounter, 1), "mmm")
    Next iCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmSuchen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrspaltige UserForm-ListBox füllen und aktuellen Wert einlesen

Problem:	Wie kann ich eine Tabelle in eine mehrspaltige UserForm-ListBox einlesen? Nach Auswahl eines Datensatzes in der ListBox soll der aktuelle Wert aus Spalte 1 in eine Tabellenblattzelle eingelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() frmListe.Show End Sub ClassModule: frmListe Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub lstMulti_Click() Range("D1") = lstMulti.Value End Sub Private Sub UserForm_Initialize() Dim iRowL As Integer iRowL = Cells(Rows.Count, 1).End(xlUp).Row lstMulti.List = Range(_ Cells(2, 1), Cells(iRowL, 3)).Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-ListBox in Abhängigkeit zur aufrufenden Spalte füllen

Problem: Wie kann ich eine UserForm-ListBox in Abhängigkeit der als Spaltenkopf angelegten Schaltfläche mit Werten füllen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmVersetzt

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdEintragen_Click()
    Dim iCounter As Integer, iRow As Integer
    iRow = Cells(Rows.Count, CInt(Me.Tag)).End(xlUp).Row
    For iCounter = 0 To lstWerte.ListCount - 1
        If lstWerte.Selected(iCounter) Then
            iRow = iRow + 1
            Cells(iRow, CInt(Me.Tag)).Value = lstWerte.List(iCounter)
        End If
    Next iCounter
End Sub

Private Sub UserForm_Activate()
    lstWerte.List = Range(Cells(3, CInt(Me.Tag)), _
        Cells(17, CInt(Me.Tag))).Value
End Sub

StandardModule: modMain

Sub CallForm()
    With frmVersetzt
        .Tag = Left(ActiveSheet.Buttons(Application.Caller).Caption, 1)
        .Show
    End With
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zufallszahl in UserForm-ListBox zentriert anzeigen

Problem: Wie kann ich eine nach dem Zufallsprinzip ermittelte Zahl in einer UserForm-ListBox immer zentriert anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmZufall

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub cmdZufall_Click()
    Dim iCount As Integer, iAct As Integer
    iAct = lstZufall.ListIndex
    Randomize
    iCount = Int((100 * Rnd) + 1)
    iCount = iCount - 1
    If iCount > 2 And iCount < 98 Then
        If iAct > iCount Then
            lstZufall.ListIndex = iCount - 2
            lstZufall.ListIndex = iCount
        Else
            lstZufall.ListIndex = iCount + 2
            lstZufall.ListIndex = iCount
        End If
    Else
        lstZufall.ListIndex = iCount
    End If
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 100
        lstZufall.AddItem iCounter
    Next iCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmZufall.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur die sichtbaren Zellen eines AutoFilter-Bereiches einlesen

Problem: Wie kann ich nur die sichtbaren Zellen eines mit AutoFilter belegten Bereiches in die ListBox einer UserForm einlesen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmVisibleCells

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim rng As Range
    Application.ScreenUpdating = False
    Set rng = Range("A1").CurrentRegion.SpecialCells(xlCellTypeVisible)
    Workbooks.Add
    rng.Copy Range("A1")
    Rows(1).Delete
    If IsEmpty(Range("A1")) Then
        Beep
        MsgBox "Es entspricht kein Datensatz dem Suchkriterium!"
        ActiveWorkbook.Close savechanges:=False
        End
    ElseIf Range("A1").CurrentRegion.Rows.Count = 1 Then
        lstVisibleCells.AddItem Range("A1").Value
    Else
        lstVisibleCells.List = Range("A1").CurrentRegion.Value
    End If
    ActiveWorkbook.Close savechanges:=False
    Application.ScreenUpdating = True
End Sub

StandardModule: basMain

Sub CallForm()
    frmVisibleCells.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattauswahl über UserForm-ListBox mit Mehrfachauswahl

Problem: Wie kann ich über eine UserForm-ListBox ausgewählte Tabellenblattregister auswählen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSheets

Private Sub cmdWeiter_Click()
    Dim iCounter As Integer
    Dim arr() As String
    Dim iItems As Integer
    For iCounter = 0 To lstSheets.ListCount - 1
        If lstSheets.Selected(iCounter) Then
            iItems = iItems + 1
            ReDim Preserve arr(1 To iItems)
            arr(iItems) = lstSheets.List(iCounter)
        End If
    Next iCounter
    If iItems > 0 Then
        Worksheets(arr).Select
    End If
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim wks As Worksheet
    For Each wks In ThisWorkbook.Worksheets
        lstSheets.AddItem wks.Name
    Next wks
End Sub

StandardModule: basMain

Sub CallForm()
    frmSheets.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus UserForm-ListBox in Tabellenblatt eintragen

Problem: Wie kann ich einen in einer UserForm-ListBox selektierten Wert nach Drücken einer Schaltfläche in ein Tabellenblatt eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAuswahl

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdEintragen_Click()
    Dim iRow As Integer
    If IsEmpty(Cells(1, 1)) Then
        iRow = 1
    Else
        iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
    End If
    Cells(iRow, 1) = lstAuswahl.Value
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 12
        lstAuswahl.AddItem Format( _
            DateSerial(1, iCounter, 1), "mmm")
    Next iCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmAuswahl.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus UserForm-ListBox in die Zwischenablage übernehmen

Problem: Wie kann ich aus einer mehrspaltigen ListBox alle markierten Werte der ersten Spalte in die Zwischenablage übernehmen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmAuslesen.Show
End Sub

ClassModule: frmAuslesen

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdAuslesen_Click()
    Dim iCounter As Integer
    Dim oClip As DataObject
    Dim sTxt As String
    Set oClip = New DataObject
    For iCounter = 1 To lstAuslesen.ListCount - 1
        If lstAuslesen.Selected(iCounter) Then
            sTxt = sTxt & lstAuslesen.List(iCounter, 0) & vbLf
        End If
    Next iCounter
    sTxt = Left(sTxt, Len(sTxt) - 1)
    oClip.SetText sTxt
    oClip.PutInClipboard
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim arr(1 To 20, 1 To 2) As Variant
    Dim iRow As Integer
    For iRow = 1 To 20
        arr(iRow, 1) = iRow
        arr(iRow, 2) = "Element " & iRow
    Next iRow
    lstAuslesen.List = arr
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus mehrspaltiger UserForm-ListBox Werte der 1. Spalte auslesen

Problem: Wie kann ich aus einer mehrspaltigen ListBox alle markierten Werte der ersten Spalte in ein Tabellenblatt auslesen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmAuslesen.Show
End Sub

ClassModule: frmAuslesen

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdAuslesen_Click()
    Dim iCounter As Integer, iRow As Integer
    Columns(1).ClearContents
    For iCounter = 1 To lstAuslesen.ListCount - 1
        If lstAuslesen.Selected(iCounter) Then
            iRow = iRow + 1
            Cells(iRow, 1) = lstAuslesen.List(iCounter, 0)
        End If
    Next iCounter
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim arr(1 To 20, 1 To 2) As Variant
    Dim iRow As Integer
    For iRow = 1 To 20
        arr(iRow, 1) = iRow
        arr(iRow, 2) = "Element " & iRow
    Next iRow
    lstAuslesen.List = arr
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrspaltige UserForm-ListBox lückenlos füllen

Problem: Wie kann ich eine mehrspaltige UserForm-ListBox aus einer Tabelle so mit Werten füllen, dass Leerzeilen nicht berücksichtigt werden?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmMultiColumn.Show
End Sub

ClassModule: frmMultiColumn

Private Sub cmdInsert_Click()
    Dim arr() As Variant
    Dim iRowL As Integer, iRow As Integer, iCol As Integer, iRowU As Integer
    lstMultiCol.Clear
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row
    For iRow = 1 To iRowL
        If Not IsEmpty(Cells(iRow, 1)) Then
            ReDim Preserve arr(0 To 2, 0 To iRowU)
            arr(0, iRowU) = Cells(iRow, 1)
            arr(1, iRowU) = Cells(iRow, 2)
            arr(2, iRowU) = Cells(iRow, 3)
            iRowU = iRowU + 1
        End If
    Next iRow
    lstMultiCol.Column = arr
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalten ein- / ausblenden nach UserForm-ListBox-Vorgabe

Problem: Wie kann ich Spalten verschiedener Tabellenblätter je nach Vorgabe in ListBoxes eines UserForm-MultiPage-Steuerelements ein- und ausblenden?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmTabellen.Show
End Sub

ClassModule: frmTabellen

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub lst1_Change()
    Call EinAus(MultiPage1.Value + 1)
End Sub

Private Sub lst2_Change()
    Call EinAus(MultiPage1.Value + 1)
End Sub

Private Sub MultiPage1_Change()
    Worksheets(MultiPage1.Value + 2).Select
End Sub

Private Sub UserForm_Initialize()
    lst1.Column = Worksheets("Tabelle1").Range("A1").CurrentRegion.Value
    lst2.Column = Worksheets("Tabelle2").Range("A1").CurrentRegion.Value
    Worksheets("Tabelle1").Range("A1").CurrentRegion.Columns.Hidden = True
    Worksheets("Tabelle2").Range("A1").CurrentRegion.Columns.Hidden = True
    Worksheets("Tabelle1").Select
    MultiPage1.Value = 0
End Sub

Private Sub EinAus(iSheet As Integer)
    Dim iCounter As Integer
    With Controls("lst" & iSheet)
        For iCounter = 0 To .ListCount - 1
            Columns(iCounter + 1).Hidden = _
                Not .Selected(iCounter)
        Next iCounter
        Cells(1, 1).Activate
    End With
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einlesen von Spaltenköpfen in UserForm-ListBox

Problem: Wie kann ich Spaltenköpfe in eine UserForm-ListBox einlesen und diese dort editieren?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmHeader

Private Sub cmdEdit_Click()
    Cells(1, lstHeader.ListIndex + 1).Value = txtSelection.Text
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub lstHeader_Click()
    txtSelection.Text = lstHeader.Value
End Sub

Private Sub UserForm_Initialize()
    lstHeader.Column = Range("A1").CurrentRegion.Value
End Sub

StandardModule: basMain

Sub CallForm()
    frmHeader.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattnamen bedingt in ListBox einlesen und Blätter auswählen

Problem: Wie kann ich beim Öffnen einer Arbeitsmappe eine ListBox mit den vorhandenen Blattnamen mit Ausnahme der Blätter "Test" und "Muster" füllen. Bei Auswahl eines Blattes soll dieses auf Schaltflächendruck ausgewählt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
StandardModule: basMain

Sub cmdAusaehlen_Click()
    Worksheets(Tabelle2.lstWks.Value).Select
End Sub

Sub cmdKopieren_Click()
    Dim sName As String
    sName = InputBox("Blattname:", , "Muster1")
    If sName = "" Then Exit Sub
    Worksheets("Muster").Copy after:=Worksheets(Worksheets.Count)
    ActiveSheet.Name = sName
    Tabelle2.lstWks.AddItem sName
End Sub

ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Dim wks As Worksheet
    Tabelle2.lstWks.Clear
    For Each wks In Worksheets
        If wks.Name <> "Test" And wks.Name <> "Muster" Then
            Tabelle2.lstWks.AddItem wks.Name
        End If
    Next wks
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen für Eintrag auswählen und UserForm-ListBox aufrufen

Problem: Ich habe in den Spalte A:E Einträge zu machen. Der Cursor soll nach jedem Eintrag eine Zelle nach rechts springen. Ist er in Spalte E angelangt, soll sich eine UserForm mit einer ListBox zur Auswahl öffnen.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    If Target.Column < 4 Then
        Target.Offset(0, 1).Select
    ElseIf Target.Column = 4 Then
        Target.Offset(0, 1).Select
        frmAuswahl.Show
    ElseIf Target.Column = 5 Then
        Target.Offset(1, -4).Select
    End If
End Sub

ClassModule: frmAuswahl

Private Sub ListBox1_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    ActiveCell = ListBox1.Value
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    With frmAuswahl.ListBox1
        .AddItem "Freund"
        .AddItem "Feind"
        .AddItem "Dienst"
        .AddItem "Urlaub"
        .ListIndex = 0
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ListBox-Auswahl in Labels übernehmen

Problem:	Wie kann ich die Auswahl aus einer UserForm-ListBox in Labels der UserForm übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmWertAusListbox Private Sub CommandButton1_Click() Unload Me End Sub Private Sub ListBox1_Change() Label1 = ListBox1.Value Label2 = Cells(ListBox1.ListIndex + 2, 2) End Sub StandardModule: basMain Sub CallForm() frmWertAusListbox.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Waren in Warenkorb legen und in Liste eintragen

Problem: Wie kann ich Waren aus einer UserForm-ListBox in eine WarenkorbListBox übertragen und in eine Einkaufsliste eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmVerkauf.Show
End Sub

ClassModule: frmVerkauf

Private Sub cmdEintragen_Click()
    Dim iRow As Integer, iRowL As Integer, iCol As Integer
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row + 1
    For iRow = 0 To lstKorb.ListCount - 1
        For iCol = 0 To 2
            Cells(iRowL + iRow, iCol + 1) = lstKorb.List(iRow, iCol)
        Next iCol
    Next iRow
    lstKorb.Clear
    Columns.AutoFit
End Sub

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub lstWaren_Click()
    Dim iCounter As Integer
    lstKorb.AddItem
    For iCounter = 0 To 2
        If iCounter < 2 Then
            lstKorb.List(lstKorb.ListCount - 1, iCounter) = _
                lstWaren.List(lstWaren.ListIndex, iCounter)
        Else
            lstKorb.List(lstKorb.ListCount - 1, iCounter) = _
                Format(lstWaren.List(lstWaren.ListIndex, iCounter), "0.00")
        End If
    Next iCounter
End Sub

Private Sub lstKorb_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    lstKorb.RemoveItem lstKorb.ListIndex
End Sub

Private Sub UserForm_Initialize()
    Dim wks As Worksheet
    Dim iRow As Integer
    Set wks = Worksheets("Waren")
    iRow = wks.Cells(Rows.Count, 1).End(xlUp).Row
    With lstWaren
        .RowSource = wks.Name & "!A2:C" & iRow
        .ColumnWidths = "40;80;60"
    End With
    lstKorb.ColumnWidths = "40;80;60"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wahlweise Elemente in UserForm-Text- oder -ListBox markieren

Problem: Wie kann ich in einer UserForm nach Betätigung von Schaltflächen wahlweise den Text in einer TextBox oder ein Element in einer ListBox markieren?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMarkieren

Private Sub cmdListMarkieren_Click()
    Dim iCounter As Integer
    lstMarkieren.SetFocus
    For iCounter = 0 To lstMarkieren.ListCount - 1
        If lstMarkieren.List(iCounter) = txtMarkieren.Text Then
            lstMarkieren.ListIndex = iCounter
            Exit Sub
        End If
    lstMarkieren.ListIndex = -1
    Next iCounter
End Sub

Private Sub cmdTextMarkieren_Click()
    With txtMarkieren
        .SetFocus
        .SelStart = 0
        .SelLength = .TextLength
    End With
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iMonth As Integer
    For iMonth = 1 To 12
        lstMarkieren.AddItem Format( _
            DateSerial(Year(Date), iMonth, 1), "mmm")
    Next iMonth
End Sub

StandardModule: basMain

Sub CallForm()
    frmMarkieren.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gruppe von UserForm-ListBoxes über eine Schleife ansprechen

Problem: Wie kann ich eine Gruppe von UserForm-ListBoxes in einer Schleife ansprechen und beim Aufruf der UserForm mit Werten füllen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmListCollection

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim col As New Collection
    Dim iList As Integer, iItem As Integer
    col.Add lstEins
    col.Add lstZwei
    col.Add lstDrei
    col.Add lstVier
    For iList = 1 To 4
        For iItem = 1 To 10
            col(iList).AddItem "Liste " & _
                iList & " - Zeile " & iItem
        Next iItem
        col(iList).ListIndex = 0
    Next iList
End Sub

StandardModule: basMain

Sub CallForm()
    frmListCollection.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachauswahl aus UserForm-ListBoxes in Tabelle eintragen

Problem: Wie kann ich eine Mehrfachauswahl aus mehreren Listboxes in eine Tabelle eintragen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code: ClassModule: frmListBoxes

```
Private Sub cmdEintragen_Click()  
    Dim cnt As Control  
    Dim iRow As Integer, iCol As Integer, iItem As Integer  
    iCol = 4  
    For Each cnt In Controls  
        iRow = 1  
        iCol = iCol + 1  
        If TypeName(cnt) = "ListBox" Then  
            For iItem = 0 To cnt.ListCount - 1  
                If cnt.Selected(iItem) Then  
                    Cells(iRow, iCol) = cnt.List(iItem)  
                    iRow = iRow + 1  
                End If  
            Next iItem  
        End If  
    Next cnt  
End Sub  
  
Private Sub cmdWeiter_Click()  
    Unload Me  
End Sub  
  
StandardModule: basMain  
  
Sub CallForm()  
    frmListBoxes.Show  
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausgewählte Werte aus UserForm-ListBoxes addieren

Problem: Wie kann ich in einer Serie von UserForm-ListBoxes ausgewählte Werte addieren?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmListBoxes

Private Sub cmdBerechnen_Click()
    Dim dSum As Double
    Dim iCounter As Integer
    For iCounter = 1 To 4
        dSum = dSum + CDBl(Controls("lstWert" & iCounter).List( _
            Controls("lstWert" & iCounter).ListIndex))
    Next iCounter
    MsgBox dSum
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim dValue As Double
    Dim iCounter As Integer, intWerte As Integer
    For iCounter = 1 To 4
        For intWerte = 1 To 10
            dValue = dValue + 1.2
            Controls("lstWert" & iCounter).AddItem dValue
        Next intWerte
        dValue = 0
        Controls("lstWert" & iCounter).ListIndex = iCounter
    Next iCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmListBoxes.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalte A mehrer Mappen in UserForm und aktives Blatt

Problem: Wie kann ich aus allen Arbeitsmappen eines Verzeichnisses den jeweiligen Inhalt von Spalte A in eine UserForm-ListBox einlesen und danach in das aktive Blatt eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDaten

Private Sub cmdWeiter_Click()
    Dim iRow As Integer
    For iRow = 0 To lstDaten.ListCount - 1
        Cells(iRow + 1, 1).Value = lstDaten.List(iRow)
    Next iRow
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    Dim iCounter As Integer, iRow As Integer
    Application.ScreenUpdating = False
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    With Application.FileSearch
        .LookIn = Range("C1").Value
        .FileType = msoFileTypeExcelWorkbooks
        .Execute
        For iCounter = 1 To .FoundFiles.Count
            Workbooks.Open .FoundFiles(iCounter), False
            With Range("A1").CurrentRegion
                For iRow = 1 To WorksheetFunction.CountA(.Columns(1))
                    frmDaten.lstDaten.AddItem Cells(iRow, 1).Value
                Next iRow
            End With
            ActiveWorkbook.Close savechanges:=False
        Next iCounter
    End With
    frmDaten.Show
ERRORHANDLER:
    Application.EnableEvents = True
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bereichsnamen in UserForm-ListBox auflisten und editieren

Problem: Wie kann ich alle Bereichsnamen einer Arbeitsmappe in einer UserForm-ListBox auflisten und ändern?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmNamen

Private Sub cmdEditieren_Click()
    Dim sTxt As String
    If txtOrtAktuell.Text = "" Then
        Beep
        MsgBox "Sie müssen zuerst eine Zeile mit Doppelklick auswählen!"
        Exit Sub
    End If
    With txtOrtAktuell
        sTxt = Right(.Text, Len(.Text) - InStr(.Text, "!"))
        sTxt = Range(sTxt).Address
        .Text = Left(.Text, InStr(.Text, "!")) & sTxt
    End With
    With lstNamen
        ActiveWorkbook.Names(.ListIndex + 1).RefersTo = txtOrtAktuell.Text
        ActiveWorkbook.Names(.ListIndex + 1).Name = txtNameAktuell.Text
        .List(.ListIndex, 0) = txtOrtAktuell
        .List(.ListIndex, 1) = txtNameAktuell
    End With
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub lstNamen_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    With lstNamen
        txtOrtAktuell.Text = .List(.ListIndex, 0)
        txtNameAktuell.Text = .List(.ListIndex, 1)
    End With
End Sub

Private Sub UserForm_Initialize()
    Dim nme As Name
    Dim arr()
    Dim iRow As Integer
    ReDim arr(0 To 1, 0)
    For Each nme In ActiveWorkbook.Names
        ReDim Preserve arr(0 To 1, iRow)
        arr(0, iRow) = nme
        arr(1, iRow) = nme.Name
        iRow = iRow + 1
    Next nme
    lstNamen.Column = arr
End Sub

StandardModule: basMain

Sub CallForm()
    frmNamen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Matrix in UserForm-ListBox einlesen

Problem: Wie kann ich eine Matrix in eine mehrspaltige UserForm-ListBox einlesen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdDialogAufruf_Click()
    frmListBox.Show
End Sub

ClassModule: frmListBox

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim arr(1 To 2, 1 To 10)
    Dim iRow As Integer, iCol As Integer
    For iCol = 1 To 2
        For iRow = 1 To 10
            arr(iCol, iRow) = _
                "Zeile " & iRow & " - Spalte " & iCol
        Next iRow
    Next iCol
    lstArray.Column = arr
End Sub

StandardModule: basMain

Sub CallForm()
    frmListBox.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auswahl aus Tabellenblatt-ListBox in Zellen übernehmen

Problem:	Wie kann ich die jeweilige Auswahl in einer Tabellenblatt-ListBox im Moment der Auswahl in die Zellen eines Arbeitsblattes übernehmen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub ListBox1_Click() Dim iCounter As Integer, iRow As Integer iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 For iCounter = 1 To 3 Cells(iRow, 0 + iCounter) = _ Tabelle2.Cells(ListBox1.ListIndex + 2, iCounter) Next iCounter End Sub</pre>

	Herbers Excel-FAQ - Die Texte Bestandteil der Excel-Lehr- und Lernmaterialien
Verzeichnisbaum aufrufen und Dateien in ListBox auflisten	
Problem:	Wie kann ich mir alle Laufwerke mit ihren Verzeichnissen anzeigen und die Exceldateien eines ausgewählten Verzeichnisses in der Listbox einer UserForm auflisten lassen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmDateiListe Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim iCounter As Integer Dim sFolder As String sFolder = GetDirectory If sFolder = "" Then End With Application.FileSearch .LookIn = sFolder .Execute For iCounter = 1 To .FoundFiles.Count lstFiles.AddItem .FoundFiles(iCounter) Next iCounter End With End Sub StandardModule: basFunctions Public Type BROWSEINFO hOwner As Long pidlRoot As Long pszDisplayName As String lpszTitle As String ulFlags As Long lpfn As Long lParam As Long iImage As Long End Type Declare Function SHGetPathFromIDList Lib "shell32.dll" _ Alias "SHGetPathFromIDListA" (ByVal pidl As Long, ByVal pszPath As String) As Long Declare Function SHBrowseForFolder Lib "shell32.dll" _ Alias "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long Function GetDirectory(Optional Msg As String) As String Dim bInfo As BROWSEINFO Dim Path As String Dim r As Long, x As Long, pos As Integer bInfo.pidlRoot = 0& If IsMissing(Msg) Then bInfo.lpszTitle = "Wählen Sie bitte einen Ordner aus." Else bInfo.lpszTitle = Msg End If bInfo.ulFlags = &H1 x = SHBrowseForFolder(bInfo) Path = Space\$(512) r = SHGetPathFromIDList(ByVal x, ByVal Path) If r Then pos = InStr(Path, Chr\$(0)) GetDirectory = Left(Path, pos - 1) Else GetDirectory = "" End If End Function StandardModule: basMain Sub CallForm() frmDateiListe.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen einer Arbeitsmappe einlesen

Problem: Wie kann ich die Bereichsnamen einer Arbeitsmappe in einer ListBox auflisten? Neben den Namen sollen die Bereichsadressen angezeigt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub DialogAufruf()
    frmNames.Show
End Sub

ClassModule: frmNames

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_Initialize()
    Dim nme As Name
    For Each nme In Names
        lstNames.AddItem nme.Name & " " & nme.RefersTo
    Next nme
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachauswahl in ListBox auslesen

Problem: Wie kann ich bei einer Mehrfachauswahl in einer ListBox die nicht ausgewählten Elemente auslesen und in einer MsgBox anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmHNoSelect.Show
End Sub

ClassModule: frmNoSelect

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim iRow As Integer
    Dim sDays As String
    For iRow = 1 To 7
        If lstWeekdays.Selected(iRow - 1) = False Then
            sDays = sDays & lstWeekdays.List(iRow - 1) & vbCrLf
        End If
    Next iRow
    MsgBox "Nicht ausgewählt wurden: " & vbCrLf & sDays
End Sub

Private Sub UserForm_Initialize()
    Dim iRow As Integer
    For iRow = 1 To 7
        lstWeekdays.AddItem Format(DateSerial(1, 1, iRow), "dddd")
    Next iRow
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateinamen in Listbox

Problem: Wie kann ich die Dateien des aktuellen Verzeichnisses in einer ListBox einer UserForm auflisten und sortieren lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmFiles

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    With Application.FileSearch
        .LookIn = CurDir
        .SearchSubFolders = False
        .Execute msoSortByFileName
        For iCounter = 1 To .FoundFiles.Count
            lstfiles.AddItem .FoundFiles(iCounter)
        Next iCounter
    End With
End Sub

StandardModule: basMain

Sub CallForm()
    frmFiles.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Als Zellkontextmenü Listbox aufrufen

Problem:	Wie kann ich das Zellkontextmenü durch eine UserForm mit einer Listbox ersetzen, aus der der ausgewählte Begriff in die aktive Zelle eingetragen wird?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle4 Private Sub Worksheet_BeforeRightClick(_ ByVal Target As Excel.Range, _ Cancel As Boolean) Cancel = True frmKontext.Show End Sub ClassModule: frmKontext Private Sub lstKontext_Change() ActiveCell = lstKontext.Value Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachauswahl aus ListBox auslesen

Problem: Wie kann ich aus einer UserForm-Listbox die ausgewählten Elemente einer Mehrfachauswahl bestimmen, die Selektierung aufheben und die Elemente löschen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAuslesen

Private Sub cmdAuslesen_Click()
    Dim iCounter As Integer
    Dim sMsg As String
    For iCounter = lstValues.ListCount - 1 To 0 Step -1
        If lstValues.Selected(iCounter) Then
            sMsg = sMsg & lstValues.List(iCounter) & vbCrLf
            lstValues.RemoveItem iCounter
        End If
    Next iCounter
    MsgBox sMsg
End Sub

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 12
        lstValues.AddItem Format(DateSerial(1, iCounter, 1), "mmm")
    Next iCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmAuslesen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattnamen in eine ListBox einlesen

Problem:	Wie kann ich die Tabellenblätter einer geschlossenen Arbeitsmappe in eine ListBox einer UserForm einlesen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmTabellen Sub cmdList_Click() Dim wks As Worksheet Dim vFile As Variant Application.ScreenUpdating = False Application.EnableEvents = False On Error GoTo ERRORHANDLER vFile = Application.GetOpenFilename("Excel-Arbeitsmappen (*.xls), *.xls") If vFile = False Then Exit Sub Workbooks.Open vFile For Each wks In ActiveWorkbook.Worksheets lstWks.AddItem wks.Name Next wks ActiveWorkbook.Close savechanges:=False ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = True End Sub Private Sub cmdContinue_Click() Unload Me End Sub StandardModule: basMain Sub CallForm() frmTabellen.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Horizontal angeordnete Elemente in eine ListBox übernehmen

Problem:	Wie kann ich horizonatal (in einer Zeile) angeordnete Elemente in eine Listbox übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() frmHorizontal.Show End Sub ClassModule: frmHorizontal Private Sub cmdContinue_Click() Unload Me End Sub Private Sub UserForm_Initialize() lstValues.Column = Range("A1").CurrentRegion.Rows(1).Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ListBox-Einträge auswechseln

Problem: Wie kann ich in XL8 über die AddItem-Methode eingefügte Einträge in einer ListBox gegen neue Begriffe austauschen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmChange.Show
End Sub

ClassModule: frmChange

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub lstDays_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    Dim iIdx As Integer
    Dim sNew As String
    sNew = InputBox("Neuen Eintrag eingeben:")
    If sNew = "" Then Exit Sub
    With lstDays
        iIdx = .ListIndex
        .RemoveItem .ListIndex
        .AddItem sNew, iIdx
    End With
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 7
        lstDays.AddItem Format(Weekday(iCounter), "dddd")
    Next iCounter
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Listbox-Eintrag durch Mausbewegung markieren

Problem: Wie kann ich bei XL8 in einer UserForm durch das Bewegen des Mauszeigers die Auswahl in einer ListBox verändern, ohne dass ein Mausklick erfolgt?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmMouseMove.Show
End Sub

ClassModule: frmMouseMove

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub lstMouse_MouseMove( _
    ByVal Button As Integer, _
    ByVal Shift As Integer, _
    ByVal X As Single, _
    ByVal Y As Single)
    If Int(Y / 4) < lstMouse.ListCount Then
        lstMouse.ListIndex = Int(Y / 4)
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateien aus Listbox auslesen und öffnen

Problem: Wie kann man Dateinamen aus einer UserForm-MultiSelect-Listbox auslesen und die zugehörigen Dateien nacheinander öffnen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmFiles.Show
End Sub

ClassModule: frmFiles

Private Sub cmdOpen_Click()
    Dim iCounter As Integer
    For iCounter = 0 To lstFiles.ListCount - 1
        If lstFiles.Selected(iCounter) Then
            Workbooks.Open lstFiles.List(iCounter)
        End If
    Next iCounter
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    With Application.FileSearch
        .LookIn = Range("B1").Value
        .FileType = msoFileTypeExcelWorkbooks
        .Execute
        For iCounter = 1 To .FoundFiles.Count
            lstFiles.AddItem .FoundFiles(iCounter)
        Next iCounter
    End With
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fundstellen in UserForm listen

Problem: Wie kann ich einen Bereich eines Tabellenblattes nach einem Teilbegriff durchsuchen lassen und die Zelladressen der Fundstellen in einem Listenfeld einer UserForm auflisten lassen?

Lösung: Geben Sie den den nachfolgenden Code in die genannten Module ein.

VBA-Code:

```
ClassModule: frmGefunden

Private Sub cmdContinue_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub FundstellenSuchen()
    Dim iRow As Integer, iRowL As Integer
    Dim sSearch As String
    sSearch = InputBox(prompt:="Suchbegriff:")
    If sSearch = "" Then Exit Sub
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row
    For iRow = 1 To iRowL
        If InStr(Cells(iRow, 1).vlaue, sSearch) Then
            frmGefunden.lstFound.AddItem Cells(iRow, 1).Value
        End If
    Next iRow
    frmGefunden.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

XL-Dateien in UserForm listen

Problem:	Wie kann ich durch Betätigung einer Schaltfläche in einer UserForm die XL-Dateinamen des aktuellen Verzeichnisses in einem sich ebenfalls in der UserForm befindlichen Listenfeld listen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basMain Sub Start() frmFileList.Show End Sub ClassModule: frmFileList Private Sub cmdList_Click() Dim intCounter As Integer With Application.FileSearch .LookIn = CurDir .FileName = "*.xls" .Execute For intCounter = 1 To .FoundFiles.Count lstFiles.AddItem .FoundFiles(intCounter) Next End With End Sub Private Sub cmdOK_Click() Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gefilterte Daten in eine ListBox einlesen

Problem:	Wie kann ich erreichen, dass nach jedem Filtervorgang die gefilterten Daten in eine ListBox eingelesen werden?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Calculate() Dim iCounter As Integer lstFilter.Clear For iCounter = 2 To WorksheetFunction.CountA(Columns(1)) If Rows(iCounter).Hidden = False Then lstFilter.AddItem Cells(iCounter, 1).Value End If Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gefilterte Daten in eine ListBox einlesen

Problem: Wie kann ich das Filtern eines Bereiches über VBA vornehmen und nur die sichtbaren Datensätze in eine mehrspaltige UserForm-ListBox einlesen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmFilter

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim sFilter As String
    Application.ScreenUpdating = False
    sFilter = InputBox("Filter:", , "Zeile 1*")
    If sFilter = "" Then Exit Sub
    Columns("A:A").AutoFilter Field:=1, Criteria1:=sFilter
    Range("A1").CurrentRegion.SpecialCells(xlCellTypeVisible).Copy
    Workbooks.Add
    Range("A1").PasteSpecial xlPasteValues
    Rows(1).Delete
    lstDesignAuswahl.List = Range("A1").CurrentRegion.Value
    ActiveWorkbook.Close savechanges:=False
    lstDesignAuswahl.ListIndex = 0
    Columns("A:A").AutoFilter
End Sub

StandardModule: basMain

Sub CallDialog()
    frmFilter.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattnamen in ein Listefeld einlesen

Problem: Wie kann ich die Tabellenblattnamen einer auszuwählenden Arbeitsmappe in ein Listefeld der aktiven Arbeitsmappe einlesen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub cmdWks_Click()  
    Dim wks As Worksheet  
    Dim arr() As String  
    Dim vName As Variant  
    Dim iCounter As Integer  
    lstWks.Clear  
    Application.ScreenUpdating = False  
    vName = Application.GetOpenFilename("Excel-Arbeitsmappen (*.xls), *.xls")  
    If vName = False Then Exit Sub  
    Workbooks.Open vName  
    ReDim arr(1 To ActiveWorkbook.Worksheets.Count)  
    For Each wks In ActiveWorkbook.Worksheets  
        iCounter = iCounter + 1  
        arr(iCounter) = wks.Name  
    Next wks  
    ActiveWorkbook.Close  
    Application.ScreenUpdating = True  
    lstWks.List = arr  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachauswahl aus ListBox auslesen

Problem:	Wie kann ich eine Mehrfachauswahl aus einem Listenfeld in ein Tabellenblatt eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basmain Sub CallDialog() frmMonths.Show End Sub ClassModule: frmMonths Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdEintragen_Click() Dim iRow As Integer, iCounter As Integer For iRow = 0 To lstMonths.ListCount - 1 If lstMonths.Selected(iRow) Then iCounter = iCounter + 1 Cells(iCounter, 1).Value = lstMonths.List(iRow) End If Next iRow Unload Me End Sub Private Sub UserForm_Initialize() Dim iMonth As Integer For iMonth = 1 To 12 lstMonths.AddItem Format(_ DateSerial(1, iMonth, 1), "mmm") Next iMonth End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Farbe bei Mausberührung wechseln

Problem: Wie kann ich die Hintergrundfarbe eines Labels in einer UserForm für den Zeitraum verändern, für den sich die Maus über das Label bewegt?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmColors

Private Sub lblcolor_MouseMove(ByVal Button As Integer, _
    ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
    lblColor.BackColor = RGB(125, 250, 250)
End Sub

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub UserForm_MouseMove(ByVal Button As Integer, _
    ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
    lblColor.BackColor = RGB(250, 125, 250)
End Sub

StandardModule: basMain

Sub CallForm()
    frmColors.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Meldung beim Öffnen und bei Zellauswahl in Spalte A

Problem: Sowohl beim Öffnen der Arbeitsmappe wie auch bei der Auswahl einer Zelle in Spalte A soll eine MsgBox erscheinen.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    MsgBox "Hallo, ich bin die Arbeitsmappe " & Me.Name
End Sub

ClassModule: Tabelle1

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Target.Column = 1 Then
        MsgBox "Hallo, ich bin die Zelle " & Target.Address
    End If
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Msg- und InputBoxes kombinieren und nach Ergebnis verzweigen

Problem: Wie kann ich über eine Kombination von Msg- und InputBoxes in verschiedene Makros in Abhängigkeit von Eingaben verzweigen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Eingeben()  
    Dim var As Variant  
    Dim sTitle As String, sEinheit As String  
    Dim sAuswahl As String  
    sTitle = "Auswahl Geschäftseinheit"  
    sAuswahl = "VA,VG,VE"  
    MsgBox "Achtung ohne Auswahl der Geschäftseinheit" & vbCrLf & _  
        "kein Vertriebszuschlag i.d. Kalkulation!", vbOKOnly + _  
        vbExclamation, sTitle  
    Do Until var = vbOK  
        sEinheit = UCase(InputBox("Bitte geben sie Ihre" & _  
            "Geschäftseinheit an: (" & _  
            sAuswahl & ") ", "Geschäftseinheit"))  
        If InStr(sAuswahl, sEinheit) Then  
            Select Case sEinheit  
                Case "VA", "VG", "VE"  
                    var = MsgBox("Sie haben " & sEinheit & " eingegeben", _  
                        vbOKCancel + vbExclamation)  
                    If var = vbOK Then  
                        Run "Kalk_" & sEinheit  
                    Else  
                        MsgBox "Sie kommen zurück zur GE Auswahl !"  
                    End If  
                End Select  
            End If  
        Loop  
    End Sub  
  
Private Sub Kalk_VA()  
    MsgBox "Kalk_VA gestartet..."  
End Sub  
  
Private Sub Kalk_VG()  
    MsgBox "Kalk_VG gestartet..."  
End Sub  
  
Private Sub Kalk_VE()  
    MsgBox "Kalk_VE gestartet..."  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalte in MultiPage-TextBoxes einlesen

Problem: Die Spalteninhalte aus A:C sollen in MultiPage-TextBoxes eingelesen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMultiPage

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer, iRow As Integer
    Dim sTxt As String
    For iCounter = 1 To 3
        For iRow = 1 To 3
            sTxt = sTxt & Cells(iRow, iCounter).Value & vbLf
        Next iRow
        sTxt = Left(sTxt, Len(sTxt) - 1)
        Controls("TextBox" & iCounter).Text = sTxt
        sTxt = ""
    Next iCounter
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmMultiPage.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In einem MultiPage-Steuerelement blättern

Problem:	Wie kann ich in einem MultiPage-Steuerelement per VBA die blättern?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmMultiPageDemo Private Sub cmdNext_Click() With MultiPage1 If .Value < .Pages.Count - 1 Then .Value = .Value + 1 Else .Value = 0 End If End With End Sub Private Sub cmdContinue_Click() Unload Me End Sub StandardModule: basMain Sub CallForm() frmMultiPageDemo.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Multiplikation in Abhängigkeit einer Optionsfeldauswahl

Problem:	In Abhängigkeit der Optionsfeldauswahl soll die Basiszahl mit dem jeweils ausgewählten Faktor multipliziert werden.
Lösung:	Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte von OptionButtons aus 2 Gruppen auslesen

Problem: Die OptionButtons sind zwei Gruppen zugeordnet. Es soll ermittelt werden, welches Optionsfeld in GoupB aktiviert ist

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Sub IfSelected()
    Dim oOle As OLEObject
    Dim oOpt As msforms.OptionButton
    For Each oOle In OLEObjects
        If TypeName(oOle.Object) = "OptionButton" Then
            Set oOpt = oOle.Object
            If oOpt And oOpt.GroupName = "GroupB" Then
                MsgBox "In GroupB ist " & oOpt.Caption & " aktiviert"
            End If
        End If
    Next oOle
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus Optionsfelder in verschiedenen Frames auslesen

Problem: Die Werte von je 3 Optionsfeldern in 3 Frames einer UserForm sollen ausgelesen und in die jeweils nächste Zeile der Tabelle eingetragen werden.

Lösung: Die Optionsfelder sollen nach dem Eintrag zurückgesetzt werden.

VBA-Code:

```
ClassModule: frmOptFrames

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim iFrm As Integer, iOpt As Integer, iRow As Integer
    iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
    For iFrm = 1 To 3
        For iOpt = 1 To 3
            Cells(iRow, (iFrm - 1) * 3 + iOpt).Value = _
                Controls("Frame" & iFrm) _
                .Controls("OptionButton" & _
                    (iFrm - 1) * 3 + iOpt).Value
            Controls("Frame" & iFrm) _
                .Controls("OptionButton" & _
                    (iFrm - 1) * 3 + iOpt).Value = False
        Next iOpt
    Next iFrm
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmOptFrames.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-Optionsfelder über eine Schleife zurücksetzen

Problem: Die Optionsfelder in einer UserForm sollen über eine Schleife zurückgesetzt werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmReset

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim cnt As Control
    For Each cnt In Controls
        If TypeName(cnt) = "OptionButton" Then
            cnt.Value = False
        End If
    Next cnt
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmReset.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-Optionsfelder prüfen

Problem: Wenn in einer UserForm im ersten Gruppenfeld das 2. und im 2. Das 3. Optionsfeld aktiviert sind, soll ein neues Tabellenblatt hinzugefügt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmOptions.Show
End Sub

ClassModule: frmOptions

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim iA As Integer, iB As Integer
    iA = opt(Frame1, 0)
    iB = opt(Frame2, 4)
    If iA = 2 And iB = 3 Then
        Worksheets.Add.Move after:=Worksheets(Worksheets.Count)
    Else
        Beep
        MsgBox "Nein, ich tue nur was, wenn die " & _
            "Optionsfelder 2 und 3 aktiviert sind!"
    End If
End Sub

Private Function opt(fme As Frame, iAct As Integer) As Integer
    Dim iCounter As Integer
    For iCounter = 1 + iAct To 4 + iAct
        If fme.Controls("OptionButton" & iCounter).Value = True Then
            opt = iCounter - iAct
            Exit Function
        End If
    Next iCounter
End Function
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Synchronisieren von Optionsfeldern in verschiedenen Blättern

Problem:	Die Optionsfelder in allen Tabellenblättern sollen synchronisiert werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gruppe von Optionsfeldern kopieren und Makro zuweisen

Problem: Wie kann ich ein Gruppenfeld mit Optionsfeldern in ein zweites Tabellenblatt kopieren und den Optionsfeldern ein Makro zuweisen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub KopierenUndZuweisen()
    Dim grp As GroupBox
    Dim opt As OptionButton
    With Worksheets("Tabelle2")
        Worksheets("Tabelle1").GroupBoxes(1).Copy
        .Paste
        Set grp = .GroupBoxes(.GroupBoxes.Count)
        grp.Left = Range("F16").Left
        grp.Top = Range("F16").Top
        Worksheets("Tabelle1").OptionButtons(1).Copy
        Worksheets(2).Paste
        Set opt = .OptionButtons(.OptionButtons.Count)
        With opt
            .Left = grp.Left + 10
            .Top = grp.Top + 10
            .OnAction = "Option_Click"
        End With
        Worksheets("Tabelle1").OptionButtons(2).Copy
        Worksheets(2).Paste
        Set opt = .OptionButtons(.OptionButtons.Count)
        With opt
            .Left = grp.Left + 10
            .Top = grp.Top + 25
            .OnAction = "Option_Click"
        End With
    End With
End Sub

Sub option_click()
    MsgBox "Hallo!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makroverzweigungen nach OptionsFeld- oder ListBox-Auswahl

Problem:	Wie kann ich in Abhängigkeit von OptionsFeld- oder ListBox-Click Makros verzweigen und die Werte ausgeben?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub ListBox1_Change() Call Message End Sub Private Sub OptionButton1_Click() Call Message End Sub Private Sub OptionButton2_Click() Call Message End Sub Private Sub OptionButton3_Click() Call Message End Sub Private Sub OptionButton4_Click() Call Message End Sub Private Sub Message() Dim oOle As OLEObject Dim iCounter As Integer For iCounter = 1 To 4 If ActiveSheet.OLEObjects(iCounter) _ .Object.Value = True Then Exit For End If Next iCounter MsgBox ActiveSheet.OLEObjects(iCounter) _ .Object.Caption & " - " & ListBox1.Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelleintrag durch **OptionButton**-Betätigung ein- und ausschalten

Problem:	Wie kann ich einen Zelleintrag durch Betätigung von OptionButtons im Tabellenblatt ein- und ausschalten?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub optSchriftAus_Change() ActiveCell.Select If optSchriftAus.Value = True Then Range("A1").NumberFormat = ";;;" Else Range("A1").NumberFormat = "General" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auslesen des ausgewählten Optionsfeldes in einer UserForm

Problem: Wie kann ich feststellen, welches Optionsfeld in einer UserForm aktiviert wurde?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAuslesen

Private Sub cmdAuslesen_Click()
    Dim cnt As Control
    Dim iCounter As Integer
    For Each cnt In Controls
        If TypeName(cnt) = "OptionButton" Then
            iCounter = iCounter + 1
            If cnt.Value = True Then Exit For
        End If
    Next cnt
    MsgBox "Optionsfeld Nr. " & iCounter & _
        " wurde ausgewählt!"
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmAuslesen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Begrenzung der RefEdit-Auswahl auf eine Zelle

Problem: Mit dem RefEdit-Steuerelement soll immer nur eine Zelle ausgewählt werden können.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmOneOnly

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub RefEdit1_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    If Range(RefEdit1.Value).Cells.Count > 1 Then
        MsgBox "Bitte nur eine Zelle auswählen!"
        RefEdit1.Value = ""
        Cancel = True
    End If
End Sub

StandardModule: Modul1

Sub CallForm()
    frmOneOnly.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über UserForm-RefEdit-Steuerelement Bereich auswählen

Problem: Wie kann ich über ein UserForm-RefEdit-Steuerelement einen Bereich auswählen und für den ausgewählten Bereich eine bedingte Formatierung festlegen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmFormatierung

Private Sub cmdFormat_Click()
    With Range(refFormat.Value)
        .FormatConditions.Delete
        .Range(refFormat.Value).Cells(1).Activate
        .FormatConditions.Add Type:=xlExpression, Formula1:= _
            "=WOCHENTAG(" & Range(refFormat.Value).Cells(1, 1) _
            .Address(False, False) & ")=7"
        .FormatConditions(1).Interior.ColorIndex = 35
        .FormatConditions.Add Type:=xlExpression, Formula1:= _
            "=WOCHENTAG(" & Range(refFormat.Value).Cells(1, 1) _
            .Address(False, False) & ")=1"
        .FormatConditions(2).Interior.ColorIndex = 42
    End With
    Unload Me
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmFormatierung.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In Abhängigkeit zur RefEdit-Auswahl UserForm aufrufen

Problem:	Wie kann ich über ein UserForm-RefEdit-Steuerelement einen Typ auswählen, dann in Abhängigkeit vom Typ eine UserForm aufrufen und die Daten eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Main() frmAuswahl.Show End Sub ClassModule: Tabelle1 Private Sub CommandButton1_Click() Call Main End Sub ClassModule: frmA Private Sub cmdEintragen_Click() With Worksheets("Ausgabe") .Range("B1") = Label4.Caption .Range("B2") = Label5.Caption .Range("B3") = TextBox1.Value End With Unload Me End Sub ClassModule: frmAuswahl Private Sub cmdContinue_Click() Unload Me If RefEdit1.Value = "" Then Exit Sub Select Case Range(RefEdit1.Value).Value Case "a": frmA.Show Case "b": frmB.Show Case "c": frmC.Show End Select End Sub ClassModule: frmB Private Sub cmdContinue_Click() With Worksheets("Ausgabe") .Range("B1") = Label4.Caption .Range("B2") = Label5.Caption .Range("B3") = TextBox1.Value End With Unload Me End Sub ClassModule: frmC Private Sub cmdContinue_Click() With Worksheets("Ausgabe") .Range("B1") = Label4.Caption .Range("B2") = Label5.Caption .Range("B3") = TextBox1.Value End With Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm in Abhängigkeit einer RefEdit-Auswahl anzeigen

Problem:	Wie kann ich in Abhängigkeit der über ein RefEdit-Feld einer UserForm getroffenen Auswahl eine zweite UserForm anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() frmAuswahl.Show End Sub ClassModule: frmAuswahl Private Sub cmdContinue_Click() Unload Me Select Case Range(RefEdit1.Value).Value Case "a": frmA.Show Case "b": frmB.Show Case "c": frmC.Show End Select End Sub ClassModule: frmA Private Sub cmdEintragen_Click() With Worksheets("Ausgabe") .Range("B1") = Label4.Caption .Range("B2") = Label5.Caption .Range("B3") = TextBox1.Value End With Unload Me End Sub ClassModule: frmB Private Sub cmdEintragen_Click() With Worksheets("Ausgabe") .Range("B1") = Label4.Caption .Range("B2") = Label5.Caption .Range("B3") = TextBox1.Value End With Unload Me End Sub ClassModule: frmC Private Sub cmdEintragen_Click() With Worksheets("Ausgabe") .Range("B1") = Label4.Caption .Range("B2") = Label5.Caption .Range("B3") = TextBox1.Value End With Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bereich in RefEdit-Element eingeben und auslesen

Problem:	Wie kann ich in einem RefEdit-Steurelement einer UserForm eine Bereichsangabe machen, indem ich den Bereich im Tabellenblatt mit der Maus selektiere?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallForm() frmRefEdit.Show End Sub ClassModule: frmRefEdit Private Sub CommandButton1_Click() MsgBox RefEdit1.Value Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Labelhintergrund mit ScrollBars festlegen

Problem: Die Hintergrundfarbe eines UserForm-Labels soll über ScrollBars unter Einsatz der RGB-Farben festgelegt werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: Modul1

Sub DialogAufruf()
    frmColors.Show
End Sub

ClassModule: frmColors

Private Sub cmdAbbrechen_Click()
    End
End Sub

Private Sub ScrollBar1_Change()
    Call SetColors
End Sub

Private Sub ScrollBar2_Change()
    Call SetColors
End Sub

Private Sub ScrollBar3_Change()
    Call SetColors
End Sub

Sub SetColors()
    Dim strA As String, strB As String, strC As String, str As String
    Dim varColor As Variant
    str = "00"
    strA = Hex(ScrollBar1.Value)
    strA = strA & String(2 - Len(strA), "0")
    strB = Hex(ScrollBar2.Value)
    strB = strB & String(2 - Len(strB), "0")
    strC = Hex(ScrollBar3.Value)
    strC = strC & String(2 - Len(strC), "0")
    varColor = str & strA & strB & strC
    Label2.Caption = ScrollBar1.Value
    Label3.Caption = ScrollBar2.Value
    Label4.Caption = ScrollBar3.Value
    Label1.BackColor = hex2dec(varColor)
End Sub

Private Sub UserForm_Initialize()

    Call SetColors
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-TextBox-Werte über ScrollBar einlesen

Problem: Wie kann ich über eine ScrollBar Tabellenblattwerte in UserFormTextBoxes einlesen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDatensatz

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub scrDatensatz_Change()
    Dim iCounter As Integer
    For iCounter = 1 To 4
        Controls("TextBox" & iCounter).Text = _
            Cells(scrDatensatz.Value, iCounter)
    Next iCounter
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 4
        Controls("TextBox" & iCounter).Text = _
            Cells(2, iCounter)
    Next iCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmDatensatz.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellwerte über SpinButton und ScrollBar hochzählen

Problem:	Wie kann ich über in einem Tabellenblatt verankerte SpinButtons und ScrollBars Zellwerte hochzählen?
Lösung:	Weisen Sie den Steuerelementen die LinkedCell-Eigenschaft zu.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einfaches Quiz mit 2 vorgegebenen Antworten

Problem:	In einem einfachen Quiz sollen Fragen mit je 2 vorgegebenen Antworten beantwortet werden. Die vorgegebenen Antworten werden über einen Zufallsgenerator auf 2 Schaltflächen verteilt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmQuiz Private Sub cmdAnswer1_Click() Call ClickAnswer(cmdAnswer1) End Sub Private Sub cmdAnswer2_Click() Call ClickAnswer(cmdAnswer2) End Sub Private Sub ClickAnswer(cmd As Control) Dim rng As Range Dim iCol As Integer lblAnswer.Caption = cmd.Caption With ThisWorkbook.Worksheets("Fragen") Set rng = .Columns("B:C").Find(cmd.Caption, lookat:=xlWhole, LookIn:=xlValues) .Cells(spnQuiz.Value, 4).Value = rng.Column End With If spnQuiz.Value = 20 Then spnQuiz.Value = 1 Else spnQuiz.Value = spnQuiz.Value + 1 End If End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdCompute_Click() Dim wks As Worksheet Dim iCount As Integer, iTrue As Integer, iAnswer As Integer Dim sTxt As String Set wks = ThisWorkbook.Worksheets("Fragen") iCount = WorksheetFunction.CountA(wks.Columns(1)) iAnswer = WorksheetFunction.CountA(wks.Columns(4)) iTrue = WorksheetFunction.CountIf(wks.Columns(4), 2) If iAnswer = 0 Then Beep MsgBox "Es wurde noch keine Frage beantwortet!" Exit Sub End If sTxt = "Anzahl Fragen beantwortet:" & vbCrLf sTxt = sTxt & iAnswer & " von " sTxt = sTxt & iCount & " = " sTxt = sTxt & Format(iAnswer / iCount, "0.00%") & vbCrLf & vbCrLf sTxt = sTxt & "Anteil richtige Antworten: " & vbCrLf sTxt = sTxt & iTrue & " von " & iAnswer & " = " sTxt = sTxt & Format(iTrue / iAnswer, "0.00%") MsgBox sTxt End Sub Private Sub cmdReset_Click() ThisWorkbook.Worksheets("Fragen").Columns(4).ClearContents End Sub Private Sub spnQuiz_Change() Dim wks As Worksheet Dim iCol As Integer, iSpan As Integer Set wks = ThisWorkbook.Worksheets("Fragen") iSpan = spnQuiz.Value lblNo.Caption = "Nr.: " & iSpan lblQuestion.Caption = wks.Cells(iSpan, 1).Value Randomize iCol = Int((2 * Rnd) + 1) If iCol = 1 Then cmdAnswer1.Caption = wks.Cells(iSpan, iCol + 1).Value cmdAnswer2.Caption = wks.Cells(iSpan, iCol + 2).Value Else cmdAnswer1.Caption = wks.Cells(iSpan, iCol + 1).Value cmdAnswer2.Caption = wks.Cells(iSpan, iCol).Value End If If Not IsEmpty(wks.Cells(iSpan, 4)) Then lblAnswer.Caption = wks.Cells(iSpan, wks.Cells(iSpan, 4).Value) Else lblAnswer.Caption = "" End If End Sub Private Sub UserForm_Initialize() Dim wks As Worksheet Set wks = ThisWorkbook.Worksheets("Fragen") With spnQuiz .Min = 1 .Max = wks.Range("A1").CurrentRegion.Rows.Count .Value = 1 wks.Columns(4).ClearContents End With End Sub StandardModule: basMain Sub CallForm() frmQuiz.Show End Sub Sub ShowQuestions() Dim wks As Worksheet Dim sPW As String Set wks = ThisWorkbook.Worksheets("Fragen") If wks.Visible = xlSheetVisible Then wks.Visible = xlSheetVeryHidden Exit Sub End If sPW = InputBox("Passwort:", , "Admin") If sPW = "" Then Exit Sub If sPW <> "Admin" Then Beep MsgBox "Sorry, keine Zugangsberechtigung!" Else wks.Visible = xlSheetVisible End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über SpinButton in UserForm-TextBox einlesen

Problem: Mit einem UserForm-SpinButton sollen die Werte aus Spalte B in eine TextBox eingelesen werden. Start ist beim letzten Wert. Die Zahlenreihe darf keine Lücke aufweisen.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSelect

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub spnColumn_Change()
    With Cells(spnColumn.Max - spnColumn.Value + 1, 2)
        txtValue.Text = .Value
        lblAddress.Caption = .Address(False, False)
    End With
End Sub

Private Sub UserForm_Initialize()
    With spnColumn
        .Min = 1
        .Max = WorksheetFunction.CountA(Columns(2))
    End With
    txtValue = Cells(WorksheetFunction.CountA(Columns(2)), 2).Value
    lblAddress.Caption = Cells(spnColumn.Max, 2).Address(False, False)
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmSelect.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen eines zweiten Blattes im ersten über Drehfeld anzeigen

Problem:	Wie kann ich mir die Zeilen des zweiten Tabellenblattes im ersten über Drehfeldauswahl anzeigen lassen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit Drehfeld durch UserForm-TextBoxes blättern und Daten auslesen

Problem:	Wie kann ich Daten aus einem Tabellenblatt Gruppenweise in TextBoxes einer UserForm ein- und auf Schaltflächendruck in das Tabellenblatt wieder aulesen, wobei die Strings wieder aufgeteilt werden müssen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmTexte Private Sub cmdAuslesen_Click() Dim iCells As Integer, iColumns As Integer, iC As Integer Dim sTxt As String iC = 1 If Controls("TextBox1").Text = "" Then Beep MsgBox "Sie müssen zuerst die Drehfeldauswahl treffen!" Exit Sub End If For iCells = 1 To 4 sTxt = Controls("TextBox" & iCells).Text For iColumns = 1 To 4 If iColumns < 4 Then Cells(12, iC) = Left(sTxt, InStr(sTxt, vbLf) - 2) sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, vbLf)) iC = iC + 1 Else Cells(12, iC) = sTxt iC = iC + 1 End If Next iColumns Next iCells End Sub Private Sub CommandButton1_Click() Unload Me End Sub Private Sub drfAuswahl_Change() Dim iCells As Integer, iColumns As Integer Dim iR As Integer, iC As Integer Dim sTxt As String iR = drfAuswahl.Value iC = 1 For iCells = 1 To 4 For iColumns = 1 To 4 sTxt = sTxt & Cells(iR, iC) & vbLf iC = iC + 1 Next iColumns sTxt = Left(sTxt, Len(sTxt) - 1) Controls("TextBox" & iCells) = sTxt sTxt = "" Next iCells End Sub StandardModule: basMain Sub CallForm() frmTexte.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Maximalwert eines SpinButtons nach Zellwert festlegen

Problem: Wie kann ich den Maximalwert eines in einem Tabellenblatt integrierten SpinButtons entsprechend einem Zellwert festlegen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle3

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    If Target.Address <> "$A$1" Then Exit Sub
    SpinButton1.Max = Target
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen bedingt durch SpinButton verschieben

Problem: Wie kann ich durch Betätigung eines SpinButtons einer UserForm Zeilen in einem Tabellenblatt nach unten oder oben verschieben?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMoveRows

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub SpinButton1_SpinUp()
    Dim sTxt As String
    With lstValues
        If .ListIndex = 0 Then Exit Sub
        sTxt = .List(.ListIndex)
        .List(.ListIndex) = .List(.ListIndex - 1)
        .List(.ListIndex - 1) = sTxt
        .ListIndex = .ListIndex - 1
        Rows(.ListIndex + 1).Insert
        Rows(.ListIndex + 3).Copy Rows(.ListIndex + 1)
        Rows(.ListIndex + 3).Delete
    End With
End Sub

Private Sub SpinButton1_SpinDown()
    Dim sTxt As String
    With lstValues
        If .ListIndex = .ListCount - 1 Then Exit Sub
        sTxt = .List(.ListIndex)
        .List(.ListIndex) = .List(.ListIndex + 1)
        .List(.ListIndex + 1) = sTxt
        .ListIndex = .ListIndex + 1
        Rows(.ListIndex + 2).Insert
        Rows(.ListIndex).Copy Rows(.ListIndex + 2)
        Rows(.ListIndex).Delete
    End With
End Sub

Private Sub UserForm_Initialize()
    Dim iRow As Integer
    iRow = 1
    Do Until IsEmpty(Worksheets("Daten").Cells(iRow, 1))
        lstValues.AddItem Cells(iRow, 1)
        iRow = iRow + 1
    Loop
    lstValues.ListIndex = SpinButton1.Value - 1
End Sub

StandardModule: basMain

Sub CallForm()
    frmMoveRows.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen und Spalten über ein Drehfeld ein- und ausblenden

Problem: Wie kann ich das Aus- und Einblenden von Zeilen und Spalten über ein Drehfeld steuern (Excel-Version XL8)?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub spnRows_Change()  
    ActiveCell.Select  
    If spnRows.Value = 0 Then  
        Rows("1:20").Hidden = False  
    Else  
        Rows("1:" & spnRows.Value).Hidden = True  
        Rows(spnRows.Value + 1 & ":20").Hidden = False  
    End If  
End Sub  
  
Private Sub spnColumns_Change()  
    ActiveCell.Select  
    If spnColumns.Value = 0 Then  
        Columns("A:Z").Hidden = False  
    Else  
        Range(Cells(1, 1), Cells(1, spnColumns.Value)). _  
            EntireColumn.Hidden = True  
        Range(Cells(1, spnColumns.Value + 1), Cells(1, 256)). _  
            EntireColumn.Hidden = False  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattzeile mit SpreadSheet-Objekt überlagern

Problem:	Durch die Überlagerung von Tabellenblattzeilen wird der Eindruck einer Fehlberechnung in Excel erweckt.
Lösung:	Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.
VBA-Code:	<pre>ClassModule: Tabelle1 Sub Festlegen() Dim iCol As Integer For iCol = 1 To Spreadsheet1.Columns.Count Spreadsheet1.Columns(iCol).ColumnWidth = 80 Next iCol End Sub Private Sub Spreadsheet1_BeforeCommand(ByVal EventInfo As OWC.SpreadsheetEventInfo) End Sub Private Sub Spreadsheet1_MouseOver(ByVal EventInfo As OWC.SpreadsheetEventInfo) 'MsgBox "Dies ist ein verborgenes SpreadSheet-Steuerlement!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle in UserForm-Spreadsheet einlesen

Problem:	In ein UserForm-SpreadSheet soll der mit Zelle A1 zusammenhängende Bereich eingelesen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub DialogAufruf() frmSpreadSheet.Show End Sub ClassModule: frmSpreadSheet Private Sub cmdCancel_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim rng As Range Dim iRow As Integer, iCol As Integer Set rng = Range("A1").CurrentRegion For iRow = 1 To rng.Rows.Count For iCol = 1 To rng.Columns.Count Spreadsheet1.Cells(iRow, iCol).Value = rng.Cells(iRow, iCol).Value Next iCol Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle in UserForm-Office-Spreadsheet übernehmen

Problem:	Die Tabelle2 soll in ein UserForm-Office-Spreadsheet übernommen werden.
Lösung:	Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmUserTable Private Sub cmdCancel_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim rngSource As Range Dim rngTarget As Object Dim rng As Range Set rngSource = Worksheets("Tabelle2").Range("A1").CurrentRegion Set rngTarget = Spreadsheet1.Cells.Range(rngSource.Address) With Spreadsheet1 rngTarget.Value = rngSource.Value End With For Each rng In rngSource.Rows(1).Columns rngTarget.Columns(rng.Column).ColumnWidth = rng.Width Next rng End Sub ClassModule: Tabelle3 Private Sub CommandButton1_Click() ActiveCell.Select frmUserTable.Show End Sub ClassModule: Tabelle1</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblatt in UserForm

Problem: Wie kann ich ein Tabellenblatt mit Zeilen und Spalten in eine Userform übernehmen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle3

Private Sub CommandButton1_Click()
    ActiveCell.Select
    frmUserTable.Show
End Sub

StandardModule: basMain

Sub CallForm()
    frmSpreadsheet.Show
End Sub

ClassModule: frmSpreadsheet

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim iRow As Integer, iCol As Integer
    For iRow = 1 To 5
        For iCol = 1 To 4
            Cells(iRow, iCol).Value = _
                Spreadsheet1.Cells(iRow, iCol).Value
        Next iCol
    Next iRow
End Sub

Private Sub UserForm_Initialize()
    Dim iRow As Integer, iCol As Integer
    For iRow = 1 To WorksheetFunction.CountA(Columns(1))
        For iCol = 1 To WorksheetFunction.CountA(Rows(1))
            Spreadsheet1.Cells(iRow, iCol).Value = _
                Cells(iRow, iCol).Value
        Next iCol
    Next iRow
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensatz eintragen oder editieren

Problem: Es soll geprüft werden, ob die Kunden-Nr. schon vorhanden ist. Wenn nein, wird der Datensatz eingetragen, andernfalls editiert.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmKunden

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdEintragen_Click()
    Dim vRow As Variant
    Dim iRow As Integer
    vRow = Application.Match(txtNo.Text, Columns(1), 0)
    If IsError(vRow) Then
        iRow = WorksheetFunction.CountA(Columns(1)) + 1
    Else
        iRow = vRow
    End If
    Cells(iRow, 1).Value = txtNo.Text
    If optHerr.Value Then
        Cells(iRow, 2).Value = "Herr"
    Else
        Cells(iRow, 2).Value = "Frau"
    End If
    Cells(iRow, 3).Value = txtNachname.Text
    Cells(iRow, 4).Value = txtVorname.Text
End Sub

StandardModule: Modul1

Sub CallForm()
    frmKunden.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über UserForm-Datensätze nur eintragen, wenn nicht vorhanden

Problem: Es soll geprüft werden, ob die Kunden-Nr. schon vorhanden ist. Wenn nein, wird der Datensatz eingetragen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmKunden

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdEintragen_Click()
    Dim iRow As Integer
    If IsError(Application.Match(txtNo.Text, Columns(1), 0)) Then
        iRow = WorksheetFunction.CountA(Columns(1)) + 1
        Cells(iRow, 1).Value = txtNo.Text
        If optHerr.Value Then
            Cells(iRow, 2).Value = "Herr"
        Else
            Cells(iRow, 2).Value = "Frau"
        End If
        Cells(iRow, 3).Value = txtNachname.Text
        Cells(iRow, 4).Value = txtVorname.Text
    Else
        MsgBox "Kundennummer ist bereits vorhanden!"
        With txtNo
            .SetFocus
            .SelStart = 0
            .SelLength = .TextLength
        End With
    End If
End Sub

StandardModule: Modul1

Sub CallForm()
    frmKunden.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Steuerelemente aus Tabellenblatt löschen

Problem:	Es sollen Steuerelemente aus Tabellenblättern gelöscht werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn zwei Schaltflächen zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub DeleteAll() Worksheets("Tabelle2").OLEObjects.Delete Worksheets("Tabelle2").Select End Sub Sub DeleteCBOs() Dim oOle As OLEObject For Each oOle In Worksheets("Tabelle3").OLEObjects If TypeName(oOle.Object) = "ComboBox" Then oOle.Delete End If Next oOle Worksheets("Tabelle3").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateinamen in TabStrip-Elemente einlesen

Problem: Wie kann ich in einer Tabelle eingetragene Dateinamen als Captions in TabStrip-Elemente einlesen? Nach Anklicken eines TabStrip-Elements soll die jeweilige Datei geöffnet werden. Die Dateieigenschaften sollen ausgelesen und angezeigt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDateien

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub TabStrip1_Change()
    Dim sPath As String
    Application.ScreenUpdating = False
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    sPath = ThisWorkbook.Path & "\"
    If Dir(sPath & TabStrip1.SelectedItem.Caption & ".xls") = "" Then
        Beep
        MsgBox "Die Datei existiert nicht!"
        Exit Sub
    End If
    Workbooks.Open sPath & TabStrip1.SelectedItem.Caption & ".xls"
    With ActiveWorkbook
        lblEigenschaften.Caption = "Titel: " & _
            .BuiltinDocumentProperties(1) & vbCrLf & _
            "Thema: " & .BuiltinDocumentProperties(2) & vbCrLf & _
            "Autor: " & .BuiltinDocumentProperties(3) & vbCrLf & _
            "Schlüsselworte: " & .BuiltinDocumentProperties(4)
        .Close savechanges:=False
    End With
ERRORHANDLER:
    Application.EnableEvents = True
    Application.ScreenUpdating = True
End Sub

Private Sub UserForm_Initialize()
    Dim intCounter As Integer
    For intCounter = 1 To 4
        TabStrip1.Tabs(intCounter - 1).Caption = Cells(intCounter, 1)
    Next intCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmDateien.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schenkungssteuern in einer UserForm berechnen

Problem:	Über einen UserForm-Dialog werden Erbschafts- und Schenkungssteuern berechnet und in die Liste eingetragen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Dim NumWert Dim NumSatz Dim NumSteuer Private Sub CbxErbe_Change() Worksheets("Hilfsfelder").Visible = True Worksheets("Hilfsfelder").Activate Range("A1").Select ActiveCell.Value = FrmErbe.CbxErbe Range("B1").Select FrmErbe.TbxSteuerkl.Value = ActiveCell.Value End Sub Private Sub CommandButton1_Click() Worksheets("Tabelle1").Activate Rows("3:3").Select Selection.Insert Shift:=x1Down Range("A3").Select Range("A3").Select ActiveCell.Value = FrmErbe.CbxErbe Range("B3").Select ActiveCell.Value = FrmErbe.TbxSteuerkl Range("C3").Select ActiveCell.Value = CDb1(NumWert) Range("D3").Select ActiveCell.Value = CDb1(NumSatz) Range("F3").Select ActiveCell.Value = CDb1(NumSteuer) FrmErbe.Hide End Sub Private Sub TbxSatz_Change() NumWert = FrmErbe.TbxWert NumSatz = FrmErbe.TbxSatz If NumWert <> "" Then If NumSatz <> "" Then NumSteuer = NumWert * NumSatz End If End If FrmErbe.TbxSteuer.Value = NumSteuer End Sub Private Sub TbxSatz_AfterUpdate() FrmErbe.TbxSatz.Text = Format(NumSatz, "#,##0.00") FrmErbe.TbxSteuer.Text = Format(NumSteuer, "#,##0.00") End Sub StandardModule: basMain Sub CallForm() FrmErbe.Show End Sub Sub EinAusblenden() With wksData If .Visible = xlSheetVeryHidden Then .Visible = xlSheetVisible Else .Visible = xlSheetVeryHidden End If End With End Sub ClassModule: frmErbe Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cboErbe_Change() If cboErbe.Value = "Eltern und Großeltern" Then Frame1.Visible = True Else Frame1.Visible = False End If Call SteuerBerechnen End Sub Private Sub cmdEintragen_Click() Dim iRow As Integer Dim sValue As String iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 sValue = txtSatz.Text sValue = WorksheetFunction.Substitute(sValue, " %", "") Cells(iRow, 1).Value = cboErbe.Value Cells(iRow, 2).Value = txtStkl.Text Cells(iRow, 3).Value = CDb1(txtWert.Text) Cells(iRow, 4).Value = CDb1(sValue) / 100 Cells(iRow, 5).Value = CDb1(txtSteuer.Text) End Sub Private Sub optErbschaft_Change() Call SteuerBerechnen End Sub Private Sub txtWert_Exit(ByVal Cancel As MSForms.ReturnBoolean) txtWert.Text = Format(txtWert.Text, "#,##0") Call SteuerBerechnen End Sub Private Sub txtWert_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer) If IsNumeric(Chr(KeyCode)) = False And KeyCode <> 9 And KeyCode <> 13 Then KeyCode = 0 End If End Sub Private Sub UserForm_Initialize() cboErbe.List = wksData.Range("Liste").Value End Sub Private Sub SteuerBerechnen() Dim lSerw As Long Dim dSatz As Double Dim iRow As Integer, iCol As Integer If cboErbe.Value = "" Then Exit Sub iRow = WorksheetFunction.Match(cboErbe.Value & "*", wksData.Columns(7), 0) For iCol = 8 To 10 If Not IsEmpty(wksData.Cells(iRow, iCol)) Then Exit For Next iCol If InStr(cboErbe.Value, "Eltern und Großeltern") Then If optSchenkung Then iRow = iRow + 1 iCol = iCol + 1 End If End If txtStkl.Text = wksData.Cells(2, iCol).Value txtFrei.Text = Format(wksData.Cells(iRow, iCol).Value, "#,##0") If txtWert = "" Then Exit Sub lSerw = CDb1(txtWert.Text) - CDb1(txtFrei.Text) If lSerw > 0 Then txtSerw.Text = Format(CDb1(txtWert.Text) - CDb1(txtFrei.Text), "#,##0") iRow = WorksheetFunction.Match(CDb1(txtSerw.Text), wksData.Columns(2), 1) iCol = WorksheetFunction.Match(txtStkl.Text, wksData.Rows(2), 0) dSatz = wksData.Cells(iRow, iCol).Value txtSatz.Text = Format(dSatz, "0.00 %") txtSteuer.Text = Format(CDb1(txtSerw.Text) * dSatz, "#,##0") Else txtSerw.Text = "0" txtSatz.Text = "0" txtSteuer.Text = "0" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Verschieben von Text diesen in Label anzeigen

Problem: Beim Verschieben von Text von einer UserForm-TextBox in eine andere soll dieser Text in einem Label angezeigt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMove

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub txtA_BeforeDragOver( _
    ByVal Cancel As MSForms.ReturnBoolean, _
    ByVal Data As MSForms.DataObject, _
    ByVal X As Single, _
    ByVal Y As Single, _
    ByVal DragState As MSForms.fmDragState, _
    ByVal Effect As MSForms.ReturnEffect, _
    ByVal Shift As Integer)
    Label1.Caption = txtA.Text
End Sub

Private Sub txtB_BeforeDragOver( _
    ByVal Cancel As MSForms.ReturnBoolean, _
    ByVal Data As MSForms.DataObject, _
    ByVal X As Single, _
    ByVal Y As Single, _
    ByVal DragState As MSForms.fmDragState, _
    ByVal Effect As MSForms.ReturnEffect, _
    ByVal Shift As Integer)
    Label1.Caption = txtB.Text
End Sub

StandardModule: Modul1

Sub CallForm()
    frmMove.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei TextBox-Eingaben Mustervergleich und Formatierung

Problem:	Bei Textfeldeingaben nach einem vorgegebenen Muster soll eine Formatierung erfolgen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmFormat Private Sub cmdCancel_Click() Unload Me End Sub Private Sub txtData_Change() If txtData.Text Like "####[a-zA-Z]" Then txtData.Text = Left(txtData.Text, 2) & "." & Mid(txtData.Text, 3, 2) & ". " & Right(txtData. Text, 1) End If End Sub StandardModule: Modul1 Sub CallForm() frmFormat.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In Namensliste TextBox-Inhalte suchen

Problem: Die in dem xlVeryHidden ausgeblendeten Blatt hinterlegte Namensliste soll gem. UserForm-TextBox-Eingaben überprüft werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: Modul1

Sub AusEinblenden()
    With Worksheets("Namen")
        If .Visible = xlVeryHidden Then
            .Visible = xlSheetVisible
        Else
            .Visible = xlVeryHidden
        End If
    End With
End Sub

Sub CallForm()
    frmName.Show
End Sub

ClassModule: frmName

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdCheck_Click()
    Dim iRow As Integer
    Dim bln As Boolean
    iRow = 1
    With ThisWorkbook.Worksheets("Namen")
        Do Until IsEmpty(.Cells(iRow, 1))
            If .Cells(iRow, 1).Value = txtName.Value Then
                If .Cells(iRow, 2).Value = txtVorname.Value Then
                    bln = True
                    Exit Do
                End If
            End If
            iRow = iRow + 1
        Loop
    End With
    If bln Then
        lblMsg.Caption = "Name wurde gefunden"
    Else
        lblMsg.Caption = "Name nicht gefunden!"
    End If
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolgen aus 1. mit Leerzeichen in 2. TextBox übernehmen

Problem:	Der in die erste TextBox einzugebende Text soll automatisch in die zweite mit einem Leerzeichen nach jedem vierten Zeichen übernommen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmSpace Private Sub cmdCancel_Click() Unload Me End Sub Private Sub txtSource_Change() txtTarget.Text = SetSpaces(txtSource.Text) End Sub StandardModule: Modul1 Function SetSpaces(sTxt As String) As String Dim iChar As Integer, iSpace As Integer Dim sText As String For iChar = 1 To Len(sTxt) sText = sText & Mid(sTxt, iChar, 1) iSpace = iSpace + 1 If iSpace = 4 Then sText = sText & " " iSpace = 0 End If Next iChar SetSpaces = sText End Function Sub CallForm() frmSpace.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

TextBox-Eintrag von Industrie- und Uhrzeit wandeln

Problem: Eine im Industrieformat eingegebene Uhrzeit soll beim Verlassen der TextBox in das Uhrzeitformat gewandelt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: Modul1

Sub CallForm()
    frmTimeFormat.Show
End Sub

ClassModule: frmTimeFormat

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub txtStart_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    txtStart.Text = Format(CDbl(txtStart.Text), "h:mm")
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolgen aus Zellen in TextBox mit Leerzeichen darstellen

Problem:	In den TextBoxes sollen die Texte aus Spalte A mit einem Leerzeichen nach jedem vierten Zeichen dargestellt werden. Die TextBoxes sind verknüpft mit den darunterliegenden Zellen, in denen auf eine benutzerdefinierte Funktion zugegriffen wird.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function SetSpaces(sTxt As String) As String Dim iChar As Integer, iSpace As Integer Dim sText As String For iChar = 1 To Len(sTxt) sText = sText & Mid(sTxt, iChar, 1) iSpace = iSpace + 1 If iSpace = 4 Then sText = sText & " " iSpace = 0 End If Next iChar SetSpaces = sText End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriff aus UserForm-TextBox in Tabelle suchen

Problem: Ein Suchbegriff aus einer UserForm-TextBox soll in Spalte A gesucht und die Fundstelle ausgewählt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSearch

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdSearch_Click()
    Dim rng As Range
    If txtSearch.Text = "" Then
        Beep
        MsgBox "Bitte Suchbegriff eingeben!"
        txtSearch.SetFocus
        Exit Sub
    End If
    Set rng = Columns(1).Find( _
        What:=txtSearch.Text, _
        After:=ActiveCell, _
        LookIn:=xlFormulas, _
        LookAt:=xlWhole, _
        SearchOrder:=xlByRows, _
        SearchDirection:=xlNext, _
        MatchCase:=False)
    If rng Is Nothing Then
        Beep
        MsgBox "Suchbegriff wurde nicht gefunden!"
        Exit Sub
    End If
    rng.Select
End Sub

StandardModule: Modul1

Sub CallForm()
    frmSearch.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei UserForm-TextBox-Eingaben Formate erzwingen

Problem: Bei UserForm-TextBoxes sollen Datumsbzw. Uhrzeitangaben in einem bestimmten Format erzwungen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDateTime

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdMessage_Click()
    MsgBox "Datum: " & Format(CDate(txtDate.Text), "dd.mm.yy") & vbCrLf & vbCrLf & _
        "Zeit: " & Format(CDate(txtTime.Text), "hh:mm")
End Sub

Private Sub txtDate_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    With txtDate
        If Not IsDate(.Text) Or _
            InStr(.Text, ":") Or _
            .TextLength <> 8 Then
            Cancel = True
            Beep
            .SetFocus
            .SelStart = 0
            .SelLength = .TextLength
        End If
    End With
End Sub

Private Sub txtTime_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    With txtTime
        If Not IsDate(.Text) Or _
            InStr(.Text, ":") = False Or _
            InStr(.Text, ".") Or _
            .TextLength <> 5 Then
            Cancel = True
            Beep
            .SetFocus
            .SelStart = 0
            .SelLength = .TextLength
        End If
    End With
End Sub

StandardModule: Modul1

Sub CallForm()
    frmDateTime.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopieren und Einfügen in UserForm ohne Excel-ShortCuts

Problem:	Kopieren eines Textfeldinhalts und Übergabe an ein anderes Textfeld mit ShortCuts, wobei die hierfür vorgesehenen ShortCuts Strg+c und Strg+v nicht verwendet werden dürfen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmCopyPaste Dim gBln As Boolean Private Sub cmdCancel_Click() Unload Me End Sub Private Sub txtSource_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer) Dim ClipAbLage As DataObject Set ClipAbLage = New DataObject If gBln And KeyCode = 116 Then ClipAbLage.SetText txtSource.Text ClipAbLage.PutInClipboard End If If KeyCode = 17 Then gBln = True Else gBln = False End Sub Private Sub txtTarget_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer) If KeyCode = 120 Then SendKeys "+v" End Sub StandardModule: Modul1 Sub CallForm() frmCopyPaste.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus Tabellenblatt-TextBoxes sollen addiert werden

Problem:	Die Werte der TextBoxes sollen addiert werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdAdd_Click() Dim txt As OLEObject Dim iSum As Integer For Each txt In OLEObjects If TypeName(txt.Object) = "TextBox" Then If txt.Object.Text <> "" Then iSum = iSum + CInt(txt.Object.Text) End If End If Next txt MsgBox "Summe: " & iSum End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm mit ShortCut aufrufen und Produktdaten eingeben

Problem:	Über die Tastenkombination Strg+a wird ein Dialog aufgerufen. Wenn sich der Cursor in einer Produktzeile (Zeile 2 oder 3) befindet, werden die Produktdaten aus der Datenbank eingelesen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub DialogAufruf() frmProdukte.Show End Sub ClassModule: frmProdukte Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim iCol As Integer, iCounter As Integer Dim sLabel As String For iCounter = 1 To 4 If Controls("Label" & iCounter).Visible Then sLabel = Controls("Label" & iCounter).Caption sLabel = Left(sLabel, Len(sLabel) - 1) Cells(ActiveCell.Row, iCounter * 2).Value = Controls("TextBox" & iCounter).Text Cells(ActiveCell.Row, (iCounter * 2) + 1).Value = sLabel End If Next iCounter End Sub Private Sub UserForm_Initialize() Dim var As Variant Dim iCounter As Integer, iColl As Integer var = Application.Match(Cells(ActiveCell.Row, 1).Value, wksData.Columns(1), 0) If Not IsError(var) Then iColl = ((WorksheetFunction.CountA(wksData.Rows(var)) - 1) / 2) For iCounter = 1 To iColl With Controls("Label" & iCounter) .Visible = True .Caption = wksData.Cells(var, (iCounter * 2) + 1).Value & ":" End With With Controls("TextBox" & iCounter) .Visible = True .Text = wksData.Cells(var, (iCounter * 2)).Value End With Next iCounter End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

TextBox-Inhalt mit und ohne Zeilenumbrüche auslesen

Problem:	Der Text der TextBox soll mit und ohne Zeilenumbrüche ausgelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn zwei Schaltflächen zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdA_Click() MsgBox txtBox.Text End Sub Private Sub cmdB_Click() Dim sTxt As String sTxt = txtBox.Text sTxt = WorksheetFunction.Substitute(sTxt, vbNewLine, " ") MsgBox sTxt End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Während der Laufzeit TextBox erstellen und auslesen

Problem: Während der Laufzeit soll eine TextBox hinzugefügt und ausgelesen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDesigner

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdCreateText_Click()
    Dim oTxt As MSForms.TextBox
    Set oTxt = Me.Controls.Add("forms.TextBox.1")
    With oTxt
        .Name = "MeineTextBox"
        .Left = 10
        .Top = 10
        .Text = "Neue TextBox"
    End With
End Sub

Private Sub cmdReadText_Click()
    MsgBox Me.Controls("MeineTextBox").Text
End Sub

Private Sub UserForm_Click()

End Sub

StandardModule: Modul1

Sub CallForm()
    frmDesigner.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einfügen eines Sonderzeichens in einer UserForm-TextBox

Problem:	An der Schreibposition des Cursors in einer UserForm-TextBox soll ein Sonderzeichen eingefügt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmInsert Dim blnCtrl As Boolean Private Sub txtInsert_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer) With txtInsert If KeyCode = 17 Then blnCtrl = True ElseIf KeyCode = 82 And blnCtrl Then .Text = Left(.Text, .SelStart) & Chr(174) & Right(.Text, Len(.Text) - .SelStart) blnCtrl = False ElseIf KeyCode = 84 And blnCtrl Then .Text = Left(.Text, .SelStart) & Chr(153) & Right(.Text, Len(.Text) - .SelStart) blnCtrl = False Else blnCtrl = False End If End With End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub UserForm_Initialize() txtInsert.SelStart = 4 End Sub StandardModule: Modul1 Sub CallForm() frmInsert.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-TextBox-Eingaben im Format [hh]:mm in Zellen übertragen

Problem:	Eingaben in UserForm-TextBoxes im Format [hh]:mm sollen in diesem Format in den Spalten A:C eingetragen werden. Die TextBoxes sollen eine Eingabekontrolle für die Muster: ##:##, ###:## und ####:## enthalten.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmHours Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdInsert_Click() Dim iRow As Integer, iCol As Integer Dim sTxt As String, sFull As String, sTime As String, sHours As String iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 For iCol = 1 To 3 sTime = Controls("TextBox" & iCol).Text sHours = Left(sTime, InStr(sTime, ":") - 1) sTxt = sHours sTxt = CInt(sTxt) Mod 24 sFull = CInt(sHours) - CInt(sTxt) sTxt = sTxt & Right(sTime, Len(sTime) - InStr(sTime, ":") + 1) Cells(iRow, iCol).Value = CDate(sTxt) + CInt(sFull) / 24 Next iCol End Sub Private Sub TextBox1_Exit(ByVal Cancel As MSForms.ReturnBoolean) Dim sTxt As String With TextBox1 sTxt = .Text If Not sTxt Like "##:##" And _ Not sTxt Like "###:##" And _ Not sTxt Like "####:##" Then Cancel = True .SetFocus .SelStart = 0 .SelLength = .TextLength End If End With End Sub Private Sub TextBox2_Exit(ByVal Cancel As MSForms.ReturnBoolean) Dim sTxt As String With TextBox2 sTxt = .Text If Not sTxt Like "##:##" And _ Not sTxt Like "###:##" And _ Not sTxt Like "####:##" Then Cancel = True .SetFocus .SelStart = 0 .SelLength = .TextLength End If End With End Sub Private Sub TextBox3_Exit(ByVal Cancel As MSForms.ReturnBoolean) Dim sTxt As String With TextBox3 sTxt = .Text If Not sTxt Like "##:##" And _ Not sTxt Like "###:##" And _ Not sTxt Like "####:##" Then Cancel = True .SetFocus .SelStart = 0 .SelLength = .TextLength End If End With End Sub StandardModule: Modul1 Sub CallForm() frmHours.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-TextBox-Einträge auf Tabellenblätter verteilen

Problem: UserForm-TextBox-Einträge sollen auf Tabellenblätter verteilt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmVerteilen

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdWks1_Click()
    Worksheets("Tabelle1").Cells(1, 1).Value = TextBox1.Text
    Worksheets("Tabelle1").Cells(2, 1).Value = TextBox2.Text
End Sub

Private Sub cmdWks2_Click()
    Worksheets("Tabelle2").Cells(1, 1).Value = TextBox3.Text
    Worksheets("Tabelle2").Cells(2, 1).Value = TextBox4.Text
End Sub

Private Sub cmdWks3_Click()
    Worksheets("Tabelle3").Cells(1, 1).Value = TextBox5.Text
    Worksheets("Tabelle3").Cells(2, 1).Value = TextBox6.Text
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    For iCounter = 1 To 6
        Controls("TextBox" & iCounter).Text = "TextBox" & iCounter
    Next iCounter
End Sub

StandardModule: Modul1

Sub CallDialog()
    frmVerteilen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit der TabTaste in Tabellenblatt-TextBoxes springen

Problem: Mit der TabTaste soll in Tabelle1 von TextBox zu TextBox gesprungen werden. Die TextBoxes sind in einer eigenen Klasse zusammengefaßt.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: Tabelle1

Dim txtBoxes(1 To 26) As New Klasse1

Private Sub Worksheet_Activate()
    Dim ctr As OLEObject
    Dim iCounter As Integer
    For Each ctr In Me.OLEObjects
        If TypeName(ctr.Object) = "TextBox" Then
            iCounter = iCounter + 1
            Set txtBoxes(iCounter).TxtGroup = ctr.Object
        End If
    Next ctr
End Sub
ClassModule: Klasse1

Public WithEvents TxtGroup As MSForms.TextBox

Private Sub TxtGroup_KeyDown(ByVal KeyCode As _
    MSForms.ReturnInteger, ByVal Shift As Integer)
    If KeyCode = vbKeyTab Then
        If TxtGroup.Index < 26 Then
            Tabelle1.OLEObjects(TxtGroup.Index + 1).Activate
        Else
            Tabelle1.OLEObjects(1).Activate
        End If
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

HTML-Text einer Internetseite in eine TextBox einlesen

Problem: In eine UserForm-TextBox soll der HTML-Text einer Internet-Seite eingelesen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: Modul1

Sub DialogAufruf()
    frmInet.Show
End Sub
ClassModule: frmInet

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim sURL As String
    sURL = "http://www.herber.de/"
    TextBox1.Text = Inet1.OpenURL(sURL)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-TextBox-Werte multiplizieren und eintragen

Problem: Der Wert aus 2 UserForm-TextBoxes soll multipliziert und in das aktive Blatt eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmEintragen

Private Sub cmdEintragen_Click()
    Dim dValue As Double
    dValue = CDbl(TextBox1.Text) * CDbl(TextBox2.Text)
    Cells(ActiveCell.Row, 3).Value = Date
    Cells(ActiveCell.Row, 4) = dValue
    Unload Me
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmEintragen.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfen, ob eine TextBox mit vorgegebenem Namen existiert

Problem:	Es soll geprüft werden, ob eine TextBox mit dem Namen txtName im aktiven Blatt existiert.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Function Pruefen(sName As String) As Boolean Dim oName As Object On Error Resume Next Set oName = OLEObjects(sName) On Error GoTo 0 If oName Is Nothing Then Err.Clear Pruefen = False Else Pruefen = True End If End Function Private Sub cmdPruefen_Click() MsgBox Pruefen("txtName") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellwerte UserForm-TextBoxwerte hinzuaddieren

Problem: Auf UserForm-Schaltflächendruck sollen zu Zellwerten TextBox-Werte hinzuaddiert werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAddieren

Private Sub cmdAddieren_Click()
    If TextBox1.Text <> "" Then
        Range("A1").Value = Range("A1").Value + CDb1(TextBox1.Text)
    End If
    If TextBox2.Text <> "" Then
        Range("B3").Value = Range("B3").Value + CDb1(TextBox2.Text)
    End If
    If TextBox3.Text <> "" Then
        Range("C6").Value = Range("C6").Value + CDb1(TextBox3.Text)
    End If
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmAddieren.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

TextBox im Arbeitsblatt löschen, wenn vorhanden

Problem:	Es soll geprüft werden, ob im aktiven Blatt eine TextBox vorhanden ist. Wenn ja, soll sie gelöscht werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub TestDelete() Dim oTxt As OLEObject Dim bln As Boolean For Each oTxt In ActiveSheet.OLEObjects If TypeName(oTxt.Object) = "TextBox" Then oTxt.Delete bln = True End If Next oTxt If bln = False Then Beep MsgBox "Keine TextBox zum Löschen vorhanden!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus geschlossener Arbeitsmappe in UserForm-TextBox

Problem: Aus einer geschlossenen Arbeitsmappe soll der Wert eines benannten Bereiches in eine UserForm-TextBox übernommen werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmImport

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim sPath As String, sFile As String, sWks As String
    Dim sRange As String
    sPath = "c:\test"
    sFile = "test.xls"
    sWks = "Tabelle1"
    sRange = "Auftraggeber"
    If Dir(sPath & "\" & sFile) = "" Then
        Beep
        MsgBox "Datei " & sFile & " nicht gefunden!"
        Exit Sub
    End If
    On Error Resume Next
    Range("IV1").Formula = "=" & sPath & _
        "\" & sFile & "]" & sWks & "!" & sRange
    txtImport.Text = Range("IV1").Value
    Range("IV1").ClearContents
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmImport.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

TextBox-Eingabe mit Formel weiterberechnen und zurückgeben

Problem: Der Wert einer TextBox soll über eine Formel weiterberechnet und an die TextBox zurückgegeben werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmCalc

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    txtBerechnen.Text = Sin(CDbl(txtBerechnen.Text))
End Sub

Private Sub UserForm_Initialize()
    txtBerechnen = 122.3
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmCalc.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach TextBoxeingabe Datensätze in UserForm-ListBox übernehmen

Problem: Bei Eingabe der vierstelligen Auftragsnummer in eine UserForm-TextBox sollen die entsprechenden Datensätze in eine UserForm-ListBox übernommen werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmValues

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub txtValue_Change()
    Dim arr() As Variant
    Dim iRow As Integer, iCol As Integer, iCounter As Integer
    If txtValue.TextLength = 4 Then
        iRow = 2
        Do Until IsEmpty(Cells(iRow, 1))
            If Cells(iRow, 1) = txtValue.Text Then
                iCounter = iCounter + 1
                ReDim Preserve arr(1 To 4, 1 To iCounter)
                For iCol = 1 To 4
                    arr(iCol, iCounter) = Cells(iRow, iCol).Value
                Next iCol
            End If
            iRow = iRow + 1
        Loop
        lstValues.Column = arr
    End If
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmValues.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Verlassen eines Textfeldes Prüfung, ob Datumseintrag

Problem: Nach einer UserForm-Textfeldeingabe soll eine Datumsprüfung erfolgen und bei negativem Ergebnis das Verlassen des Textfeldes verhindert werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: Modul1

Sub DialogAufruf()
    frmEingabe.Show
End Sub

ClassModule: frmEingabe

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub txtEingabe_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    With txtEingabe
        If Not IsDate(.Text) Then
            .Text = "Date only!"
            .SelStart = 0
            .SelLength = .TextLength
            Cancel = True
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriffe in UserForm eingeben, Fundstelle importieren

Problem: Über eine UserForm sollen Suchbegriffe eingegeben und eine Suche gestartet werden. War die Suche erfolgreich, ist das Suchergebnis in Tabelle2 einzutragen. Existiert ein Hyperlink, ist das zugehörige Bild einzufügen.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: frmInsertPicture

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim pct As Picture
    Dim rng As Range
    Set rng = Cells.Find(TextBox1.Text, LookIn:=xlValues, lookat:=xlWhole)
    If rng Is Nothing Then
        Beep
        MsgBox "Suchbegriff wurde nicht gefunden!"
    End If
    With Worksheets("Tabelle2")
        .Select
        .Range("B1").Select
        .Range("A1").Value = rng.Value
        If Dir(rng.Hyperlinks(1).Address) <> "" Then
            Set pct = .Pictures.Insert(rng.Hyperlinks(1).Address)
        End If
    End With
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmInsertPicture.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-TextBox-Inhalte mit Grafik-Hyperlinks einfügen

Problem: Über eine UserForm sollen TextBox-Inhalte zusammen mit Hyperlinks zu auszuwählenden Grafiken in das Tabellenblatt übernommen werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: frmHyperLink

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim iCounter As Integer
    Dim sTag As String
    Rows(1).Clear
    For iCounter = 1 To 3
        sTag = Controls("CommandButton" & iCounter).Tag
        Cells(1, iCounter).Value = Controls("TextBox" & iCounter).Text
        If sTag <> "False" And sTag <> "" Then
            ActiveSheet.Hyperlinks.Add _
                Anchor:=Cells(1, iCounter), _
                Address:=sTag
        End If
    Next iCounter
    Unload Me
End Sub

Private Sub CommandButton1_Click()
    CommandButton1.Tag = SelectPicture
End Sub

Private Sub CommandButton2_Click()
    CommandButton2.Tag = SelectPicture
End Sub

Private Sub CommandButton3_Click()
    CommandButton3.Tag = SelectPicture
End Sub

Private Function SelectPicture()
    Dim var As Variant
    var = Application.GetOpenFilename("Bild-Dateien (*.gif), *.gif")
    If var = False Then
        SelectPicture = "False"
    Else
        SelectPicture = var
    End If
End Function

StandardModule: Modul1

Sub CallForm()
    frmHyperLink.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit Tab-Taste von TextBox zu TextBox springen

Problem:	Wie kann ich mich in einem Tabellen mit der Tab-Taste von einer TextBox zur anderen bewegen?
Lösung:	Den nachfolgenden Code in das Klassenmodul der Tabelle einfügen.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub TextBox1_KeyDown(ByVal KeyCode As _ MSForms.ReturnInteger, ByVal Shift As Integer) If KeyCode = vbKeyTab Then TextBox2.Activate End Sub Private Sub TextBox2_KeyDown(ByVal KeyCode As _ MSForms.ReturnInteger, ByVal Shift As Integer) If KeyCode = vbKeyTab Then TextBox3.Activate End Sub Private Sub TextBox3_KeyDown(ByVal KeyCode As _ MSForms.ReturnInteger, ByVal Shift As Integer) If KeyCode = vbKeyTab Then TextBox1.Activate End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auf Doppelte prüfen, bevor Tabelleneintrag

Problem:	Vor dem Eintrag eines UserForm-TextBox-Wertes in eine Tabelle soll geprüft werden, ob der Wert bereits vorhanden ist.
Lösung:	Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.
VBA-Code:	<pre>ClassModule: frmValues Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim var As Variant Dim intRow As Integer var = Application.Match(CDbl(txtValue.Text), Columns(1), 0) If Not IsError(var) Then MsgBox "Wert ist bereits vorhanden!" Else intRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 Cells(intRow, 1) = txtValue.Text End If End Sub StandardModule: basMain Sub CallForm() frmValues.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in jeweils aktive UserForm-TextBox eintragen

Problem: Ein Wert soll in die gerade aktive TextBox einer UserForm eingetragen werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: frmTextBoxes

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim txt As String
    txt = InputBox("Wert eintragen:", , 12)
    If txt = "" Then End
    gtxtBox.Text = txt
End Sub

Private Sub TextBox1_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    Set gtxtBox = Controls("TextBox1")
End Sub

Private Sub TextBox2_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    Set gtxtBox = Controls("TextBox2")
End Sub

Private Sub TextBox3_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    Set gtxtBox = Controls("TextBox3")
End Sub

Private Sub TextBox4_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    Set gtxtBox = Controls("TextBox4")
End Sub

StandardModule: basMain

Public gtxtBox As MSForms.TextBox

Sub CallForm()
    frmTextBoxes.Show
End Sub
```

Anzeige eines Zahlwortes synchron zur TextBox-Zahleneingabe

Problem:	Synchron zu Eingaben in eine UserForm-TextBox soll in einer zweiten TextBox das zugehörige Zahlwort angezeigt werden.
Lösung:	Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.
VBA-Code:	ClassModule: Tabelle1 Private Sub cmdDialogAufruf_Click() frmZahlwort.Show End Sub ClassModule: frmZahlwort Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub txtNumber_Change() txtWord.Text = ZahlWort(CDbl(txtNumber.Text)) End Sub StandardModule: basMain Sub CallForm() frmZahlwort.Show End Sub Function ZahlWort(dblZahl As Double, Optional bolArt As Variant) Dim arrArt As Variant Dim intCounter As Integer, intCts As Integer Dim strWert As String, strTmp As String, strSuffix As String On Error Resume Next If bolArt = 1 Then If Err = 0 Then intCts = (dblZahl - Fix(dblZahl)) * 100 strSuffix = " " & Format(CStr(intCts), "00") & "/100" Else bolArt = 0 End If End If On Error GoTo 0 dblZahl = Fix(dblZahl) strTmp = Right(CStr(dblZahl), 3) strWert = Part(strTmp) For intCounter = 1 To 4 strTmp = CStr(dblZahl) Select Case Len(strTmp) Case Is < 1 + 3 * intCounter ZahlWort = strWert & strSuffix Exit Function Case Is < 4 + 3 * intCounter strTmp = Left(strTmp, Len(strTmp) - intCounter * 3) Case Else strTmp = Left(Right(strTmp, 3 + intCounter * 3), 3) End Select Select Case intCounter Case 1: arrArt = Array("tausend", "eintausend") Case 2: arrArt = Array("millionen", "einemillion") Case 3: arrArt = Array("milliarden", "einemilliarde") Case 4: arrArt = Array("billionen", "einebillion") End Select If Right(CStr(dblZahl), 3) = "000" Then strWert = Part(strTmp) & arrArt(0) ElseIf CInt(strTmp) = 1 Then strWert = arrArt(1) & strWert Else strWert = Part(strTmp) & arrArt(0) & strWert End If Next intCounter ZahlWort = strWert & strSuffix End Function Private Function Part(strPart As String) As String Dim arrA As Variant, arrB As Variant, arrC As Variant Dim strTmp As String arrA = Array("null", "eins", "zwei", "drei", "vier", "fünf", "sechs", "sieben", "acht", "neun") arrB = Array("elf", "zwölf", "dreizehn", "vierzehn", "fünfzehn", "sechzehn", "siebzehn", "achtzehn", "neunzehn") arrC = Array("zehn", "zwanzig", "dreißig", "vierzig", "fünfzig", "sechzig", "siebzig", "achtzig", "neunzig") If Len(strPart) = 1 Then strTmp = arrA(CInt(Right(strPart, 1))) ElseIf Right(strPart, 2) = "00" Then If Left(strPart, 1) = "1" Then Part = "einhundert" Else Part = arrA(CInt(Left(strPart, 1))) & "hundert" End If Exit Function ElseIf Mid(strPart, Len(strPart) - 1, 1) = "0" Then strTmp = arrA(CInt(Right(strPart, 1))) ElseIf Mid(strPart, Len(strPart) - 1, 1) = "1" Then If CInt(Right(strPart, 1)) <> 0 Then strTmp = arrB(CInt(Right(strPart, 2)) - 11) Else strTmp = arrC(CInt(Mid(strPart, Len(strPart) - 1, 1)) - 1) End If ElseIf CInt(Mid(strPart, Len(strPart) - 1, 1)) > 1 Then Select Case Right(strPart, 1) Case "0" strTmp = arrC(CInt(Mid(strPart, Len(strPart) - 1, 1)) - 1) Case "1" strTmp = "einund" & _<br arrc(cint(mid(strpart, -="" 1))="" 1)<br="" 1,="" len(strpart)=""></br arrc(cint(mid(strpart,> Case Else strTmp = arrA(CInt(Right(strPart, 1))) & "und" & _<br arrc(cint(mid(strpart, -="" 1))="" 1)<br="" 1,="" len(strpart)=""></br arrc(cint(mid(strpart,> End Select End If If Len(strPart) = 3 Then Select Case Left(strPart, 1) Case "0" Case "1" strTmp = "einhundert" & strTmp Case Else strTmp = arrA(CInt(Left(strPart, 1))) & "hundert" & strTmp End Select End If Part = strTmp End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabe in Tabellenblatt-TextBox begrenzen

Problem:	Nach der Eingabe von 3 Zeichen in die TextBox sollen diese in die erste freie Zelle in Spalte A geschrieben werden. Danach ist die TextBox zu leeren, wobei sie wieder den Focus bekommt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub TextBox1_Change() Dim iRow As Integer iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 If TextBox1.TextLength = 3 Then Cells(iRow, 1).Value = TextBox1.Text TextBox1.Text = "" TextBox1.Activate End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datum aus TextBox an erste frei Stelle eintragen

Problem: Ein Datum aus einer UserForm-TextBox soll bei aktivierter CheckBox an die erste freie Stelle einer Datumsreihe eingefügt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmInsertDate

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdEintragen_Click()
    Dim rng As Range
    If chbInsert.Value = False Then Exit Sub
    If Not IsDate(CDate(txtDate.Text)) Then
        MsgBox "Kein gültiges Datum!"
        txtDate.SetFocus
        Exit Sub
    End If
    For Each rng In Worksheets("Data").Range("A5:A5,D6:D12").Cells
        If IsEmpty(rng) Then
            rng = CDate(txtDate.Text)
            End
        End If
    Next rng
End Sub

StandardModule: Modul1

Sub CallForm()
    frmInsertDate.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Markierung einer Datumsreihe gem. UserForm-TextBoxes

Problem: Eine sich aus UserForm-TextBoxes ergebende Datumsreihe soll markiert werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDateInsert

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim rngStart As Range, rngEnd As Range
    Set rngStart = Columns(1).Find( _
        what:=DateValue(txtStart.Text), LookIn:=xlFormulas)
    Set rngEnd = Columns(1).Find( _
        what:=DateValue(txtEnd.Text), LookIn:=xlFormulas)
    Columns("B").Clear
    Range(rngStart, rngEnd).Offset(0, 1).Value = "X"
End Sub

StandardModule: basMain

Sub CallForm()
    frmDateInsert.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eigene Klasse für UserForm-TextBoxes bilden

Problem:	Bei einer UserForm mit 4 TextBoxes soll bei der Eingabe des 8. Zeichens geprüft werden, ob es sich um ein gültiges Datum handelt. Alle TextBoxes sind gemeinsam über ein Change-Ereignis anzusprechen.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdDialogAufruf_Click() frmSampleClass.Show End Sub ClassModule: frmSampleClass Dim txtBoxes(1 To 4) As New Klasse1 Private Sub cmdCancel_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim intCounter As Integer For intCounter = 1 To 4 Set txtBoxes(intCounter).TxtGroup = Controls("TextBox" & intCounter) Next intCounter End Sub ClassModule: Klasse1 Public WithEvents TxtGroup As MSForms.TextBox Private Sub TxtGroup_Change() With TxtGroup If Len(.Text) < 8 Then Exit Sub If Not IsDate(.Text) Then Beep MsgBox "Bitte gültiges Datum eingeben!" .SelStart = 0 .SelLength = .TextLength End If End With End Sub StandardModule: basMain Sub CallForm() frmSampleClass.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingetragenen Wert in TextBox speichern

Problem:	Wie kann ich erreichen, daß der in einer UserForm-TextBox eingetragene Wert beim nächsten Aufruf der UserForm wieder angezeigt wird?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmSetValue Private Sub cmdCancel_Click() gvar = txtB.Text Unload Me End Sub StandardModule: basMain Public gvar As Variant Sub CallForm() frmSetValue.Show With ThisWorkbook.VBProject _ .VBComponents("frmSetValue").Designer .Controls("txtA").Value = gvar End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datumreihe aus TextBox-Einträgen generieren

Problem:	Wie kann ich aus TextBox-Einträgen (Start- und Enddatum) in einer UserForm in einer Tabelle eine Zahlenreihe generieren?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmDate Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim datStart As Date, datEnde As Date Dim iRow As Integer datStart = DateValue(txtStart.Text) datEnde = DateValue(txtEnde.Text) For iRow = 1 To datEnde - datStart + 1 Cells(iRow, 1) = datStart + iRow - 1 Next iRow Unload Me End Sub StandardModule: basMain Sub CallForm() frmDate.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Focus und Selection zwischen zwei TextBoxes wechseln

Problem:	Der Focus soll nach Eingaben zwischen zwei TextBoxes wechseln. Die Einträge sollen selektiert erscheinen.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() frmFocus.Show End Sub ClassModule: frmFocus Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub txtA_AfterUpdate() With txtB .SetFocus .SelStart = 0 .SelLength = Len(.Text) End With End Sub Private Sub txtB_AfterUpdate() txtA.SetFocus With txtA .SetFocus .SelStart = 0 .SelLength = Len(.Text) End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabebegrenzung und keine Zifferneingabe

Problem: Die Eingabelänge einer Tabellenblatt-TextBox soll begrenzt werden. Desweiteren sollen nur Ziffern zugelassen werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der Tabelle ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub TextBox1_Change()
    Dim sTxt As String
    sTxt = TextBox1.Text
    If Right(sTxt, 1) Like "[!0-9]" = False Then
        MsgBox "Nur Buchstaben erlaubt!"
        TextBox1.Text = Left(sTxt, Len(sTxt) - 1)
    End If
End Sub

StandardModule: Modul1

Sub TextPostanschrift_BeiÄnderung()

Application.ScreenUpdating = False
dim PostS As String
Dim EingabeText As String
EingabeText = PostS.Text

If IsNumeric(EingabeText) Then
MsgBox "Nur Text ist erlaubt!"
PostS.Text = EingabeNumeric
If Numeric > 0 Then
Text = Left$(Eingabe, Len(EingabeNumeric) - 1)
Else

If Len(PostS) > 3 Then
MsgBox "Text zu Lang!"

End If
End If
End If
Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalteninhalt in UserForm-TextBox mit ScrollBalken einlesen

Problem:	Der Inhalt aus Spalte A soll in eine UserForm-TextBox mit vertikalen Scroll-Balken eingelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmText Private Sub cmdCancel_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim lngRow As Long Dim strTxt As String lngRow = 1 Do Until IsEmpty(Cells(lngRow, 1)) strTxt = strTxt & Cells(lngRow, 1) & vbCrLf lngRow = lngRow + 1 Loop txt.Text = Left(strTxt, Len(strTxt) - 1) End Sub StandardModule: basMain Sub CallForm() frmText.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriff abfragen und Fundzeile in UserForm-TextBoxes

Problem: Ein Suchbegriff soll abgefragt und in der Tabelle gesucht werden. Die Daten der Fundzeile sind in UserForm-TextBoxes einzulesen und anzuzeigen.

Lösung: Den nachfolgenden Code in die benannten Module einfügen.

VBA-Code:

```
ClassModule: frmTexte

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    Dim rng As Range
    Dim iCounter As Integer
    Dim sSearch As String
    sSearch = InputBox("Suchbegriff:", , "Zeile 5 - Spalte 3")
    If sSearch = "" Then Exit Sub
    Set rng = Cells.Find( _
        what:=sSearch, lookat:=xlWhole, LookIn:=xlValues)
    If rng Is Nothing Then
        Beep
        MsgBox "Suchbegriff wurde nicht gefunden!"
        Exit Sub
    End If
    For iCounter = 1 To 6
        frmTexte.Controls("TextBox" & iCounter).Text = _
            Cells(rng.Row, iCounter)
    Next iCounter
    frmTexte.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabebegrenzungen für TextBoxes

Problem: Wie kann ich in 3 UserForm-TextBoxes verschiedene, wertmäßige Eingabebegrenzungen festlegen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmTxtCntr

Private Sub cmdOK_Click()
    Range("D10").Value = CDBl(txtC.Text)
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub txtA_Change()
    If txtA.Text = "" Then Exit Sub
    If Right(txtA.Text, 1) Like "[!0-9]" Then
        txtA.Text = Left(txtA.Text, txtA.TextLength - 1)
    End If
    If txtA.TextLength = 2 Then
        If CInt(txtA.Text) >= 20 And CInt(txtA.Text) <= 50 Then
            txtB.SetFocus
        Else
            Beep
            MsgBox "Nur Ganzzahlen zwischen 20 und 50 erlaubt!"
            With txtA
                .SelStart = 0
                .SelLength = .TextLength
            End With
        End If
    End If
End Sub

Private Sub txtB_Change()
    If txtB.Text = "" Then Exit Sub
    If IsNumeric(txtB.Text) = False Then
        txtB.Text = Left(txtB.Text, txtB.TextLength - 1)
    End If
    If txtB.TextLength = 4 Then
        If CDBl(txtB.Text) >= 0.5 And CInt(txtB.Text) <= 30 Then
            txtC.Text = (CInt(txtA.Text) + CDBl(txtB.Text)) ^ 5 * 0.25
            cmdOK.Visible = True
            cmdOK.SetFocus
        Else
            Beep
            MsgBox "Nur Zahlen zwischen 0,5 und 30 erlaubt!"
            With txtB
                .SelStart = 0
                .SelLength = .TextLength
            End With
        End If
    End If
End Sub

StandardModule: basMain

Sub CallForm()
    frmTxtCntr.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblatt-TextBox-Werte in Abhängigkeit zu Drehfeld

Problem:	Die Werte in den TextBoxes sollen sich in Abhängigkeit zum Wert des Drehfeldes mit den Zellinhalten aus den Spalten A:C füllen.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus UserForm-TextBox suchen und Fundstelle kopieren

Problem: Eine in eine UserForm-TextBox einzugebende Zahl soll in Spalte 1 gesucht und die Fundzeile bis Spalte R in eine neue Arbeitsmappe kopiert werden. Die Tabellen 1 bis 5 sind zu durchsuchen.

Lösung: Den nachfolgenden Code in die benannten Module einfügen.

VBA-Code:

```
ClassModule: frmSearch

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdSearch_Click()
    If txtSearch.Text = "" Then Exit Sub
    Call MultiSuche(CLng(txtSearch.Text))
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmSearch.Show
End Sub

Sub MultiSuche(strSearch As Long)
    Dim wks As Worksheet
    Dim rng As Range
    Dim iRow As Integer
    Dim sFind As String
    Workbooks.Add
    For Each wks In ThisWorkbook.Worksheets
        Set rng = wks.Cells.Find(strSearch)
        If Not rng Is Nothing Then
            sFind = rng.Address
            Do
                iRow = iRow + 1
                wks.Range(wks.Cells(rng.Row, 1), _
                    wks.Cells(rng.Row, 18)).Copy _
                    Cells(iRow, 1)
                Set rng = wks.Cells.FindNext(After:=rng)
                If rng.Address = sFind Then Exit Do
            Loop
        End If
    Next wks
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wörter eines Satzes in Indizliste suchen

Problem:	Jedes Wort einer Supportfrage soll in einer Indizliste gesucht und es soll die entsprechende Antwort angezeigt werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() frmSearch.Show End Sub ClassModule: frmSearch Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdSearch_Click() Dim rng As Range Dim var As Variant Dim arr() As Integer Dim iRow As Integer, iCounter As Integer Dim sWord As String, sTxt As String Dim bln As Boolean sTxt = txtSearch.Text If Right(sTxt, 1) = "?" Or Right(sTxt, 1) = "." Then _ sTxt = Left(sTxt, Len(sTxt) - 1) ReDim arr(1 To 1) Worksheets("Data").Rows("1:" & Rows.Count).Hidden = False Do Until sTxt = "" If InStr(sTxt, " ") Then sWord = Left(sTxt, InStr(sTxt, " ") - 1) sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, " ")) Else sWord = sTxt End If iRow = 2 With Worksheets("Data") Do Until IsEmpty(.Cells(iRow, 1)) Set rng = .Range(.Cells(iRow, 3), _ .Cells(iRow, 256)) _ .Find(sWord, lookat:=xlPart, _ LookIn:=xlValues) If Not rng Is Nothing Then var = Application.Match(iRow, arr, 0) If IsError(var) Then iCounter = iCounter + 1 ReDim Preserve arr(1 To iCounter) arr(iCounter) = iRow End If End If End If iRow = iRow + 1 End With Loop If InStr(sTxt, " ") = False Then If bln Then Exit Do Else bln = True End If Loop Worksheets("Data").Rows("2:" & Rows.Count).Hidden = True For iRow = 1 To iCounter Worksheets("Data").Rows(arr(iRow)).Hidden = False Next iRow Worksheets("Data").Select Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Plausibilisttsprfung bei Datum&Zeit-Eingaben in UserForm-TextBox

Problem: Wie kann ich prfen, ob Datum&Zeit-Eingaben in eine UerForm-TextBox fon der Syntax her korrekt sind?

Lsung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDatumZeit

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub TextBox1_BeforeUpdate( _
    ByVal Cancel As MSForms.ReturnBoolean)
    If Len(TextBox1.Text) < 16 Then Cancel = True
End Sub

Private Sub TextBox1_Change()
    If Len(TextBox1.Text) = 16 Then
        If IsDate(TextBox1.Text) = False Then
            MsgBox "Keine gltige Eingabe!"
        Else
            MsgBox "Datum: " & Left(TextBox1.Text, 10) & _
                vbLf & "Zeit: " & _
                Right(TextBox1.Text, 5)
        End If
    End If
End Sub

Private Sub TextBox1_KeyPress( _
    ByVal KeyAscii As MSForms.ReturnInteger)
    If Chr(KeyAscii) Like "[0-9.: ]" = False Then
        KeyAscii = 0
    End If
End Sub

StandardModule: basMain

Sub CallForm()
    frmDatumZeit.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe schliessen, wenn kein Eintrag in TextBox erfolgt

Problem: Wie kann ich eine Arbeitsmappe automatisch schließen lassen, wenn der Anwender 1 Minute lang keinen Eintrag in einer UserForm-TextBox tätigt?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: modMain

Public NextTime As Date, dteStart As Date

Sub Updateclock()
    NextTime = Now + TimeValue("00:00:10")
    Application.OnTime NextTime, "Updateclock"
    If Time - dteStart > TimeSerial(0, 1, 0) Then Call StopClock
End Sub

Sub StopClock()
    Application.OnTime NextTime, "Updateclock", , False
    Unload frmAutomatic
    ThisWorkbook.Close savechanges:=True
End Sub

Sub CallForm()
    frmAutomatic.Show
End Sub

ClassModule: frmAutomatic

Private Sub txtEingabe_KeyUp( _
    ByVal KeyCode As MSForms.ReturnInteger, _
    ByVal Shift As Integer)
    dteStart = Time
End Sub

Private Sub UserForm_Activate()
    dteStart = Time
    Call Updateclock
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Autofilter-Kriterien über UserForm vorgeben

Problem: Wie kann ich Autofilter-Kriterien über UserForm-TextBox-Einträge vorgeben?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmFilter

Private Sub cmdFilter_Click()
    If ActiveSheet.FilterMode = False Then
        ActiveSheet.FilterMode = True
    End If
    Range("A1").AutoFilter _
        field:=1, _
        Criteria1:="=" & txtFilter.Text, _
        Operator:=xlOr
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmFilter.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Passwortabfrage mit ShortCut-Aufruf

Problem:	Wie kann ich mit der Tastenkombination Strg+w einen Dialog zur Passwordeingabe aufrufen? Bei Eingabe des richtigen Passwortes soll bei allen Arbeitsblättern der Blattschutz aufgehoben werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmPassword Private Sub txtPassword_Change() Dim wks As Worksheet If Len(txtPassword.Text) = 8 Then If txtPassword = "Passwort" Then For Each wks In Worksheets wks.Unprotect "Passwort" Next wks Else MsgBox "War wohl nix!" txtPassword.Text = "" End If End If End Sub StandardModule: basMain Sub UserFormStart() frmPassword.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichen in UserForm-TextBox durch ScrollBar-Schieber markieren

Problem: Wie kann ich die einzelnen Zeichen einer UserForm-TextBox entsprechend den gewählten ScrollBar-Werten markieren lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSelect

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub scbSelect_Change()
    scbSelect.SetFocus
    txtSelect.SelStart = scbSelect.Value
    txtSelect.SelLength = 1
    txtSelect.SetFocus
End Sub

Private Sub UserForm_Initialize()
    scbSelect.Max = Len(txtSelect.Text) - 1
End Sub

StandardModule: basMain

Sub CallForm()
    With frmSelect
        With .txtSelect
            .SetFocus
            .SelStart = 0
            .SelLength = 1
        End With
        .Show
    End With
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text aus letzter Zelle mit Inhalt in UserForm-TextBox anzeigen

Problem:	Wie kann ich mir in einer UserForm-TextBox beim Dialogaufruf den letzten in Spalte A eingetragenen Wert anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmLastValue Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub UserForm_Initialize() txtLastValue.Text = Cells(_ Cells(Rows.Count, 1).End(xlUp).Row, 1).Text End Sub StandardModule: basMain Sub CallForm() frmLastValue.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text aus UserForm-TextBox in Tabelle suchen

Problem: Wie kann ich Werte aus einer UserForm-TextBox in einem Tabellenblatt suchen und die Fundzeile in UserForm-Labels ausgeben lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSuchen

Private Sub cmdSuchen_Click()
    Dim rng As Range
    Dim dValue As Double
    Dim iCounter As Integer
    If Not IsNumeric(txtWert.Value) Then
        MsgBox "Nur Zahlen erlaubt!"
        Exit Sub
    End If
    dValue = CDbl(txtWert.Value)
    Set rng = Columns("A").Find(dValue, _
        lookat:=xlWhole, LookIn:=xlValues)
    If rng Is Nothing Then
        MsgBox "Suchwert wurde nicht gefunden!"
        For iCounter = 6 To 9
            Controls("Label" & iCounter).Caption = ""
        Next iCounter
    Else
        For iCounter = 0 To 3
            Controls("Label" & iCounter + 6).Caption = _
                rng.Offset(0, iCounter)
        Next iCounter
    End If
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmSuchen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-TextBox-Werte auf Numeric prüfen und in Tabelle eintragen

Problem: Wie kann ich UserForm-TextBox-Werte darauf überprüfen, ob es sich um Zahlen handelt und bei Zutreffen in Tabellen eintragen. Handelt es sich nicht um eine Zahl, soll eine entsprechende Meldung ausgegeben und der Wert in der TextBox markiert werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmPruefen

Private Sub cmdEintragen_Click()
    Dim iCounter As Integer
    For iCounter = 1 To 3
        With Controls("TextBox" & iCounter)
            If IsNumeric(.Text) = False Then
                MsgBox "Bitte Wert in TextBox" & iCounter & " berichtigen!"
                .SetFocus
                .SelStart = 0
                .SelLength = Len(.Text)
            End If
        End With
    Next iCounter
    For iCounter = 1 To 3
        Cells(1, iCounter) = CDBl(Controls("TextBox" & iCounter).Text)
    Next iCounter
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmPruefen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-TextBox-Werte einer ListBox hinzufügen

Problem:	Nach der Eingabe von 2 Ziffern in eine UserForm-TextBox soll der Wert in eine ListBox eingetragen werden und der Cursor zur nächsten TextBox springen.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmTextToList Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub txtBox1_Change() Call Eintragenlistbox(1) End Sub Private Sub txtBox2_Change() Call Eintragenlistbox(2) End Sub Private Sub txtBox3_Change() Call Eintragenlistbox(3) End Sub Private Sub Eintragenlistbox(intNo As Integer) If Len(Controls("txtBox" & intNo)) <> 2 Then Exit Sub Controls("lstBox" & intNo).AddItem Controls("txtBox" & intNo).Text Select Case intNo Case 1: txtBox2.SetFocus Case 2: txtBox3.SetFocus Case 3: txtBox1.SetFocus End Select Controls("txtBox" & intNo).Text = "" End Sub StandardModule: basMain Sub CallForm() frmTextToList.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Im Textfeld nur Eingabe von 2 Ziffern zulassen, dann zum nächsten

Problem: Wie kann ich erreichen, dass der Cursor nach der Eingabe von 2 Ziffern in eine TextBox zur nächsten springt. Der Wert soll beim Verlassen der jeweiligen TextBox in der dem TextBox-Wert entsprechenden Spalte eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub TextBox1_Change()
    If Len(TextBox1.Text) <> 2 Then Exit Sub
    Call EintragenWert(Val(TextBox1.Text), 5)
    TextBox2.Activate
End Sub

Private Sub TextBox2_Change()
    If Len(TextBox2.Text) <> 2 Then Exit Sub
    Call EintragenWert(Val(TextBox2.Text), 6)
    TextBox3.Activate
End Sub

Private Sub TextBox3_Change()
    If Len(TextBox3.Text) <> 2 Then Exit Sub
    Call EintragenWert(Val(TextBox3.Text), 7)
    TextBox1.Activate
End Sub

Sub EintragenWert(iValue As Integer, iBox As Integer)
    ActiveCell.Select
    Cells(iBox, iValue) = iValue
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabulator in UserForm-TextBox simulieren

Problem:	In einer UserForm-TextBox kann kein Tabulator eingesetzt werden. Kann ich dieses Problem umgehen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmTabulator Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub spnTabulator_Change() txtSpin.Value = spnTabulator.Value End Sub Private Sub txtTabulator_KeyDown(ByVal _ KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer) If KeyCode = 9 Then KeyCode = 0 txtTabulator.Text = txtTabulator.Text & _ String(spnTabulator.Value, " ") End If End Sub StandardModule: basMain Sub CallForm() frmTabulator.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte zweiter UserForm-TextBoxes multiplizieren

Problem: Wie kann ich die Werte zweier UserForm-TextBoxes multipliziert in einer dritten TextBox ausgeben?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmBerechnen

Private Sub cmdBerechnen_Click()
    txtSumme.Text = CDbl(txtA.Text) * CDbl(txtB.Text)
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmBerechnen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abfrage zum Kopierbereich von Startzelle, Zeilen- und Spaltenanzahl

Problem: Wie kann ich über eine UserForm einen Kopierbereich festlegen? Abgefragt werden sollen Startzelle, Zeilen- und Spaltenanzahl.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdDialogAufruf_Click()
    frmAuswahl.Show
End Sub

ClassModule: frmAuswahl

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim rngSource As Range, rngTarget As Range, rngStart As Range
    Set rngStart = Worksheets("Tabelle1").Range(txtStart.Text)
    Set rngSource = Worksheets("Tabelle1").Range _
        (Cells(rngStart.Row, rngStart.Column), _
        Cells(rngStart.Row + txtRows.Text - 1, _
        rngStart.Column + txtColumns.Text - 1))
    rngSource.Copy Worksheets("Tabelle2").Range("A1")
    Worksheets("Tabelle2").Select
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmAuswahl.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neues Tabellenblatt und Name über UserForm ermitteln

Problem: Wie kann ich der aktiven Arbeitsmappe ein neues Tabellenblatt hinzufügen und dessen Name in einer UserForm festlegen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmNeuesBlatt

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdNeuesBlatt_Click()
    Dim wks As Worksheet
    On Error Resume Next
    Set wks = Worksheets(txtNeuesBlatt.Text)
    If Err > 0 Or wks Is Nothing Then
        Worksheets.Add after:=Worksheets(Worksheets.Count)
        ActiveSheet.Name = txtNeuesBlatt.Text
        Worksheets(2).Select
    Else
        Beep
        MsgBox "Blatt besteht schon!"
    End If
    On Error GoTo 0
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmNeuesBlatt.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textvariable in eine UserForm-TextBox eintragen

Problem: Wie kann ich eine öffentliche Textvariable über eine Schaltfläche in eine UserForm-Textbox eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmTest

Private Sub cmdTest_Click()
    txtTest.Text = sTest
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Public sTest As String

Sub CallForm()
    sTest = "Hallo"
    frmTest.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bestimmte Zeichen in einer UserForm-TextBox markieren

Problem: Wie kann ich bestimmte Zeichen in einer UserForm-Textbox so markieren, daß sie überschrieben werden können?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMarkieren

Private Sub cmdMarkieren_Click()
    With txtMarkieren
        .SetFocus
        .SelStart = 8
        .SelLength = 5
    End With
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmMarkieren.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In einer UserForm-TextBox die Eingabe von Sonderzeichen verhindern

Problem:	Wie kann ich in einer UserForm-TextBox die Eingabe von Sonderzeichen verhindern?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmTextPruefung Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub txtPruefung_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger) If Chr(KeyAscii) Like "[0-9 a-z A-Z]" = False Or _ KeyAscii = 32 Then Beep MsgBox "Nur Zahlen und Buchstaben erlaubt!" KeyAscii = 0 End If End Sub StandardModule: basMain Sub CallForm() frmTextPruefung.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus UserForm-TextBox und aus Tabelle gerundet anzeigen

Problem: Wie kann ich Werte aus einer UserForm-TextBox und aus einer Tabelle in einer weiteren UserForm-TextBox auf zwei Stellen gerundet anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmRunden

Private Sub cmdAusTabelle_Click()
    txtRunden.Value = WorksheetFunction.Round( _
        Range("A1").Value, 2)
End Sub

Private Sub cmdAusTextBox_Click()
    txtRunden.Value = WorksheetFunction.Round( _
        CDb1(txtBasis.Value), 2)
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmRunden.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriff aus einer UserForm-Textbox in Tabelle suchen und melden

Problem: Wie kann ich einen Suchbegriff aus einer UserForm-TextBox in Spalte A einer Tabelle suchen lassen und die Fundstelle durch eine MsgBox anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSuchen

Private Sub cmdSuchen_Click()
    Dim rng As Range
    Set rng = Columns(1).Find( _
        what:=txtSuchbegriff.Text, _
        lookat:=xlPart, _
        LookIn:=xlFormulas)
    If rng Is Nothing Then
        MsgBox "Suchbegriff wurde nicht gefunden!"
    Else
        MsgBox "Suchbegriff wurde in Zeile" & rng.Row & " gefunden!"
    End If
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmSuchen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Paßwortabfrage mit UserForm

Problem: Wie kann ich über eine UserForm ein Paßwort abfragen und je nach Wahrheitsgehalt der Antwort die Steuerung vornehmen? Das Paßwort darf nicht angezeigt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmPasswort

Private Sub cmdAbbrechen_Click()
    MsgBox "Kein Zugang!"
    Unload Me
End Sub

Private Sub cmdOK_Click()
    If txtPasswort.Text = "Paßwort" Then
        MsgBox "Alles klar!"
        Unload Me
    Else
        MsgBox "War wohl nix!"
        txtPasswort.Text = ""
        txtPasswort.SetFocus
    End If
End Sub

Private Sub UserForm_Initialize()
    txtPasswort.SetFocus
End Sub

StandardModule: basMain

Sub CallForm()
    frmPasswort.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus UserForm-TextBox in Tabelle suchen und in ListBox listen

Problem: Wie kann ich einen in einer UserForm-TextBox eingegebenen Wert in einem Tabellenblattbereich suchen und alle Fundwerte in einer ListBox der UserForm auflisten lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSuchen

Private Sub cmdSuchen_Click()
    Dim rngFind As Range, rngFirst As Range
    lstFind.Clear
    Set rngFind = ActiveSheet.Cells.Find( _
        what:=txtSuchen.Text, _
        lookat:=xlPart, _
        LookIn:=xlValues)
    If rngFind Is Nothing Then
        Beep
        MsgBox "Kein Suchbegriff gefunden!"
        Exit Sub
    End If
    Set rngFirst = rngFind
    Do
        lstFind.AddItem rngFind
        Set rngFind = ActiveSheet.Cells.FindNext(rngFind)
    Loop While Not rngFind Is Nothing And _
        rngFind.Address <> rngFirst.Address
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmSuchen.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Warnung bei Erreichung eines Minuswertes in UserForm

Problem: In einer UserForm lasse ich in einer TextBox Werte - auch Minuswerte - addieren. Wie kann ich eine Warnmeldung ausgeben lassen, wenn in dieser TextBox ein Minuswert erreicht wird?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAddieren

Private Sub cmdAddieren_Click()
    txtSumme.Value = CDbl(txtSumme.Text) + CDbl(txtEingabe.Text)
    If CDbl(txtSumme.Text) < 0 Then
        Beep
        MsgBox "Summenwerte unter 0 nicht erlaubt!"
        txtSumme.Value = 0
    End If
    txtEingabe.Value = ""
    txtEingabe.SetFocus
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub cmdZurueck_Click()
    txtSumme.Value = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmAddieren.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus einer UserForm-Textfeld-Serie in Arbeitsblatt

Problem: Wie kann ich die Werte aus einer Serie von UserForm-Textfeldern in die Spalte eines Tabellenblattes übertragen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmTextUebertragen

Private Sub cmdUebertragen_Click()
    Dim cnt As Control
    Dim iRow As Integer
    iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
    For Each cnt In Controls
        If TypeName(cnt) = "TextBox" Then
            Cells(iRow, 1).Value = cnt.Text
            iRow = iRow + 1
        End If
    Next cnt
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmTextUebertragen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-TextBoxes ein- und ausblenden

Problem: Wie kann ich UserForm-TextBoxes durch Schaltflächendruck ein- und ausblenden?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAusEin

Private Sub cmdAlle_Click()
    Dim iCounter As Integer
    If cmdAlle.Caption = "Alle Namen ausblenden" Then
        For iCounter = 0 To 7
            Controls(iCounter).Visible = False
        Next iCounter
        cmdAlle.Caption = "Alle Namen einblenden"
    Else
        For iCounter = 0 To 7
            Controls(iCounter).Visible = True
        Next iCounter
        cmdAlle.Caption = "Alle Namen ausblenden"
    End If
End Sub

Private Sub cmdEinsUndZwei_Click()
    If cmdEinsUndZwei.Caption = "Namen 1 und 2 ausblenden" Then
        lblName1.Visible = False
        lblName2.Visible = False
        txtName1.Visible = False
        txtName2.Visible = False
        cmdEinsUndZwei.Caption = "Namen 1 und 2 einblenden"
    Else
        lblName1.Visible = True
        lblName2.Visible = True
        txtName1.Visible = True
        txtName2.Visible = True
        cmdEinsUndZwei.Caption = "Namen 1 und 2 ausblenden"
    End If
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmAusEin.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In einer UserForm Berechnungen aufaddieren

Problem: Wie kann ich in einer UserForm-TextBox die in einer anderen TextBox berechneten Werte aufaddieren lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmBerechnen

Private Sub cmdBerechnen_Click()
    Dim dValueA As Double, dValueB As Double
    Dim dBerechnung As Double, dTotal As Double
    dValueA = CDbl(txtWert1.Text)
    dValueB = CDbl(txtWert2.Text)
    txtBerechnung.Value = dValueA + dValueB
    dBerechnung = CDbl(txtBerechnung.Text)
    If txtTotal.Text <> "" Then
        dTotal = CDbl(txtTotal.Text)
    Else
        dTotal = 0
    End If
    txtTotal.Value = dTotal + dBerechnung
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmBerechnen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus UserForm-TextBox in bestimmten Formate übertragen

Problem: Wie kann ich in einer TextBox einer UserForm eingegebene Werte in von Optionsfeldern abhängigen Formaten in eine Zelle übertragen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmFormatAuswahl

Private Sub cmdEintragen_Click()
    Dim dValue As Double
    If optWaehrung.Value = True Then
        Range("A1").NumberFormat = "#,##0.00 $"
    ElseIf optKolonne.Value = True Then
        Range("A1").NumberFormat = "#,##0.00"
    ElseIf optSingle.Value = True Then
        Range("A1").NumberFormat = "0"
    Else
        Range("A1").NumberFormat = "0.00%"
    End If
    dValue = CDbl(txtWert.Text)
    If optProzent.Value = True Then dValue = dValue / 100
    Range("A1") = dValue
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmFormatAuswahl.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktuelle Zeit über UserForm eintragen und Rennzeit berechnen

Problem: Wie kann ich über eine UserForm eine Startnummer auswählen, in der entsprechenden Tabellenzeile die aktuelle Zeit eintragen und die Rennzeit berechnen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmRennZeit

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdEintragen_Click()
    Cells(cboNummer.ListIndex + 1, 3) = Now
End Sub

Private Sub UserForm_Initialize()
    cboNummer.ListIndex = 1
End Sub

StandardModule: basMain

Sub CallForm()
    frmRennZeit.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit TextBox-Werten in UserForm rechnen

Problem: Wie kann ich Werte aus mehreren TextBoxes mit einem in einer weiteren TextBox stehenden Faktor multiplizieren und das Ergebnis in einer TextBox anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdDialogAufruf_Click()
    frmBerechnen.Show
End Sub

ClassModule: frmBerechnen

Private Sub cmdBerechnen_Click()
    Dim cnt As Control
    Dim dValue As Double
    For Each cnt In Controls
        If Left(cnt.Name, 1) = "S" Then
            dValue = dValue + CDBl(cnt.Text)
        End If
    Next cnt
    Controls("Total").Text = dValue * Controls("Faktor").Value
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmBerechnen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vereinfachte Datumseingabe über UserForm

Problem:	Wie kann ich Datumsangaben über eine UserForm möglichst vereinfachen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmDate Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub txtDatum_Change() Dim dat As Date Dim iRow As Integer Dim sTxt As String sTxt = txtDatum.Text If sTxt = "" Then Exit Sub If Right(sTxt, 1) Like "[0-9]" = False Then Beep sTxt = Left(sTxt, Len(sTxt) - 1) txtDatum.Text = sTxt End If If Len(sTxt) = 4 Then On Error Resume Next dat = DateSerial(Year(Date), Right(sTxt, 2), Left(sTxt, 2)) If Err > 0 Then MsgBox "Falsches Datum!" Err = 0 txtDatum.Text = "" Exit Sub End If iRow = WorksheetFunction.CountA(Columns(1)) + 1 Cells(iRow, 1) = dat txtDatum.Text = "" End If End Sub StandardModule: basMain Sub CallForm() frmDate.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabekontrolle bei UserForm

Problem: Wie kann ich die Eingaben in einer UserForm-TextBox auf Ziffern, Groß- oder Kleinbuchstaben begrenzen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmEingabekontrolle

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub TextBox1_Change()
    Dim sTxt As String
    sTxt = TextBox1.Text
    If Len(sTxt) = 0 Then Exit Sub
    If CheckBox1.Value = True And Right(sTxt, 1) _
        Like "[0-9]" Then Exit Sub
    If CheckBox2.Value = True And Right(sTxt, 1) _
        Like "[A-Z]" Then Exit Sub
    If CheckBox3.Value = True And Right(sTxt, 1) _
        Like "[a-z]" Then Exit Sub
    TextBox1.Text = Left(sTxt, Len(sTxt) - 1)
End Sub

StandardModule: basMain

Sub CallForm()
    frmEingabekontrolle.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in UserForm-TextBox eintragen und selektieren

Problem: Wie kann ich beim Aufruf einer UserForm in eine TextBox einen Wert eintragen und selektieren lassen, damit dieser gleich überschrieben werden kann?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSelektieren

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    With TextBox1
        .Value = "Januar"
        .SelStart = 0
        .SelLength = .TextLength
    End With
End Sub

StandardModule: basMain

Sub CallForm()
    frmSelektieren.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Umwandlung eines TextBox-Wertes in eine Zahl

Problem: Wenn in einer UserForm-TextBox Zahlen mit Komma als Dezimaltrenner eingegeben werden, wird der Wert bei einer Übernahme auf CommandButton-Befehl als Text eingetragen. Wie kann ich das verhindern?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CommandButton1_Click()
    frmDezimaltrenner.Show
End Sub

ClassModule: frmDezimaltrenner

Private Sub CommandButton1_Click()
    Worksheets("Tabelle1").Range("A1") = _
        WorksheetFunction.Substitute(TextBox1.Value, ",", ".")
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmDezimaltrenner.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bedingte Übergabe des Focus und Setzen der Schreibmarke im Dialog

Problem: Wie kann ich je nach Eingabe in EditBox 1 den Focus an EditBox2 oder EditBox3 übergeben und dort die Schreibmarke an eine bestimmte Stelle setzen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmSelect.Show
End Sub

ClassModule: frmSelect

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub cmdFirst_Click()
    With txtFirst
        .SetFocus
        .SelStart = 3
        .SelLength = 2
    End With
End Sub

Private Sub cmdSecond_Click()
    With txtSecond
        .SetFocus
        .SelStart = 6
        .SelLength = 4
    End With
End Sub

Private Sub UserForm_Initialize()
    txtFirst.Text = Format( _
        DateSerial(Year(Date), Month(Date), 12), "dd.mm.yyyy")
    txtSecond.Text = Format( _
        DateSerial(Year(Date), Month(Date), 25), "dd.mm.yyyy")
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Paßwortabfrage

Problem:	Wie kann ich eine einfache Paßwortabfrage über InputBox starten?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Passwort() Dim sWord As String Do While InputBox("Passwort:") <> "MeinPasswort" Beep If MsgBox("War wohl nix!", vbOKCancel) = vbCancel Then End Loop MsgBox "Paßwort stimmt!" End Sub</pre>



Artikel aus Auswahltextbox in Sammeltextbox verschieben

Problem:	Wie kann ich Eintragungen in der einen Textbox auswählen und in die andere übertragen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	StandardModule: basMain Sub CallForm() frmMonths.Show End Sub ClassModule: frmMonths Private Sub cmdContinue_Click() Unload Me End Sub Private Sub cmdHin_Click() Dim arr() As String Dim var As Variant Dim iRow As Integer If lstTarget.ListCount > 0 Then ReDim arr(1 To lstTarget.ListCount) For iRow = 0 To lstTarget.ListCount - 1 arr(iRow + 1) = lstTarget.List(iRow) Next iRow For iRow = 0 To lstSource.ListCount - 1 If lstSource.Selected(iRow) Then var = Application.Match(lstSource.List(iRow), arr, 0) If IsError(var) Then ReDim Preserve arr(1 To UBound(arr) + 1) arr(UBound(arr)) = lstSource.List(iRow) End If End If Next iRow lstTarget.List = arr Else For iRow = 0 To lstSource.ListCount - 1 If lstSource.Selected(iRow) Then lstTarget.AddItem lstSource.List(iRow) End If Next iRow End If End Sub Private Sub cmdHer_Click() Dim iRow As Integer For iRow = lstTarget.ListCount - 1 To 0 Step -1 If lstTarget.Selected(iRow) Then lstTarget.RemoveItem iRow End If Next iRow End Sub Private Sub cmdAll_Click() lstTarget.List = lstSource.List End Sub Private Sub cmdRemove_Click() lstTarget.Clear End Sub Private Sub lstTarget_DblClick(ByVal Cancel As MSForms.ReturnBoolean) lstTarget.RemoveItem lstTarget.ListIndex End Sub Private Sub lstSource_DblClick(ByVal Cancel As MSForms.ReturnBoolean) Call cmdHin_Click End Sub Private Sub UserForm_Initialize() Dim iMonth As Integer For iMonth = 1 To 12 lstSource.AddItem Format(DateSerial(1, iMonth, 1), "mmm") Next iMonth End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dialog-Datumseingabe vereinfachen und kontrollieren

Problem: Wie kann ich die Datumseingabe im Format TTMMJJJJ in eine DialogEditBox kontrollieren und weiterverarbeiten lassen? Nach der Eingabe von 8 Zeichen soll der Focus an die nächste EditBox weitergegeben werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmDate.Show
End Sub

ClassModule: frmDate

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub txtDate_Change()
    Dim sDate As String
    sDate = txtDate.Text
    If sDate = "" Then Exit Sub
    If Not Right(sDate, 1) Like "[0-9]" Then
        sDate = Left(sDate, Len(sDate) - 1)
        txtDate.Text = sDate
    End If
    If Len(sDate) = 6 Then
        sDate = Format(DateSerial( _
            "20" & Right(sDate, 2), _
            Mid(sDate, 3, 2), _
            Left(sDate, 2) _
            ), "dd.mm.yy")
        txtDate.Text = sDate
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufmaße erfassen und eintragen

Problem: Wie kann ich über einen Dialog Aufmaßdaten eingeben und diese in eine Tabelle eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmAufmass.Show
End Sub

ClassModule: frmAufmass

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim vRow As Variant, vCol As Variant
    vCol = Application.Match(TextBox1.Text, Rows(1), 0)
    If IsError(vCol) Then
        MsgBox "Blatt wurde nicht gefunden!"
        Exit Sub
    End If
    vRow = Application.Match(TextBox2.Text, Columns(1), 0)
    If IsError(vRow) Then
        MsgBox "LV-Position wurde nicht gefunden!"
        Exit Sub
    End If
    Cells(vRow, vCol).Value = TextBox3.Text
End Sub

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_Initialize()
    TextBox1.Value = "Blatt-Nr. 4"
    TextBox2.Value = "LV-Position 12"
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabelänge in TextBox begrenzen und Werte aus Liste übernehmen

Problem: Wie kann ich die Eingabelänge in einer Textbox auf 4 begrenzen? Nach Eingabe des 4. Zeichens soll anhand der Eintragung ein Wert aus einer zweiten Tabelle gesucht und in der ersten Tabelle eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmInsert.Show
End Sub

ClassModule: frmInsert

Private Sub txtInsert_Change()
    Dim vRow As Variant
    Dim iRow As Integer
    If txtInsert.TextLength = 4 Then
        With Worksheets("Datenbank")
            vRow = .Application.Match(txtInsert.Text, .Columns(1), 0)
            If Not IsError(vRow) Then
                iRow = WorksheetFunction.CountA(Columns(1)) + 1
                Cells(iRow, 1).Value = .Cells(vRow, 1).Value
                Cells(iRow, 2).Value = .Cells(vRow, 2).Value
            Else
                Beep
                MsgBox "Wert wurde nicht gefunden!"
            End If
        End With
        With txtInsert
            .SetFocus
            .SelStart = 0
            .SelLength = .TextLength
        End With
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abfrage von Rechenaufgaben zeitgesteuert nach dem Zufallsprizip

Problem:	Wie kann ich zeitgesteuert Rechenaufgaben abfragen. Die Basiszahlen und die Operanden sollen durch das Zufallsprinzip bestimmt und die Ergebnisse sollen ausgewertet werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basMain Public NextTime As Date Sub CallForm() frmRechnen.Show End Sub Sub Verbergen() Worksheets("Memory").Visible = xlVeryHidden End Sub Sub NeueAufgabe() Dim vOperand As Variant Dim iOperand As Integer, iFactorA As Integer, iFactorB As Integer Dim iRowL As Integer Dim sOperand As String Randomize vOperand = Array("+", "-", "*", "/") iOperand = Int((4 * Rnd) + 0) sOperand = vOperand(iOperand) iFactorA = Int((99 * Rnd) + 1) iFactorB = Int((9 * Rnd) + 1) frmRechnen.lblAufgabe.Caption = iFactorA & " " & sOperand & " " & iFactorB Worksheets("Memory").Range("AA1").Formula = "=" & frmRechnen.lblAufgabe.Caption NextTime = Now + TimeSerial(0, 0, frmRechnen.SpinButton1.Value) Application.OnTime _ earliesttime:=NextTime, _ procedure:="NeueAufgabe" End Sub Sub RechnenBeenden() On Error GoTo ERRORHANDLER Application.OnTime _ earliesttime:=NextTime, _ procedure:="NeueAufgabe", _ schedule:=False ERRORHANDLER: With Worksheets("Memory") If WorksheetFunction.CountA(.Columns(1)) = 0 Then Exit Sub MsgBox "Von " & WorksheetFunction.CountA(.Columns(1)) & _ " Aufgaben wurden " & WorksheetFunction.Sum(.Columns(1)) & _ " richtig gelöst." & vbLf & _ "Dies entspricht " & Format(WorksheetFunction.Sum(.Columns(1)) / _ WorksheetFunction.CountA(.Columns(1)), "0.0%") .Columns(1).ClearContents End With End Sub ClassModule: frmRechnen Private Sub cmdContinue_Click() Unload Me Call RechnenBeenden End Sub Private Sub cmdLoesung_Click() Dim iRow As Integer If txtLoesung.Text = "" Then Exit Sub With Worksheets("Memory") iRow = WorksheetFunction.CountA(.Columns(1)) + 1 If CDbl(txtLoesung.Text) = .Range("AA1").Value Then .Cells(iRow, 1).Value = 1 Else .Cells(iRow, 1).Value = 0 End If End With txtLoesung.Text = "" txtLoesung.SetFocus End Sub Private Sub cmdStart_Click() Call NeueAufgabe End Sub Private Sub SpinButton1_Change() Label4.Caption = SpinButton1.Value & " sec." End Sub Private Sub UserForm_Initialize() SpinButton1.Value = 10 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text aus UserForm in eine Zelle schreiben

Problem: Wie kann ich einen TextBox-Wert in einer UserForm während dem Editieren in eine Zelle schreiben lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmEdit

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub txtEdit_Change()
    Range("A1").Value = txtEdit.Text
End Sub

Private Sub UserForm_Initialize()
    txtEdit.Text = Range("A1").Value
End Sub

StandardModule: basMain

Sub CallForm()
    frmEdit.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textfelder in UserForm beim Start vorbelegen

Problem:	Wie kann ich 5 Textfelder in einer UserForm beim Aufruf des Dialogs mit fortlaufenden Werten belegen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmPreSet Private Sub cmdContinue_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim iCounter As Integer For iCounter = 1 To 5 Controls("txt" & iCounter).Text = iCounter Next iCounter End Sub StandardModule: basMain Sub CallForm() frmPreSet.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur AJM in TextBox akzeptieren und Anzahl zählen

Problem: Wie kann ich erreichen, daß eine TextBox in einer UserForm nur die Zeichen A, J und M akzeptiert und die Anzahl dieser Zeichen als Label anzeigt?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmEingabe

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub txtEingabe_Change()
    Dim arr(1 To 3) As Integer
    Dim iCounter As Integer
    Dim sTxt As String
    sTxt = txtEingabe.Text
    If sTxt = "" Then Exit Sub
    If Not Right(sTxt, 1) Like "A" And _
        Not Right(sTxt, 1) Like "J" And _
        Not Right(sTxt, 1) Like "M" Then
        sTxt = Left(sTxt, Len(sTxt) - 1)
        txtEingabe.Text = sTxt
    End If
    For iCounter = 1 To Len(sTxt)
        Select Case Mid(sTxt, iCounter, 1)
            Case "A": arr(1) = arr(1) + 1
            Case "J": arr(2) = arr(2) + 1
            Case "M": arr(3) = arr(3) + 1
        End Select
    Next iCounter
    For iCounter = 1 To 3
        Controls("Label" & iCounter).Caption = _
            "Anzahl Zeichen " & iCounter & ": " & arr(iCounter)
    Next iCounter
End Sub

StandardModule: basMain

Sub CallForm()
    frmEingabe.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

TextBoxes- und Label-Werte in Abhängigkeit verändern

Problem:	Wie kann ich in einer UserForm die Werte von Labels und TextBoxes in Abhängigkeit voneinander verändern?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle11 Private Sub CommandButton1_Click() UserForm1.Show End Sub ClassModule: frmKredit Private Sub cmdContinue_Click() Unload Me End Sub Private Sub txtTilgung_Change() Dim dRate As Double, dTilgung As Double Dim iPreis As Integer If txttilgung.Text = "" Then txtrate.Text = "" Label2.Caption = "" Exit Sub End If iPreis = CInt(Label1.Caption) dRate = CDbl(txttilgung.Text) / 100 txtrate.Text = WorksheetFunction.RoundUp(iPreis * dRate, 2) dTilgung = CDbl(txtrate.Text) Label2.Caption = WorksheetFunction.RoundUp(iPreis / dTilgung, 0) End Sub Private Sub txtRate_Change() Dim iPreis As Integer Dim dTilgung As Double If txtrate.Text = "" Then txtrate.Text = "" Label2.Caption = "" Exit Sub End If iPreis = CInt(Label1.Caption) dTilgung = CDbl(txtrate.Text) txttilgung.Text = WorksheetFunction.RoundUp(dTilgung * 100 / iPreis, 2) Label2.Caption = WorksheetFunction.RoundUp(iPreis / dTilgung, 0) End Sub StandardModule: basMain Sub CallForm() frmKredit.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Termine in Jahrestabelle eintragen

Problem: Wie kann ich Reitstunden mit Angaben über Uhrzeit, Reiter und Pferd in einen Jahreskalender eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmTermin.Show
End Sub

ClassModule: frmTermin

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim vRow As Variant
    Dim iCol As Integer
    If IsDate(txtDate) = False Then
        Beep
        MsgBox "Kein gültiges Datum!"
        Exit Sub
    End If
    vRow = Application.Match(CDate(txtDate.Text), Columns(1), 0)
    If IsError(vRow) Then
        Beep
        MsgBox "Datum wurde nicht gefunden!"
        Exit Sub
    End If
    iCol = WorksheetFunction.CountA(Rows(vRow)) + 1
    Cells(vRow, iCol).Value = txtTime.Text
    Cells(vRow, iCol + 1).Value = txtReiter.Text
    Cells(vRow, iCol + 2).Value = txtGaul.Text
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur Buchstaben in eine UserForm-TextBox

Problem: Wie kann ich die Eingabemöglichkeit in einer UserForm-TextBox auf Buchstaben begrenzen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmBuchstaben

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub txtChars_Change()
    Dim sTxt As String
    If txtChars.Text = "" Then Exit Sub
    sTxt = txtChars.Text
    If Not Right(sTxt, 1) Like "[a-z]" And Not _
        Right(sTxt, 1) Like "[A-Z]" Then
        sTxt = Left(sTxt, Len(sTxt) - 1)
        txtChars.Text = sTxt
    End If
End Sub

StandardModule: basMain

Sub CallForm()
    frmBuchstaben.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text aus UserForm-Textbox in Variable einlesen

Problem: Wie kann ich den Text einer UserForm-Textbox in eine Variable einlesen und weiterverwerten? Die Eingabe in die Textbox soll begrenzt sein.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmVariables

Private Sub txtVariables_Change()
    Dim sVariable As String
    Dim dVariable As Double
    With txtVariables
        If Not Right(.Text, 1) Like "[0-9,]" Then
            .Text = Left(.Text, .TextLength - 1)
        End If
        If Len(.Text) = 5 Then
            sVariable = .Text
            dVariable = CDBl(.Text)
            MsgBox _
                "Text: " & sVariable & vbCrLf & _
                "Wert: " & dVariable * 2
        End If
    End With
End Sub

Private Sub cmdContinue_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmVariables.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nichtnumerische Eingaben in TextBox verhindern

Problem: Wie kann ich die Eingabe von nichtnumerischen Zeichen in eine TextBox einer UserForm verhindern?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
Private Sub txtNumber_Change()  
    If Len(txtNumber.Text) = 0 Then Exit Sub  
    If Not Right(txtNumber, 1) Like "[0-9]" Then  
        Beep  
        MsgBox "Nur Zahlen bitte!"  
        With txtNumber  
            .SetFocus  
            .SelStart = txtNumber.TextLength - 1  
            .SelLength = 1  
        End With  
    End If  
End Sub  
  
StandardModule: basMain  
  
Sub CallForm()  
    frmZahlen.Show  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bestimmte Anzahl von Zeichen zulassen

Problem: Wie kann ich die in eine TextBox einzutragende Zeichenfolge auf 6 Zeichen begrenzen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frm6Zeichen

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub txt6Chars_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    If Len(txt6Chars.Text) < 6 Then
        Beep
        MsgBox "6 Zeichen eingeben!"
        With txt6Chars
            .SetFocus
            .SelStart = .TextLength
        End With
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabe von positiven Zahlen erzwingen

Problem: Wie kann ich die Eingabe von positiven Zahlen in eine UserFormTextBox erzwingen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

ClassModule: frmNurPositiv

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub txtValue_KeyDown( _
    ByVal KeyCode As MSForms.ReturnInteger, _
    ByVal Shift As Integer)
    Dim sTxt As String
    If KeyCode = 189 Then
        KeyCode = 0
    End If
End Sub

StandardModule: basMain

Sub CallForm()
    frmNurPositiv.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur numerische Eingaben zulassen

Problem: Wie kann ich Excel veranlassen, in einem Textfeld in einer UserForm nur numerische Eingaben zuzulassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CommandButton1_Click()
    frmNurZahlen.Show
End Sub

StandardModule: basMain

Sub CallForm()
    frmNurZahlen.Show
End Sub

ClassModule: frmNurZahlen

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub txtNumber_Change()
    Dim sTxt As String
    sTxt = txtNumber.Text
    If sTxt = "" Then Exit Sub
    If Right(sTxt, 1) Like "[0-9]" = False Then
        txtNumber.Text = Left(sTxt, Len(sTxt) - 1)
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text aus Textbox in Zellen aufteilen

Problem:	Wie kann ich den Text in einer UserForm-Textbox je Zeile auf verschiedene Zellen aufteilen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub DialogAufruf() frmText.Show End Sub ClassModule: frmText Private Sub cmdEintragen_Click() Dim iRow As Integer Dim sTxt As String sTxt = txtText.Text sTxt = WorksheetFunction.Substitute(sTxt, vbLf, "") Do iRow = iRow + 1 If InStr(sTxt, vbCr) Then Cells(iRow, 1).Value = Left(sTxt, InStr(sTxt, vbCr) - 1) sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, vbCr)) Else Cells(iRow, 1).Value = sTxt Exit Do End If Loop Unload Me End Sub Private Sub UserForm_Initialize() txtText.Text = _ "Hans W. Herber" & vbCr & _ "Im Schönblick 5" & vbCr & _ "53567 Asbach" & vbCr & _ "http://www.herber.de" & vbCr & _ "herber@herber.de" cmdEintragen.SetFocus End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen von TextBoxes hochzählen

Problem: Wie kann ich die Namen von TextBoxes - Form: TEXT01 - auf Knopfdruck um jeweils 1 hochzählen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub TextFelderUpDate()  
    Dim txtBox As TextBox  
    Dim vTxt As Variant  
    Dim iCounter As Integer  
    For iCounter = ActiveSheet.TextBoxes.Count To 1 Step -1  
        Set txtBox = ActiveSheet.TextBoxes(iCounter)  
        vTxt = CInt(Right(txtBox.Name, 2)) + 1  
        txtBox.Name = Left(txtBox.Name, 4) & Format(vTxt, "00")  
        txtBox.Text = txtBox.Name  
    Next iCounter  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Freiraum am linken Rand von MS-FORMS-Textfeldern löschen

Problem: Wie kann ich den standardmäßig vorgegebenen Freiraum am linken Rand von MS-Forms-Textfeldern unterdrücken?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMargin

Private Sub cmdMargin_Click()
    With txtMargin
        .SelectionMargin = Not .SelectionMargin
        .SetFocus
    End With
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmMargin.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text aus Textbox in Variable

Problem:	Wie weise ich einer Variablen den Inhalt einer Textbox zu?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: basMain Sub CallDialog() frmText.Show End Sub ClassModule: frmText Private Sub cmdOK_Click() Dim dValue As Double dValue = txtValue.Text MsgBox prompt:="TextBox-Wert: " & dValue Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert eines UserForm-ToggleButtons auslesen

Problem:	Der Wert eines ToggleButtons wird ausgelesen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmTgl Private Sub cmdCancel_Click() Unload Me End Sub Private Sub tgl1_Click() MsgBox tgl1.Value End Sub StandardModule: Modul1 Sub CallForm() frmTgl.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einlesen von Arbeitsmappen in ein TreeView-Steuerelement

Problem:	Die geöffneten Arbeitsmappen und deren Arbeitsblätter sollen in ein TreeView-Steuerelement eingelesen werden. Beim Anklicken eines Elementes soll das jeweilige Objekt aktiviert werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmTreeViewSample Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub TreeView1_Click() With TreeView1.SelectedItem If .Children = False Then Workbooks(.Parent.Text).Worksheets(.Text).Activate Else Workbooks(TreeView1.SelectedItem.Text).Activate End If End With End Sub Private Sub UserForm_Initialize() Dim wkb As Workbook Dim wks As Worksheet Dim ndeMain As Node, ndeSecond As Node Dim iCounter As Integer With TreeView1 For Each wkb In Workbooks iCounter = iCounter + 1 Set ndeMain = .Nodes.Add(Text:=wkb.Name) For Each wks In wkb.Worksheets Set ndeSecond = .Nodes.Add(relative:=ndeMain, _ relationship:=tvwChild, _ Text:=wks.Name) ndeSecond.Sorted = True Next wks Next wkb ndeMain.Expanded = True ndeMain.Sorted = True End With End Sub StandardModule: basMain Sub CallForm() frmTreeViewSample.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

GetOpenFilename-Methode aufrufen und Form modal belassen

Problem:	Wenn aus einer UserForm heraus die GetOpenFilename-Methode aufgerufen wird, bleibt die UserForm nicht modal und ein Tabellenblatt kann ausgewählt werden. Es handelt sich um einen Bug. Dieses Beispiel stellt einen WorkAround dar.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmMode Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdSelect_Click() Dim vFile As Variant frmMode.Hide vFile = Application.GetOpenFilename frmMode.Show End Sub StandardModule: Modul1 Sub CallForm() frmMode.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateiauswahl über UserForm

Problem:	In einer UserForm soll eine Datei zum Öffnen ausgewählt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmFiles Private Sub cmdOpen_Click() If Dir(txtFile.Text) = "" Then Beep MsgBox "Datei wurde nicht gefunden!" Else Workbooks.Open txtFile.Text End If Unload Me End Sub Private Sub cmdSelect_Click() Dim vFile As Variant vFile = Application.GetOpenFilename("Excel-Dateien (*.xls), *.xls") If vFile = False Then Exit Sub txtFile.Text = vFile End Sub StandardModule: Modul1 Sub CallForm() frmFiles.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm erstellen, anzeigen und löschen

Problem: Auf Schaltflächen-Klick wird eine UserForm erstellt, angezeigt und wieder gelöscht.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub CreateForm()
    Dim frmTemp
    Dim oTextBox As Msforms.TextBox
    Dim oBtnA As Msforms.CommandButton
    Dim oBtnB As Msforms.CommandButton
    Dim sCode As String
    Application.VBE.MainWindow.Visible = False
    Set frmTemp = ThisWorkbook.VBProject.VBComponents.Add(3)
    frmTemp.Properties("Width") = 135
    frmTemp.Properties("Height") = 115
    frmTemp.Properties("Caption") = "Nur ein Test"
    Set oTextBox = frmTemp.Designer.Controls.Add("forms.TextBox.1")
    With oTextBox
        .Left = 5
        .Top = 5
        .Width = 120
        .Height = 20
        .Text = "Dies ist ein Test"
    End With
    Set oBtnA = frmTemp.Designer.Controls.Add("forms.CommandButton.1")
    With oBtnA
        .Left = 5
        .Top = 30
        .Width = 120
        .Height = 20
        .Caption = "Einlesen"
    End With
    Set oBtnB = frmTemp.Designer.Controls.Add("forms.CommandButton.1")
    With oBtnB
        .Left = 5
        .Top = 60
        .Width = 120
        .Height = 20
        .Caption = "Abbrechen"
    End With
    sCode = "Private Sub CommandButton1_Click()" & vbCrLf
    sCode = sCode & "    MsgBox ""Wert aus der TextBox: "" & TextBox1.Text" & vbCrLf
    sCode = sCode & "End Sub" & vbCrLf & vbCrLf
    sCode = sCode & "Private Sub CommandButton2_Click()" & vbCrLf
    sCode = sCode & "    Unload Me" & vbCrLf
    sCode = sCode & "End Sub" & vbCrLf
    With frmTemp.CodeModule
        .AddFromString sCode
    End With
    VBA.UserForms.Add(frmTemp.Name).Show
    ThisWorkbook.VBProject.VBComponents.Remove VBComponent:=frmTemp
End Sub
```

Steuerelementwechsel bei Eingaben

Problem:	Nach Ausfüllen der letzten TextBox erfolgt Eintragung, die Werte in den TextBoxes werden zurückgesetzt und die erste TextBox wird zum Eintrag des nächsten Datensatz ausgewählt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	StandardModule: Modul1 <pre>Sub callform() frmEintragen.Show End Sub ClassModule: frmEintragen Dim bln As Boolean Private Sub cmdCancel_Click() Unload Me End Sub Private Sub TextBox2_Change() If TextBox2.TextLength = 4 Then TextBox3.SetFocus End Sub Private Sub TextBox3_Change() If TextBox3.TextLength = 4 Then TextBox4.SetFocus End Sub Private Sub TextBox4_Change() If TextBox4.TextLength = 4 Then TextBox5.SetFocus End Sub Private Sub TextBox2_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger) If KeyAscii < 48 Or KeyAscii > 57 Then KeyAscii = 0 End Sub Private Sub TextBox3_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger) If KeyAscii < 48 Or KeyAscii > 57 Then KeyAscii = 0 End Sub Private Sub TextBox4_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger) If KeyAscii < 48 Or KeyAscii > 57 Then KeyAscii = 0 End Sub Private Sub TextBox6_Exit(ByVal Cancel As MSForms.ReturnBoolean) Dim iTxt As Integer, iRow As Integer Dim sTxt As String If bln Then bln = False Exit Sub End If For iTxt = 1 To 6 If iRow < Cells(Rows.Count, iTxt).End(xlUp).Row + 1 Then iRow = Cells(Rows.Count, iTxt).End(xlUp).Row + 1 End If Next iTxt If iRow = 108 Then Beep MsgBox "Kein Eintrag mehr möglich!" End End If For iTxt = 1 To 6 sTxt = Controls("TextBox" & iTxt).Text Select Case iTxt Case 1, 5, 6 Cells(iRow, iTxt).Value = sTxt Case Else Cells(iRow, iTxt).Value = TimeSerial(CInt(Left(sTxt, 2)), CInt(Right(sTxt, 2)), 0) End Select Controls("TextBox" & iTxt).Text = "" Next iTxt bln = True TextBox1.SetFocus End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm mit und ohne Memory aufrufen

Problem: Eine UserForm soll ausgeblendet und mit Werten wieder aufgerufen werden, wobei alle Textfelder zu leeren sind.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmHide

Private Sub cmdHide_Click()
    Hide
End Sub

Private Sub cmdUnload_Click()
    Unload Me
End Sub

Private Sub UserForm_Activate()
    Dim cntr As Control
    For Each cntr In Controls
        If TypeName(cntr) = "TextBox" Then
            cntr.Text = ""
        End If
    Next cntr
    If lblCalls.Caption = "" Then
        lblCalls.Caption = "1"
    Else
        lblCalls.Caption = CStr(CInt(lblCalls.Caption) + 1)
    End If
End Sub

StandardModule: Modul1

Sub CallForm()
    frmHide.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei ComboBox-Auswahl Zellformel neu berechnen

Problem:	In UserForm-ComboBoxes sollen Werte aus den Spalten A und B gelistet werden. Bei einer ComboBox-Auswahl soll die Formel in Zelle F2 neu berechnet und der Wert in eine UserForm-TextBox übertragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmSetValue Private Sub cboA_Change() Range("D2").Value = cboA.ListIndex + 1 txtValue.Text = Range("F2").Value End Sub Private Sub cboB_Change() Range("E2").Value = cboB.ListIndex + 1 txtValue.Text = Range("F2").Value End Sub Private Sub cmdOK_Click() Unload Me End Sub Private Sub UserForm_Initialize() cboA.List = Range("A1").CurrentRegion.Columns(1).Value cboB.List = Range("A1").CurrentRegion.Columns(2).Value End Sub StandardModule: Modul1 Sub CallForm() frmSetValue.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textimport in UserForm-Elemente

Problem: Bei Eingabe einer 7-stelligen Zeichenfolge in eine UserForm-TextBox soll der String einer Textdatei gesucht werden. Bei Auffinden sind die Elemente des Datensatzes in Labels einzutragen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmSearch

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub txtSearch_Change()
    Dim iCounter As Integer, iFile As Integer
    Dim sFile As String, sTxt As String
    sFile = ThisWorkbook.Path & "\Source.txt"
    With txtSearch
        If .TextLength = 7 Then
            iFile = FreeFile
            Open sFile For Input As iFile
            Do Until EOF(iFile)
                Line Input #1, sTxt
                If Left(sTxt, 7) = .Text Then
                    Label1.Caption = Left(sTxt, 7)
                    Label2.Caption = Mid(sTxt, 8, 8)
                    Label3.Caption = Mid(sTxt, 16, 4)
                    Label4.Caption = Mid(sTxt, 20, 4)
                    Exit Do
                End If
            Loop
            Close
            If Label1.Caption = "" Then
                For iCounter = 1 To 4
                    Controls("Label" & iCounter).Caption = "Nicht gefunden"
                Next iCounter
            End If
            ElseIf .TextLength < 2 Then
                For iCounter = 1 To 4
                    Controls("Label" & iCounter).Caption = ""
                Next iCounter
            End If
        End With
    End Sub

StandardModule: Modul1

Sub CallForm()
    frmSearch.Show
End Sub
```

Lagerbewegungen erfassen und dokumentieren

Problem:	Lagerbewegungen sollen über eine UserForm erfaßt und dokumentiert werden.
Lösung:	Geben Sie den nachfolgenden Code in die genannten Module ein.
VBA-Code:	ClassModule: frmEdit Private Sub cboItems_Change() lblItems.Caption = Worksheets("Lager").Cells(cboItems.ListIndex + 2, 3).Value End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim iRow As Integer With wksLager If optAusgabe.Value = True Then .Cells(cboItems.ListIndex + 2, 3).Value = .Cells(cboItems.ListIndex + 2, 3).Value - CInt(txtPcs.Value) ElseIf optRueckgabe.Value = True Then .Cells(cboItems.ListIndex + 2, 3).Value = .Cells(cboItems.ListIndex + 2, 3).Value + CInt(txtPcs.Value) Else Beep MsgBox "Bitte Ein- oder Ausgabe wählen!" Exit Sub End If End With With wksBewegung iRow = .Cells(.Rows.Count, 1).End(xlUp).Row + 1 .Cells(iRow, 1).Value = cboItems.Value .Cells(iRow, 2).Value = Worksheets("Lager").Cells(cboItems.ListIndex + 2, 2).Value .Cells(iRow, 3).Value = CInt(txtPcs.Text) .Cells(iRow, 4).Value = txtName.Value .Cells(iRow, 5).Value = Now If optAusgabe.Value = True Then .Cells(iRow, 6).Value = "Ausgabe" Else .Cells(iRow, 6).Value = "Rückgabe" End If End With Unload Me End Sub Private Sub UserForm_Initialize() With wksLager cboItems.List = _ .Range(.Cells(2, 1), .Cells(.Range("A1").CurrentRegion.Rows.Count, 2)).Value End With cboItems.ListIndex = 0 End Sub StandardModule: basMain Sub CallForm() frmEdit.Show End Sub Sub LagerEinAus() With wksLager If .Visible = xlSheetVisible Then .Visible = xlSheetVeryHidden Else .Visible = xlSheetVisible .Select End If End With End Sub Sub BewegungEinAus() With wksBewegung If .Visible = xlSheetVisible Then .Visible = xlSheetVeryHidden Else .Visible = xlSheetVisible .Select End If End With End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Start- und Enddatum aus UserForm-TextBoxes als Filterkriterien

Problem: Die in UserForm-TextBoxes festgelegten Datum-Filterkriterien sollen beim Filtern zugrundegelegt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmFilter

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Range("A1").CurrentRegion.AutoFilter _
        field:=3, _
        Criteria1:=">=" & CDate(txtStart.Text), _
        Operator:=xlAnd, _
        Criteria2:="<=" & CDate(txtEnd.Text)
End Sub

StandardModule: Modul1

Sub Filtern()
    frmFilter.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In zweiter UserForm Werte aus erster UserForm anzeigen

Problem:	Eine erste UserForm soll eine zweite aufrufen, in deren TextBox die Auswahl einer ComboBox aus der ersten UserForm angezeigt wird.
Lösung:	Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmFirst Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() frmSecond.Show End Sub Private Sub UserForm_Initialize() Dim iMonth As Integer For iMonth = 1 To 12 cboValues.AddItem Format(DateSerial(1, iMonth, 1), "mmm") Next iMonth End Sub ClassModule: frmSecond Private Sub cmdOK_Click() Unload Me End Sub Private Sub UserForm_Initialize() txtValue.Text = frmFirst.cboValues.Value End Sub StandardModule: Modul1 Sub DialogAufruf() frmFirst.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen Dialog mit Dateiname und Dateidatum

Problem: Beim Öffnen der Arbeitsmappe soll ein Dialog mit Dateiname und Dateidatum angezeigt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    frmOpen.Show
End Sub

ClassModule: frmOpen

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim sTxt As String, sDate As String
    sDate = FileDateTime(ThisWorkbook.FullName)
    sTxt = "Programmname: " & ThisWorkbook.Name & vbCrLf
    sTxt = sTxt & "Versionsdatum: " & sDate
    lblVersion.Caption = sTxt
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auf UserForm-Schaltflächendruck Focus auf Tabelle setzen

Problem:	Auf UserForm-Schaltflächendruck soll der Focus auf die Tabelle gesetzt werden. Nur ab XL9, da die Modal-Eigenschaft erst ab dieser Version eingestellt werden kann.
Lösung:	Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub DialogAufruf() frmFocus.Show End Sub ClassModule: frmFocus Private Declare Function FindWindow Lib "user32" _ Alias "FindWindowA" (ByVal lpClassName As String, _ ByVal lpWindowName As String) As Long Private Declare Function SetForegroundWindow _ Lib "user32" (ByVal hwnd As Long) As Long Private m_hWndXl As Long Private Sub cmdFocus_Click() SetForegroundWindow m_hWndXl ActiveCell.Select End Sub Private Sub UserForm_Initialize() Application.Caption = "My Unique Caption" m_hWndXl = FindWindow("XLMAIN", Application.Caption) Application.Caption = Empty End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm starten, wenn Cursor in Zeile 5

Problem:	Wenn der Cursor in Zeile 5 gesetzt wird, soll eine UserForm mit den Einträgen am Ende der Zeile und der Möglichkeit einer freien Texteingabe angezeigt werden.
Lösung:	Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim iCol As Integer Application.EnableEvents = False If Selection.Cells.Count > 1 Then GoTo ERRORHANDLER On Error Resume Next iCol = Range("IV5").CurrentRegion.Column - 1 If Not Intersect(Target, Range(Cells(5, 1), Cells(5, iCol))) Is Nothing Then frmOptions.Show End If ERRORHANDLER: Application.EnableEvents = True End Sub ClassModule: frmOptions Dim bFlag As Boolean Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() Dim iRow As Integer If txtOption.Value = "" Then For iRow = 0 To lstOptions.ListCount - 1 If lstOptions.Selected(iRow) Then ActiveCell.Value = lstOptions.List(iRow) Exit For End If Next iRow Else ActiveCell.Value = txtOption.Text End If Unload Me End Sub Private Sub lstOptions_Change() If bFlag = True Then Exit Sub txtOption.Text = "" End Sub Private Sub txtOption_Change() Dim iRow As Integer If txtOption.Value = "" Or txtOption.TextLength > 1 Then Exit Sub bFlag = True For iRow = 0 To lstOptions.ListCount - 1 lstOptions.Selected(iRow) = False Next iRow End Sub Private Sub txtOption_Exit(ByVal Cancel As MSForms.ReturnBoolean) bFlag = False End Sub Private Sub UserForm_Initialize() Dim iCounter As Integer, iStart As Integer iStart = Range("IV5").CurrentRegion.Column For iCounter = iStart To 256 lstOptions.AddItem Cells(5, iCounter).Value Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-Schaltfläche während der Laufzeit hinzufügen

Problem: Während der Laufzeit soll einer UserForm eine Schaltfläche mit zugehörigem Ereigniscode hinzugefügt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdDialogAufruf_Click()
    frmButtons.Show
End Sub

ClassModule: frmButtons

Dim oEvents As New Collection

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdNewBtn_Click()
    Dim oNew As clsbtn
    Set oNew = New clsbtn
    If Controls.Count > 2 Then Exit Sub
    Set oNew.oBtn = frmButtons.Controls.Add( _
        "Forms.CommandButton.1", "cmdNew")
    With oNew.oBtn
        .Top = cmdNewBtn.Top
        .Left = cmdNewBtn.Left
        .Width = cmdNewBtn.Width
        .Height = cmdNewBtn.Height
        .Caption = "Klick mich"
    End With
    cmdNewBtn.Top = cmdCancel.Top
    cmdCancel.Top = cmdCancel.Top + 35
    Me.Height = Me.Height + 38
    oEvents.Add oNew
End Sub

ClassModule: clsBtn

Public WithEvents oBtn As MSForms.CommandButton

Private Sub oBtn_Click()
    MsgBox "Ich bin die neue Schaltfläche!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzeigen einer UserForm für den Zeitraum von 5 Sekunden

Problem: Wie kann ich eine UserForm für den Zeitraum von 5 Sekunden anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmMsg

Private Sub UserForm_Activate()
    Application.Wait Now + TimeSerial(0, 0, 5)
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmMsg.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dialog nach 3 Sekunden schließen

Problem: Ein Dialog mit einer Ja/Nein-Abfrage soll aufgerufen werden. Fällt der Anwender in den ersten 3 Sekunden nach dem Aufruf keine Entscheidung, wird die Betätigung der Ja-Schaltfläche angenommen und der Dialog geschlossen.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Public gdat As Date

Sub CallForm()
    frmHilfe.Show
End Sub

Sub Beenden()
    MsgBox "Aufgabe wird ausgeführt!"
    If gdat >= Now Then
        Application.OnTime EarliestTime:=gdat, _
            Procedure:="Beenden", Schedule:=False
    End If
    Unload frmHilfe
End Sub

ClassModule: frmHilfe

Private Sub cmdJa_Click()

    Call Beenden
    Unload Me
End Sub

Private Sub cmdCancel_Click()
    If gdat >= Now Then
        Application.OnTime EarliestTime:=gdat, _
            Procedure:="Beenden", Schedule:=False
    End If
    Unload Me
End Sub

Private Sub UserForm_Activate()
    gdat = Now + TimeSerial(0, 0, 3)
    Application.OnTime gdat, "Beenden", Schedule:=True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Größe einer UserForm einschl. der Elemente verändern

Problem: Wie kann ich während der Laufzeit die Breite einer UserForm einschließlich ihrer Elemente über eine Bildlaufleiste verändern?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmDynamic

Private Sub cmdDynamic_Click()
    Unload Me
End Sub

Private Sub scrDynamic_Change()
    Me.Width = scrDynamic.Value + 24
    scrDynamic.Width = scrDynamic.Value
    lstDynamic.Width = scrDynamic.Value
    cmdDynamic.Width = scrDynamic.Value
End Sub

StandardModule: basMain

Sub CallForm()
    frmDynamic.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm unter Berücksichtigung der Auflösung maximieren

Problem: Eine UserForm soll nach Auslesen der Bildschirmauflösung maximiert werden.

Lösung: Den nachfolgenden Code in die benannten Module einfügen.

VBA-Code:

```
StandardModule: basMain

Declare Function GetDeviceCaps Lib "gdi32" (ByVal hdc As Long, _
    ByVal nIndex As Long) As Long
Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long
Declare Function ReleaseDC Lib "user32" (ByVal hwnd As Long, _
    ByVal hdc As Long) As Long

Const HORZRES = 8
Const VERTRES = 10

Function ScreenResolution()
    Dim lRval As Long
    Dim lDc As Long
    Dim lHSize As Long
    Dim lVSize As Long
    lDc = GetDC(0&)
    lHSize = GetDeviceCaps(lDc, HORZRES)
    lVSize = GetDeviceCaps(lDc, VERTRES)
    lRval = ReleaseDC(0, lDc)
    ScreenResolution = lHSize & "x" & lVSize
End Function

Sub CallForm()
    frmFullSize.Show
End Sub

ClassModule: frmFullSize

Private Sub CommandButton1_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim sSize As String
    sSize = ScreenResolution
    With Me
        .Width = Left(sSize, InStr(sSize, "x") - 1)
        .Height = Right(sSize, Len(sSize) - _
            InStr(sSize, "x"))
        .Left = 0
        .Top = 0
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Programmsteuerung für mehrere aufrufende UserForms

Problem:	Wie kann ich ein von mehreren UserForms aufzurufendes Programm je nach aufrufender Form verzweigen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmDrei Private Sub cmdEintragen_Click() Call DatenEintragen(Me) End Sub Private Sub cmdWeiter_Click() Unload Me End Sub ClassModule: frmEins Private Sub cmdEintragen_Click() Call DatenEintragen(Me) End Sub Private Sub cmdWeiter_Click() Unload Me End Sub ClassModule: frmZwei Private Sub cmdEintragen_Click() Call DatenEintragen(Me) End Sub Private Sub cmdWeiter_Click() Unload Me End Sub StandardModule: basMain Sub DatenEintragen(frm) Dim iCounter As Integer, intCol As Integer Select Case frm.Name Case "frmEins": intCol = 1 Case "frmZwei": intCol = 2 Case "frmDrei": intCol = 3 End Select For iCounter = 1 To 3 frm.Controls("TextBox" & iCounter).Text = _ Cells(iCounter, intCol) Next iCounter End Sub Sub AufrufFormA() frmEins.Show End Sub Sub AufrufFormB() frmZwei.Show End Sub Sub AufrufFormC() frmDrei.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm als Fortschrittsanzeige

Problem:	Wie kann ich eine UserForm als Fortschrittsanzeige einsetzen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmProgress Private Sub cmdStart_Click() Dim dRow As Double Application.ScreenUpdating = False Me.Caption = "Bitte warten..." For dRow = 1 To 10000 If dRow Mod 10 = 0 Then lblProgress.Width = 222 * (dRow / 10000) fmeProgress.Caption = Format(dRow / 10000, "0%") DoEvents End If Cells(dRow, 1) = "Zeile " & dRow Next dRow Cells.ClearContents Unload Me End Sub Private Sub UserForm_Initialize() With lblProgress .Width = 0 .Left = lblBackGround.Left .Top = lblBackGround.Top End With End Sub StandardModule: basMain Sub CallForm() frmProgress.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Von zweiter UserForm die erste aufrufen

Problem:	Wie kann ich nach der Eingabe eines Wertes in einer UserFormTextBox die nächste UserForm aufrufen und von dort wieder zur ersten zurückspringen?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmFirst Private Sub cmdAusblenden_Click() Me.Hide frmSecond.Show End Sub ClassModule: frmSecond Private Sub cmdAusblenden_Click() Unload Me frmFirst.Show End Sub Private Sub cmdBeenden_Click() Unload frmFirst Unload Me End Sub StandardModule: basMain Sub CallForm() frmFirst.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zwischen zwei UserForms wechseln, wobei sich die 1. Aktualisiert

Problem: Wie kann ich zwischen zwei UserForms wechseln, wobei jeweils nur eine sichtbar bleibt und sich die erste nach Eintragungen automatisch aktualisiert?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmEintrag

Private Sub cmdEintragen_Click()
    Dim intRow As Integer
    intRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
    Cells(intRow, 1) = "Zeile " & intRow
    frmEintrag.Hide
    frmListe.Show
End Sub

ClassModule: frmListe

Private Sub cmdWeiter_Click()
    Unload Me
    End
End Sub

Private Sub cmdZumEintrag_Click()
    frmListe.Hide
    frmEintrag.Show
End Sub

Private Sub UserForm_Activate()
    With cbbListe
        .List = ActiveSheet.Range("A1").CurrentRegion.Value
        .ListIndex = .ListCount - 1
    End With
End Sub

Private Sub UserForm_Initialize()
    cbbListe.List = ActiveSheet.Range("A1").CurrentRegion.Value
    cbbListe.ListIndex = 0
End Sub

StandardModule: basMain

Sub CallForm()
    frmListe.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zu einer aufrufenden UserForm zurückkehren

Problem:	Wie kann ich erreichen, daß eine über eine zweite UserForm aufgerufene UserForm erkennt, welches das aufrufende Formular war und zu diesem wieder zurückkehren?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmAufruf1 Private Sub cmdAufruf1_Click() frmAufruf1.Hide With frmMain .cmdZurueck.Tag = 1 .cmdZurueck.Enabled = True End With frmMain.Show End Sub ClassModule: frmAufruf2 Private Sub cmdAufruf2_Click() frmAufruf2.Hide With frmMain .cmdZurueck.Tag = 2 .cmdZurueck.Enabled = True End With frmMain.Show End Sub ClassModule: frmMain Private Sub cmdBeenden_Click() Dim frmAct As UserForm For Each frmAct In UserForms Unload frmAct Next frmAct End Sub Private Sub cmdZuAufruf2_Click() frmMain.Hide frmAufruf2.Show End Sub Private Sub cmdZuAufruf1_Click() frmMain.Hide frmAufruf1.Show End Sub Private Sub cmdZurueck_Click() frmMain.Hide UserForms(cmdZurueck.Tag).Show End Sub StandardModule: basMain Sub CallForm() frmMain.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

AcrobatReader mit *.pdf-Dokument über UserForm starten

Problem: Wie kann ich über eine UserForm den AcrobatReader mit einem in einer Excel-Zelle benannten *.pdf-Datei starten?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmAcrobat

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub cmdAufruf_Click()
    Shell Chr(34) & Range("B1").Value & Chr(34) & _
        Chr(34) & Range("B2").Value & Chr(34), vbNormalFocus
End Sub

StandardModule: basMain

Sub CallForm()
    frmAcrobat.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm als Fortschrittsanzeige

Problem: Wie kann ich während eines Programmablaufes eine UserForm als Fortschrittsanzeige anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmFortschrittsanzeige

Private Sub cmdFortschrittsanzeige_Click()
    Dim iCounter As Integer
    For iCounter = 1 To 10
        Controls(iCounter).BackColor = 6723891
        Application.Wait Now + TimeSerial(0, 0, 1)
    Next iCounter
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

StandardModule: basMain

Sub CallForm()
    frmFortschrittsanzeige.Show
End Sub
```

Dialoggesteuerte Druckmehrblattauswahl

Problem:	Wenn ich mehrere Arbeitsblätter auswähle und den Druckbefehl gebe wird mir in der Kopfzeile als Gesamtzahl der Blätter die Gesamtzahl der Druckseiten aller ausgewählten Arbeitsblätter ausgegeben. Wie kann ich das auf die Seiten des jeweiligen Arbeitsblattes beschränken?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain <pre>Sub PrintSheets() Dim frmNew As Object Dim wks As Worksheet Dim chb As MSForms.CheckBox Dim cmd As MSForms.CommandButton Dim arr() As String Dim iCounter As Integer, iDrive As Integer Dim iTop As Integer Dim sCode As String Application.VBE.MainWindow.Visible = False On Error Resume Next Set frmNew = ThisWorkbook.VBProject.VBComponents("frmWks") If Err = 0 Then GoTo ERRORHANDLER On Error GoTo 0 Set frmNew = ThisWorkbook.VBProject.VBComponents.Add(3) iTop = 5 For Each wks In Worksheets Set chb = frmNew.Designer.Controls.Add("forms.CheckBox.1") With chb .Top = iTop .Left = 5 .Width = 100 .Caption = "Blatt " & wks.Name End With iTop = iTop + 20 Next wks Set cmd = frmNew.Designer.Controls.Add("forms.CommandButton.1") With cmd .Top = iTop .Left = 5 .Width = 100 .Height = 25 .Caption = "OK" .Name = "cmdOK" End With iTop = iTop + 30 Set cmd = frmNew.Designer.Controls.Add("forms.CommandButton.1") With cmd .Top = iTop .Left = 5 .Width = 100 .Height = 25 .Caption = "Abbrechen" .Name = "cmdCancel" End With With frmNew .Properties("Width") = 118 .Properties("Height") = iTop + 50 .Properties("Caption") = "Arbeitsblätter" .Properties("Name") = "frmWks" End With sCode = "Private Sub cmdCancel_Click" & vbCrLf sCode = sCode & " Unload Me" & vbCrLf sCode = sCode & "End Sub" & vbCrLf & vbCrLf sCode = sCode & "Private Sub cmdOK_Click" & vbCrLf sCode = sCode & " Dim arr() as String" & vbCrLf sCode = sCode & " Dim iArr as Integer" & vbCrLf sCode = sCode & " Dim iCounter As Integer" & vbCrLf sCode = sCode & " For iCounter = 1 to " & Worksheets.Count & vbCrLf sCode = sCode & " If Controls(""CheckBox"" & iCounter).Value" sCode = sCode & " = True Then" & vbCrLf sCode = sCode & " iArr = iArr + 1" & vbCrLf sCode = sCode & " Redim Preserve arr(1 to iArr)" & vbCrLf sCode = sCode & " arr(iArr) = Worksheets(iCounter).Name" & vbCrLf sCode = sCode & " End If" & vbCrLf sCode = sCode & " Next iCounter" & vbCrLf sCode = sCode & " If iArr > 0 Then" & vbCrLf sCode = sCode & " Unload Me" & vbCrLf sCode = sCode & " For iCounter = 1 to iArr" & vbCrLf sCode = sCode & " Worksheets(arr(iCounter)).PrintPreview" & vbCrLf sCode = sCode & " Next iCounter" & vbCrLf sCode = sCode & " Else" & vbCrLf sCode = sCode & " Unload Me" & vbCrLf sCode = sCode & " End If" & vbCrLf sCode = sCode & "End Sub" & vbCrLf ThisWorkbook.VBProject.VBComponents("frmWks").CodeModule.AddFromString sCode ERRORHANDLER: VBA.UserForms.Add(frmNew.Name).Show ThisWorkbook.VBProject.VBComponents.Remove ThisWorkbook.VBProject.VBComponents(frmNew.Name) Set frmNew = Nothing Exit Sub End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verhindern des Schließens einer UserForm mittels Schließkreuz

Problem:	Wie kann ich verhindern, daß der Anwender eine UserForm mittels des rechts oben im Dialog angebrachten Schließkreuzes abbrechen kann?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmClose Private Sub cmdContinue_Click() Unload Me End Sub Private Sub UserForm_QueryClose _ (Cancel As Integer, CloseMode As Integer) If CloseMode <> 1 Then Cancel = 1 End Sub StandardModule: basMain Sub CallForm() frmClose.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamischen Dialog erstellen

Problem: Wie kann ich mir in einem Dialog in Abhängigkeit der Aktivierung eines Optionsfeldes ein Label oder eine EditText anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
StandardModule: basMain

Sub CallForm()
    frmDynamisch.Show
End Sub

ClassModule: frmDynamisch

Private Sub cmdContinue_Click()
    Unload Me
End Sub

Private Sub OptionButton1_Change()
    If OptionButton1.Value = True Then
        Me.Width = 226.5
    Else
        Me.Width = 110.25
    End If
End Sub

Private Sub UserForm_Initialize()
    Dim iRow As Integer
    For iRow = 1 To 10
        ListBox1.AddItem "Zeile " & iRow
    Next iRow
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingangsbildschirm anzeigen

Problem: Wie kann ich beim Öffnen einer Arbeitsmappe einen Stardialog anzeigen lassen, der eine vorzugebende Zeit am Bildschirm verweilt?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    frmSplash.Show
End Sub

ClassModule: frmSplash

Private Sub UserForm_Activate()
    Application.Wait Now + TimeSerial(0, 0, 2)
    Unload Me
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wenn kein Eintrag, UserForm nicht schließen

Problem:	Wie kann ich verhindern, daß eine UserForm geschlossen wird, bevor ein Textfeld einen Eintrag erhält?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallForm() frmErzwingen.Show End Sub ClassModule: frmErzwingen Private Sub cmdContinue_Click() If txtZwang.Text = "" Then Beep MsgBox "Eintrag im Textfeld erforderlich!" txtZwang.SetFocus Else Unload Me End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm an Excel-Fenster anpassen

Problem:	Wie kann ich eine UserForm so an das Excel-Fenster anpassen, daß dieses überlagert wird?
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: frmCover Private Sub CommandButton1_Click() Unload Me End Sub Private Sub UserForm_Initialize() With frmCover .Height = Application.Height .Width = Application.Width End With End Sub StandardModule: basMain Sub CallForm() frmCover.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Excel für die Zeit eines Dialogaufrufes ausblenden

Problem:	Wie kann ich Excel beim Einblenden eines Dialogs ausblenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() Application.Visible = False Application.VBE.MainWindow.Visible = False frmContinues.Show Application.Visible = True End Sub ClassModule: frmContinues Private Sub cmdContinues_Click() Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafiken an eine Seite anpassen

Problem:	Wie kann ich Grafiken so verkleinern, daß sie auf eine Druckseite passen? Die Option "Anpassen" aus dem "Seite-enrichten"-Dialog soll nicht zur Lösung herangezogen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Anpassen() Dim oPct As Picture MsgBox "Vor der Anpassung" ActiveSheet.PrintPreview Set oPct = ActiveSheet.Pictures(1) While ExecuteExcel4Macro("Get.Document(50)") > 1 oPct.Height = oPct.Height * 90 / 100 oPct.Width = oPct.Width * 90 / 100 Wend MsgBox "Nach der Anpassung" ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckvorschau gem. Kontrollkästchen-Auswahl

Problem:	Die Tabellen gem. Kontrollkästchen sollen in der Druckvorschau angezeigt werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub DruckVorschau() Dim chb As CheckBox For Each chb In ActiveSheet.CheckBoxes If chb.Value = xlOn Then Worksheets(chb.Caption).PrintPreview End If Next chb End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blatt nur drucken, wenn Zeichenfolge im Namen

Problem: Es sollen nur die Arbeitsblätter gedruckt werden, die ein Tabelle in ihrem Namen tragen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub PrintSheets()
    Dim wks As Worksheet
    For Each wks In Worksheets
        If InStr(wks.Name, "Tabelle") Then
            wks.PrintPreview
        End If
    Next wks
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblatt mit dem aktuellen Monatsnamen drucken

Problem:	Wie kann ich auf Schaltflächendruck das Tabellenblatt mit dem aktuellen Monatsnamen drucken lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AnlegenMonate() Dim iMonth As Integer For iMonth = 1 To 12 Worksheets.Add.Move after:=Worksheets(Worksheets.Count) ActiveSheet.Name = Format(DateSerial(1, iMonth, 1), "mmm") Next iMonth Worksheets(1).Select End Sub Sub MonatDrucken() Worksheets(Format(DateSerial(_ 1, Month(Date), 1), "mmm")).PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druck der in einer UserForm-ListBox ausgewählten Arbeitsblätter

Problem:	Wie kann ich die in einer UserForm-ListBox per Mehrfachauswahl ausgewählten Tabellenblätter drucken lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CallForm() frmDrucken.Show End Sub ClassModule: frmDrucken Private Sub cmdAbbrechen_Click() Unload Me End Sub Private Sub cmdDrucken_Click() Dim arrWks() Dim iCounter As Integer, iCount As Integer For iCounter = 0 To lstDrucken.ListCount - 1 If lstDrucken.Selected(iCounter) Then Worksheets(lstDrucken.List(iCounter)).PrintOut End If Next iCounter Unload Me End Sub Private Sub UserForm_Initialize() Dim wks As Worksheet For Each wks In Worksheets lstDrucken.AddItem wks.Name Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblätter in umgekehrter Reihenfolge drucken

Problem:	Wie kann ich alle Arbeitsblätter einer Arbeitsmappe in umgekehrter Reihenfolge ausdrucken?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub UmgekehrtDrucken() Dim iCounter As Integer For iCounter = Worksheets.Count To 1 Step -1 Worksheets(iCounter).PrintPreview Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Sichtbare Blätter drucken

Problem: Wie kann ich alle eingeblendeten Arbeitsblätter einer Arbeitsmappe über ein VBA-Makro ausdrucken?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub PrintVisibles()
    Dim wks As Worksheet
    For Each wks In Worksheets
        If wks.Visible = True Then
            wks.PrintPreview
        End If
    Next wks
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Dateien eines Verzeichnisses drucken

Problem:	Wie kann ich über VBA alle Arbeitsmappen eines Ordners ausdrucken lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AlleDrucken() Dim iCounter As Integer Application.ScreenUpdating = False Application.EnableEvents = False On Error GoTo ERRORHANDLER With Application.FileSearch .LookIn = Range("B1").Value .FileType = msoFileTypeExcelWorkbooks .Execute For iCounter = 1 To .FoundFiles.Count Workbooks.Open .FoundFiles(iCounter), False ActiveWorkbook.PrintOut ActiveWorkbook.Close savechanges:=False Next iCounter End With ERRORHANDLER: Application.ScreenUpdating = True Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur Wert eines Kombinationsfeldes, nicht das Feld drucken

Problem: Es soll nur der ausgewählte Wert eines Kombinationsfeldes, nicht das Feld selbst gedruckt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub GetPrint()
    ActiveSheet.PrintPreview
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit variablem oberer und unterer Rahmenlinie drucken

Problem:	Alle Seiten sollen mit variablen oberen und unteren Rahmen gedruckt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub IsPageBreak() Dim rng As Range Dim vRowBrk As Variant Dim lRow As Long Dim iBreak As Integer Set rng = Range("A1").CurrentRegion Cells.Borders.LineStyle = xlLineStyleNone rng.BorderAround LineStyle:=xlContinuous, Weight:=xlMedium iBreak = 1 Do vRowBrk = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(64)," & iBreak & ")") iBreak = iBreak + 1 If IsError(vRowBrk) Then Exit Do rng.Rows(vRowBrk - 1).Borders(xlEdgeBottom).Weight = xlMedium rng.Rows(vRowBrk).Borders(xlEdgeTop).Weight = xlMedium Loop ActiveSheet.PrintPreview Cells.Borders.LineStyle = xlLineStyleNone End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckbereich der 1. Seite ermitteln

Problem:	Der Druckbereich der 1. Seite soll über MessageBox gemeldet werden. Wenn es sich horizontal nicht um mehrere Seiten handelt, muss der Code angepasst werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub FirstPageAddress() Dim rng As Range Dim iRow As Integer, iCol As Integer, iCounter As Integer Application.ScreenUpdating = False iRow = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(64),1)") - 1 iCol = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(65),1)") - 1 Set rng = Range(Cells(1, 1), Cells(iRow, iCol)) MsgBox rng.Address End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckbereichsauswahl über CheckBoxes

Problem: Die Druckvorschau soll gemäß den mit CheckBoxes ausgewählten Druckoptionen erfolgen. Alle zu druckenden Zellen müssen an den in der Tabelle sichtbaren Positionen gedruckt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdPrint_Click()
    Dim rng As Range
    Dim sAddress As String
    Application.ScreenUpdating = False
    ActiveCell.Activate
    sAddress = Selection.Address
    ActiveSheet.Copy
    If chbHeader.Value = False Then
        With ActiveSheet.PageSetup
            .LeftHeader = ""
            .CenterHeader = ""
            .RightHeader = ""
        End With
    End If
    If chbFooter.Value = False Then
        With ActiveSheet.PageSetup
            .LeftFooter = ""
            .CenterFooter = ""
            .RightFooter = ""
        End With
    End If
    If chbSelection.Value = True Then
        For Each rng In ActiveSheet.UsedRange.Cells
            If Intersect(rng, ActiveSheet.Range(sAddress)) Is Nothing Then
                rng.NumberFormat = ";;;;"
            End If
        Next rng
    End If
    ActiveSheet.PrintPreview
    ActiveWorkbook.Close savechanges:=False
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Zelleingabe Druckvorgang starten

Problem:	Bei Eingabe der Monatsanzahl in Zelle B2 soll der Gesamttext des Bereiches A1:B3 gedruckt werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Address <> "\$B\$2" Then Exit Sub ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vor dem Druck Zeilen einfügen und danach wieder löschen

Problem: Vor dem Druck nach soll nach jedem Datensatz eine Zeile eingefügt und danach wieder gelöscht werden.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code: StandardModule: Modul1

```
Sub LeerZeilen()  
    Dim iRowL As Integer, iRow As Integer  
    Application.ScreenUpdating = False  
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row  
    For iRow = iRowL To 1 Step -1  
        Rows(iRow).Insert  
    Next iRow  
    ActiveSheet.PrintPreview  
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row - 1  
    For iRow = iRowL To 1 Step -2  
        Rows(iRow).Delete  
    Next iRow  
    Application.ScreenUpdating = True  
End Sub  
  
Sub ZeilenHoehe()  
    With Range("A1").CurrentRegion.Rows  
        .RowHeight = .RowHeight * 2  
        ActiveSheet.PrintPreview  
        .RowHeight = .RowHeight / 2  
    End With  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abfrage vor Start- und Enddatum eines zu druckenden Bereiches

Problem:	Das Start- und Enddatum eines zum Drucken auszuwählenden Bereiches sollen über InputBox abgefragt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub DatePrint() Dim varStart As Variant, varEnd As Variant Dim dStart As Double, dEnd As Double Dim sStart As String, sEnd As String sStart = InputBox("Start:", , "12.02.01") If sStart = "" Then Exit Sub sEnd = InputBox("Ende:", , "15.03.01") If sEnd = "" Then Exit Sub On Error Resume Next dStart = CDbl(DateValue(sStart)) dEnd = CDbl(DateValue(sEnd)) If Err > 0 Then Err.Clear MsgBox "Ungültige Eingaben!" Exit Sub End If If dEnd < dStart Then Beep MsgBox "Das Enddatum darf nicht kleiner " & _ "als das Startdatum sein!" End If varStart = Application.Match(dStart, Columns(1), 0) varEnd = Application.Match(dEnd, Columns(1), 0) ActiveSheet.PageSetup.PrintArea = _ Range(Cells(varStart, 1), Cells(varEnd, 1)).Address ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen nicht drucken, wenn Spalte C leer oder 0

Problem:	Zeilen mit leeren Zellen oder Nullwerten in Spalte sollen zum Druck ausgeblendet werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Drucken() Dim iRowL As Integer, iRow As Integer iRowL = Cells(Rows.Count, 3).End(xlUp).Row For iRow = 1 To iRowL If IsEmpty(Cells(iRow, 3)) Or Cells(iRow, 3).Value = 0 Then Rows(iRow).Hidden = True End If Next iRow ActiveSheet.PrintPreview Rows.Hidden = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druck einer Tabelle mit wechselndem Wiederholungszeilentext

Problem:	Die Tabelle soll mit wechselndem Wiederholungszeilentext gedruckt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub VariableZeile() Dim iCount As Integer, iCounter As Integer Range("A1").Select iCount = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(50),1)") ActiveSheet.PageSetup.PrintTitleRows = "1:1" For iCounter = 1 To iCount Range("A1").Value = "Wiederholungszeile " & iCounter & ". Seite" ActiveSheet.PrintOut from:=iCounter, To:=iCounter Next iCounter Range("A1").Value = "Wiederholungszeile" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einfügen von Zeilen vor und nach Zellen mit Fettdruck

Problem: Vor dem Ausdruck sollen vor die erste und nach der letzten Zelle mit Fettdruck neue Zeilen eingefügt werden. Start über die normale Drucken-Schaltfläche oder über das Menü Druck.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_BeforePrint(Cancel As Boolean)
    Dim iFirst As Integer, iSecond As Integer, iRow As Integer
    iRow = 1
    Do Until IsEmpty(Cells(iRow, 1))
        If Cells(iRow, 1).Font.Bold = True Then
            If iFirst = 0 Then iFirst = iRow
        Else
            If iFirst > 0 Then
                iSecond = iRow
                Exit Do
            End If
        End If
        iRow = iRow + 1
    Loop
    Rows(iSecond).Insert
    With Cells(iSecond, 1)
        .Value = "Nach den fetten Zeilen"
        .Font.Bold = False
    End With
    Rows(iFirst).Insert
    Cells(iFirst, 1).Value = "Vor den fetten Zeilen"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Markierten Bereich drucken

Problem:	Der gelb markierte Bereich soll gedruckt werden. Die Positionierung ist so vorzunehmen, als würde das ganze Blatt gedruckt, Rahmen sind in den Druck nicht, Einstellung für der Zeilen- und Spaltenhöhe sind einzubehiehen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BereichDrucken() Dim rng As Range Application.ScreenUpdating = False ActiveSheet.Copy For Each rng In Range("A1").CurrentRegion If rng.Interior.ColorIndex <> 6 Then rng.ClearContents End If Next rng Cells.Borders.LineStyle = xlLineStyleNone ActiveSheet.PrintPreview ActiveWorkbook.Close savechanges:=False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckbereich in Abhängigkeit zur Einträgen in einer Spalte

Problem:	Der Druckbereich soll in Abhängigkeit der Anzahl von Einträgen in Spalte B festgelegt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub BereichFestlegen() Dim rng As Range Set rng = Range(Cells(1, 1), _ Cells(Cells(1, 2).End(xlDown).Row, 8)) ActiveSheet.PageSetup.PrintArea = rng.Address ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktuellen Bereich in den Spalten A:F als Druckbereich

Problem:	Wie kann ich den Bereich von Zelle A1 bis zur letzten Zelle mit Inhalt in Spalte F als Druckbereich festlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Drucken() Dim iCol As Integer, iRow As Integer For iCol = 1 To 4 If Cells(Rows.Count, iCol).End(xlUp).Row > iRow Then iRow = Cells(Rows.Count, iCol).End(xlUp).Row End If Next iCol ActiveSheet.PageSetup.PrintArea = Range("A1:F" & iRow).Address ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Den in einer Zelle genannten Bereich eines Blattes drucken

Problem:	Wie kann ich den in einer Zelle genannten Druckbereich eines anderen Tabellenblattes über VBA ausdrucken?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Drucken() Dim swks As String, sRng As String, sAdr As String sAdr = Range("A1").Value swks = Left(sAdr, InStr(sAdr, "!") - 1) sRng = Right(sAdr, Len(sAdr) - InStr(sAdr, "!")) Worksheets(swks).Range(sRng).PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text aus UserForm-TextBox drucken

Problem: Wie kann ich einen in einer UserForm-TextBox stehenden Text auf Schaltflächendruck auf den Drucker ausgeben?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: frmDrucken

Private Sub cmdDrucken_Click()
    Unload Me
    Application.ScreenUpdating = False
    Workbooks.Add 1
    Range("A1").Value = txtEingabe.Text
    ActiveSheet.PrintPreview
    ActiveWorkbook.Close savechanges:=False
    Application.ScreenUpdating = True
End Sub

StandardModule: basMain

Sub CallForm()
    frmDrucken.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckstart- und endzeilen über InputBox abfragen

Problem:	Wie kann ich die Druckstart- und Endzeilen über InputBox abfragen und den ausgewählten Bereich als Druckvorschau anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BereichsAuswahl() Dim rng As Range Dim vStart As Variant, vEnd As Variant vStart = Application.InputBox(_ prompt:="Startzeile:", Default:=50, Type:=1) If vStart = "" Then Exit Sub vEnd = Application.InputBox(_ prompt:="Startzeile:", Default:=150, Type:=1) If vEnd = "" Then Exit Sub Set rng = Range("A" & vStart & ":G" & vEnd) rng.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckbereich bis zum ersten Nullwert festlegen

Problem:	Wie kann ich den Druckbereich so festlegen, daß er den Bereich des aktiven Tabellenblattes nur bis zum ersten Nullwert in Spalte A umfaßt?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DruckBereichFestlegen() Dim iRow As Integer iRow = 1 Do Until Cells(iRow + 1, 1).Value = 0 iRow = iRow + 1 Loop Worksheets("Tabelle1").PageSetup.PrintArea = _ ActiveSheet.Name & "!" & _ Range(Cells(1, 1), Cells(iRow, 11)).Address ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Label in Zelle A1 bei jedem Ausdruck um 1 hochzählen

Problem:	Wie kann ich eine Zeichenfolge in der Form 1999-1001 bei jedem Ausdruck um 1 höherzählen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SerienDruck() Dim intCounter As Integer Application.ScreenUpdating = False For intCounter = 1001 To 1003 ActiveSheet.PrintOut Range("A1").Value = Left(Range("A1"), 5) & _ CInt(Right(Range("A1"), 4) + 1) Next intCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckbereich variabel gestalten

Problem:	Wie kann ich durch die Eingabe einer Zahl zwischen verschiedenen Druckbereichen hin- und herschalten?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Jede zweite Spalte drucken

Problem:	Wie kann ich veranlassen, daß nur jede zweite Spalte gedruckt wird?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul2 Sub SpaltenDrucken() Dim iCol As Integer Application.ScreenUpdating = False For iCol = 1 To ActiveSheet.UsedRange.Columns.Count If iCol Mod 2 <> 0 Then Columns(iCol).Hidden = True Next iCol ActiveSheet.PrintPreview ActiveSheet.Columns.Hidden = False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur Zeilen mit einer "2" drucken

Problem:	Wie kann ich veranlassen, daß nur die Zeilen gedruckt werden, die in spalte A eine 2 enthalten?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Drucken() Dim iRow As Integer, iRowL As Integer Application.ScreenUpdating = False iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL If Cells(iRow, 1) <> 2 Then Rows(iRow).Hidden = True End If Next iRow ActiveSheet.PrintPreview Rows.Hidden = False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bereiche in Abhängigkeit eines Zellinhaltes drucken

Problem:	Wie kann ich die Druckbereiche in Abhängigkeit des Inhalts einer Zelle verändern. Wenn in Zelle A1 "1. Bereich" steht, soll der 1., wenn dort "2. Bereich" steht, der 2. gedruckt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DruckeAlternativ() If Range("A1").Value = "1. Bereich" Then ActiveSheet.PageSetup.PrintArea = "A1:F20" ElseIf Range("A1").Value = "2. Bereich" Then ActiveSheet.PageSetup.PrintArea = "G1:L20" Else Exit Sub End If ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Leere Zeilen beim Druck ausblenden

Problem:	Wie kann ich alle leere Zeilen beim druck ausblenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DruckZeilen() Dim iRow As Integer, iRowL As Integer Application.ScreenUpdating = False iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL If IsEmpty(Cells(iRow, 1)) Then Rows(iRow).Hidden = True End If Next iRow ActiveSheet.PrintPreview Rows.Hidden = False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Videokassettenaufkleber drucken

Problem: Wie kann ich aus einer Video-Liste heraus die zugehörigen Aufkleber drucken?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim iRow As Integer
    If Target.Column <> 2 Then Exit Sub
    Cancel = True
    With Worksheets("LabelDruck")
        iRow = WorksheetFunction.CountA(.Columns(1)) + 1
        If iRow > 12 Then Exit Sub
        .Cells(iRow, 1).Value = Target.Offset(0, -1).Value
        .Cells(iRow, 2).Value = Target.Value
        .Cells(iRow, 3).Value = Target.Offset(0, 3).Value
        .Cells(iRow, 4).Value = Target.Offset(0, 4).Value
    End With
End Sub

ClassModule: Tabelle3

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim var As Variant
    If Target.Cells.Count > 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Target.Column <> 1 Then Exit Sub
    If Target.Row > 12 Then Exit Sub
    With Worksheets("Videoliste")
        var = Application.Match(Target.Value, .Columns(1), 0)
        If Not IsError(var) Then
            Target.Offset(0, 1).Value = .Cells(var, 2).Value
            Target.Offset(0, 2).Value = .Cells(var, 5).Value
            Target.Offset(0, 3).Value = .Cells(var, 6).Value
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Überträge bei Mehrseitendruck

Problem:	Wie kann ich beim Druck mehrerer Seiten Überträge erstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MehrSeitenDruck() Dim rngSource As Range, rngTarget As Range Dim iPage As Integer Set rngSource = Worksheets("Quellblatt").Range("A1").CurrentRegion Set rngTarget = Range("A2:F59") For iPage = 1 To rngSource.Rows.Count Step 58 rngTarget.ClearContents rngTarget.Value = rngSource.Range(_ rngSource.Cells(iPage, 1), _ rngSource.Cells(iPage + 58 - 1, 6)).Value ActiveSheet.PrintPreview Next iPage rngTarget.ClearContents End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schmale Spalten nebeneinander drucken

Problem: Die Spalten A und B sind sehr schmal. Wie kann ich den Ausdruck so automatisieren, daß diese in drei Kolonnen nebeneinander gedruckt werden?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub MultiSpaltenDruck()  
    Dim rng As Range  
    Dim iRow As Integer, iCountR As Integer  
    Dim iRowT As Integer, iColT As Integer  
    Dim iCounter As Integer  
    Application.ScreenUpdating = False  
    Set rng = Range("A1").CurrentRegion  
    iCountR = 55  
    Workbooks.Add 1  
    iRow = 1  
    iRowT = 1  
    iColT = 1  
    Do While iRow <= rng.Rows.Count  
        For iCounter = 1 To 3  
            rng.Range(rng.Cells(iRow, 1), _  
                rng.Cells(iRow + iCountR - 1, 2)).Copy _  
                Cells(iRowT, iColT)  
            iRow = iRow + iCountR  
            iColT = iColT + 2  
        Next iCounter  
        ActiveSheet.PrintPreview  
        iColT = 1  
    Loop  
    ActiveWorkbook.Close savechanges:=False  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrbereichsauswahl auf eine Seite drucken

Problem: Bei einer Mehrbereichsauswahl druckt Excel jede Seite auf einem extra Blatt. Wie kann ich die Bereiche so zusammenfassen, daß nur ein Blatt gedruckt wird?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub MehrBereichsDruck()
    Dim rng As Range, rngAct As Range
    Dim iRow As Integer, iCounter As Integer
    Application.ScreenUpdating = False
    Set rng = Selection
    Workbooks.Add 1
    For Each rngAct In rng.Areas
        iCounter = iCounter + 1
        If iRow > 0 Then
            iRow = Cells(Rows.Count, 1).End(xlUp).Row + 2
        Else
            iRow = 1
        End If
        Cells(iRow, 1).Value = iCounter & ". Bereich:"
        rngAct.Copy Cells(iRow + 1, 1)
    Next rngAct
    ActiveSheet.PrintPreview
    ActiveWorkbook.Close savechanges:=False
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zuerst Vorder-, dann Rückseitendruck

Problem:	Wie kann ich zuerst alle Vorder-, dann alle Rückseiten drucken lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DblShPrint() Dim varPB As Variant Dim iCounter As Integer, iPage As Integer, iSheet As Integer Dim iRow As Integer, iRowL As Integer Application.ScreenUpdating = False iRowL = Cells.Find("*", Range("a1"), , , xlByRows, xlPrevious).Row iSheet = 1 iRow = 1 For iPage = 1 To 2 For iCounter = 1 To ExecuteExcel4Macro("GET.DOCUMENT(50)") varPB = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(64), " & _ & iSheet & ")") If IsError(varPB) Then varPB = iRowL + 1 If iPage = 1 And iSheet Mod 2 <> 0 Or _ iPage = 2 And iSheet Mod 2 = 0 Then Range(Cells(iRow, 1), Cells(varPB - 1, 5)) _ .PrintPreview End If iSheet = iSheet + 1 iRow = varPB Next iCounter iSheet = 1 iRow = 1 If iPage = 1 Then MsgBox "Bitte Blätter einlegen!" Next iPage End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen mit leeren Formeln beim Druck unterdrücken

Problem:	Wenn ich ein Blatt ausdrücke, werden Zeilen mit Zellen, in denen sich Formeln mit Leerstrings befinden mit ausgedruckt. Wie kann ich das verhindern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub OhneAusblenden() ActiveSheet.PrintPreview End Sub Sub FuerDruckAusblenden() Dim rng As Range Dim iRow As Integer, iCol As Integer Dim bln As Boolean Set rng = Range("A1").CurrentRegion For iRow = 1 To rng.Rows.Count bln = False For iCol = 1 To rng.Columns.Count If Cells(iRow, iCol).Value <> "" Then bln = True Exit For End If Next iCol If bln = False Then rng.Rows(iRow).Hidden = True End If Next iRow ActiveSheet.PrintPreview Rows.Hidden = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nummern vorgeben und drucken

Problem: Wie kann ich über einen Dialog einen fünfstelligen Von/Bis-Nummernbereich vorgeben, die jeweilige Nummer in eine Zelle eintragen und das Blatt ausdrucken lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub DruckDialog()  
    Dim sTxt As String  
    Dim lFrom As Long, lTo As Long, lCounter As Integer  
    sTxt = InputBox("Von - bis:", , "30001-30005")  
    If sTxt = "" Then Exit Sub  
    If InStr(sTxt, "-") = False Then Exit Sub  
    If Not IsNumeric(Left(sTxt, 5)) Then Exit Sub  
    If Not IsNumeric(Right(sTxt, Len(sTxt) - InStr(sTxt, "-"))) _  
        Then Exit Sub  
    lFrom = Left(sTxt, 5)  
    lTo = Right(sTxt, Len(sTxt) - InStr(sTxt, "-"))  
    For lCounter = lFrom To lTo  
        With Worksheets("Tabelle1")  
            .Range("F7") = lCounter  
            .PrintPreview  
        End With  
    Next lCounter  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Leere Zeilen beim Drucken unterdrücken

Problem:	Wie kann ich die Anzeige aller leeren Zeilen beim Drucken unterdrücken?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Start() Dim iRow As Integer, iRowL As Integer Application.ScreenUpdating = False ActiveSheet.DisplayAutomaticPageBreaks = False iRowL = Cells.Find("*", Range("A1"), , , xlByRows, xlPrevious).Row For iRow = iRowL To 1 Step -1 If WorksheetFunction.CountA(Rows(iRow)) = 0 Then Rows(iRow).Hidden = True End If Next iRow ActiveSheet.PrintOut Rows.Hidden = False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen vom Druck ausschließen

Problem: Wie kann ich einen Zellbereich vom Druck ausschließen? Der auszuschließende Bereich soll über einen Dialog festgelegt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub BereichFestlegen()  
    Const csMsg As String = "Auszuschließenden Bereich selektieren:"  
    Dim rng As Range  
    Dim sTxt As String  
    Set rng = Application.InputBox(prompt:=csMsg, Type:=8)  
    sTxt = rng.NumberFormat  
    rng.NumberFormat = ";;;"  
    ActiveSheet.Range("A1:A50").PrintPreview  
    rng.NumberFormat = sTxt  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrbereichsauswahl auf eine Seite drucken.

Problem:	Wenn ich eine Mehrbereichsauswahl drucke, verwendet Excel für jeden Bereich eine Seite, obwohl alle Bereiche auf einer Seite Platz fänden. Wie kann ich das ändern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MehrBereichsDruck() Dim wksSource As Worksheet, wksTarget As Worksheet Dim rng As Range Dim iRow As Integer, intRng As Integer Application.ScreenUpdating = False Set wksSource = ActiveSheet Set wksTarget = Worksheets.Add wksSource.Select iRow = 1 With wksTarget For Each rng In Selection.Areas intRng = intRng + 1 .Cells(iRow, 1) = "Bereich Nr. " & intRng rng.Copy .Cells(iRow + 2, 1).PasteSpecial Paste:=xlValues iRow = .Cells.Find("*", Range("A1"), , , _ xlByRows, xlPrevious).Row + 2 Next rng .Columns.AutoFit .PrintPreview Application.DisplayAlerts = False .Delete Application.DisplayAlerts = True End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateiname vorgeben bei Druck in Printdatei

Problem: Wenn ich einen Ausdruck in eine Druckdatei umleite, habe ich über VBA keine Möglichkeit, den Dateinamen vorzugeben. Ist dieses Problem zu umgehen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub TestDruck()
    SendKeys "test.prn{enter}"
    ActiveWindow.SelectedSheets.PrintOut _
        Copies:=1, PrintToFile:=True, Collate:=True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktuellen Dateipfad aktualisieren

Problem: Wie kann ich veranlassen, daß der aktuelle volle Dateiname bei jedem Druck in die Fußzeile übernommen wird?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforePrint(Cancel As Boolean)
    ActiveSheet.PageSetup.LeftFooter = _
        ThisWorkbook.FullName
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Drucker auswählen und nach dem Drucken zurücksetzen

Problem:	Vor dem Druck der Tabelle soll ein Dialog zur Druckerauswahl aufgerufen werden. Nach dem Ausdruck wird der Drucker wieder zurückgesetzt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SelectPrinter() Dim sPrinter As String sPrinter = Application.ActivePrinter Application.Dialogs(xlDialogPrinterSetup).Show ActiveSheet.PrintPreview Application.ActivePrinter = sPrinter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einstellen des Drucker-Papiereinzug-Schachts

Problem:	Wie kann ich den Papiereinzugschacht des Druckers über VBA einstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SchachtFestlegen() Application.SendKeys "%di%o{tab 2}{down 2}{enter 2}" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zum Druck auf Fax-Drucker umschalten und wieder zurück

Problem: Wie kann ich zum Ausdruck einer Tabelle auf den Fritz-Fax-Drucker umschalten und nach dem Ausdruck wieder zum vorher aktiven Drucker zurückkehren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Drucken()
    Dim sPrinter As String
    Application.ScreenUpdating = False
    sPrinter = Application.ActivePrinter
    Application.ActivePrinter = _
        "FRITZfax Drucker auf FRITZ!fax:"
    Worksheets("Tabelle1").PrintOut
    Application.ActivePrinter = sPrinter
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktiven Drucker in UserForm wechseln

Problem:	Wie kann ich in einem UserForm-Dialog den aktiven Drucker wechseln?
Lösung:	Nachfolgendes Makro in das Klassenmodul der UserForm einfügen, dem Code entsprechende Steuerelemente hinzufügen und starten.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdStart_Click() frmPrinter.Show End Sub ClassModule: frmPrinter Private Sub cmdChangePrinter_Click() Me.Hide Application.Dialogs(xlDialogPrinterSetup).Show lblPrinter.Caption = Application.ActivePrinter Me.Show End Sub Private Sub CommandButton1_Click() Unload Me End Sub Private Sub UserForm_Activate() lblPrinter.Caption = Application.ActivePrinter End Sub StandardModule: basMain Sub CallForm() frmPrinter.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckereinstellungen verändern

Problem:	Wie kann ich bei den Druckereinstellungen den Druckschacht vorgeben?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub PrintSettings() SendKeys "%e%z{down 2}{enter 2}" Application.Dialogs(xlDialogPrint).Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Teilbereiche als Wiederholungszeilen anlegen

Problem:	Die Zeilen 1:11 sollen unter Weglassung der Zeilen 5:6 als Wiederholungszeilen festgelegt werden.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub PrintSheet() With ActiveSheet .PageSetup.PrintTitleRows = "\$1:\$11" .Rows("5:6").Hidden = True .PrintPreview .Rows("5:6").Hidden = False End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopf- und Fußzeileneintrag vor dem Drucken

Problem: Vor dem Drucken wird über das BeforePrintEreignis der Text gem. Spalte B eingetragen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_BeforePrint(Cancel As Boolean)
    If ActiveSheet.Name = "Tabelle1" Then
        ActiveSheet.PageSetup.LeftFooter = Range("B1").Value
        ActiveSheet.PageSetup.CenterFooter = Range("B2").Value
        ActiveSheet.PageSetup.RightFooter = Range("B3").Value
    Else
        ActiveSheet.PageSetup.LeftFooter = ""
        ActiveSheet.PageSetup.CenterFooter = ""
        ActiveSheet.PageSetup.RightFooter = ""
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dokumententitel und UserName in Kopfzeile übernehmen

Problem: Dokumententitel und Name des Autors sollen in die Druckkopfzeile übernommen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub PrintSheet()
    Dim sHead As String
    With ThisWorkbook.BuiltinDocumentProperties
        sHead = "Title: " & .Item(1) & vbCrLf & "Author: " & .Item(3)
    End With
    ActiveSheet.PageSetup.LeftHeader = sHead
    ActiveSheet.PrintPreview
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopfzeile nur auf erstem Blatt drucken

Problem:	Wie kann ich erreichen, dass nur auf dem ersten Blatt eine Kopfzeile gedruckt wird?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Drucken() Dim iPage As Integer For iPage = 1 To ExecuteExcel4Macro("GET.DOCUMENT(50)") With Worksheets("Tabelle1") If iPage = 1 Then .PageSetup.LeftHeader = "Mein Dokument" Else .PageSetup.LeftHeader = "" End If .PrintOut from:=iPage, To:=iPage End With Next iPage End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Pfad- und Arbeitsblattname in Fußzeile eintragen

Problem:	Wie kann ich automatisch bei jedem Druck den kompletten Pfad und den jeweiligen Tabellenblattnamen in die Fußzeile eintragen lassen? Änderungen des Blattnamens und Verschiebungen der Arbeitsmappe in ein anderes Verzeichnis sollen berücksichtigt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforePrint(Cancel As Boolean) Dim wks As Worksheet For Each wks In Worksheets wks.PageSetup.LeftFooter = ThisWorkbook.FullName _ & " - " & wks.Name Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die Druckblätter einer Arbeitsmappe durchnummerieren

Problem:	Wie kann ich beim Ausdruck aller Blätter einer Arbeitsmappe die einzelnen Blätter durchnummerieren lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Seitenumbruch() Dim wks As Worksheet Dim iCount As Integer, iCounter As Integer, iPage As Integer For Each wks In ThisWorkbook.Worksheets wks.Activate wks.PageSetup.PrintArea = wks.UsedRange.Address iCount = iCount + ExecuteExcel4Macro("Get.Document(50)") Next wks For Each wks In ThisWorkbook.Worksheets wks.Activate For iPage = 1 To ExecuteExcel4Macro("Get.Document(50)") iCounter = iCounter + 1 wks.PageSetup.LeftHeader = "Blatt " & _ iCounter & " von " & iCount & " Blatt" wks.PrintOut from:=iPage, to:=iPage Next iPage Next wks Worksheets(1).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fußzeile mit durchgehenden Seitennummern über alle Tabellen

Problem:	Wie kann ich alle Arbeitblätter einer Arbeitsmappe drucken lassen und in den Fußzeilen über alle Tabellen fortlaufende Blattnummern mit der Gesamtblattanzahl angeben lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AlleSeiten() Dim wks As Worksheet Dim iCountW As Integer, iCountP As Integer, iAct As Integer Application.ScreenUpdating = False For Each wks In ThisWorkbook.Worksheets wks.Select iCountW = iCountW + ExecuteExcel4Macro("INDEX(GET.DOCUMENT(50),1)") Next wks For Each wks In ThisWorkbook.Worksheets wks.Select For iAct = 1 To ExecuteExcel4Macro("INDEX(GET.DOCUMENT(50),1)") iCountP = iCountP + 1 wks.PageSetup.LeftFooter = _ "Blatt " & iCountP & " von " & _ iCountW & " Blatt" wks.PrintOut From:=iAct, To:=iAct Next iAct Next wks Worksheets("Tabelle1").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datum im Format TTTT, TT. MMMM JJJJ in Fußzeile eingeben

Problem: Wie kann ich das aktuelle Datum im Format TTTT, TT. MMMM JJJJ in die Fußzeile eingeben?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub LangesDatum()
    Dim sDate As String
    sDate = Format(Date, "dddd, dd. mmmm yyyy")
    ActiveSheet.PageSetup.LeftFooter = sDate
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopf-/Fußzeilen-Festlegungen übernehmen

Problem:	Wie kann ich die Kopf-/Fußzeilenfestlegungen des einen Tabellenblattes in ein zweites übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub KFZeilenUebernehmen() Dim psQuelle As PageSetup, psZiel As PageSetup Set psQuelle = Worksheets("Tabelle1").PageSetup Set psZiel = Worksheets("Tabelle2").PageSetup With psZiel .LeftHeader = psQuelle.LeftHeader .CenterHeader = psQuelle.CenterHeader .RightHeader = psQuelle.RightHeader .LeftFooter = psQuelle.LeftFooter .CenterFooter = psQuelle.CenterFooter .RightFooter = psQuelle.RightFooter End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Letztes Speicherdatum in die Fußzeile aller Tabellenblätter

Problem:	Wie kann ich das letzte Speicherdatum in die Fußzeile aller Tabellenblätter eintragen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeSave(_ ByVal SaveAsUI As Boolean, Cancel As Boolean) Dim wks As Worksheet For Each wks In ThisWorkbook.Worksheets wks.PageSetup.LeftFooter = _ "Gespeichert am: " & Format(Date, "dd.mm.yyyy") Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Seitennummer und Seitenanzahl per VBA in Kopfzeile

Problem: Wie kann ich Seitenanzahl und Seitennummer per VBA in die Kopfzeile eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub KopfZeile()
    With ActiveSheet
        .PageSetup.LeftHeader = "Seite &P von &N"
        .PrintPreview
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Unterschiedliche Schriftformatierung in Kopf- / Fußzeile

Problem:	Wie kann ich Eintragungen in der Kopf- bzw. Fußzeile in verschiedenen Schriftgrößen bzw. fett/nicht fett darstellen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Druck() With ActiveSheet .PageSetup.LeftHeader = "&B&14 Hallo, " _ & "&B&10 ich bin da!" .PrintPreview End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei- incl. Pfadname in Kopfzeile

Problem:	Wie kann ich den kompletten Dateinamen - als mit Pfad - in die Kopfzeile eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FusszeileVollerName() ActiveSheet.PageSetup.LeftHeader = _ ThisWorkbook.FullName ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datum im Format DD.MM.JJJJ in Fußzeile

Problem:	Wie kann ich das aktuelle Datum im Format DD.MM.JJJJ automtisch in die Fußzeile eingeben lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub DatumInFussZeile() ActiveSheet.PageSetup.LeftFooter = _ Format(Date, "dd.mm.yyyy") ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anwendername in Fußzeile eintragen

Problem:	Wie kann ich den Anwendernamen in die Fußzeile eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Verfasser() With Worksheets("Tabelle1") .PageSetup.LeftFooter = Application.UserName .PrintPreview End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über VBA Schriftgröße von Fußzeilen bestimmen

Problem:	Wie kann ich per VBA die Schriftgröße einer Fußzeile bestimmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Druck() Dim sTxt As String sTxt = ActiveWorkbook.FullName With ActiveSheet .PageSetup.LeftFooter = "&24" & sTxt .PrintPreview End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fußzeilen beim Öffnen eintragen

Problem:	Wie kann ich in die Fußzeile des 1. Arbeitsblattes jeder Arbeitsmappe, die ich öffne, den Pfad, die Seitenzahl und den UserNamen eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Oeffnen() Const csFilter As String = "Excel-Dateien(*.xls), *.xls" Dim vFile As Variant vFile = Application.GetOpenFilename(dFilter) If vFile = False Then Exit Sub Workbooks.Open vFile With ActiveWorkbook.ActiveSheet.PageSetup .LeftFooter = ActiveWorkbook.Path .CenterFooter = "&N" .RightFooter = Application.UserName End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Randeinstellung der Druck-Kopfzeile

Problem: Wie kann ich über Menü DATEI / SEITE EINRICHTEN den linken Rand der rechten Kopfzeile festlegen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub KopfzeilenRand()
    Dim sRow As String
    sRow = String(35, " ") & "MeineKopfzeile"
    Worksheets("Tabelle1").PageSetup.LeftHeader = sRow
    Worksheets("Tabelle1").PrintPreview
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Kopf- / Fußzeilen-Festlegungen löschen

Problem:	Wie kann ich über VBA alle Einstellungen für Kopf und Fußzeilen in der gesamten Arbeitsmappe löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub KopfFussWeg() Dim wks As Worksheet For Each wks In ThisWorkbook.Worksheets With wks.PageSetup .LeftHeader = "" .CenterHeader = "" .RightHeader = "" .LeftFooter = "" .CenterFooter = "" .RightFooter = "" End With Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopfzeile in Buchform drucken

Problem: Bei Büchern sind die Kopfzeilen in der Regel jeweils links und rechts angeordnet. Wie kann ich dieses Layout bei Excel übernehmen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub MyOldHeader()
    Dim iCount As Integer, iCounter As Integer
    Dim bln As Boolean
    bln = True
    iCount = ExecuteExcel4Macro("get.document(50)")
    For iCounter = 1 To iCount
        With ActiveSheet.PageSetup
            If bln = True Then
                .LeftHeader = "Header"
                .RightHeader = ""
            Else
                .LeftHeader = ""
                .RightHeader = "Header"
            End If
        End With
        ActiveWindow.SelectedSheets.PrintOut _
            From:=iCounter, To:=iCounter, Copies:=1
        bln = Not bln
    Next iCounter
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrzeilige Kopf- / Fußzeile

Problem:	Wie kann ich über ein VBA-Makro 3 Linien in die Kopf- und 2 Linien in die Fußzeile einfügen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basmain Sub LinesInHeFo() Dim sSep As String, sText As String sSep = Application.Rept("_", 20) With ActiveSheet.PageSetup sText = sSep & vbCrLf & sSep .CenterFooter = sText sText = sText & vbCrLf & sSep .CenterHeader = sText End With ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ein- und Auslesen der Seiteneinrichtung / Orientierung

Problem:	In UserForm-Optionsfelder soll die Seiteneinrichtungsfestlegung für die Orientierung ein und ausgelesen werden.
Lösung:	Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.
VBA-Code:	<pre>ClassModule: frmOrientation Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() If optPortrait.Value = True Then ActiveSheet.PageSetup.Orientation = xlPortrait Else ActiveSheet.PageSetup.Orientation = xlLandscape End If Unload Me End Sub Private Sub UserForm_Initialize() If ActiveSheet.PageSetup.Orientation = xlPortrait Then optPortrait.Value = True Else optLandscape.Value = True End If End Sub StandardModule: basMain Sub CallForm() frmOrientation.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckseite des aktuellen und aller Blätter ermitteln

Problem: Die Druckseite des aktiven Blattes und die Anzahl der Gesamtdruckblätter der Arbeitsmappe sollen ermittelt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub GetPageIdx()  
    Dim wksStart As Worksheet  
    Dim wks As Worksheet  
    Set wksStart = ActiveSheet  
    Application.ScreenUpdating = False  
    Dim iPage As Integer, iCount As Integer  
    For Each wks In Worksheets  
        wks.Select  
        iCount = iCount + ExecuteExcel4Macro("INDEX(GET.DOCUMENT(50),1)")  
        If wks.Name = wksStart.Name Then iPage = iCount  
    Next wks  
    wksStart.Select  
    MsgBox "Blatt " & iPage & " von " & iCount  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur die letzte Seite drucken

Problem:	Nur die letzte Seite soll gedruckt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub PrintLastSite() Dim iSite As Integer iSite = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(50),1)") ActiveSheet.PrintOut from:=iSite, to:=iSite End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckdaten auf Druckblatt sammeln

Problem:	Daten aus dem Quellblatt sollen so in das als Druckvorlage dienende Sammelblatt übernommen werden, dass auf jeder Druckseite der Bereich C7:D37 als Kopfzeile und danach Gruppen von Spalten gesammelt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Foto() Dim wks As Worksheet Set wks = ActiveSheet With Worksheets("Quellblatt") .Range("C7:D37").CopyPicture Appearance:=xlScreen, Format:=xlBitmap ActiveSheet.Paste Destination:=Worksheets("Sammelblatt").Range("A1") .Range("E7:L37").CopyPicture Appearance:=xlScreen, Format:=xlBitmap wks.Paste Destination:=Range("A32") wks.PrintPreview wks.Shapes(wks.Shapes.Count).Delete .Range("O7:Y37").CopyPicture Appearance:=xlScreen, Format:=xlBitmap wks.Paste Destination:=Range("A32") wks.PrintPreview wks.Shapes(wks.Shapes.Count).Delete .Range("AB7:AB37").CopyPicture Appearance:=xlScreen, Format:=xlBitmap wks.Paste Destination:=Range("A32") wks.PrintPreview End With wks.Shapes(wks.Shapes.Count).Delete wks.Shapes(wks.Shapes.Count).Delete End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckseiten und Anzahl der Ausdrücke über UserForm festlegen

Problem: Wie kann ich die zu druckenden Seiten und die Anzahl der Ausdrücke über einen UserForm-Dialog festlegen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: modMain

Sub CallForm()
    frmDrucken.Show
End Sub

ClassModule: frmDrucken

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdDrucken_Click()
    ActiveSheet.PrintOut _
        From:=CInt(txtFrom.Text), _
        To:=CInt(txtTo.Text), _
        Copies:=CInt(txtCopies.Text)
    Unload Me
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blatt auswählen und Zellen nacheinander drucken

Problem:	Über eine InputBox soll ein Arbeitsblattname abgefragt werden. In dem ausgewählten Blatt soll ein vorgegebener Bereich auf Inhalt geprüft und - wenn nicht leer - zellenweise gedruckt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SerieDrucken() Dim rng As Range Dim swks As String swks = InputBox("Tabellenblatt:", , "Ausgaben") If swks = "" Then Exit Sub With Worksheets(swks) For Each rng In .Range("A1:A5").Cells If Not IsEmpty(rng) Then Range("A1").Value = rng.Value ActiveSheet.PrintPreview End If Next rng End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe öffnen, erstes Blatt drucken, Mappe schließen

Problem:	Wie kann ich über VBA eine Arbeitsmappe öffnen, das erste Blatt drucken und die Mappe wieder schließen, ohne daß sie angezeigt wird?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub OeffnenDruckenSchliessen() Dim vFile As Variant Application.ScreenUpdating = False Application.EnableEvents = False On Error GoTo ERRORHANDLER vFile = Application.GetOpenFilename(_ "Excel-Arbeitsmappe (*.xls), *.xls") If vFile = False Then GoTo ERRORHANDLER Workbooks.Open vFile Worksheets(1).PrintOut ActiveWorkbook.Close savechanges:=False ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Erste Druckseite in eine neue Arbeitsmappe kopieren

Problem: Wie kann ich exakt die erste Druckseite des aktiven Blattes in eine neue Arbeitsmappe kopieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub ErsteSeiteKopieren()  
    Dim rng As Range  
    Dim iRow As Integer, iCol As Integer, iCounter As Integer  
    Application.ScreenUpdating = False  
    iRow = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(64),1)") - 1  
    iCol = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(65),1)") - 1  
    Set rng = Range(Cells(1, 1), Cells(iRow, iCol))  
    Workbooks.Add  
    rng.Copy Range("A1")  
    For iCounter = 1 To rng.Columns.Count  
        Columns(iCounter).ColumnWidth = rng.Columns.ColumnWidth  
    Next iCounter  
    For iCounter = 1 To rng.Rows.Count  
        Rows(iCounter).RowHeight = rng.Rows.RowHeight  
    Next iCounter  
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textreihe auf mehrere Spalten mit fester Seitenlänge verteilen

Problem: Wie kann ich eine feste Anzahl von Zeilen in einer Spalte auf mehrere Spalten mit einer festgelegten Seitenlänge verteilen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub TextInSpalten()  
    Dim rng As Range  
    Dim iRow As Integer, iCountR As Integer, iCol As Integer  
    Application.ScreenUpdating = False  
    Set rng = Range("A1").CurrentRegion  
    iCountR = 55  
    Workbooks.Add  
    For iRow = 1 To rng.Rows.Count Step iCountR  
        iCol = iCol + 1  
        rng.Range(rng.Cells(iRow), _  
            rng.Cells(iRow + iCountR - 1)).Copy Cells(1, iCol)  
    Next iRow  
    ActiveSheet.PrintPreview  
    ActiveWorkbook.Close savechanges:=False  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten von mehreren Blättern auf ein Druckblatt

Problem: Wie kann ich die Daten aus mehreren Tabellenblättern auf einem Druckblatt sammeln?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub PrintPlus()
    Dim rng As Range
    Dim iWks As Integer, iRow As Integer
    Workbooks.Add 1
    iRow = 1
    For iWks = 2 To 4
        Cells(iRow, 1).Value = "Blatt " & _
            ThisWorkbook.Worksheets(iWks).Name
        Set rng = ThisWorkbook.Worksheets(iWks) _
            .Range("A1").CurrentRegion
        rng.Copy Cells(iRow + 1, 1)
        iRow = iRow + rng.Rows.Count + 2
    Next iWks
    ActiveSheet.PrintPreview
    ActiveWorkbook.Close savechanges:=False
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die Druckseite drucken, in der sich der Cursor befindet

Problem:	Wie kann ich die Druckseite drucken lassen, in der sich aktuell der Cursor befindet?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub PrintActPage() Dim varPB As Variant Dim iPage As Integer, iHPage As Integer, iVPage As Integer Do Until ActiveCell.Row + 1 <= varPB iHPage = iHPage + 1 varPB = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(64)," & iHPage & ")") If IsError(varPB) Then Exit Do Loop varPB = 0 Do Until IsError(varPB) iPage = iPage + 1 varPB = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(64)," & iPage & ")") If IsError(varPB) Then Exit Do Loop varPB = 0 Do Until ActiveCell.Column + 1 <= varPB iVPage = iVPage + 1 varPB = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(65)," & iVPage & ")") If IsError(varPB) Then Exit Do Loop iPage = (iPage * (iVPage - 1)) + iHPage ActiveSheet.PrintOut from:=iPage, to:=iPage End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druck aus mehreren Tabellenblättern auf eine Druckseite

Problem:	Wie kann ich in XL8 Bereiche aus mehreren Arbeitsblättern auf eine Druckseite zusammenfassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Foto() Dim iWks As Integer, iRow As Integer iRow = 1 For iWks = 3 To 5 Worksheets(iWks).Range("A1:D7").CopyPicture _ Appearance:=xlScreen, _ Format:=xlBitmap Worksheets("Sammelblatt").Paste _ Destination:=Worksheets("Sammelblatt").Cells(iRow, 1) ActiveSheet.Shapes(iWks - 1).Select Selection.Formula = Worksheets(iWks).Name & "!A1:D7" iRow = iRow + 8 Next iWks Range("A1").Select ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachauswahl im Hoch- und Querformat drucken

Problem:	Wie kann ich über VBA den 1. Bereich einer Mehrfachauswahl im Hoch-, den 2. Im Querformat drucken?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Drucken() Dim rngA As Range, rngB As Range Set rngA = Selection.Areas(1) Set rngB = Selection.Areas(2) With ActiveSheet.PageSetup .PrintArea = rngA.Address .Orientation = xlPortrait ActiveSheet.PrintPreview .PrintArea = rngB.Address .Orientation = xlLandscape ActiveSheet.PrintPreview End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckseitenlinien im Tabellenblatt

Problem: Nach dem Druck- oder dem Vorschaubefehl erscheinen im Tabellenblatt gestrichelte Rahmen um die Druckseiten. Wie kann ich das verhindern?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub PrintMarker()
    With ActiveSheet
        .PrintPreview
        .DisplayAutomaticPageBreaks = False
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Seitenanzahl eintragen

Problem: Wie kann ich in einer Zelle die Seitenanzahl des aktiven Arbeitsblattes eintragen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub PageCount()
    Range("B1").Value = _
        ExecuteExcel4Macro("Get.Document(50)")
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckseite anpassen

Problem:	Wie kann ich den Druckbereich per VBA auf eine Seite festlegen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FitPage() With ActiveSheet.PageSetup .Zoom = False .FitToPagesWide = 1 .FitToPagesTall = 1 End With ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten sortieren und bedingt drucken

Problem:	Die Werte aus A:B aus den Datenblättern sollen sortiert, in das Druckblatt kopiert und dort gedruckt werden. Wenn der Schlüssel in Zelle E2 den Wert 0 hat, wird nicht gedruckt.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Drucken() Dim iWks As Integer Application.ScreenUpdating = False For iWks = 1 To 1 With Worksheets(CStr(iWks)) If .Range("E2") <> 0 Then Worksheets("Print").Cells.Clear .Columns("A:B").NumberFormat = "dd.mm.yyyy" .Range("A2").AutoFilter Field:=1, _ Criteria1:="=" & Format(CDate(.Range("E1").Value), "dd.mm.yyyy") .Range(.Cells(1, 1), .Cells(.Cells(1, 2).End(xlDown).Row, 2)).Copy _ Worksheets("Print").Range("A2") With Worksheets("Print") .Visible = True .PrintPreview .Visible = xlVeryHidden End With .AutoFilterMode = False End If End With Next iWks Application.ScreenUpdating = True End Sub Sub BlaetterEinAus() Worksheets("1").Visible = xlSheetVeryHidden Worksheets("Print").Visible = xlSheetVeryHidden 'Worksheets("1").Visible = xlSheetVisible 'Worksheets("Print").Visible = xlSheetVisible End Sub</pre>

	Herbers Excel-FAQ - Die Texte Bestandteil der Excel-Lehr- und Lernmaterialien Spendenbescheinigungen als Serienbrief drucken <table><tr><td>Problem:</td><td>An eine Adressliste sollen Spendenbescheinigungen gedruckt werden.</td></tr><tr><td>Lösung:</td><td>Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.</td></tr><tr><td>VBA-Code:</td><td> StandardModule: Modul1 Sub Bescheinigung() Dim iRow As Integer iRow = 2 With Worksheets("Adressen") Do Until IsEmpty(.Cells(iRow, 1)) Range("A8:A11").ClearContents Range("A8").Value = .Cells(iRow, 2).Value & " " & .Cells(iRow, 1).Value Range("A9").Value = .Cells(iRow, 3).Value Range("A11").Value = .Cells(iRow, 4).Value & " " & .Cells(iRow, 5).Value Range("A28").Value = .Cells(iRow, 2).Value & " " & .Cells(iRow, 1).Value Range("A29").Value = .Cells(iRow, 3).Value Range("A30").Value = .Cells(iRow, 4).Value & " " & .Cells(iRow, 5).Value Range("A36").Value = "XXX " & .Cells(iRow, 6).Value & " DM /" & _ ZahlWort(.Cells(iRow, 6).Value) & "/11.09.01 XXX" iRow = iRow + 1 ActiveSheet.PrintPreview Loop End With End Sub StandardModule: Modul2 Function ZWort(dZahl As Double, Optional bln As Boolean) Dim dRest As Double dRest = WorksheetFunction.Round((dZahl - Fix(dZahl)), 2) * 100 dZahl = Fix(dZahl) BisNeunzehn = Array("", "ein", "zwei", "drei", "vier", _ "fünf", "sechs", "sieben", "acht", "neun", "zehn", _ "elf", "zwölf", "dreizehn", "vierzehn", "fünfzehn", _ "sechzehn", "siebzehn", "achtzehn", "neunzehn") Zehner = Array("", "zehn", "zwanzig", "dreißig", _ "vierzig", "fünfzig", "sechzig", "siebzig", _ "achtzig", "neunzig") Tausender = Array("", "tausend", "millionen", "milliarden") If dRest = 0 Then ZWort = Text(dZahl) Else If bln Then ZWort = Text(dZahl) & " " & dRest & "/00" Else ZWort = Text(dZahl) End If End If End Function Private Function Wort(wert As Integer) As String Dim h As Integer h = wert Mod 100 If h < 20 Then Wort = BisNeunzehn(h) Else Wort = BisNeunzehn(h Mod 10) & IIf(h Mod 10 > 0, "und", "") & _ Zehner(Int(h / 10)) End If h = (wert Mod 1000 - h) / 100 If h > 0 Then Wort = BisNeunzehn(h) & "hundert" & Wort End Function Private Function Text(wert As Double) Dim l As Integer, i As Integer, p As Integer If InStr(1, Str(wert), ",") = 0 And InStr(1, Str(wert), ".") = 0 Then For i = 1 To 1 + Int(Len(Str(wert)) / 3) p = Val("0" & Mid("000" + Str(wert), _ Len("000" & Str(wert)) - i * 3 + 1, 3)) If p > 0 Then Text = Wort(p) & Tausender(i - 1) & Text Next Else Text = "#Ganzzahl!" End If If Right(Text, 3) = "ein" Then Text = Text & "s" End Function </td></tr></table>	Problem:	An eine Adressliste sollen Spendenbescheinigungen gedruckt werden.	Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.	VBA-Code:	StandardModule: Modul1 Sub Bescheinigung() Dim iRow As Integer iRow = 2 With Worksheets("Adressen") Do Until IsEmpty(.Cells(iRow, 1)) Range("A8:A11").ClearContents Range("A8").Value = .Cells(iRow, 2).Value & " " & .Cells(iRow, 1).Value Range("A9").Value = .Cells(iRow, 3).Value Range("A11").Value = .Cells(iRow, 4).Value & " " & .Cells(iRow, 5).Value Range("A28").Value = .Cells(iRow, 2).Value & " " & .Cells(iRow, 1).Value Range("A29").Value = .Cells(iRow, 3).Value Range("A30").Value = .Cells(iRow, 4).Value & " " & .Cells(iRow, 5).Value Range("A36").Value = "XXX " & .Cells(iRow, 6).Value & " DM /" & _ ZahlWort(.Cells(iRow, 6).Value) & "/11.09.01 XXX" iRow = iRow + 1 ActiveSheet.PrintPreview Loop End With End Sub StandardModule: Modul2 Function ZWort(dZahl As Double, Optional bln As Boolean) Dim dRest As Double dRest = WorksheetFunction.Round((dZahl - Fix(dZahl)), 2) * 100 dZahl = Fix(dZahl) BisNeunzehn = Array("", "ein", "zwei", "drei", "vier", _ "fünf", "sechs", "sieben", "acht", "neun", "zehn", _ "elf", "zwölf", "dreizehn", "vierzehn", "fünfzehn", _ "sechzehn", "siebzehn", "achtzehn", "neunzehn") Zehner = Array("", "zehn", "zwanzig", "dreißig", _ "vierzig", "fünfzig", "sechzig", "siebzig", _ "achtzig", "neunzig") Tausender = Array("", "tausend", "millionen", "milliarden") If dRest = 0 Then ZWort = Text(dZahl) Else If bln Then ZWort = Text(dZahl) & " " & dRest & "/00" Else ZWort = Text(dZahl) End If End If End Function Private Function Wort(wert As Integer) As String Dim h As Integer h = wert Mod 100 If h < 20 Then Wort = BisNeunzehn(h) Else Wort = BisNeunzehn(h Mod 10) & IIf(h Mod 10 > 0, "und", "") & _ Zehner(Int(h / 10)) End If h = (wert Mod 1000 - h) / 100 If h > 0 Then Wort = BisNeunzehn(h) & "hundert" & Wort End Function Private Function Text(wert As Double) Dim l As Integer, i As Integer, p As Integer If InStr(1, Str(wert), ",") = 0 And InStr(1, Str(wert), ".") = 0 Then For i = 1 To 1 + Int(Len(Str(wert)) / 3) p = Val("0" & Mid("000" + Str(wert), _ Len("000" & Str(wert)) - i * 3 + 1, 3)) If p > 0 Then Text = Wort(p) & Tausender(i - 1) & Text Next Else Text = "#Ganzzahl!" End If If Right(Text, 3) = "ein" Then Text = Text & "s" End Function
Problem:	An eine Adressliste sollen Spendenbescheinigungen gedruckt werden.						
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.						
VBA-Code:	StandardModule: Modul1 Sub Bescheinigung() Dim iRow As Integer iRow = 2 With Worksheets("Adressen") Do Until IsEmpty(.Cells(iRow, 1)) Range("A8:A11").ClearContents Range("A8").Value = .Cells(iRow, 2).Value & " " & .Cells(iRow, 1).Value Range("A9").Value = .Cells(iRow, 3).Value Range("A11").Value = .Cells(iRow, 4).Value & " " & .Cells(iRow, 5).Value Range("A28").Value = .Cells(iRow, 2).Value & " " & .Cells(iRow, 1).Value Range("A29").Value = .Cells(iRow, 3).Value Range("A30").Value = .Cells(iRow, 4).Value & " " & .Cells(iRow, 5).Value Range("A36").Value = "XXX " & .Cells(iRow, 6).Value & " DM /" & _ ZahlWort(.Cells(iRow, 6).Value) & "/11.09.01 XXX" iRow = iRow + 1 ActiveSheet.PrintPreview Loop End With End Sub StandardModule: Modul2 Function ZWort(dZahl As Double, Optional bln As Boolean) Dim dRest As Double dRest = WorksheetFunction.Round((dZahl - Fix(dZahl)), 2) * 100 dZahl = Fix(dZahl) BisNeunzehn = Array("", "ein", "zwei", "drei", "vier", _ "fünf", "sechs", "sieben", "acht", "neun", "zehn", _ "elf", "zwölf", "dreizehn", "vierzehn", "fünfzehn", _ "sechzehn", "siebzehn", "achtzehn", "neunzehn") Zehner = Array("", "zehn", "zwanzig", "dreißig", _ "vierzig", "fünfzig", "sechzig", "siebzig", _ "achtzig", "neunzig") Tausender = Array("", "tausend", "millionen", "milliarden") If dRest = 0 Then ZWort = Text(dZahl) Else If bln Then ZWort = Text(dZahl) & " " & dRest & "/00" Else ZWort = Text(dZahl) End If End If End Function Private Function Wort(wert As Integer) As String Dim h As Integer h = wert Mod 100 If h < 20 Then Wort = BisNeunzehn(h) Else Wort = BisNeunzehn(h Mod 10) & IIf(h Mod 10 > 0, "und", "") & _ Zehner(Int(h / 10)) End If h = (wert Mod 1000 - h) / 100 If h > 0 Then Wort = BisNeunzehn(h) & "hundert" & Wort End Function Private Function Text(wert As Double) Dim l As Integer, i As Integer, p As Integer If InStr(1, Str(wert), ",") = 0 And InStr(1, Str(wert), ".") = 0 Then For i = 1 To 1 + Int(Len(Str(wert)) / 3) p = Val("0" & Mid("000" + Str(wert), _ Len("000" & Str(wert)) - i * 3 + 1, 3)) If p > 0 Then Text = Wort(p) & Tausender(i - 1) & Text Next Else Text = "#Ganzzahl!" End If If Right(Text, 3) = "ein" Then Text = Text & "s" End Function						



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Seriendruck

Problem: Wie kann ich einen Seriendruck mit Adressdaten aus einer Adressdatei per VBA ablaufen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub SerienDruck()
    Dim wks As Worksheet
    Dim iRow As Integer
    Set wks = Worksheets("Datenbank")
    iRow = 1
    Do Until IsEmpty(wks.Cells(iRow, 1))
        Range("Adresse").ClearContents
        Range("Firma1") = wks.Cells(iRow, 2)
        Range("Firma2") = wks.Cells(iRow, 3)
        Range("Partner") = wks.Cells(iRow, 4)
        Range("Strasse") = wks.Cells(iRow, 5)
        Range("Ort") = wks.Cells(iRow, 6) & " " & _
            wks.Cells(iRow, 7)
        ActiveSheet.PrintPreview
        iRow = iRow + 1
    Loop
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenumbrüchen über VBA setzen

Problem:	Manuelle Zeilenumbrüche werden gesetzt, bereits vorhandene entfernt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetPageBreaks() Dim iRow As Integer, iCounter As Integer iRow = 56 With ActiveSheet .DisplayAutomaticPageBreaks = True For iCounter = .HPageBreaks.Count To 1 Step -1 If .HPageBreaks(iCounter).Type = xlPageBreakManual Then .HPageBreaks(iCounter).Delete End If Next iCounter Do While iRow < Range("A1").CurrentRegion.Rows.Count .HPageBreaks.Add Rows(iRow) iRow = iRow + 50 Loop .PrintPreview End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenumbruch bei Wertänderung

Problem:	Gemäß den Werten in Spalte A sollen die Seitenumbrüche gesetzt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetPageBreaks() Dim iRow As Integer iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) If Cells(iRow, 1).Value <> Cells(iRow - 1, 1).Value Then Cells(iRow, 1).PageBreak = xlPageBreakManual End If iRow = iRow + 1 Loop ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Seitenumbruch nach Zeilen mit bestimmtem Inhalt einfügen

Problem:	An den Zeilen mit dem Wert XYZ in Spalte A soll ein Seitenwechsel eingefügt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Seitenwechsel() Dim iRow As Integer For iRow = WorksheetFunction.CountA(Columns(1)) To 1 Step -1 If Cells(iRow, 1).Value = "XYZ" Then Tabelle1.HPageBreaks.Add Cells(iRow, 1) End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vor dem Drucken an den Seitenumbrüchen die Seitennummer eintragen

Problem:	In den jeweiligen Zellen der horizontalen Seitenwechsel sollen die Seitenzahlen vor dem Drucken eingetragen und danach gelöscht werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub DruckSeiten() Dim iRow As Integer, iRowL As Integer, iPage As Integer iRowL = Cells(Rows.Count, 1).End(xlUp).Row ActiveSheet.PageSetup.PrintArea = Range("A1:B" & iRowL).Address Rows(iRowL).Select ActiveSheet.DisplayAutomaticPageBreaks = True Range("B1").Value = "Seite 1" For iPage = 1 To ActiveSheet.HPageBreaks.Count ActiveSheet.HPageBreaks(iPage) _ .Location.Offset(0, 1).Value = "Seite " & iPage + 1 Next iPage ActiveSheet.PrintPreview Columns("B").ClearContents Range("A1").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Seitenumbruch ermitteln und Übertrag erstellen

Problem:	Wie kann ich in einer längeren Liste den Seitenumbruch ermitteln und in der Zelle rechts neben der Wertezelle in der letzten Zeile einen Übertrag eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Seitenumbruch() Dim varPB As Variant Dim iPage As Integer, iRowL As Integer iRowL = Cells(Rows.Count, 1).End(xlUp).Row iPage = 1 Do While IsError(varPB) = False varPB = ExecuteExcel4Macro("INDEX(GET.DOCUMENT(64)," & iPage & ")") If IsError(varPB) Then Exit Sub Else Cells(varPB - 1, 2) = Application.Sum(Range(Cells(1, 1), _ Cells(varPB - 1, 1))) End If iPage = iPage + 1 Cells(iRowL, 2) = Application.Sum(Range(Cells(1, 1), Cells(iRowL, 1))) Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vertikalen Seitenumbruch verhindern

Problem:	Wie kann ich bei einem in der Länge mehrseitigem Blatt vertikale Seitenumbrüche verhindern? Bei Wertänderung in der letzten Spalte soll jeweils ein horizontaler Seitenumbruch eingefügt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Drucken() Dim iCounter As Integer Application.ScreenUpdating = False With ActiveSheet .PageSetup.Zoom = 100 ActiveWindow.View = xlPageBreakPreview Do While .VPageBreaks.Count > 0 .PageSetup.Zoom = .PageSetup.Zoom - 1 ActiveWindow.View = xlNormalView ActiveWindow.View = xlPageBreakPreview Loop iCounter = 2 Do Until IsEmpty(Cells(iCounter, 12)) If Cells(iCounter, 12).Value <> _ Cells(iCounter - 1, 12).Value Then .HPageBreaks.Add Before:=Cells(iCounter, 12) End If iCounter = iCounter + 1 Loop .PrintPreview End With ActiveWindow.View = xlNormalView Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm drucken

Problem: Wie kann ich eine UserForm drucken?

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: frmPrintAndMsg

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdPrintForm_Click()
    Me.PrintForm
End Sub

StandardModule: basMain

Sub CallForm()
    frmPrintAndMsg.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druck mit und ohne Wasserzeichen

Problem:	Die Tabelle soll einmal normal und einmal mit dem Wasserzeichen "Duplikat" gedruckt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub PrintWithDuplicate() With ActiveSheet .Shapes("Duplikat").Visible = False .PrintOut .Shapes("Duplikat").Visible = True .PrintOut .Shapes("Duplikat").Visible = False End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckzoom auf das Maximum hochfahren

Problem:	Wie kann ich den Druckzoom auf das Maximum einer Seite hochfahren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZoomUp() Dim iPages As Integer, iZoom As Integer iZoom = 100 iPages = ExecuteExcel4Macro("Get.Document(50)") With ActiveSheet .PageSetup.Zoom = 95 Do Until iPages > 1 iZoom = iZoom + 5 .PageSetup.Zoom = iZoom iPages = ExecuteExcel4Macro("Get.Document(50)") Loop .PageSetup.Zoom = iZoom - 5 .PrintPreview End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zwischensumme in Fußzeile

Problem:	Wie kann ich die Summe in die Fußzeile übernehmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Seitenumbruch() Dim varPB As Variant Dim iPage As Integer, iRowL As Integer Dim dSumA As Currency, dSumB As Currency iRowL = Cells(Rows.Count, 2).End(xlUp).Row iPage = 1 Do Until IsError(varPB) varPB = ExecuteExcel4Macro(_ "INDEX(GET.DOCUMENT(64)," & iPage & ")") If IsError(varPB) Then Exit Do dSumA = WorksheetFunction.Sum(Range(_ Cells(1, 2), Cells(varPB - 1, 2))) With ActiveSheet.PageSetup .LeftHeader = "Zwischentotal:" .CenterHeader = Format(dSumB, "#,##0.00" & " €") .LeftFooter = "Zwischentotal:" .CenterFooter = Format(dSumA, "#,##0.00" & " €") End With ActiveSheet.PrintOut From:=iPage, To:=iPage dSumB = dSumA iPage = iPage + 1 Loop dSumB = dSumA dSumA = WorksheetFunction.Sum(Range(_ Cells(1, 2), Cells(iRowL, 2))) With ActiveSheet.PageSetup .LeftHeader = "Zwischentotal:" .CenterHeader = Format(dSumB, "#,##0.00" & " €") .LeftFooter = "Zwischentotal:" .CenterFooter = Format(dSumA, "#,##0.00" & " €") End With ActiveSheet.PrintOut From:=iPage, To:=iPage End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Rechnung mit Überträgen

Problem:	Wie kann ich eine mehrseitige Leistungsliste beim Drucken auf ein Erste-Seite- und mehrere Folgeseiten-Rechnungsformular mit der Errechnung von Überträgen drucken lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub RechnungSchreiben() Dim wksSource As Worksheet, wksSecond As Worksheet Dim dValue As Double Dim iRowL As Integer, iPrint As Integer, iCount As Integer Dim iPage As Integer Set wksSource = Worksheets("Eingabe") Set wksSecond = Worksheets("RechnungFF") Range("A29:G54").ClearContents iRowL = wksSource.Cells(Rows.Count, 1).End(xlUp).Row Range("A29:A54").Value = wksSource.Range("A2:A27").Value Range("F29:F54").Value = wksSource.Range("C2:C27").Value Range("G29:G54").Value = wksSource.Range("B2:B27").Value If iRowL < 27 Then Call SetBottom(ActiveSheet, True) ActiveSheet.PrintPreview Else Call SetBottom(ActiveSheet, False) ActiveSheet.PrintPreview dValue = Range("H55").Value iPage = 1 For iPrint = 28 To iRowL Step 47 With wksSecond .Range("A6, A8:G53, H7").ClearContents If iPrint + 47 <= iRowL Then Call SetBottom(wksSecond, False) Else Call SetBottom(wksSecond, True) End If iPage = iPage + 1 .Range("A6").Value = "Blatt " & iPage & _ " von " & Range("D26").Value .Range("H7").Value = dValue .Range("A8:A54").Value = wksSource.Range(_ "A" & iPrint & ":A" & iPrint + 46).Value .Range("F8:F54").Value = wksSource.Range(_ "C" & iPrint & ":C" & iPrint + 46).Value .Range("G8:G54").Value = wksSource.Range(_ "B" & iPrint & ":B" & iPrint + 46).Value .PrintPreview dValue = .Range("H55").Value End With Next iPrint End If End Sub Private Sub SetBottom(wks As Worksheet, bln As Boolean) If bln Then wks.Range("A55").Value = "Nettobetrag:" wks.Range("A56").Value = "Mwst.:" wks.Range("A57").Value = "Rechnungsbetrag:" wks.Range("G56").Value = Worksheets("Data").Range("Mwst").Value wks.Range("H56").FormulaR1C1 = "=R[-1]C*RC[-1]" wks.Range("H57").FormulaR1C1 = "=R[-2]C+R[-1]C" With wks.Range("A57:H57") .Borders(xlEdgeTop).Weight = xlThin .Borders(xlEdgeBottom).LineStyle = xlDouble End With Else wks.Range("A55").Value = "Übertrag:" With wks.Range("A56:H57") .ClearContents .Borders.LineStyle = xlLineStyleNone End With End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckbereich alternativ festlegen

Problem:	Wie kann ich den Druckbereich alternativ festlegen
Lösung:	Kann nur anhand einer Beispielarbeitsmappe dargestellt werden.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Deaktivieren und Aktivieren von Drag&Drop

Problem:	Bei Aktivierung dieser Arbeitsmappe soll das Drag&Drop-Feature ab-, beim Deaktivieren wieder auf den ursprünglichen Zustand zurückgesetzt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Public gbln As Boolean Private Sub Workbook_Activate() gbln = Application.CellDragAndDrop Application.CellDragAndDrop = False End Sub Private Sub Workbook_Deactivate() On Error Resume Next Application.CellDragAndDrop = gbln If Err > 0 Then Application.CellDragAndDrop = True End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Aktivierung einer Arbeitsmappe ShortCut zuweisen

Problem:	Bei Aktivierung dieser Arbeitsmappe soll dem ShortCut Strg+x ein Makro zugewiesen und bei Deaktivierung soll die Zuweisung wieder entfernt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Activate() Application.MacroOptions macro:="MeinMakro", _ Description:="", ShortcutKey:="x" End Sub Private Sub Workbook_Deactivate() Application.MacroOptions macro:=ThisWorkbook.Name & "!MeinMakro", _ Description:="", ShortcutKey:="" End Sub StandardModule: basMain Sub MeinMakro() MsgBox "Ich bin ein Makro!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Aktivierung einer bestimmten Tabelle in Vollbildmodus schalten

Problem: Wie kann ich bei der Aktivierung einer bestimmten Tabelle in den Vollbildmodus und beim Verlassen wieder zurück schalten?

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul von Tabelle2 ein

VBA-Code:

```
ClassModule: Tabelle3

Private Sub Worksheet_Activate()
    Application.DisplayFullScreen = True
End Sub

Private Sub Worksheet_Deactivate()
    Application.DisplayFullScreen = False
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eintrag von Werten in verschiedene Arbeitsmappen

Problem:	Beim Schließen der Arbeitsmappe sollen die Zeilen in Abhängigkeit der Werte in Spalte G in verschiedene Arbeitsmappen eingetragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Dim wks As Worksheet Dim iRow As Integer, iRowL As Integer, iRowT As Integer Dim sPath As String, sNo As String, sFile As String, sNoOld As String sPath = Application.DefaultFilePath & "\" If MsgBox("Verteilung vornehmen?", vbQuestion + vbYesNo) = vbNo Then Exit Sub Application.ScreenUpdating = False Worksheets("Tabelle1").Select Set wks = ActiveSheet Range("A1").CurrentRegion.Sort key1:=Range("G2"), order1:=xlAscending, header:=xlYes iRowL = wks.Cells(Rows.Count, 1).End(xlUp).Row For iRow = 2 To iRowL sNoOld = sNo Select Case wks.Cells(iRow, 7).Value Case Is <= 2500 sNo = "1" Case Is <= 5000 sNo = "2" Case Is <= 7500 sNo = "3" Case Else sNo = "4" End Select If wks.Cells(iRow, 7).Value <> Cells(iRow - 1, 7).Value Then sFile = "Pruefen_" & sNo & ".xls" If Dir(sPath & sFile) <> "" Then Workbooks.Open sPath & sFile Else Workbooks.Add 1 wks.Range("A1:H1").Copy Range("A1") End If End If Do While Int(wks.Cells(iRow, 7).Value / 2500) = Int(wks.Cells(iRow + 1, 7).Value / 2500) Or iRow = iRowL iRowT = Cells(Rows.Count, 1).End(xlUp).Row + 1 Range(Cells(iRowT, 1), Cells(iRowT, 8)).Value = wks.Range(wks.Cells(iRow, 1), wks.Cells (iRow, 8)).Value iRow = iRow + 1 If iRow > iRowL Then Exit Do Loop ActiveWorkbook.SaveAs sPath & sFile ActiveWorkbook.Close savechanges:=False Next iRow Application.ScreenUpdating = True End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelklick-Feature für Tabelle und Symbolleisten ausschalten

Problem:	Die Doppelklick-Aktionen sollen ausgeschaltet werden. Dies soll sich auch auf den Doppelklick im Symbolleistenbereich beziehen (erst ab XL8 SR2).
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call DoppelklickEinschalten End Sub StandardModule: basMain Sub DoppelklickAusschalten() Application.OnDoubleClick = "Ausschalten" End Sub Sub DoppelklickEinschalten() Application.OnDoubleClick = "" End Sub Sub Ausschalten() End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Speichernabfrage beim Schließen der Arbeitsmappe

Problem: Wie kann ich bei jedem Schließen einer bestimmten Arbeitsmappe abfragen, ob die Datei unter dem in Zelle A1 stehenden Namen gespeichert werden soll? Wenn die Datei bereits besteht, soll eine Sicherheitsabfrage erfolgen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Dim vSave As Variant
    Dim sFile As String
    sFile = ThisWorkbook.Worksheets("Tabelle1").Range("B1").Value
    If Right(sFile, 3) <> "xls" Then sFile = sFile & ".xls"
    If Dir(sFile) <> "" Then
        vSave = MsgBox("Datei überschreiben?", vbYesNoCancel)
        If vSave = vbYes Then
            Application.DisplayAlerts = False
            ThisWorkbook.SaveAs sFile
            Application.DisplayAlerts = True
        ElseIf vSave = vbCancel Then
            Cancel = True
        Else
            Cancel = False
        End If
    Else
        ThisWorkbook.SaveAs sFile
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Schließen der Arbeitsmappe Datum und Zeit protokollieren

Problem: Wie kann ich bei jedem Schließen einer Arbeitsmappe in einem Arbeitsblatt dieser Mappe Datum und Zeit protokollieren lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Dim iRow As Integer
    With Worksheets("Tabelle1")
        iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
        .Cells(iRow, 1) = Date
        .Cells(iRow, 2) = Time
    End With
    On Error GoTo ERRORHANDLER
    ThisWorkbook.Save
    Exit Sub
ERRORHANDLER:
    MsgBox "Die Datei konnte nicht gespeichert werden!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick Buchung vornehmen

Problem:	Bei Doppelklick auf eine Buchungszeile soll der Bereich A:G dieser Zeile in die erste freie Zeile des in Spalte A genannten Kontos kopiert werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean) Dim wks As Worksheet Dim iRow As Integer If Target.Row = 1 Then Exit Sub If IsEmpty(Cells(Target.Row, 1)) Then Exit Sub On Error Resume Next Set wks = Worksheets(CStr(Cells(Target.Row, 1).Value)) If Err > 0 Then Err.Clear Else iRow = wks.Cells(wks.Rows.Count, 1).End(xlUp).Row + 1 wks.Range(wks.Cells(iRow, 1), wks.Cells(iRow, 7)).Value = Range(Cells(Target.Row, 1), Cells (Target.Row, 7)).Value End If End Sub Private Sub Worksheet_SelectionChange(ByVal Target As Range) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick eine Tabelle sortieren

Problem:	Bei Doppelklick auf ein Datum in Spalte A soll die Tabelle im Blatt Termine so sortiert werden, dass das doppelgeklickte Datum oben steht.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean) Dim wks As Worksheet Dim vRow As Variant, iRow As Integer If Target.Column <> 1 Then Exit Sub If IsEmpty(Target) Then Exit Sub Cancel = True Set wks = Worksheets("Termine") vRow = Application.Match(CDbl(Target.Value), wks.Columns(1), 0) If IsError(vRow) Then Beep MsgBox "An diesem Tag keinen Termin gefunden!" Exit Sub End If iRow = 2 Do Until IsEmpty(wks.Cells(iRow, 1)) If CDbl(wks.Cells(iRow, 1).Value) = CDbl(Target.Value) Then wks.Cells(iRow, 3).Value = 1 Else wks.Cells(iRow, 3).Value = 2 End If iRow = iRow + 1 Loop wks.Range("A1").CurrentRegion.Sort key1:=wks.Range("C2"), order1:=xlAscending, header:=xlYes wks.Columns(3).ClearContents Application.Goto wks.Range("A1"), True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick Hilfetext als Kommentar anzeigen.

Problem: Bei Doppelklick auf eine Zelle soll aus einer Textdatei ein zugehöriger Hilfetext aus- und in einen Kommentar eingelesen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
    Dim cmt As Comment
    Dim sHelp As String, sTxt As String, sFile As String
    If Intersect(Target, Range("A1:C1")) Is Nothing Then Exit Sub
    Cancel = True
    sFile = ThisWorkbook.Path & "\12help" & Target.Column & ".txt"
    If Dir(sFile) = "" Then
        sHelp = "Keine Hilfedatei gefunden!"
    Else
        Close
        Open sFile For Input As #1
        Do Until EOF(1)
            Line Input #1, sTxt
            sHelp = sHelp & sTxt & vbCrLf
        Loop
        Close
    End If
    If Not ActiveCell.Comment Is Nothing Then
        ActiveCell.Comment.Delete
    End If
    Application.DisplayCommentIndicator = xlCommentAndIndicator
    Set cmt = ActiveCell.AddComment(sHelp)
    cmt.Shape.TextFrame.AutoSize = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick AcrobatReader mit Datei aufrufen

Problem: Aus Doppelklick auf Zelle B2 soll Acrobat Reader mit der in Zelle B2 genannten Datei aufgerufen werden. Die Routine soll auch in Excel 7.0 funktionieren.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
StandardModule: basMain

Sub auto_open()
    ActiveSheet.OnDoubleClick = "CallReader"
End Sub

Sub auto_close()
    ActiveSheet.OnDoubleClick = ""
End Sub

Sub CallReader()
    If ActiveCell.Address <> "$B$2" Then Exit Sub
    If Dir(Range("B1").Value) = "" Or Dir(Range("B2").Value) = "" Then
        Beep
        MsgBox "Bitte die Pfade und Dateinamen überprüfen!"
        Exit Sub
    End If
    Shell Chr(34) & Range("B1").Value & Chr(34) & _
        Chr(34) & Range("B2").Value & Chr(34), vbNormalFocus
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

DoubleClick-Ereignis für alle Arbeitsmappen initialisieren

Problem:	Das Doppelklickereignis soll für alle geöffneten Arbeitsmappe gelten.
Lösung:	Geben Sie den nachfolgenden Code in das genannte Klassen- und das Standardmodul ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Set MyApp.xlApp = application End Sub ClassModule: clsApp Public WithEvents xlApp As application Private Sub xlApp_SheetBeforeDoubleClick(_ ByVal Sh As Object, _ ByVal Target As Range, _ Cancel As Boolean) Cancel = True MsgBox "Zelle: " & Target.Address(False, False) & vbLf & _ "Arbeitsblatt: " & Sh.Name & vbLf & _ "Arbeitsmappe " & Sh.Parent.Name End Sub StandardModule: basMain Public MyApp As New clsApp</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick Gebühren aus Tabelle anzeigen

Problem:	Bei Doppelklick auf Erwachsener oder Jugendlicher soll in Abhängigkeit der aktuellen Uhrzeit eine MessageBox mit den Gebühren der Tabelle gezeigt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean) Dim cValue As Currency If Intersect(Target, Range("D2:D3")) Is Nothing Then Exit Sub Cancel = True Select Case Time Case Is >= 22 / 24 MsgBox "Kein Telefonat möglich!" Case Is >= 18 / 24 MsgBox "Gebühren: " & Cells(Target.Row, 3).Text Case Is >= 14 / 24 MsgBox "Gebühren: " & Cells(Target.Row, 2).Text Case Is >= 10 / 24 MsgBox "Gebühren: " & Cells(Target.Row, 1).Text End Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick transponierte Werteübernahme und Druck

Problem:	Bei Doppelklick auf eine Zeile sollen die Daten der Zeile in Tabelle 2 transponiert übernommen und gedruckt werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean) Dim iCounter As Integer Cancel = True With Worksheets("Tabelle2") For iCounter = 1 To 5 .Cells(iCounter, 1) = Cells(Target.Row, iCounter) Next iCounter .PrintPreview End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick WAV-Datei abspielen

Problem: Bei Doppelklick auf einen Dateinamen in Spalte A soll die dort genannte WAV-Datei abgespielt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    If Target.Column <> 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Cancel = True
    If Dir(Target.Value) = "" Then
        Beep
        MsgBox "WAV-Datei wurde nicht gefunden!"
        Exit Sub
    End If
    Call sndPlaySound32(Target.Value, 1)
End Sub

StandardModule: basMain

Declare Function sndPlaySound32 Lib "winmm.dll" _
    Alias "sndPlaySoundA" (ByVal lpszSoundName _
    As String, ByVal uFlags As Long) As Long
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick gelinkte Arbeitsmappe öffnen

Problem: Bei Doppelklick auf Zelle E13 soll die dort gelinkte Arbeitsmappe geöffnet werden. Anpassung des Pfades/Dateinamens erforderlich.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim sTxt As String
    Cancel = True
    On Error GoTo ERRORHANDLER
    sTxt = ActiveCell.Formula
    sTxt = Left(sTxt, InStr(sTxt, "]") - 1)
    sTxt = WorksheetFunction.Substitute(sTxt, "'", "")
    sTxt = WorksheetFunction.Substitute(sTxt, "=", "")
    sTxt = WorksheetFunction.Substitute(sTxt, "[", "")
    If Dir(sTxt) <> "" Then
        Workbooks.Open sTxt
    Else
        MsgBox "Datei " & sTxt & " wurde nicht gefunden!"
    End If
ERRORHANDLER:
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen-Aktionen bei Doppelklick

Problem:	Bei Doppelklick auf eine Zelle soll der Wert aus Zeile 1 der doppelgeklickten Spalte mit dem Wert der Zeilennummer multipliziert und in einer MsgBox angezeigt werden. Gleichzeitig soll sich der Hintergrund der Zelle rot färben.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Public dblValue As Double Private Sub Worksheet_BeforeDoubleClick(_ ByVal Target As Range, Cancel As Boolean) Cancel = True If dblValue = 0 Then dblValue = Cells(1, Target.Column) * Target.Row MsgBox dblValue Target.Interior.ColorIndex = 3 Else Target.Value = dblValue Target.Interior.ColorIndex = 50 dblValue = 0 End If End Sub Private Sub Worksheet_SelectionChange(ByVal Target As Range) dblValue = 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Artikelnummer über InputBox suchen und eintragen

Problem: Bei Doppelklick in eine Zelle in den Spalten A:B soll sich eine InputBox zur Eingabe einer Artikelnummer öffnen. Diese Artikelnummer soll in Spalte A gesucht und die Zelle neben der Fundstelle soll ausgewählt werden

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_BeforeDoubleClick( _  
    ByVal Target As Range, Cancel As Boolean)  
    Dim var As Variant  
    Dim sTxt As String  
    If Target.Column > 2 Then Exit Sub  
    Cancel = True  
    sTxt = InputBox("Artikelnummer:", , "234")  
    If sTxt = "" Then Exit Sub  
    sTxt = "*" & sTxt & "*"  
    var = Application.Match(sTxt, Columns(1), 0)  
    If Not IsError(var) Then  
        Cells(var, 2).Select  
    End If  
    sTxt = InputBox("Eingabe:", , "Ein Text")  
    If sTxt = "" Then Exit Sub  
    Cells(var, 2) = sTxt  
    Call Worksheet_BeforeDoubleClick(ActiveCell, False)  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makroaufruf durch Doppelklick in Abhängigkeit von der Zelladresse

Problem: In Abhängigkeit einer doppelgeklickten Zelle sollen verschiedene Makros aufgerufen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    If Target.Row <> 7 Or Target.Column > 5 Then Exit Sub
    Cancel = True
    Select Case Target.Column
        Case 1: Call Makro1
        Case 2: Call Makro2
        Case 3: Call Makro3
        Case 4: Call Makro4
        Case 5: Call Makro5
    End Select
End Sub

StandardModule: basMain

Sub Makro1()
    MsgBox "Ich bin Makro1"
End Sub

Sub Makro2()
    MsgBox "Ich bin Makro2"
End Sub

Sub Makro3()
    MsgBox "Ich bin Makro3"
End Sub

Sub Makro4()
    MsgBox "Ich bin Makro4"
End Sub

Sub Makro5()
    MsgBox "Ich bin Makro5"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick aufrufende Zelle in UserForm nennen

Problem: Bei Doppelklick im Bereich A1:A10 soll eine UserForm aufgerufen und die aufrufende Zelle dort genannt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    If Intersect(Target, Range("A1:A10")) Is Nothing Then Exit Sub
    Cancel = True
    frmCaller.Show
End Sub

ClassModule: frmCaller

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    If OptionButton1.Value = True Then
        ActiveCell.Value = "Ein Wert"
    End If
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    frame1.Caption = "Eintrag in Zelle " & _
        ActiveCell.Address(False, False)
End Sub

StandardModule: basMain

Sub DialogAufruf()
    frmCaller.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellwert nach Doppelklick in Tabelle suchen

Problem:	Der Suchbegriff aus Zelle C5 soll bei Doppelklick auf diese Zelle in den Spalten D:F gesucht und die Fundstelle selektiert werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_BeforeDoubleClick(_ ByVal Target As Range, Cancel As Boolean) Dim rng As Range If Target.Address <> "\$B\$1" Then Exit Sub Cancel = True Set rng = Columns("D:F").Find(Range("B1").Value, _ lookat:=xlWhole, LookIn:=xlValues) If Not rng Is Nothing Then rng.Select Else Beep MsgBox "Suchbegriff wurde nicht gefunden!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick zum Monatsanfang springen

Problem:	Wie kann ich erreichen, dass bei Doppelklick auf einen Monatsnamen in Zeile 1 der Cursor zu dem jeweiligen Monat in der Tabelle springt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_BeforeDoubleClick(_ ByVal Target As Range, Cancel As Boolean) Cancel = True Application.Goto Range(ActiveCell.Value).Offset(0, -1), True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In geschützter Zelle in geschütztem Blatt Werterhöhung

Problem: Wie kann ich in geschützten Zellen eines geschützten Arbeitsblattes über Doppelklick Werterhöhungen vornehmen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    Cancel = True
    ActiveSheet.Unprotect
    Target.Value = Target.Value + 1
    ActiveSheet.Protect
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten der aktuellen Zeile in der Excel-Datenmaske anzeigen

Problem: Wie kann ich mir die Daten der aktuellen Zeile bei Doppelklick in der Excel-Datenmaske anzeigen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim rng As Range
    Cancel = True
    Set rng = ActiveCell.CurrentRegion.Rows(ActiveCell.Row)
    SendKeys "{down " & ActiveCell.Row - 2 & "}"
    ActiveSheet.ShowDataForm
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über Doppelklick aus Preisliste in Bestellformular einfügen

Problem: Durch Doppelklick auf eine Preislistenposition soll diese in ein Bestellformular übernommen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
StandardModule: basMain

Sub Loeschen()
    Worksheets("Bestellung").Range("A11:F" & Rows.Count).ClearContents
End Sub

Sub Kopieren()
    Dim objButton As Button
    ActiveSheet.Copy
    ActiveSheet.Unprotect
    ActiveSheet.Buttons.Delete
    Range("A9") = "Email: buero-pc-bedarf@t-online.de"
    ActiveSheet.Protect
End Sub

ClassModule: Tabelle2

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim iRowL As Integer, iRow As Integer, iCounter As Integer
    Cancel = True
    iRow = Target.Row
    If iRow < 8 Then GoTo ErrorHandler
    If IsEmpty(Cells(iRow, 1)) Then GoTo ErrorHandler
    With Worksheets("Bestellung")
        iRowL = .Cells(Rows.Count, 1).End(xlUp).Row + 1
        For iCounter = 1 To 4
            .Cells(iRowL, iCounter) = Cells(iRow, iCounter)
        Next iCounter
        .Cells(iRowL, 5).Value = 0
        .Cells(iRowL, 6).FormulaR1C1 = "=sum(RC[-2]*RC[-1])"
        .Range("B8") = Date
    End With
End

ErrorHandler:
    MsgBox "Sie müssen einen Artikel doppelklicken"
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblatt durch Zell-Doppelklick auswählen

Problem: Wie kann ich bei Doppelklick auf eine Zelle mit Tabellenblattnamen zum jeweiligen Arbeitsblatt springen

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle5

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    If Intersect(Target, Range("A1:A4")) Is Nothing Then Exit Sub
    Worksheets(Target.Value).Select
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Durch DoppelKlick Spalte in zweites Arbeitsblatt kopieren

Problem: Wie kann ich durch Doppelklicken auf eine Zelle die gesamte Spalte in ein zweites Arbeitsblatt kopieren?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim iCol As Integer
    Cancel = True
    With Worksheets("Aufträge")
        If IsEmpty(.Range("A1")) Then
            iCol = 1
        Else
            iCol = .Cells(1, Columns.Count) _
                .End(xlToLeft).Column + 1
        End If
        ActiveCell.EntireColumn.Copy .Cells(1, iCol)
    End With
    Application.CutCopyMode = False
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick Kontext-Menü mit Listeneinträgen anzeigen

Problem:	Wie kann ich mir bei Doppelklick auf eine Zelle ein Kontextmenü mit Einträgen aus einer Liste einer zweiten Tabelle anzeigen lassen? Das ausgewählte Element soll in die Zelle eingetragen und die Anzahl der in einer Zeile vorhandenen Elemente soll in der zweiten Tabelle gezählt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call DeleteCmdBar End Sub ClassModule: Tabelle1 Private Sub Worksheet_BeforeDoubleClick(_ ByVal Target As Excel.Range, Cancel As Boolean) Dim oBar As CommandBar Dim oBtn As CommandBarButton Dim wks As Worksheet Dim iCol As Integer Cancel = True Set wks = Worksheets("Daten") Call DeleteCmdBar Set oBar = Application.CommandBars.Add(_ Name:="StringInsert", _ Position:=msoBarPopup, _ MenuBar:=False, _ temporary:=True) iCol = 1 Do Until IsEmpty(wks.Cells(1, iCol)) Set oBtn = oBar.Controls.Add With oBtn .Caption = wks.Cells(1, iCol).Value .Style = msoButtonCaption .OnAction = "GetValue" End With iCol = iCol + 1 Loop CommandBars("StringInsert").ShowPopup End Sub StandardModule: basMain Sub GetValue() ActiveCell.Value = _ Application.CommandBars("StringInsert") _ .Controls(Application.Caller(1)).Caption End Sub Sub DeleteCmdBar() On Error Resume Next Application.CommandBars("StringInsert").Delete On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenweise von Blatt zu Blatt kopieren

Problem: Wie kann ich bei Doppelklick und/oder Schaltflächendruck die aktive Zeile in ein anderes Tabellenblatt kopieren?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
StandardModule: basMain

Sub CopyRow()
    Call CopyNow(ActiveCell.Row)
End Sub

Sub CopyNow(iRow As Integer)
    Dim wks As Worksheet
    Dim iRowL As Integer
    Application.ScreenUpdating = False
    Set wks = Worksheets("Tabelle2")
    iRowL = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1
    Rows(iRow).Copy wks.Rows(iRowL)
    Application.CutCopyMode = False
    wks.Columns.AutoFit
    Application.ScreenUpdating = True
End Sub

ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Excel.Range, Cancel As Boolean)
    Cancel = True
    Call CopyNow(Target.Row)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Editieren von Zellen durch DoppelKlick verhindern

Problem: Wie kann ich verhindern, daß der Zellinhalt nach Doppelklick editiert werden kann, ohne einen Blattschutz zu setzen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Excel.Range, _
    Cancel As Boolean)
    Cancel = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick Ordner im Explorer öffnen

Problem:	Wie kann ich bei Doppelklick auf eine Zelle den Explorer mit dem in der Zelle stehenden Verzeichnis öffnen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_BeforeDoubleClick(_ ByVal Target As Range, Cancel As Boolean) If Target.Address <> "\$A\$1" Then Exit Sub Cancel = True Shell "explorer " & Target.Value, vbMaximizedFocus End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick Textbox mit Text einfügen

Problem:	Wie kann ich bei einem Doppelklick auf eine Zelle in Abhängigkeit vom Zelleninhalt eine Textbox mit Text einfügen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_BeforeDoubleClick(_ ByVal Target As Range, Cancel As Boolean) Dim oTextBox As Object If Target.Column <> 1 Then Exit Sub If Target.Cells.Count > 1 Then Exit Sub Cancel = True With Target.Offset(0, 1) Set oTextBox = Me.TextBoxes.Add(_ .Left, .Top, .Width, .Height) End With oTextBox.Text = "Text zu " & Target.Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auf Doppelklick Zeilenumbruch und "X"

Problem:	Wie kann ich auf Doppelklick einen Zeilenumbruch so erzeugen, daß in der ersten Zeile ein "X" und in der zweiten Zeile der ursprüngliche Text steht?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean) If Target.Row <> 1 Or Target.Column > 31 Then Exit Sub Cancel = True If InStr(Target.Value, vbCrLf) Then Target.Value = WorksheetFunction.Substitute(Target.Value, vbCrLf & "X", "") Else Target.Value = Target.Value & vbCrLf & "X" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Teilergebnisse in Zahlenreihe eintragen und löschen

Problem:	Wie kann ich in einer Zahlenreihe auf Doppelklick Teilergebnisse eintragen bzw. löschen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean) If Target.Column <> 1 Then Exit Sub Cancel = True If InStr(Target.Formula, "SUBTOTAL") Then Rows(Target.Row & ":" & Target.Row + 1).Delete Else Rows(Target.Row & ":" & Target.Row + 1).Insert ActiveCell.Formula = "=SUBTOTAL(9,A1:A" & Target.Row - 3 & ")" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten von Tabelle1 auf Tabelle2 übertragen

Problem: Wie kann ich Daten einer Zeile aus Tabelle1 durch doppelklicken in die erste freie Zeile von Tabelle2 übertragen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim iRow As Integer
    If WorksheetFunction.CountA(Rows(Target.Row)) = 0 Then
        Exit Sub
    Else
        Cancel = True
    End If
    With Worksheets("Tabelle2")
        If IsEmpty(.Range("A1")) Then
            iRow = 1
        Else
            iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
        End If
        .Rows(iRow).Value = Target.EntireRow.Value
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile einfügen und Formeln und Werte übernehmen

Problem:	Wie kann ich in ein Tabellenblatt eine neue Zeile so einfügen, daß die Formeln und Werte der Zeile oberhalb übernommen werden?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_BeforeDoubleClick(_ ByVal Target As Excel.Range, _ Cancel As Boolean) Dim iRow As Integer If WorksheetFunction.CountA(Rows(Target.Row)) = 0 Then Exit Sub End If Cancel = True iRow = Target.Row Rows(iRow).Insert Rows(iRow - 1).Copy _ Destination:=Rows(iRow & ":" & iRow + 1) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzeige eines Datensatzes durch Klick mit der rechten Maustaste

Problem: Beim Klicken mit der rechten Maustaste in den Spalten A:E soll der Datensatz der activen Zeile in einer MsgBox angezeigt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeRightClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim intCounter As Integer
    Dim txt As String
    If Target.Column > 5 Then Exit Sub
    Cancel = True
    For intCounter = 1 To 5
        txt = txt & Cells(Target.Row, intCounter) & vbLf
    Next intCounter
    MsgBox txt
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur speichern, wenn alle Formularfelder ausgefüllt sind

Problem:	Die Arbeitsmappe darf sich nur speichern lassen, wenn alle Eingabefelder ausgefüllt sind.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean) If WorksheetFunction.CountA(_ Worksheets("Tabelle1").Range("D5,D7,D9,D11,D13")) < 5 Then Beep MsgBox "Bitte alle Eingabefelder ausfüllen!" Cancel = True End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Speichern Schaltfläche erstellen

Problem: Nach dem Speichern der Arbeitsmappe soll eine Schaltfläche erstellt werden, die nach deren Aufruf wieder gelöscht wird.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
StandardModule: basMain

Sub Message()
    MsgBox ThisWorkbook.FullName
    ActiveSheet.Buttons(2).Delete
End Sub

Private Sub CreateButton()
    Dim btn As Button
    Dim dWidth As Double, dHeight As Double
    dWidth = Range("C6").Width + Range("D6").Width
    dHeight = Range("C6").Height + Range("C7").Height
    Set btn = Tabelle1.Buttons.Add(Range("C6").Left, _
        Range("C6").Top, dWidth, dHeight)
    btn.OnAction = "Message"
    btn.Caption = "Klick mich"
    btn.Visible = True
End Sub

Sub Speichern()
    ThisWorkbook.Save
    Call CreateButton
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellhintergrund in Abhängigkeit von Zelleintrag festlegen.

Problem:	Bei Eintragungen in Zelle A1 bzw. A2 sollen die Folgezellen in Spalte A mit einem roten (r) oder gelben (g) Hintergrund belegt werden. Lösung A über das Change-Ereignis, B über die bedingte Formatierung.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Intersect(Target, Range("A1:A2")) Is Nothing Then Exit Sub If IsEmpty(Range("A1")) Or IsEmpty(Range("A2")) Then Exit Sub Columns(1).Interior.ColorIndex = xlNone Select Case Range("A2").Value Case "g" Range(Cells(3, 1), Cells(Range("A1").Value + 2, 1)) _ .Interior.ColorIndex = 6 Case "r" Range(Cells(3, 1), Cells(Range("A1").Value + 2, 1)) _ .Interior.ColorIndex = 3 End Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Filter vor Speichern zurücksetzen

Problem:	Beim Speichern der Arbeitsmappe sollen die gewählten Filterkriterien in allen Blättern zurückgesetzt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeSave(_ ByVal SaveAsUI As Boolean, _ Cancel As Boolean) Dim wks As Worksheet For Each wks In Worksheets If wks.FilterMode = True Then wks.ShowAllData End If Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattschutz beim Speichern setzen

Problem: Nach dem Speichern sollen Zellen mit Inhalt nicht mehr geändert werden können.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_BeforeSave( _  
    ByVal SaveAsUI As Boolean, Cancel As Boolean)  
    Dim rng As Range  
    ActiveSheet.Unprotect  
    For Each rng In ActiveSheet.UsedRange.Cells  
        If Not IsEmpty(rng) Then rng.Locked = True  
    Next rng  
    ActiveSheet.Protect  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Keine Wiederholung fett geschriebener Zahlen

Problem:	In fett geschriebene Zahlen dürfen sich in dem mit A1 zusammenhängenden Bereich nicht wiederholen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim rng As Range, oCell As Range If Target.Font.Bold = False Then Exit Sub If Target.Value = "" Then Exit Sub If IsNumeric(Target.Value) = False Then Exit Sub On Error GoTo ERRORHANDLER Application.EnableEvents = False With Range("A1").CurrentRegion For Each oCell In .Cells If oCell.Value = Target.Value And oCell.Font.Bold And oCell.Address <> Target.Address Then Beep MsgBox "Der eingegebene Wert existiert bereits!" Target.ClearContents Exit For End If Next oCell End With ERRORHANDLER: Application.EnableEvents = True End Sub Sub a() Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Webabfrage periodisch aktualisieren und Ergebnisse speichern

Problem:	Eine WebAbfrage wird stündlich aktualisiert. Die importierten Werte werden in eine Arbeitsmappe in das Standardverzeichnis gespeichert.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Calculate() Dim wkb As Workbook Application.ScreenUpdating = False ThisWorkbook.Worksheets(1).Copy Set wkb = ActiveWorkbook On Error GoTo ERRORHANDLER Application.EnableEvents = False With wkb With .Worksheets(1) .Columns("B:IV").Delete .Cells.Interior.ColorIndex = xlColorIndexNone .Range("A1").CurrentRegion.TextToColumns _ Destination:=Range("A1"), _ DataType:=xlDelimited, _ TextQualifier:=xlDoubleQuote, _ ConsecutiveDelimiter:=False, _ Tab:=False, _ Semicolon:=True, _ Comma:=False, _ Space:=False, Other:=False .Columns.AutoFit .Range("A65536").ClearContents End With .SaveAs Application.DefaultFilePath & "\TxtImport_" & Format(Now, "yymmdd_hhmmss") & ".xls", xlWorkbookNormal .Close savechanges:=False End With ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In Auswahlabhängigkeit unterschiedliche Werte anzeigen

Problem: In Abhängigkeit der Auswahl in Zelle B9 sollen unterschiedliche Grafiken angezeigt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Calculate()  
    If ActiveCell.Address <> "$B$8" Then Exit Sub  
    ActiveSheet.Shapes("Archiv").Visible = Range("B8").Value - 1  
    ActiveSheet.Shapes("Antworten").Visible = Not Range("B8").Value - 1  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen in Abhängigkeit der aktuellen Uhrzeit markieren

Problem: A1 soll von 22:01-06:00, B1 von 06:01 bis 14:00 und C1 von 14:01 bis 22:00 Uhr eine Kreuzmarkierung erhalten, Aktualisierung bei jeder Zelleingabe in Spalte A.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Calculate()
    Select Case Hour(Time)
        Case 22 To 24, 0 To 6
            With Range("A1")
                .Borders(xlDiagonalDown).Weight = xlThin
                .Borders(xlDiagonalUp).Weight = xlThin
            End With
            With Range("B1")
                .Borders(xlDiagonalDown).LineStyle = xlNone
                .Borders(xlDiagonalUp).LineStyle = xlNone
            End With
            With Range("C1")
                .Borders(xlDiagonalDown).LineStyle = xlNone
                .Borders(xlDiagonalUp).LineStyle = xlNone
            End With
        Case 6 To 14
            With Range("B1")
                .Borders(xlDiagonalDown).Weight = xlThin
                .Borders(xlDiagonalUp).Weight = xlThin
            End With
            With Range("A1")
                .Borders(xlDiagonalDown).LineStyle = xlNone
                .Borders(xlDiagonalUp).LineStyle = xlNone
            End With
            With Range("C1")
                .Borders(xlDiagonalDown).LineStyle = xlNone
                .Borders(xlDiagonalUp).LineStyle = xlNone
            End With
        Case 14 To 22
            With Range("C1")
                .Borders(xlDiagonalDown).Weight = xlThin
                .Borders(xlDiagonalUp).Weight = xlThin
            End With
            With Range("A1")
                .Borders(xlDiagonalDown).LineStyle = xlNone
                .Borders(xlDiagonalUp).LineStyle = xlNone
            End With
            With Range("B1")
                .Borders(xlDiagonalDown).LineStyle = xlNone
                .Borders(xlDiagonalUp).LineStyle = xlNone
            End With
    End Select
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopfzeile einer Autofilter-Tabelle gelb hinterlegen

Problem:	Beim Filtern einer Spalte soll die Kopfzeile gelb hinterlegt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Calculate() Dim flt As Filter Dim iCol As Integer For Each flt In ActiveSheet.AutoFilter.Filters iCol = iCol + 1 If flt.On Then Cells(1, iCol).Interior.ColorIndex = 6 Else Cells(1, iCol).Interior.ColorIndex = _ xlColorIndexNone End If Next flt End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makroaufruf bei Betätigung des Autofilters

Problem:	Bei Betätigung des Autofilters soll ein Makro aufgerufen werden.
Lösung:	Auslösung in Verbindung der Formel in E1 und der Calculate-Ereignis-Prozedur im Klassenmodul der Tabelle.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Calculate() MsgBox "Hallo!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Formelkalkulation Zellwert erhöhen

Problem:	Nach Kalkulation der Formel in Zelle B1 soll der Wert aus Zelle B1 um das fünffache erhöht in Zelle C1 eingetragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Calculate() Range("C1").Value = Range("B1").Value * 5 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Meldung nach Einfügen oder Löschen von Zeilen

Problem:	Nach dem Einfügen oder Löschen von Zeilen oberhalb der jetzigen Zeile 21 soll eine entsprechende Meldung angezeigt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() gintRow = Range("Last").Row End Sub ClassModule: Tabelle1 Public gintRow As Integer Private Sub Worksheet_Calculate() If Range("Last").Row < gintRow Then MsgBox "Es wurde eine Zeile gelöscht!" ElseIf Range("Last").Row > gintRow Then MsgBox "Es wurde eine Zeile eingefügt!" End If gintRow = Range("Last").Row End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Erreichen eines vorgegebenen Wertes Datum eintragen

Problem:	Wenn sich der Wert der Formel in Spalte A auf > 10 erhöht, soll in B1 das aktuelle Datum mit Zeit eingetragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Calculate() Dim iRow As Integer, iRowL As Integer iRowL = Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL If Cells(iRow, 1) > 10 Then Cells(iRow, 2).Value = Now End If Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In Abhängigkeit von einem Formelwert Makro starten

Problem: Wenn ein durch eine Formel ermittelter Zellwert einen Grenzwert übersteigt, soll ein Makro gestartet werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Calculate()
    If Range("C1").Value > 100 Then
        MsgBox "Wert in Zelle C1 hat 100 überstiegen!"
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-Aufruf, wenn Formel-Wert Grenze überschreitet

Problem:	Wie kann ich immer dann, wenn ein durch eine Formel ermittelter Wert eine Grenze übersteigt, eine UserForm anzeigen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: frmMeldung Private Sub CommandButton1_Click() Unload Me End Sub ClassModule: Tabelle1 Private Sub Worksheet_Calculate() If Range("C2").Value > 100 Then frmMeldung.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Signal, wenn sich Formelwert über eine Grenze bewegt

Problem:	Wie kann ich ein Signal ertönen lassen, wenn sich der Wert einer Formel auf über 0 erhöht?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Calculate() If Range("C1").Value > 0 Then Beep End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Wertüberschreitung Makro aufrufen

Problem: Wie kann ich bei der Überschreitung eines bestimmten Wertes in einer eine Formel enthaltenden Zelle ein Makro aufrufen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Calculate()
    If Range("A1").Value > 4 Then
        MsgBox "Der Wert von 4 wurde überschritten!"
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Meldung bei Werteänderung

Problem: Wie kann ich immer dann, wenn sich der Wert in Zelle C1 ändert, eine Meldung anzeigen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Calculate()
    If gdValue <> Range("C1").Value Then
        MsgBox "Der Wert von C1 hat sich von " & _
            & gdValue & " auf " & Range("C1").Value & _
            " geändert."
        gdValue = Range("C1").Value
    End If
End Sub

ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    gdValue = _
        Worksheets("Tabelle1").Range("C1").Value
End Sub

StandardModule: basMain

Public gdValue As Double
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrundfarbe bei Eingabezeile ändern

Problem:	Bei Eingaben im Bereich B:2:D10 soll der Hintergrund der Eingabezeile und der Summenzeile in der gleichen Zeile auf rot und die Schrift auf fett gesetzt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Intersect(Target, Range("B2:D10")) Is Nothing Then Exit Sub 'Wenn die bestehende Markierung nicht aufgehoben werden soll, 'die beiden folgenden Zeilen auskommentieren Range("B2:E10").Interior.ColorIndex = xlColorIndex.xlColorIndexNone Range("B2:E10").Font.Bold = False Target.Interior.ColorIndex = 3 Target.Font.Bold = True Cells(Target.Row, 5).Interior.ColorIndex = 3 Cells(Target.Row, 5).Font.Bold = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Korrespondierende Werte aus Tabelle und Datum eintragen

Problem:	Bei Einträgen in Spalte B gem. Gültigkeitsliste werden in den Spalten C:E die korrespondierenden Werte aus Tabelle Artikel und in Spalte A das aktuelle Datum mit Zeit eingetragen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 2 Then Exit Sub On Error GoTo ERRORHANDLER Cells(Target.Row, 1).Value = Now With Worksheets("Artikel") Cells(Target.Row, 3).Value = WorksheetFunction.VLookup(Target.Value, .Columns("A:D"), 2, 0) Cells(Target.Row, 4).Value = WorksheetFunction.VLookup(Target.Value, .Columns("A:D"), 3, 0) Cells(Target.Row, 5).Value = WorksheetFunction.VLookup(Target.Value, .Columns("A:D"), 4, 0) End With ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anpassung des Lagerbestands bei Ein- und Ausgang

Problem:	Bei Eingaben in die Spalten C:D soll sich der Lagerbestand in Spalte E anpassen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 3 And Target.Column <> 4 Then Exit Sub If IsEmpty(Target) Then Exit Sub If Target.Column = 3 Then Cells(Target.Row, 5).Value = Cells(Target.Row, 5).Value + Target.Value Else Cells(Target.Row, 5).Value = Cells(Target.Row, 5).Value - Target.Value End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Festlegung der Hintergrundfarbe nach Werteingabe

Problem: Bei Eingabe eines Buchstabens in Spalte A soll sich die Hintergrundfarbe gemäß der Tabelle Colors ändern.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim vRow As Variant
    If Target.Column <> 1 Then Exit Sub
    If Target.Cells.Count > 1 Then Exit Sub
    If IsEmpty(Target) Then
        Target.Interior.ColorIndex = xlColorIndexNone
    Else
        With Worksheets("Colors")
            vRow = Application.Match(UCase(Target.Value), .Columns(1), 0)
            If Not IsError(vRow) Then
                Target.Interior.ColorIndex = .Cells(vRow, 1).Interior.ColorIndex
            End If
        End With
    End If
End Sub

StandardModule: Modul1

Sub FarbenAnlegen()
    Dim iRow As Integer
    With Worksheets("Colors")
        For iRow = 1 To 26
            With Cells(iRow, 1)
                .Value = Chr(iRow + 64)
                .Interior.ColorIndex = iRow + 1
            End With
        Next iRow
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makroverzweigung in Abhängigkeit einer Werte-Auswahl

Problem: Bei Auswahl eines Wertes in Zelle C6 soll zu verschiedenen Makros verzweigt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Address <> "$C$6" Then Exit Sub
    Run Target.Value
End Sub

StandardModule: Modul1

Sub A()
    MsgBox "Ich bin Makro A"
End Sub

Sub B()
    MsgBox "Ich bin Makro B"
End Sub

Sub C()
    MsgBox "Ich bin Makro C"
End Sub

Sub D()
    MsgBox "Ich bin Makro D"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fehlermeldung, wenn nicht der erwartete Wert eingetragen wird

Problem:	Die Formel in Zelle F2 soll eine Fehlermeldung anzeigen, wenn im Bereich A1:C10 ein anderer Wert als der in Zelle F1 genannte eingetragen wird.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Umwandlung von Komma in Punkt

Problem:	Bei Eingaben mit Kommata in Spalte A wird das Komma automatisch in einen Punkt umgewandelt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 1 Then Exit Sub If IsEmpty(Target) Then Exit Sub On Error GoTo ERRORHANDLER Application.EnableEvents = False If InStr(Target.Value, ",") Then Target.NumberFormat = "@" Target.Value = Application.Substitute(Target.Text, ",", ".") End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eingaben prüfen, ob der Wert schon in anderer Tabelle

Problem:	Bei Eingaben in den Tabellen 1 bis 5 wird überprüft, ob der eingegebene Wert bereits an gleicher Stelle in einem anderen Tabellenblatt enthalten ist. Wenn ja, wird die Eingabe nicht akzeptiert.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range) Dim iWks As Integer Dim bln As Boolean If Left(Sh.Name, 7) <> "Tabelle" Then Exit Sub If IsEmpty(Target) Then Exit Sub For iWks = 1 To 5 If Worksheets(iWks).Name <> Target.Parent.Name Then If Worksheets(iWks).Range(Target.Address).Value = Target.Value Then bln = True Exit For End If End If Next iWks If bln Then Beep MsgBox "Sorry, dieser Wert ist schon in Tabelle" & iWks & " enthalten!" Application.EnableEvents = False Target.ClearContents Application.EnableEvents = True End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eingabe in Spalte B nächsthöhere Nummer in Spalte A

Problem:	Bei Eingaben in Spalte B soll in Spalte A automatisch die nächsthöhere Nummer von Spalte A vergeben werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 2 Then Exit Sub On Error Resume Next Application.EnableEvents = False If IsEmpty(Target) Then Target.Offset(0, -1).ClearContents Else Target.Offset(0, -1).Value = WorksheetFunction.Max(Columns(1)) + 1 Application.EnableEvents = True End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppeleingaben verhindern

Problem:	In Spalte A werden Doppeleingaben über das Change-Ereignis verhindert, in Spalte B durch das Excel-Feature Menü DATEN / GÜLTIGKEIT
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim rng As Range If Target.Column <> 1 Then Exit Sub If IsEmpty(Target) Then Exit Sub If Target.Cells.Count > 1 Then Exit Sub Application.EnableEvents = False On Error GoTo ERRORHANDLER If Application.CountIf(Range("A:A"), Target.Value) > 1 Then Target.ClearContents End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Formate und Zellinhalte ausblenden

Problem: Im Bereich B2:B6 sollen Formate und Zellinhalte nicht sichtbar sein. Dies gilt auch beim Überschreiben der Zellen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    If Intersect(Target, Range("B2:D2")) Is Nothing Then Exit Sub
    On Error GoTo ERRORHANDLER
    Application.ScreenUpdating = True
    Application.EnableEvents = False
    With Range("B2:D2")
        .ClearFormats
        .NumberFormat = ";;;"
    End With
ERRORHANDLER:
    Application.ScreenUpdating = False
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrundfarbe gem. Werte/Farbindex festlegen

Problem: Bei Zahleneingaben in Spalte A sollen die Hintergrundfarbenfestlegung gem. Tabelle Colors erfolgen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim vRow As Variant
    If Target.Column <> 1 Then Exit Sub
    If Target.Cells.Count > 1 Then
        If WorksheetFunction.CountA(Target) = 0 Then
            Target.Interior.ColorIndex = xlColorIndexNone
            Exit Sub
        Else
            Exit Sub
        End If
    End If
    With Worksheets("Colors")
        If IsEmpty(Target) Then
            Target.Interior.ColorIndex = xlColorIndexNone
        Else
            vRow = Application.Match(Target.Value, .Columns(1), 0)
            If Not IsError(vRow) Then
                Target.Interior.ColorIndex = .Cells(vRow, 1).Interior.ColorIndex
            Else
                Target.Interior.ColorIndex = xlColorIndexNone
            End If
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wenn in Zeile 1 ein F und in Zeile 2 ein Ur, dann F in Klammern

Problem:	Werden in Zeile 1 ein F und in Zeile 2 ein Ur eingesetzt, soll das F in Klammern gesetzt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Row > 2 Then Exit Sub Application.EnableEvents = False On Error GoTo ERRORHANDLER If Cells(1, Target.Column) = "F" And Cells(2, Target.Column) = "Ur" Then Cells(1, Target.Column) = "(F)" End If If Cells(1, Target.Column) = "(F)" And Cells(2, Target.Column) <> "Ur" Then Cells(1, Target.Column) = "F" End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wenn Wert vorhanden, Wert in Klammern setzen

Problem: Wird in Zeile 1 ein Wert aus Tabelle Data, Spalte A und in Zeile 2 ein Wert aus Spalte Data, Spalte B eingesetzt, soll der Wert in Zeile 1 in Klammern gesetzt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim wks As Worksheet
    Dim iCol As Integer
    Dim sText As String
    If Target.Row > 2 Then Exit Sub
    Set wks = Worksheets("Data")
    iCol = Target.Column
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    If Not IsError(Application.Match(Cells(1, iCol).Value, wks.Columns(1), 0)) And _
        Not IsError(Application.Match(Cells(2, iCol).Value, wks.Columns(2), 0)) Then
        Cells(1, iCol) = "(" & Cells(1, iCol).Value & ")"
    End If
    sText = Cells(1, iCol)
    sText = Replace(sText, "(", "")
    sText = Replace(sText, ")", "")
    If Not IsError(Application.Match(sText, wks.Columns(1), 0)) And _
        IsError(Application.Match(Cells(2, iCol).Value, wks.Columns(2), 0)) Then
        Cells(1, iCol) = sText
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Farbänderung bei Änderung des Wahrheitswertes

Problem: Bei Änderung des Wahrheitswertes in Zelle B1 soll sich die Hintergrundfarbe dieser Zelle ändern. Die Bedingte Formatierung soll nicht eingesetzt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    With Range("B1")
        If .Value = True Then
            .Interior.ColorIndex = 34
        Else
            .Interior.ColorIndex = 36
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabekontrolle bei Datumswerten

Problem: Bei Eingaben im Bereich A2:B3 sollen Nicht-Datumswerte abgelehnt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle2

```
Private Sub Worksheet_Change(ByVal Target As Range)
    If Intersect(Target, Range("A2:B3")) Is Nothing Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Target.Cells.Count > 1 Then Exit Sub
    Application.EnableEvents = False
    If Not IsDate(Target.Value) Then
        MsgBox "Bitte ein gültiges Datum eingeben!"
        Target.ClearContents
    End If
    On Error GoTo ERRORHANDLER
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Produktaktivierung Abmessungen eintragen

Problem:	Bei Eingabe eines "x" in Spalte B sollen die Abmessungen des jeweiligen Produkts aus dem Blatt Daten ausgelesen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abruf des Korrespondenzwertes per VBA

Problem:	Bei Eingaben eines Landes in Spalte A soll in Spalte B der Korrespondenzwert aus der Data-Tabelle angezeigt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 1 Then Exit Sub If Target.Cells.Count > 1 Then Exit Sub If IsEmpty(Target) Then Target.Offset(0, 1).ClearContents End If Application.EnableEvents = False On Error GoTo ERRORHANDLER Target.Offset(0, 1).Value = WorksheetFunction.VLookup(_ Target.Value, Worksheets("Data").Columns("A:B"), 2, 0) ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Shapes in Abhängigkeit von Zellwerten färben

Problem: Die Shapes in Zeile 2 sollen sich bei Werten in Zeile 1 < 0 rot, bei > 0 grün färben.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim shp As Shape
    If Intersect(Target, Range("A1:F1")) Is Nothing Then Exit Sub
    For Each shp In Shapes
        If shp.TopLeftCell.Address = Target.Offset(1, 0).Address Then
            Select Case Target.Value
                Case Is < 0
                    shp.Fill.ForeColor.SchemeColor = 10
                Case Is > 0
                    shp.Fill.ForeColor.SchemeColor = 17
                Case Else
                    shp.Fill.ForeColor.SchemeColor = 0
            End Select
        End If
    Next shp
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabe eines X nur einmal zulassen und Wert aufrufen

Problem: In den Zellen B22, I22 und Q22 sollen nur einmal die Eingabe eines "X" zugelassen werden. In Zelle V22 soll in Abhängigkeit der X-Eintragung der Wert 2, 4 oder 6 erscheinen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim rng As Range, rngAct As Range
    Dim iCounter As Integer
    Set rng = Range("B22,I22,Q22")
    If Intersect(Target, rng) Is Nothing Then Exit Sub
    For Each rngAct In rng.Cells
        If rngAct.Value = "X" Then
            iCounter = iCounter + 1
            If iCounter > 1 Then
                On Error GoTo ERRORHANDLER
                Application.EnableEvents = False
                rngAct.ClearContents
            End If
        End If
    Next rngAct
ERRORHANDLER:
    Application.EnableEvents = True
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)

End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verknüpfungen bei x-Eingabe erstellen bzw. löschen

Problem:	Wenn in Zelle A1 ein "x" eingegeben wird, sollen in Zeile 1 von Tabelle2 Verknüpfungen Zur Zeile 1 in Tabelle1 erstellt werden. Bei Eingabe eines anderen Wertes sollen diese wieder gelöscht werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) If Target.Address <> "\$A\$1" Then Exit Sub If UCase(Target.Value) = "X" Then Worksheets("Tabelle2").Range("A1:IV1").Formula = "=Tabelle1!A1" Else Worksheets("Tabelle2").Rows(1).ClearContents End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Cursor nach Zelleingaben nach Vorgabe positionieren

Problem: Nach Eingaben in Zelle A1 soll der Cursor auf Zelle C1, nach Eingabe in C1 auf Zelle E1 springen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Address <> "$A$1" And Target.Address <> "$C$1" Then Exit Sub
    Target.Offset(0, 2).Select
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Begrenzung der Textlänge über Ereignis-Programmierung

Problem:	In Spalte A soll die Textlänge nach Eingaben auf 35 Zeichen reduziert werden. Lösung über Ereignisprogrammierung.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 1 Then Exit Sub On Error GoTo ERRORHANDLER If Len(Target.Value) > 35 Then Application.EnableEvents = False Target.Value = Left(Target.Value, 35) End If ERRORHANDLER: Application.EnableEvents = True End Sub Private Sub Worksheet_SelectionChange(ByVal Target As Range) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beschränkung der Eingabemöglichkeit auf 32 Zellen

Problem: Im Bereich A1:D88 dürfen nur 32 Zellen mit Werten belegt werden. Wird versucht, einen 33. Wert einzugeben, darf dieser nicht übernommen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Intersect(Target, Range("A1:D88")) Is Nothing Then Exit Sub
    On Error GoTo ERRORHANDLER
    Application.EnableEvents = False
    If WorksheetFunction.CountA(Range("A1:D88")) > 32 Then
        MsgBox "Nur 32 Eingaben erlaubt!"
        Target.ClearContents
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblattübergreifendes Verhindern von Doppeleintragungen

Problem: Es soll verhindert werden, dass je Zeile Doppeleintragungen über alle Tabellen im Bereich A1:B20 erfolgen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
    Dim wks As Worksheet
    If Intersect(Target, Range("A1:B20")) Is Nothing Then Exit Sub
    For Each wks In Worksheets
        If wks.Name <> Sh.Name Then
            If WorksheetFunction.CountIf( _
                wks.Range(wks.Cells(Target.Row, 1), _
                wks.Cells(Target.Row, 100)), Target.Value) > 0 Then
                Beep
                MsgBox "Eingabe schon vorhanden in Blatt " & wks.Name & "!"
                Application.EnableEvents = False
                Target.Value = ""
                Application.EnableEvents = True
            End If
        End If
    Next wks
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen bei Eingaben leeren

Problem: Bei Eingaben sollen die übrigen Werte der jeweiligen Zeile gelöscht werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim iCol As Integer, iRowT As Integer, iColT As Integer
    If Intersect(Target, Range("A1:D5")) Is Nothing Then Exit Sub
    iRowT = Target.Row
    iColT = Target.Column
    On Error GoTo ERRORHANDLER
    Application.EnableEvents = False
    For iCol = 1 To iColT - 1
        Cells(iRowT, iCol).ClearContents
    Next iCol
    For iCol = iColT + 1 To 4
        Cells(iRowT, iCol).ClearContents
    Next iCol
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eingabe Grafiken in alle Tabellen und Makro zuweisen

Problem: Bei Eingaben in Zelle B1 soll die dort genannte Grafikdatei in alle Tabellen unter Hinzufügen einer Makrozuweisung eingefügt werden. Bestehende Grafiken sind zu löschen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim oPic As Picture
    Dim iWks As Integer
    Dim sFile As String
    If Target.Address <> "$B$1" Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    sFile = Range("B1").Value
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Grafikdatei wurde nicht gefunden!"
        Exit Sub
    End If
    Call DeleteShapes
    For iWks = Me.Index To Worksheets.Count
        Set oPic = Worksheets(iWks).Pictures.Insert(sFile)
        With oPic
            .Left = 100
            .Top = 120
            .Width = 80
            .Height = 120
            .OnAction = "GoToWks"
        End With
    Next iWks
End Sub

StandardModule: Modul1

Sub DeleteShapes()
    Dim wks As Worksheet
    For Each wks In Worksheets
        wks.Pictures.Delete
    Next wks
End Sub

Sub GoToWks()
    Worksheets(Range("B2").Value).Select
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Vergleich Kommentar einfügen

Problem: Wie kann ich bei Eingaben in Spalte A diese mit Spalte C vergleichen und bei gleichen Werten den Wert von Spalte B als Kommentar einfügen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim cmt As Comment
    If Target.Column <> 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Target.Cells.Count > 1 Then Exit Sub
    If Not Target.Comment Is Nothing Then
        Target.Comment.Delete
    End If
    If Target.Value = Cells(Target.Row, 3).Value Then
        Set cmt = Target.AddComment(Cells(Target.Row, 2).Value)
        cmt.Shape.TextFrame.AutoSize = True
    End If
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)

End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werterhöhung einer Zelle bei Eingabe in einer anderen

Problem: Der Wert in Zelle A20 soll sich bei Eingaben in Zelle A1 um den dort eingegebenen Wert verringern.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Address <> "$A$1" Then Exit Sub
    Application.EnableEvents = False
    Range("A20").Value = Range("A20").Value - Range("A1").Value
    On Error GoTo ERRORHANDLER
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Gültigkeitsfeldauswahl Zeile einfügen

Problem:	Bei der Auswahl eines Gültigkeitsfeldes in Spalte G soll an der Position der aktiven Zelle eine neue Zeile eingefügt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 7 Then Exit Sub If Target.Cells.Count > 1 Then Exit Sub If Target.Value = "Neue Zeile" Then Rows(ActiveCell.Row).Insert Target.ClearContents End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eine WENN-Formel nur bei der ersten Werteänderung berechnen

Problem: Die WENN-Formel in B1 soll nur bei der ersten Werteänderung in A1 berechnet werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Calculate()
    With Range("B1")
        If .HasFormula Then
            .Value = .Value
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Eingabe eines S keine weiteren Eingaben möglich

Problem: Bei Eingabe eines S in Zelle D4 sollen im Bereich D4:Z4 keine weiteren Eingaben möglich sein.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Address <> "$D$4" Then Exit Sub
    ActiveSheet.Unprotect
    If Target.Value = "S" Then
        Range("D4:Z4").Locked = True
    End If
    ActiveSheet.Protect
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eingabe Formel in der Nebenspalte nach unten kopieren

Problem: Bei Eingaben in Spalte A sollen die Formeln in Spalte B aus der Zeile über der Eingabezeile in die Eingabezeile kopiert werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Column <> 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    On Error GoTo ERRORHANDLER
    Application.EnableEvents = False
    Range(Cells(Target.Row - 1, 2), Cells(Target.Row - 1, 256)).Copy _
        Range(Cells(Target.Row, 2), Cells(Target.Row, 256))
    Application.CutCopyMode = False
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Umwandlung einer Zeichenfolge in Spalte A in Großbuchstaben

Problem:	Alle Eingaben in Spalte A sollen in Großbuchstaben umgewandelt werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If IsEmpty(Target) Then Exit Sub If Target.Cells.Count > 1 Then Exit Sub If Target.Column > 1 Then Exit Sub Application.EnableEvents = False Target.Value = UCase(Target.Value) Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrwertsteuer und Bruttobetrag nach Nettowerteingabe

Problem:	Bei Eingaben im Bereich E6:E50 soll in Spalte F die 16%ige Mehrwertsteuer und in Spalte G der Bruttobetrag ausgeworfen werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim rngInput As Range Set rngInput = Range("E6:E50") On Error GoTo ERRORHANDLER If Target.Cells.Count > 1 Then Exit Sub If Intersect(rngInput, ActiveCell) Is Nothing Then Exit Sub Application.EnableEvents = False If IsEmpty(Target) Then Range(Target, Target.Offset(0, 2)).ClearContents Else Target.Offset(0, 1) = Target.Value * 0.16 Target.Offset(0, 2) = Target.Value + Target.Offset(0, 1).Value End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Fußballergebnisse eingeben und Tabelle sortieren

Problem:	Im Bereich B2:E5 werden Spielergebnisse eingegeben. Punkte und Tore werden automatisch errechnet und es wird nach Tabellenplatz sortiert.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Intersect(Target, Range("B2:E5")) Is Nothing Then Exit Sub On Error GoTo 0 If Cells(Target.Row, 1).Value = Cells(1, Target.Column).Value Then Application.EnableEvents = False Beep MsgBox "Hier sind keine Eingaben erlaubt!" Target.ClearContents Application.EnableEvents = True Exit Sub End If If InStr(Target.Value, ":") = False Then Application.EnableEvents = False Beep MsgBox "Ungültige Eingabe!" Target.ClearContents Application.EnableEvents = True Exit Sub End If Call Sortieren ERRORHANDLER: Application.EnableEvents = True End Sub StandardModule: Modul1 Function Points(rng As Range) As Integer Dim oCell As Range Dim iPoints As Integer, iAct As Integer Dim sTxt As String For Each oCell In rng.Cells If Not IsEmpty(oCell) Then sTxt = oCell.Value iAct = Left(sTxt, InStr(sTxt, ":") - 1) - _ Right(sTxt, Len(sTxt) - InStr(sTxt, ":")) Select Case iAct Case Is > 0 iPoints = iPoints + 3 Case Is = 0 iPoints = iPoints + 1 End Select End If Next oCell Points = iPoints End Function Function Goals(rng As Range) As Integer Dim oCell As Range Dim iGoals As Integer, iAct As Integer Dim sTxt As String For Each oCell In rng.Cells If Not IsEmpty(oCell) Then sTxt = oCell.Value iAct = Left(sTxt, InStr(sTxt, ":") - 1) - _ Right(sTxt, Len(sTxt) - InStr(sTxt, ":")) iGoals = iGoals + iAct End If Next oCell Goals = iGoals End Function Function GoalsCount(rng As Range) As Integer Dim oCell As Range Dim iGoals As Integer, iAct As Integer Dim sTxt As String For Each oCell In rng.Cells If Not IsEmpty(oCell) Then sTxt = oCell.Value iAct = Left(sTxt, InStr(sTxt, ":") - 1) iGoals = iGoals + iAct End If Next oCell GoalsCount = iGoals End Function Sub Sortieren() Range("B1").CurrentRegion.Sort _ key1:=Range("F2"), order1:=xlDescending, _ key2:=Range("G2"), order2:=xlDescending, _ key3:=Range("H2"), order3:=xlDescending End Sub Sub Schuetzen() ActiveSheet.Protect userinterfaceonly:=True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makroverzweigung im Change-Ereignis

Problem: Eine Change-Ereignis soll in Abhängigkeit der aufrufenden Zelle auf verschiedene Makros verzweigen. Erster Bereich ist A1:B10, zweiter C1:D10.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Not Intersect(Target, Range("A1:B10")) Is Nothing Then
        Call ErsteRoutine
    ElseIf Not Intersect(Target, Range("C1:D10")) Is Nothing Then
        Call ZweiteRoutine
    End If
End Sub

Sub ErsteRoutine()
    MsgBox "Aufruf aus dem Bereich A1:B10!"
End Sub

Sub ZweiteRoutine()
    MsgBox "Aufruf aus dem Bereich C1:D10!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Im Change-Ereignis Eingabe in oder Leeren von Zellen erkennen

Problem: Im Change-Ereignis soll erkannt werden, ob ein Eintrag erfolgte oder ob Werte in einer einzelnen Zelle oder in einem Bereich gelöscht wurden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim rng As Range
    Dim bln As Boolean
    For Each rng In Target.Cells
        If rng.Value <> "" Then bln = True
    Next rng
    If bln = False Then
        Beep
        MsgBox "Es wurde gelöscht!"
    Else
        MsgBox "Es erfolgte ein Eintrag!"
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Laufende Eintragungen mit revolvierender Numerierung

Problem:	Numerierungen in Spalte A sollen bei Eingaben in Spalte B automatisch eingetragen werden. Ist Zeile 18 erreicht, wird die Eintragung in Zeile 1 mit der nächstfolgenden Nummer fortgesetzt.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 2 Then Exit Sub If IsEmpty(Target) Then Exit Sub Application.EnableEvents = False If Target.Row = 1 Then If Not IsEmpty(Cells(19, 1)) Then Target.Offset(0, -1) = Cells(19, 1) + 1 Else Target.Offset(0, -1) = 1 End If Else Target.Offset(0, -1) = Target.Offset(-1, -1) + 1 End If Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Durch VBA erfolgte Eintragung nach 30 Sekunden löschen.

Problem: Bei einem Eintrag in Zelle B1 soll - wenn Zelle A1 nicht leer ist - der Bereich A1:B1 nach 30 Sekunden gelöscht werden.

Lösung: Mit Alt+F11 in die Entwicklungsumgebung wechseln und in die genannten Module den jeweiligen Code eingeben.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Address <> "$B$1" Then Exit Sub
    If IsEmpty(Target) Or IsEmpty(Target.Offset(0, -1)) Then Exit Sub
    Application.OnTime Now + TimeSerial(0, 0, 30), "DeleteContents"
End Sub

StandardModule: basMain

Sub DeleteContents()
    Worksheets("Tabelle1").Range("A1:B1").ClearContents
End Sub

Sub Eintragen()
    Range("B1").Value = "Hallo"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabewerte in Spalte einsortieren

Problem:	Der in C1 einzugebende Wert soll in eine Zehnergruppen-Zahlenreihe mit Einzellenlücken in Spalte A zwischen die Zehnerwerte eingeordnet werden, wobei bereits vorhandene Werte zu überschreiben sind.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim var As Variant If Target.Address <> "\$C\$1" Then Exit Sub If IsEmpty(Target) Then Exit Sub var = Application.Match(_ CInt(Left(CStr(Target.Value), Len(Target.Value) - 1)) _ * 10, Columns(1), 0) If Not IsError(var) Then Application.EnableEvents = False On Error GoTo ERRORHANDLER Cells(var + 1, 1).Value = Target.Value End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelleinträge den Werten in einer zweiten Zelle hinzuaddieren

Problem: In B1 eingegebene Zahlen sollen dem Wert in Zelle A1 hinzuaddiert werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Address <> "$B$1" Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    Range("A1").Value = Range("A1").Value + Target.Value
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werteprüfung und -meldung bei Zelleingaben

Problem:	Bei Eingaben in der mit dem Bereichsnamen "Test" belegten Zelle A1 soll überprüft werden, ob der Wert zwischen 1 und 3 liegt. Wenn ja, soll eine Meldung mit dem Wert erscheinen, wenn nein soll er gelöscht werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Intersect(Target, Range("Test")) Is Nothing Then Exit Sub If IsEmpty(Target) Then Exit Sub If Target.Value < 1 Or Target.Value > 3 Then Beep MsgBox "Nur Eingaben zwischen 1 und 3 erlaubt!" Application.EnableEvents = False On Error GoTo ERRORHANDLER Target.ClearContents Else MsgBox "Eingabe in Zelle ""Test"": " & Range("Test").Value End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eintrag Übernahme der Formel aus Nebenzelle

Problem: Bei einer Eingabe in Spalte E soll die in der Parallelzelle in Spalte D hinterlegte Formel eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim strFormula As String
    If Target.Column <> 5 Or IsEmpty(Target) Then Exit Sub
    strFormula = Target.Offset(0, -1).Formula
    strFormula = WorksheetFunction.Substitute(strFormula, ".", ",")
    Application.EnableEvents = False
    Target.Value = "'" & strFormula
    Application.EnableEvents = True
End Sub

ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim sFormula As String
    If Target.Column <> 5 Or IsEmpty(Target) Then Exit Sub
    sFormula = Target.Offset(0, -1).Value
    sFormula = Application.Substitute(sFormula, ",", ".")
    Application.EnableEvents = False
    If Left(sFormula, 1) = "+" Then
        Target.Value = "=" & sFormula
    Else
        Target.Value = sFormula
    End If
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eintrag Zeile in 2. Blatt kopieren und Tabellen sortieren

Problem:	Bei Einträgen in Spalte A von Tabelle1 soll der Eintrag automatisch in Tabelle2 übernommen werden. Beide Tabellen sind zu sortieren.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 1 Then Exit Sub Worksheets("Tabelle2").Range(Target.Address).Value = Target.Value Range("A1").Sort key1:=Range("A2"), order1:=xlAscending, header:=xlYes With Worksheets("Tabelle2") .Range("A1").Sort key1:=.Range("A2"), order1:=xlAscending, header:=xlYes End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abfrage der Auswahl aus einem Gültigkeits-Listefeld

Problem:	Die Auswahl aus einem Gültigkeits-Listefeld soll über eine Select-Case-Anweisung abgefragt werden. Nur ab XL9 lauffähig.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Address <> "\$C\$1" Then Exit Sub Select Case Target.Value Case "Januar", "Februar", "März", "April", "Mai", "Juni" MsgBox "1. Jahreshälfte" Case Else MsgBox "2. Jahreshälfte" End Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingaben auf Werte zwischen 100 und 200 beschränken

Problem:	Eingaben in Spalte A sollen auf Werte zwischen 100 und 200 beschränkt werden. Die Routine soll auch in XL5 / 7 lauffähig sein.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 1 Then Exit Sub If IsEmpty(Target) Then Exit Sub If Not IsNumeric(ActiveCell.Value) Then Application.EnableEvents = False On Error GoTo ERRORHANDLER Target.ClearContents ElseIf Target.Value < 100 Or Target.Value > 200 Then Beep MsgBox "Der Eingabewert liegt außerhalb des " & vbLf & _ "empfohlenen Bereiches!", vbExclamation End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Raumdoppelbelegungen verhindern

Problem: Nach Eingabe der Raumnummer soll geprüft werden, ob der Raum zum vorgesehenen Zeitraum frei ist.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim iRow As Integer, iRowL As Integer
    If Target.Column <> 3 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row
    For iRow = 2 To iRowL
        If iRow <> Target.Row Then
            If Cells(iRow, 2) = Cells(Target.Row, 2) And _
                Cells(iRow, 3) = Cells(Target.Row, 3) Then
                MsgBox "Sorry, schon belegt!"
                Target.ClearContents
            End If
        End If
    Next iRow
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrundfarbe nach Farbindex-Liste festlegen

Problem:	Bei Eingaben in Spalte A soll - wenn der Eingabewert in Spalte B von Tabelle FarbIndex gefunden wird - der Zellohintergrund mit dem dort festgelegten Farbindex hinterlegt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim var As Variant If Target.Column <> 1 Then Exit Sub With Worksheets("FarbIndex") var = Application.Match(Target.Value, .Columns(1), 0) If Not IsError(var) Then Target.Interior.ColorIndex = .Cells(var, 2) End If End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Blattnamenseintrag Zeile in jeweiliges Blatt verschieben

Problem: Bei Eintrag eines Arbeitsblattnamens in Spalte D soll die Zeile des Eintrags in das jeweilige Tabellenblatt verschoben werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim wks As Worksheet
    Dim iRow As Integer
    If Target.Column <> 4 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    On Error Resume Next
    Set wks = Worksheets(Target.Value)
    If Err > 0 Or wks Is Nothing Then
        MsgBox "Tabelle wurde nicht gefunden!"
        Exit Sub
    End If
    On Error GoTo 0
    With wks
        iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
        .Range(.Cells(iRow, 1), .Cells(iRow, 3)).Value = _
            Range(Cells(Target.Row, 1), Cells(Target.Row, 3)).Value
    End With
    Rows(Target.Row).Delete
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in externer Mappe suchen und mit Format übernehmen.

Problem: Bei einer Eingabe eines Wertes in Spalte B soll dieser Wert in einer anderen Arbeitsmappe in Zeile1 und der Wert aus Spalte A dort in Spalte A gesucht werden. Die Formatierung aus der Fundzelle soll übernommen werden.

Lösung: Den nachfolgenden Code in die genannten Elemente einfügen.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Column = 2 Then Call ReadFormatting(Target.Offset(0, -1))
End Sub

StandardModule: Modul1

Sub ReadFormatting(rng As Range)
    Dim vRow As Variant, vCol As Variant
    Dim sFile As String
    Application.ScreenUpdating = False
    sFile = ThisWorkbook.Path & "\test1.xls"
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Testarbeitsmappe ist nicht vorhanden!"
        Exit Sub
    End If
    Workbooks.Open sFile, False
    Worksheets("160901").Select
    vRow = Application.Match(rng.Value, Columns(1), 0)
    vCol = Application.Match(rng.Offset(0, 1).Value, Rows(1), 0)
    If Not IsError(vRow) And Not IsError(vCol) Then
        Cells(vRow, vCol).Copy
        rng.Offset(0, 2).PasteSpecial xlPasteFormats
        Application.CutCopyMode = False
    End If
    ActiveWorkbook.Close savechanges:=False
    Application.ScreenUpdating = True
End Sub

Sub SetFormula()
    Dim sFile As String, sFormula As String
    sFile = "" & ThisWorkbook.Path & "\[test1.xls]160901'!"
    sFormula = "=INDEX(" & sFile & "$A$1:$F$6,MATCH(A1," & _
        sFile & "A:A,0),MATCH(B1," & sFile & "1:1,0))"
    Range("C1").Formula = sFormula
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Setzen der Hintergrundfarbe nach Wertevergleich

Problem: Bei Eingaben oder Änderungen von Formelwerten soll eine Überprüfung mit den Vorgängerwerten stattfinden. Ist der neue Zellwert höher, soll die Hintergrundfarbe auf rot, ist er niedriger, soll sie auf gelb gesetzt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim vNew As Variant, vOld As Variant
    If Intersect(Target, Range("B3:C3")) Is Nothing Then Exit Sub
    vNew = Target.Value
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    Application.Undo
    vOld = Target.Value
    Target.Value = vNew
    If vNew > vOld Then
        Target.Interior.ColorIndex = 3
    ElseIf vNew < vOld Then
        Target.Interior.ColorIndex = 6
    Else
        Target.Interior.ColorIndex = xlColorIndexNone
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in zweiter Tabelle suchen und als Kommentar ausgeben

Problem: Bei Eingaben in Spalte A soll der Eingabetext in Spalte A von Tabelle2 gesucht und die Werte der Nachbarzelle sollen in der Eingabezelle als Kommentar angezeigt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim wks As Worksheet
    Dim var As Variant
    Dim sTxt As String
    If Target.Column <> 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Set wks = Worksheets("Tabelle2")
    var = Application.Match(Target, wks.Columns(1), 0)
    On Error Resume Next
    Target.Comment.Delete
    On Error GoTo 0
    If IsError(var) Then Exit Sub
    sTxt = "Spalte B: " & wks.Cells(var, 2) & vbLf
    sTxt = sTxt & "Spalte C: " & wks.Cells(var, 3) & vbLf
    sTxt = sTxt & "Spalte D: " & wks.Cells(var, 4)
    If sTxt <> "" Then
        ActiveCell.AddComment sTxt
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Listen von Eingabedaten in einer anderen Tabelle

Problem:	Eingaben in Spalte B des Daten-Blattes sollen in einer zweiten Tabelle bis gesamt 10 Stück gelistet werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) Dim var As Variant Dim iCol As Integer If IsEmpty(Target) Then Exit Sub If Target.Column <> 2 Or Target.Row > 20 Then Exit Sub With Worksheets("Tabelle1") var = Application.Match(_ Target.Offset(0, -1).Value, .Columns(1), 0) If Not IsError(var) Then iCol = WorksheetFunction.CountA(.Rows(var)) + 1 .Cells(var, iCol).Value = Target.Value End If End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Meldung bei Überschreitung einer Spaltensumme

Problem:	Wenn die Summe der Werte im Eingabe-Spaltenbereich von Zelle 9 bis Zelle 22 den Wert von 7,5 überschreitet, soll eine Meldung ausgegeben werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column < 4 Or Target.Column > 34 Then Exit Sub If Target.Row < 9 Or Target.Row > 22 Then Exit Sub If WorksheetFunction.Sum(Range(Cells(9, Target.Column), _ Cells(22, Target.Column))) > 7.5 Then Beep MsgBox "Vorsicht, der Wert von 7,5 wurde überschritten!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus Gültigkeitsliste durch eine Zahl ersetzen

Problem: Die Auswahl aus einer Gültigkeitsliste in Spalte D soll durch eine Zahl ersetzt werden. Nur lauffähig ab XL10.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Column <> 4 Then Exit Sub
    Application.EnableEvents = False
    Target.Value = WorksheetFunction.VLookup( _
        Target.Value, Range("A1:B12"), 2, 0)
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabe von Fehlerwerten verhindern

Problem:	Im Bereich A1:B6 soll die Eingabe von Fehlerwerten verhindert und im Fehlerfall eine entsprechende MsgBox angezeigt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim var As Variant If IsEmpty(Target) Then Exit Sub If Intersect(Target, Range("Berechnungen")) _ Is Nothing Then Exit Sub var = Application.Sum(Range("Berechnungen")) If IsError(var) Then Beep MsgBox "Bitte korrekte Werte eingeben!" Target.Value = "" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Zelleingabe andere Zellen füllen

Problem:	Bei Eingabe von x oder y im Bereich K21:K51 sollen in den jeweiligen Zeilen in den Spalten C und D Bindestriche eingetragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Intersect(Target, Range("A1:A10")) Is Nothing Then Exit Sub If Target.Value <> "x" And Target.Value <> "y" Then Target.Offset(0, 1).ClearContents Target.Offset(0, 2).ClearContents Else Target.Offset(0, 1) = "-" Target.Offset(0, 2) = "-" End If End Sub</pre>



Eintragung von Reklamationen nach Reklamationscode

Problem:	Nach Eingabe der TourNr, LfdNr und Reklamationscode sollen die Daten aus den Spalten C, D und F der Tourenliste zusammen mit der Reklamationsart in das Blatt Reklamationen eingefügt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) Dim vTour As Variant, vRek As Variant Dim iRow As Integer, iRowT As Integer If IsEmpty(Target) Then Exit Sub Select Case Target.Row Case 12 vTour = Application.Match(Target, _ Worksheets("Touren").Columns(1), 0) If IsError(vTour) Then Beep MsgBox "Die Tour wurde nicht gefunden!" Range("E12").Select End End If Case 13 vTour = Application.Match(Range("E12"), _ Worksheets("Touren").Columns(1), 0) With Worksheets("Touren") Do If .Cells(vTour, 2) = Target Then iRow = vTour Exit Do End If vTour = vTour + 1 Loop Until .Cells(vTour, 1) <> _ .Cells(vTour - 1, 1) End With If iRow = 0 Then Beep MsgBox "Die LfdNr wurde nicht gefunden!" Range("E13").Select End End If Case 14 vTour = Application.Match(Range("E12"), _ Worksheets("Touren").Columns(1), 0) With Worksheets("Touren") Do If .Cells(vTour, 2) = Range("E13") Then iRow = vTour Exit Do End If vTour = vTour + 1 Loop Until .Cells(vTour, 1) <> .Cells(vTour - 1, 1) End With vRek = Application.Match(Target, _ Worksheets("Code").Columns(1), 0) If IsError(vRek) Then Beep MsgBox "Der Reklamationscode wurde nicht gefunden!" Range("E14").Select End End If With Worksheets("Reklamationen") iRowT = .Cells(Rows.Count, 1).End(xlUp).Row + 1 .Cells(iRowT, 1) = Worksheets("Touren").Cells(iRow, 3) .Cells(iRowT, 2) = Worksheets("Touren").Cells(iRow, 4) .Cells(iRowT, 3) = Worksheets("Touren").Cells(iRow, 6) .Cells(iRowT, 4) = Worksheets("Code").Cells(vRek, 2) End With End Select End Sub Private Sub Msg() Beep MsgBox "Eingabe nicht korrekt!" End End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Änderungen in Protokollblatt festhalten

Problem:	Änderungen in diesem Arbeitsblatt sollen unter Angabe der Zelleadresse und des vorherigen Wertes im Blatt "Protokoll" protokolliert werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim vNew As Variant, vOld As Variant Dim iRow As Integer If Intersect(Target, Range("A1:E12")) Is Nothing Then Exit Sub vNew = Target.Value Application.EnableEvents = False On Error GoTo ERRORHANDLER Application.Undo vOld = Target.Value Target.Value = vNew With Worksheets("Protokoll") iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1 .Cells(iRow, 1).Value = Target.Address(False, False) .Cells(iRow, 2).Value = vOld End With ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Memory in Zellkommentar

Problem: Bei Zellveränderungen mit Wert soll der Zellhintergrund gelb, ohne Wert grün belegt werden. Die jeweils vorhergehenden Werte sind in einem Zell-Kommentar zu speichern.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Application.DisplayCommentIndicator = xlCommentIndicatorOnly
    Range("A2").Select
    Range("A1").Select
End Sub

ClassModule: Tabelle1

Public var As Variant

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Value <> var Then
        If Not ActiveCell.Comment Is Nothing Then
            ActiveCell.Comment.Text ActiveCell.Comment.Text & vbCrLf & var
        Else
            ActiveCell.AddComment CStr(var)
        End If
        If Not IsEmpty(Target) Then
            Target.Interior.ColorIndex = 6
        Else
            Target.Interior.ColorIndex = 4
        End If
    End If
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Selection.Cells.Count > 1 Then Exit Sub
    var = Target.Value
End Sub

Sub ReSetComments()
    Dim cmt As Comment
    For Each cmt In ActiveSheet.Comments
        cmt.Delete
    Next cmt
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Automatische Sortierung bei Zelleintrag

Problem:	Wenn in den Spalten A:D eine Eintragung erfolgt und alle vier Spalten in der relevanten Zeile ausgefüllt sind, soll eine Sortierung erfolgen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column > 4 Then Exit Sub If WorksheetFunction.CountA(_ Range(Cells(Target.Row, 1), _ Cells(Target.Row, 4))) = 4 Then Columns("A:D").Sort _ key1:=Range("A2"), _ order1:=xlAscending, header:=True End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nummerngruppe prüfen und melden

Problem:	Bei Eingaben in Spalte A soll abgeprüft werden, ob die Nummerngruppe der ersten 3 Zeichen in Spalte A im Blatt "Daten" existiert. Ergebnisausgabe als Meldung.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim rng As Range Dim sNo As String If Target.Column <> 1 Then Exit Sub sNo = Left(Target.Value, 3) Set rng = Worksheets("Daten").Columns(1).Find(sNo, _ lookat:=xlPart, LookIn:=xlValues) If Not rng Is Nothing Then MsgBox "Nummerngruppe " & sNo & " existiert" Else MsgBox "Nummerngruppe " & sNo & " existiert nicht" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktion nach Auswahl in Gültigkeitsliste

Problem: In Abhängigkeit der Auswahl im Gültigkeitsfeld B1 soll die Anzahl der Tage im Gültigkeitsfeld D1 festgelegt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der Tabelle ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim intCounter As Integer
    Dim strDays As String
    If Target.Address <> "$B$1" Then Exit Sub
    Select Case Range("B1").Value
        Case "Januar", "März", "Mai", "Juli", "August", "Oktober", "Dezember"
            For intCounter = 1 To 31
                strDays = strDays & intCounter & ","
            Next intCounter
        Case "April", "Juni", "September", "November"
            For intCounter = 1 To 30
                strDays = strDays & intCounter & ","
            Next intCounter
        Case "Februar"
            If Year(Date) Mod 4 = 0 Then
                For intCounter = 1 To 29
                    strDays = strDays & intCounter & ","
                Next intCounter
            Else
                For intCounter = 1 To 28
                    strDays = strDays & intCounter & ","
                Next intCounter
            End If
        End Select
    strDays = Left(strDays, Len(strDays) - 1)
    With Range("D1").Validation
        .Delete
        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
            xlBetween, Formula1:=strDays
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob Daten schon vorhanden und Meldung

Problem: Bei Eingaben in Spalte C soll überprüft werden, ob in einer anderen Zeile der gleiche Name mit dem gleichen Datum und der gleichen Uhrzeit vorkommt. Wenn ja, soll eine Meldung ausgegeben werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Column <> 3 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Range("IV1").FormulaArray = "=sum((C1:C5000="" & _
        Target.Value & """)*(A1:A5000="" & _
        Target.Offset(0, -2).Address(False, False) & ")*(B1:B5000="" & _
        Target.Offset(0, -1).Address(False, False) & ")*1)"
    If Range("IV1") > 1 Then
        MsgBox "Sorry, Wert schon vorhanden!"
        Application.EnableEvents = False
        On Error GoTo ERRORHANDLER
        Target.ClearContents
    End If
    Range("IV1").ClearContents
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eintrag von X Zeile in andere Tabelle kopieren

Problem: Bei Eintragung eines X in Spalte R soll die entsprechende Zeile kopiert und am Ende der Tabelle in Tabelle2 eingetragen werden. Im Quellblatt ist die Zeile zu löschen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim iRow As Integer
    If Target.Column <> 5 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If UCase(Target.Value) = "X" Then
        With Worksheets("Tabelle2")
            iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
            Rows(Target.Row).Copy .Rows(iRow)
            Rows(Target.Row).Delete
        End With
    End If
    Application.CutCopyMode = False
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Erfüllen einer Bedingungen Farbe für 1 Minute

Problem:	Wenn nach einer Eingabe in Zelle A1 der Wert von A1 <> dem Wert von B1, soll sich der Hintergrund von Zelle B1 für 1 Minute auf gelb stellen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Address <> "\$A\$1" Then Exit Sub If Range("A1").Value <> Range("B1").Value Then Range("B1").Interior.ColorIndex = 6 Application.Wait Now + TimeSerial(0, 1, 0) Range("B1").Interior.ColorIndex = xlColorIndexNone End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

SVERWEIS als Ereignisprozedur

Problem: Bei Eintrag einer Nummer in Spalte A sollen - falls vorhanden - die entsprechenden Werte aus tab1 in die Spalten B:C eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle2

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim var As Variant
    If Target.Column <> 1 Then Exit Sub
    With Application
        var = .VLookup(Target.Value, _
            Worksheets("tab1").Columns("A:C"), 2, 0)
        If Not IsError(var) Then
            Application.EnableEvents = False
            On Error GoTo ERRORHANDLER
            Target.Offset(0, 1) = .VLookup(Target.Value, _
                Worksheets("tab1").Columns("A:C"), 2, 0)
            Target.Offset(0, 2) = .VLookup(Target.Value, _
                Worksheets("tab1").Columns("A:C"), 3, 0)
        End If
    End With
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile in Abhängigkeit zur Eingabe in Blätter kopieren

Problem:	Bei Eingabe eines "K" in Zelle A1 soll die Zeile in Arbeitsblatt "ab", bei einem "W" in Blatt "xy" kopiert werden. Die Hintergrundfarbe soll in beiden Fällen auf rot gesetzt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle3 Private Sub Worksheet_Change(ByVal Target As Range) Dim wks As Worksheet Dim lRow As Long If Target.Address <> "\$A\$1" Then Exit Sub Select Case Target.Value Case "K" Set wks = Worksheets("xy") Case "W" Set wks = Worksheets("ab") Case Else Range("A1").Interior.ColorIndex = xlColorIndexNone Exit Sub End Select With wks lRow = .Cells(.Rows.Count, 1).End(xlUp).Row + 1 .Rows(lRow).Value = Rows(1).Value End With Range("A1").Interior.ColorIndex = 3 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrundfarbe in Abhängigkeit vom Wochentag

Problem:	Bei Datumseingaben in Spalte A soll die Hintergrundfarbe in Abhängigkeit vom Wochentag festgelegt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If IsEmpty(Target) Then Exit Sub If IsDate(Target.Value) = False Then Exit Sub Select Case Weekday(Target) Case 1: Target.Interior.ColorIndex = 3 Case 2: Target.Interior.ColorIndex = 4 Case 3: Target.Interior.ColorIndex = 5 Case 4: Target.Interior.ColorIndex = 6 Case 5: Target.Interior.ColorIndex = 7 Case 6: Target.Interior.ColorIndex = 8 Case 7: Target.Interior.ColorIndex = 9 End Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eintrag Datum aus Zellwerten zusammenstellen

Problem: Bei Eintragung einer Zahl in Zelle E10 soll automatisch das Datum unter Zugrundelegung des Wertes aus L1 als Jahr und L2 als Monat eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If IsEmpty(Target) Then Exit Sub
    If Target.Address = "$E$1" Then
        Application.EnableEvents = False
        On Error GoTo ERRORHANDLER
        Target.Value = DateSerial(Range("B1").Value, _
            Range("B2").Value, Target.Value)
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus Eingabeblatt in jeweiliges Monatsblatt eintragen

Problem: Eintragungen in den Zellen C8:E8 sollen in dem in Zelle B8 genannten Arbeitsblatt an die nächste freie Position eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim wks As Worksheet
    Dim iRow As Integer
    If Intersect(Target, Range("C8:E8")) Is Nothing Then Exit Sub
    On Error Resume Next
    Set wks = Worksheets(Cells(Target.Row, 2).Value)
    If Err > 0 Or wks Is Nothing Then
        Err.Clear
        Beep
        MsgBox "Das Arbeitsblatt " & Cells(Target.Row, 2).Value & _
            " existiert nicht!"
        Exit Sub
    End If
    On Error GoTo 0
    iRow = wks.Cells(Rows.Count, Target.Column - 2) _
        .End(xlUp).Row + 1
    wks.Cells(iRow, Target.Column - 2).Value = Target.Value
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabemöglichkeit auf 255 Zeichen je Zelle begrenzen

Problem: Wie kann ich verhindern, dass mehr als 255 Zeichen in eine Zelle eingegeben werden?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim sTxt As String
    sTxt = Target.Value
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    If Len(sTxt) > 255 Then
        Target.Value = Left(sTxt, 255)
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Zelleingabe automatische Sortierung

Problem:	Wie kann ich erreichen, dass nach Zelleingaben eine Tabelle automatisch neu sortiert wird?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 1 Then Exit Sub Range("A1").Sort _ Key1:=Range("A2"), _ Order1:=xlAscending, _ Header:=xlGuess, _ OrderCustom:=1, _ MatchCase:=False, _ Orientation:=xlTopToBottom End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelleingaben in Gruppen von 5 Zeichen aufteilen

Problem: Wie kann ich Eingaben in Zelle A1 in Gruppen von jeweils fünf Zeichen aufteilen und diese in Zellen eintragen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim iCounter As Integer, iMax As Integer, iRow As Integer
    Dim sTxt As String
    If Target.Column <> 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    iMax = 5
    sTxt = Target.Text
    For iCounter = 1 To Len(sTxt)
        If Len(sTxt) > 5 Then
            Target.Offset(iRow, 0) = Left(sTxt, 5)
            iRow = iRow + 1
            sTxt = Right(sTxt, Len(sTxt) - 5)
        End If
    Next iCounter
    Target.Offset(iRow, 0) = sTxt
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingaben in Abhängigkeit von dritten Zellen zulassen / verhindern

Problem:	Wie kann ich Zelleingaben in einem vorgegebenen Bereich davon abhängig machen, ob sich in einer bestimmten Zelle ein Eintrag befindet?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Intersect(Target, Range("A15:T60")) Is Nothing Then Exit Sub If Target.Address = "\$C\$15" Or Target.Address = _ "\$T\$15" Then Exit Sub If IsEmpty(Target) Then Exit Sub If Not IsEmpty(Range("C15")) And IsEmpty(Range("T15")) Then MsgBox "Zuerst Zelle T15 ausfüllen!" Application.EnableEvents = False On Error GoTo ERRORHANDLER Target.ClearContents End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafik nach Eingabe des Dateinamens einfügen

Problem: Wie kann ich nach Eingabe eines Grafikdateinamens in Spalte A diese Grafik im Tabellenblatt einfügen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle2

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim pct As Picture
    If Target.Column <> 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    On Error GoTo ErrorHandler
    Set pct = ActiveSheet.Pictures.Insert _
        (Range("C1").Value & "\" & _
        Target.Value & ".gif")
    With pct
        .Left = Range("B4").Left
        .Top = Range("B4").Top
    End With
    Exit Sub
ErrorHandler:
    MsgBox "Grafik ist nicht vorhanden!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus Matrix in zweitem Blatt auflisten

Problem:	Wie kann ich bei Eingabe die Werte einer Matrix im ersten Arbeitsblatt im zweiten Arbeitsblatt als Liste aufführen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim rng As Range Dim iRow As Integer If Intersect(Target, Range("B2:J21")) Is Nothing Then Exit Sub With Worksheets("Ziel") Set rng = .Columns("A").Find(_ Cells(Target.Row, 1).Value & "_" & Cells(1, Target.Column).Value, _ lookat:=xlWhole, LookIn:=xlValues) If rng Is Nothing Then iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1 .Cells(iRow, 1) = Cells(Target.Row, 1).Value & "_" & _ Cells(1, Target.Column).Value .Cells(iRow, 2) = Target.Value Else rng.Offset = Target.Value End If End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datum über Zehnertastatur im Modus 1-1 als TT.MM.JJJJ ausgeben

Problem:	Ich gebe in einem Tabellenblatt das Datum über die Zehnertastatur im Modus 1-1 ein. Als Ergebnis wird "01. Jan" angezeigt. Wie kann ich erreichen, dass die Ausgabe als TT.MM.JJJJ erfolgt?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.NumberFormat = "d-mmm" Then Target.NumberFormat = "dd.mm.yyyy" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Übernahme von Daten nach Eingabe eines Monatsnamens

Problem: Wenn ich in eine bestimmte Zelle einen Monatsnamen eingebe, möchte ich, dass aus einer zweiten Tabelle die zugehörigen Daten in die aktuelle Tabelle eingelesen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle3

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim var As Variant
    Dim iRowL As Integer, iRow As Integer, iRowT As Integer
    If Target.Address <> "$B$1" Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Range("A2:B65536").ClearContents
    var = Application.Match(Target.Value, Worksheets("Daten").Rows(1), 0)
    If IsError(var) Then
        MsgBox "Monat wurde nicht gefunden!"
        Exit Sub
    End If
    With Worksheets("Daten")
        iRowL = .Cells(Rows.Count, var).End(xlUp).Row
        iRowT = 1
        For iRow = 2 To iRowL
            If Not IsEmpty(.Cells(iRow, var)) Then
                iRowT = iRowT + 1
                Cells(iRowT, 1) = .Cells(iRow, 1)
                Cells(iRowT, 2) = .Cells(iRow, var)
            End If
        Next iRow
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Stundeneintrag Meldung bei Grenzüberschreitung

Problem: Wie kann ich mir bei der Eintragung von Stunden bei Erreichen eines Grenzwertes eine Meldung anzeigen und den Eintrag auf 0 zurücksetzen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Column > 5 Or Target.Row = 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    If WorksheetFunction.Sum(Range(Cells(Target.Row, 1), _
        Cells(Target.Row, 5))) > 8 Then
        MsgBox "Wer mehr als 8 Stunden arbeitet," & _
            "den soll der Teufel holen!"
        Target.Value = 0
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingaben in B:D automatisch mit 6 multiplizieren

Problem: Wie kann ich erreichen, dass Eingaben in den Spalten B:D automatisch mit 6 multipliziert werden?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Column > 2 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Not IsNumeric(Target.Value) Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    Target.Value = Target.Value * 6
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einen von einer Schlüsselziffer abhängigen Bereich kopieren

Problem:	Wie kann ich erreichen, daß bei Eingabe einer 1 ein Bereich aus Tabelle "Maske1", bei Eingabe einer 2 ein Bereich aus Tabelle "Maske2" in das aktuelle Blatt kopiert wird?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Address <> "\$B\$1" Then Exit Sub If IsEmpty(Target) Then Exit Sub Application.EnableEvents = False On Error GoTo ERRORHANDLER If Target.Value = 1 Then Worksheets("Maske1").Range("C5:G18").Copy Range("C5:G18") ElseIf Target.Value = 2 Then Worksheets("Maske2").Range("C5:G18").Copy Range("C5:G18") End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eingabe von "x" Zelle unterhalb in fett

Problem:	Wie kann ich veranlassen, dass bei der Eingabe eines x die Zelle unterhalb als fett formatiert wird und diese Formatierung bei einer anderen Angabe wieder aufgehoben wird?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column > 2 Then Exit Sub If Target.Value = "x" Then Target.Offset(2, 0).Font.Bold = True Else Target.Offset(2, 0).Font.Bold = False End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wechselseitige Buchung in zwei Konten

Problem: Wie kann ich auf zwei Konten - bspw. Kasse und Bank - wechselseitig buchen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

ClassModule: Tabelle3

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim intRow As Integer
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    If Target.Column = 2 Then
        With Worksheets(Target.Offset(0, -1).Value)
            intRow = .Cells(Rows.Count, 3).End(xlUp).Row + 1
            .Cells(intRow, 4) = Target
            .Cells(intRow, 3) = Target.Parent.Name
        End With
    ElseIf Target.Column = 4 Then
        With Worksheets(Target.Offset(0, -1))
            intRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
            .Cells(intRow, 2) = Target
            .Cells(intRow, 1) = Target.Parent.Name
        End With
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```

ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim iRow As Integer
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    If Target.Column = 2 Then
        With Worksheets(Target.Offset(0, -1).Value)
            iRow = .Cells(Rows.Count, 3).End(xlUp).Row + 1
            .Cells(iRow, 4) = Target
            .Cells(iRow, 3) = Target.Parent.Name
        End With
    ElseIf Target.Column = 4 Then
        With Worksheets(Target.Offset(0, -1).Value)
            iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
            .Cells(iRow, 2) = Target
            .Cells(iRow, 1) = Target.Parent.Name
        End With
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Pfeilfarbe- und Bewegung abhängig von Zellwerten

Problem:	Wie kann ich einen Pfeil mit folgenden Bedingungen manipulieren: - Wenn A1<B1 dann Pfeil rot nach unten - Wenn A1=B1 dann Pfeil gelb nach links - Wenn A1>B1 dann Pfeil grün nach oben
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Intersect(Target, Range("A1:B1")) Is Nothing Then Exit Sub With Shapes("Autoform 1") If Range("A1").Value > Range("B1") Then .Rotation = 90# .Fill.ForeColor.SchemeColor = 50 ElseIf Range("A1").Value < Range("B1") Then .Rotation = 270# .Fill.ForeColor.SchemeColor = 53 Else .Rotation = 0# .Fill.ForeColor.SchemeColor = 13 End If End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamische Postleitzahldatenbank

Problem: Wie kann ich mir eine dynamische Postleitzahldatenbank anlegen? Bei Eingabe einer bereits bekannten PLZ soll der zugehörige Ort eingetragen werden, ist die PLZ nicht bekannt, soll ein Dialog zur Ortseingabe aufgerufen und PLZ und Ort sollen in die PLZDatenbank eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim var As Variant
    Dim iRow As Integer
    Dim sOrt As String
    If Target.Column <> 1 Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    If IsEmpty(Target) Then
        Target.Offset(0, 1).ClearContents
        Exit Sub
    End If
    With Worksheets("PLZ")
        var = Application.Match(Target.Value, .Columns(1), 0)
        If Not IsError(var) Then
            Target.Offset(0, 1).Value = .Cells(var, 2).Value
        Else
            sOrt = InputBox("Bitte Ort eingeben:")
            If sOrt = "" Then Exit Sub
            iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
            .Cells(iRow, 1).Value = Target.Value
            .Cells(iRow, 2).Value = sOrt
            Target.Offset(0, 1).Value = sOrt
        End If
    End With
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fahrzeugbelegungsplan nach Datumseintragung aktualisieren

Problem:	Wie kann ich in einem Fahrzeugbelegungsplan nach Eintragung eines Enddatums im relevanten Zeitraum den jeweiligen Fahrer eintragen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) Dim varStart As Variant, varEnd As Variant Dim iRow As Integer, iCol As Integer If Target.Column <> 3 Then Exit Sub If IsEmpty(Target) Then Exit Sub If Not IsDate(Target) Then Exit Sub With Worksheets("Tabelle2") varStart = Application.Match(_ CDb1(Target.Offset(0, -1).Value), .Columns(1), 0) varEnd = Application.Match(_ CDb1(Target.Value), .Columns(1), 0) If IsError(varStart) Or IsError(varEnd) Then _ GoTo ERRORHANDLER iCol = WorksheetFunction.Match(_ Cells(Target.Row, 1).Value, .Rows(1), 0) For iRow = varStart To varEnd .Cells(iRow, iCol).Value = Target.Offset(0, 1).Value Next iRow End With Exit Sub ERRORHANDLER: MsgBox "Der Datumsbereich wurde nicht gefunden!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eintragung UserForm mit aktualisierten Werten anzeigen

Problem: Wie kann ich bei Einträgen in Spalte A eine UserForm aufrufen, in der in einer ListBox die aktualisierten Werte angezeigt werden?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    If Target.Column <> 1 Then Exit Sub
    With frmAutomatic
        .lstAutomatic.List = Range("A1").CurrentRegion.Value
        .Show
    End With
End Sub

ClassModule: frmAutomatic

Private Sub cmdWeiter_Click()
    Unload Me
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle nach Datumseintrag automatisch sortieren

Problem: Wie kann ich eine Tabelle nach einem Datumseintrag in Spalte A automatisch sortieren lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Column <> 1 Then Exit Sub
    Range("A1").CurrentRegion.Sort _
        Key1:=Range("A2"), _
        Order1:=xlAscending, _
        Header:=xlYes, _
        OrderCustom:=1, _
        MatchCase:=False, _
        Orientation:=xlTopToBottom
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Midi-Datei beim Öffnen einer Arbeitsmappe abspielen

Problem: Wie kann ich beim Öffnen einer Arbeitsmappe eine Midi-Datei, deren Pfad und Name in Zelle A1 stehen, in Excel abspielen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    With Tabelle1
        .MediaPlayer1.AutoStart = True
        .MediaPlayer1.FileName = .Range("A1").Value
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eintragungen im 1. Blatt in andere Blätter übernehmen

Problem:	Eintragungen in Spalte B des zweiten Tabellenblattes sollen in allen Blättern in Spalte E übernommen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle9 Private Sub Worksheet_Change(ByVal Target As Excel.Range) Dim iWks As Integer If Target.Column <> 2 Then Exit Sub For iWks = Me.Index + 1 To Worksheets.Count Worksheets(iWks).Cells(Target.Row, 5) = Target.Value Next iWks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eingabe andere Werte löschen

Problem: Wie kann ich bei Eingabe eines Wertes in einem definierten Bereich die übrigen in diesem Bereich bereits vorhandenen Werte löschen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim rng As Range
    If Intersect(Target, Range("A1:E1")) Is Nothing Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    For Each rng In Range("A1:E1").Cells
        If Intersect(rng, Target) Is Nothing Then
            rng.ClearContents
        End If
    Next rng
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelleingaben in nicht sichtbarem Blatt protokollieren

Problem: Wie kann ich jede Zelleingabe mit Angabe des Anwendernamens, des Datums und der Zeit in einem nicht sichtbare Tabellenblatt protokollieren?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
StandardModule: basMain

Sub HistorieEinAus()
    With Worksheets("History")
        If .Visible = True Then
            .Visible = xlVeryHidden
        Else
            .Visible = True
        End If
    End With
End Sub

ClassModule: Tabelle3

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim lRow As Long
    If Target.Column > 2 Then Exit Sub
    If Target.Cells.Count > 1 Then Exit Sub
    With Worksheets("History")
        lRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1
        .Cells(lRow, 1).Value = Target.Address(False, False)
        .Cells(lRow, 2).Value = Now
        .Cells(lRow, 3).Value = Application.UserName
        .Cells(lRow, 4).Value = Target.Value
    End With
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)

End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchwert in zweiter Tabelle suchen und Fundzeile eintragen

Problem: Wie kann ich den Wert einer Eingabezeile in einer zweiten Tabelle suchen lassen und die Fundzeile in die Eingabetabelle übernehmen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle2

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim wks As Worksheet
    Dim rng As Range
    If Target.Address <> "$B$6" Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Set wks = Worksheets("Tabelle2")
    Set rng = wks.Cells.Find( _
        what:=Target.Value, _
        LookIn:=xlValues, _
        lookat:=xlWhole)
    If rng Is Nothing Then
        Range("A9:F9").ClearContents
    Else
        Range("A9:F9").Value = _
            wks.Range(wks.Cells(rng.Row, 1), _
                wks.Cells(rng.Row, 6)).Value
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabe auf bereits vorhanden prüfen und, wenn nein, einfügen

Problem:	Wie kann ich Zelleingaben darauf überprüfen, ob der Wert in einer Liste bereits vorhanden ist und diesen, wenn nicht, an das Ende der Liste anfügen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) Dim iRow As Integer If Target.Address <> "\$A\$1" Then Exit Sub If IsEmpty(Target) Then Exit Sub If WorksheetFunction.CountIf(Columns(1), Target.Value) > 1 Then MsgBox "Wert ist schon vorhanden!" Else iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 Cells(iRow, 1).Value = Target.Value End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eintrag in Spalte A aktuelle Zeile kopieren

Problem:	Wie kann ich veranlassen, daß bei einem beliebigen Eintrag in Spalte A die jeweilige Zeile in einer zweiten Tabelle in die erste frei Zeile eingetragen wird?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) Dim iRow As Integer, iRowL As Integer If Target.Column <> 1 Then Exit Sub If IsEmpty(Target) Then Exit Sub iRow = Target.Row With Worksheets("Tabelle2") If IsEmpty(.Range("A1")) Then iRowL = 1 Else iRowL = .Cells(Rows.Count, 1).End(xlUp).Row + 1 End If Range(Cells(iRow, 1), Cells(iRow, 5)).Copy .Cells(iRowL, 1) .Columns.AutoFit End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eingaben in Spalte B in Spalte A den Maximalwert anzeigen

Problem: Wie kann ich in Spalte A bei Eingaben in Spalte B den jeweiligen Maximalwert in Spalte B auflisten lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim iRow As Integer
    If Target.Column <> 2 Then Exit Sub
    iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
    Cells(iRow, 1) = WorksheetFunction.Max(Columns(Target.Column))
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Elemente der Gültigkeitsprüfung immer sortiert anzeigen

Problem: Wie kann ich die Elemente der Gültigkeitsprüfung immer in sortierter Reihenfolge anzeigen lassen, ohne daß der Listenbereich jedesmal manuell sortiert werden muß?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    If Intersect(Target, Range("A1:A10")) Is Nothing Then Exit Sub
    Range("A1:A10").Sort _
        Key1:=Range("A1"), _
        Order1:=xlAscending, _
        Header:=xlNo, _
        OrderCustom:=1, _
        MatchCase:=False, _
        Orientation:=xlTopToBottom
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eingabe von "j" "Ja" und von "n" "Nein" schreiben

Problem:	Wie kann ich bei Eingabe von "j" in Spalte A die Eingabe in "Ja" und von "n" "Nein" umwandeln lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Excel.Range) If Target.Column <> 1 Then Exit Sub If IsEmpty(Target) Then Exit Sub If Target.Value = "j" Then Target.Value = "Ja" ElseIf Target.Value = "n" Then Target.Value = "Nein" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bis zur 12. Zeile Cursor nach unten, dann nächste Spalte ansetzen

Problem: Wie kann ich festlegen, daß der Cursor bis zur 12. Zeile nach Eingaben immer die untere, danach die erste der nächsten Spalte auswählt?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Row < 12 Then
        Target.Offset(1, 0).Select
    Else
        Cells(1, Target.Column + 1).Select
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, welche Fahrer in welcher Zeit frei sind

Problem: Wie kann ich durch einfache Eingabe eines Datums, einer Anfangs- und Endzeit feststellen, welche Fahrer für diesen Zeitraum eine Tour übernehmen können und welche belegt sind?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle4

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim iWks As Integer, iRow As Integer
    Dim iColA As Integer, iColB As Integer
    Dim iTime As Integer
    Dim Besetzt As Boolean
    If Intersect(Target, Range("B1:B3")) Is Nothing Then Exit Sub
    If IsEmpty(Range("B1")) Or IsEmpty(Range("B3")) Or _
        IsEmpty(Range("B3")) Then Exit Sub
    For iWks = 3 To Worksheets.Count
        Besetzt = False
        With Worksheets(iWks)
            iRow = WorksheetFunction.Match( _
                CDbl(Range("B1").Value), .Columns(1))
            iColA = .Rows("1").Find(TimeValue(Range("B2").Text), _
                LookIn:=xlFormulas, lookat:=xlWhole).Column
            iColB = .Rows("1").Find(TimeValue(Range("B3").Text), _
                LookIn:=xlFormulas, lookat:=xlWhole).Column
            For iTime = iColA To iColB
                If Not IsEmpty(.Cells(iRow, iTime)) Then
                    Cells(iWks - 2, 4) = "Besetzt"
                    Besetzt = True
                    Exit For
                End If
            Next iTime
            If Besetzt = False Then Cells(iWks - 2, 4) = "Frei"
        End With
    Next iWks
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Eintragungen Wert des Eintrags plus Vorgangswert anzeigen

Problem:	Wie kann ich mir bei Eingaben in Spalte A den aktuellen Wert plus dem vorher in der Zelle befindlichen Wert anzeigen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Excel.Range) If Not IsNumeric(Target) Or IsEmpty(Target) Then Exit Sub If Target.Column <> 1 Then Exit Sub Dim dValue As Double, dPlus As Double Application.EnableEvents = False On Error GoTo ERRORHANDLER dPlus = Target.Value Application.Undo dValue = dPlus + Target.Value Target.Value = dPlus MsgBox "Gesamtwert: " & dValue ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

CheckBoxes in einer Tabelle über Zelleingaben steuern

Problem: Wie kann ich eine Serie von CheckBoxes in einem Tabellenblatt in Abhängigkeit von Zelleingaben steuern?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim shp As Shape
    Dim iCounter As Integer
    If Intersect(Target, Range("A1:A10")) Is Nothing Then Exit Sub
    For iCounter = 1 To 10
        If WorksheetFunction.CountIf(Columns(1), iCounter) > 0 Then
            OLEObjects(iCounter).Object.Value = True
        Else
            OLEObjects(iCounter).Object.Value = False
        End If
    Next iCounter
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert einer bestimmten Zelle merken

Problem:	Wie kann ich erreichen, daß sich Excel den Wert einer bestimmten Zelle merkt? Nach dem Überschreiben des ursprünglichen Wertes soll dieser jedesmal angezeigt werden, wenn der Cursor sich in dieser Zelle befindet.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(_ ByVal Target As Excel.Range) Dim varNew As Variant If Target.Address <> "\$D\$5" Then Exit Sub varB = varA varA = Target End Sub Private Sub Worksheet_SelectionChange(_ ByVal Target As Excel.Range) If Target.Address <> "\$D\$5" Then Exit Sub varA = Target MsgBox varA End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrundfarbe nach Eingabewert

Problem:	Wie kann ich bei die Zellhintergrundfarbe bei Eingabewerten von über 10 auf Gelb und andernfalls auf Normal setzen lassen? Die bedingte Formatierung soll nicht eingesetzt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(_ ByVal Target As Excel.Range) If Intersect(Target, Range("A1:B10")) _ Is Nothing Then Exit Sub If Target.Value > 10 Then Target.Interior.ColorIndex = 6 Else Target.Interior.ColorIndex = xlNone End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Eingabe eines Kurzbegriffes Werte in Nebenzellen eintragen

Problem:	Wie kann ich veranlassen, daß nach der Eingabe eines Kurzbegriffes in den beiden Zellen rechts neben der Eingabezelle Daten aus einem zweiten Tabellenblatt in Abhängigkeit vom Kurzbegriff übernommen werden?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) Dim var As Variant If Target.Column <> 1 Then Exit Sub If Target.Cells.Count > 1 Then Exit Sub Application.EnableEvents = False On Error GoTo ERRORHANDLER With Worksheets("Tabelle2") If Not IsEmpty(Target) Then var = Application.Match(Target.Value, .Columns(1), 0) If Not IsError(var) Then Target.Offset(0, 1).Value = .Cells(var, 2).Value Target.Offset(0, 2).Value = .Cells(var, 3).Value End If Else Range(Target, Target.Offset(0, 2)).ClearContents End If End With ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellen für die Werktage eines Monats erstellen

Problem:	Wie kann ich durch die Eintragung einer Jahres- bzw. Monatszahl für jeden Tag des jeweiligen Monats ein Arbeitsblatt generieren lassen? Die Arbeitsblätter sollen als Name den Tag und den Wochen tag tragen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Excel.Range) Dim iYear As Integer, iMonth As Integer, iDay As Integer Application.ScreenUpdating = False If Intersect(Target, Range("B1:B2")) Is Nothing Then Exit Sub If IsEmpty(Range("B1")) Or IsEmpty(Range("B2")) Then Exit Sub iYear = Range("B1") iMonth = Range("B2") For iDay = 1 To Day(DateSerial _ (iYear, iMonth + 1, 0)) If WeekDay(DateSerial(iYear, _ iMonth, iDay), 2) < 6 Then Worksheets("Tag").Copy after:=Worksheets _ (Worksheets.Count) ActiveSheet.Name = iDay & " " & Format(_ DateSerial(iYear, iMonth, iDay), "dddd") End If Next iDay Worksheets(1).Select Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Den vorhergehenden Zellwert in einer zweiten Zelle zeigen

Problem: Wie kann ich bei Eingaben in eine Zelle den jeweils vorhergehenden Zellwert in einer zweiten Zelle anzeigen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim var As Variant
    If Target.Address <> "$A$1" Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    var = Target.Value
    Application.Undo
    Range("A2").Value = Range("A1").Value
    Target.Value = var
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einfacher Zellbezug greift auf geschlossene Arbeitsmappe zu

Problem: Wie kann ich durch Eingabe eines einfachen Zellbezugs - =F17 - auf eine geschlossene Arbeitsmappe zugreifen, deren Name in der ersten Zeile der Eingabespalte steht?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim sPath As String, sFile As String, sFormula As String
    If Target.Row <> 3 Then Exit Sub
    If IsEmpty(Target.Offset(-1, 0)) Or _
        IsEmpty(Target.Offset(-1, 0)) Then Exit Sub
    sPath = Range("B6").Value & "\"
    sFile = Cells(1, Target.Column).Value & ".xls"
    If Dir(sPath & sFile) = "" Then
        Beep
        MsgBox "Datei wurde nicht gefunden!"
        Exit Sub
    End If
    sFormula = "'" & sPath & "[" & sFile & "]"
    sFormula = sFormula & Cells(2, Target.Column).Value & "!"
    sFormula = sFormula & Cells(3, Target.Column).Value
    With Cells(4, Target.Column)
        .Formula = "=" & sFormula
        .Value = .Value
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelleingaben automatisch auf volle 1,2-Teile runden

Problem:	Wie kann ich Zelleingaben automatisch auf volle 1,2 runden lassen? Aus 1,1 soll 1,2, aus 3,1 soll 3,6 werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Column <> 2 Then Exit Sub If IsEmpty(Target) Then Exit Sub If Not IsNumeric(Target.Value) Then Exit Sub Application.EnableEvents = False On Error GoTo ERRORHANDLER Target.Value = Target.Value * 1.2 ERRORHANDLER: End Sub Private Sub Worksheet_SelectionChange(ByVal Target As Range) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zugriff auf geschlossene Arbeitsmappe

Problem:	Wie kann ich auf Zellen einer geschlossenen Arbeitsmappe zugreifen, ohne jedesmal den vollen Dateinamen angeben zu müssen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Excel.Range) Dim sTxt As String If IsEmpty(Target) Then Exit Sub If Target.Cells.Count > 1 Then Exit Sub If Target.Address <> "\$E\$1" Then Exit Sub Application.EnableEvents = False On Error GoTo ERRORHANDLER sTxt = "" & Range("B1").Value sTxt = sTxt & "[" & Range("B2").Value & "]" sTxt = sTxt & Range("B3").Value & "!" sTxt = sTxt & Target.Value Target.Formula = "=" & sTxt Target.Value = Target.Value ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prozentuale Erhöhung über mehrere Arbeitsblätter hinweg

Problem: Wie kann ich erreichen, daß sich alle Werte in mehreren Tabellen nach Eingabe einer Zahl in einer bestimmten Zelle prozentual um diese Zahl erhöhen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle2

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim iWks As Integer
    If Target.Address <> "$E$1" Then Exit Sub
    If Target.Cells.Count > 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Application.ScreenUpdating = False
    Range("E1").Copy
    For iWks = 2 To 5
        Worksheets(iWks).Range("A1:D21").PasteSpecial _
            Paste:=xlValues, Operation:=xlMultiply, _
            SkipBlanks:=False, Transpose:=False
    Next iWks
    Application.CutCopyMode = False
    Range("A1").Select
    Application.ScreenUpdating = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wiederherstellung des alten Zellinhaltes bei unerlaubten Werten

Problem: Wie kann ich den ursprünglichen Zellinhalt wiederherstellen, wenn in eine Zelle, in der nur Ziffern erlaubt sind, Zeichenfolgen eingegeben wurden?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    If Not IsNumeric(Target.Value) Then
        Application.Undo
        Beep
        MsgBox "Nur Zahlen erlaubt!"
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Faxbestellung auf Grundlage einer Artikelliste erstellen

Problem: Wie kann ich eine Faxbestellung auf Grundlage einer Artikelliste erstellen? Das Bestellformular soll sowohl vervollständigt werden, wenn ich im Formular eine Artikelnummer eingebe wie auch, wenn ich in der Artikelliste eine Stückzahl eintrage.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim wks As Worksheet
    Dim var As Variant
    If Target.Column <> 1 Then Exit Sub
    If Target.Row < 8 Then Exit Sub
    If Target.Cells.Count > 1 Then Exit Sub
    Set wks = Worksheets("Artikelliste")
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    If IsEmpty(Target) Then
        Range(Cells(Target.Row, 1), _
            Cells(Target.Row, 5)).ClearContents
    Else
        var = Application.Match(Target.Value, wks.Columns(1), 0)
        If Not IsError(var) Then
            Target.Offset(0, 1).Value = wks.Cells(var, 2).Value
            Target.Offset(0, 3).Value = wks.Cells(var, 3).Value
            Target.Offset(0, 4).Formula = "=RC[-2]*RC[-1]"
        End If
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```

```
ClassModule: Tabelle3

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim wks As Worksheet
    Dim iRow As Integer
    If Target.Column <> 4 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Target.Cells.Count > 1 Then Exit Sub
    Set wks = Worksheets("Fax-Bestellung")
    iRow = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1
    wks.Cells(iRow, 1).Value = Cells(Target.Row, 1).Value
    wks.Cells(iRow, 2).Value = Cells(Target.Row, 2).Value
    wks.Cells(iRow, 3).Value = Target.Value
    wks.Cells(iRow, 4).Value = Cells(Target.Row, 3).Value
    wks.Cells(iRow, 5).Formula = "=RC[-2]*RC[-1]"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabe der Artikelnummer und Rabattberechnung

Problem:	Wie kann ich durch die Eingabe einer Artikelnummer den Import der Warenbenennung und des Preises aus anderen Tabellenblätter veranlassen? Bei den preisen soll eine Rabattberechnung anhand einer vorhandenen Rabatt-Tabelle vorgenommen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) Dim wks As Worksheet, wksRabatt As Worksheet Dim varArtikel As Variant, varRabatt As Variant Dim dValue As Double, dNetto As Double Dim iWks As Integer If Target.Column <> 1 Then Exit Sub If IsEmpty(Target) Then Exit Sub If Target.Cells.Count > 1 Then Exit Sub Set wksRabatt = Worksheets("Rabatt") For iWks = 3 To 5 varArtikel = Application.Match(Target.Value, Worksheets(iWks).Columns(1), 0) If Not IsError(varArtikel) Then Set wks = Worksheets(iWks) Exit For End If Next iWks If IsError(varArtikel) Then Exit Sub Else dValue = wks.Cells(varArtikel, 3).Value varRabatt = Application.Match(wks.Name, wksRabatt.Columns(4), 0) If Not IsError(varRabatt) Then dValue = dValue - (dValue * wksRabatt.Cells(varRabatt, 5).Value / 100) End If varRabatt = Application.Match(Target.Value, wksRabatt.Columns(1), 0) If Not IsError(varRabatt) Then dValue = dValue - (dValue * wksRabatt.Cells(varRabatt, 2).Value / 100) End If Target.Offset(0, 1).Value = wks.Cells(varArtikel, 2).Value Target.Offset(0, 2).Value = dValue End If End Sub Private Sub Worksheet_SelectionChange(ByVal Target As Range) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensatz nach Eingabe eines Kurzbegriffes

Problem: Wie kann ich einen Datensatz nach Eingabe eines Kurzbegriffes abrufen? Wenn ich beispielsweise "web" eingebe, soll die Anschrift der Firma Weber GmbH & Co. KG eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim wks As Worksheet
    Dim var As Variant
    Dim iRow As Integer
    If Target.Address <> "$A$1" Then Exit Sub
    Set wks = Worksheets("Datenbank")
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    If IsEmpty(Target) Then
        Range("A1:A4").ClearContents
    Else
        var = Application.Match( _
            "*" & Target.Value & "*", wks.Columns(1), 0)
        If Not IsError(var) Then
            For iRow = 1 To 3
                Cells(iRow, 1).Value = wks.Cells(var, iRow).Value
            Next iRow
            Cells(4, 1).Value = wks.Cells(var, 4).Value & " " & _
                wks.Cells(var, 5).Value
        Else
            Range("A1:A4").ClearContents
        End If
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Eingabe Werte aus Tabellen Übernehmen

Problem: Wie kann ich je nach Eingabe einer Artikelnummer Werte aus verschiedenen Tabellen übernehmen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle3

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim wksWaren As Worksheet, wksPlatten As Worksheet
    Dim var As Variant
    Dim iRow As Integer, iCol As Integer
    Set wksWaren = Worksheets("Warenliste")
    Set wksPlatten = Worksheets("Platten")
    If Target.Rows.Count > 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    Select Case Target.Column
        Case 1
            If Target.Value <> "AAA" Then
                var = Application.Match( _
                    Target.Value, wksWaren.Columns(1), 0)
                If IsError(var) Then
                    MsgBox "Artikelnummer nicht gefunden," & _
                        " bitte Neueingabe!"
                    Target.ClearContents
                    Exit Sub
                End If
                Target.Offset(0, 1) = wksWaren.Cells(var, 2).Value
                Target.Offset(0, 4) = wksWaren.Cells(var, 3).Value
                Target.Offset(1, 0).Select
            Else
                Target.Offset(0, 1) = "Tischlerplatte"
                Target.Offset(0, 2).Select
            End If
        Case 3
            If Target.Offset(0, -2) = "AAA" Then _
                Target.Offset(0, 1).Select
        Case 4
            iRow = WorksheetFunction.Match(Target.Offset(0, -1).Value, _
                wksPlatten.Columns(1))
            iCol = WorksheetFunction.Match(Target.Value, wksPlatten.Rows(3))
            Target.Offset(0, 1).Value = wksPlatten.Cells(iRow, iCol).Value
            Target.Offset(1, -3).Select
    End Select
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Datensätze >= Eingabewert filtern

Problem: Wie kann ich alle Datensätze, deren Wert in der Eingabespalte >= der aktuellen Eingabe ist, in ein zweites Tabellenblatt filtern?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle11

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim iRow As Integer, iRowL As Integer
    If Target.Column <> 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Not IsNumeric(Target.Value) Then Exit Sub
    iRow = 2
    Do Until IsEmpty(Cells(iRow, 1))
        If Cells(iRow, 1).Value >= Target.Value And _
            iRow <> Target.Row Then
            With Worksheets("Tabelle2")
                iRowL = .Cells(Rows.Count, 1).End(xlUp).Row + 1
                .Range(.Cells(iRowL, 1), .Cells(iRowL, 4)).Value = _
                    Range(Cells(iRow, 1), Cells(iRow, 4)).Value
            End With
        End If
        iRow = iRow + 1
    Loop
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Balkendiagramm simulieren

Problem: Wie kann ich in einem Tabellenblatt die Hintergrundfarbe von Zellen bei Zelleingaben so verändern, daß der Eindruck eines Balkendiagramms entsteht?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle11

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim iCounter As Integer
    If Target.Row <> 18 Then Exit Sub
    If Target.Column Mod 3 = 1 Or _
        Target.Column > 18 Then Exit Sub
    If IsNumeric(Target.Value) = False Then
        Beep
        MsgBox "Nur Zahlen zwischen 0 und 10 erlaubt!"
        End
    End If
    If Target.Value > 10 Or Target.Value < 0 Then
        Beep
        MsgBox "Nur Zahlen zwischen 0 und 10 erlaubt!"
        End
    End If
    For iCounter = 11 To 11 - Target.Value + 1 Step -1
        With Cells(iCounter, Target.Column)
            If IsEmpty(Target.Offset(0, -1)) Then
                .Interior.ColorIndex = 5
                .Font.ColorIndex = 3
            Else
                .Interior.ColorIndex = 3
                .Font.ColorIndex = 5
            End If
        End With
    Next iCounter
    For iCounter = 11 - Target.Value To 2 Step -1
        With Cells(iCounter, Target.Column)
            If IsEmpty(Target.Offset(0, -1)) Then
                .Interior.ColorIndex = xlNone
                .Font.ColorIndex = 3
            Else
                .Interior.ColorIndex = xlNone
                .Font.ColorIndex = 3
            End If
        End With
    Next iCounter
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Warnung, wenn Wert bereits vorhanden

Problem: Wie kann ich bei Eingabe eines in einer Spalte bereits vorhandenen Wertes eine Warnmeldung anzeigen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle11

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    If Target.Column <> 4 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Target.Cells.Count > 1 Then Exit Sub
    If WorksheetFunction.CountIf(Columns(4), Target.Value) > 1 Then
        If MsgBox("Lieferschein schon eingetragen!" & _
            vbCrLf & "Dennoch eintragen?", _
            vbCritical + vbYesNo) = vbNo Then
            Application.EnableEvents = False
            Target.ClearContents
            Application.EnableEvents = True
        End If
    End If
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalt und Kommentar nach Eingabe festlegen

Problem: Wie kann ich den Inhalt einer Zelle nach Eingabe eines Kurzbegriffes durch Festlegungen in einem zweiten Tabellenblatt ersetzen und der Zelle einen dort hinterlegten Kommentar zuordnen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle21

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim wks As Worksheet
    Dim oCmt As Comment
    Dim var As Variant
    Set wks = Worksheets("Kürzel")
    If Target.Column <> 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Target.Cells.Count > 1 Then Exit Sub
    Application.DisplayCommentIndicator = xlCommentIndicatorOnly
    var = Application.Match(Target.Value, wks.Columns(1), 0)
    If Not IsError(var) Then
        Application.EnableEvents = False
        Target.Value = wks.Cells(var, 2).Value
        If Not Target.Comment Is Nothing Then
            Target.Comment.Delete
        End If
        Set oCmt = Target.AddComment(wks.Cells(var, 3).Value)
        oCmt.Shape.TextFrame.AutoSize = True
        On Error GoTo ERRORHANDLER
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

MsgBox mit Werten aus geschlossener Arbeitsmappe

Problem: Wie kann ich mir beim Eintragen einer Artikelnummer die Daten dieses Artikels aus einer geschlossenen Arbeitsmappe in einer MsgBox anzeigen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle12

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim sPath As String, sFile As String, sWks As String
    Dim sFormula As String, sFormulaA As String, sFormulaB As String
    If Target.Column <> 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Not IsNumeric(Target.Value) Then Exit Sub
    If Target.Value > 20 Then Exit Sub
    sPath = ThisWorkbook.Path
    sFile = "Test1.xls"
    If Dir(sPath & "\" & sFile) = "" Then
        Beep
        MsgBox "Testdatei Test1.xls wurde nicht gefunden!"
        Exit Sub
    End If
    sWks = "048199"
    sFormula = "VLOOKUP(A" & Target.Row & ", "
    sFormula = sFormula & "'" & sPath & "\" & sFile & "]" & sWks & "'"
    sFormulaA = sFormula & "A:C,2,0)"
    sFormulaB = sFormula & "A:C,3,0)"
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    Target.Offset(0, 1).Formula = "=" & sFormulaA
    Target.Offset(0, 2).Formula = "=" & sFormulaB
    With Range(Target.Offset(0, 1), Target.Offset(0, 2))
        .Value = .Value
    End With
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zwei Tabellen wechselseitig ändern

Problem:	Wie kann ich Eintragungen in der einen automatisch in der anderen Tabelle anzeigen lassen? Ein zusätzliches Problem ist, daß die eine Tabelle ein Auszug der anderen ist, d.h. sie hat nur einen Teil der Spalten der ersten Tabelle.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) Dim iCol As Integer iCol = Target.Column If iCol <> 1 And iCol <> 4 And iCol <> 7 Then Exit Sub Call Eintragen(Me, Target.Row, Target.Column) End Sub ClassModule: Tabelle3 Private Sub Worksheet_Change(ByVal Target As Range) Dim iCol As Integer iCol = Target.Column If iCol <> 1 And iCol <> 2 And iCol <> 3 Then Exit Sub Call Eintragen(Me, Target.Row, Target.Column) End Sub StandardModule: basMain Sub Eintragen(wks As Worksheet, iRow As Integer, iCol As Integer) Dim wksA As Worksheet If wks.Name = "Tabelle1" Then Set wksA = Worksheets("Tabelle2") Else Set wksA = Worksheets("Tabelle1") End If Select Case wks.Name Case "Tabelle2" Select Case iCol Case 1 wksA.Cells(iRow, 1).Value = wks.Cells(iRow, iCol).Value Case 4 wksA.Cells(iRow, 2).Value = wks.Cells(iRow, iCol).Value Case 7 wksA.Cells(iRow, 3).Value = wks.Cells(iRow, iCol).Value End Select Case Else Select Case iCol Case 1 wksA.Cells(iRow, 1).Value = wks.Cells(iRow, iCol).Value Case 2 wksA.Cells(iRow, 4).Value = wks.Cells(iRow, iCol).Value Case 3 wksA.Cells(iRow, 7).Value = wks.Cells(iRow, iCol).Value End Select End Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Monatsblatt anlegen

Problem: Wie kann ich durch Eingabe der Monats- bzw. Jahreszahl automatisch die Tage als Zahl und im Format DDD eintragen lassen. Die Routine soll automatisch die Anzahl der Tage erkennen und die Wochenenden farbig hinterlegen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim lDay As Long, lStart As Long, lEnd As Long
    Dim iDay As Integer
    If Target.Address <> "$B$1" And _
        Target.Address <> "$D$1" Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    With Rows("2:3")
        .ClearContents
        .ClearFormats
        .HorizontalAlignment = xlCenter
    End With
    lStart = DateSerial(Range("D1").Value, Range("B1").Value, 1)
    lEnd = DateSerial(Range("D1").Value, Range("B1").Value + 1, 0)
    For lDay = lStart To lEnd
        iDay = iDay + 1
        Cells(2, iDay).Value = iDay
        Cells(3, iDay).Value = Format(lDay, "dddd")
        If Weekday(lDay) = 7 Then
            Cells(3, iDay).Interior.ColorIndex = 34
        ElseIf Weekday(lDay) = 1 Then
            Cells(3, iDay).Interior.ColorIndex = 35
        End If
    Next lDay
    Me.Name = Format(DateSerial(1, Range("B1").Value, 1), "mmm")
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabeerleichterung bei Datumseingaben

Problem: Wie kann ich Datumseingaben im Format TTMMJJ tätigen und Excel zur Umwandlung ins Format TT.MM.JJ veranlassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle2

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim vTxt As Variant
    If Target.Column <> 1 Then Exit Sub
    If Len(CStr(CDbl(Target.Value))) < 5 Then Exit Sub
    If Not IsDate(Target.Value) Then Exit Sub
    vTxt = CStr(CDbl(Target.Value))
    If Len(vTxt) = 5 Then vTxt = "0" & vTxt
    vTxt = DateSerial(Right(vTxt, 2), Mid(vTxt, 3, 2), Left(vTxt, 2))
    On Error GoTo ERRORHANDLER
    Application.EnableEvents = False
    Target.Value = vTxt
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ListBox bei Eintrag erweitern

Problem: Wie kann ich erreichen, daß sich die ListFillRange einer ListBox bei Einträgen in der Liste automatisch erweitert. Die ListBox befindet sich in dem Tabellenblatt mit der Liste.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim iRow As Integer
    If Target.Column <> 1 Or Target.Row = 1 Then Exit Sub
    iRow = Cells(Rows.Count, 1).End(xlUp).Row
    ListBox1.ListFillRange = Range("A2:A" & iRow).Address
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eintragung dokumentieren

Problem:	In Spalte A steht der Warenbestand und in Spalte B das Datum. Wenn der Warenbestand geändert wird, soll automatisch das Datum aktualisiert werden. Wie mache ich das?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Target.Row = 1 Then Exit Sub If Target.Column <> 1 Then Exit Sub Target.Offset(0, 1) = Now End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Änderungsdatum eintragen

Problem:	Ich möchte in Spalte AA bei jeder Änderung im Tabellenblatt das Datum der Änderung und den jeweiligen Anwendernamen eintragen lassen. Wie mache ich das?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim iRow As Integer If Target.Column = 1 Then Exit Sub If IsEmpty(Range("A1")) Then iRow = 1 Else iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 End If Cells(iRow, 1) = "Geändert von " & _ Application.UserName & _ " am " & _ Format(Now, "dd.mm.yy - hh:mm:ss") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Änderung der Kopfzeile bei Zelleintrag

Problem:	Wie kann ich in XL8 automatisch die Kopfzeile ändern lassen, wenn sich die Einträge in Zelle B6 oder D7 ändern?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Dim sAdr As String sAdr = Target.Address If sAdr <> "\$B\$6" And sAdr <> "\$D\$7" Then Exit Sub sAdr = CStr(Range("B6")) & "/" & CStr(Range("D7")) ActiveSheet.PageSetup.LeftHeader = sAdr ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vervollständigen von Zelleingaben

Problem: Wie kann ich veranlassen, daß nach Eingabe einer Zahl ein bestimmter Text in die Eingabezelle geschrieben wird?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle2

```
Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Cells.Count > 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Target.Column <> 1 Then Exit Sub
    Application.EnableEvents = False
    On Error GoTo ERRORHANDLER
    Select Case Target.Value
        Case 4: Target = "4 - Äpfel"
        Case 5: Target = "5 - Birnen"
        Case 6: Target = "6 - Schornsteinfeger"
        Case 7: Target = "7 - Pfirsische"
    End Select
ERRORHANDLER:
    Application.EnableEvents = True
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)

End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellwert in Kommentar übernehmen

Problem:	Zelleinträge sollen in Kommentare eingefügt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	ClassModule: Tabelle1

```
Private Sub Worksheet_Change( _  
    ByVal Target As Excel.Range)  
    Dim cmt As Comment  
    Application.DisplayCommentIndicator = _  
        xlCommentIndicatorOnly  
    If Not Target.Comment Is Nothing Then  
        Target.Comment.Delete  
    End If  
    Set cmt = Target.AddComment( _  
        Text:="Zellinhalt: " & Target.Value)  
    cmt.Shape.TextFrame.AutoSize = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingaben aufaddieren

Problem:	Wie kann ich Zelleingaben in einer zweiten Zelle addieren und in einer dritten Zelle zählen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Excel.Range) If Target.Address <> "\$A\$1" Then Exit Sub If Target.Cells.Count > 1 Then Exit Sub If IsEmpty(Target) Then Exit Sub If Not IsNumeric(Target.Value) Then Exit Sub Application.EnableEvents = False On Error GoTo ERRORHANDLER Target.Offset(0, 1).Value = _ Target.Offset(0, 1).Value + Target.Value Target.Offset(0, 2).Value = _ Target.Offset(0, 2).Value + 1 ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabe in Großbuchstaben wandeln

Problem:	Wie kann ich erreichen, daß Zelleingaben automatisch in Großbuchstaben umgewandelt werden?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Excel.Range) If IsEmpty(Target) Then Exit Sub If Intersect(Target, Range("B3:F20")) _ Is Nothing Then Exit Sub Application.EnableEvents = False On Error GoTo ERRORHANDLER Target = UCase(Target) ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makroaufruf abhängig vom Zellwert

Problem: Wie kann ich bei XL8 in Abhängigkeit vom Zellwert der Eingabezelle ein bestimmtes Makro aufrufen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    If Target.Address <> "$A$1" Then Exit Sub
    If IsNumeric(Target) = False Then Exit Sub
    Select Case Target
        Case 1: Call Macro1
        Case 2: Call Macro2
    End Select
End Sub

Sub Macro1()
    MsgBox "Ich bin Makro1"
End Sub

Sub Macro2()
    MsgBox "Ich bin Makro2"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makros in Abhängigkeit vom Zellennamen aufrufen

Problem: Wie kann ich eine Reihe von Unterprogrammen in Abhängigkeit vom Namen der Eingabezelle aufrufen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Cells.Count > 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    If Intersect(Target, Range("B1:B3")) Is Nothing Then Exit Sub
    On Error GoTo ERRORHANDLER
    Application.EnableEvents = False
    Select Case Target.Name.Name
        Case "Gewicht": Call Gewicht
        Case "Alter": Call Alter
        Case "Hautfarbe": Call Hautfarbe
    End Select
ERRORHANDLER:
    Application.EnableEvents = True
End Sub

StandardModule: basMain

Sub Gewicht()
    ActiveCell.Value = 125
End Sub

Sub Alter()
    ActiveCell.Value = 54
End Sub

Sub Hautfarbe()
    ActiveCell.Value = "Pink"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei falscher Eingabe alten Wert wiederherstellen

Problem: Ich möchte den Benutzer zwingen, nur numerische Werte in einen bestimmten Bereich einzugeben. Gibt er einen nichtnumerischen Wert ein, soll der alte Wert wiederhergestellt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
StandardModule: basMain

Public gvValue As Variant

ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    If ActiveCell.Column = 256 Then
        Range("A1").Select
    Else
        ActiveCell.Offset(0, 1).Select
        ActiveCell.Offset(0, -1).Select
    End If
End Sub

ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    If Not IsNumeric(Target.Value) Then
        Target.Value = gvValue
    End If
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)
    gvValue = ActiveCell.Value
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelleingaben speichern

Problem:	In der Zelle A1 werden laufend Daten aktualisiert. Damit die alten Daten nicht verlorengehen, möchte ich diese in Spalte B mit dem Erstellungsdatum in Spalte C und die Erstellungsuhrzeit in Spalte D protokollieren.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) Dim iRow As Integer If Target.Address <> "\$A\$1" Then Exit Sub iRow = Cells(Rows.Count, 2).End(xlUp).Row + 1 If IsEmpty(Cells(1, 2)) Then iRow = 1 Cells(iRow, 2) = Target.Value Cells(iRow, 3) = Date Cells(iRow, 4) = Time End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gültigkeitsliste in Abhängigkeit einer Werteliste

Problem: Im Bereich A7:D11 dürfen über eine Auswahlliste immer nur die Werte ausgewählt werden können, die in der jeweiligen Spalte im Vergleich zum Bereich A1:D5 noch frei sind.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim iRow As Integer, iCol As Integer
    Dim sVal As String
    If Target.Cells.Count > 1 Then Exit Sub
    If Intersect(Target, Range("A7:D11")) Is Nothing Then Exit Sub
    iCol = Target.Column
    For iRow = 1 To 5
        If IsError( _
            Application.Match( _
                Cells(iRow, iCol).Value, _
                Range(Cells(7, iCol), Cells(11, iCol)), 0)) Then
            If sVal <> "" Then
                sVal = sVal & "," & Cells(iRow, iCol).Value
            Else
                sVal = Cells(iRow, iCol).Value
            End If
        End If
    Next iRow
    If sVal = "" Then
        Range(Cells(7, iCol), Cells(11, iCol)).Validation.Delete
    Else
        For iRow = 7 To 11
            With Cells(iRow, iCol).Validation
                .Delete
                .Add _
                Type:=xlValidateList, _
                AlertStyle:=xlValidAlertStop, _
                Operator:=xlBetween, _
                Formula1:=sVal
                .IgnoreBlank = True
            End With
        Next iRow
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei jedem Aufruf eine andere Prozedur aufrufen.

Problem: Bei jedem Schaltflächenklick soll eine andere Prozedur aufgerufen werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdAction_Click()
    Run ThisWorkbook.Name & "!" & "MyAction" & Range("C4").Value
    With Range("C4")
        If .Value = Range("B1").Value Then
            .Value = 1
        Else
            .Value = .Value + 1
        End If
    End With
End Sub

StandardModule: Modul1

Sub MyAction1()
    MsgBox "Aktion Nr. 1"
End Sub

Sub MyAction2()
    MsgBox "Aktion Nr. 2"
End Sub

Sub MyAction3()
    MsgBox "Aktion Nr. 3"
End Sub

Sub MyAction4()
    MsgBox "Aktion Nr. 4"
End Sub

Sub MyAction5()
    MsgBox "Aktion Nr. 5"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Feststellen, ob eine Schaltfläche mit Shift-Taste gedrückt wurde

Problem: Wie kann ich feststellen, ob eine Tabellenblattschaltfläche mit oder ohne gehaltene Hochschalttaste gedrückt wurde?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdVerzweigung_MouseDown( _
    ByVal Button As Integer, ByVal Shift As Integer, _
    ByVal X As Single, ByVal Y As Single)
    If Shift = 0 Then Call Meldung1 Else Call Meldung2
End Sub

Sub Meldung1()
    MsgBox "Shift-Taste nicht gedrückt!"
End Sub

Sub Meldung2()
    MsgBox "Shift-Taste gedrückt!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schaltflächenfarbe beim Überfahren mit der Maus ändern

Problem: Beim Überfahren der Schaltflächen mit der Maus sollen Ampelfarben angezeigt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CommandButton1_MouseMove( _
    ByVal Button As Integer, _
    ByVal Shift As Integer, _
    ByVal X As Single, _
    ByVal Y As Single)
    CommandButton1.BackColor = vbRed
    CommandButton2.BackColor = vbWhite
    CommandButton3.BackColor = vbWhite
End Sub

Private Sub CommandButton2_MouseMove( _
    ByVal Button As Integer, _
    ByVal Shift As Integer, _
    ByVal X As Single, _
    ByVal Y As Single)
    CommandButton1.BackColor = vbWhite
    CommandButton2.BackColor = vbYellow
    CommandButton3.BackColor = vbWhite
End Sub

Private Sub CommandButton3_MouseMove( _
    ByVal Button As Integer, _
    ByVal Shift As Integer, _
    ByVal X As Single, _
    ByVal Y As Single)
    CommandButton1.BackColor = vbWhite
    CommandButton2.BackColor = vbWhite
    CommandButton3.BackColor = vbGreen
End Sub

Private Sub Label1_MouseMove( _
    ByVal Button As Integer, _
    ByVal Shift As Integer, _
    ByVal X As Single, _
    ByVal Y As Single)
    CommandButton1.BackColor = vbWhite
    CommandButton2.BackColor = vbWhite
    CommandButton3.BackColor = vbWhite
End Sub

StandardModule: Modul1

Sub Dummy()
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle alle 10 Sekunden sortieren

Problem:	Eine Tabelle soll alle 10 Sekunden neu sortiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn zwei Schaltflächen zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) On Error Resume Next Call SortierungStop End Sub StandardModule: basMain Public Const ciIntervall As Integer = 10 Public Const dsMacro As String = "Berechnen" Public gdNextTime As Double Public gvar As Variant Sub SortierungStart() gdNextTime = Now + TimeSerial(0, 0, ciIntervall) Application.OnTime gdNextTime, dsMacro End Sub Sub Berechnen() Range("A1").CurrentRegion.Sort key1:=Range("A2"), order1:=xlAscending Call SortierungStart End Sub Sub SortierungStop() On Error Resume Next Application.OnTime earliesttime:=gdNextTime, _ procedure:=dsMacro, schedule:=False Application.Calculation = gvar End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafischen Webcounter zeilenweise einlesen

Problem:	Excel/VBA-Beispiel: Grafischen Webcounter zeilenweise einlesen
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn zwei Schaltflächen zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call StopCounter End Sub StandardModule: Modul1 Public Const gsMacro As String = "UpdateCounter" Public gdNextTime As Double Private Declare Function URLDownloadToFile Lib "urlmon" _ Alias "URLDownloadToFileA" (_ ByVal pCaller As Long, _ ByVal szURL\$, _ ByVal szFileName\$, _ ByVal dwReserved As Long, _ ByVal lpfnCB As Long) As Long Sub StartCounter() Dim iIntervall As Integer iIntervall = Range("C1").Value gdNextTime = Now + TimeSerial(0, 0, iIntervall) Application.OnTime earliesttime:=gdNextTime, _ procedure:=gsMacro, schedule:=True End Sub Sub StopCounter() On Error Resume Next Application.OnTime earliesttime:=gdNextTime, _ procedure:=gsMacro, schedule:=False End Sub Private Sub UpdateCounter() Dim pct As Picture Dim lResult As Long Dim sUrl\$, sLocalFile\$ Dim iRow As Integer Dim sFile As String iRow = Cells(Rows.Count, 1).End(xlUp).Row + 1 Cells(iRow, 1).Select sUrl = "http://s89.gratiscounter.de/hit.cgi?24374" sFile = Application.DefaultFilePath & "\counter.gif" lResult = URLDownloadToFile(0, sUrl, sFile, 0, 0) Cells(iRow, 1).Value = "X" Set pct = ActiveSheet.Pictures.Insert(sFile) Rows(iRow).RowHeight = pct.Height Call StartCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeit in vorgegebenem Intervall hochzählen

Problem:	Die Zeit in Zelle B1 soll im Sekunden-Intervall um die in Zelle E1 genannten Minuten hochgezählt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Public Const gsMacro As String = "UpdateClock" Public gdNextTime As Double Sub StartClock() Dim iIntervall As Integer iIntervall = Range("E1").Value gdNextTime = Now + TimeSerial(0, 0, 1) Application.OnTime earliesttime:=gdNextTime, _ procedure:=gsMacro, schedule:=True End Sub Sub UpdateClock() Dim wks As Worksheet Range("B1").Value = Range("B1").Value + _ TimeSerial(0, Range("E1").Value, 0) Call StartClock End Sub Sub StopClock() On Error Resume Next Application.OnTime earliesttime:=gdNextTime, _ procedure:=gsMacro, schedule:=False End Sub ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call StopClock End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung im Sekundenabstand, ob eine Datei geändert wurde

Problem: Im Sekundenabstand soll geprüft werden, ob zu einer Excel-Arbeitsmappe Änderungen durchgeführt wurden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Public NextTime As Date
Public datUpDate As Date

Sub Updateclock()
    Dim datUpDateNew As Date
    NextTime = Now + TimeValue("00:00:01")
    Application.OnTime NextTime, "Updateclock"
    datUpDateNew = VBA.FileDateTime(Range("B1").Value)
    If datUpDateNew > datUpDate Then
        datUpDate = datUpDateNew
        MsgBox "Dateiänderung um " & datUpDate
    End If
End Sub

Sub StopClock()
    Application.OnTime NextTime, "Updateclock", , False
End Sub
```

Tabelle blinken lassen

Problem:	Auf Schaltflächendruck soll eine Tabelle zum Blinken gebracht und auf weiteren Schaltflächendruck soll das Blinken beendet werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	StandardModule: Modul1 <pre>Sub StartCellFlasher() Dim FlashingRangeCol As New Collection FlashingRangeCol.Add Range("A1") FlashingRangeCol.Add Range("A3") FlashingRangeCol.Add Range("B10") FlashingRangeCol.Add Range("C14") FlashingRangeCol.Add Range("C15:D36") FlashingRangeCol.Add Range("E5:E125") Call fncStartCellFlashing(FlashingRangeCol, 400) End Sub Sub StopCellFlasher() Call fncStopCellFlashing End Sub</pre> StandardModule: modCellFlasher <pre>Private Declare Function FindWindow _ Lib "user32" _ Alias "FindWindowA" _ (_ ByVal lpClassName As String, _ ByVal lpWindowName As String _) _ As Long Private Declare Function SetTimer _ Lib "user32" _ (_ ByVal hWnd As Long, _ ByVal nIDEvent As Long, _ ByVal uElapse As Long, _ ByVal lpTimerFunc As Long _) _ As Long Private Declare Function KillTimer _ Lib "user32" _ (_ ByVal hWnd As Long, _ ByVal nIDEvent As Long _) _ As Long Private Declare Function GetCurrentVbaProject _ Lib "vba332.dll" _ Alias "EbGetExecutingProj" _ (_ hProject As Long _) _ As Long Private Declare Function GetFuncID _ Lib "vba332.dll" _ Alias "TipGetFunctionId" _ (_ ByVal hProject As Long, _ ByVal strFunctionName As String, _ ByRef strFunctionID As String _) _ As Long Private Declare Function GetAddr _ Lib "vba332.dll" _ Alias "TipGetLpfnOfFunctionId" _ (_ ByVal hProject As Long, _ ByVal strFunctionID As String, _ ByRef lpfn As Long _) _ As Long Private WindowsTimer As Long Private FlashingRangeCollection As Collection Public Function fncStartCellFlashing _ (_ FlashingRangeCol As Collection, _ FlashingPeriod As Integer _) _ fncStopWindowsTimer Set FlashingRangeCollection = New Collection Set FlashingRangeCollection = FlashingRangeCol fncWindowsTimer CLng(FlashingPeriod) End Function Public Function fncStopCellFlashing() Dim aCell As Variant On Error Resume Next fncStopWindowsTimer For Each aCell In FlashingRangeCollection aCell.Font.Color = vbBlack Next aCell Set FlashingRangeCollection = Nothing End Function Private Function fncWindowsTimer(_ TimeInterval As Long _) As Boolean Dim WindowsTimer As Long WindowsTimer = 0 If Val(Application.Version) > 8 Then WindowsTimer = SetTimer _ (_ hWnd:=FindWindow("XLMAIN", Application.Caption), _ nIDEvent:=0, _ uElapse:=TimeInterval, _ lpTimerFunc:=AddrOf_cbkCustomTimer _) _ Else WindowsTimer = SetTimer _ (_ hWnd:=FindWindow("XLMAIN", Application.Caption), _ nIDEvent:=0, _ uElapse:=TimeInterval, _ lpTimerFunc:=AddrOf("cbkCustomTimer") _) _ End If fncWindowsTimer = CBool(WindowsTimer) End Function Private Function fncStopWindowsTimer() KillTimer _ hWnd:=FindWindow("XLMAIN", Application.Caption), _ nIDEvent:=WindowsTimer End Function Private Function cbkCustomTimer _ (_ ByVal Window_hWnd As Long, _ ByVal WindowsMessage As Long, _ ByVal EventID As Long, _ ByVal SystemTime As Long _) _ As Long Dim aCell As Variant Static aFlag As Boolean On Error Resume Next If aFlag = True Then For Each aCell In FlashingRangeCollection aCell.Font.Color = vbBlack Next aCell aFlag = False Else For Each aCell In FlashingRangeCollection aCell.Font.Color = aCell.Interior.Color Next aCell aFlag = True End If End Function Private Function AddrOf _ (_ CallbackFunctionName As String _) _ As Long , Dim aResult As Long, CurrentVBProject As Long, strFunctionID As String, _ AddressOfFunction As Long, UniCbKFunctionName As String , 'convert the name of the function to Unicode system UniCbKFunctionName = StrConv(CallbackFunctionName, vbUnicode) , 'if the current VBProjects exists ... If Not GetCurrentVbaProject(CurrentVBProject) = 0 Then '... get the function ID of the callback function based on its name, _ in order to ensure that the function exists aResult = GetFuncID _ (_ hProject:=CurrentVBProject, _ strFunctionName:=UniCbKFunctionName, _ strFunctionID:=strFunctionID _) _ 'if the function exists ... If aResult = 0 Then '... get a pointer to the callback function based on strFunctionID aResult = GetAddr _ (_ CurrentVBProject, _ strFunctionID, _ lpfn:=AddressOfFunction _) _ 'if we have got the pointer pass it to the result of the function If aResult = 0 Then AddressOfFunction End If End If End If End Function Function AddrOf_cbkCustomTimer() As Long AddrOf_cbkCustomTimer = vbaPass(AddressOf cbkCustomTimer) End Function Function vbaPass(AddressOfFunction As Long) As Long vbaPass = AddressOfFunction End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensatz alle 10 Sekunden gruppenweise eintragen.

Problem:	Die Daten aus Zelle A1 sollen alle 10 Sekunden in den Spalten B:M gruppenweise untereinander eingetragen werden. Ein Gruppe besteht aus 6 Datenpaaren.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Call StopEintragen End Sub StandardModule: basMain Public Const giIntervall As Integer = 10 Public Const gsMacro As String = "WerteEintragen" Public gdNextTime As Double Sub StartEintragen() gdNextTime = Now + TimeSerial(0, 0, giIntervall) Application.OnTime earliesttime:=gdNextTime, _ procedure:=gsMacro, schedule:=True End Sub Private Sub WerteEintragen() Dim iRow As Integer, intCol As Integer iRow = RealLastCell(ActiveSheet).Row If WorksheetFunction.CountA(_ Range(Cells(iRow, 2), Cells(iRow, 13))) = 12 Then iRow = iRow + 1 intCol = 2 Else intCol = WorksheetFunction.CountA(_ Range(Cells(iRow, 2), Cells(iRow, 13))) + 2 End If Cells(iRow, intCol) = Now Cells(iRow, intCol + 1) = Range("A1").Value Call StartEintragen End Sub Sub StopEintragen() On Error Resume Next Application.OnTime earliesttime:=gdNextTime, _ procedure:=gsMacro, schedule:=False End Sub Private Function RealLastCell(TheSheet As Worksheet) As Range Dim ExcelLastCell As Range Dim Row%, Col%, LastRowWithData%, LastColWithData% Set ExcelLastCell = TheSheet.Cells.SpecialCells(xlLastCell) LastRowWithData = ExcelLastCell.Row Row = ExcelLastCell.Row Do While Application.CountA(TheSheet.Rows(Row)) = 0 And Row <> 1 Row = Row - 1 Loop LastRowWithData = Row LastColWithData = ExcelLastCell.Column Col = ExcelLastCell.Column Do While Application.CountA(_ TheSheet.Columns(Col)) = 0 And Col <> 1 Col = Col - 1 Loop LastColWithData = Col Set RealLastCell = TheSheet.Cells(Row, Col) End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Meldung nach vorgegebener Minutenzahl

Problem:	Nach einer in einer UserForm-TextBox einzugebenden Minutenzahl soll eine Meldung erfolgen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>StandardModule: basMain Sub DialogAufruf() frmMessage.Show End Sub Sub Meldung() MsgBox "Ich sollte mich melden!" End Sub ClassModule: frmMessage Private Sub cmdOK_Click() Application.OnTime Now + _ TimeSerial(0, CInt(txtMinute.Text), 0), "Meldung" Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe 5 Sekunden nach dem Öffnen speichern und schließen

Problem: Wie kann ich eine Arbeitsmappe 5 Sekunden, nachdem sie geöffnet wurde, automatisch speichern und schließen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Application.OnTime Now + _
        TimeSerial(0, 0, 5), "AutomatischSpeichern"
End Sub

StandardModule: basMain

Sub AutomatischSpeichern()
    ThisWorkbook.Close savechanges:=True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Rechner zu einem vorgegebenen Zeitpunkt ausschalten

Problem:	Wie kann ich zu einem bestimmten Zeitpunkt alle Programme beenden und den Rechner ausschalten lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Declare Function ExitWindowsEx Lib "user32" _ (ByVal uFlags&, ByVal wReserved&) Global Const EWX_FORCE = 8 Global Const EWX_LOGOFF = 0 Global Const EWX_REBOOT = 2 Global Const EWX_SHUTDOWN = 1 Sub TschuessEinleiten() Dim varTschuess As Variant varTschuess = InputBox("Wann soll der Rechner ausgeschaltet werden?", , _ Format(Now + TimeSerial(0, 1, 0), "hh:mm:ss")) If varTschuess = "" Then Exit Sub Application.OnTime TimeValue(varTschuess), "Tschuess" End Sub Sub Tschuess() Dim LResult LResult = ExitWindowsEx(EWX_SHUTDOWN, 0&) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktuell verbrachte Arbeitszeit anzeigen

Problem: Wie kann ich mir in einer Zelle die aktuell verbrachte Tagesarbeitszeit anzeigen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call StopClock
End Sub

Private Sub Workbook_Open()
    Call StartClock
End Sub

StandardModule: basMain

Public Const gsMacro As String = "UpdateClock"
Public gdNextTime As Double

Sub PauseStart()
    Dim iRow As Integer
    iRow = Cells(Rows.Count, 2).End(xlUp).Row + 1
    Cells(iRow, 2) = Now()
End Sub

Sub PauseEnde()
    Dim iRow As Integer
    iRow = Cells(Rows.Count, 3).End(xlUp).Row + 1
    Cells(iRow, 3) = Now()
End Sub

Sub StartClock()
    gdNextTime = Now + TimeSerial(0, 1, 0)
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=True
End Sub

Sub UpdateClock()
    ThisWorkbook.Worksheets("Arbeitszeit").Range("D3").Calculate
    Call StartClock
End Sub

Sub StopClock()
    On Error Resume Next
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=False
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeit eintragen und Tabelle minütlich als HTML-Datei speichern

Problem:	Wie kann ich die Zeitangabe in einer Zelle und der Statusbar jede Sekunde aktualisieren und die Tabelle jede Minute als HTMLDatei speichern lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Public Const gsMacroA As String = "UpdateClock" Public Const gsMacroB As String = "UpdateHTML" Public gdNextTimeA As Double Public gdNextTimeB As Double Sub StartClock() gdNextTimeA = Now + TimeSerial(0, 0, 1) Application.OnTime earliesttime:=gdNextTimeA, _ procedure:=gsMacroA, schedule:=True End Sub Private Sub UpdateClock() Application.DisplayStatusBar = True ThisWorkbook.Worksheets("Uhrzeit").Range("A1").Calculate Application.StatusBar = "Zeit: " & Format(Time, "hh:mm:ss") Call StartClock End Sub Sub StopClock() On Error Resume Next Application.OnTime earliesttime:=gdNextTimeA, _ procedure:=gsMacroA, schedule:=False Application.StatusBar = False End Sub Sub StartHTML() Dim iIntervall As Integer iIntervall = ThisWorkbook.Worksheets("Uhrzeit").Range("D1").Value gdNextTimeB = Now + TimeSerial(0, iIntervall, 0) Application.OnTime earliesttime:=gdNextTimeB, _ procedure:=gsMacroB, schedule:=True End Sub Private Sub UpdateHTML() On Error GoTo ERRORHANDLER Close Open ThisWorkbook.Worksheets("Uhrzeit").Range("D2").Value & _ "\zeit.htm" For Output As #1 Print #1, "<html><body><H1>" Print #1, "Diese Datei wurde um " & Range("A1").Text & " Uhr gespeichert." Print #1, "</H1></body></html>" Close Call StartHTML Exit Sub ERRORHANDLER: MsgBox "Die Datei konnte nicht erstellt werden!" Call StopHTML End Sub Sub StopHTML() On Error Resume Next Application.OnTime earliesttime:=gdNextTimeB, _ procedure:=gsMacroB, schedule:=False End Sub ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call StopClock Call StopHTML End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Inhalt einer Zelle in Minutenabstand in einer Spalte auflisten

Problem: Zelle A1 enthält eine DDE-Verknüpfung zu einem externen Programm, die sich in Minutenabstand aktualisiert. Wie kann ich die aktualisierten Daten minütlich in Spalte C zur späteren Weiterverwendung auflisten lassen? In Spalte B sollen das jeweils aktuelle Datum und Zeit eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Public Const gsMacro As String = "UpdateClock"
Public gdNextTime As Double

Sub StartClock()
    Dim iIntervall As Integer
    iIntervall = Range("E1").Value
    gdNextTime = Now + TimeSerial(0, 0, iIntervall)
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=True
End Sub

Sub UpdateClock()
    Dim wks As Worksheet
    Dim iRow As Integer
    Set wks = ThisWorkbook.Worksheets("Tabelle1")
    iRow = wks.Cells(Rows.Count, 2).End(xlUp).Row + 1
    Cells(iRow, 2) = Now
    Cells(iRow, 3) = Range("A1").Value
    Call StartClock
End Sub

Sub StopClock()
    On Error Resume Next
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=False
End Sub

ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call StopClock
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fortlaufende Uhrzeit in Zelle und Statusleiste zeigen

Problem: Wie kann ich in einer Zelle und in der Statusleiste die jeweils aktuelle Uhrzeit anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Public Const gsMacro As String = "UpdateClock"
Public gdNextTime As Double

Sub UpdateClock()
    Application.DisplayStatusBar = True
    Worksheets("Tabelle1").Range("A1").Calculate
    Application.StatusBar = "Zeit: " & Format(Time, "hh:mm:ss")
    Call StartClock
End Sub

Sub StartClock()
    gdNextTime = Now + TimeSerial(0, 0, 1)
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=True
End Sub

Sub StopClock()
    On Error Resume Next
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=False
    Application.StatusBar = False
End Sub

ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call StopClock
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelle blinken lassen

Problem: Wie kann ich die Hintergrundfarbe einer Zelle jede Sekunde wechseln?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call BlinkenEnde
End Sub

StandardModule: basMain

Public Const giIntervall As Integer = 1
Public Const gsMacro As String = "Blinken"
Public gdNextTime As Double

Sub BlinkenEin()
    gdNextTime = Now + TimeSerial(0, 0, giIntervall)
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=True
End Sub

Private Sub Blinken()
    With ActiveCell
        If .Interior.ColorIndex = 6 Then
            .Interior.ColorIndex = 9
            .Font.ColorIndex = 6
        Else
            .Interior.ColorIndex = 6
            .Font.ColorIndex = 9
        End If
    End With
    Call BlinkenEin
End Sub

Sub BlinkenEnde()
    On Error Resume Next
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=False
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe alle 5 Minuten neu berechnen

Problem: Wie kann ich eine Arbeitsmappe alle 5 Minuten neu berechnen lassen?

Lösung: Die nachfolgenden Makros in die genannten Module einfügen.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    Call BerechnenStop
    Application.Calculation = gvar
End Sub

Private Sub Workbook_Open()
    gvar = Application.Calculation
End Sub

StandardModule: basMain

Public Const ciIntervall As Integer = 10
Public Const dsMacro As String = "Berechnen"
Public gdNextTime As Double
Public gvar As Variant

Sub BerechnenStart()
    Application.Calculation = xlManual
    gdNextTime = Now + TimeSerial(0, 0, ciIntervall)
    Application.OnTime gdNextTime, dsMacro
End Sub

Sub Berechnen()
    Application.Calculate
    Call BerechnenStart
End Sub

Sub BerechnenStop()
    On Error Resume Next
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=dsMacro, schedule:=False
    Application.Calculation = gvar
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makros zeitgesteuert aufrufen

Problem: Wie kann ich in Abhängigkeit vom Zeitpunkt des Öffnens einer Arbeitsmappe Makros aufrufen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_Open()  
    Select Case Hour(Time)  
        Case Is < 9  
            Call Routine9h  
        Case Is < 15  
            Call Routine15h  
        Case Is < 20  
            Call Routine20h  
    End Select  
End Sub  
  
StandardModule: basMain  
  
Sub Routine9h()  
    MsgBox "Es ist vor 09:00h"  
End Sub  
  
Sub Routine15h()  
    MsgBox "Es ist vor 15:00h"  
End Sub  
  
Sub Routine20h()  
    MsgBox "Es ist vor 20:00h"  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen der Arbeitsmappe auf aktuelles Datum prüfen

Problem:	Beim Öffnen der Arbeitsmappe wird überprüft, ob das aktuelle Datum in einem der Arbeitsblätter vorhanden ist.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim wks As Worksheet Dim rng As Range For Each wks In Worksheets Set rng = wks.Cells.Find(Date, LookIn:=xlValues, lookat:=xlWhole) If Not rng Is Nothing Then MsgBox "Aktuelles Datum ist in der Arbeitsmappe enthalten!" Exit For End If Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spaltenweiser Blattschutz in Abhängigkeit des Anwendernamens

Problem: Der Blattschutz soll in Abhängigkeit des Anwendernamens (Menü EXTRAS / OPTIONEN / Register ALLGEMEIN / ANWENDERNAME) spaltenweise gesetzt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_Open()  
    Dim rng As Range  
    Worksheets("Tabelle1").Unprotect "HWH"  
    Set rng = Range("A1").CurrentRegion  
    rng.Locked = True  
    Select Case Application.UserName  
        Case "Anwender1": rng.Columns(1).Locked = False  
        Case "Anwender2": rng.Columns(2).Locked = False  
        Case "Hans W. Herber": rng.Columns(3).Locked = False  
        Case "Anwender4": rng.Columns(4).Locked = False  
        Case "Anwender5": rng.Columns(5).Locked = False  
    End Select  
    Worksheets("Tabelle1").Protect "HWH"  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe ausgeblendet öffnen

Problem:	Die Arbeitsmappe soll beim Öffnen automatisch ausgeblendet werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Windows(ThisWorkbook.Name).Visible = False MsgBox "Sie können die Arbeitsmappe einblenden über:" & _ vbCrLf & "Menü FENSTER / EINBLENDEN" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Funktionstaste F11 beim Öffnen ein- und beim Schließen aus

Problem: Beim Öffnen der Arbeitsmappe wird die Funktion der Taste F11 aus- und beim Schließen wieder eingeschaltet.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Application.OnKey "{F11}"
End Sub

Private Sub Workbook_Open()
    Application.OnKey "{F11}", ""
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen der Arbeitsmappe ComboBox-Wert setzen

Problem:	Beim Öffnen der Arbeitsmappe soll der Monat des Dropdownfelds auf den Monat nach dem aktuellen Datum und das Jahr auf das Jahr des kommenden Monats eingestellt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim iMonth As Integer, iYear As Integer If Month(Date) = 12 Then iMonth = 1 iYear = Year(Date) + 1 Else iMonth = Month(Date) + 1 iYear = Year(Date) End If With Worksheets("Tabelle1") .Range("B1").Value = iMonth .Range("B3").Value = Year(Date) End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorlagendialog beim Öffnen einer Arbeitsmappe aufrufen

Problem: Beim Öffnen von Excel soll automatisch der Dialog zur Auswahl der Vorlagen aufgerufen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Dim oBtn As CommandBarButton
    Set oBtn = Application.CommandBars.FindControl(ID:=18)
    oBtn.Execute
    ThisWorkbook.Close savechanges:=False
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Berechnungsmodus ändern und beim Schließen zurücksetzen.

Problem:	Der eingestellte Berechnungsmodus wird beim Öffnen der Arbeitsmappe gespeichert, auf Manuell gestellt und bei Schließen der Arbeitsmappe wieder zurückgesetzt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Application.Calculation = Worksheets("Data").Range("A1").Value End Sub Private Sub Workbook_Open() Worksheets("Data").Range("A1").Value = Application.Calculation Application.Calculation = xlCalculationManual End Sub StandardModule: Modul1 Sub AusEinblenden() Worksheets("Data").Visible = xlVeryHidden Worksheets("Data").Visible = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen der Arbeitsmappe Bestellnummer eintragen

Problem:	Beim Öffnen der Arbeitsmappe soll in Zelle B1 die nächste Bestellnummer eingetragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim vNo As Variant Dim iCounter As Integer Dim sPath As String On Error GoTo ERRORHANDLER sPath = "c:\temp" With Application.FileSearch .Filename = "*.xls" .SearchSubFolders = False .LookIn = sPath .Execute msoSortByFileName End With If .FoundFiles.Count > 0 Then vNo = Dir(.FoundFiles(.FoundFiles.Count)) Else vNo = 0 End If With Range("B1") .NumberFormat = "@" .Value = Format(CInt(Left(vNo, 4)) + 1, "0000") End With Exit Sub ERRORHANDLER: MsgBox "Sie müssen das Verzeichnis anpassen!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makroaufruf mit F6

Problem: In der Workbook_Open-Prozedur soll die Funktionstaste 6 für einen Makroaufruf festgelegt werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Application.OnKey "{F6}"
End Sub

Private Sub Workbook_Open()
    Application.OnKey "{F6}", "CopyAbove"
End Sub

StandardModule: Modul1

Sub CopyAbove()
    Dim wks As Worksheet
    Dim rng As Range
    On Error Resume Next
    Set wks = ActiveSheet
    If Err > 0 Or wks Is Nothing Then Exit Sub
    Application.ScreenUpdating = False
    If Selection.Row = 1 Then Exit Sub
    If Selection.Rows.Count > 1 Then Exit Sub
    For Each rng In Selection.Cells
        rng.Offset(-1, 0).Copy rng
    Next rng
    Application.CutCopyMode = False
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei jedem Öffnen der Arbeitsmappe Wert um 1 erhöhen

Problem: Bei jedem Öffnen der Arbeitsmappe soll die Rechnungsnummer um 1 erhöht werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_Open()  
    With Range("RechnNr")  
        .Value = .Value + 1  
    End With  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe öffnen und nach Beenden des Mausklicks schließen

Problem:	Bei Mausklick auf die Schaltfläche soll eine Datei geöffnet, über den Maushaltezeitraum angezeigt und nach Beenden des Haltens wieder geschlossen werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdLoadFile_MouseDown(_ ByVal Button As Integer, _ ByVal Shift As Integer, _ ByVal X As Single, _ ByVal Y As Single) Dim sFile As String sFile = Range("B1").Value If Dir(sFile) = "" Then Beep MsgBox "Datei wurde nicht gefunden!" Exit Sub End If Workbooks.Open sFile, False, True End Sub Private Sub cmdLoadFile_MouseUp(_ ByVal Button As Integer, _ ByVal Shift As Integer, _ ByVal X As Single, _ ByVal Y As Single) ActiveWorkbook.Close savechanges:=False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Meldung bei Datumsüberschreitung

Problem: Wenn die Kalenderwochen-Differenz zwischen dem Wert in Zelle A1 und dem aktuellen Datum in Zelle B1 größer als 2 ist, soll beim Öffnen der Arbeitsmappe eine Meldung ausgegeben werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    With Worksheets("Tabelle1")
        If .Range("D1").Value - .Range("C1").Value > 2 Then
            MsgBox "Zwei Kalenderwochen wurden überschritten!"
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur Eintragungen in aktuellem Monat zulassen

Problem:	In Spalte B dürfen immer nur die Zellen editierbar sein, deren Nachbar in Spalte A ein Datum des aktuellen Monats trägt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim wks As Worksheet Dim iRowL As Integer, iRow As Integer Set wks = Worksheets("Tabelle1") wks.Unprotect iRowL = wks.Cells(Rows.Count, 1).End(xlUp).Row For iRow = 1 To iRowL If Month(wks.Cells(iRow, 1)) = Month(Date) Then wks.Cells(iRow, 2).Locked = False Else wks.Cells(iRow, 2).Locked = True End If Next iRow ActiveSheet.Protect End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen einer Arbeitsmappe erstes Blatt auswählen

Problem: Beim Öffnen einer Arbeitsmappe soll automatisch das erste Blatt selektiert werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_Open()  
    Worksheets(1).Activate  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit F6 Werte von einer zur anderen Zelle transferieren

Problem:	Wie kann ich mit dem Drücken der Funktionstaste F6 den Wert von Zelle A1 in Zelle B1 übertragen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Application.OnKey "{F6}" End Sub Private Sub Workbook_Open() Application.OnKey "{F6}", "ChangeValue" End Sub StandardModule: basMain Sub ChangeValue() Range("B1").Value = Range("A1").Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten in Abhängigkeit zum aktuellen Datum löschen

Problem: Wie kann ich beim Öffnen der Arbeitsmappe in einer Tabelle alle Daten löschen, die vor dem aktuellen Datum liegen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_Open()  
    Dim iCol As Integer  
    For iCol = 2 To 17  
        With Worksheets("Tabelle1")  
            If .Cells(3, iCol).Value < Date - 5 Then  
                .Range(.Cells(4, iCol), .Cells( _  
                    Rows.Count, iCol)).ClearContents  
            End If  
        End With  
    Next iCol  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Terminmeldung beim Öffnen der Arbeitsmappe

Problem: Wie kann ich veranlassen, daß beim Öffnen einer Arbeitsmappe eine Terminliste überprüft wird und bei Übereinstimmung mit dem aktuellen Tag eine Meldung erfolgt?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Dim wks As Worksheet
    Dim var As Variant
    Set wks = Worksheets("Tabelle1")
    var = Application.Match(CDbl(Date), wks.Columns(1), 0)
    If Not IsError(var) Then
        MsgBox "Heutiger Termin:" & vbCrLf & _
            wks.Cells(var, 2).Value
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Öffnen Standardschriftgröße prüfen und Zoom festlegen

Problem:	Wie kann ich beim Öffnen einer Arbeitsmappe die Standardschriftgröße prüfen und in Abhängigkeit hiervon den Zoomfaktor festlegen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Select Case Application.StandardFontSize Case 8, 9: ActiveWindow.Zoom = 120 Case 10, 11: ActiveWindow.Zoom = 100 Case 12, 14: ActiveWindow.Zoom = 80 End Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen der Arbeitsmappe Passwort abfragen

Problem:	Wie kann ich beim Öffnen einer Arbeitsmappe ein Passwort abfragen und - wenn korrekt - den Blattschutz aller Arbeitsblätter aufheben?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean) Dim wks As Worksheet For Each wks In ThisWorkbook.Worksheets wks.Protect "MeinPasswort" Next wks ThisWorkbook.Save End Sub Private Sub Workbook_Open() Dim wks As Worksheet Dim strPassword strPassword = InputBox("Bitte Passwort eingeben:", , "MeinPasswort") If strPassword <> "MeinPasswort" Then Exit Sub For Each wks In ThisWorkbook.Worksheets wks.Unprotect "MeinPasswort" Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen zweites Fenster erzeugen und anordnen

Problem: Wie kann ich beim Öffnen einer Arbeitsmappe ein zweites Fenster zur 2. Tabelle der Arbeitsmappe öffnen, beiden Fenster in einer bestimmten Größe anzeigen lassen und beim Schließen der Arbeitsmappe den alten Zustand wiederherstellen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Public gvarWndState

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    Windows(Windows.Count).Close
    ActiveWindow.WindowState = gvarWndState
    On Error GoTo 0
End Sub

Private Sub Workbook_Open()
    Dim wndFirst As Window
    gvarWndState = ActiveWindow.WindowState
    Set wndFirst = ActiveWindow
    With ActiveWindow
        .WindowState = xlNormal
        .Top = 50
        .Left = 50
        .Width = 200
        .Height = 200
    End With
    ActiveWindow.NewWindow
    With ActiveWindow
        .WindowState = xlNormal
        .Top = 50
        .Left = 260
        .Width = 200
        .Height = 200
    End With
    Worksheets("Tabelle2").Select
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen der Arbeitsmappe eine UserForm anzeigen

Problem:	Wie kann ich beim Öffnen einer Arbeitsmappe eine UserForm mit einer OK-Schaltfläche anzeigen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() frmStart.Show End Sub ClassModule: frmStart Private Sub cmdWeiter_Click() Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Tabellenaktivierung im Extras-Menü eigenen Menüpunkt anzeigen

Problem: Wie kann ich im Extras-Menü bei der Anzeige dieser Arbeitsmappe einen eigenen Menüpunkt anzeigen und beim Fensterwechsel wieder ausblenden lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_WindowActivate(ByVal Wn As Excel.Window)
    Dim oBar As CommandBar
    Dim oPopUp As CommandBarPopup
    Dim oBtn As CommandBarButton
    Set oBar = Application.CommandBars("Worksheet Menu Bar")
    Set oPopUp = oBar.FindControl(ID:=30007)
    Set oBtn = oPopUp.Controls.Add
    With oBtn
        .Caption = "Meldung"
        .OnAction = "mnuMeldung"
        .Style = msoButtonCaption
    End With
End Sub

Private Sub Workbook_WindowDeactivate(ByVal Wn As Excel.Window)
    Dim oBar As CommandBar
    Dim oPopUp As CommandBarPopup
    On Error Resume Next
    Set oBar = Application.CommandBars("Worksheet Menu Bar")
    Set oPopUp = oBar.FindControl(ID:=30007)
    oPopUp.Controls("Meldung").Delete
    On Error GoTo 0
End Sub

StandardModule: basMain

Sub mnuMeldung()
    MsgBox "Ich bin die Meldung!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Start-UserForm zeitgesteuert anzeigen

Problem:	Wie kann ich beim Öffnen einer Arbeitsmappe eine UserForm für eine vorgegebene Zeit anzeigen lassen und die Arbeitsmappe nach der Anzeige wieder schließen?
Lösung:	Geben Sie den Ereigniscode in die genannten Klassenmodule ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() frmHWH.Show End Sub StandardModule: modMain Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long) Sub Warten() Dim i% For i = 1 To 10 frmHWH.Label1.Caption = i frmHWH.Repaint Sleep 1000 Next i End Sub ClassModule: frmHWH Private Sub UserForm_Activate() frmHWH.Repaint Call Warten Unload Me ThisWorkbook.Close savechanges:=False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Workbook_Open-Prozedur beim VBA-Aufruf ausschalten

Problem:	Wie kann ich beim Öffnen einer Arbeitsmappe per VBA die Workbook_Open-Routine ausschalten?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub OpenWkb() On Error GoTo ERRORHANDLER Application.EnableEvents = False Workbooks.Open FileName:=Range("B1").Value ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datum in Registry eintragen und auslesen

Problem: Wie kann ich beim ersten Öffnen einer Arbeitsmappe in die Registry ein Datum eintragen lassen? Dieses Datum soll bei jedem weiteren Öffnen dieser Arbeitsmappe abgefragt werden. Ist es älter als 35 Tage, soll die Arbeitsmappe geschlossen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Dim sAbfrage As String
    sAbfrage = GetSetting( _
        appname:="MeinProg", _
        section:="Valid", _
        key:="Exit")
    If sAbfrage = "" Then
        SaveSetting _
            appname:="MeinProg", _
            section:="Valid", _
            key:="Exit", _
            Setting:=Date
        MsgBox "Heutiges Datum wurde eingetragen!"
    Else
        If CDate(sAbfrage) < Date - 35 Then
            ThisWorkbook.Close savechanges:=False
        Else
            MsgBox "Diese Arbeitsmappe wurde am" & vbCrLf & _
                sAbfrage & " das erste mal geöffnet!"
        End If
    End If
End Sub

StandardModule: basMain

Sub Loeschen()
    On Error Resume Next
    DeleteSetting "MeinProg"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen der Arbeitsmappe bedingt Datum eintragen

Problem:	Wie kann ich veranlassen, daß beim Öffnen einer Arbeitsmappe geprüft wird, ob die Zellen A1 und B1 Werte enthalten? Wenn A1 nicht leer und B1 leer ist, soll in B1 das aktuelle Datum eingetragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() With Worksheets("Tabelle1") If Not IsEmpty(.Range("A1")) And _ IsEmpty(.Range("B1")) Then .Range("B1") = Date End If End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen einer Arbeitsmappe WAV-Datei abspielen

Problem: Wie kann ich beim Start einer Arbeitsmappe automatisch eine WAV-Datei abspielen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Call WAvalsStart
End Sub

StandardModule: basMain

Declare Function sndPlaySound32 Lib "winmm.dll" _
    Alias "sndPlaySoundA" (ByVal lpszSoundName _
    As String, ByVal uFlags As Long) As Long

Sub WAvalsStart()
    On Error GoTo ERRORHANDLER
    Call sndPlaySound32("c:\windows\drumroll.wav", 1)
End Sub

ERRORHANDLER:
    MsgBox "Bitte Pfad und Namen der WAV-Datei anpassen!"
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Geburtstagsmeldung bei Öffnen der Arbeitsmappe

Problem:	Wie kann ich mir jedesmal beim Öffnen einer bestimmten Mappe anzeigen lassen, wer Geburtstag hat?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim dat As Date Dim iRow As Integer Dim sTxt As String iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) dat = Cells(iRow, 4).Value If Month(dat) = Month(Date) And _ Day(dat) = Day(Date) Then sTxt = sTxt & Cells(iRow, 2).Value & " " & _ Cells(iRow, 1).Value & " wird " & _ Cells(iRow, 3).Value & vbCrLf End If iRow = iRow + 1 Loop If sTxt <> "" Then MsgBox "Geburtstage:" & vbCrLf & sTxt Else MsgBox "Es liegt kein Geburtstag an!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Symbolleisten beim Öffnen aus-, beim Schließen einblenden

Problem: Wie kann ich alle Symbolleisten beim Öffnen der Arbeitsmappe ausund beim Schließen wieder einblenden lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Dim oBar As CommandBar
    Dim iRow As Integer
    For Each oBar In Application.CommandBars
        If oBar.Visible And oBar.Type <> _
            msoBarTypeMenuBar Then
            iRow = iRow + 1
            With ThisWorkbook.Worksheets("CmdBars")
                .Cells(iRow, 1).Value = oBar.Name
                oBar.Visible = False
            End With
        End If
    Next oBar
End Sub

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call CmdBarsEin
End Sub

StandardModule: basMain

Sub WksAus()
    Worksheets("CmdBars").Visible = xlVeryHidden
End Sub

Sub CmdBarsEin()
    Dim iRow As Integer
    iRow = 1
    With ThisWorkbook.Worksheets("CmdBars")
        Do Until IsEmpty(.Cells(iRow, 1))
            Application.CommandBars(.Cells(iRow, 1) _
                .Value).Visible = True
            iRow = iRow + 1
        Loop
        .Columns(1).Clear
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In Funktionskategorie einordnen

Problem: Wie kann ich eine benutzerdefinierte Funktion in die Funktionskategorie "Finanzmathematik" einordnen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Sub Workbook_Open()
    Application.MacroOptions _
        macro:="DM2EURO", _
        Description:="Euro-Umrechnung", _
        Category:=1
End Sub

StandardModule: basMain

Function DM2EURO(Wert)
    DM2EURO = WorksheetFunction.Round(Wert / 1.95583, 2)
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zur Zelle mit aktuellem Datum scrollen

Problem:	In Spalte A habe ich Datumseinträge. Wie kann ich veranlassen, daß beim Öffnen der Arbeitsmappe die Zeile, in der sich die Zelle mit dem aktuellen Datum befindet, als oberste Zeile angezeigt wird?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim iRow As Integer With Worksheets("Tabelle1") iRow = WorksheetFunction.Match(CDbl(Date), .Columns(1), 0) Application.Goto .Cells(iRow, 1), True End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Laufende Rechnungsnummer

Problem: Wie kann ich in ein Rechnungsformular die Rechnungsnummer aktualisieren lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_Open()  
    With Worksheets("Tabelle1").Range("G10")  
        .Value = .Value + 1  
    End With  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Heinweis, daß 4 Wochen vergangen sind

Problem: Wie kann ich mich beim Öffnen einer Arbeitsmappe darauf hinweisen lassen, daß ab einem bestimmten Datum jeweils 4 Wochen vergangen sind?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    If Date - Worksheets("Tabelle1") _
        .Range("A1").Value >= 28 Then
        MsgBox "Schon wieder 4 wOchen älter geworden!"
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verfalldatum abgelaufen

Problem: Wie kann ich das Öffnen einer Arbeitsmappe nach einem bestimmten Datum verhindern.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    If Date > DateSerial(2002, 7, 31) Then
        Beep
        MsgBox "Verfalldatum abgelaufen!"
        ActiveWorkbook.Close savechanges:=False
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fehler in Workbook_Open-Prozedur abfangen

Problem: Wie kann ich verhindern, daß die Workbook_Open-Prozedur abbricht, wenn eine zu ladende Arbeitsmappe nicht gefunden wird?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_Open()  
    Dim sFile As String  
    sFile = ThisWorkbook.Path & "\test.xls"  
    If Dir(sFile) = "" Then  
        MsgBox "Kann Datei '" & sFile & "' nicht finden -" _  
            & vbCrLf & "bitte manuell laden!"  
    Else  
        Workbooks.Open sFile  
        MsgBox "Bitte Eintragung vornehmen!"  
    End If  
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Start des Makros nach Laden der Arbeitsmappe

Problem: Wie bringe ich ein Makro automatisch nach Laden der Arbeitsmappe ans Laufen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
StandardModule: basMain

Sub StatistikProgramm()
    MsgBox "Ich bin das Statistikprogramm"
End Sub

ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    If MsgBox( _
        prompt:="Statistikprogramm öffnen?", _
        Buttons:=vbQuestion + vbYesNo) = vbYes Then
        Call StatistikProgramm
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beispiele zum Einsatz des SelectionChange-Ereignisses

Problem:	In 15 Tabellenblättern werden Beispiele zum Einsatz des SelectionChange-Ereignisses gezeigt.
Lösung:	Geben Sie den Ereigniscode in die Klassenmodule der Arbeitsblätter ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim rng As Range Set rng = Range("B5:C10") If Intersect(Target.Cells(1), rng) Is Nothing Or _ Intersect(Target.Cells(Target.Cells.Count), rng) _ Is Nothing Then rng.Select End Sub ClassModule: Tabelle2 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim rng As Range Set rng = Range("B1,B3,B5,B7") If Selection.Cells.Count > 1 Then ActiveCell.Select If Intersect(ActiveCell, rng) Is Nothing Then _ rng.Cells(1).Select End Sub ClassModule: Tabelle3 Private Sub Worksheet_SelectionChange(ByVal Target As Range) If Selection.Cells.Count > 1 Then ActiveCell.Select End Sub ClassModule: Tabelle4 Private Sub Worksheet_Activate() gvCmt = Application.DisplayCommentIndicator Application.DisplayCommentIndicator = xlCommentAndIndicator End Sub Private Sub Worksheet_Deactivate() Application.DisplayCommentIndicator = gvCmt End Sub Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim cmt As Comment If Target.Cells.Count > 1 Then ActiveCell.Select For Each cmt In Comments cmt.Delete Next cmt If Target.HasFormula Then Set cmt = Target.AddComment(Target.FormulaLocal) cmt.Shape.TextFrame.AutoSize = True End If End Sub StandardModule: Modul1 Public gvCmt As Variant Public grngOld As Range Public gdblRowHeight As Double Public gintSize As Integer Sub MacroOut() MsgBox "Sie waren soeben in Zelle D6!" End Sub Sub MacroIn() MsgBox "Sie befinden sich in Zelle D6!" End Sub ClassModule: Tabelle5 Private Sub Worksheet_SelectionChange(ByVal Target As Range) With Animation1 If Target.Address = "\$E\$4" Then .Visible = True .Open ThisWorkbook.Path & "\ykohl.avi" Else .Visible = False End If End With End Sub ClassModule: Tabelle6 Private Sub Worksheet_SelectionChange(ByVal Target As Range) With TextBoxes(1) .Top = Target.Top .Left = Target.Left .Text = Target.Address(False, False) End With End Sub ClassModule: Tabelle7 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Cells.Interior.ColorIndex = xlColorIndexNone Target.Interior.ColorIndex = 6 End Sub ClassModule: Tabelle8 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Cells(WorksheetFunction.CountA(Columns(1)) + 1, 1).Value = _ Target.Address(False, False) End Sub ClassModule: Tabelle9 Dim bln As Boolean Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range) Dim rng As Range Set rng = Range("D6") If Target.Address = rng.Address Then bln = True If bln = True And Target.Address <> rng.Address Then Call MacroOut bln = False End If End Sub ClassModule: Tabelle10 Private Sub Worksheet_SelectionChange(ByVal Target As Range) If Target.Address = "\$D\$6" Then Call MacroIn End Sub ClassModule: Tabelle11 Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range) Application.EnableEvents = False Range(Target, Target.Offset(0, 9)).Select Application.EnableEvents = True End Sub ClassModule: Tabelle12 Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range) On Error Resume Next grngOld.Font.Size = gintSize Rows(grngOld.Row).RowHeight = gdblRowHeight gintSize = Target.Font.Size gdblRowHeight = Rows(Target.Row).RowHeight If IsEmpty(Target) Then Exit Sub Target.Font.Size = 20 Set grngOld = Target Rows(Target.Row).AutoFit End Sub ClassModule: Tabelle13 Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range) If Target.Column > 3 And Target.Column < 7 Then Application.Cursor = xlWait Else Application.Cursor = xlDefault End If End Sub ClassModule: Tabelle14 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim strList If Target.Column <> 1 Then Exit Sub If Range("B1").Value <> "Monate" And _ Range("B1").Value <> "Tage" Then Exit Sub Select Case Range("B1").Value Case "Monate" strList = "Monate" Case "Tage" strList = "Tage" End Select With Target.Validation .Delete .Delete .Add _ Type:=xlValidateList, _ AlertStyle:=xlValidAlertStop, _ Operator:=xlBetween, _ Formula1:="=" & strList End With End Sub ClassModule: Tabelle15 Private Sub Worksheet_SelectionChange(ByVal Target As Range) If Target.Address = "\$A\$7" Then Range("A15").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auflistung der jeweils zuletzt ausgewählten 10 Zellen

Problem: Die zuletzt ausgewählten 10 aktivierten Zellen werden aufgelistet.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Dim iRow As Integer
    With ThisWorkbook.Sheets("Memory")
        iRow = WorksheetFunction.CountA(.Columns(1)) + 1
        .Cells(iRow, 1).Value = Target.Cells(1).Address(False, False)
        If iRow >= 11 Then
            .Range("A1:A10").Value = .Range("A2:A11").Value
            .Range("A11").ClearContents
        End If
    End With
End Sub

Sub a()
    ThisWorkbook.Sheets("Memory").Columns(1).ClearContents
End Sub

StandardModule: Modul1

Sub Eintragen()
    Dim iRow As Integer
    For iRow = 1 To 10
        Cells(iRow, 1).Value = ThisWorkbook.Sheets("Memory").Cells(iRow, 1).Value
    Next iRow
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile markieren, in der sich der Cursor befindet

Problem:	Die Zeile, in der sich der Cursor befindet, soll markiert werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Rows(Target.Row).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Änderung der Schriftgröße bei Zellauswahl

Problem:	Wenn sich in Zelle I2 eine Zahl befindet, soll sich bei Auswahl dieser Zelle die Schriftgröße ändern.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) If Target.Address = "\$I\$2" Then If IsEmpty(Target) Then Target.Font.Size = 10 Exit Sub End If If IsNumeric(Target.Value) Then Target.Font.Size = 24 Else Target.Font.Size = 10 End If Else Range("I2").Font.Size = 10 End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte bei Zellbewegungen in andere Tabellen übernehmen

Problem:	Bei Zellbewegungen sollen Werte in die in der Tabelle genannten Arbeitsblätter übernommen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim wks As Worksheet Dim iRow As Integer If Target.Row = 1 Then Exit Sub If Target.Column > 2 Then Exit Sub If IsEmpty(Cells(Target.Row, 2)) Then Exit Sub On Error Resume Next Set wks = Worksheets("final data" & CStr(Cells(Target.Row, 2).Value)) If Not wks Is Nothing Then iRow = wks.Cells(Rows.Count, 1).End(xlUp).Row + 1 wks.Rows(iRow).Value = Rows(Target.Row).Value End If On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte der aktiven Zeile in einer fixen Zeile anzeigen

Problem: In Zeile 20 sollen die Werte der jeweils aktiven Zeile dargestellt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Selection.Cells.Count > 1 Then Exit Sub
    Rows(20).Value = Target.EntireRow.Value
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Zellauswahl UserForm mit Artikelnummern aufrufen

Problem: Bei Auswahl einer Zelle in Spalte A soll eine UserForm mit ComboBox-Daten aus der Data-Blatt aufgerufen werden. Die UserForm-Eintragungen sollen in die Zellen übernommen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Target.Cells.Count > 1 Then Exit Sub
    If Target.Row = 1 Then Exit Sub
    If Target.Column <> 1 Then Exit Sub
    frmArtikel.Show
End Sub

ClassModule: frmArtikel

Private Sub txtArtNo_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    Dim var As Variant
    var = ActiveCell.Offset(0, 1).Formula = "=vlookup([RC-1],data!a:c,2,0)"
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    ActiveCell.Value = cboArtikel.List(cboArtikel.ListIndex, 0)
    ActiveCell.Offset(0, 1).Value = cboArtikel.List(cboArtikel.ListIndex, 1)
    ActiveCell.Offset(0, 2).Value = txtStueck.Value
    On Error GoTo ERRORHANDLER:
    Application.EnableEvents = False
    ActiveCell.Offset(1, 0).Select
    cboArtikel.SetFocus
ERRORHANDLER:
    Application.EnableEvents = True
End Sub

Private Sub UserForm_Initialize()
    cboArtikel.List = Worksheets("Data").Range("A1").CurrentRegion.Value
    cboArtikel.ListIndex = 0
    txtStueck = ActiveCell.Offset(0, 2).Value
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Zellauswahl Wert aus anderer Zelle eintragen

Problem: Bei Auswahl einer oder mehrerer Zellen im Bereich A5:A31 soll in die Zellauswahl der Wert aus Zelle U5 eingetragen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Not Intersect(Target, Range("$A$5:$A$31")) Is Nothing Then
        Selection.Value = Range("U5").Value
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wenn Formel, dann Löschtaste deaktivieren

Problem: Wenn sich der Cursor auf einer Zelle mit Formel befindet, soll die Löschtaste deaktiviert, sonst aktiviert sein.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Deactivate()
    Application.OnKey "{del}"
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Target.HasFormula Then
        Application.OnKey "{del}", ""
    Else
        Application.OnKey "{del}"
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei der Auswahl eines doppelt vorkommenden Wertes Farbe setzen

Problem: Bei der Auswahl eines in einer Liste doppelt vorkommenden Wertes soll sich die Hintergrundfarbe ändern.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Dim rng As Range
    Set rng = Range("E6:G10")
    If Target.Cells.Count > 1 Then Exit Sub
    If Intersect(Target, rng) Is Nothing Then Exit Sub
    If WorksheetFunction.CountIf(rng, Target.Value) > 1 Then
        Target.Interior.ColorIndex = 6
        Target.Font.Bold = True
    Else
        Target.Interior.ColorIndex = xlColorIndexNone
        Target.Font.Bold = False
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Cursor-Position rechts und links markieren

Problem: Rechts und links des Cursors sollen > bzw. <-Zeichen erscheinen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Target.Cells.Count > 1 Then Exit Sub
    Cells.Replace "<", ""
    Cells.Replace ">", ""
    Select Case Target.Column
        Case 1
            Target.Offset(0, 1).Value = "<"
        Case 256
            Target.Offset(0, -1).Value = "<"
        Case Else
            Target.Offset(0, 1).Value = "<"
            Target.Offset(0, -1).Value = ">"
    End Select
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamische Gültigkeitsliste

Problem:	Bei einer Zellauswahl in Spalte A soll eine Gültigkeitsliste aus Blatt Daten ohne doppelte Einträge erstellt, bei Zellauswahl in Spalte B sollen nur die im Blatt Daten dem jeweiligen Hersteller zugeordneten Produkte angezeigt werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Activate() Dim col As New Collection Dim iRow As Integer Dim sVal As String iRow = 2 On Error Resume Next With Worksheets("Daten") Do Until IsEmpty(.Cells(iRow, 1)) col.Add .Cells(iRow, 1).Value, .Cells(iRow, 1).Value iRow = iRow + 1 Loop End With On Error GoTo 0 For iRow = 1 To col.Count If iRow = 1 Then sVal = col(iRow) Else sVal = sVal & "," & col(iRow) End If Next iRow With Columns(1).Validation .Delete .Add Type:=xlValidateList, _ AlertStyle:=xlValidAlertStop, Operator:= _ xlBetween, Formula1:=sVal .IgnoreBlank = True .InCellDropdown = True End With End Sub Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim rng As Range Dim iRow As Integer Dim sVal As String If Target.Column <> 2 Then Exit Sub If IsEmpty(Target.Offset(0, -1)) Then Exit Sub With Worksheets("Daten") .Range("A1").CurrentRegion.AutoFilter field:=1, _ Criteria1:=Target.Offset(0, -1).Value For iRow = 2 To .Range("A1").CurrentRegion.Rows.Count If .Rows(iRow).Hidden = False Then If sVal = "" Then sVal = .Range("A1").CurrentRegion.Cells(iRow, 2).Value Else sVal = sVal & "," & .Range("A1") _ .CurrentRegion.Cells(iRow, 2).Value End If End If Next iRow With Target.Validation .Delete .Add Type:=xlValidateList, _ AlertStyle:=xlValidAlertStop, Operator:= _ xlBetween, Formula1:=sVal .IgnoreBlank = True .InCellDropdown = True End With .AutoFilterMode = False End With End Sub Sub Gueltigkeit() End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kommentartexte über SVERWEIS ermitteln und anzeigen

Problem:	Bei jedem Zellwechsel im dunkelblau hinterlegten Bereich soll ein mit SVERWEIS aus dem Blatt "Kommentare" gelieferter Kommentar erscheinen.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim cmt As Comment Dim rng As Range For Each cmt In ActiveSheet.Comments cmt.Delete Next cmt If Intersect(Target, Range("A1:D12")) Is Nothing Then Exit Sub Set cmt = Target.AddComment(_ WorksheetFunction.VLookup(Target.Value, _ Worksheets("Kommentare").Columns("A:B"), 2, 0)) cmt.Shape.TextFrame.AutoSize = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auswahl von Zellen in mehreren Zeilen verhindern

Problem:	Die Auswahl von Zellen in mehreren Zeilen soll verhindert werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) If Selection.Rows.Count > 1 Then Target.Cells(1).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wenn Zelle A7 ausgewählt wird, Cursor auf A15

Problem:	Immer dann, wenn die Zelle A7 ausgewählt wird, soll der Cursor auf Zelle A15 springen.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) If Target.Address = "\$A\$7" Then Range("A15").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makrostart bei Auswahl einer Zelle in Zeile 3

Problem:	Bei Auswahl der 3. Zeile soll ein Makro gestartet werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) If Target.Row = 3 Then Beep MsgBox "Sie befinden sich in Zeile 3!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gültigkeitsliste ohne Lerrzellen

Problem:	Die Gültigkeitsliste in Zelle A1 soll die Daten aus Spalte A aus Tabelle2 ohne die Leerzeilen enthalten und sich anpassen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim intRow As Integer, intLastRow As Integer Dim txt As String If Target.Address <> "\$A\$1" Then Exit Sub With Worksheets("Tabelle2") intLastRow = .Cells(Rows.Count, 1).End(xlUp).Row For intRow = 1 To intLastRow If Not IsEmpty(.Cells(intRow, 1)) Then txt = txt & .Cells(intRow, 1) & "," End If Next intRow End With txt = Left(txt, Len(txt) - 1) With Range("A1").Validation .Delete .Add _ Type:=xlValidateList, _ AlertStyle:=xlValidAlertStop, _ Operator:=xlBetween, _ Formula1:=txt End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gültigkeitsliste nach Kriterium festlegen

Problem:	Wenn in Zelle B1 "Monate" steht, soll sich die Gültigkeitsliste aus Spalte A auf Monate, bei "Tage" auf Wochentage beschränken.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim sList If Target.Column <> 1 Then Exit Sub If Range("B1").Value <> "Monate" And _ Range("B1").Value <> "Tage" Then Exit Sub Select Case Range("B1").Value Case "Monate" sList = "Monate" Case "Tage" sList = "Tage" End Select With Target.Validation .Delete .Delete .Add _ Type:=xlValidateList, _ AlertStyle:=xlValidAlertStop, _ Operator:=xlBetween, _ Formula1:="=" & sList End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellschutz nur für Formeln

Problem: Es sollen nur die Zellen mit Formeln geschützt werden. Bei Neueingaben von Formeln soll dies erkannt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Dim rng As Range
    For Each rng In Target.Cells
        If rng.HasFormula Then
            ActiveSheet.Protect
            Exit Sub
        Else
            ActiveSheet.Unprotect
        End If
    Next rng
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neuberechnung nach Änderung der Hintergrundfarbe

Problem: Nach Änderung der Hintergrundfarbe und Zellwechsel im Bereich A1:C8 soll die Formel in Zelle D1 neu berechnet werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_SelectionChange _
    (ByVal Target As Range)
    Range("D1").Calculate
End Sub

StandardModule: Modul1

Function Farbe(rngBereich As Object, iColor As Integer)
    Dim iCounter As Integer
    Dim rngAct As Range
    Application.Volatile
    For Each rngAct In rngBereich
        If rngAct.Interior.ColorIndex = iColor Then
            iCounter = iCounter + 1
        End If
    Next rngAct
    Farbe = iCounter
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abhängiger Aufruf einer Gültigkeitsliste

Problem: Nur dann, wenn die Werte in den Zellen A1:C1 jeweils 10 betragen, soll bei Auswahl der Zellen A3, C3, A5 und C5 die Liste aus D1:D12 als Gültigkeitsliste festgelegt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Range("A1") = 10 And _
       Range("B1") = 10 And _
       Range("C1") = 10 Then
        Range("A3,C3,A5,C5").Validation.Delete
        Range("A3,C3,A5,C5").Validation.Add _
            Type:=xlValidateList, _
            AlertStyle:=xlValidAlertStop, _
            Operator:=xlBetween, _
            Formula1:="=$D$1:$D$12"
    Else
        Range("A3,C3,A5,C5").Validation.Delete
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gültigkeitsliste bedingt aufklappen lassen

Problem: Wie kann ich bei Auswahl von Zelle B3 eine Gültigkeitsliste dann aufklappen lassen, wenn Zelle A1 nicht leer ist?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If ActiveCell.Address <> "$B$3" Then Exit Sub
    If Not IsEmpty(Range("A1")) Then
        SendKeys "%{down}"
        With Target.Validation
            .Delete
            .Add Type:=xlValidateList, _
                AlertStyle:=xlValidAlertStop, Operator:= _
                xlBetween, Formula1:="=$E$1:$E$12"
            .IgnoreBlank = True
            .InCellDropdown = True
            .InputTitle = ""
            .ErrorTitle = ""
            .InputMessage = ""
            .ErrorMessage = ""
            .ShowInput = True
            .ShowError = True
        End With
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile mit Cursor farblich hervorheben

Problem: Wie kann ich die Zeile, in der sich der Cursor befindet, farblich hervorheben lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Rows.Interior.ColorIndex = xlColorIndexNone
    Rows(Target.Row).Interior.ColorIndex = 6
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Cursor in bestimmtem Zellbereich in Eieruhr verwandeln

Problem:	Wie kann ich den Cursor immer dann, wenn er sich in den Spalten D:F befindet, in eine Eieruhr verwandeln?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(_ ByVal Target As Excel.Range) If Target.Column > 3 And Target.Column < 7 Then Application.Cursor = xlWait Else Application.Cursor = xlDefault End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schriftgröße der aktiven Zelle bei Auswahl ändern

Problem: Wie kann ich die Schriftgröße bei Auswahl einer Zelle ändern und beim Verlassen wieder zurücksetzen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Public rng As Range
Public dHeight As Double
Public iSize As Integer

Private Sub Worksheet_SelectionChange( _
    ByVal Target As Excel.Range)
    On Error Resume Next
    rng.Font.Size = iSize
    Rows(rng.Row).RowHeight = dHeight
    iSize = Target.Font.Size
    dHeight = Rows(Target.Row).RowHeight
    If IsEmpty(Target) Then Exit Sub
    Target.Font.Size = 20
    Set rng = Target
    Rows(Target.Row).AutoFit
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit der aktiven Zelle jeweil 9 weitere in der Zeile markieren

Problem:	Wie kann ich jedesmal, wenn ich eine Zelle auswähle, die neun sich rechts der aktiven Zelle befindlichen Zellen mit auswählen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(_ ByVal Target As Excel.Range) On Error GoTo ERRORHANDLER Application.EnableEvents = False Range(Target, Target.Offset(0, 9)).Select ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Feste Hintergrundfarbe der aktiven Zelle bei Cursorbewegungen

Problem: Wie kann ich die Hintergrundfarbe der aktiven Zellen variabel gestalten, daß sie sich bei Cursorbewegungen anpaßt.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Public grng As Range
Public giOld As Integer, giNew As Integer

Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)
    On Error Resume Next
    giNew = Target.Interior.ColorIndex
    Target.Interior.ColorIndex = 6
    grng.Interior.ColorIndex = giOld
    giOld = giNew
    Set grng = Target
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die Ereignisse Zellauswahl, Eingabe und Doppelklick

Problem: Wie kann ich Makroanweisungen zu den Ereignissen Zellauswahl, Eingabe und Doppelklick aufrufen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle12

Private Sub Worksheet_SelectionChange( _
    ByVal Target As Excel.Range)
    If Target.Address = "$B$1" Then
        MsgBox "Bin jetzt in Zelle " & Target.Address
    End If
End Sub

Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    If Target.Address = "$B$2" Then
        MsgBox "Sie haben den Wert '" & Target.Value & "' eingegeben!"
    End If
End Sub

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Excel.Range, Cancel As Boolean)
    Cancel = True
    If Target.Address = "$B$3" Then
        MsgBox "Sie befinden sich in Zelle " & Target.Address
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auswahl eines Zellbereiches verhindern

Problem: Wie kann ich das Auswählen eines bestimmten Zellbereiches verhindern?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle12

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Not Application.Intersect(Range("A1:E10"), _
        Range(Target.Address)) _
        Is Nothing Then
        Range("F1").Select
        Beep
        MsgBox "Danger, sacra, hob i g'soagt!", , _
            Application.UserName
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makro nach Verlassen einer Zelle ausführen

Problem:	Wie kann ich Excel dazu veranlassen, nach Verlassen einer bestimmten Zelle ein Makro auszuführen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Dim bln As Boolean Private Sub Worksheet_SelectionChange _ (ByVal Target As Excel.Range) Dim rng As Range Set rng = Range("D6") If Target.Address = rng.Address Then bln = True If bln = True And Target.Address <> rng.Address Then Call MyMacro bln = False End If End Sub Sub MyMacro() MsgBox "Sie waren soeben in Zelle D6!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Blattwechsel in die Zelle mit dem aktuellen Datum springen

Problem: Beim Blattwechsel soll der Cursor immer zum aktuellen Datum in Spalte A springen.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_SheetActivate(ByVal Sh As Object)
    Dim var As Variant
    var = Application.Match(CDbl(Date), Columns(1), 0)
    If IsError(var) Then Exit Sub
    Cells(var, 1).Select
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Blattaktivierung Einlesen der Arbeitsblätter in ListBox

Problem:	Bei jedem Aktivieren eines Arbeitsblattes sollen die Blattnamen der Arbeitsmappe neu in ein Listenfeld eingelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Activate() Dim wks As Worksheet Dim lst As ListBox Set lst = ActiveSheet.ListBoxes("Listenfeld 1") lst.RemoveAllItems For Each wks In Worksheets lst.AddItem wks.Name Next wks End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Meldung bei Blattwechsel

Problem:	Bei Blattwechsel soll eine Meldung mit dem Namen des aktiven Blattes angezeigt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_SheetActivate(ByVal Sh As Object) MsgBox "Sie befinden sich in Blatt " & Sh.Name End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelleingabe automatisch in Kopfzeile übernehmen

Problem: Wie kann ich Eingaben in bestimmte Zellen eines Arbeitsblattes automatisch als Kopfzeile festlegen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_SheetChange( _
    ByVal Sh As Object, ByVal Target As Range)
    If Sh.Name = "Text" Then Exit Sub
    If Target.Address <> "$A$1" Then Exit Sub
    Sh.PageSetup.LeftHeader = Range("A1").Value
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Synchrone Steuerung von CheckBoxes in 2 Tabellen

Problem:	Wie kann ich CheckBoxes in zwei Tabellen synchronisieren? Bei Aktivierung oder Deaktivierung einer CheckBox in der einen Tabelle soll die gleiche Einstellung in der anderen erfolgen.
Lösung:	Nur anhand einer Beipspielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Access-Datenbank öffnen und deren Makro starten

Problem:	Es soll eine Access-Datenbank geöffnet, ein dort befindliches Makro gestartet und die Datenbank wieder geschlossen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallAccessMacro() Dim accApp As Object Dim sFile As String sFile = Range("B1").Value If Dir(sFile) = "" Then Beep MsgBox "Access-Datenbank wurde nicht gefunden!" Else Set accApp = CreateObject("Access.Application") accApp.OpenCurrentDatabase sFile accApp.Run "Meldung" accApp.CloseCurrentDatabase Set accApp = Nothing End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Office-Assistenten ansprechen

Problem:	Wie kann ich in XL8 den Office-Assistenten ansprechen und einsetzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Billyboy() Dim oBln As Balloon On Error GoTo ERRORHANDLER Set oBln = Assistant.NewBalloon With oBln .BalloonType = msoBalloonTypeBullets .Mode = msoModeModal .Button = msoButtonSetOK .Icon = msoIconNone .Labels(1).Text = "Die deutschsprachige " & _ "Microsoft-Knowledge-Base ..." .Show .Labels(1).Text = "... hat so einen liebenswerten " & _ "Touch ins Amateurhafte - ..." .Show .Labels(1).Text = "... es ist schön zu wissen, " & _ "daß es in unserer harten Welt ..." .Show .Labels(1).Text = "... doch noch Nischen für solche " & _ "Exceler gibt ..." .Show .Icon = msoIconAlert Beep .Labels(1).Text = "... Billyboy, mach weiter so!" .Show End With Exit Sub ERRORHANDLER: MsgBox "Der Assistent ist nicht installiert!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichentabelle StandBy

Problem:	Wie kann ich aus Excel heraus jederzeit die Zeichentabelle aufrufen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ZeichenTabelle() Shell "charmap.exe", 1 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

dBase-Datei laden, bearbeiten, speichern

Problem: Wie kann ich eine dBase-Datei über VBA laden, bestimmte Werte neu berechnen und die Datei wieder als dBase-Datei speichern?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Konvert()
    Dim rng As Range
    Dim sFile As String
    Application.ScreenUpdating = False
    On Error GoTo ERRORHANDLER
    sFile = Range("B1").Value
    If Dir(sFile) = "" Then
        Beep
        MsgBox "dBase-Datei wurde nicht gefunden!"
    Else
        Workbooks.Open sFile
        For Each rng In Range("A2:B3")
            rng.Value = rng.Value * 2
        Next rng
        Application.DisplayAlerts = False
        ActiveWorkbook.SaveAs sFile, FileFormat:=xlDBF4
        ActiveWorkbook.Close
        Application.DisplayAlerts = True
    End If
ERRORHANDLER:
    Application.ScreenUpdating = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick Telefonnummer wählen

Problem: Bei Doppelklick auf eine Zelle in Spalte A soll die dort stehende Telefonnummer angewählt werden. Die Umgebungseinstellungen in Windows müssen hierfür vorgesehen sein.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    If Target.Column > 1 Then Exit Sub
    Cancel = True
    Application.SendKeys ActiveCell.Value & "~"
    Shell "Dialer", vbNormalFocus
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Emailrundschriften mit vorgegebenem Text über Outlook

Problem: An den in Spalte A gelisteten Empfängerkreis ist ein Outlook-Email-Rundschriften mit dem Text aus einer TextBox zu richten.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code: StandardModule: basMain

```
Sub Sample()  
    Dim olApp As Object  
    Dim olMailItm As Object  
    Dim iCounter As Integer  
    Dim Dest As Variant  
    Dim SDest As String  
    Set olApp = CreateObject("Outlook.Application")  
    Set olMailItm = olApp.CreateItem(0)  
    With olMailItm  
        SDest = ""  
        For iCounter = 1 To _  
            WorksheetFunction.CountA(Columns(1))  
            If SDest = "" Then  
                SDest = Cells(iCounter, 1).Value  
            Else  
                SDest = SDest & ";" & Cells(iCounter, 1).Value  
            End If  
        Next iCounter  
        .BCC = SDest  
        .Subject = "Kundeninfo"  
        .Body = ActiveSheet.TextBoxes(1).Text  
        .Send  
    End With  
    Set olMailItm = Nothing  
    Set olApp = Nothing  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen der Arbeitsmappe Email versenden

Problem:	Beim Öffnen der Arbeitsmappe soll über Outlook eine Mail an einen vorgegebenen Email-Adressaten gesandt werden.
Lösung:	Bitte beachten: In der Entwicklungsumgebung muss ein Verweis zur Microsoft Outlook Object Library hergestellt sein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim oOL As Object Dim oOLMsg As Object Dim oOLRecip As Object Dim sUser As String Set oOL = CreateObject("Outlook.Application") Set oOLMsg = oOL.CreateItem(0) sUser = cptName If cptName = "" Then sUser = Application.UserName With oOLMsg Set oOLRecip = .Recipients.Add("hans@herber.de") oOLRecip.Type = olTo .Subject = "LogMeldung Datei " & ThisWorkbook.Name .Body = "Die Arbeitsmappe " & ThisWorkbook.Name & _ " wurde von " & sUser & " am " & Date & " um " & Time & _ " geöffnet." & vbCrLf & vbCrLf .Send End With oOLRecip.Resolve Set oOLRecip = Nothing Set oOLMsg = Nothing Set oOL = Nothing End Sub StandardModule: Modul1 Declare Function GetUserName Lib "advapi32.dll" _ Alias "GetUserNameA" (ByVal lpBuffer As String, _ nSize As Long) As Long Function cptName() Dim Buffer As String * 100 Dim BuffLen As Long BuffLen = 100 GetUserName Buffer, BuffLen End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrere Arbeitsmappen per Outlook an einen Empfänger

Problem:	Wie kann ich über Outlook gleichzeitig mehrere Arbeitsmappen an einen vorgegebenen Empfänger senden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: modMain Sub SendMessage() Dim oOL As Object Dim oOLMsg As Object Dim oOLRecip As Object Dim oOLAttach As Object Dim iRow As Integer Set oOL = CreateObject("Outlook.Application") Set oOLMsg = oOL.CreateItem(0) With oOLMsg Set oOLRecip = .Recipients.Add(Range("E1").Value) iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) Set oOLAttach = .Attachments.Add(Cells(iRow, 1).Value) iRow = iRow + 1 Loop .Subject = Format(Date, "dd.mm.yy") & " - " & Format(Time, "hh:mm:ss") .Body = "Beiliegend die Excel-Dateien" .Send End With Set oOLRecip = Nothing Set oOLMsg = Nothing Set oOL = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In Abhängigkeit von einem Formelwert Mail mit Outlook versenden

Problem: Wenn ein durch eine Formel ermittelter Zellwert einen Grenzwert übersteigt, soll eine Info-Mail mit Outlook versandt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Calculate()
    If Range("A1") > 100 Then Call SendMessage
End Sub

StandardModule: basMain

Sub SendMessage()
    Dim oOL As Object
    Dim oOLMsg As Object
    Dim oOLRecip As Object
    Set oOL = CreateObject("Outlook.Application")
    Set oOLMsg = oOL.CreateItem(0)
    With oOLMsg
        Set oOLRecip = .Recipients.Add(Range("G1").Value)
        .Subject = "Zellwert > 100"
        .Body = "Zellwert: " & ActiveSheet.Range("A1")
        .Importance = 0
        .Send
    End With
    oOLRecip.Resolve
    Set oOLMsg = Nothing
    Set oOLRecip = Nothing
    Set oOL = Nothing
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Emailversand mit Outlook auf der Basis von Excel-Daten

Problem: In einer Excel-Tabelle habe ich spaltenweise folgende Angaben: Email-Adressen, Dateinamen, Betreff, Text Wie kann ich auf Basis dieser Daten Outlook starten und die Emails mit den genannten Anlagen und Angaben in den entsprechenden Feldern versenden?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Verteilen()  
    Dim oOL As Object  
    Dim oOLMsg As Object  
    Dim oOLRecip As Object  
    Dim oOLAttach As Object  
    Dim iRow As Integer, iCounter As Integer  
    Dim sFile As String, sRec As String, sSub As String  
    Dim sBody As String  
    Dim bln  
    Application.ScreenUpdating = False  
    Application.DisplayStatusBar = True  
    bln = Application.DisplayStatusBar  
    iRow = Cells(Rows.Count, 1).End(xlUp).Row  
    Set oOL = CreateObject("Outlook.Application")  
    For iCounter = 2 To iRow  
        sRec = Cells(iCounter, 1)  
        sFile = Cells(iCounter, 2)  
        sSub = Cells(iCounter, 3)  
        sBody = Cells(iCounter, 4)  
        Application.StatusBar = "Sende Datei " & sFile & " an " & sRec & "..."  
        Set oOLMsg = oOL.CreateItem(0)  
        With oOLMsg  
            Set oOLRecip = .Recipients.Add(sRec)  
            .Subject = sSub  
            .Body = sBody & vbCrLf & vbCrLf  
            Set oOLAttach = .Attachments.Add(sFile)  
            .Send  
        End With  
        oOLRecip.Resolve  
    Next iCounter  
    Set oOL = Nothing  
    Application.StatusBar = False  
    Application.DisplayStatusBar = bln  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Inhalt von Zelle A1 mit Outlook versenden

Problem: Wie kann ich den Inhalt von Zelle A1 an eine vorgegebene Email-Adresse über Outlook versenden?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub SendMessage()  
    Dim oOL As Object  
    Dim oOLMsg As Object  
    Dim oOLRecip As Object  
    Dim sAddress As String  
    sAddress = Range("D1").Value  
    Set oOL = CreateObject("Outlook.Application")  
    Set oOLMsg = oOL.CreateItem(0)  
    With oOLMsg  
        Set oOLRecip = .Recipients.Add(sAddress)  
        .Subject = "Dies ist ein Outlook-Test"  
        .Body = ActiveSheet.Range("A1").Value  
        .Importance = 1  
        .Send  
    End With  
    oOLRecip.Resolve  
    Set oOLRecip = Nothing  
    Set oOLMsg = Nothing  
    Set oOL = Nothing  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Inhalt einer Tabelle periodisch mit Outlook versenden

Problem:	Wie kann ich eine Tabelle automatisch alle 30 Minuten an einen bestimmten Empfänger versenden? Der Versand soll über Outlook durchgeführt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Public Const gsMacro As String = "SendEmail" Public gdNextTime As Double Sub StartEmail() Dim iIntervall As Integer iIntervall = Range("B2").Value gdNextTime = Now + TimeSerial(0, 0, iIntervall) Application.OnTime earliesttime:=gdNextTime, _ procedure:=gsMacro, schedule:=True End Sub Sub StopEmail() On Error Resume Next Application.OnTime earliesttime:=gdNextTime, _ procedure:=gsMacro, schedule:=False End Sub Private Sub SendEmail() Dim oOL As Object Dim oOLMsg As Object Dim oOLRecip As Object Dim oOLAttach As Object Dim iRow As Integer, iCol As Integer Dim sTxt As String iRow = 1 iCol = 1 With Worksheets("Tabelle2") Do Until IsEmpty(.Cells(iRow, iCol)) Do Until IsEmpty(.Cells(iRow, iCol)) sTxt = sTxt & .Cells(iRow, iCol) & " " iCol = iCol + 1 Loop sTxt = WorksheetFunction.Trim(sTxt) & vbCrLf iCol = 1 iRow = iRow + 1 Loop End With Set oOL = CreateObject("Outlook.Application") Set oOLMsg = oOL.CreateItem(0) With oOLMsg Set oOLRecip = .Recipients.Add("hans@herber.de") .Subject = Format(Date, "dd.mm.yy") & " - " & Format(Time, "hh:mm:ss") .Body = sTxt .Importance = 1 For Each oOLRecip In .Recipients oOLRecip.Resolve Next .Send End With Set oOL = Nothing Call StartEmail End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eine Tabelle periodisch mit dem Standard-Emailprogramm versenden

Problem: Wie kann ich eine Tabelle automatisch alle 30 Minuten an einen bestimmten Empfänger versenden?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: modMain

Public Const gsMacro As String = "SendEmail"
Public gdNextTime As Double

Sub StartEmail()
    Dim iIntervall As Integer
    iIntervall = Range("B2").Value
    gdNextTime = Now + TimeSerial(0, 0, iIntervall)
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=True
End Sub

Sub StopEmail()
    On Error Resume Next
    Application.OnTime earliesttime:=gdNextTime, _
        procedure:=gsMacro, schedule:=False
End Sub

Private Sub SendEmail()
    Dim wks As Worksheet
    Dim iRow As Integer
    Dim sFile As String, sAddress As String
    Application.ScreenUpdating = False
    sAddress = Range("B1").Value
    ThisWorkbook.Worksheets("Tabelle2").Copy
    ActiveWorkbook.SendMail sAddress, Date
    ActiveWorkbook.Close savechanges:=False
    Call StartEmail
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus Excel heraus eine Email ohne Arbeitsmappenanlage versenden

Problem:	Wie kann ich aus Excel heraus das Email-Programm starten und Email-Adresse, Betreff sowie den Text vorgeben?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Private Declare Function ShellExecute Lib "Shell32.dll" _ Alias "ShellExecuteA" (ByVal hWnd As Long, _ ByVal lpOperation As String, ByVal lpFile As String, _ ByVal lpParameters As String, ByVal lpDirectory As String, _ ByVal nShowCmd As Long) As Long Private Sub Mail(sAdr As String, Optional sSub As String, _ Optional sBody As String) Call ShellExecute(0&, "Open", "mailto:" + sAdr + _ "?Subject=" + sSub + "&Body=" + sBody, "", "", 1) End Sub Sub MailVersenden() Dim sAddress As String, sSubject As String, sTxt As String sAddress = Range("B1").Value sSubject = Range("B2").Value sTxt = Range("B3").Value Call Mail(sAddress, sSubject, sTxt) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattbereich als Email versenden

Problem:	Wie kann ich einen vorgegebenen Bereich eines Arbeitsblattes per Email versenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub EmailVersand() Dim rng As Range Dim sAddress As String Application.ScreenUpdating = False Set rng = Range("A3:F18") sAddress = Range("B1").Value Workbooks.Add 1 rng.Copy Range("A1") Columns.AutoFit ActiveWorkbook.SendMail sAddress, "Test" ActiveWorkbook.Close savechanges:=False Application.ScreenUpdating = True End Sub</pre>



Arbeitsmappe zeitgesteuert per Email an einen Verteiler versenden

Problem:	Wie kann ich zeitgesteuert eine bestimmte Arbeitsmappe öffnen, Daten per VBA-Code eintragen lassen, per Email versenden und wieder schließen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdStart_Click() Call Versenden End Sub ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call StopEmail End Sub StandardModule: basMain Public Const gsMacro As String = "SendEmail" Public gdNextTime As Double Private Sub SendEmail() Dim wks As Worksheet Dim iRow As Integer Dim sFile As String Application.ScreenUpdating = False sFile = Range("D2").Value If Dir(sFile) = "" Then Beep MsgBox "Zu versendende Datei nicht gefunden!" Else Application.EnableEvents = False On Error GoTo ERRORHANDLER Workbooks.Open Filename:=sFile, updatelinks:=False iRow = 1 With ThisWorkbook.Worksheets("Tabelle1") Do Until IsEmpty(.Cells(iRow, 1)) ActiveWorkbook.SendMail .Cells(iRow, 1).Value, Date iRow = iRow + 1 Loop End With ActiveWorkbook.Close savechanges:=False End If Call StartEmail ERRORHANDLER: Application.EnableEvents = True Application.ScreenUpdating = True End Sub Sub StartEmail() Dim iIntervall As Integer iIntervall = Range("D1").Value gdNextTime = Now + TimeSerial(0, 0, iIntervall) Application.OnTime earliesttime:=gdNextTime, _ procedure:=gsMacro, schedule:=True End Sub Sub StopEmail() On Error Resume Next Application.OnTime earliesttime:=gdNextTime, _ procedure:=gsMacro, schedule:=False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe als Email versenden

Problem:	Wie kann ich aus XL8 heraus die aktive Arbeitsmappe per Email versenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CommandButton1_Click() Call Senden End Sub StandardModule: basMain Sub Senden() ThisWorkbook.SendMail _ Recipients:="hans@herber.de", _ Subject:="Ein Test" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei senden

Problem:	Wie kann ich über VBA eine Exceldatei über das Standard-Mailprogramm versenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Book4Bill() ActiveWorkbook.SendMail Recipients:= _ "bgates@microsoft.com", _ Subject:="Hi Bill" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Druckbereiche als GIF-Grafiken exportieren

Problem: Wie kann ich die Druckbereiche verschiedener Tabellenblätter als GIF-Grafiken exportieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub ScreenShot()
    Dim wks As Worksheet
    Dim cht As Chart
    Dim iCounter As Integer
    Dim sPath As String
    sPath = Application.Path & "\"
    For iCounter = 2 To Worksheets.Count
        Set wks = Worksheets(iCounter)
        wks.Range(wks.PageSetup.PrintArea).CopyPicture _
            Appearance:=xlScreen, _
            Format:=xlPicture
        Set cht = Charts.Add
        cht.ChartArea.Clear
        On Error Resume Next
        cht.Paste
        On Error GoTo 0
        cht.Export sPath & wks.Name & ".gif"
        Application.DisplayAlerts = False
        cht.Delete
        Application.DisplayAlerts = True
    Next iCounter
    sPath = Left(sPath, Len(sPath) - 1)
    MsgBox "Die Grafiken wurden im Verzeichnis " & _
        sPath & " gespeichert!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Diagramm exportieren und in Tabellenblatt importieren

Problem:	Wie kann ich ein Excel-Diagramm als *.GIF-Grafik exportieren, in ein Tabellenblatt importieren und die Grafik wieder löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DiaExport() Dim pct As Picture Dim sFile As String sFile = Application.Path & "\test.gif" Sheets("Diagramm1").Export sFile Application.DisplayAlerts = False Sheets("Diagramm1").Delete Application.DisplayAlerts = True Worksheets.Add after:=Worksheets(Worksheets.Count) Set pct = ActiveSheet.Pictures.Insert(sFile) Kill sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufruf von Textdateien protokollieren

Problem:	Bei Doppelklick auf die Zellen in A1:A3 werden die dort genannten Dateien im Explorer aufgerufen. Der Vorgang wird in einer Log-Datei dokumentiert.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean) Cancel = True If Dir(Target.Value) = "" Then Exit Sub Close Open Application.DefaultFilePath & "\hyp.log" For Append As #1 Print #1, Target.Value & vbTab & Now Close Shell "explorer " & Target.Value, vbMaximizedFocus End Sub Private Sub Worksheet_FollowHyperlink(ByVal Target As Hyperlink) Close Open Application.DefaultFilePath & "\hyp.log" For Append As #1 Print #1, Target.Address & vbTab & Now Close End Sub StandardModule: Modul1 Sub WriteLog() End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafikprogramm aufrufen und Bild laden

Problem:	Wie kann ich bei Doppelklick auf eine Zelle ein Grafikprogramm aufrufen und die Grafikdatei mit dem in der Zelle stehenden Namen laden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_BeforeDoubleClick(_ ByVal Target As Range, Cancel As Boolean) Dim sFile As String Cancel = True If Target.Address <> "\$B\$1" Then Exit Sub sFile = Target.Value If Dir(sFile) = "" Then Beep MsgBox "Grafikdatei wurde nicht gefunden!" Else On Error GoTo ERRORHANDLER Shell "pbrush.exe " & Target.Value, 1 End If Exit Sub ERRORHANDLER: MsgBox "PBrush ist nicht installiert!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalt als Unicode-Zeichen in HTML-Tabelle schreiben

Problem:	Der Text aus Zelle A1 wird in einer HTMLDatei gespeichert, wobei die slowakischen Sonderzeichen in Unicode-Nummern umgewandelt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Start() Dim wks As Worksheet Dim iRow As Integer Dim sTxt As String, sFile As String Application.ScreenUpdating = False sFile = Application.DefaultFilePath & "\unicode.html" ActiveSheet.Copy iRow = 2 Set wks = ThisWorkbook.Worksheets("Data") Do Until IsEmpty(wks.Cells(iRow, 1)) Cells.Replace _ what:=wks.Cells(iRow, 1).Value, _ replacement:=wks.Cells(iRow, 2).Value, _ lookat:=xlPart, _ MatchCase:=True iRow = iRow + 1 Loop Open sFile For Output As #1 Print #1, "<html><head>" Print #1, "<meta HTTP-EQUIV=""Content-Type"" CONTENT=""text/html; charset=UTF-8"">" Print #1, "</head>" Print #1, "<body>" Print #1, Cells(1, 1).Value Print #1, "</body></html>" Close ActiveWorkbook.Close savechanges:=False Shell "explorer " & sFile, vbNormalFocus Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBA-generierte HTML-Sequenz HTML-formatiert in Tabelle

Problem:	Wenn Sie aus einem HTML-Dokument Text kopiert haben, können Sie diesen über die Zwischenablage in ein ExcelArbeitsblatt einfügen. Die gleiche Funktionalität wird hier für über VBA generierten HTML-Code zur Verfügung gestellt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Private Declare Function RegisterClipboardFormat Lib "user32" _ Alias "RegisterClipboardFormatA" (ByVal lpString As String) _ As Long Sub GetHTML() HTMLToClipboard "<html><body><table><tr width=""240"">" & _ "<td bgcolor=""blue"">" & _ "Dies ist ein Beispieltext</td></tr></table></body></html>" End Sub Public Sub HTMLToClipboard(HTMLText As String) Dim nCFHTML As Long Dim nClipboardText As String nCFHTML = RegisterClipboardFormat("HTML Format") nClipboardText = "Version:0.9" & vbCrLf nClipboardText = nClipboardText & "StartHTML:-1" & vbCrLf nClipboardText = nClipboardText & "EndHTML:-1" & vbCrLf nClipboardText = nClipboardText & "StartFragment:000081" & vbCrLf nClipboardText = nClipboardText & "EndFragment:000000" & vbCrLf nClipboardText = nClipboardText & HTMLText & vbCrLf nClipboardText = Replace(nClipboardText, "000000", _ Format\$(Len(nClipboardText), "000000")) With New DataObject .Clear .SetText StrConv(nClipboardText, vbFromUnicode), nCFHTML .PutInClipboard End With Range("A1").Select ActiveSheet.Paste End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hierarchisch aufgebautes Inhaltsverzeichnis als HTML-File

Problem:	Aus einer Liste in den Spalten A:C soll ein hierarchisch aufgebautes Inhaltsverzeichnis als HTML-Datei aufgebaut werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: Modul1 <pre>Sub CreateHTML() Dim iRow As Integer, iStage As Integer, iCounter As Integer, iPage As Integer Dim sFile As String sFile = Application.DefaultFilePath & "\test.htm" Close Open sFile For Output As #1 Print #1, "<html>" Print #1, "<head>" Print #1, "<style type='\"text/css\"'>" Print #1, " body { font-size:12px;font-family:tahoma } " Print #1, "</style>" Print #1, "</head>" Print #1, "<body>" iRow = 2 Do While WorksheetFunction.CountA(Rows(iRow)) > 0 If Not IsEmpty(Cells(iRow, 1)) Then For iCounter = 1 To iStage Print #1, "" iStage = iStage - 1 Next iCounter Print #1, "" Print #1, "" & Cells(iRow, 1).Value & "" iPage = iPage + 1 If iStage < 1 Then iStage = iStage + 1 End If End If If Not IsEmpty(Cells(iRow, 2)) Then For iCounter = 2 To iStage Print #1, "" iStage = iStage - 1 Next iCounter Print #1, "" Print #1, "" & Cells(iRow, 2).Value & "" iPage = iPage + 1 If iStage < 2 Then iStage = iStage + 1 End If End If If Not IsEmpty(Cells(iRow, 3)) Then If iStage < 3 Then Print #1, "" End If Print #1, "" & Cells(iRow, 3).Value & "" iPage = iPage + 1 If iStage < 3 Then iStage = iStage + 1 End If End If iRow = iRow + 1 Loop For iCounter = 2 To iStage Print #1, " " iStage = iStage - 1 Next iCounter Print #1, "</body>" Print #1, "</html>" Close Shell "hh " & vbCrLf & sFile, vbMaximizedFocus End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafik-Dateinamen aus Zelle auslesen und HTML-Datei erstellen

Problem: Über die erste Schaltfläche soll eine HTML-Datei mit der Anzeige der in Zelle A1 genannten Grafik erstellt werden, die über die zweite Schaltfläche aufgerufen werden kann.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code: StandardModule: Modul1

```
Sub CreateHTML()  
    Dim sPath As String  
    sPath = Application.Path  
    Close  
    Open sPath & "\test.htm" For Output As #1  
    Print #1, "<html><body>"  
    Print #1, "<img src="" & Range("A1").Value & "">"  
    Print #1, "</body></html>"  
    Close  
End Sub  
  
Sub CallHTML()  
    Dim appIE As Object  
    Set appIE = CreateObject("InternetExplorer.Application")  
    appIE.navigate Application.Path & "\test.htm"  
    appIE.Visible = True  
    Set appIE = Nothing  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Von allen Arbeitsblättern der Mappe HTML-Dateien anlegen

Problem:	Von allen Arbeitsblättern der Arbeitsmappe sollen HTML-Dateien angelegt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub XL2HTML() Dim wks As Worksheet Dim iRow As Integer, iCol As Integer Dim sFile As String, sPath As String sPath = Application.Path Close For Each wks In ActiveWorkbook.Worksheets wks.Select sFile = sPath & "\" & wks.Name & ".htm" iRow = 1 Open sFile For Output As #1 Print #1, "<html><body><table border=1>" Do Until IsEmpty(Cells(iRow, 1)) iCol = 1 Print #1, "<tr>" Do Until IsEmpty(Cells(1, iCol)) If Not IsEmpty(Cells(iRow, iCol)) Then If iRow = 1 Then Print #1, "<th>" & Cells(iRow, iCol) & "</th>" Else Print #1, "<td>" & Cells(iRow, iCol) & "</td>" End If Else Print #1, "<td>&nbsp;</td>" End If iCol = iCol + 1 Loop Print #1, "</tr>" iRow = iRow + 1 Loop Print #1, "</table></body></html>" Close Next wks Worksheets(2).Select MsgBox "Die Dateien wurden im Verzeichnis " & sPath & " gespeichert!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Titel einer HTML-Seite auslesen

Problem:	Aus einer HTML-Seite soll der Titel ausgelesen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub ReadTitle() Dim iFile As Integer Dim sFile As String, sTxt As String Dim bln As Boolean sFile = Range("B1").Value If Dir(sFile) = "" Then Beep MsgBox "Datei " & sFile & " nicht gefunden!" Exit Sub End If iFile = FreeFile Open sFile For Input As iFile Do Until EOF(1) Line Input #iFile, sTxt If InStr(UCase(sTxt), "<TITLE>") Then sTxt = Right(sTxt, Len(sTxt) - 7) sTxt = Left(sTxt, InStr(UCase(sTxt), "</TITLE>") - 1) bln = True Exit Do End If Loop Close iFile If bln Then MsgBox sTxt Else MsgBox "Keinen Titel gefunden!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus Zelleintragungen Hyperlinks und HTML-Tabelle erstellen

Problem:	Aus den Texten in den Spalten A und B sollen zum einen Excel-Hyperlinks und zum anderen eine HTML-Seite erstellt werden.
Lösung:	Nur anhand einer Beispieldarstellungsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

HTML-Datei im Internet-Explorer mit fester Fenstergrösse

Problem:	Wie kann ich eine HTML-Datei auswählen und mir diese im InternetExplorer mit einer festen Fenstergrösse ohne das Explorer-Menü anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ExplorerOeffnen() Dim oExplorer As Object Dim var As Variant var = Application.GetOpenFilename("HTML-Dateien (*.htm), *.htm") If var = False Then Exit Sub Set oExplorer = CreateObject("InternetExplorer.Application") With oExplorer .Navigate var .StatusBar = False .MenuBar = False .ToolBar = False .Visible = True .Resizable = False .Offline = True .Width = 750 .Height = 550 End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

HTML-Datei aus gefilterten Daten mit Hyperlinks aufbauen

Problem:	Wie kann ich mit dem Autofilter gefilterte Daten so in einem HTMLDatei überführen, daß auch die Hyperlinks übernommen werden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Visible2HTML() Dim rng As Range Dim hyp As Hyperlink Dim iRow As Integer, iCol As Integer, iFile As Integer Dim sFile As String Set rng = Range("A1").CurrentRegion iFile = FreeFile sFile = Application.Path & "\testhtml.htm" Open sFile For Output As iFile Print #iFile, "<html>" Print #iFile, "<head>" Print #iFile, "<title>Bedingte HTML-Übernahme</title>" Print #iFile, "<style>" Print #iFile, " th" Print #iFile, " {" Print #iFile, " font-family=tahoma,verdana;" Print #iFile, " font-size=12px;" Print #iFile, " font-weight=bold" Print #iFile, " }" Print #iFile, "" Print #iFile, " td" Print #iFile, " {" Print #iFile, " font-family=tahoma,verdana;" Print #iFile, " font-size=12px;" Print #iFile, " }" Print #iFile, "</style>" Print #iFile, "<head>" Print #iFile, "<body>" Print #iFile, "<table border=1 cellpadding=3 cellspacing=1>" Print #iFile, " <tr>" For iCol = 1 To rng.Columns.Count Print #iFile, " <th bgcolor=#ffffe0>" & Cells(1, iCol).Value & "</th>" Next iCol Print #iFile, " </tr>" For iRow = 2 To rng.Rows.Count If Rows(iRow).Hidden = False Then Print #iFile, " <tr>" For iCol = 1 To rng.Columns.Count If Cells(iRow, iCol).Hyperlinks.Count > 0 Then Set hyp = Cells(iRow, iCol).Hyperlinks(1) End If If hyp Is Nothing Then Print #iFile, " <td>" & Cells(iRow, iCol).Value & "</td>" Else Print #iFile, " <td>" & Cells(iRow, iCol).Value & "</td>" End If Set hyp = Nothing Next iCol Print #iFile, " </tr>" End If Next iRow Print #iFile, "</table>" Print #iFile, "</body>" Print #iFile, "</html>" Close iFile Shell "explorer " & sFile, vbMaximizedFocus End Sub Sub a() Close End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur Zeilen mit bestimmtem Inhalt in HTML-Datei konvertieren

Problem:	Wie kann ich aus einer Tabelle nur die Zeilen in eine HTML-Datei konvertieren, in denen eine Zelle einen bestimmten Inhalt hat?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BedingtZuHTML() Dim rng As Range Dim var As Variant Dim iRow As Integer, iCol As Integer, iFile As Integer Dim sFile As String Set rng = Range("A1").CurrentRegion iFile = FreeFile sFile = Application.Path & "\testhtml.htm" Open sFile For Output As iFile Print #iFile, "<html>" Print #iFile, "<head>" Print #iFile, "<title>Bedingte HTML-Übernahme</title>" Print #iFile, "<style>" Print #iFile, " th" Print #iFile, " {" Print #iFile, " font-family=tahoma,verdana;" Print #iFile, " font-size=12px;" Print #iFile, " font-weight=bold" Print #iFile, " }" Print #iFile, "" Print #iFile, " td" Print #iFile, " {" Print #iFile, " font-family=tahoma,verdana;" Print #iFile, " font-size=12px;" Print #iFile, " }" Print #iFile, "</style>" Print #iFile, "<head>" Print #iFile, "<body>" Print #iFile, "<table border=1 cellpadding=3 cellspacing=1>" Print #iFile, " <tr>" For iCol = 1 To rng.Columns.Count Print #iFile, " <th bgcolor=#ffffe0>" & Cells(1, iCol).Value & "</th>" Next iCol Print #iFile, " </tr>" For iRow = 1 To rng.Rows.Count var = Application.Match("Zu HTML", Rows(iRow), 0) If Not IsError(var) Then Print #iFile, " <tr>" For iCol = 1 To rng.Columns.Count Print #iFile, " <td>" & Cells(iRow, iCol).Value & "</td>" Next iCol Print #iFile, " </tr>" End If Next iRow Print #iFile, "</table>" Print #iFile, "</body>" Print #iFile, "</html>" Close iFile Shell "explorer " & sFile, vbMaximizedFocus End Sub</pre>



Dateinamen eines Verzeichnisses als HTML-Datei

Problem:	Wie kann ich alle Dateien eines anzugebenden Verzeichnisses in einer HTML-Datei auflisten und einen Link zu ihnen erstellen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Sub Dir2HTML() Dim iCounter As Integer, iFile As Integer Dim sPath As String, sFile As String iFile = FreeFile sPath = Range("B1").Value sFile = Range("B2").Value On Error GoTo ERRORHANDLER Open sFile For Output As iFile Print #iFile, "<html>" Print #iFile, "<head>" Print #iFile, "<title>Dateien im Verzeichnis " & sPath & "</title>" Print #iFile, "<style>" Print #iFile, "th" Print #iFile, "{" Print #iFile, "font-family=tahoma,verdana;" Print #iFile, "font-size=12px;" Print #iFile, "font-weight=bold" Print #iFile, "}" Print #iFile, "" Print #iFile, "td" Print #iFile, "{" Print #iFile, "font-family=tahoma,verdana;" Print #iFile, "font-size=12px;" Print #iFile, "}" Print #iFile, "</style>" Print #iFile, "<head>" Print #iFile, "<body>" Print #iFile, "<table border=1 cellpadding=3 cellspacing=1>" Print #iFile, "<tr><th colspan=2 bgcolor=#ffffe0>" & _ Dateiliste des Ordners " & sPath & "</th></tr>" Print #iFile, "<tr><th align=left>Dateidatum</th>" & _ "<th align=left>Dateiname</th></tr>" With Application.FileSearch .LookIn = sPath .Execute For iCounter = 1 To .FoundFiles.Count Print #iFile, "<tr><td>" & _ FileDateTime(.FoundFiles(iCounter)) & _ "</td><td>" & _ "" & _ .FoundFiles(iCounter) & _ "</td></tr>" Next iCounter End With Print #iFile, "</table>" Print #iFile, "</body>" Print #iFile, "</html>" Close iFile Shell "explorer " & sFile, vbMaximizedFocus Exit Sub ERRORHANDLER: MsgBox "Die HTML-Datei konnte nicht angelegt werden!" End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Excel zu HTML

Problem:	Wie kann ich eine Excel-Tabelle möglichst schnell in das HTML-Format konvertieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub XL2HTML() Dim iRow As Integer, iCol As Integer Dim sFile As String sFile = Range("K1").Value Close On Error GoTo ERRORHANDLER Open sFile For Output As #1 On Error GoTo 0 Print #1, "<html>" Print #1, "<header>" Print #1, "<style type='\"text/css\"'>" Print #1, "td {" Print #1, " font-size:9pt;" Print #1, " font-family:tahoma,verdana;" Print #1, " background-color:#ffffe0" Print #1, "}" Print #1, "</style>" Print #1, "</header>" Print #1, "<body bgcolor=#d0d0d0><center>" Print #1, "<table bgcolor=#003000 border=3" & _ " cellpadding=3 cellspacing=3>" For iCol = 1 To 8 Print #1, " <td>" & Cells(1, iCol).Text & "</td>" Next iCol For iRow = 2 To 7 Print #1, " <tr>" For iCol = 1 To 8 If Not IsEmpty(Cells(iRow, iCol)) Then If iCol < 3 Then Print #1, " <td>" & Cells(iRow, iCol).Text & "</td>" Else Print #1, " <td align=right>" & _ Cells(iRow, iCol).Text & "</td>" End If Else Print #1, " <td>&nbsp;</td>" End If Next iCol Print #1, " </tr>" Next iRow Print #1, "</table></center>" Print #1, "</body>" Print #1, "</html>" MsgBox "Datei wurde angelegt:" & vbLf & sFile End ERRORHANDLER: MsgBox "Bitte den Verzeichnisnamen überprüfen!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Redirect-Adressen aus Internetseiten auslesen

Problem: Die in den Internetdateien aus Spalte A festgelegten Redirect-Adressen sollen in Spalte B eingetragen werden.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Sub MessageAbrufen()  
    Dim IE As Object  
    Dim iRow As Integer  
    iRow = 1  
    Do Until IsEmpty(Cells(iRow, 1))  
        Set IE = CreateObject("InternetExplorer.Application")  
        IE.Visible = True  
        IE.Navigate Cells(iRow, 1).Value  
        Application.Wait Now + TimeSerial(0, 0, 2)  
        SendKeys "{tab}^c%{F4}"  
        Application.Wait Now + TimeSerial(0, 0, 2)  
        Cells(iRow, 2).Select  
        ActiveSheet.Paste  
        Set IE = Nothing  
        iRow = iRow + 1  
    Loop  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriffe an Google übergeben und Ergebnis aufrufen

Problem:	Über eine UserForm soll Google nach einem Suchbegriff suchen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Suchen_Click() Dim sSuche As String Dim sLink1 As String Dim sLink2 As String sLink1 = "http://www.google.de/search?hl=de&ie=ISO-8859-1&oe=ISO-8859-1&q=+" sLink2 = "&btnG=Suche&meta=" Call Shell("C:\Programme\Internet Explorer\IEXPLORE.EXE") End Sub ClassModule: frmSearch Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdSearch_Click() Dim sLink1 As String Dim sLink2 As String sLink1 = "http://www.google.de/search?hl=de&ie=ISO-8859-" & _ "1&oe=ISO-8859-1&q=+" sLink2 = "&btnG=Suche&meta=" Shell ("C:\Programme\Internet Explorer\IEXPLORE.EXE " & _ sLink1 & txtSearch.Text & sLink2), vbMaximizedFocus Unload Me End Sub StandardModule: Modul2 Sub CallForm() frmSearch.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausländische UStNr nach dem WDDX-Verfahren prüfen

Problem:	Die Gültigkeit der Steuernummer in Spalte A soll im WDDX-Verfahren geprüft werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub IsUStIDOK() Dim vRow As Variant Dim iQuery As Integer, iChar As Integer Dim sQuery As String, sNo As String, sTxt As String iQuery = 2 Do Until IsEmpty(Cells(iQuery, 1)) sQuery = "http://wddx.bff-online.de/ustid.php?eigene_id=" & _ Range("H1").Value & "&abfrage_id=" & Cells(iQuery, 1).Value With ActiveSheet.QueryTables.Add(Connection:= _ "URL;" & sQuery, Destination:=Range("IV1")) .Name = "ustid" .Refresh BackgroundQuery:=False End With sTxt = Range("IV3").Value For iChar = 1 To Len(sTxt) - 20 If Mid(sTxt, iChar, 20) Like "<string>###</string>" Then sNo = Mid(sTxt, iChar + 8, 3) Exit For End If Next iChar If IsNumeric(sNo) Then vRow = Application.Match(CInt(sNo), Worksheets("Codes").Columns(1), 0) If Not IsError(vRow) Then Cells(iQuery, 2).Value = Worksheets("Codes").Cells(vRow, 2).Value End If Else Cells(iQuery, 2).Value = "Problem: " & sNo End If Columns("IV").Delete iQuery = iQuery + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Automatische Passworteingabe in Internet-Formular

Problem:	Über den IExplorer-Dialog soll eine automatische Passworteingabe erfolgen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallSearch() frmSuche.Show End Sub ClassModule: frmSuche Private Sub cmdBack_Click() WebBrowser1.Navigate "http://www.herber.de/form_test1.html" cmdBack.Enabled = False cmdOK.Enabled = True End Sub Private Sub cmdCancel_Click() Unload Me End Sub Private Sub cmdOK_Click() WebBrowser1.Document.all("username").innertext = "Hans W. Herber" WebBrowser1.Document.all("password").innertext = "Herbers Passwort" WebBrowser1.Document.Forms(0).Item("Submit").Click cmdOK.Enabled = False End Sub Private Sub UserForm_Initialize() WebBrowser1.Navigate "http://www.herber.de/form_test1.html" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Pingwerte von WebServern auslesen

Problem:	Die Pingwerte einer Liste von WebServern sollen in eine Tabelle eingetragen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub PingHWSites() Dim rng As Range Dim v As Variant Dim iCounter As Integer Dim sTxt As String, sTime As String, sFile As String sFile = Application.DefaultFilePath & "\Ping.txt" Close Set rng = Range("A2") Do Until IsEmpty(rng.Value) On Error Resume Next If InStr(Application.OperatingSystem, "NT") Then v = Shell("cmd.exe /C ping " & rng.Value & ">" & sFile) Else MsgBox "Bitte beachten:" & vbCrLf & "Die Routine funktioniert" & _ " nur auf NT-Systemen zuverlässig!" v = Shell("command.com /C ping " & rng.Value & ">" & sFile) End If If Err = 0 Then Do DoEvents AppActivate v Loop Until Err <> 0 On Error GoTo 0 Open sFile For Input As #1 Do Until EOF(1) Line Input #1, sTxt If InStr(sTxt, "Minimum") Then sTime = Right(sTxt, Len(sTxt) - InStr(sTxt, "Minimum") - 9) sTime = Left(sTime, InStr(sTime, ",") - 1) rng.Offset(0, 1).Value = sTime End If If InStr(sTxt, "Maximum") Then sTime = Right(sTxt, Len(sTxt) - InStr(sTxt, "Maximum") - 9) sTime = Left(sTime, InStr(sTime, ",") - 1) rng.Offset(0, 2).Value = sTime End If If InStr(sTxt, "Mittelwert") Then sTime = Right(sTxt, Len(sTxt) - InStr(sTxt, "Mittelwert") - 11) rng.Offset(0, 3).Value = sTime End If Loop Close #1 Else rng.Offset(0, 1) = "Failed to shell" End If Set rng = rng.Offset(1, 0) Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafikdatei aus dem Internet downloaden und in Tabelle einfügen

Problem: In das aktive Blatt soll eine downzuladende Grafik aus dem Internet eingefügt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Private Declare Function URLDownloadToFile Lib "urlmon" _
    Alias "URLDownloadToFileA" ( _
        ByVal pCaller As Long, _
        ByVal szURL$, _
        ByVal szFileName$, _
        ByVal dwReserved As Long, _
        ByVal lpfnCB As Long) As Long

Sub Downl()
    Dim pct As Picture
    Dim lResult As Long
    Dim sURL As String, sLocalFile As String
    Range("B7").Select
    sURL = Range("B1").Value
    sLocalFile = Application.DefaultFilePath & "start2.gif"
    lResult = URLDownloadToFile(0, sURL, sLocalFile, 0, 0)
    Set pct = ActiveSheet.Pictures.Insert(sLocalFile)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Download und Öffnen einer Arbeitsmappe

Problem: Eine Excel-Arbeitsmappe soll aus dem Internet downgeladen und danach (nicht im Browser) geöffnet werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Private Declare Function URLDownloadToFile Lib "urlmon" _
    Alias "URLDownloadToFileA" ( _
        ByVal pCaller As Long, _
        ByVal szURL$, _
        ByVal szFileName$, _
        ByVal dwReserved As Long, _
        ByVal lpfnCB As Long) As Long

Sub Downl()
    Dim lResult As Long
    Dim sURL$, sLocalFile$
    sURL = "http://www.herber.de/bbs/texte/11names.xls"
    sLocalFile = Application.DefaultFilePath & "\11names.xls"
    lResult = URLDownloadToFile(0, sURL, sLocalFile, 0, 0)
    Workbooks.Open Application.DefaultFilePath & "\11names.xls"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausführbare Datei downloaden und starten

Problem: Aus dem Internet soll eine ausführbare Datei downgeladen und danach gestartet werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Private Declare Function URLDownloadToFile Lib "urlmon" _
    Alias "URLDownloadToFileA" ( _
        ByVal pCaller As Long, _
        ByVal szURL$, _
        ByVal szFileName$, _
        ByVal dwReserved As Long, _
        ByVal lpfnCB As Long) As Long

Sub Downl()
    Dim lResult As Long
    Dim sURL$, sLocalFile$
    sURL = "http://217.160.138.156/andere/bean.exe"
    sLocalFile = Application.DefaultFilePath & "\bean.exe"
    lResult = URLDownloadToFile(0, sURL, sLocalFile, 0, 0)
    Shell Application.DefaultFilePath & "\bean.exe", vbNormalFocus
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Internet-Tabelle aktualisieren

Problem:	Die Tabelle soll auf Schaltflächenklick aktualisiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Aufruf() WebAufruf ("http://www.herber.de/cgi-bin/ul_hits.pl") End Sub Function WebAufruf(wName\$) With ActiveSheet.QueryTables.Add(Connection:= _ "URL;" & wName, _ Destination:=Range("A1")) .FieldNames = False .RefreshStyle = xlInsertDeleteCells .RowNumbers = False .FillAdjacentFormulas = False .RefreshOnFileOpen = False .HasAutoFormat = True .BackgroundQuery = True .TablesOnlyFromHTML = True .Refresh BackgroundQuery:=False .SavePassword = False .SaveData = True End With End Function Sub Aktualisieren() Worksheets("Daten").Range("A1").QueryTable.Refresh _ BackgroundQuery:=False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Binärdatei aus dem Internet downloaden und speichern

Problem:	Aus dem Internet soll eine Binärdatei auf den Rechner geladen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Private Declare Function URLDownloadToFile Lib "urlmon" _ Alias "URLDownloadToFileA" (_ ByVal pCaller As Long, _ ByVal szURL\$, _ ByVal szFileName\$, _ ByVal dwReserved As Long, _ ByVal lpfnCB As Long) As Long Sub Downl() Dim lResult As Long Dim sURL\$, sLocalFile\$ sURL = "http://www.excel-cd.de/downloads/xlBasics.zip" sLocalFile = Application.DefaultFilePath & "xlbasics.zip" lResult = URLDownloadToFile(0, sURL, sLocalFile, 0, 0) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Excel-Arbeitsmappe aus dem Internet öffnen und verschieben

Problem:	Eine Excel-Tabelle aus dem Internet soll importiert und in diese Arbeitsmappe verschoben werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ImportMove() Application.ScreenUpdating = False Application.VBE.MainWindow.Visible = False On Error GoTo ERRORHANDLER Workbooks.Open "http://www.herber.de/bbs/texte/10filter.xls" With ThisWorkbook ActiveSheet.Move after:=.Worksheets(.Worksheets.Count) .Worksheets(.Worksheets.Count).Name = "Import" End With ERRORHANDLER: Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

HTML-Seiten den Favoriten hinzufügen

Problem:	Den Favoriten soll ein neuer Ordner mit Internetadressen hinzugefügt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SetFavorites() Dim rng As Range Dim sPath As String sPath = WindowsFolder & "\Favoriten" On Error Resume Next MkDir sPath sPath = sPath & "\" & ActiveSheet.Name MkDir sPath On Error GoTo 0 Close For Each rng In Range("A1").CurrentRegion.Cells If rng.Hyperlinks.Count > 0 Then Open sPath & "\" & rng.Value & ".URL" For Output As #1 Print #1, CreateLink(rng.Hyperlinks(1).Address) Close End If Next rng MsgBox "Die Favoriten wurden angelegt!" End Sub Private Function CreateLink(sLink As String) Dim sTxt As String sTxt = "[Default]" & vbCrLf sTxt = sTxt & "BASEURL=" & sLink & vbCrLf & vbCrLf sTxt = sTxt & "[InternetShortcut]" & vbCrLf sTxt = sTxt & "URL=" & sLink & vbCrLf CreateLink = sTxt End Function StandardModule: basFunctions Declare Function GetWindowsDirectory Lib "kernel32.dll" _ Alias "GetWindowsDirectoryA" (_ ByVal lpBuffer As String, ByVal nSize As Long) As Long Function WindowsFolder() Dim lLength As Long Dim sWinDir As String sWinDir = Space(255) lLength = GetWindowsDirectory(sWinDir, 255) WindowsFolder = Left(sWinDir, lLength) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Animierte Grafik in Tabellenblatt anzeigen

Problem: In einem Tabellenblatt soll eine animierte Grafik aus dem Internet angezeigt werden. Der IExplorer ab 5.0 muss installiert sein.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Tabelle1.WebBrowser1.Navigate _
        "http://www.herber.de/images/cd.gif"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suche im Internet in Herbers Excel-Archiv

Problem:	Wie kann ich Herbers Excel-Archiv im Internet durchsuchen?
Lösung:	Den nachstehenden Code in den genannten Modulen platzieren.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallWW() frmWW.Show End Sub ClassModule: frmWW Private Sub cmdCancel_Click() Unload Me End Sub Private Sub WebBrowser1_NavigateComplete2(_ ByVal pDisp As Object, URL As Variant) Me.Caption = "Suche in Herbers Excel/VBA-Archiv" End Sub Private Sub UserForm_Initialize() Dim num As String, as_q As String, _ as_epq As String, as_eq As String, as_oq As String num = Range("num").Value as_q = Range("as_q").Value as_epq = Range("as_epq").Value as_eq = Range("as_eq").Value as_oq = Range("as_oq").Value With WebBrowser1 .Navigate "http://www.google.de/search?as_q=" & _ as_q & "&num=" & num & "&hl=de&btnG=Google-Suche&as_epq=" & _ as_epq & "&as_oq=" & as_oq & " &as_eq=" & as_eq & _ "&lr=&as_ft=i&as_filetype=&as_qdr=all&as_occt=any&as_dt=i" & _ "&as_sitesearch=www.herber.de" End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

HTML-Seite aus dem Internet aufrufen und als Textdatei speichern

Problem:	Eine Internet-HTML-Seite soll ausgelesen und als Textdatei im Verzeichnis dieser Arbeitsmappe gespeichert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Aufruf() Call URL_Load("http://www.kicker.de") End Sub Private Sub URL_Load(ByVal sURL As String) Dim appIE As Object Dim sTxt As String Set appIE = CreateObject("InternetExplorer.Application") appIE.navigate sURL Do: Loop Until appIE.Busy = False Do: Loop Until appIE.Busy = False sTxt = appIE.document.documentElement.outerHTML Set appIE = Nothing Close Open ThisWorkbook.Path & "\test.txt" For Output As #1 Print #1, sTxt Close MsgBox "Der Text wurde gespeichert unter:" & vbCrLf & _ Application.Path & "\test.txt" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus verschiedenen Zellwerten zusammengesetzte Webadresse aufrufen

Problem:	Die Web-Adresse http://www.herber.de soll um die Verzeichnisse und den Dateinamen in Spalte A ergänzt und aufgerufen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CommandButton1_Click() Dim iRow As Integer Dim sPath As String sPath = "http://www.herber.de" iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) sPath = sPath & "/" & Cells(iRow, 1) iRow = iRow + 1 Loop sPath = Left(sPath, Len(sPath) - 1) ActiveWorkbook.FollowHyperlink _ Address:=sPath, _ NewWindow:=True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Google zu einem Suchbegriff gefundene Links auflisten

Problem:	Bei Google soll eine Suchanfrage gestartet und aus der Ergebnisliste alle Links in einer Tabelle gelistet werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub HTMLOAuslesen() Dim wks As Worksheet Dim lRow As Integer, iRow As Integer, iRowT As Integer Dim sSearch As String sSearch = "http://www.google.de/search?" sSearch = sSearch & "q=excel&hl=de&btnG=Google-Suche" sSearch = sSearch & "&meta=lr%3Dlang_de" Application.ScreenUpdating = False Workbooks.Open sSearch Set wks = ActiveSheet lRow = wks.Cells(Rows.Count, 1).End(xlUp).Row Workbooks.Add For iRow = 1 To lRow If wks.Cells(iRow, 1).Hyperlinks.Count = 1 Then iRowT = iRowT + 1 Cells(iRowT, 1) = wks.Cells(iRow, 1).Hyperlinks(1).Address End If Next iRow wks.Parent.Close savechanges:=False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Internet-Prüfung, ob Domaines noch verfügbar sind

Problem:	Es soll geprüft werden, ob die in Spalte A gelisteten Domaines noch verfügbar sind.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub AdressePruefen() Dim iRow As Integer Dim sTxt As String For iRow = 1 To 6 sTxt = MessageLaden("http://www.ripe.net/perl/whois?query=" & _ & Cells(iRow, 1) & "&.submit=Anfrage+senden") If InStr(sTxt, "descr: ") Then sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, "descr: ") - 12) sTxt = Left(sTxt, InStr(sTxt, "descr: ") - 1) sTxt = WorksheetFunction.Clean(sTxt) Else sTxt = "Noch frei" End If If sTxt = "" Then sTxt = "Noch frei" Cells(iRow, 2) = sTxt Next iRow MsgBox "Bin fertig!" End Sub Function MessageLaden(URL As String) As String Dim IEApp As Object Dim IEDocument As Object Dim i As Long Set IEApp = CreateObject("InternetExplorer.Application") IEApp.Visible = False IEApp.Navigate URL Do: Loop Until IEApp.Busy = False Do: Loop Until IEApp.Busy = False Set IEDocument = IEApp.Document MessageLaden = IEDocument.Body.innerText IEApp.Quit Set IEDocument = Nothing Set IEApp = Nothing End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datei zum FTP-Server up- und vom FTP-Server downloaden

Problem:	Eine Excel-Datei mit vorgegebenem Namen soll zum WWW-Server hochund vom Server heruntergeladen werden. Nach dem Download ist die Datei zu öffnen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Sub DownIndexXLS() Dim iFile As Integer Dim sLog As String iFile = FreeFile sLog = Left(Range("B1").Value, 3) & "LogIn.txt" Open sLog For Output As iFile Print #iFile, Range("B3").Value Print #iFile, Range("B4").Value Print #iFile, "cd herber" Print #iFile, "cd bbs" Print #iFile, "cd texte" Print #iFile, "binary" Print #iFile, "get " & Range("B2").Value Print #iFile, "quit" Close iFile Call Win32WaitTilFinished(_ "ftp -s:" & sLog & " " & Range("B5").Value) Workbooks.Open Range("B1").Value & "\" & Range("B2").Value End Sub Sub UpIndexXLS() Dim iFile As Integer Dim sLog As String, sFile As String iFile = FreeFile sLog = Left(Range("B1").Value, 3) & "LogIn.txt" Open sLog For Output As #iFile Print #iFile, Range("B3").Value Print #iFile, Range("B4").Value Print #iFile, "cd herber" Print #iFile, "cd bbs" Print #iFile, "cd texte" Print #iFile, "binary" Print #iFile, "put " & Range("B1").Value & "\" & Range("B2").Value Print #iFile, "quit" Close Shell "ftp -s:" & sLog & " " & Range("B5").Value, vbNormalFocus MsgBox "Daten wurden hochgeladen!" End Sub StandardModule: basFunctions Public Const PROCESS_QUERY_INFORMATION = &H400 Public Const WAIT_TIMEOUT = &H102& Declare Function OpenProcess Lib "kernel32" (ByVal dwDesiredAccess As Long, _ ByVal bInheritHandle As Long, ByVal dwProcessId As Long) As Long Declare Function WaitForSingleObject Lib "kernel32" (ByVal hHandle As Long, _ ByVal dwMilliseconds As Long) As Long Sub Win32WaitTilFinished(ProgEXE As String) Dim ProcessID As Long Dim hProcess As Long Dim RetVal As Long ProcessID = Shell(ProgEXE, vbHide) hProcess = OpenProcess(PROCESS_QUERY_INFORMATION, False, ProcessID) Do DoEvents RetVal = WaitForSingleObject(hProcess, 50) Loop Until RetVal <> WAIT_TIMEOUT End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriffe aus Tabelle im Excel-WebArchiv suchen

Problem: Ein Suchbegriff soll auf Herbers Excel-Server gesucht und die Fundstellen sollen im Browser angezeigt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub SuchString()
    Dim sSearch As String
    sSearch = "http://www.herber.de/cgi-bin/searchstruct.pl?begriff="
    sSearch = sSearch & Range("A1").Value
    ActiveWorkbook.FollowHyperlink sSearch
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Internetdaten abrufen und in MsgBox anzeigen

Problem: Daten aus dem Web sollen im Hintergrund über den Internet-Explorer abgerufen, in Excel übernommen und in einer MsgBox angezeigt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub MessageAbrufen()
    MsgBox MessageLaden("www.herber.de/forum/archiv/24to28/t26542.htm")
End Sub

Function MessageLaden(URL As String) As String
    Dim IEApp As Object
    Dim IEDocument As Object
    Dim i As Long
    Set IEApp = CreateObject("InternetExplorer.Application")
    IEApp.Visible = False
    IEApp.Navigate URL
    Do: Loop Until IEApp.Busy = False
    Do: Loop Until IEApp.Busy = False
    Set IEDocument = IEApp.Document
    MessageLaden = IEDocument.Body.innerText
    IEApp.Quit
    Set IEDocument = Nothing
    Set IEApp = Nothing
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktuellen Wechselkurs abrufen

Problem:	Wie kann ich den aktuellen Wechselkurs aus dem Internet abrufen?
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: basMain Sub WechselKurse() Range("B4").Value = Date Range("B5").Value = Time Worksheets("Tabelle2").QueryTables(1).Refresh End Sub Sub EinAusblenden() Worksheets("Tabelle2").Visible = True 'Worksheets("Tabelle2").Visible = xlVeryHidden End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblatt-Hyperlink über UserForm aufrufen

Problem:	Wie kann ich über eine UserForm-Schaltfläche einen Hyperlink aus dem aktiven Arbeitsblatt und die damit verbundene Datei aufrufen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: frmHyperlink Private Sub cmdHyperlink_Click() Unload Me If Dir(Range("A1").Hyperlinks(1).Address) = "" Then Beep MsgBox "Hyperlink-Adresse wurde nicht gefunden!" Else Range("A1").Hyperlinks(1).Follow End If End Sub Private Sub cmdWeiter_Click() Unload Me End Sub StandardModule: basMain Sub CallForm() frmHyperlink.Show End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auf UserForm-CommandButton-Klick eine Internetseite aufrufen

Problem:	Wie kann ich durch das Klicken auf einen CommandButton in einer UserForm eine Internetseite anwählen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CommandButton1_Click() frmInternet.Show End Sub StandardModule: basMain Sub CallForm() frmInternet.Show End Sub ClassModule: frmInternet Private Sub cmdInternet_Click() ActiveWorkbook.FollowHyperlink _ Address:="http://www.herber.de", _ NewWindow:=True Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Web-Abfrage ohne *.iqy-Datei

Problem: Wie kann ich eine Web-Abfrage starten, ohne vorher eine Abfrage(*.iqy-)Datei anzulegen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CommandButton1_Click()
    Call Aufruf
End Sub

StandardModule: basMain

Sub Aufruf()
    Dim tbl As QueryTable
    Dim sSearch As String
    sSearch = "http://de.finance.yahoo.com/q?s=NSUG.DE&d=t"
    Worksheets.Add after:=Worksheets(Worksheets.Count)
    Set tbl = ActiveSheet.QueryTables.Add( _
        Connection:="URL;" & sSearch, _
        Destination:=Range("A1"))
    With tbl
        .FieldNames = False
        .RefreshOnFileOpen = False
        .Refresh BackgroundQuery:=False
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalt als Internet-Suchbegriff

Problem:	Wie kann ich in XL8 über einen Internet-Hyperlink eine Suchmaschine aufrufen und einen in der aktiven Zelle stehenden begriff suchen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CommandButton1_Click() Call WebSuche End Sub StandardModule: basMain Sub WebSuche() Dim sSearch As String, sWorth As String Application.ScreenUpdating = False sSearch = "http://www.google.de/search?q=" sSearch = sSearch & Range("A1").Value sSearch = sSearch & "&ie=UTF-8&oe=UTF-8&hl=de&btnG=Google-" sSearch = sSearch & "Suche&meta=lr%3Dlang_de" With Range("A1") sWorth = .Value .Value = sSearch .Hyperlinks.Add Range("A1"), sSearch .Hyperlinks(1).Follow NewWindow:=False, AddHistory:=True .Hyperlinks(1).Delete .ClearContents .Value = sWorth End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Import aus dem Internet

Problem: Wie kann ich eine HTML-Seite aus dem Internet öffnen, einen Zelleninhalt herausgreifen und diesen in das aktive Arbeitsblatt übertragen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub HTML()
    Dim rng As Range
    Dim sPage As String, sTxt As String
    Application.ScreenUpdating = False
    sPage = "http://www.herber.de/mailling/extern.htm"
    Workbooks.Open Filename:=sPage
    Set rng = Columns(6).Find( _
        what:="232", _
        LookIn:=xlValues, _
        lookat:=xlPart)
    ThisWorkbook.Worksheets("Tabelle1").Range("A1").Value = _
        rng.Offset(0, 1).Value
    ActiveWorkbook.Close savechanges:=False
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Web-Seiten mit Excel95 öffnen

Problem: Ab Excel8 können Hyperlinks in Tabellenblätter eingefügt werden. Gibt es eine vergleichbare Möglichkeit auch bei Excel7?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub WebSite()
    Dim var As Variant
    var = Shell("Start " & Range("A1").Value)
    Application.WindowState = xlMinimized
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick Textdateien in Editor öffnen

Problem:	Bei Doppelklick auf A1 sollen 3 Dateien mit der in Zelle D3 genannten Suffix aus dem in Zelle D1 genannten Verzeichnis mit der in Zelle D2 genannten Anwendung geöffnet werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean) Dim sFile As String, sPath As String, sTxt As String, sExe As String, sSfx As String If Target.Address <> "\$A\$1" Then Exit Sub Cancel = True sPath = Range("D1").Value sExe = Range("D2").Value sSfx = Range("D3").Value sFile = Range("A1").Value sFile = WorksheetFunction.Substitute(sFile, " ", "") Do While InStr(sFile, ",") sTxt = sPath & "\" & Left(sFile, InStr(sFile, ",") - 1) & "." & sSfx If Dir(sTxt) <> "" Then Shell sExe & " " & sTxt, vbNormalFocus sFile = Right(sFile, Len(sFile) - InStr(sFile, ",")) Loop sTxt = sPath & "\" & sFile & "." & sSfx If Dir(sTxt) <> "" Then Shell sExe & " " & sTxt, vbNormalNoFocus End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text in NotePad schreiben, speichern, in Excel aufrufen

Problem:	Wie kann ich aus Excel heraus NotePad aufrufen, einen Text schreiben, diesen speichern, Notepad schliessen und die Textdatei in Excel laden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Public Const PROCESS_QUERY_INFORMATION = &H400 Public Const WAIT_TIMEOUT = &H102& Declare Function OpenProcess Lib "kernel32" (ByVal dwDesiredAccess As Long, _ ByVal bInheritHandle As Long, ByVal dwProcessId As Long) As Long Declare Function WaitForSingleObject Lib "kernel32" (ByVal hHandle As Long, _ ByVal dwMilliseconds As Long) As Long Sub Aufruf() Dim sFile As String sFile = Application.Path & "\testtext.txt" If Dir(sFile) <> "" Then Kill sFile SendKeys "Hallo Hubert!%du" & sFile & "%s", False SendKeys "%de" WartenBisFertig "NotePad" Workbooks.OpenText sFile MsgBox "Weiter" ActiveWorkbook.Close Kill sFile End Sub Sub WartenBisFertig(strEXE As String) Dim ProcessID As Long Dim hProcess As Long Dim RetVal As Long ProcessID = Shell(strEXE, vbNormalFocus) hProcess = OpenProcess(PROCESS_QUERY_INFORMATION, False, ProcessID) Do DoEvents RetVal = WaitForSingleObject(hProcess, 50) Loop Until RetVal <> WAIT_TIMEOUT End Sub</pre>



Trainingsplan in einen Outlook-Kalender einlesen

Problem:	Der Trainingsplan wird in einen neu angelegten Ordner "Training" im Outlook-Kalender übertragen. In einer zweiten Prozedur wird der Kalender ausgelesen.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn zwei Schaltflächen zu.
VBA-Code:	StandardModule: Modul1 Sub WriteCalendar() Dim olApp As Outlook.Application Dim olNS As Outlook.NameSpace Dim olCal As Outlook.MAPIFolder Dim olApt As AppointmentItem Dim iRow As Integer Set olApp = CreateObject("Outlook.Application") Set olNS = olApp.GetNamespace("MAPI") On Error Resume Next Set olCal = olNS.GetDefaultFolder(olFolderCalendar).Folders("Training") If Err Then Set olCal = olNS.GetDefaultFolder(olFolderCalendar).Folders.Add("Training") Err.Clear End If iRow = 4 Do Until IsEmpty(Cells(iRow, 1)) Set olApt = olApp.CreateItem(olAppointmentItem) With olApt .Start = Cells(iRow, 1).Value + Cells(iRow, 3).Value .End = Cells(iRow, 1).Value + Cells(iRow, 4).Value .Subject = "Trainingstag" .Location = Cells(iRow, 5).Value .Body = Cells(iRow, 6).Value & " mitbringen" .BusyStatus = olBusy .ReminderMinutesBeforeStart = 120 .ReminderSet = True .Save .Move olCal End With iRow = iRow + 1 Loop ERRORHANDLER: Set olApt = Nothing Set olCal = Nothing Set olNS = Nothing Set olApp = Nothing End Sub Sub ReadCalendar() Dim olApp As Outlook.Application Dim olNS As Outlook.NameSpace Dim olCal As Outlook.MAPIFolder Dim olApt As AppointmentItem Dim dat As Date Dim iRow As Integer Dim sTxt As String Application.ScreenUpdating = False On Error GoTo ERRORHANDLER Workbooks.Add 1 Set olApp = CreateObject("Outlook.Application") Set olNS = olApp.GetNamespace("MAPI") Set olCal = olNS.GetDefaultFolder(olFolderCalendar).Folders("Training") ThisWorkbook.Worksheets(1).Range("A1:F3").Copy Range("A1") iRow = 4 For Each olApt In olCal.Items dat = olApt.Start Cells(iRow, 1).Value = Fix(dat) Cells(iRow, 2).Value = Format(Fix(dat), "ddd") Cells(iRow, 3).Value = CDb1(dat) - Fix(dat) dat = olApt.End Cells(iRow, 4).Value = CDb1(dat) - Fix(dat) Cells(iRow, 5).Value = olApt.Location sTxt = olApt.Body Cells(iRow, 6).Value = Left(sTxt, InStr(sTxt, " ") - 1) iRow = iRow + 1 Next olApt Columns("A").NumberFormat = "dd.mm.yy" Columns("C:D").NumberFormat = "hh:mm" Columns.AutoFit ERRORHANDLER: Application.ScreenUpdating = True Set olApt = Nothing Set olCal = Nothing Set olNS = Nothing Set olApp = Nothing End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappen gem. Liste an Email-Adressen versenden

Problem: An die in Spalte B genannten Emailadressen sollen die Arbeitsmappen aus Spalte A mit dem Standardmailprogramm versandt werden.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SendWkbs()
    Dim wks As Worksheet
    Dim iRowA As Integer, iRowB As Integer
    Application.ScreenUpdating = False
    Set wks = ActiveSheet
    iRowA = 2
    Do Until IsEmpty(wks.Cells(iRowA, 1))
        iRowB = 2
        Do Until IsEmpty(wks.Cells(iRowB, 2))
            Workbooks.Open wks.Cells(iRowA, 1).Value, False, True
            ActiveWorkbook.SendMail wks.Cells(iRowB, 2).Value, Subject:="Test"
            ActiveWorkbook.Close savechanges:=False
            iRowB = iRowB + 1
        Loop
        iRowA = iRowA + 1
    Loop
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblatt an variable Email-Adresse per Outlook versenden

Problem: Das in Zelle C1 genannte Arbeitsblatt soll mit dem in Zelle B1 genannten Betreff an die in Zelle A1 genannte Emailadresse gesandt werden.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SendTab()
    Dim wks As Worksheet
    Application.ScreenUpdating = False
    Set wks = ActiveSheet
    Worksheets(Range("C1").Value).Copy
    ActiveWorkbook.SendMail wks.Range("A1").Value, wks.Range("B1").Value
    ActiveWorkbook.Close savechanges:=False
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachauswahl per Outlook versenden

Problem: Eine Mehrfachauswahl aus der nebenstehenden Tabelle soll über Outlook versandt werden. Die Einträge sind mit einem Vorspann untereinander anzuordnen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub SendMessage()  
    Dim oOL As Object  
    Dim oOLMsg As Object  
    Dim oOLRecip As Object  
    Dim rng As Range, oCell As Range  
    Dim sAddress As String, sTxt As String  
    sAddress = "hans@herber.de"  
    sTxt = "Erste Zeile" & vbCrLf & "Zweite Zeile" & vbCrLf  
    For Each rng In Selection.Areas  
        sTxt = sTxt & vbCrLf  
        For Each oCell In rng.Cells  
            sTxt = sTxt & oCell.Text & vbCrLf  
        Next oCell  
    Next rng  
    Set oOL = CreateObject("Outlook.Application")  
    Set oOLMsg = oOL.CreateItem(0)  
    With oOLMsg  
        Set oOLRecip = .Recipients.Add(sAddress)  
        .Subject = "Dies ist ein Outlook-Test"  
        .Body = sTxt  
        .Send  
    End With  
    oOLRecip.Resolve  
    Set oOLRecip = Nothing  
    Set oOLMsg = Nothing  
    Set oOL = Nothing  
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Geschäftstelefon-Nummer in Outlook-Kontaktordner editieren

Problem: Die Geschäftstelefonangabe des in Zelle B2 genannten Email-Kontakts soll editiert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub ContactEdit()
    Dim oOL As Object
    Dim olNS As Object
    Dim fdContacts As Object
    Dim fdItems As Object
    Dim fdItem As Object
    Set oOL = CreateObject("Outlook.Application")
    Set olNS = oOL.GetNamespace("MAPI")
    Set fdContacts = olNS.GetDefaultFolder(olFolderContacts)
    Set fdItems = fdContacts.Items
    Set fdItem = fdItems(Range("B1").Value)
    On Error GoTo ERRORHANDLER
    fdItem.BusinessTelephoneNumber = Range("B2").Value
    fdItem.Save
ERRORHANDLER:
    Set fdItem = Nothing
    Set fdItems = Nothing
    Set fdContacts = Nothing
    Set olNS = Nothing
    Set oOL = Nothing
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Unterschiedliche Texte an unterschiedliche Email-Adressen

Problem: An die in Spalte A genannten Email-Adressen sollen die Texte aus Spalte C mit dem Betreff aus Spalte B gesandt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SendList()
    Dim oOL As Object
    Dim oOLMsg As Object
    Dim oOLRecip As Object
    Dim iRow As Integer
    Dim sRec As String, sSub As String, sBody As String
    iRow = 2
    Set oOL = CreateObject("Outlook.Application")
    Do Until IsEmpty(Cells(iRow, 1))
        sRec = Cells(iRow, 1).Value
        sSub = Cells(iRow, 2).Value
        Set oOLMsg = oOL.CreateItem(0)
        With oOLMsg
            Set oOLRecip = .Recipients.Add(sRec)
            .Subject = sSub
            .body = Cells(iRow, 3).Value
            .Send
            oOLRecip.Resolve
        End With
        iRow = iRow + 1
    Loop
    Set oOL = Nothing
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten aus Outlook-Adressbuch in Tabelle einlesen

Problem:	In die Spalten A:C sollen Daten aus dem Outlook-Adressbuch eingefügt werden. Das Outlook-Adressbuch wird aufgerufen und die in das Listefeld "Mail-Empfänger" verschobenen Adressen werden ausgelesen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub cmd_ShowAdrBook_Click() Dim objSession As MAPI.Session Dim objRecipients As MAPI.Recipients Dim objRecipient As MAPI.Recipient Dim iRow As Integer Set objSession = CreateObject("MAPI.Session") objSession.Logon Set objRecipients = objSession.AddressBook(_ Recipients:=objRecipients, _ Title:="Wählen Sie den oder die Empfänger") iRow = 1 If Not objRecipients Is Nothing Then For Each objRecipient In objRecipients iRow = iRow + 1 Cells(iRow, 1).Value = objRecipient.Name Cells(iRow, 2).Value = objRecipient.Address Cells(iRow, 3).Value = objRecipient.Type Next objRecipient End If ActiveWorkbook.Save objSession.Logoff End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Rundschreiben mit vorgegebenem Text an Verteiler senden

Problem:	Ein Rundschreiben mit dem Text aus Blatt MyText soll an den in Spalte A genannten Verteiler gesandt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Sample() Dim olApp As Object Dim olMailItm As Object Dim iCounter As Integer, iRow As Integer Dim Dest As Variant Dim SDest As String, txt As String, sSubject As String sSubject = Range("Betreff").Value Set olApp = CreateObject("Outlook.Application") Set olMailItm = olApp.CreateItem(0) With Worksheets("MyText") iRow = .Cells(Rows.Count, 1).End(xlUp).Row For iCounter = 1 To iRow If Not IsEmpty(.Cells(iCounter, 1)) Then txt = txt & .Cells(iCounter, 1).Text & vbNewLine Else txt = txt & vbNewLine End If Next iCounter End With With olMailItm SDest = "" For iCounter = 1 To _ Range("A1").End(xlDown).Row If SDest = "" Then SDest = Cells(iCounter, 1).Value Else SDest = SDest & ";" & Cells(iCounter, 1).Value End If Next iCounter .BCC = SDest .Subject = sSubject .Body = txt .Send End With Set olMailItm = Nothing Set olApp = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Titel und Anmerkungen aus Outlook-Adressbuch auslesen

Problem: Die benutzerdefinierten Funktionen in den Spalten B und C sollen Titel und Anmerkungen zum Outlooknamen aus dem Outlook-Adressbuch ermitteln.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: Modul1

```
Function GetOL(sContact As String, iContact As Integer)
    Dim oOl As Object
    Dim oNsp As Object
    Dim oFolder As Object
    Dim oContact As Object
    Set oOl = CreateObject("Outlook.Application")
    Set oNsp = oOl.GetNamespace("MAPI")
    Set oFolder = oNsp.Folders("Persönliche Ordner").Folders("Kontakte")
    Set oContact = oFolder.Items(sContact)
    Select Case iContact
        Case 1: GetOL = oContact.Title
        Case 2: GetOL = oContact.Body
    End Select
    Set oContact = Nothing
    Set oFolder = Nothing
    Set oNsp = Nothing
    Set oOl = Nothing
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nachricht versenden und zur Nachverfolgung kennzeichnen

Problem: An die in Zelle B1 genannte Adresse soll eine Email mit Anlage versandt werden, wobei die Nachricht zu Nachverfolgung gekennzeichnet werden soll.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SendAndSetFlag()
    Dim oOL As Object
    Dim oOLMsg As Object
    Dim oOLRecip As Object
    Dim oOLAttach As Object
    Dim sFile As String, sRec As String, sSub As String
    Dim sBody As String
    Set oOL = CreateObject("Outlook.Application")
    sRec = Range("B1").Value
    sFile = Range("B2").Value
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Datei wurde nicht gefunden!"
        Exit Sub
    End If
    sSub = "Textdatei"
    Set oOLMsg = oOL.CreateItem(0)
    With oOLMsg
        Set oOLRecip = .Recipients.Add(sRec)
        .Subject = sSub
        Set oOLAttach = .Attachments.Add(sFile)
        .FlagStatus = olFlagMarked
        .Send
        oOLRecip.Resolve
    End With
    Set oOL = Nothing
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kommentar in Outlook-Kalender eintragen

Problem: Im Outlook-Kalender soll unter erstem Eintrag zu dem in Zelle genannten Datum der in Zelle B2 genannte Kommentar eingetragen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SetOLComment()
    Dim olApp As Outlook.Application
    Dim olNS As Outlook.Namespace
    Dim olAI As Outlook.AppointmentItem
    Dim olFolder As Outlook.MAPIFolder
    Set olApp = New Outlook.Application
    Set olNS = olApp.GetNamespace("MAPI")
    Set olFolder = olNS.GetDefaultFolder(olFolderCalendar)
    For Each olAI In olFolder.Items
        If Format(olAI.Start, "dd.mm.yyyy") = Format(Range("B1").Value, "dd.mm.yyyy") Then
            olAI.Body = Range("B2").Value
            olAI.Save
        End If
    Next olAI
    Set olAI = Nothing
    Set olNS = Nothing
    Set olApp = Nothing
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausgewählten Bereich als interaktive Tabelle versenden

Problem:	Der ausgewählte Bereich der Tabelle soll als interaktive Tabelle per Outlook versandt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Sel2Mail() Dim ol As Object Dim oMail As Object Dim oFs As Object Dim oFile As Object Dim oTxtStrm As Object Dim sSubject As String, sBody As String, sRecipient As String Dim sPath As String sPath = Application.DefaultFilePath & "\html.htm" If ActiveWorkbook.PublishObjects.Count = 1 Then ActiveWorkbook.PublishObjects(1).Delete End If ActiveWorkbook.PublishObjects.Add _ SourceType:=xlSourceRange, _ Filename:=sPath, _ Sheet:=ActiveSheet.Name, _ Source:=Selection.Address, _ HtmlType:=xlHtmlCalc ActiveWorkbook.PublishObjects(1).Publish create:=True Set oFs = CreateObject("Scripting.FileSystemObject") Set oFile = oFs.GetFile(sPath) Set oTxtStrm = oFile.OpenAsTextStream sBody = oTxtStrm.ReadAll Set ol = CreateObject("Outlook.Application") Set oMail = ol.CreateItem(0) oMail.Save oMail.To = Worksheets("Data").Range("B1").Value oMail.subject = Worksheets("Data").Range("B2").Value oMail.HTMLBody = sBody oMail.Importance = 1 oMail.Sensitivity = 1 oMail.Send Set ol = Nothing Set oMail = Nothing Set oFs = Nothing Set oFile = Nothing Set oTxtStrm = Nothing Kill sPath End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl der Mails in einem vorgegebenem Outlook-Ordner ermitteln

Problem: Aus dem Outlook-Pfad "Persönliche Ordner/ Posteingang" soll die Anzahl der Emails ermittelt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SendMessage()
    MsgBox "Vorhandene Mails: " & GrapText
End Sub

Function GrapText() As Long
    Dim objOutlook As Object
    Dim objnSpace As Object
    Dim objFolder As Object
    Set objOutlook = CreateObject("Outlook.Application")
    Set objnSpace = objOutlook.GetNamespace("MAPI")
    Set objFolder = objnSpace.Folders("Persönliche Ordner").Folders("Posteingang")
    GrapText = objFolder.Items.Count
    Set objFolder = Nothing
    Set objnSpace = Nothing
    Set objOutlook = Nothing
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfen, ob ein Name im Outlook-Adressbuch existiert

Problem:	Im Outlook-Adressbuch soll der in Zelle B1 eingetragene Name gesucht und das Ergebnis der Suche in einer MessageBox angezeigt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub NameExists() Dim oOL As Object Dim oAdr As Object Dim oEntry As Object Dim iCounter As Integer Dim bln As Boolean bln = Application.DisplayStatusBar Application.DisplayStatusBar = True Set oOL = CreateObject("Outlook.Application") Set oAdr = oOL.Session.AddressLists("Kontakte") For Each oEntry In oAdr.AddressEntries If iCounter Mod 10 = 0 Then Application.StatusBar = _ "Prüfe Adresse Nr. " & iCounter & "..." End If If Range("B1").Value = oEntry.Name Then MsgBox "Adresse wurden gefunden!" GoTo FINISH End If iCounter = iCounter + 1 Next oEntry MsgBox "Name wurde nicht gefunden!" FINISH: Set oOL = Nothing Application.StatusBar = False Application.DisplayStatusBar = bln End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mails eines Outlook-Ordners in einen anderen verschieben

Problem:	Die Mails eines Outlook-Ordners sollen in einen anderen Ordner verschoben werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub MoveMsgs() Dim oOutlook As Object Dim oNSpace As Object Dim oFolderA As Object Dim oFolderB As Object Dim oMsg As Object Dim iCounter As Integer, iCount As Integer Set oOutlook = CreateObject("Outlook.Application") Set oNSpace = oOutlook.GetNamespace("MAPI") Set oFolderA = oNSpace.folders("Persönliche Ordner") _ .folders("Gelöschte Objekte") Set oFolderB = oNSpace.folders("Persönliche Ordner") _ .folders("Temp") iCount = oFolderA.Items.Count If iCount > 0 Then For iCounter = 1 To iCount Set oMsg = oFolderA.Items(1) oMsg.Move oFolderB Next iCounter End If Set oNSpace = Nothing Set oFolderA = Nothing Set oFolderB = Nothing Set oMsg = Nothing Set oOutlook = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einem Outlook-Mailkontakt eine Anmerkung hinzufügen

Problem:	Einem Outlook-Mailkontakt soll eine Anmerkung hinzugefügt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub xl2ol() Dim oOl As Outlook.Application Dim oNsp As Outlook.NameSpace Dim oFolder As Outlook.MAPIFolder Dim oContact As Outlook.ContactItem Set oOl = CreateObject("Outlook.Application") Set oNsp = Outlook.GetNamespace("MAPI") Set oFolder = oNsp.Folders("Persönliche Ordner").Folders("Kontakte") Set oContact = oFolder.Items("Hans W. Herber") oContact.Body = "Hallo" oContact.Save Set oContact = Nothing Set oFolder = Nothing Set oNsp = Nothing Set oOl = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Outlook-Nachrichten in Excel-Tabellen einlesen

Problem: Die Nachrichten eines Outlook-Ordners sollen in hinzuzufügende Tabellenblätter eingelesen werden. Der Outlook-Pfad muss angepasst werden.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code: StandardModule: Modul1

```
Sub GrapText()  
    Dim objOutlook As Object  
    Dim objnSpace As Object  
    Dim objFolder As Object  
    Dim objMsg As Object  
    Dim intCounter As Integer, intCount As Integer, iRow As Integer  
    Dim sTxt As String  
    Application.ScreenUpdating = False  
    Set objOutlook = CreateObject("Outlook.Application")  
    Set objnSpace = objOutlook.GetNamespace("MAPI")  
    Set objFolder = objnSpace.folders("Persönliche Ordner").folders("Dank")  
    intCount = objFolder.Items.Count  
    If intCount > 0 Then  
        For intCounter = 1 To intCount  
            Set objMsg = objFolder.Items(intCounter)  
            Worksheets.Add after:=Worksheets(Worksheets.Count)  
            objMsg.SaveAs ThisWorkbook.Path & "\temp.txt", olTXT  
            Close  
            iRow = 0  
            Open ThisWorkbook.Path & "\temp.txt" For Input As #1  
            Do Until EOF(1)  
                iRow = iRow + 1  
                Line Input #1, sTxt  
                Cells(iRow, 1).Value = "" & sTxt  
            Loop  
            Close  
        Next intCounter  
        Kill ThisWorkbook.Path & "\temp.txt"  
    End If  
    Set objnSpace = Nothing  
    Set objFolder = Nothing  
    Set objMsg = Nothing  
    Set objOutlook = Nothing  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einlesen des Outlook-Adressbuches in eine UserForm-ListBox

Problem:	Wie kann ich das Outlook-Adressbuch in eine UserForm-ListBox einlesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DialogAufruf() frmOutlook.Show End Sub ClassModule: frmOutlook Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub UserForm_Initialize() Dim oOL As Object Dim oAdr As Object Dim oEntry As Object Dim arr() As String Dim iCounter As Integer Dim bln As Boolean bln = Application.DisplayStatusBar Application.DisplayStatusBar = True Set oOL = CreateObject("Outlook.Application") Set oAdr = oOL.Session.AddressLists("Kontakte") For Each oEntry In oAdr.AddressEntries If iCounter Mod 10 = 0 Then Application.StatusBar = _ "Lese Adresse Nr. " & iCounter & " ein..." End If ReDim Preserve arr(1 To 2, iCounter) arr(1, iCounter) = oEntry.Name arr(2, iCounter) = oEntry.Address iCounter = iCounter + 1 Next oEntry lstAdressen.Column = arr Set oOL = Nothing Application.StatusBar = False Application.DisplayStatusBar = bln End Sub</pre>



Diagramm zu PowerPoint-Präsentation konvertieren und aufrufen

Problem:	Auf Grundlage des Diagramms soll eine PowerPoint-Präsentation erstellt und aufgerufen werden. In der Entwicklungsumgebung muss ein Verweis zur Microsoft PowerPointObjektbibliothek hergestellt sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: Modul1 Sub CreatePPT() Dim ppt As Object Dim oPPT As PowerPoint.Application Dim oPrs As PowerPoint.Presentation Dim oSld As PowerPoint.Slide Dim oPct As PowerPoint.Shape Dim oTxt As PowerPoint.Shape Dim sPath As String sPath = Application.DefaultFilePath & "/xl2ppt.ppt" Set oPPT = CreateObject("PowerPoint.Application") Set oPrs = oPPT.Presentations.Add(msoCTrue) Set oSld = oPrs.Slides.Add(1, ppLayoutTitleOnly) oSld.Shapes.Title.TextFrame.TextRange.Text = "Diagramm-Test" ActiveSheet.ChartObjects(1).Copy Set oTxt = oSld.Shapes(1) With oTxt .Left = 50 .Top = 0 .Width = 650 .Height = 50 With .TextFrame With .TextRange With .Font .Name = "Arial" .Size = 24 .Bold = msoCTrue End With End With .AutoSize = ppAutoSizeShapeToFitText End With End With With oTxt.AnimationSettings .Animate = msoTrue .EntryEffect = ppEffectBoxIn .TextLevelEffect = ppAnimateByAllLevels .AnimateBackground = msoTrue .TextUnitEffect = ppAnimateByCharacter .AdvanceMode = ppAdvanceOnTime End With Set oPct = oSld.Shapes.Paste(1) oPrs.SaveAs sPath oPrs.Close Set oPct = Nothing Set oTxt = Nothing Set oSld = Nothing Set oPrs = Nothing Set oPPT = Nothing Application.Wait Now + TimeSerial(0, 0, 3) Set ppt = GetObject(sPath) ppt.SlideShowSettings.Run End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

PowerPoint mit SlideShow starten und danach schließen

Problem:	PowerPoint soll mit einer SlideShow gestartet und nach Beendigung der SlideShow wieder geschlossen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub PowerPointStarten() Dim ppApp As Object Dim ppP As Object Dim sFile As String sFile = ThisWorkbook.Path & "\3ulrich.ppt" If Dir(sFile) = "" Then Beep MsgBox "Die Datei " & sFile & " existiert nicht!" Exit Sub End If Set ppApp = CreateObject("PowerPoint.Application") ppApp.Visible = msoTrue Set ppP = ppApp.Presentations.Open(sFile) ppP.SlideShowSettings.Run ppApp.Quit Set ppP = Nothing Set ppApp = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen PowerPoint starten und danach Mappe aktivieren

Problem:	Beim Laden der Arbeitsmappe soll Powerpoint gestartet und nach dessen Beendigung die Mappe wieder aktiviert werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Call Win32WaitTilFinished("powerpnt.exe") End Sub StandardModule: basMain Public Const PROCESS_QUERY_INFORMATION = &H400 Public Const WAIT_TIMEOUT = &H102& Declare Function OpenProcess Lib "kernel32" (ByVal dwDesiredAccess As Long, _ ByVal bInheritHandle As Long, ByVal dwProcessId As Long) As Long Declare Function WaitForSingleObject Lib "kernel32" (ByVal hHandle As Long, _ ByVal dwMilliseconds As Long) As Long Sub Win32WaitTilFinished(ProgEXE As String) Dim ProcessID As Long Dim hProcess As Long Dim RetVal As Long ProcessID = Shell(ProgEXE, vbMaximizedFocus) hProcess = OpenProcess(PROCESS_QUERY_INFORMATION, False, ProcessID) Do DoEvents RetVal = WaitForSingleObject(hProcess, 50) Loop Until RetVal <> WAIT_TIMEOUT End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Währung in der Systemsteuerung / Ländereinstellungen umstellen

Problem: Wie kann ich in der Systemsteuerung bei den Ländereinstellungen die Währung über VBA umstellen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub DialogAufrufen()
    frmWaehrung.Show
End Sub

Sub WaehrungsAufruf(strWaehrung As String)
    Dim strEuro As String
    Application.ScreenUpdating = False
    strWaehrung = strWaehrung & "%b{enter}"
    Shell "RunDll32.exe Shell32.dll,Control_RunDLL intl.cpl,,2"
    SendKeys strWaehrung, True
End Sub

ClassModule: frmWaehrung

Private Sub cmdAbbrechen_Click()
    Unload Me
End Sub

Private Sub cmdEinstellen_Click()
    Dim strWaehrung As String
    If optDM.Value = True Then
        strWaehrung = "DM"
    ElseIf optEuro.Value = True Then
        strWaehrung = ChrW(8364)
    Else
        strWaehrung = "$"
    End If
    Unload Me
    Call WaehrungsAufruf(strWaehrung)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolgen in Textdatei suchen und ersetzen

Problem: Die in Zelle B1 genannte Textdatei soll im Ordner dieses Verzeichnisses gesucht und geöffnet werden. Der Suchbegriff aus Zelle B2 ist durch den Ersatzbegriff aus Zelle B3 zu ersetzen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub SubstituteSave()  
    Dim arr() As String  
    Dim iCounter As Integer  
    Dim sSource As String, sTarget As String, sTxtA As String  
    Dim sTxtB As String, sTxt As String, sPath As String  
    sPath = ThisWorkbook.Path & "\"  
    sSource = sPath & Range("B1").Value  
    sTarget = sPath & Range("B4").Value  
    sTxtA = Range("B2").Value  
    sTxtB = Range("B3").Value  
    Close  
    Open sSource For Input As #1  
    Do Until EOF(1)  
        Line Input #1, sTxt  
        If InStr(sTxt, sTxtA) Then  
            sTxt = Replace(sTxt, sTxtA, sTxtB)  
        End If  
        iCounter = iCounter + 1  
        ReDim Preserve arr(1 To iCounter)  
        arr(iCounter) = sTxt  
    Loop  
    Close  
    Open sTarget For Output As #1  
    For iCounter = 1 To UBound(arr)  
        Print #1, arr(iCounter)  
    Next iCounter  
    Close  
    On Error GoTo ERRORHANDLER  
    Shell "notepad " & sTarget, vbMaximizedFocus  
    Exit Sub  
ERRORHANDLER:  
    MsgBox "Job erledigt!"  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In jeder Zeile einer Textdatei 1. und letztes Zeichen entfernen

Problem: Bei der in Zelle B1 genannten Datei sollen in jeder Datenzeile das 1. und letzte Zeichen entfernt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub EditText()  
    Dim arr() As String  
    Dim lRow As Long  
    Dim sPath As String, sTxt As String  
    sPath = Range("B1").Value  
    If Dir(sPath) = "" Then  
        Beep  
        MsgBox "Datei " & sPath & " wurde nicht gefunden!"  
        Exit Sub  
    End If  
    Close  
    Open sPath For Input As #1  
    Do Until EOF(1)  
        Line Input #1, sTxt  
        sTxt = Right(sTxt, Len(sTxt) - 1)  
        sTxt = Left(sTxt, Len(sTxt) - 1)  
        lRow = lRow + 1  
        ReDim Preserve arr(1 To lRow)  
        arr(lRow) = sTxt  
    Loop  
    Close  
    Open sPath For Output As #1  
    For lRow = 1 To UBound(arr)  
        Print #1, arr(lRow)  
    Next lRow  
    Close  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelt vorkommende Anführungszeichen in Textdatei ersetzen

Problem: Bei der in Zelle B1 genannten Datei sollen doppelt vorkommende Anführungszeichen durch einzelne ersetzt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub EditText()  
    Dim arr() As String  
    Dim lRow As Long  
    Dim sPath As String, sTxt As String  
    sPath = Range("B1").Value  
    If Dir(sPath) = "" Then  
        Beep  
        MsgBox "Datei " & sPath & " wurde nicht gefunden!"  
        Exit Sub  
    End If  
    Close  
    Open sPath For Input As #1  
    Do Until EOF(1)  
        Line Input #1, sTxt  
        sTxt = Replace(sTxt, """""", "")  
        lRow = lRow + 1  
        ReDim Preserve arr(1 To lRow)  
        arr(lRow) = sTxt  
    Loop  
    Close  
    Open sPath For Output As #1  
    For lRow = 1 To UBound(arr)  
        Print #1, arr(lRow)  
    Next lRow  
    Close  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdateien vergleichen und Unterschiede dokumentieren

Problem:	Zwei Dateien mit der gleichen Zeilenanzahl werden Zeile für Zeile miteinander verglichen. Die Zeilen der ersten Datei, die Unterschiede zur jeweiligen Zeile der zweiten Datei aufweisen, werden in eine dritte Datei geschrieben.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Anlegen() Dim iFile As Integer, iRow As Integer, iCol As Integer Dim sFile As String iFile = FreeFile For iCol = 1 To 2 sFile = Application.Path & "\test" & iCol & ".txt" Open sFile For Output As iFile iRow = 1 Do Until IsEmpty(Cells(iRow, iCol)) Print #iFile, Cells(iRow, iCol).Value iRow = iRow + 1 Loop Close iFile Next iCol MsgBox "Die Testdateien wurden angelegt!" End Sub Sub Vergleich() Dim iCounter As Integer, iFile As Integer Dim sPath As String, txtA As String, txtB As String sPath = Application.Path & "\" iFile = FreeFile Open sPath & "test1.txt" For Input As iFile Open sPath & "test2.txt" For Input As iFile + 1 Open sPath & "test3.txt" For Output As iFile + 2 Do Until EOF(1) iCounter = iCounter + 1 Line Input #iFile, txtA Line Input #iFile + 1, txtB If txtA <> txtB Then Print #iFile + 2, txtA & " - " & txtB End If Loop Close sPath = sPath & "test3.txt" Workbooks.OpenText _ Filename:=sPath, _ DataType:=xlDelimited, _ tab:=False, _ semicolon:=False, _ comma:=False, _ Space:=False, _ other:=False Columns.AutoFit MsgBox "Weiter" ActiveWorkbook.Close savechanges:=False Kill sPath End Sub</pre>



Zeile in *.ini Datei austauschen

Problem:	Wie kann ich in einer *.ini Datei eine Zeile austauschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Sub CreateFile() Dim iFile As Integer, iCounter As Integer Dim sFile As String, sTxt As String iFile = FreeFile sFile = Application.Path & "\textttest.txt" Open sFile For Output As iFile For iCounter = 1 To 10 Print #iFile, "Zeile" & iCounter Next iCounter Close iFile Workbooks.OpenText _ Filename:=sFile, _ DataType:=xlDelimited, _ tab:=False, _ semicolon:=False, _ comma:=False, _ Space:=False, _ other:=False MsgBox "Weiter" ActiveWorkbook.Close savechanges:=False End Sub Sub ChangeFile() Dim iFile As Integer, iCounter As Integer Dim sFileA As String, sFileB As String, sTxt As String iFile = FreeFile sFileA = Application.Path & "\textttest.txt" sFileB = Application.Path & "\temp.txt" Open sFileA For Input As iFile Open sFileB For Output As iFile + 1 Do Until EOF(iFile) Line Input #iFile, sTxt iCounter = iCounter + 1 If iCounter = 5 Then Print #iFile + 1, "Fünfte Zeile" Else Print #iFile + 1, sTxt End If Loop Close Kill sFileA Name sFileB As sFileA Workbooks.OpenText _ Filename:=sFileA, _ DataType:=xlDelimited, _ tab:=False, _ semicolon:=False, _ comma:=False, _ Space:=False, _ other:=False MsgBox "Weiter" ActiveWorkbook.Close savechanges:=False Kill sFileA End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In importierter Textdatei englische Werte in deutsche wandeln

Problem:	In einer importierten Textdatei befinden sich Zahlenwerte in englischer Schreibweise und Minuswerte mit angefügtem Minuszeichen. Wie kann ich diese Werte in deutsche Werte konvertieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub InWerte() Dim rng As Range For Each rng In ActiveSheet.UsedRange.Cells If Not IsEmpty(rng) Then If Right(rng.Text, 1) = "-" Then If IsNumeric(Left(rng.Text, Len(rng.Text) - 1)) Then rng.Value = "-" & Left(rng.Text, Len(rng.Text) - 1) End If End If rng.Formula = rng.FormulaLocal End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Variablen in einer temporären Textdatei zwischenspeichern

Problem: Ich möchte Variablen über das Schließen der Arbeitsmappe hinaus für einen späteren Aufruf deponieren. Wie mache ich das, wenn ich sie nicht in der Arbeitsmappe selbst speichern möchte?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub VariablenStore()
    Dim arr(1 To 10) As String
    Dim iCounter As Integer, iFile As Integer
    Dim sFile As String
    sFile = Application.Path & "\\VarStore.txt"
    For iCounter = 1 To 10
        arr(iCounter) = "Zeile" & iCounter
    Next iCounter
    iFile = FreeFile
    Open sFile For Output As iFile
    For iCounter = 1 To 10
        Print #iFile, arr(iCounter)
    Next iCounter
    Close iFile
    MsgBox "Variablen gespeichert unter:" & vbCrLf & sFile
End Sub

Sub VariablenLesen()
    Dim iFile As Integer
    Dim sFile As String, sTxt As String, sText As String
    iFile = FreeFile
    sFile = Application.Path & "\\VarStore.txt"
    Open sFile For Input As iFile
    Do Until EOF(iFile)
        Line Input #iFile, sTxt
        sText = sText & sTxt & vbCrLf
    Loop
    Close iFile
    MsgBox sText
    Kill sFile
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdateien verbinden

Problem: Wie kann ich mehrere Textdateien, die die Namen "test1.txt", "test2.txt" usw. tragen, miteinander verbinden?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Anlegen()  
    Dim iFile As Integer  
    Dim sFileA As String, sFileB As String  
    sFileA = Application.Path & "\texttestA.txt"  
    sFileB = Application.Path & "\texttestB.txt"  
    iFile = FreeFile  
    Open sFileA For Output As iFile  
    Print #iFile, "Dateiname: " & sFileA  
    Close iFile  
    Open sFileB For Output As iFile  
    Print #iFile, "Dateiname: " & sFileB  
    Close iFile  
    MsgBox "Dateien wurden angelegt!"  
End Sub  
  
Sub Verbinden()  
    Dim iFile  
    Dim sFileA As String, sFileB As String, sTxt As String  
    iFile = FreeFile  
    sFileA = Application.Path & "\texttestA.txt"  
    sFileB = Application.Path & "\texttestB.txt"  
    Open sFileA For Append As iFile  
    Open sFileB For Input As iFile + 1  
    Do Until EOF(iFile + 1)  
        Line Input #iFile + 1, sTxt  
        Print #iFile, sTxt  
    Loop  
    Close  
    Workbooks.OpenText _  
        Filename:=sFileA, _  
        DataType:=xlDelimited, _  
        tab:=False, _  
        semicolon:=False, _  
        comma:=False, _  
        Space:=False, _  
        other:=False  
    MsgBox "Weiter"  
    ActiveWorkbook.Close savechanges:=False  
    Kill sFileA  
    Kill sFileB  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einige Zeilen in einer sequentiellen Datei ersetzen

Problem: Wie kann ich mehrere Zeilen einer sequentiellen Textdatei durch Zellinhalte eines Excel-Arbeitsblattes ersetzen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub CreateFile()  
    Dim rng As Range  
    Dim iFile As Integer, iRow As Integer, iCol As Integer  
    Dim sFile As String, sTxt As String  
    sFile = Application.Path & "\textttest.txt"  
    iFile = FreeFile  
    Open sFile For Output As iFile  
    For iRow = 1 To 10  
        For iCol = 1 To 5  
            sTxt = sTxt & "Zeile" & iRow & " - Spalte" & iCol & ","  
        Next iCol  
        sTxt = Left(sTxt, Len(sTxt) - 1)  
        Print #iFile, sTxt  
        sTxt = ""  
    Next iRow  
    Close  
    Workbooks.OpenText _  
        Filename:=sFile, _  
        DataType:=xlDelimited, _  
        tab:=False, _  
        semicolon:=False, _  
        comma:=True, _  
        Space:=False, _  
        other:=False  
    Columns.AutoFit  
    MsgBox "Weiter"  
    For Each rng In ThisWorkbook.Worksheets("Tabelle1") _  
        .Range("A1:E10").Cells  
        If Not IsEmpty(rng) Then  
            Range(rng.Address).Value = rng.Value  
        End If  
    Next rng  
    MsgBox "Weiter"  
    ActiveWorkbook.Close savechanges:=False  
    Kill sFile  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die Zeilen mit Wert in Spalte C als CSV-Datei speichern

Problem: Nur die Zeilen, die einen Wert in Spalte C beinhalten, sollen in einer Semikolonseparierten CSV-Datei gespeichert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SaveAsCSV()
    Dim iRow As Integer, iCol As Integer, iRowL As Integer, iColL As Integer
    Dim sTxt As String, sSep As String, sFile As String
    iRowL = Cells(Rows.Count, 1).End(xlUp).Row
    iColL = Cells(1, 256).End(xlToLeft).Column
    sFile = Application.DefaultFilePath & "\test_csv.csv"
    sSep = ";"
    Close
    Open sFile For Output As #1
    For iRow = 1 To iRowL
        If Not IsEmpty(Cells(iRow, 3)) Then
            For iCol = 1 To iColL
                sTxt = sTxt & Cells(iRow, iCol).Value & sSep
            Next iCol
            sTxt = Left(sTxt, Len(sTxt) - 1)
            Print #1, sTxt
            sTxt = ""
        End If
    Next iRow
    Close
    MsgBox "Datei wurde angelegt:" & vbLf & sFile
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdatei in Unicode-Text konvertieren

Problem:	Die in Zeile B1 genannte Datei soll in eine Unicode-Datei konvertiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SaveAsUniCode() Dim arr() As String Dim iCounter As Integer Dim sText As String, sPath As String sPath = Range("B1").Value Close Open sPath For Input As #1 Do Until EOF(1) Line Input #1, sText iCounter = iCounter + 1 ReDim Preserve arr(1 To iCounter) arr(iCounter) = StrConv(sText, vbUnicode) Loop Close Open sPath For Output As #1 For iCounter = 1 To UBound(arr) Print #1, arr(iCounter) Next iCounter Close End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattinhalte in getrennten Textdateien speichern

Problem:	Die Texte aus den 3 Tabellenblättern sollen in 3 Textdateien mit den Tabellenblattnamen exportiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub TextExport() Dim rng As Range Dim iWks As Integer, iRow As Integer, iCol As Integer Dim sTxt As String, sPath As String sPath = Application.DefaultFilePath & "\" For iWks = 1 To Worksheets.Count Open sPath & Worksheets(iWks).Name & ".txt" For Output As #1 Set rng = Worksheets(iWks).Range("A1").CurrentRegion For iRow = 1 To rng.Rows.Count For iCol = 1 To rng.Columns.Count sTxt = sTxt & Worksheets(iWks).Cells(iRow, iCol).Value & vbTab Next iCol Print #1, Left(sTxt, Len(sTxt) - 1) sTxt = "" Next iRow Close #1 Next iWks MsgBox "Sie finden die Textdateien im Ordner " & Left(sPath, Len(sPath) - 1) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm-TextBoxes-Texte in Textdateien speichern

Problem: Aus UserForm-TextBoxes sollen Texte in Textdateien gespeichert werden, deren Name in Abhängigkeit einer Optionsfeldauswahl erfolgt.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmRubriken

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim iOption As Integer
    Dim sPath As String, sFile As String
    sPath = Application.DefaultFilePath & "\"
    For iOption = 1 To 4
        If Controls("OptionButton" & iOption) Then
            Exit For
        End If
    Next iOption
    Select Case iOption
        Case 1: sFile = "TextOptionEins.txt"
        Case 2: sFile = "TextOptionZwei.txt"
        Case 3: sFile = "TextOptionDrei.txt"
        Case 4: sFile = "TextOptionVier.txt"
    End Select
    Close
    Open sPath & sFile For Output As #1
    For iOption = 1 To 4
        Print #1, Controls("TextBox" & iOption).Text
    Next iOption
    Close
    MsgBox "Die Datei wurde angelegt:" & vbCrLf & sPath & sFile
    Unload Me
End Sub

StandardModule: Modul1

Sub CallForm()
    frmRubriken.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattdaten einschließlich Kommentare in Textdatei

Problem: Die in der Tabelle vorhandenen Kommentare sollen ebenso wie die Zellinhalte - durch öffnende und schließende Klammern gekennzeichnet - in eine Textdatei übernommen werden.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code:

```
StandardModule: Modul1

Sub Comment2Text()
    Dim cmt As Comment
    Dim rng As Range
    Dim iRow As Integer, iCol As Integer
    Dim sTxt As String
    On Error GoTo ERRORHANDLER
    Set rng = Range("A1").CurrentRegion
    Open Application.Path & "\test.txt" For Output As #1
    For iRow = 1 To rng.Rows.Count
        For iCol = 1 To rng.Columns.Count
            If Not Cells(iRow, iCol).Comment Is Nothing Then
                sTxt = sTxt & Cells(iRow, iCol).Text & _
                    "(" & Cells(iRow, iCol).Comment.Text & ")" & ";"
            Else
                sTxt = sTxt & Cells(iRow, iCol).Text & ";"
            End If
        Next iCol
        sTxt = Left(sTxt, Len(sTxt) - 1)
        Print #1, sTxt
        sTxt = ""
    Next iRow
    Close
    MsgBox "Textdatei wurde erstellt!"
    Exit Sub
ERRORHANDLER:
MsgBox "Textdatei konnte nicht erstellt werden!"
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten in einer Textdatei ohne Anführungszeichen speichern

Problem:	Wenn die Arbeitsmappe manuell gespeichert wird, fügt Excel Anführungszeichen hinzu. Dieses Problem soll umgangen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Export() Dim rng As Range Dim iFile As Integer, iRow As Integer, iCol As Integer Dim sFile As String, sTxt As String Set rng = Range("A1").CurrentRegion sFile = Application.Path & "\\testtext.txt" iFile = FreeFile Open sFile For Output As iFile For iRow = 1 To rng.Rows.Count For iCol = 1 To rng.Columns.Count sTxt = sTxt & Cells(iRow, iCol).Value & vbTab Next iCol sTxt = Left(sTxt, Len(sTxt) - 1) Print #iFile, sTxt sTxt = "" Next iRow Close iFile Workbooks.OpenText _ Filename:=sFile, _ DataType:=xlDelimited, _ tab:=True, _ semicolon:=False, _ comma:=False, _ Space:=False, _ other:=False MsgBox "Weiter" ActiveWorkbook.Close savechanges:=False Kill sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenweises Speichern von Texten

Problem: Die Zellinhalte aus Spalte A sollen jeweils in eine Textdatei gem. Dateinamen aus Spalte B gespeichert werden. Die Pfadangabe steht in Zelle E1.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub XLS2TXT()
    Dim iRow As Integer
    On Error GoTo ERRORHANDLER
    iRow = 1
    Close
    Do Until IsEmpty(Cells(iRow, 1))
        Open Range("E1").Value & "\" & _
            Cells(iRow, 2) & ".txt" For Output As #1
        Print #1, Cells(iRow, 1).Value
        Close
        iRow = iRow + 1
    Loop
    MsgBox "Die Dateien wurden angelegt!"
    Exit Sub
ERRORHANDLER:
    MsgBox "Die Dateien konnten nicht angelegt werden!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattinhalt in sequentielle Textdatei ein- und auslesen

Problem: Wie kann ich den Inhalt der Spalten A:B in eine Textdatei ein- und später wieder auslesen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub IntTextDatei()  
    Dim rng As Range  
    Dim iRow As Integer, iCol As Integer, iFile As Integer  
    Dim sFile As String, sTxt As String  
    Set rng = Range("A1").CurrentRegion  
    sFile = Application.Path & "/testtext.txt"  
    iFile = FreeFile  
    Open sFile For Output As iFile  
    For iRow = 1 To rng.Rows.Count  
        For iCol = 1 To rng.Columns.Count  
            sTxt = sTxt & Cells(iRow, iCol).Value & ","  
        Next iCol  
        sTxt = Left(sTxt, Len(sTxt) - 1)  
        Print #iFile, sTxt  
        sTxt = ""  
    Next iRow  
    Close iFile  
    rng.ClearContents  
    MsgBox "Habe die Daten in eine Textdatei eingelesen!"  
End Sub
```

```
Sub AusTextDatei()  
    Dim iRow As Integer, iFile As Integer  
    Dim sFile As String, sTxtA As String, sTxtB As String  
    sFile = Application.Path & "/testtext.txt"  
    If Dir(sFile) = "" Then  
        MsgBox "Die Daten wurden nicht eingelesen!"  
        Exit Sub  
    End If  
    iFile = FreeFile  
    Open sFile For Input As iFile  
    Do Until EOF(iFile)  
        Input #iFile, sTxtA, sTxtB  
        iRow = iRow + 1  
        Cells(iRow, 1).Value = sTxtA  
        Cells(iRow, 2).Value = sTxtB  
    Loop  
    Close  
    Kill sFile  
    MsgBox "Habe die Daten ausgelesen!"  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenumbrüche in eine *.CSV-Datei übernehmen

Problem: Wie kann ich erzwungene Zeilenumbrüche in einer Zelle in eine CSV-Datei übernehmen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub cmdCSV_Click()
    Dim iFile As Integer
    Dim sFile As String, sDaten As String
    sFile = Application.Path & "\testtext.csv"
    iFile = FreeFile
    sDaten = Range("A2").Value
    Open sFile For Output As iFile
    Do
        Print #iFile, Left(sDaten, InStr(sDaten, vbCrLf) - 1)
        sDaten = Right(sDaten, Len(sDaten) - InStr(sDaten, vbCrLf))
    Loop While InStr(sDaten, vbCrLf) > 0
    Print #iFile, sDaten
    Close iFile
    Workbooks.Open sFile
    MsgBox "Weiter"
    ActiveWorkbook.Close savechanges:=False
    Kill sFile
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Texte mit unterschiedlichen Feldlängen als Textdatei speichern

Problem:	Wie kann ich Zellinhalte korrigieren und dann spaltenabhängig mit unterschiedlichen Feldlängen in eine *.dat-Textdatei speichern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Export() Dim rng As Range Dim iRow As Integer, iCol As Integer, iFile As Integer Dim iLen As Integer Dim sFile As String, sTxt As String, sText As String Set rng = Range("A1").CurrentRegion sFile = Application.Path & "\textttest.txt" iFile = FreeFile Open sFile For Output As iFile For iRow = 2 To rng.Rows.Count For iCol = 1 To rng.Columns.Count iLen = Cells(1, iCol).Value sText = Cells(iRow, iCol).Text If Len(sText) > iLen Then sText = Left(sText, iLen) Else sText = sText & String(iLen - Len(sText), " ") End If sTxt = sTxt & sText & ", " Next iCol sTxt = Left(sTxt, Len(sTxt) - 1) Print #iFile, sTxt sTxt = "" Next iRow Close Workbooks.OpenText _ Filename:=sFile, _ DataType:=xlDelimited, _ tab:=False, _ semicolon:=False, _ comma:=True, _ Space:=False, _ other:=False Columns.AutoFit MsgBox "Weiter" ActiveWorkbook.Close savechanges:=False Kill sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattinhalt unter Weglassung einer Spalte als Textdatei

Problem: Wie kann ich einen Tabellenblattbereich unter Aussparung einer bestimmten Spalte als Textdatei mit einem Leerzeichen als Feldtrenner speichern?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub AlsTextSpeichern()  
    Dim rng As Range  
    Dim iFile As Integer, iRow As Integer, iCol As Integer  
    Dim sFile As String, sTxt As String, sText As String  
    Set rng = Range("A1").CurrentRegion  
    iFile = FreeFile  
    sFile = Application.Path & "\textttest.txt"  
    Open sFile For Output As iFile  
    For iRow = 1 To rng.Rows.Count  
        For iCol = 1 To rng.Columns.Count  
            Select Case iCol  
                Case 1, 2, 4  
                    sTxt = sTxt & Cells(iRow, iCol).Text & ";"  
                Case 5  
                    sTxt = sTxt & Cells(iRow, iCol).Text  
            End Select  
        Next iCol  
        Print #iFile, sTxt  
        sTxt = ""  
    Next iRow  
    Close  
    Workbooks.OpenText _  
        Filename:=sFile, _  
        DataType:=xlDelimited, _  
        tab:=False, _  
        semicolon:=True, _  
        comma:=False, _  
        Space:=False, _  
        other:=False  
    Columns.AutoFit  
    MsgBox "Weiter"  
    ActiveWorkbook.Close savechanges:=False  
    Kill sFile  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalte in Verbindung mit Variablen in Textdatei schreiben

Problem:	Wie kann ich Zellinhalte eines vorgegebenen Bereiches mit einer String-Variablen verknüpfen und in eine sequentielle Textdatei schreiben?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub TextExport() Dim iFile As Integer, iRow As Integer Dim sFile As String, sUser As String sFile = Application.Path & "\testtext.txt" sUser = Application.UserName iFile = FreeFile Open sFile For Output As iFile For iRow = 1 To 10 Print #iFile, Cells(iRow, 1).Value _ & " - " & sUser Next iRow Close iFile Workbooks.OpenText _ Filename:=sFile, _ DataType:=xlDelimited, _ tab:=False, _ semicolon:=False, _ comma:=False, _ Space:=False, _ other:=False MsgBox "Weiter" ActiveWorkbook.Close savechanges:=False Kill sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Export in *.CSV-Datei mit Kommata als Feldtrenner

Problem:	Wenn ich ein Tabellenblatt manuell als CSV-Datei speichere, wird das Semikolon als Feldtrenner eingesetzt. Wie kann ich stattdessen das Komma einsetzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub AlsCsvSpeichern() Dim sFile As String ActiveSheet.Copy sFile = Application.Path & "\testtext.csv" ActiveWorkbook.SaveAs _ FileName:=sFile, _ FileFormat:=xlCSV ActiveWorkbook.Close savechanges:=False Workbooks.Open sFile Columns.AutoFit MsgBox "Weiter" ActiveWorkbook.Close savechanges:=False Kill sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Export in Textdatei mit festen Feldlängen

Problem:	Wie kann ich einen Excel-Bereich mit einer festen Feldlänge für alle Spalten in eine Textdatei exportieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Export() Dim rng As Range Dim iLength As Integer, iFile As Integer Dim iRow As Integer, iCol As Integer Dim sFile As String, sTxt As String, sText As String Set rng = Range("A1").CurrentRegion iLength = 20 iFile = FreeFile sFile = Application.Path & "\testtext.txt" Open sFile For Output As iFile For iRow = 1 To rng.Rows.Count For iCol = 1 To rng.Columns.Count sTxt = Cells(iRow, iCol).Text sText = sText & sTxt & String(20 - Len(sTxt), " ") Next iCol Print #iFile, sText sText = "" Next iRow Close Workbooks.OpenText _ Filename:=sFile, _ DataType:=xlFixedWidth, _ tab:=False, _ semicolon:=False, _ comma:=False, _ Space:=False, _ other:=False, _ FieldInfo:=Array(Array(0, 1), Array(20, 1), _ Array(40, 1), Array(60, 1)) MsgBox "Weiter" ActiveWorkbook.Close savechanges:=False Kill sFile End Sub StandardModule: Modul1 Sub Makro1() Makro1 Makro Makro am 19.09.2002 von Hans W. Herber aufgezeichnet Workbooks.OpenText Filename:="C:\Temp\test.txt", Origin:=xlWindows, _ StartRow:=1, DataType:=xlFixedWidth, FieldInfo:=Array(Array(0, 1), Array(3, _ 1), Array(22, 1), Array(29, 1)) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textprogramm starten und Inhalte von Zelle A1 eingeben

Problem:	Wie kann ich NOTEPAD starten und dort automatisch den Inhalt der Zelle A1 eingeben?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CellToApplication() Application.SendKeys Range("A1").Value, True Shell "Notepad.exe", 3 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdatei mit vorgegebenen Feldlängen anlegen

Problem:	Wie kann ich eine Textdatei mit vorgegebenen Feldlängen anlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Aufruf() Dim sPath As String sPath = Application.DefaultFilePath & "\Test.txt" Call FesteFeldBreite(Array(12, 14, 16), sPath) End Sub Private Sub FesteFeldBreite(_ vArr As Variant, sFile As String) Dim iRow As Integer, iCol As Integer, iCount As Integer Dim sTxt As String, sAct As String Close Open sFile For Output As #1 With Range("A1").CurrentRegion For iRow = 1 To .Rows.Count For iCol = 1 To .Columns.Count iCount = vArr(iCol - 1) sAct = Cells(iRow, iCol).Value sAct = sAct & String(iCount - Len(sAct), " ") sTxt = sTxt & sAct Next iCol Print #1, sTxt sTxt = "" Next iRow End With Close MsgBox "Die Textdatei wurde angelegt:" & vbCrLf & sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fehlermeldung abfangen und in sequentielle Datei schreiben

Problem: Wie kann ich Fehlermeldungen abfangen und die Fehlernummer mit der Meldung in eine sequentielle Textdatei schreiben lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub FehlerProtokoll()
    Dim iCount As Integer
    Dim sFile As String
    sFile = Application.Path & "\MyError.txt"
    On Error GoTo ERRORHANDLER
    iCount = 40000
    End
ERRORHANDLER:
    Open sFile For Output As #1
    Print #1, "FehlerNummer: " & Err
    Print #1, "Fehler: " & Error
    Close
    Workbooks.OpenText sFile
    MsgBox "Weiter"
    ActiveWorkbook.Close savechanges:=False
    Kill sFile
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textexport in CSV-Datei mit beliebigen Feldtrennern

Problem:	Wenn ich eine Excel-Tabelle als CSV-Datei speichere, wird das Semikolon als Feldseparator gesetzt, steuere ich diesen Vorgang über ein englischsprachiges Makro, das Komma. Wie kann ich das ändern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Export() Dim iRow As Integer, iCol As Integer Dim sTxt As String, sSep As String sSep = Range("G2").Value On Error GoTo ERRORHANDLER Close Open Range("G1").Value For Output As #1 On Error GoTo 0 For iRow = 1 To 5 For iCol = 1 To 5 sTxt = sTxt & Cells(iRow, iCol).Value & sSep Next iCol sTxt = Left(sTxt, Len(sTxt) - 1) Print #1, sTxt sTxt = "" Next iRow MsgBox "Datei wurde angelegt:" & vbCrLf & Range("G1").Value End ERRORHANDLER: MsgBox "Bitte überprüfen Sie den Verzeichnisnamen!" End Sub</pre>



Werte aus Textdatei importieren und umwandeln

Problem:	Aus einer Textdatei sollen Werte importiert und umgewandelt werden. Handelt es sich bei dem rechten Zeichen im Wert um ein H, ist der Wert positiv, bei einem S ist er negativ. Die Werte werden in die Tabelle eingetragen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: Modul1 Sub Importieren() Dim arr As Variant Dim vValue As Variant Dim iRow As Integer, iCol As Integer Dim sFile As String, sTxt As String sFile = ThisWorkbook.Path & "\setminusminus.txt" If Dir(sFile) = "" Then Beep MsgBox "Kann die Textdatei " & sFile & " nicht finden!" Exit Sub End If Close Open sFile For Input As #1 Do Until EOF(1) iRow = iRow + 1 Line Input #1, sTxt arr = fncSplit(sTxt, vbTab) For iCol = 0 To UBound(arr) vValue = arr(iCol) If Right(vValue, 1) = "H" Then vValue = CDb(Left(vValue, Len(vValue) - 1)) ElseIf Right(vValue, 1) = "S" Then vValue = CDb(Left(vValue, Len(vValue) - 1)) * -1 End If Cells(iRow, iCol + 1).Value = vValue Next iCol Loop Close End Sub Function fncSplit(_ SplitText As String, _ Delimiter As Variant _) As Variant Dim iFile As Integer, iCounter As Integer Dim arr() As String Dim sTxt As String Do While SplitText <> "" ReDim Preserve arr(iCounter) If InStr(SplitText, Delimiter) Then sTxt = Left(SplitText, _ InStr(SplitText, Delimiter) - 1) arr(iCounter) = sTxt SplitText = Right(SplitText, Len(SplitText) - _ InStr(SplitText, Delimiter)) Else arr(iCounter) = SplitText SplitText = "" End If iCounter = iCounter + 1 Loop fncSplit = arr End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Textdateien eines Verzeichnisses importieren

Problem: Aus dem in Zelle B1 genannten Verzeichnis sollen alle Dateien gem. dem Filter in Zelle B2 in diese Arbeitsmappe importiert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub TextSerienImport()  
    Dim wkb As Workbook  
    Dim iCounter As Integer  
    Dim sPath As String, sPattern As String  
    Application.ScreenUpdating = False  
    sPath = Range("B1").Value  
    sPattern = Range("B2").Value  
    Set wkb = ThisWorkbook  
    With Application.FileSearch  
        .NewSearch  
        .LookIn = sPath  
        .Filename = sPattern  
        .Execute msoSortByFileName  
        For iCounter = 1 To .FoundFiles.Count  
            Workbooks.OpenText .FoundFiles(iCounter)  
            ActiveSheet.Move after:=wkb.Worksheets(wkb.Worksheets.Count)  
        Next iCounter  
    End With  
    Worksheets(1).Select  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdatei in ein neues Tabellenblatt der Arbeitsmappe importieren

Problem:	Die in Zelle B1 genannte Textdatei wird in ein neues Blatt dieser Arbeitsmappe importiert.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub TextImport() Dim wkb As Workbook Dim sFile As String Set wkb = ThisWorkbook sFile = Range("B1").Value If Dir(sFile) = "" Then Beep MsgBox "Datei wurde nicht gefunden!" Else Workbooks.OpenText sFile ActiveSheet.Move after:=wkb.Worksheets(wkb.Worksheets.Count) End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Import einer Textdatei mit mehr als 256 Spalten

Problem:	Eine Textdatei mit über 256 Spalten soll so importiert werden, dass die über das jeweilige Blatt hinausgehenden Spalten in Folgeblätter eingetragen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SplitImport() Dim wks As Worksheet Dim arr() As Variant Dim lRow As Long Dim iCol As Integer Dim sFile As String, sSeparator As String, sTxt As String Application.ScreenUpdating = False Set wks = ActiveSheet sFile = "c:\temp\test.txt" If Dir(sFile) = "" Then Beep MsgBox "Die Textdatei wurde nicht gefunden!" Exit Sub End If sSeparator = vbTab Close Open sFile For Input As #1 Do Until EOF(1) Line Input #1, sTxt Set wks = Worksheets(1) lRow = lRow + 1 iCol = 1 Do While InStr(sTxt, sSeparator) If iCol > 256 Then If wks.Index = Worksheets.Count Then Worksheets.Add after:=Worksheets(Worksheets.Count) Set wks = ActiveSheet Else Set wks = Worksheets(wks.Index + 1) End If iCol = 1 End If wks.Cells(lRow, iCol).Value = Left(sTxt, InStr(sTxt, sSeparator) - 1) sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, sSeparator)) iCol = iCol + 1 Loop wks.Cells(lRow, iCol).Value = sTxt Loop Close Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdatei über QueryTables importieren

Problem: Die in Zelle B1 genannte Textdatei soll als QueryTable importiert werden. Die Abfragetabelle wird nach dem Import gelöscht.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub TextImport()  
    Dim sFile As String  
    sFile = Range("B1").Value  
    If Dir(sFile) = "" Then  
        Beep  
        MsgBox "Die Textdatei wurde nicht gefunden!"  
        Exit Sub  
    End If  
    With ActiveSheet.QueryTables.Add(Connection:="TEXT;" & sFile & "", _  
        Destination:=Range("A3"))  
        .Name = "test"  
        .FieldNames = True  
        .RowNumbers = False  
        .FillAdjacentFormulas = False  
        .PreserveFormatting = True  
        .RefreshOnFileOpen = False  
        .RefreshStyle = xlInsertDeleteCells  
        .SavePassword = False  
        .SaveData = True  
        .AdjustColumnWidth = True  
        .RefreshPeriod = 0  
        .TextFilePromptOnRefresh = False  
        .TextFilePlatform = xlWindows  
        .TextFileStartRow = 1  
        .TextFileParseType = xlDelimited  
        .TextFileTextQualifier = xlTextQualifierDoubleQuote  
        .TextFileConsecutiveDelimiter = False  
        .TextFileTabDelimiter = True  
        .TextFileSemicolonDelimiter = False  
        .TextFileCommaDelimiter = False  
        .TextFileSpaceDelimiter = False  
        .TextFileColumnDataTypes = Array(1)  
        .Refresh BackgroundQuery:=False  
    End With  
    ActiveSheet.QueryTables(1).Delete  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdateien mit einer Zeile in Tabelle einlesen

Problem: Aus dem in Zelle B1 genannten Verzeichnis sollen die Texte aus allen *.txt-Dateien in eine neue Arbeitsmappe eingelesen werden. Jede Textdatei besteht nur aus einer Zeile.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub ImportTextFiles()
    Dim iRow As Integer
    Dim sPath As String, stxt As String
    Application.ScreenUpdating = True
    sPath = Range("B1").Value
    Close
    Workbooks.Add 1
    With Application.FileSearch
        .NewSearch
        .Filename = "*.txt"
        .LookIn = sPath
        .Execute
    On Error Resume Next
    For iRow = 1 To .FoundFiles.Count
        Open .FoundFiles(iRow) For Input As #1
        Line Input #1, stxt
        Cells(iRow, 1).Value = stxt
        Close
    Next iRow
    End With
    On Error GoTo 0
    Columns.AutoFit
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdatei auslesen und in MsgBox anzeigen

Problem:	Die Datei aus Zelle B1 soll eingelesen und in einer MsgBox ausgegeben werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetText() Dim sTxt As String, sText As String, sPath As String sPath = Range("B1").Value If Dir(sPath) = "" Then Beep MsgBox "Datei wurde nicht gefunden!" Exit Sub End If Close Open sPath For Input As #1 Do Until EOF(1) Line Input #1, sTxt sText = sText & sTxt & vbCrLf Loop Close sText = Left(sText, Len(sText) - 1) MsgBox sText End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBA-Import einer Textdatei mit Semikoli als Feldtrenner

Problem: Die in Zelle P1 genannte Textdatei soll importiert werden. Es handelt sich um eine Texttdatei mit Semikoli als Feldtrenner.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub TextImport()  
    Dim iRow As Integer, iCol As Integer  
    Dim sFile As String, sTxt As String  
    sFile = Range("P1").Value  
    If Dir(sFile) = "" Then  
        Beep  
        MsgBox "Datei wurde nicht gefunden!"  
        Exit Sub  
    End If  
    iRow = 1  
    iCol = 1  
    Close  
    Open sFile For Input As #1  
    Do Until EOF(1)  
        Line Input #1, sTxt  
        Do While InStr(sTxt, ";")  
            Cells(iRow, iCol).Value = Left(sTxt, InStr(sTxt, ";") - 1)  
            sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, ";"))  
            iCol = iCol + 1  
        Loop  
        Cells(iRow, iCol).Value = sTxt  
        iRow = iRow + 1  
        iCol = 1  
    Loop  
    Close  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdatei mit 250.000 Zeilen anlegen und importieren

Problem:	Es soll zuerst ein Textdatei mit 250.000 Zeilen angelegt werden, die dann in mehrere Arbeitsblätter zu importieren ist.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Txt2Sheets() Dim lRow As Long Dim iWks As Integer Application.ScreenUpdating = False FileCopy "c:\test.txt", "c:\test1.txt" For lRow = 1 To Fix(250000 / 65536) + 1 iWks = iWks + 1 Application.StatusBar = "Importiere " & iWks & ". Blatt..." Workbooks.OpenText _ Filename:="c:\Test1.txt", _ Origin:=xlWindows, _ StartRow:=1, _ DataType:=xlFixedWidth, FieldInfo:=Array(0, 1) With ThisWorkbook ActiveSheet.Move after:=.Worksheets(.Worksheets.Count) ActiveSheet.Name = "Blatt" & iWks End With Call DeleteText Next lRow Kill "c:\test1.txt" Application.StatusBar = False Application.ScreenUpdating = True Worksheets("Tabelle1").Select MsgBox "Job erledigt!" End Sub Private Sub DeleteText() Dim lCounter As Long Dim sTxt As String Open "c:\test1.txt" For Input As #1 Open "c:\test2.txt" For Output As #2 Do Until EOF(1) lCounter = lCounter + 1 Line Input #1, sTxt If lCounter > 65536 Then Print #2, sTxt End If Loop Close Kill "c:\test1.txt" Name "c:\test2.txt" As "c:\test1.txt" End Sub Sub CreateText() Dim lCounter As Long Open "c:\test.txt" For Output As #1 For lCounter = 1 To 250000 If lCounter Mod 10000 = 0 Then Application.StatusBar = "Lege Zeile " & lCounter & " an..." End If Print #1, "Zeile " & lCounter Next lCounter Close Application.StatusBar = False MsgBox "Textdatei wurde angelegt!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die erste Zahl in Klammern aus Textdatei auslesen

Problem:	Aus einer Textdatei soll jeweils die erste in Klammern stehende Zeichenfolge ausgelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ParseTxt() Dim var As Variant Dim lCounter As Long Dim iRow As Integer Dim sFile As String, sTxt As String Dim bln As Boolean Application.ScreenUpdating = False bln = Application.DisplayStatusBar Application.DisplayStatusBar = True sFile = Range("E1").Value If Dir(sFile) = "" Then Beep MsgBox "Datei " & sFile & " wurde nicht gefunden!" Exit Sub End If On Error GoTo ERRORHANDLER Close Open sFile For Input As #1 Do Until EOF(1) lCounter = lCounter + 1 If lCounter Mod 100 = 0 Then Application.StatusBar = "Bearbeite Zeile " & lCounter & "..." End If Line Input #1, sTxt sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, "(")) sTxt = Left(sTxt, InStr(sTxt, ")") - 1) var = Application.Match(sTxt, Columns(1), 0) If IsError(var) Then iRow = iRow + 1 Cells(iRow, 1).Value = sTxt Cells(iRow, 2).Value = 1 Else Cells(var, 2).Value = Cells(var, 2).Value + 1 End If Loop ERRORHANDLER: Close Application.StatusBar = False Application.DisplayStatusBar = bln Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text in Textdateien suchen und Fundzeilen in Tabelle ausgeben

Problem:	Alle Textdateien eines Verzeichnisses sollen nach einem Suchbegriff durchsucht und die Fundzeilen in diese Tabelle eingetragen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub GrepText() Dim fs As FileSearch Dim lRow As Long, lFile As Long Dim iCounter As Integer Dim sDir As String, sTxt As String, sBegriff As String Application.ScreenUpdating = False sDir = Range("H1").Value sBegriff = Range("H3").Value Columns("A").ClearContents Set fs = Application.FileSearch With fs .LookIn = sDir .Filename = Range("H2").Value .Execute For iCounter = 1 To .FoundFiles.Count lFile = lFile + 1 If lFile Mod 100 = 0 Then Application.StatusBar = _ "Bearbeite Datei " & lFile & "..." Open .FoundFiles(iCounter) For Input As #1 Do Until EOF(1) Line Input #1, sTxt If InStr(sTxt, sBegriff) Then lRow = lRow + 1 Cells(lRow, 1).Value = sTxt End If Loop Close Next iCounter End With Application.StatusBar = False Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fragen aus Textdateien in UserForm-Label einlesen

Problem: Je nach aufrufender UserForm-Schaltfläche sollen aus Textdateien Fragen in ein Label-Feld eingelesen werden.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmQuestions

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOne_Click()
    Call GetText(1)
End Sub

Private Sub cmdTwo_Click()
    Call GetText(2)
End Sub

Private Sub cmdThree_Click()
    Call GetText(3)
End Sub

Private Sub GetText(iQuestion As Integer)
    Dim sFile As String, sTxt As String, sQuestion As String
    sFile = ThisWorkbook.Path & "\Frage" & iQuestion & ".txt"
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Die Textdatei " & sFile & " existiert nicht!"
        Exit Sub
    End If
    Close
    Open sFile For Input As #1
    Do Until EOF(1)
        Line Input #1, sTxt
        sQuestion = sTxt & vbCrLf
    Loop
    Close
    lblQuestion.Caption = sQuestion
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmQuestions.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdateien auslesen und in Werte in Zellen eintragen

Problem:	Aus den Textdateien "Text1.txt" bis "Text3.txt" im Verzeichnis dieser Arbeitsmappen sollen jeweils 3 Zeilen ausgelesen und in einem Tabellenblatt eingetragen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Auslesen() Dim iCounter As Integer, iLine As Integer Dim sTxt As String Close Range("A1").Value = "Name" Range("B1").Value = "Vorname" Range("C1").Value = "Strasse" Rows(1).Font.Bold = True For iCounter = 1 To 3 iLine = 0 Open ThisWorkbook.Path & "\text" & iCounter & ".txt" For Input As #1 For iLine = 1 To 3 Line Input #1, sTxt sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, " ")) Cells(iCounter + 1, iLine).Value = sTxt Next iLine Close Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdatei mit mehr als 256 Spalten in Tabelle einlesen

Problem:	Eine Textdatei mit einer unbekannten, über 256 hinausgehenden Zahl von Spalten soll so eingelesen werden, dass die über 256 Spalten hinausgehenden Felder jeweils 600 Zeilen tiefer angeordnet werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub TextImport() Dim arr As Variant Dim iRow As Integer, iCol As Integer, iColT As Integer, iPlus As Integer Dim sFile As String, sTxt As String sFile = ThisWorkbook.Path & "\test_import.txt" Close Open sFile For Input As #1 Do Until EOF(1) Line Input #1, sTxt arr = fncSplit(sTxt, ",") For iCol = 0 To UBound(arr) iRow = Fix(iCol / 256) * 600 + 1 + iPlus iColT = iCol Mod 256 + 1 Cells(iRow, iColT).Value = arr(iCol) Next iCol iPlus = iPlus + 1 Loop Close MsgBox "Job erledigt!" End Sub Function fncSplit(_ SplitText As String, _ Delimiter As Variant _) As Variant Dim iFile As Integer, iCounter As Integer Dim arr() As String Dim sTxt As String Do While SplitText <> "" ReDim Preserve arr(iCounter) If InStr(SplitText, Delimiter) Then sTxt = Left(SplitText, _ InStr(SplitText, Delimiter) - 1) arr(iCounter) = sTxt SplitText = Right(SplitText, Len(SplitText) - _ InStr(SplitText, Delimiter)) Else arr(iCounter) = SplitText SplitText = "" End If iCounter = iCounter + 1 Loop fncSplit = arr End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafik in Internet Explorer laden

Problem: Wie kann ich eine Grafik, deren Name in Zelle A1 steht, in den Internet Explorer laden?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub OpenIE()  
    Dim ie As Object  
    Dim sFile As String  
    sFile = Range("A1").Value  
    If Dir(sFile) = "" Then  
        Beep  
        MsgBox "Grafik wurde nicht gefunden!"  
        Exit Sub  
    End If  
    Set ie = CreateObject("InternetExplorer.Application")  
    ie.Navigate sFile  
    ie.Visible = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdatei nach Begriff durchsuchen und Fundzeile importieren

Problem: Wie kann ich eine Textdatei nach einem bestimmten Begriff durchsuchen lassen und die Zeile, in der der Suchbegriff gefunden wurde, in ein Excel-Arbeitsblatt importieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub CreateText()  
    Dim iFile As Integer, iCounter As Integer  
    Dim sFile As String  
    iFile = FreeFile  
    sFile = Application.Path & "\textttest.txt"  
    Open sFile For Output As iFile  
    For iCounter = 1 To 10  
        If iCounter = 5 Then  
            Print #iFile, "Ein Hallo"  
        Else  
            Print #iFile, "Zeile " & iCounter  
        End If  
    Next iCounter  
    Close iFile  
    Workbooks.OpenText _  
        Filename:=sFile, _  
        DataType:=xlDelimited, _  
        tab:=False, _  
        semicolon:=False, _  
        comma:=False, _  
        Space:=False, _  
        other:=False  
    MsgBox "Weiter"  
    ActiveWorkbook.Close savechanges:=False  
End Sub  
  
Sub TextImport()  
    Dim iFile As Integer  
    Dim sSearch As String, sTxt As String  
    Dim sFile As String  
    If Dir(sFile) = "" Then  
        Beep  
        MsgBox "Sie müssen zuerst eine Textdatei anlegen!"  
        Exit Sub  
    End If  
    iFile = FreeFile  
    sFile = Application.Path & "\textttest.txt"  
    sSearch = "Hallo"  
    Open sFile For Input As iFile  
    Do Until EOF(1)  
        Input #iFile, sTxt  
        If InStr(sTxt, sSearch) Then  
            Range("A1") = "Gefunden: " & sTxt  
            Exit Do  
        End If  
    Loop  
    Close iFile  
    Kill sFile  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Textimport Strings in Zahlen wandeln

Problem:	Nach dem Import von Textdateien von einem Großrechner in Excel entsteht das Problem, daß trotz richtiger Zellformatierung ein Teil der Zahlen nur als Zeichenfolgen übernommen werden. Wie kann die Umwandlung erfolgen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub InWerte() With Range("A1").CurrentRegion .Formula = .Formula End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textdatei als externe Datei per VBA einbinden

Problem:	Wie kann ich mit VBA in XL8 eine Textdatei als externe Daten in ein Arbeitsblatt einbinden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub TextLink() Dim txt As OLEObject Dim sFile As String sFile = Range("B1").Value If Dir(sFile) = "" Then Beep MsgBox "Textdatei wurde nicht gefunden!" Else Set txt = ActiveSheet.OLEObjects.Add(_ FileName:=sFile, _ Link:=True, _ DisplayAsIcon:=False) End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Rechnungsnummer aus sequentieller Datei übernehmen

Problem:	Wie kann ich die Rechnungsnummer aus einer Textdatei abrufen und diese dort automatisch hochzählen lassen, so daß mir beim nächsten Aufruf die nächsthöhere Nummer zur Verfügung steht?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub InvoiceIni() Dim sFile As String, sNo As String Dim iNo As Integer, iFile As Integer iFile = FreeFile sFile = Application.Path & "\Invoice.ini" If Dir(sFile) <> "Invoice.ini" Then iNo = 1 Open sFile For Output As iFile Print #iFile, iNo Close Else Open sFile For Input As iFile Do Until EOF(iFile) Input #1, sNo Loop Close iNo = CInt(sNo) + 1 Open sFile For Output As iFile Print #iFile, iNo Close End If Range("A1").Value = iNo End Sub Sub ResetNo() Dim sFile As String sFile = Application.Path & "\Invoice.ini" If Dir(sFile) <> "" Then Kill sFile Range("A1").Value = 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten aus Textdatei gezielt in Zellen übernehmen

Problem: Wie kann ich Daten aus einer Textdatei gezielt in bestimmte Zellen eines Arbeitsblattes schreiben lassen? Der Feldseparator ist das Komma.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub TextImport()  
    Dim arr(1 To 6) As String  
    Dim iFile As Integer, iCounter As Integer  
    Dim sFile As String  
    sFile = Application.Path & "\textttest.txt"  
    iFile = FreeFile  
    Open sFile For Output As iFile  
    Print #iFile, "Zeile 1, Zeile 2, Zeile 3, Zeile 4, Zeile 5, Zeile 6"  
    Close  
    Open sFile For Input As iFile  
    Do Until EOF(iFile)  
        Input #iFile, arr(1), arr(2), arr(3), arr(4), arr(5), arr(6)  
    Loop  
    Close  
    Kill sFile  
    For iCounter = 1 To 6  
        Cells(iCounter, 1).Value = arr(iCounter)  
    Next iCounter  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textimport mit Semikolon als Feldseparator

Problem:	Wenn ich eine *.txt-Datei mit Semikolon als Feldtrenner manuell öffne, wird dieser erkannt, öffne ich die Datei mit Makro, wird das Semikolon nicht erkannt. Wie kann ich das ändern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub OpenText() Dim sFile As String sFile = Application.Path & "\textttest.txt" Open sFile For Output As #1 Print #1, "Text 1; Text 2; Text 3" Close Workbooks.OpenText _ FileName:=sFile, _ DataType:=xlDelimited, _ semicolon:=True MsgBox "Weiter" ActiveWorkbook.Close savechanges:=False Kill sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textimport aus Text-Datei mit Smikoli als Feldtrenner

Problem:	Wenn ich eine Text-Datei manuell importiere, wird das Semikolon als Feldseparator erkannt, steuere ich diese Vorgang über ein englischsprachiges Makro, ist dies nicht der Fall. Wie kann ich das ändern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Import() Dim sFile As String sFile = Range("B1").Value If Dir(sFile) = "" Then MsgBox "Datei wurde nicht gefunden!" Exit Sub End If Workbooks.OpenText sFile, _ DataType:=xlDelimited, Semicolon:=True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Buchstaben aus Textdatei entfernen und neu speichern

Problem: Aus der in Zelle B1 genannten Textdatei sollen alle Buchstaben entfernt werden, wobei das Ergebnis in der in Zelle B2 genannten Datei zu speichern ist.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub SearchAndChange()
    Dim FSO As Object
    Dim oFile As Object
    Dim oOFile As Object
    Dim oStrm As Object
    Dim oOStrm As Object
    Dim iRow As Integer
    Dim sTxt As String, sSource As String, sTarget As String
    Set FSO = New Scripting.FileSystemObject
    sSource = Range("B1").Value
    sTarget = Range("B2").Value
    Set oFile = FSO.GetFile(sSource)
    Set oStrm = oFile.OpenAsTextStream(ForReading)
    sTxt = oStrm.ReadAll
    oStrm.Close
    sTxt = CharFilter(sTxt)
    FSO.CreateTextFile sTarget, True
    Set oOFile = FSO.GetFile(sTarget)
    Set oOStrm = oOFile.OpenAsTextStream(ForWriting)
    oOStrm.Write sTxt
    oOStrm.Close
    MsgBox sTxt
    iRow = iRow + 1
    Set FSO = Nothing
End Sub

Function CharFilter(sTxt As String) As String
    Dim aRegExp As Object
    Set aRegExp = CreateObject("vbscript.regexp")
    aRegExp.Pattern = "[a-zA-Zäöü]"
    aRegExp.Global = True
    CharFilter = aRegExp.Replace(sTxt, "")
End Function
```



Dateinamen einlesen, Texte editieren

Problem:	Namen von Textdateien werden eingelesen und die zugehörigen Dateien werden editiert.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn Schaltflächen zu.
VBA-Code:	StandardModule: Modul1 <pre>Sub ListFiles() Dim sPath As String Dim iCounter As Integer sPath = GetDirectory("Verzeichnis auswählen:") If sPath = "" Then Exit Sub Range("A2:A65536").ClearContents With Application.FileSearch .NewSearch .Filename = "*.dxf" .LookIn = sPath .Execute .MatchTextExactly = True For iCounter = 1 To .FoundFiles.Count Cells(iCounter + 1, 1).Value = Dir(.FoundFiles(iCounter)) Next iCounter End With Range("H1").Value = sPath End Sub</pre> <pre>Sub SearchAndChange() Dim FSO As Object Dim oFile As Object Dim oOFile As Object Dim oStrm As Object Dim oOStrm As Object Dim iRow As Integer Dim sTxt As String, sSource As String, sTarget As String Set FSO = New Scripting.FileSystemObject If IsEmpty(Range("H1")) Or IsEmpty(Range("H2")) Then Beep MsgBox "Quell- oder Zielverzeichnis fehlen!" Exit Sub End If iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) sSource = Range("H1").Value & "\" & Cells(iRow, 1).Value sTarget = Range("H2").Value & "\" & Cells(iRow, 4).Value Set oFile = FSO.GetFile(sSource) Set oStrm = oFile.OpenAsTextStream(ForReading) sTxt = oStrm.ReadAll oStrm.Close If InStr(sTxt, Cells(iRow, 2).Value) Then sTxt = Replace(sTxt, Cells(iRow, 2).Value, Cells(iRow, 3).Value) FSO.CreateTextFile sTarget, True Set oOFile = FSO.GetFile(sTarget) Set oOStrm = oOFile.OpenAsTextStream(ForWriting) oOStrm.Write sTxt oOStrm.Close Else Cells(iRow, 5).Value = False End If MsgBox sTxt iRow = iRow + 1 Loop Set FSO = Nothing End Sub</pre> StandardModule: Modul2 <pre>Public Type BROWSEINFO hOwner As Long pidlRoot As Long pszDisplayName As String lpszTitle As String ulFlags As Long lpfn As Long lParam As Long iImage As Long End Type</pre> <pre>Declare Function SHGetPathFromIDList Lib "shell32.dll" _ Alias "SHGetPathFromIDListA" (ByVal pidl As Long, _ ByVal pszPath As String) As Long</pre> <pre>Declare Function SHBrowseForFolder Lib "shell32.dll" _ Alias "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long</pre> <pre>Function GetDirectory(Optional msg) As String Dim bInfo As BROWSEINFO Dim Path As String Dim r As Long, x As Long, pos As Integer bInfo.pidlRoot = 0& If IsMissing(msg) Then bInfo.lpszTitle = "Wählen Sie bitte einen Ordner aus." Else bInfo.lpszTitle = msg End If bInfo.ulFlags = &H1 x = SHBrowseForFolder(bInfo) Path = Space\$(512) r = SHGetPathFromIDList(ByVal x, ByVal Path) If r Then pos = InStr(Path, Chr\$(0)) GetDirectory = Left(Path, pos - 1) Else GetDirectory = "" End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufruf eines VBScripts aus VBA

Problem:	Der Aufruf eines VBScripts (mit einfacher Meldung) soll demonstriert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Aufruf() Dim v As Variant Dim sPath As String sPath = ThisWorkbook.Path & "\message.vbs" If Dir(sPath) = "" Then Beep MsgBox "Datei wurde nicht gefunden!" Exit Sub End If v = Shell(_ "wscript c:\hwh\excel\support\forum\message.vbs ""Hans W. Herber"" " _ & Now, vbHide) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateiinformationen auslesen

Problem:	Die Dateiinformationen der in Zelle B1genannten Datei sollen ausgelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu. <pre>StandardModule: Modul1 Sub ReadProperties() Dim objWMIService As Object Dim colFiles As Object Dim objFile As Object Dim strPath As String, strComputer As String strPath = Range("B1").Value If Dir(strPath) = "" Then Beep MsgBox "Die Datei wurde nicht gefunden!" Exit Sub End If strPath = WorksheetFunction.Substitute(strPath, "\", "\\") strComputer = "." Set objWMIService = GetObject("winmgmts:" _ & "{impersonationLevel=impersonate}!\\\" & strComputer & "\root\cimv2") Set colFiles = objWMIService.ExecQuery _ ("Select * from CIM_Datafile Where name = '" & strPath & "'") For Each objFile In colFiles With objFile Cells(3, 2).Value = .AccessMask Cells(4, 2).Value = .Archive Cells(5, 2).Value = .Compressed Cells(6, 2).Value = .CompressionMethod Cells(7, 2).Value = .CreationDate Cells(8, 2).Value = .CSName Cells(9, 2).Value = .Drive Cells(10, 2).Value = .EightDotThreeFileName Cells(11, 2).Value = .Encrypted Cells(12, 2).Value = .EncryptionMethod Cells(13, 2).Value = .Extension Cells(14, 2).Value = .Filename Cells(15, 2).Value = .FileSize Cells(16, 2).Value = .FileType Cells(17, 2).Value = .FSName Cells(18, 2).Value = .Hidden Cells(19, 2).Value = .LastAccessed Cells(20, 2).Value = .LastModified Cells(21, 2).Value = .Manufacturer Cells(22, 2).Value = .Name Cells(23, 2).Value = .Path Cells(24, 2).Value = .Readable Cells(25, 2).Value = .System Cells(26, 2).Value = .Version Cells(27, 2).Value = .Writeable End With Next objFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamische Abschlußlinie unter benutzten Bereich ziehen

Problem: Unterhalb eines benutzten Tabellenblattbereiches soll eine dynamische Linie gezogen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und binden Sie die Funktionen in die bedingte Formatierung ein.

VBA-Code:

```
StandardModule: Modul1

Function RealLastCell() As Variant
    Dim rng As range
    Dim arr(1 To 2)
    Dim Row%, Col%
    Set rng = range("H25")
    Row = rng.Row
    Do While Application.CountA(Rows(Row)) = 0 And Row <> 1
        Row = Row - 1
    Loop
    arr(1) = Row
    Col = rng.Column
    Do While Application.CountA(Columns(Col)) = 0 And Col <> 1
        Col = Col - 1
    Loop
    arr(2) = Col
    RealLastCell = arr
End Function

Function GetRow() As Integer
    Dim arr As Variant
    arr = RealLastCell()
    GetRow = arr(1)
End Function

Function GetColumn() As Integer
    Dim arr As Variant
    arr = RealLastCell()
    GetColumn = arr(2)
End Function
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Excel-Arbeitsmappe mit WinZip packen und entpacken

Problem: Wie kann ich eine Excel-Arbeitsmappe mit WinZip packen bzw. entpacken?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Entpacken()
    Dim sXLPPath As String, sZIPPath As String
    Dim sWinZipPath As String
    sXLPPath = Range("B2").Value
    sZIPPath = Left(sXLPPath, Len(sXLPPath) - 3) & ".zip"
    If Dir(sZIPPath) = "" Then
        Beep
        MsgBox "Gepackte Excel-Arbeitsmappe nicht gefunden!"
        Exit Sub
    End If
    sWinZipPath = Range("B1").Value
    If Dir(sWinZipPath) = "" Then
        Beep
        MsgBox "WinZip wurde nicht gefunden!"
        Exit Sub
    End If
    Shell sWinZipPath & " -e " & sZIPPath
    Application.Wait Now + TimeSerial(0, 0, 2)
    Workbooks.Open sXLPPath
    Kill sZIPPath
End Sub

Sub Verpacken()
    Dim sXLPPath As String, sZIPPath As String
    Dim sWinZipPath As String
    sXLPPath = Range("B2").Value
    If Dir(sXLPPath) = "" Then
        Beep
        MsgBox "Excel-Arbeitsmappe nicht gefunden!"
        Exit Sub
    End If
    sZIPPath = Left(sXLPPath, Len(sXLPPath) - 3) & ".zip"
    sWinZipPath = Range("B1").Value
    If Dir(sWinZipPath) = "" Then
        Beep
        MsgBox "WinZip wurde nicht gefunden!"
        Exit Sub
    End If
    Shell sWinZipPath & " -a " & sZIPPath & " " & sXLPPath
    Application.Wait Now + TimeSerial(0, 0, 2)
    Kill sXLPPath
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Word mit leerem Dokument und Druckdialog aufrufen

Problem:	Word soll mit einem leeren Dokument und dem Druckdialog geöffnet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Aufrufen() Dim appWord As Object Dim wdDoc As Object Set appWord = CreateObject("Word.Application") Set wdDoc = appWord.Documents.Add appWord.Dialogs(88).Show appWord.Visible = True Set wdDoc = Nothing Set appWord = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

OLE-Verbindung aufbauen und Verb senden

Problem:	Es soll eine OLEObject-Verbindung zu dem in Zelle B1 genannten Word-Dokument hergestellt und das Verb zum Öffnen des Dokuments gesendet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SendVerb() Dim oOle As OLEObject Dim sFile As String sFile = Range("B1").Value If Dir(sFile) = "" Then Beep MsgBox ("Datei wurde nicht gefunden!") Exit Sub End If Set oOle = Worksheets(1).OLEObjects.Add(FileName:= _ sFile, Link:=True, DisplayAsIcon:=False) oOle.Verb (xlVerbOpen) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Standardmailprogramm über Word feststellen

Problem: Es soll festgestellt werden, welches Standard-Mailprogramm festgelegt wurde. Da Excel-VBA eine solche Abfrage nicht vorsieht, erfolgt sie über eine WordInstanz.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub ReadRegistry()
    Dim appWd As Object
    Dim iTrenner As Integer
    Dim ID As String
    Set appWd = CreateObject("Word.Application")
    ID = appWd.System.PrivateProfileString("", _
        "HKEY_LOCAL_MACHINE\SOFTWARE\Classes\mailto\shell\open\command", "")
    iTrenner = InStr(1, ID, ".exe", vbTextCompare)
    ID = Left(ID, iTrenner + 4)
    If InStr(1, ID, "%ProgramFiles%") > 0 Then
        iTrenner = InStr(1, ID, "%ProgramFiles%")
        ID = Right(ID, Len(ID) - iTrenner - Len("%ProgramFiles%"))
        ID = Environ("ProgramFiles") & "\" & ID
    ElseIf InStr(1, ID, "%SystemDrive%") > 0 Then
        iTrenner = InStr(1, ID, "%ProgramFiles%")
        ID = Right(ID, Len(ID) - iTrenner - Len("%SystemDrive%"))
        ID = Environ("SystemDrive") & "\" & ID
    End If
    Set appWd = Nothing
    If Right(ID, 1) = Chr(34) Then ID = Left(ID, Len(ID) - 1)
    MsgBox ID
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wordmakro mit Parameter aufrufen

Problem:	An ein Wordmakro soll ein Parameter übergeben und von dort zurückgegeben werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallWordMacro() Dim wdApp As Object Dim wdDoc As Object Dim vWert As Variant On Error GoTo ERRORHANDLER vWert = "1,95583" Set wdApp = CreateObject("Word.Application") Set wdDoc = wdApp.documents.Open(_ ThisWorkbook.Path & "\WordMakro.doc") On Error Resume Next wdDoc.CustomDocumentProperties.Add _ Name:="Wert", _ Value:=vWert, _ LinkToContent:=False, _ Type:=msoPropertyTypeString If Err > 0 Then wdDoc.CustomDocumentProperties("Wert").Value = vWert End If wdApp.documents.Open ThisWorkbook.Path & "\WordMakro.doc" wdApp.Run "WD.Modul1.Meldung" vWert = CDb1(wdDoc.CustomDocumentProperties("Wert")) MsgBox vWert ERRORHANDLER: wdApp.Quit Set wdDoc = Nothing Set wdApp = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in Word-Dokument einlesen und dieses Drucken

Problem: Der Bereich A1:C1 soll in ein Word-Dokument eingelesen und von dort gedruckt werden. Die Aktion soll im Hintergrund ablaufen.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code:

```
StandardModule: Modul1

Sub PrintWord()
    Dim wdApp As Object
    Dim wdDoc As Object
    Dim sFile As String
    sFile = Range("F1").Value
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Datei wurde nicht gefunden!"
    Else
        Set wdApp = CreateObject("Word.Application")
        wdApp.Visible = False
        Set wdDoc = wdApp.Documents.Open(sFile)
        Range("A1:C1").Copy
        wdDoc.Range.Paste
        wdDoc.PrintOut
        Application.Wait Now + TimeSerial(0, 0, 5)
        wdDoc.Close savechanges:=False
        wdApp.Quit
        Set wdDoc = Nothing
        Set wdApp = Nothing
        Application.CutCopyMode = False
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Das in einem Gültigkeitsfeld ausgewählte Word-Dokument öffnen

Problem: Das ausgewählte Word-Dokument soll in Word geöffnet werden. Routine funktioniert nur ab XL2000, da das Change-Ereignis bei XL97 durch eine Gültigkeitsauswahl nicht ausgelöst wird.

Lösung: Die Ereignis-Programmierung befindet sich im Klassenmodul der Tabelle.

VBA-Code: StandardModule: basMain

```
Sub OpenWordDoc(sFile As String)
    Dim WordObj As Object
    Dim WordDoc As Object
    On Error GoTo w
    AppActivate "Microsoft Word"
    GoTo n
w:
    Set WordObj = CreateObject("Word.Application")
n:
    Set WordDoc = WordObj.Documents.Open(sFile)
    WordObj.Visible = True
End Sub

ClassModule: Tabelle3

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim sPath As String
    If Target.Address <> "$D$9" Then Exit Sub
    sPath = "c:\eigene dateien\" & Target.Value
    Call OpenWordDoc(sPath)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle geöffneten Tasks auslesen

Problem:	Alle geöffneten Tasks sollen über den Word-Tasks-Befehl in diese Tabelle eingelesen werden. In der Entwicklungsumgebung muss ein Verweis zur Microsoft Word Object Library hergestellt sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CreateMsgTasks() Dim appWord As Object Dim iCounter As Integer Set appWord = CreateObject("Word.Application") For iCounter = 1 To appWord.Tasks.Count Cells(iCounter, 1) = appWord.Tasks(iCounter) Next iCounter Set appWord = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Excel-Zellinhalte in Worddokument einlesen um per Email versenden

Problem:	Wie kann ich den Inhalt einer Excel-Tabelle in ein Worddokument einfügen und dieses automatisch per Email versenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SendMessage() Dim oWord As Object Dim oDoc As Object Dim oOL As Object Dim oOLMsg As Object Dim oOLRecip As Object Dim oOLAttach As Object Dim iRow As Integer Dim sFile As String sFile = Application.Path & "\test.doc" ThisWorkbook.Worksheets("Tabelle2").UsedRange.Copy Set oWord = CreateObject("Word.Application") Set oDoc = oWord.Documents.Add oDoc.Range.Paste oDoc.SaveAs sFile oWord.Quit Set oDoc = Nothing Set oWord = Nothing Set oOL = CreateObject("Outlook.Application") Set oOLMsg = oOL.CreateItem(0) With oOLMsg iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) Set oOLRecip = .Recipients.Add(Cells(iRow, 1)) oOLRecip.Type = 1 iRow = iRow + 1 Loop Set oOLAttach = .Attachments.Add(sFile) .Subject = Format(Date, "dd.mm.yy") & " - " & Format(Time, "hh:mm:ss") .Body = "Beiliegend der Excel-Text" .Importance = 1 For Each oOLRecip In .Recipients oOLRecip.Resolve Next .Send End With Set oOL = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ersten Absatz eines Word-Dokumentes in Tabelle einfügen

Problem:	Wie kann ich in das aktive Tabellenblatt den ersten Absatz eines Word-Dokumentes einfügen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub WordImport() Dim appWord As Object Dim docWord As Object Dim rngWord As Object Dim sFile As String sFile = Range("B1").Value If Dir(sFile) = "" Then Beep MsgBox "Word-Dokument wurde nicht gefunden!" Exit Sub End If Set appWord = CreateObject("Word.Application") Set docWord = appWord.Documents.Open(sFile) Set rngWord = docWord.Paragraphs(1).Range rngWord.Copy Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Paste appWord.Quit Set rngWord = Nothing Set docWord = Nothing Set appWord = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wordmakro aus Excel heraus aufrufen

Problem: Wie kann ich aus Excel heraus Word mit einem bestimmten Dokument laden und ein dort befindliches Makro starten?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdWordMakro_Click()
    Call WDTest
End Sub

StandardModule: basMain

Sub WordTest()
    Dim appWord As Object
    Dim sFile As String
    sFile = Range("B1").Value
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Word-Dokument wurde nicht gefunden!"
    Else
        Set appWord = CreateObject("Word.Application")
        appWord.Documents.Open sFile
        appWord.Run "Project.Modul1." & Range("B2").Value
        appWord.Quit
        Set appWord = Nothing
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus Excel heraus ein Worddokument drucken

Problem: Wie kann ich aus Excel heraus Word veranlassen, ein bestimmtes Dokument zu drucken?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cmdWordDruck_Click()
    Call WordDrucken
End Sub

StandardModule: basMain

Sub WordDrucken()
    Dim wrdApp As Object
    Dim wrdDoc As Object
    Dim sFile As String
    sFile = Range("B1").Value
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Worddokument wurde nicht gefunden!"
    Else
        Set wrdApp = CreateObject("Word.Application")
        Set wrdDoc = wrdApp.Documents.Open(sFile)
        wrdDoc.PrintOut
        Application.Wait Now + TimeSerial(0, 0, 5)
        wrdApp.Quit
        Set wrdDoc = Nothing
        Set wrdApp = Nothing
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Word mit bestimmtem Dokument starten

Problem: Wie kann ich Word mit einem im gleichen Verzeichnis wie die aufrufende Excel-Arbeitsmappe liegenden Dokument starten?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub WordOeffnen()
    Dim sFile As String
    sFile = Range("B1").Value
    If Dir(sFile) <> "" Then
        Shell "winword.exe "" & sFile & """, 1
    Else
        MsgBox "Datei nicht gefunden!"
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Unterverzeichnisse eines vorgegebenen Pfades auslesen

Problem:	Es sollen alle Unterverzeichnisse des in Zelle B1 genannten Pfades genannt werden. WindowsScriptingHost muss installiert sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ReverseFolders() Dim FS0 As Object Set FS0 = CreateObject("Scripting.FileSystemObject") ShowSubFolders FS0.GetFolder(Range("B1").Value) End Sub Sub ShowSubFolders(Folder) Dim Subfolder As Object For Each Subfolder In Folder.SubFolders MsgBox Subfolder.Path ShowSubFolders Subfolder Next End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Programmversion eines beliebigen Programms feststellen

Problem:	Die Programmversion der in Zelle B1 genannten Anwendung wird gemeldet. WindowsScriptingHost muss installiert sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ProgrammVersion() Dim objFSO As Object Set objFSO = CreateObject("Scripting.FileSystemObject") MsgBox objFSO.GetFileVersion(Range("B1").Value) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Absolute Adresse eines Hyperlinks feststellen

Problem:	Aus dem Hyperlink in Zelle A1 soll die absolute Adresse ermittelt werden. WindowsScriptingHost muss installiert sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetFileName() Dim fs As Object Dim f As Object Dim iChar As Integer Dim sFolder As String, sOrdner As String, sFile As String Dim sPathR As String, sPathA As String sPathR = ThisWorkbook.Path sFolder = Range("A1").Hyperlinks(1).Address If InStr(sFolder, "..\") Then For iChar = Len(sFolder) To 1 Step -1 If Mid(sFolder, iChar, 1) = "\" Then Exit For Next iChar sFile = Right(sFolder, Len(sFolder) - iChar) sOrdner = WorksheetFunction.Substitute(sFolder, "..\", "") sOrdner = Left(sOrdner, InStr(sOrdner, "\")) Set fs = CreateObject("Scripting.FileSystemObject") Do While InStr(sFolder, "..\") Set f = fs.GetFolder(sPathR) sPathR = WorksheetFunction.Substitute(sPathR, "\" & f.Name, "", 1) sFolder = WorksheetFunction.Substitute(sFolder, "..\", "", 1) Loop Set fs = Nothing Set f = Nothing sFile = sPathR & "\" & sOrdner & sFile Else sFile = ThisWorkbook.Path & "\" & sFile End If MsgBox sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBScript mit Parameter aus VBA aufrufen

Problem: Ein VBScript soll mit den Argumenten Anwendername und Datum/Zeit aufgerufen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub VbsAufruf()
    Dim sFile As String
    Dim sArg As String
    sArg = "" & Application.UserName & Chr(34) & Now & ""
    sFile = ThisWorkbook.Path & "\message.vbs "
    Shell "WScript.exe " & sFile & sArg, vbNormalFocus
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ein Meldungsfenster für 5 Sekunden anzeigen

Problem:	Excel/VBA-Beispiel: Ein Meldungsfenster für 5 Sekunden anzeigen
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub TempMsg() Dim wsh As New IWshShell_Class Dim sMsg As String, sTitle As String sMsg = "Eine Anzeige für 5 Sekunden" sTitle = "Message-Demo" wsh.Popup sMsg, 5, sTitle, 7 + vbInformation Set wsh = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Registry-Einträge schreiben, lesen, löschen

Problem:	Excel/VBA-Beispiel: Registry-Einträge schreiben, lesen, löschen
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub RegWrite() Dim wsh As New IWshShell_Class With wsh .RegWrite "HKCU\Software\HWH\", "Dies ist eine Voreinstellung" .RegWrite "HKCU\Software\HWH\MyString", "Eine Zeichenfolge" .RegWrite "HKCU\Software\HWH\MyNumString", 12345 .RegWrite "HKCU\Software\HWH\MyDwordValue", 12345, "REG_DWORD" .RegWrite "HKCU\Software\HWH\MyBinaryValue", 12345, "REG_BINARY" End With Set wsh = Nothing End Sub Sub RegRead() Dim wsh As New IWshShell_Class With wsh MsgBox .RegRead("HKCU\Software\HWH\") & vbCrLf & _ .RegRead("HKCU\Software\HWH\MyString") & vbCrLf & _ .RegRead("HKCU\Software\HWH\MyNumString") & vbCrLf & _ .RegRead("HKCU\Software\HWH\MyDwordValue") End With Set wsh = Nothing End Sub Sub RegDelete() Dim wsh As New IWshShell_Class wsh.RegDelete "HKCU\Software\HWH\" ' wsh.RegDelete "HKCU\Software\HWH\MyString" ' wsh.RegDelete "HKCU\Software\HWH\MyNumString" ' wsh.RegDelete "HKCU\Software\HWH\MyDwordValue" ' wsh.RegDelete "HKCU\Software\HWH\MyBinaryValue" ' wsh.RegDelete "HKCU\Software\HWH\" Set wsh = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob ein Laufwerk auf dem Rechner existiert

Problem:	Es soll geprüft werden, ob dass in Zelle B1 genannte Laufwerk auf dem Rechner existiert. Hierzu muß in Der Entwicklungsumgebung ein Verweis zum Windows Scripting Host geschaltet sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CheckDriveExists() Dim FSobj As Object Set FSobj = CreateObject("Scripting.FileSystemObject") If FSobj.DriveExists(Range("B1").Value) Then MsgBox "Existiert!" Else MsgBox "Existiert nicht!" End If Set FSobj = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

DeskTop-Link zu einer Excel-Arbeitsmappe erstellen

Problem:	Es soll ein DeskTop-Link zu einer in Zelle B1 genannten Excel-Arbeitsmappe erstellt werden. In der Entwicklungsumgebung muss ein Verweis zum Windows Scripting Host geschaltet sein.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub MakeDesktopShortcut() Dim wsh As New IWshShell_Class Dim SC As IWshShortcut_Class Dim sPath As String, sFile As String sPath = wsh.SpecialFolders.Item("Desktop") sFile = Range("B1").Value Set SC = wsh.CreateShortcut(sPath & "\test.lnk") SC.TargetPath = sFile SC.Hotkey = "CTRL+ALT+Z" SC.Save Set wsh = Nothing End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eine Reihe von Beispielen zum Feature "Bedingte Formatierung"

Problem:	Eine Reihe von Beispielen zum Feature "Bedingte Formatierung" in 7 Tabellenblättern
Lösung:	Geben Sie die benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function IsProtect(rng As Range) As Boolean If rng.Locked Then IsProtect = True End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalte mit aktuellem Datum markieren

Problem:	Die Spalte mit dem aktuellen Datum in Zeile 1 soll gelb markiert werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>ClassModule: UserForm1 Private Sub CommandButton1_Click() End Sub Private Sub TextBox5_Exit(ByVal Cancel As MSForms.ReturnBoolean) If TextBox5.TextLength = 0 Then Exit Sub If TextBox5.TextLength < 6 Then MsgBox "Bitte mindestens eine 6-stellige Zahl angeben!" Exit Sub End If If TextBox5.TextLength > 6 And TextBox5.TextLength < 12 Then MsgBox "Bitte eine 6- oder 12-stellige Zahl angeben!" Exit Sub End If If TextBox5.TextLength = 6 Then TextBox5.Text = Format(TextBox5.Text, "### ###") Else TextBox5.Text = Format(TextBox5.Text, "## ### ### ### @") End If End Sub Private Sub UserForm_Click() End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeitdarstellung als Zeitstrahl

Problem:	In einer Tabelle werden die Zeiten gemäß der Tabelle in A1:G5 als Zeitstrahlen dargestellt.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub cmdHyperlink_Click() ActiveWorkbook.FollowHyperlink "http://www.excel-cd.de/i_bestellung.html", NewWindow:=True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Zellen mit Formeln finden

Problem:	In einem vorgegebenen Bereich sollen alle Zellen mit Formeln über die Hintergrundfarbe markiert werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>StandardModule: Modul1 Function IsFormula(rng As Range) As Boolean IsFormula = rng.HasFormula End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelle markieren, wenn zwei Bedingungen erfüllt sind

Problem:	In Spalte B sollen alle Werte markiert werden, wenn in Spalte D die Differenz größer als 2 ist und in Spalte E kein ok steht.
Lösung:	Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte, die auch in anderen Spalten vorkommen, markieren

Problem: Alle Werte, die in anderen Spalten ebenfalls vorkommen, sollen markiert werden.

Lösung: Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.

VBA-Code:

```
StandardModule: Modul1

Sub Markierne()
    Dim rng As Range
    Set rng = Range("A1").CurrentRegion
    Dim iRow As Integer, iColC As Integer, iCol As Integer

End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen unterhalb Samstagen und Sonntagen grau hinterlegen

Problem: Die Zellen unter Samstag und Sonntag sollen grau hinterlegt werden.

Lösung: Nur anhand einer Beispielarbeitsmappe darstellbar.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Negative Zahlen ausblenden

Problem: Negative Zahlen in Spalte A sollen ausgeblendet werden.

Lösung: Folgende Formel in der Bedingten Formatierung: `=A1<0`

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelt vorkommende Werte farblich kennzeichnen

Problem:	In Spalte A sollen doppelt vorkommende Werte farblich gekennzeichnet werden.
Lösung:	Bei der bedingten Formatierung einzutragende Formel: =ZÄHLENWENN(A:A;A1)>1
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte markieren, die größer als Parallelwerte sind

Problem:	Die Werte in Spalte C, die größer als die Parallelwerte in Spalte B sind, sollen mit grünem Hintergrund belegt werden.
Lösung:	Unter "Bedingte Formatierung" folgende Formel eingeben: =C2>B2 und Format festlegen.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Unterschiedliche Namen mit Rahmenlinien trennen

Problem:	Unterschiedliche Namen in einer Liste sollen durch Rahmenlinien getrennt werden.
Lösung:	Folgende Formel unter "Bedingte Formatierung" einsetzen und Format festlegen: =A1<>A2
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hervorhebung von Anwesenheitszeiten

Problem: Über eine gelbe Hintergrundmarkierung soll anhand von Anfangs- und Endzeiten sichtbar gemacht werden, wer zu welcher Stunde anwesend ist.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code: StandardModule: basFunctions

```
Function MA(rngAll As Range, iHour As Integer)
    Dim rng As Range
    Dim iCount As Integer
    For Each rng In rngAll.Cells
        If rng.Interior.ColorIndex = 6 And _
            Hour(Cells(1, rng.Column).Value) = iHour And _
            rng.Offset(0, 1).Interior.ColorIndex = 6 Then
            If Hour(Cells(1, rng.Column - 1).Value) <> iHour Or _
                rng.Offset(0, -1).Interior.ColorIndex <> 6 Then
                iCount = iCount + 1
            End If
        End If
    Next rng
    MA = iCount
End Function

Function MA1(rngAll As Range, iHour As Integer)
    Dim iRow As Integer, iCol As Integer, iCount As Integer
    iCol = WorksheetFunction.CountA(Rows(1)) - 1
    For iRow = 2 To rngAll.Rows.Count
        If iHour / 24 >= Cells(iRow, iCol).Value And _
            iHour / 24 < Cells(iRow, iCol + 1).Value Then
            iCount = iCount + 1
        End If
    Next iRow
    MA1 = iCount
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrundfarbe bei Null-Bestand ändern.

Problem:	Bei dem Wert "0 Bestand" in Zelle B1 soll die Hintergrundfarbe auf gelb wechseln. Lösung über Menü FORMAT / BEDINGTE FORMATIERUNG
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelt vorkommende Werte in Spalte A markieren

Problem:	Alle in Spalte A doppelt vorkommenden Werte sollen markiert werden. Lösung über Menü FORMAT / BEDINGTE FORMATIERUNG
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle ungeraden Zahlen markieren

Problem:	Alle ungeraden Zahlen in Spalte A sollen mit einem roten Hintergrund versehen werden. Lösung über Menü FORMAT / BEDINGTE FORMATIERUNG.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen in Abhängigkeit des Summen-Teilergebnisses markieren

Problem:	Alle Werte in Spalte B, bei denen das Summen-Teilergebnis größer/gleich 5 ist, sollen farblich hervorgehoben werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensätze unter Maximum ausblenden

Problem:	Es soll in Spalte A:B immer nur die Zeile sichtbar sein, die das höchste Datum enthält. Lösung über Bedingte Formatierung.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrundfarbe in Abhängigkeit des Zellwertes

Problem:	Die Hintergrundfarbe eines Prozentbereiches soll in Abhängigkeit des jeweiligen Wertes gewählt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenvergleich über die bedingte Formatierung

Problem:	In einer Tabelle sollen Abweichungen in Tabelle2 zu Tabelle3 angezeigt werden. Lösung über die Kombination von Formeln und bedingter Formatierung.
Lösung:	Nur anhand einer Beispieldarstellungsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Letzte Zeile mit einem bestimmten Anfangsbuchstaben markieren

Problem:	Die Zelle mit dem letzten Eintrag zu einem Anfangsbuchstaben soll einen unteren Rahmen bekommen. Lösung über Menü FORMAT / BEDINGTE FORMATIERUNG.
Lösung:	Nur anhand einer Beispieldarstellungsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kennzeichnung von Lottogewinnzahlen

Problem:	Über die bedingte Formatierung sollen die jeweiligen Lottogewinnzahlen gekennzeichnet werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datumsbereich gem. Start- und Enddatum markieren

Problem:	Die Zellen, deren Datumswerte im Bereich der Anfang- und Endwerte einer zweiten Tabelle liegen, sollen markiert werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeitabhängige Zellmarkierungen über bedingte Formatierung setzen

Problem:	A1 soll von 22:01-06:00, B1 von 06:01 bis 14:00 und C1 von 14:01 bis 22:00 Uhr einen Rahmen erhalten.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Überschrittene Zahlungstermine markieren

Problem:	Die überschrittenen Zahlungstermine sollen farblich markiert werden.
Lösung:	Lösung über bedingte Formatierung; nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Markierung von Lotto-Treffern bei einer Tippgemeinschaft.

Problem:	In den Tippzeilen einer Tippgemeinschaft sollen die Lottozahlentreffer gelb markiert werden. Lösung über Menü FORMAT / BEDINGTE FORMATIERUNG.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Farbliche Kennzeichnung nach Werte-Verhältnis

Problem:	Die Verkaufspreise, die kleiner als der Einkaufspreis * 200% sind, sollen grün, die, die größer als Einkaufspreis * 260% sind, blau gekennzeichnet werden.
Lösung:	Erste Formel für die bedingte Formatierung: =B2<(A2*2)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Min- und Max-Werte automatisch markieren

Problem:	In einer Kalorienverbrauchstabelle sollen die Min- und Max-Werte automatisch markiert werden.
Lösung:	Formel in bedingter Formatierung: =C2=MAX(C:C)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blaue Linie, wenn Wert ungleich dem Wert der Vorgängerzelle

Problem:	Wenn der Wert in einer Zelle in Spalte A <> dem Wert der darunterliegenden Zelle, soll eine blaue Linie über den gesamten Datenbereich gezogen werden.
Lösung:	Folgende Formel in der bedingten Formatierung: =\$A1<>\$A2
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Markierung von Formeln über bedingte Formatierung

Problem:	Alle Zellen in Spalte A, die weder leer sind noch eine Formel enthalten, sollen gelb markiert werden. Lösung über benutzerdefinierte Funktion im Modul basMain in Verbindung mit der bedingten Formatierung über Menü FORMAT.
Lösung:	Formel für die bedingte Formatierung: =UND(istformel(A1)=FALSCH;A1<>"")
VBA-Code:	<pre>StandardModule: Modul1 Function IstFormel(rng As Range) If rng.HasFormula Then IstFormel = True End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrundfarbe einer Zeile, wenn bestimmter Wert in 1 Zelle

Problem:	Wie kann ich eine Zeile in A:H farbig markieren, wenn eine Zelle in der Zeile ein "x" beinhaltet?
Lösung:	Erste Formel für die bedingte Formatierung: =ISTFEHLER(VERGLEICH("x";1:1;0))=FALSCH
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Maximalwerte im selektierten Bereich markieren

Problem:	Wie kann ich die Maximalwerte im selektierten Bereich über die bedingte Formatierung violett markieren lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub BedingtFormat() With Selection .FormatConditions.Delete .FormatConditions.Add Type:=xlExpression, Formula1:= _ "=" & Range("Bereich").Cells(1).Address(False, False) & _ "=MAX(" & Selection.Address & ")" .FormatConditions(1).Interior.ColorIndex = 39 End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hintergrund setzen, wenn mehr als 9 Einträge

Problem:	Bei mehr als 9 zusammenhängenden Einträgen in Spalte B soll der jeweils erste Eintrag gelb markiert werden.
Lösung:	Formel für die bedingten Formatierung: =ANZAHL2(B1:B9)>8
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bedingte Formatierung in Abhängigkeit eines Zellkommentars

Problem: Wie kann ich eine bedingte Formatierung von einem Zellkommentar in einem anderen Blatt abhängig machen?

Lösung: Den Code in ein Standardmodul eingeben.

VBA-Code:

```
StandardModule: basMain

Function Test(rng As Range) As Boolean
    If rng.Comment.Text = "Test" Then
        Test = True
    Else
        Test = False
    End If
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile mit Inhalt mit gelbem Hintergrund

Problem:	Wie kann ich alle Zeilen mit Inhalt gelb hinterlegen?
Lösung:	Formel für bedingte Formatierung: =ANZAHL2(\$A2:\$J2)>0
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Jede zweite Zeile mit Hintergrundfarbe belegen

Problem:	Wie kann ich jede zweite Zeile mit einer Hintergrundfarbe belegen?
Lösung:	Formel für die bedingte Formatierung: =REST(ZEILE();2)=1
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ersten mit einem neuen Buchstaben beginnenden Namen fett schreiben

Problem:	Wie kann ich veranlassen, dass in einer Spalte der erste mit einem neuen Buchstaben beginnende Name mit dem Schriftattribut fett dargestellt wird.
Lösung:	Formel für die bedingte Formatierung: =LINKS(A1;1)<>LINKS(A65536;1)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datumsbereich in Abhängigkeit von 3 Kriterien markieren

Problem:	In Zeile 1 befinden sich Kopfzeilen, unterhalb davon Datumsreihen. In den Zellen H1:H3 befinden sich als Kriterien Kopfzeilenwert, Start- und Enddatum. Bei Änderungen der Eintragungen soll sich die Markierung des Datumsbereiches den geänderten Daten anpassen.
Lösung:	Formel für die bedingte Formatierung: =UND(A\$1=\$H\$1;A2>=\$H\$2;A2<=\$H\$3)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wochenenden und Feiertage in Jahresliste markieren

Problem:	Wie kann ich in einer Monatsliste die Wochenenden und Feiertage so markieren, dass sich die Markierung bei Eingabe eines anderen Jahres oder Monats automatisch anpasst?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktuelle Geburtstage beim Öffnen der Arbeitsmappe anzeigen

Problem:	Wie kann ich erreichen, daß mir beim Öffnen einer Arbeitsmappe die Zellen mit aktuellen Geburtstagen gelb und die mit Werten bis eine Woche vorher als Vorwarnung blau hinterlegt werden?
Lösung:	Formel A für die bedingte Formatierung: =UND(MONAT(D1)=MONAT(HEUTE()));TAG(D1)=TAG(HEUTE()))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Zellen in Spalte A ohne Eintrag gelb hinterlegen

Problem:	Wie kann ich erreichen, dass alle Zellen in Spalte A, in denen kein Wert steht, gelb hinterlegt sind. Wird ein Wert eingetragen, soll die Hintergrundfarbe gelöscht werden.
Lösung:	Formel für die bedingte Formatierung: =ISTLEER(A1)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelt vorkommende Werte markieren

Problem:	Wie kann ich alle in einer Spalte doppelt vorkommenden Werte markieren lassen?
Lösung:	Formel für die bedingte Formatierung in A2: =ZÄHLENWENN(A:A;A2)>1
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über bedingte Formatierung Hintergrundfarbe bei Zelleingabe ändern

Problem:	Bei der Eingabe einer 7 in Spalte A sollen die sieben Zellen rechts der Eingabezelle mit der Hintergrundfarbe gelb belegt werden.
Lösung:	Formel für die bedingte Formatierung: <code>= \$A1=7</code>
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über die bedingte Formatierung den Minimalwert markieren

Problem:	Wie kann ich über die bedingte Formatierung in Spalte A einer Tabelle den jeweilig kleinsten Wert markieren lassen?
Lösung:	Formel für die bedingte Formatierung: =A1=MIN(A:A)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tagesliste eines Monats nach Eingaben aktualisieren und formatieren

Problem:	Wie kann ich die Tagesliste eines Monats nach Eingabe der Monatsoder Jahreszahl automatisch aktualisieren und Samstage und Sonntage unterschiedlich markieren lassen?
Lösung:	Nur anhand einer Beipsielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bedingte Formatierung auslesen

Problem:	Wie kann ich die Bedingungen der bedingten Formatierung und die Auswirkung auf die formatierte Zelle auslesen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FormCond() Dim iFormula As Integer, iColorA As Integer Dim iColorB As Integer With Range("A1").FormatConditions(1) iFormula = .Formula1 iColorA = .Font.ColorIndex iColorB = Range("A1").Font.ColorIndex If iColorB = -4105 Then iColorB = "'Automatisch'" If Range("A1").Value = iFormula Then MsgBox "Bedingung '" & iFormula & "' ist erfüllt" _ & vbCrLf & "Die aktuelle Schriftfarbe ist " & iColorA Else MsgBox "Bedingung '" & iFormula & "' ist nicht erfüllt" _ & vbCrLf & "Die aktuelle Schriftfarbe ist " & iColorB End If End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellfarbe in verknüpfter Zelle ändern

Problem:	Wie kann ich in einer verknüpften Zelle die Zellfarbe ändern, wenn in der Quellzelle ein neuer Wert eingegeben wird? Quell- und Zielzelle befinden sich in verschiedenen Arbeitsmappen.
Lösung:	Formel in der bedingten Formatierung: =Quellzelle>10
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeitstrahl

Problem:	Wie kann ich veranlassen, daß sich nach Eingabe eines Datums die Zellinnenfarbe der nachfolgenden Zelle entsprechend der Anzahl der Tage ändert?
Lösung:	Formel für die bedingte Formatierung: =A1<=\$A\$2
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinnenfarbe bei Eingabe eines Sonntags verändern

Problem:	Wie kann ich veranlassen, daß sich die Zellinnenfarbe bei Eingabe eines Sonntagsdatums automatisch verändert?
Lösung:	Formel in der bedingten Formatierung: =WOCHENTAG(A1)=1
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Farbanpassung nach Eingabewert

Problem:	Wie kann ich in Abhängigkeit von Eintragungen in Spalte A in Spalte B die Schriftfarbe verändern?
Lösung:	Formel für die bedingte Formatierung: =A1="Lieferant A";
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Minimalwerte je Zeile markieren

Problem:	Wie kann ich die Zellen mit den Minimalwerten je Zeile mit einem roten Hintergrund belegen?
Lösung:	Formel für die bedingte Formatierung: =UND(A1=MIN(1:1);A1>0)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen bei Minuswerten schraffieren

Problem:	Wie kann ich bei der Zahleneingabe bei Minuswerten die Zelle schraffieren und die Schraffierung bei Pluswerten wieder aufheben?
Lösung:	Formel für die bedingte Formatierung: =A1<0
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schriftfarbe rot, wenn Datum > Heute

Problem:	Wie kann ich die Schriftfarbe einer Zelle automatisch auf rot setzen lassen, wenn das darin enthaltene Datum größer als das aktuelle Datum ist?
Lösung:	Formel für die bedingte Formatierung: =UND(A1>HEUTE();ISTZAHL(A1))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gemäss Gültigkeits-Listenauswahl Beträge addieren

Problem:	In Zelle C8 wird über eine Formel die Addition der Werte aus Tabelle Daten gem. Name, Start- und Endmonat ermittelt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach DropDown-Auswahl Währungswerte umstellen

Problem:	Über Dropdownfeld-Auswahl in Zelle E2 werden die Werte aus der ausgeblendeten Spalte A wahlweise in EURO oder USD dargestellt.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Für ausgewählten Bereich per VBA Gültigkeitskriterien setzen

Problem:	Über Schaltflächenklick sollen in Spalte A für den ausgewählten Bereich Gültigkeitsbedingungen so gesetzt werden, dass nur Zahlen > 0 und bis einer Nachkommastelle akzeptiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetValidation() Dim sStart As String sStart = Selection.Cells(1).Address(False, False) With Selection.Validation .Delete .Add Type:=xlValidateCustom, AlertStyle:=xlValidAlertStop, Operator:= _ xlBetween, Formula1:= _ "=AND(ROUND(MOD(" & sStart & ",0.1),12)=0," & sStart & ">0)" .IgnoreBlank = True .InCellDropdown = True End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur positive Ganzzahlen akzeptieren

Problem: In Spalte A sollen nur positive Ganzzahlen als Eingabe akzeptiert werden.

Lösung: Bei Gültigkeit folgende Formel eintragen: =UND(REST(A1;1)=0;A1>0)

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textlänge in einer Spalte auf 40 Zeichen begrenzen

Problem:	Die Textlänge in Spalte A ist über Menü DATEN / GÜLTIGKEIT auf 40 Zeichen begrenzt worden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Begrenzung der Textlänge bei Eingaben

Problem:	In Spalte A soll die Textlänge auf 35 Zeichen begrenzt werden. Lösung über Menü DATEN / GÜLTIGKEIT
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamische Gültigkeitsliste

Problem:	In Zelle B2 soll eine Gültigkeitsliste angezeigt werden, deren Werte in Abhängigkeit zu dem in Zelle A2 eingetragenen Wert stehen.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabebegrenzung beim Addieren

Problem:	In Zelle G1 befinde sich eine Addition der Zellen E1 und F1. Die Eingaben in E1 und F1 sollen auf einen Maximalwert von 5 in Zelle G1 begrenzt werden. Lösung über Menü DATEN / GÜLTIGKEIT.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Calculate() If Range("G3").Value > 5 Then Application.Undo End If End Sub StandardModule: Modul1 Function Undo_mk() Application.Undo End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Außer Kommata und Semikoli keine anderen Zeichen doppelt eingeben

Problem:	In Spalte C dürfen außer Kommata und Semikoli keine Zeichen doppelt eingegeben werden. Lösung über Menü DATEN / GÜLTIGKEIT.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kein Eintrag in F oder T, wenn in Spalte C ein C steht

Problem:	Wenn in Spalte C in B steht, darf in Spalte F weder ein T noch ein F akzeptiert werden. Lösung über Menü DATEN / GÜLTIGKEIT.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vervollständigung einer Adresse nach Gültigkeitsfeld-Auswahl

Problem: Nach Auswahl eines Namens in einem Gültigkeitsfeld soll sich die restliche Adresse über Formeln aktualisieren. Die Gültigkeitsauswahl soll bei Neueinträgen angepasst werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    Dim iRow As Integer
    iRow = Cells(Rows.Count, 1).End(xlUp).Row
    Range("A2:A" & iRow).Name = "Adressen"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Keine doppelten Werte und nur Werte mit 10 Zeichen akzeptieren

Problem:	Die Werte in Spalte A dürfen sich nicht wiederholen und müssen 10 Zeichen lang sein. Lösung über Menü DATEN / GÜLTIGKEIT / BENUTZERDEFINIERT
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kein Eintrag, bevor ein Wert in Zelle A1 enthalten ist

Problem:	Wie kann ich Zelleingaben in Spalte A verhindern, solange in Zelle A1 noch kein Wert steht?
Lösung:	Formel bei Gültigkeit: <code>=A\$1<>"&quot;</code>
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Doppelte Eingabe von Werten verhindern

Problem:	Wie kann ich verhindern, dass in Spalte A Werte eingegeben werden, die dort in einer anderen Zelle bereits vorhanden sind? Außerdem soll die Eingaben von Zeichenfolgen mit einem "y" verhindert werden.
Lösung:	Formel für Gültigkeit: =UND(ZÄHLENWENN(A:A;A1)=1;ISTFEHLER(FINDEN("y";A1)))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gültigkeitsauswahl über VBA auslesen

Problem: Wie kann ich die Auswahl aus einem Gültigkeits-DropDown-Feld per VBA auslesen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Calculate()
    If ActiveSheet.Name <> Me.Name Then Exit Sub
    If Intersect(Range("A1:A10"), ActiveCell) Is Nothing Then Exit Sub
    MsgBox "Auswahl: " & ActiveCell.Value
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einmalige Eingabe zulassen

Problem:	Wie kann ich in die Zellen eines Arbeitsblattes nur einmalige Eingaben zulassen?
Lösung:	Formel in Menü DATEN / GÜLTIGKEIT: =ANZAHL2(A1:IV65536)=1
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wechselweise den Eintrag einer 1 verhindern

Problem:	Wie kann ich in XL8 verhindern, daß in einer Zelle in Spalte A eine "1" eingetragen wird, wenn diese in der danebenliegenden Zelle von Spalte B schon vorhanden ist? Dies soll auch wechselweise gelten.
Lösung:	Formel über Menü DATEN / GÜLTIGKEIT: =B1<>1 bzw. =A1<>1
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrfachnennung melden

Problem:	In eine Tabelle gebe ich Autokennzeichen ein, wobei Mehrfacheintragungen möglich sind. Ist der Eintrag schon einmal vorhanden, soll eine MsgBox die Anzahl dieser Eintragungen nennen. Wie löse ich das über VBA-Code?
Lösung:	Formel bei DATEN / GÜLTIGKEIT: =ZÄHLENWENN(A:A;A1)=1
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nichtnumerische Eingaben verhindern

Problem:	Wie kann ich verhindern, daß der Anwender nur Ganzzahlen größer als 0 eingibt?
Lösung:	Geben Sie die Beschränkung über Menü DATEN / GÜLTIGKEIT... ein
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Berechnung der Druckverluste in Rohrnetzen

Problem:	Berechnung der Druckverluste in Rohrnetzen (Luft, Heizung) nach einer Gleichung von Coolbrook-White.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beispiel für Zielwertsuche

Problem:	Wie ist die Zielwertsuche zu handhaben?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen verbinden und senkrecht ausrichten

Problem:	Über festgelegte Variablen für den Zeilenindex sollen Zellen verbunden und mit senkrecht ausgerichtetem Text versehen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Verbinden() Dim iRowL As Integer, iRowC As Integer iRowL = 30 iRowC = 20 With Range(Cells(iRowL - iRowC, 1), Cells(iRowL, 1)) .Merge .Value = "Monatsaufstellung" .Orientation = 90 .Font.Bold = True .VerticalAlignment = xlCenter End With Columns(1).AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fehlermeldung #NV ausblenden

Problem:	Wie kann man den Fehlerwert #NV so ausblenden, daß die übrigen Fehlerwerte (z.B. #WERT) sichtbar bleiben?
Lösung:	Lösung über Menü FORMAT / BEDINGTE FORMATIERUNG / Formel: =UND(ISTFEHLER(A1); ISTFEHL(A1)=FALSCH)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Automatische Datumformatierung unterdrücken

Problem:	Wenn ich in einem Bereich im Zahlenformat "Text" formatierter Zellen über das Menü BEARBEITEN / ERSETZEN die Kommata durch Punkte ersetze, wird ein Teil der Zahlen als Datum ("Apr 98") formatiert. Wie kann ich das verhindern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ReplaceComma() Dim rng As Range For Each rng In Range("A1:C2") If InStr(rng.Value, ",") <> 0 Then rng.Value = Application.Substitute(rng, ",", ".") End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verhinderung der Formatumwandlung durch Excel

Problem:	Wenn ich Zeichenfolgen ersetzen lassen, nimmt Excel manchmal eine automatische Datum-Formatierung vor. Kann das verhindert werden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NoFormatting() On Error GoTo ERRORHANDLER ActiveSheet.TransitionFormEntry = True Cells.Replace _ What:="20", _ Replacement:="19", _ LookAt:=xlPart, _ SearchOrder:=xlByRows, _ MatchCase:=False ERRORHANDLER: ActiveSheet.TransitionFormEntry = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Linien setzen und löschen

Problem: Einem ausgewählten Bereich sollen Linien hinzugefügt bzw. vorhandene gelöscht werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub LinienEinAus()
    With Selection.Borders
        If .LineStyle = xlContinuous Then
            .LineStyle = xlNone
        Else
            .LineStyle = xlContinuous
            .Weight = xlThin
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Formate jenseits der letzten benutzten Zelle löschen

Problem: Über den ShortCut STRG+q sollen alle Formate in den Spalten rechts und den Zeilen unterhalb der letzten benutzten Zelle gelöscht und die letzte benutzte Zelle ausgewählt werden.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben und über STRG+q starten.

VBA-Code: StandardModule: Modul1

```
Sub ClearFormats()  
    Dim rng As Range  
    Set rng = RealLastCell(ActiveSheet)  
    Range(rng.Offset(0, 1), Cells(1, Columns.Count)).EntireColumn.Delete  
    Rows(CStr(rng.Row + 1) & ":" & Rows.Count).Delete  
    rng.Select  
End Sub  
  
Function RealLastCell(TheSheet As Worksheet) As Range  
    Dim ExcelLastCell As Range  
    Dim Row%, Col%, LastRowWithData%, LastColWithData%  
    Application.ScreenUpdating = False  
    Set ExcelLastCell = TheSheet.Cells.SpecialCells(xlLastCell)  
    LastRowWithData = ExcelLastCell.Row  
    Row = ExcelLastCell.Row  
    Do While Application.CountA(TheSheet.Rows(Row)) = 0 And Row <> 1  
        Row = Row - 1  
    Loop  
    LastRowWithData = Row  
    LastColWithData = ExcelLastCell.Column  
    Col = ExcelLastCell.Column  
    Do While Application.CountA(TheSheet.Columns(Col)) = 0 And Col <> 1  
        Col = Col - 1  
    Loop  
    LastColWithData = Col  
    Set RealLastCell = TheSheet.Cells(Row, Col)  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen verbinden, ohne dass Werte verlorengehen

Problem:	Die Zellen aus Spalte A und B sollen verbunden werden, ohne dass die Werte aus Spalte B verlorengehen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Verbinden() Dim iRow As Integer iRow = 1 Do Until IsEmpty(Cells(iRow, 1)) Cells(iRow, 1) = Cells(iRow, 1) & " - " & Cells(iRow, 2) Cells(iRow, 2).ClearContents Range(Cells(iRow, 1), Cells(iRow, 2)).Merge iRow = iRow + 1 Loop Columns(1).AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spalten bedingt summieren und Zellen formatieren

Problem: Wenn in eine Zelle der ersten Zeile einer Spalte eine 12 eingegeben wird, soll eine Addition der Spalte erfolgen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    Dim iRow As Integer, iCol As Integer
    If IsEmpty(Target) Then Exit Sub
    If Target.Row = 1 And Target.Value = 12 Then
        Application.EnableEvents = False
        On Error GoTo ERRORHANDLER
        iCol = Target.Column
        iRow = Cells(Rows.Count, iCol).End(xlUp).Row
        Range(Cells(2, iCol), Cells(iRow, iCol)) _
            .NumberFormat = "#,##0.00"
        Cells(iRow + 1, iCol).Formula = _
            "=sum(R2C" & iCol & ":R" & iRow & "C" & iCol & ")"
    End If
ERRORHANDLER:
    Application.EnableEvents = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auslesen der aktuell in der Zwischenablage vorhandenen Formate

Problem:	Die aktuell in der Zwischenablage vorhandenen Formate sollen ausgelesen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: Modul1 <pre>Sub TestClipboard() Dim vaFormats As Variant, dFormat As Variant vaFormats = Application.ClipboardFormats For Each dFormat In vaFormats Select Case dFormat Case xlClipboardFormatBIFF MsgBox "xlClipboardFormatBIFF (" & dFormat & ")" Case xlClipboardFormatBIFF2 MsgBox "xlClipboardFormatBIFF2 (" & dFormat & ")" Case xlClipboardFormatBIFF3 MsgBox "xlClipboardFormatBIFF3 (" & dFormat & ")" Case xlClipboardFormatBIFF4 MsgBox "xlClipboardFormatBIFF4 (" & dFormat & ")" Case xlClipboardFormatBinary MsgBox "xlClipboardFormatBinary (" & dFormat & ")" Case xlClipboardFormatBitmap MsgBox "xlClipboardFormatBitmap (" & dFormat & ")" Case xlClipboardFormatCGM MsgBox "xlClipboardFormatCGM (" & dFormat & ")" Case xlClipboardFormatCSV MsgBox "xlClipboardFormatCSV (" & dFormat & ")" Case xlClipboardFormatDIF MsgBox "xlClipboardFormatDIF (" & dFormat & ")" Case xlClipboardFormatDspText MsgBox "xlClipboardFormatDspText (" & dFormat & ")" Case xlClipboardFormatEmbeddedObject MsgBox "xlClipboardFormatEmbeddedObject (" & dFormat & ")" Case xlClipboardFormatEmbedSource MsgBox "xlClipboardFormatEmbedSource (" & dFormat & ")" Case xlClipboardFormatLink MsgBox "xlClipboardFormatLink (" & dFormat & ")" Case xlClipboardFormatLinkSource MsgBox "xlClipboardFormatLinkSource (" & dFormat & ")" Case xlClipboardFormatLinkSourceDesc MsgBox "xlClipboardFormatLinkSourceDesc (" & dFormat & ")" Case xlClipboardFormatMovie MsgBox "xlClipboardFormatMovie (" & dFormat & ")" Case xlClipboardFormatNative MsgBox "xlClipboardFormatNative (" & dFormat & ")" Case xlClipboardFormatObjectDesc MsgBox "xlClipboardFormatObjectDesc (" & dFormat & ")" Case xlClipboardFormatObjectLink MsgBox "xlClipboardFormatObjectLink (" & dFormat & ")" Case xlClipboardFormatOwnerLink MsgBox "xlClipboardFormatOwnerLink (" & dFormat & ")" Case xlClipboardFormatPICT MsgBox "xlClipboardFormatPICT (" & dFormat & ")" Case xlClipboardFormatPrintPICT MsgBox "xlClipboardFormatPrintPICT (" & dFormat & ")" Case xlClipboardFormatRTF MsgBox "xlClipboardFormatRTF (" & dFormat & ")" Case xlClipboardFormatScreenPICT MsgBox "xlClipboardFormatScreenPICT (" & dFormat & ")" Case xlClipboardFormatStandardFont MsgBox "xlClipboardFormatStandardFont (" & dFormat & ")" Case xlClipboardFormatStandardScale MsgBox "xlClipboardFormatStandardScale (" & dFormat & ")" Case xlClipboardFormatSYLK MsgBox "xlClipboardFormatSYLK (" & dFormat & ")" Case xlClipboardFormatTable MsgBox "xlClipboardFormatTable (" & dFormat & ")" Case xlClipboardFormatText MsgBox "xlClipboardFormatText (" & dFormat & ")" Case xlClipboardFormatToolFace MsgBox "xlClipboardFormatToolFace (" & dFormat & ")" Case xlClipboardFormatToolFacePICT MsgBox "xlClipboardFormatToolFacePICT (" & dFormat & ")" Case xlClipboardFormatVALU MsgBox "xlClipboardFormatVALU (" & dFormat & ")" Case xlClipboardFormatWK1 MsgBox "xlClipboardFormatWK1 (" & dFormat & ")" Case Else MsgBox "Zwischenablage ist leer (" & dFormat & ")" End Select Next End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Differenz aus Start- und Endzeit in Sekunden

Problem:	Wie kann ich die Differenz aus einer Start- und Endzeit in Sekunden darstellen lassen?
Lösung:	Formatieren Sie die Zelle benutzerdefiniert als [ss]
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Jahreskalender und bedingte Formatierung

Problem:	Wie kann ich ein Jahreskalenderblatt anlegen und Wochenenden, den aktuellen Tag und Feiertage farblich markieren, ohne VBA einzusetzen? Bei Eingabe eines neuen Jahres soll eine Neuberechnung erfolgen.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eingabeerleichterung bei Zeitformaten

Problem:	Wie kann ich Excel dazu bewegen, eine Eingabe in Form von hhmmss als hh:mm:ss (Stunden:Minuten:Sekunden) zu erkennen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Excel.Range) Dim sTxt As String If Target.Column <> 1 Then Exit Sub On Error GoTo ERRORHANDLER Application.EnableEvents = False sTxt = Target.Value If sTxt Like "#####" Then Target.NumberFormat = "hh:mm:ss" Target.Value = TimeSerial(_ Left(sTxt, 2), _ Mid(sTxt, 3, 2), _ Right(sTxt, 2)) Else Target.NumberFormat = "General" End If ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datumsformatkonvertierung (von Heinrich Stuerzl)

Problem: Wie kann ich importierte Datumsformate auf das Datumsformat meines Rechners umstellen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function GetDate(sDate As String) As String
    Dim iDay As Integer, iMonth As Integer, iYear As Integer
    Dim sDlm As String
    If sDate Like "##?##?#####" = False Then
        Beep
        GetDate = "Kein gültiger Datumswert!"
        Exit Function
    End If
    iDay = CInt(Mid(sDate, 1, 2))
    iMonth = Mid(sDate, 4, 2)
    iYear = Mid(sDate, 7, 4)
    sDlm = Application.International(xlDateSeparator)
    Select Case Application.International(xlDateOrder)
        Case 0
            GetDate = Format(CDate(iMonth & sDlm & _
                & iDay & sDlm & iYear), "Short Date")
        Case 1
            GetDate = Format(CDate(iDay & sDlm & _
                iMonth & sDlm & iYear), "Short Date")
        Case 2
            GetDate = Format(CDate(iYear & sDlm & iMonth & _
                sDlm & iDay), "Short Date")
    End Select
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopfzelle mit aktivem Autofilter markieren

Problem:	Wie kann ich die Kopfzelle der Spalten, in denen der Autofilter aktiv ist, markieren?
Lösung:	Den VBA-Code in ein Standardmodul eintragen und der Formel in der bedingten Formatierung zuweisen.
VBA-Code:	<pre>StandardModule: Modul1 Function FilterStatus() As Boolean Application.Volatile If ActiveSheet.AutoFilter.Filters(Application.Caller.Column).On Then FilterStatus = True End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelle mit kleinsten Wert markieren

Problem:	Wie kann ich bei Eingaben in Spalte D jedesmal den kleinsten Wert in dieser Spalte mit einer gelben Hintergrundfarbe belegen?
Lösung:	Formel in der bedingten Formatierung: =A1=MIN(A:A)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zelle mit Maximalwert markieren

Problem:	Wie kann ich die Hintergrundfarbe der Zelle mit dem Maximalwert aus Spalte A gelb markieren?
Lösung:	Formel für die bedingte Formatierung: =A1=MAX(A:A)
VBA-Code:	<pre>StandardModule: mHWH Sub MaxMark() With Application Cells(.Match(.Max(Columns(1), 1), _ Columns(1), 0), 1).Interior _ .ColorIndex = 3 End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinnenfarben über VBA festlegen

Problem:	Wie kann ich per VBA von der Farbpalette abweichende Zellinnenfarben festlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SetColor() ActiveWorkbook.Colors(41) = RGB(112, 144, 122) Range("A1:A20").Interior.ColorIndex = 41 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schriftart und -größe eines Kommentars festlegen

Problem:	Wie kann ich die Schriftart und -größe eines Kommentars per VBA festlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub KommentarSchrift() Dim cmt As Comment Set cmt = ActiveCell.AddComment("Mein Kommentar") With cmt.Shape.TextFrame.Characters.Font .Name = "Courier" .Size = 12 .Bold = False End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Besuchsliste für Kunden verschiedener Kategorien führen

Problem:	Wie kann ich eine Besuchsliste für Kunden 3 verschiedener Kategorien führen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle in Excel verfügbaren Schriften listen

Problem:	Wie kann ich in Spalte A alle in Excel verfügbaren Schriften auflisten und in der entsprechenden Schriftart formatieren lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SchriftAuslesen() Dim cnt As CommandBarControl Dim iCounter As Integer Application.ScreenUpdating = False Set cnt = Application.CommandBars.FindControl(ID:=1728) For iCounter = 1 To cnt.ListCount With Cells(iCounter, 1) .Value = cnt.List(iCounter) .Font.Name = cnt.List(iCounter) End With Next iCounter Columns(1).AutoFit Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schriftgröße beim Kopieren verdoppeln

Problem:	Wie kann ich beim Kopieren über VBA die Schriftgröße verdoppeln?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Uebertragen() Dim iSize As Integer iSize = Range("A1").Font.Size * 2 With Worksheets("Tabelle2") Range("A1").Copy .Range("A1") .Range("A1").Font.Size = iSize .Select End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spaltenbreite anpassen

Problem:	Wie kann ich die Spaltenbreite nach dem Zellwert einstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FitColumns() Dim intCounter As Integer For intCounter = 1 To 3 Columns(intCounter).ColumnWidth = _ Cells(1, intCounter).Value Next intCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Formatvorlagen löschen

Problem:	Wie kann ich alle Formatvorlagen - außer "Standard" - über VBA löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DeleteStyles() Dim sty As Style For Each sty In ThisWorkbook.Styles If sty.BuiltIn = False Then sty.Delete Next sty End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wasserzeichen in Tabelle einfügen

Problem:	Wie kann ich ein Wasserzeichen in eine Tabelle einfügen?
Lösung:	Fügen Sie den Wasserzeichentext in ein auf halbtransparent gesetztes Wordartobjekt ein.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahlen-Formatierungs-Dialog aus UserForm aufrufen

Problem: Über eine UserForm-Schaltfläche wird der Zahlenformatierungsdialog aufgerufen und der Bereich C1:D12 gemäß der Formatauswahl formatiert.

Lösung: Geben Sie den nachstehenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: frmFormat

Dim sFormat As String

Private Sub cmdCancel_Click()
    ActiveCell.NumberFormat = sFormat
    Unload Me
End Sub

Private Sub cmdFormat_Click()
    sFormat = ActiveCell.NumberFormat
    Application.Dialogs(xlDialogFormatNumber).Show
End Sub

Private Sub cmdOK_Click()
    Range("C1:D12").NumberFormat = ActiveCell.NumberFormat
    ActiveCell.NumberFormat = sFormat
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Range("A1").Select
End Sub

StandardModule: Modul1

Sub DialogAufruf()
    frmFormat.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle gem. hinterlegten Zahlenformaten formatieren

Problem:	Die Spalten dieser Tabelle sollen gemäß den Zahlenformat-Vorgaben aus Tabelle "Formate" formatiert werden.
Lösung:	In Tabelle "Formate" stehen in der ersten Zeile die Zahlenformate, in der zweiten die jeweiligen Spaltennummern.
VBA-Code:	<pre>StandardModule: basMain Sub Formatieren() Dim wks As Worksheet Dim iCol As Integer Dim sCol As String Dim sTxt As String Set wks = Worksheets("Formate") iCol = 1 Do Until IsEmpty(wks.Cells(1, iCol)) sTxt = wks.Cells(2, iCol) Do Until InStr(sTxt, ",") = 0 sCol = Left(sTxt, InStr(sTxt, ",") - 1) Columns(CInt(sCol)).NumberFormat = wks.Cells(1, iCol).Value sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, ",")) Loop Columns(CInt(sTxt)).NumberFormat = wks.Cells(1, iCol).Value iCol = iCol + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahlenformatierung mit Punkt nach der 1. Ziffer

Problem:	Wie kann ich Zellen so formatieren, daß nach der ersten Ziffer ein Punkt gesetzt wird? Beispiele: 1 = 1., 10 = 1.0, 123455 = 1.234455
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FormatEin() Worksheets("Data").OnEntry = "Formatting" End Sub Sub FormatAus() Worksheets("Data").OnEntry = "" End Sub Sub Formatting() Application.Caller.NumberFormat = "0\." & _ String(Len(CStr(Application.Caller)) - 1, "0") End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vor eine Aufzählung Punkte setzen

Problem:	Wie kann ich vor eine Auswahl von Zellen Aufzählungspunkte setzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub PunkteSetzen() Dim rng As Range For Each rng In Selection.Cells If Not IsEmpty(rng) Then rng.Value = " " & Chr(163) & " " & rng.Value End If rng.Characters(Start:=4, Length:=1) _ .Font.Name = "Wingdings" Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenumbruch in Formel erzwingen

Problem:	Wie kann ich einen Zeilenumbruch in einer Formel erzwingen? Ich möchte Adressdaten, die sich in den Spalten "Anrede", "Vorname", "Name", "Straße", "PLZ" und "Ort" (Spalten A-F) befinden, so verknüpfen, daß ein ordnungsgemäßes Adressfeld entsteht.
Lösung:	Formel in Blatt Tabelle1: =A1&ZEICHEN(10)&B1&" "&C1&ZEICHEN(10)&D1&ZEICHEN(10)&ZEICHEN(10)&E1&" "&F1
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anfügen des Zeichens für Durchmesser

Problem:	Wie kann ich an einen vorhandenen Text das Zeichen für "Durchmesser" anhängen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DiaMeters() Dim rng As Range For Each rng In Selection.Cells rng.Value = rng.Value & " " & Chr(198) rng.Characters(Start:=Len(rng.Value), _ Length:=1).Font.Name = "Symbol" Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilenhöhe bei verbundenen Zellen anpassen

Problem: Wenn ich Zellen mit Zeilenumbrüchen verbinde, passt sich die Zeilenhöhe nicht an. Wie kann ich die Anpassung erreichen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub AutoFitMergedCellRowHeight()  
    Dim CurrentRowHeight As Single, MergedCellRgWidth As Single  
    Dim CurrCell As Range  
    Dim ActiveCellWidth As Single, PossNewRowHeight As Single  
    Dim iX As Integer  
    If ActiveCell.MergeCells Then  
        With ActiveCell.MergeArea  
            If .Rows.Count = 1 And .WrapText = True Then  
                Application.ScreenUpdating = False  
                CurrentRowHeight = .RowHeight  
                ActiveCellWidth = ActiveCell.ColumnWidth  
                For Each CurrCell In Selection  
                    MergedCellRgWidth = CurrCell.ColumnWidth + _  
                        MergedCellRgWidth  
                    iX = iX + 1  
                Next  
                MergedCellRgWidth = MergedCellRgWidth + (iX - 1) * 0.71  
                .MergeCells = False  
                .Cells(1).ColumnWidth = MergedCellRgWidth  
                .EntireRow.AutoFit  
                PossNewRowHeight = .RowHeight  
                .Cells(1).ColumnWidth = ActiveCellWidth  
                .MergeCells = True  
                .RowHeight = IIf(CurrentRowHeight > PossNewRowHeight, _  
                    CurrentRowHeight, PossNewRowHeight)  
            End If  
        End With  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einfügen von Werten in Abhängigkeit eines Blattnamens

Problem:	In den Bereich A15:D22 sollen die Daten aus dem in Zelle B2 benannten Tabellenblatt, Bereich A1:D8, eingefügt werden.
Lösung:	Nur anhand einer Beispieldarstellungsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Erhöhen der Positionsnummer bei Listeneintrag

Problem:	Bei neuen Eingaben in Spalte C soll sich die Positionsnummer in Spalte B für die gesamte Wertetabelle jeweils um 1 erhöhen.
Lösung:	Folgende Formel: =WENN(C3<>"";ANZAHL2(INDIREKT("&c1:c&"&ZEILE()-1));&"")
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Zeileneintrag Anzeige von "erledigt"

Problem:	Wie kann ich erreichen, dass in Spalte D ein Erledigt-Vermerk erscheint, wenn in der jeweiligen Zeile ein Eintrag erfolgt?
Lösung:	Folgende Formel: =WENN(ANZAHL2(A2:C2)>0;"erledigt";"")
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Letzte Zelle einer Spalte mit Inhalt aus geschlossener Arbeitsmappe

Problem:	Wie kann ich die Anzahl der Zellen mit Inhalt in einer im gleichen Verzeichnis liegenden Arbeitsmappe feststellen?
Lösung:	Folgende Formel: =ANZAHL2('C:\CD0209\Beispiele\[000197v.xls]Text'!\$A:\$A)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suche von Korrespondenzwerten in zwei Spalten

Problem:	Die Daten aus Spalte C sollen in den Spalten A und B gesucht und der Korrespondenzwert soll in Spalte D ausgegeben werden.
Lösung:	Datstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Jahre, Monate und Tage von Zeiträumen ermitteln

Problem:	Über die nicht dokumentierte DATEDIF-Funktion werden im Bereich D1:D3 die Jahre, Monate und Tage von Zeiträumen ermittelt.
Lösung:	Formel zur Ermittlung der Jahre: =DATEDIF(A1;B1;"y")
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBA-DateDiff und Excel-DATEDIF

Problem:	Die VBA-DateDiff-Funktion entspricht in ihren Ergebnissen nicht der undokumentierten Excel-DATEDIF-Funktion. Wie kann ich die Letztere in VBA einbinden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DateSpan() Dim iVBA As Integer iVBA = DateDiff(_ interval:="yyyy", _ date1:=Range("A1").Value, _ date2:=Range("B1").Value) With Range("IV1") .Formula = "=DateDif(A1,B1,""y"")" MsgBox prompt:="Excel: " & .Value & " Jahre" .ClearContents End With MsgBox prompt:="VBA: " & iVBA & " Jahre" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Julianisches Datum

Problem:	Wie ermittle ich das Julianische Datum in der Schreibweise "yyddd"?
Lösung:	Diese Formel ergibt das Julianische Datum des aktuellen Tages: =RECHTS(JAHR(HEUTE());2)&DATEDIF(DATUM(JAHR(HEUTE());1;1)-1;HEUTE();"d")
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Umrechnung von Monats- in Jahreswerte

Problem:	Die Formeln in Spalte C sollen die Monatswerte aus Spalte B in Jahreswerte umrechnen, wobei das Jahr aus F1 unter Berücksichtigung der Schaltjahre zugrundegelegt werden soll.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Formel zur Ermittlung von Ostersonntag

Problem:	Wie kann der Ostersonntag eines beliebigen Jahres ohne VBA-Einsatz ermittelt werden?
Lösung:	Wie kann der Ostersonntag eines beliebigen Jahres ohne VBA-Einsatz ermittelt werden?
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einem Datum eine bestimmte Anzahl von Jahren hinzufügen

Problem:	Dem Datum in Zelle A2 soll die Anzahl der Jahre aus Zelle B2 zugefügt werden.
Lösung:	Folgende Formel: =DATUM(JAHR(A2)+B2;MONAT(A2);TAG(A2))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Darlehensverzinsung bei variablen Rückzahlungsraten

Problem:	Die Darlehensverzinsung bei variablen Rückzahlungsterminen und -raten soll ermittelt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auflistung der Kalenderwochen eines variablen Jahres

Problem:	Die Kalenderwochen von/bis der einzelnen Monate sollen gelistet werden, wobei das Jahr - Zelle B1 - variabel gehalten werden soll. Kein Einsatz von VBA.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte mit begrenzter Haltbarkeit

Problem:	Die Formel in Zelle E2 soll nur bis zum 31.10.2001 > 0 bringen. Test durch Umstellung der Systemzeit. Danach die Formel kurz mit F2 editieren.
Lösung:	Lösung über eine Formel im verborgenen Namen INTEREST. Formel:
VBA-Code:	<pre>StandardModule: Modul1 Sub NamenEin() For Each nme In ThisWorkbook.Names nme.Visible = True Next nme End Sub Sub NamenAus() For Each nme In ThisWorkbook.Names nme.Visible = False Next nme End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Berechnung von Zeitspanne und Mittelwert

Problem:	Wie kann ich eine Zeitspanne und den daraus resultierenden Mittelwert je Stunde errechnen? Der Endzeitpunkt kann nach Mitternacht liegen.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Varianten zur Ermittlung des Wochentages eines Datums

Problem:	Auf welche Arten kann ich den Wochentag eines Datums anzeigen lassen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Lohnberechnung mit Zuschlagsätzen

Problem:	Wie baue ich eine Tabelle auf, die eine automatische Lohnberechnung mit Berücksichtigung von Zuschlagsätzen vornimmt?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Restschuld unter Berücksichtigung der Zinseszinsberechnung

Problem:	Wie kann ich mir eine Tabelle aufbauen, die bei festem Zinssatz und festen monatlichen Rückzahlungen eine monatliche Restschuldberechnung unter Berücksichtigung der angefallenen Zinsen durchführt?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anwendung von DBMAX auf Datum und Uhrzeit

Problem:	In Tabelle2 soll zum einen der Maximal-Temperaturwert eines vorgegebenen Tages und zum anderen die zugehörige Uhrzeit errechnet werden.
Lösung:	Nur anhand einer Beispieldarstellungsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Handhabung der Datenbankfunktion DBSUMME()

Problem:	Wie kann ich die Datenbankfunktion DBSUMME() handhaben, um bei der Summierung eines Datenbankbereiches mehrere Kriterien zugrunde zu legen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Formeln mit Zelladressen speichern

Problem: Wie kann ich alle Formeln eines Tabellenblattes mit den jeweils zugehörigen Zelladressen als Textdatei speichern?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub SaveFormulas()
    Dim rng As Range
    Dim iFile As Integer
    Dim sFile As String
    sFile = Application.Path & "\formulas.txt"
    iFile = FreeFile
    Open sFile For Output As iFile
    For Each rng In Range("A1").CurrentRegion.Cells
        If rng.HasFormula Then
            Print #iFile, _
                rng.Address; Tab; _
                rng.Formula, Tab; _
                rng.FormulaLocal
        End If
    Next rng
    Close iFile
    Workbooks.OpenText _
        Filename:=sFile, _
        DataType:=xlDelimited, _
        tab:=True, _
        semicolon:=False, _
        comma:=False, _
        Space:=False, _
        other:=False
    MsgBox "Weiter"
    ActiveWorkbook.Close savechanges:=False
    Kill sFile
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wahlauswertung nach Hare-Niemeyer

Problem:	Wie kann ich die Stimmenzählung nach Hare-Niemeyer in eine Excel-Tabelle integrieren
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus einer Tabelle die Daten nach Ort und Datum abgreifen

Problem:	Im Bereich B12:G13 sollen die Daten des in der Liste aus Zelle B11 ausgewählten Ortes aus der oberen Tabelle abgerufen werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fracht gem. Entfernung und Gewicht berechnen

Problem: Aufgrund der Km- und KG-Angaben soll die Fracht gem. Frachttabelle errechnet werden.

Lösung: Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopfzeilen anhand von Max / Min-Werten ermitteln

Problem:	Zu den Werten im Bereich F2:I2 werden im Bereich F4:I4 die Kopfzeilen aus der Matrix in A1:D2 geliefert.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung der jeweils letzten 5 Datensätze

Problem:	Im Bereich D2:E6 sollen die jeweils letzten 5 Datensätze aus den Spalten A:B dargestellt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Preisberechnung über Materialliste

Problem:	Aus der Kombination Bauteil, Materialcode, Anzahl und Größe soll der Preis aufgrund der Basispreise aus Blatt Materialliste ermittelt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Transponierte Werteübernahme

Problem:	Werte sollen in einem zweiten Blatt transponiert dargestellt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Preisermittlung aus Artikel- / Kundenmatrix

Problem:	Unter Angabe des Kunden und des Artikels soll ein Preis aus einer Tabelle ermittelt werden.
Lösung:	Folgende Formel: =INDEX(Kunden!A1:Z2000;VERGLEICH(B2;Kunden!A:A;0);VERGLEICH(A2;Kunden!1:1;0))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchen der Nachbarzelle der Zelle mit dem Maximalwert

Problem:	Der Wert der Zelle links neben der Zelle mit dem Maximalwert soll ermittelt werden.
Lösung:	Folgende Formel: =INDEX(A:A;VERGLEICH(MAX(B:B);B:B;0);1)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausflugszeiten aus einer Reisematrix ermitteln

Problem:	Wie kann ich nach Eingabe von zwei Kriterien aus einer Reisematrix die jeweils gültige Abfahrtszeit ermitteln?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in Abhängigkeit einer Gültigkeitslistenauswahl ermitteln

Problem:	In Zelle H10 befindet sich ein Gültigkeitsbegrenzung. In den Zellen links neben H10 sollen die zur Auswahl in H10 korrespondierenden Werte eingetragen werden. Nur ab XL9 funktionsfähig.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen in Abhängigkeit einer Punktwertung ausgeben

Problem:	In Spalte A stehen Namen, in Spalte B Punkte. Die niedrigste Punktezahl bedeutet den ersten Platz. Wie kann ich in einer anderen Spalte die Namen nach ihrer Rangfolge listen lassen?
Lösung:	Formel: =INDEX(\$A\$1:\$A\$5;VERGLEICH(KKLEINSTE(\$B\$1:\$B\$5;D1);\$B\$1:\$B\$5;0);1)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schulnote anhand einer Punktetabelle errechnen

Problem:	Wie kann ich ein Schulnote für eine gegebene Punktezahl anhand einer Punktetabelle ermitteln.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Medallien-Verteilung nach Punkten

Problem:	Wie kann ich die Medallien-Verteilung bei Schulsportfesten festlegen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopfzeilenwerte angekreuzter Zellen multiplizieren

Problem:	Wie kann ich die Kopfzeilenwerte der mit einem Kreuz versehenen Zellen mit einem in der gleichen Zeile stehenden Multiplikator multiplizieren lassen?
Lösung:	Formel: =INDEX(VERGLEICH("x";A2:F2);1)*G2
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Den jeweils letzten Wert einer Zahlenreihe als Multiplikator

Problem:	Wie kann ich den jeweils letzten Wert einer Tabellen-Zahlenreihe als Multiplikator einsetzen?
Lösung:	Folgende Formel: =INDEX(A:A;ANZAHL2(A:A);1)*B1
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Portokosten aus einer Gewichts- / Gebietsmatrix errechnen

Problem:	Eine Matrix enthält eine Gewichts- und eine Gebietsstaffelung. Wie kann ich durch Eingabe dieser beiden Faktoren die Portokosten errechnen lassen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalt in zweitem Blatt ermitteln

Problem:	Wie kann ich den Zellinhalt in einem zweiten Blatt in der gleichen Spalte aber einer anderen Zeile als die der aktiven Zelle ermitteln?
Lösung:	Folgende Formel: =INDEX(Tabelle2!1:16384;25;SPALTE()) Ermittelt wird der Wert aus Zeile 25
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle gem. Baugruppenbezeichnungen summieren

Problem:	Die Formeln in Spalte C sollen die Preise aus Spalte B gem. den Baugruppen aus Spalte A summieren.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte gem. Start- und Endmonat summieren

Problem:	Die Gradanteile per Monat sind auf Grundlage der Angaben im Bereich E1:E2 in Zelle E4 zu summieren.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus jeder 20. Zeile ermitteln

Problem:	Es sollen die Werte aus jeder 20. Zeile ermittelt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Monatsbereiche aus Folgeblättern summieren

Problem:	Aus den Folgeblättern sollen die Werte gem. den in Spalte A genannten Monate summiert werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Variablen auf mehrere Arten in Funktionen integrieren

Problem:	Die Basisformel aus D2 soll im ersten Schritt mit den Variablen aus Spalte A (Formel in D3) und im zweiten als benannte Formel (Formel in D4) erstellt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit SVERWEIS Fundstellen ab vorgegebenen Zeilen abrufen

Problem:	Aus Tabelle2, Spalte A sollen SVERWEIS-Fundstellen ab der in Spalte B genannten Zeile abgerufen werden.
Lösung:	Formel: =SVERWEIS(\$A\$1;INDIREKT(""tabelle2!a"&B2+1&";:b65536"");2;0)
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in Monats/Namen-Ergebnismatrix übernehmen

Problem:	In Zeile 1 stehen die Monate, in Spalte A die Namen. Die Formel soll aus dem Monatsblatt die Werte zum jeweiligen Namen übernehmen.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	<pre>StandardModule: Modul1 Sub BlaetterAnlegen() Dim iCounter As Integer For iCounter = 1 To 12 Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Name = Format(DateSerial(1, iCounter, 1), "mmm") Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Addition bis zum aktuellen Datum

Problem:	Wie kann ich, wenn in Spalte A das fortlaufende Datum und in Spalte B zugehörige Werte stehen, diese Werte bis zum jeweils aktuellen Datum addieren?
Lösung:	Folgende Formel: =SUMME(INDIREKT("B1:B"&VERGLEICH(HEUTE();A:A;0)))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit der Indirekt-Funktion auf geschlossene Arbeitsmappe zugreifen

Problem: Wie kann ich das Problem umgehen, dass ich weder mit der INDIREKT-Funktion noch mit VBA auf geschlossene Arbeitsmappen zugreifen kann?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Importieren()
    Dim rng As Range
    Dim sFormula As String, sPath As String
    Dim sWkb As String, sWks As String
    sPath = ThisWorkbook.Path
    sWkb = Range("F1").Value
    If Dir(sPath & "\" & sWkb) = "" Then
        Beep
        MsgBox "Testdatei wurde nicht gefunden!"
    End If
    sWks = Range("F2").Value
    sFormula = "=" & sPath & "\"
    sFormula = sFormula & "[" & sWkb & "]"
    sFormula = sFormula & sWks & "!"
    For Each rng In Selection.Cells
        rng.Formula = sFormula & rng.Address
    Next rng
    With Selection
        .Value = .Value
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Übersichtsblatt mit Angabe, welche Namen in welcher Tabelle

Problem:	Ich habe 2 Blätter mit je einer Spalte für Vor- und Nachnamen. Wie kann ich mir in einem Übersichtsblatt unter dem Tabellenblattnamen ein X anzeigen lassen, in dem sowohl Vor- als auch Nachname in einer Zeile vorhanden sind?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

KGRÖSSTE über VBA eintragen

Problem:	Wie kann ich die ersten 10 Ränge der Werte aus Spalte A über VBA in das Arbeitsblatt eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub KErmittelnWert() Dim iCounter As Integer For iCounter = 1 To 21 Cells(iCounter, 2) = Application.Large(Columns("A"), iCounter) Next iCounter End Sub Sub KErmittelnFormel() Dim iCounter As Integer For iCounter = 1 To 21 Cells(iCounter, 2).Formula = "=LARGE(A:A," & iCounter & ")" Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorgegebene Anzahl niedrigster Werte anzeigen

Problem:	Wie kann ich mir in Spalte B die in Zelle C3 genannte Anzahl der niedrigsten Werte aus Spalte A anzeigen lassen?
Lösung:	Folgende Formel: =WENN(\$C\$3>=ZEILE()-2;KKLEINSTE(A:A;ZEILE()-2);"")
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Email-Adresse aus Vorname, Name und Domain bilden

Problem:	Aus dem 1. Buchstaben des Vornamens, des Nachnamens und der Domain soll eine Emailadresse in Kleinschreibung ermittelt werden.
Lösung:	Folgende Formel: =KLEIN(LINKS(A2;1)&"&B2&"&"@&"&C2)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Postleitzahl vom Ort trennen

Problem:	In einer Adressdatei stehen Postleitzahlen und Orte in der gleichen Zelle. Die Postleitzahl erscheint mal in der Form "53567" und mal als "D-53567". Wie kann ich diese Zeichenfolge nach Postleitzahlen und Orten trennen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summe ohne Maximalwert und Kennzeichnung des Max-Wertes

Problem:	In Spalte F sollen die Werte aus den jeweiligen Zeilen ohne Berücksichtigung des Maximalwertes summiert werden. Der jeweilige Maximalwert soll mit einer Hintergrundfarbe belegt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Weiteren Verlauf prognostizieren

Problem:	In B2:B12 befinden sich die jeweiligen Auftragseingänge des Vorjahres, in C2:C10 die des aktuellen Jahres. Aufgrund der Monatsverteilung aus Spalte B soll für die restlichen Monate des laufenden Jahres (C11:C13) eine Prognose erstellt werden.

Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mittelwert aus den 6 höchsten Werten in Spalte A ermitteln

Problem:	Wie kann ich den Mittelwert aus den 6 höchsten Werten in Spalte A ermitteln?
Lösung:	Formel: =MITTELWERT(KGRÖSSTE(A:A;{1;2;3;4;5;6}))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mittelwert einer Zahlenreihe bis zur Zeile mit der Formel

Problem:	Wie kann ich den Mittelwert einer Zahlenreihe bis zur Eingabezeile der Formel ermitteln?
Lösung:	Formel: =MITTELWERT(INDIREKT("&A1:A&ZEILE()"))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus einer Reihe von 10-minütigen Messungen h-Mittelwerte auslesen

Problem:	Wie kann ich aus einer Reihe von 10-minütigen Messungen die jeweiligen stündlichen Mittelwerte auslesen lassen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nettoarbeitstage und bewegliche Feiertage

Problem:	Wie kann ich die Nettoarbeitstage eines Zeitraumes unter Berücksichtigung der beweglichen Feiertage ermitteln?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen nach Rangfolge ordnen

Problem:	In Tabelle E:F soll die Rangfolge aus Tabelle A:C nach Rängen geordnet wiedergegeben werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wetten auf Wahlergebnisse

Problem: Jeder Wetter wettet auf das amtliche Endergebnis der Wahlen. Gewonnen hat der Wetter mit der geringsten Gesamtabweichung vom amtlichen Endergebnis.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:

```
StandardModule: Modul1

Sub Makro1()

    Makro1 Makro
    Makro am 01.09.2004 von Hans W. Herber aufgezeichnet


    With Selection
        .HorizontalAlignment = xlCenterAcrossSelection
        .VerticalAlignment = xlBottom
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .ShrinkToFit = False
        .MergeCells = False
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Skatrangliste erstellen

Problem:	Es soll eine Skatrangliste nach einzelnen Spieltagen und kumuliert erstellt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>StandardModule: modMain Sub Filtern() With ActiveSheet.UsedRange .Sort Key1:=Range("A2"), Order1:=xlAscending, Header:=xlGuess, _ OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom .AdvancedFilter Action:=xlFilterCopy, CopyToRange:= _ Cells(1, .Columns.Count + 1), Unique:=False .EntireColumn.Delete End With End Sub ClassModule: DieseArbeitsmappe</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Errechnung von Schulnoten anhand einer Punktetabelle

Problem:	Wie kann ich die Schulnote im Verhältnis zur maximal erreichbaren Punktezahl festlegen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte auf 9er-Endungen ab-/auf-/runden

Problem: Die Werte im Bereich B2:D5 werden auf 9er-Endungen auf-, ab- oder gerundet.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bestehender Formel eine 5er-Rundung hinzufügen

Problem:	Der Formel in C1 soll eine Rundungsformel zur 5er-Rundung hinzugefügt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetRound() Dim sFormula sFormula = ActiveCell.Formula sFormula = Right(sFormula, Len(sFormula) - 1) sFormula = "=round(" & sFormula & "*2,1)/2" ActiveCell.Formula = sFormula End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Runden auf 0,05

Problem: Es soll in der zweiten Kommastelle auf 0,5 gerundet werden.

Lösung: Folgende Formel: =RUNDEN(A1*2;1)/2

VBA-Code: Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Rundungsbeispiele

Problem:	Wie sind die Rundungsfunktionen zu handhaben.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verlauf prognostizieren (über Funktion SCHÄTZER)

Problem:	In B2:B12 befinden sich die jeweiligen Auftragseingänge des Vorjahres, in C2:C10 die des aktuellen Jahres. Aufgrund der Monatsverteilung aus Spalte B soll für die restlichen Monate des laufenden Jahres (C11:C13) eine Prognose erstellt werden.

Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auslesen von Schnittstellenwerten

Problem:	Aus einer Tabelle sollen Schnittstellenwerte ausgelesen werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einsatz des Schnittstellen-Operators

Problem:	Ermittlung ein Schnittstelle mit Hilfe des Schnittstellen-Operators. Alternativ dazu die Ermittlung über VBA-Code.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zugriff auf Zellen unter Verwendung eines Bereichsnamens

Problem:	Wie kann ich auf Zellen eines zweiten Arbeitsblattes unter Verwendung eines Bereichsnamens und der INDIREKT-Funktion zugreifen und dabei den Bereich transponieren?
Lösung:	Folgende Formel: =INDEX(Test;SPALTE())
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nutzwertanalyse beim Haukauf

Problem:	Als Entscheidungsvorbereitung für einen Hauskauf soll eine Nutzwertanalyse durchgeführt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Stückelung nach EURO

Problem:	Der Betrag in Zelle A3 soll gemäß den in Euro zur Verfügung stehenden Noten und Münzen gestückelt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Wochentagen summieren

Problem:	In Spalte F sollen die summierten Werte des jeweiligen Wochentages aus Spalte E angezeigt werden.
Lösung:	Formel in Zelle F1: =SUMMEWENN(B:B;E1;C:C)
VBA-Code:	<pre>StandardModule: Modul1 Sub ImportCSV() Dim sPath As String, sFile As String sPath = ThisWorkbook.Path & "\" If Day(Date) > 1 Then sFile = Format(DateSerial(Year(Date), Month(Date), 1), "myyyy") & ".csv" Else sFile = Format(DateSerial(Year(Date), Month(Date) - 1, 1), "myyyy") & ".csv" End If Workbooks.Open sPath & sFile End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Umsatzaufteilung nach Lieferanten

Problem: Die Lieferantenumsätze aus Spalte B sollen nach Lieferanten aufgeteilt werden.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Stunden tagesweise addieren

Problem: Aus der Stempelliste sollen die Stunden eines jeweiligen Tages addiert werden.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in Abhängigkeit zur Währungsformatierung summieren

Problem:	Die Werte in Spalte C sollen in Abhängigkeit der Währungsformatierung summiert werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summen zwischen zwei Zeiträumen ermitteln

Problem:	Wie kann ich den in Spalte B stehenden Umsatz eines bestimmten Zeitraums ermitteln? Der Starttag steht in Zelle C2, der Endtag in Zelle D2 und die Datumsreihe befindet sich in Spalte A.
Lösung:	Folgende Formel: =SUMMEWENN(A:A;">="&C2;B:B)-SUMMEWENN(A:A;">"&D2;B:B)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrwertsteuerberechnung von Brutto und Summierung der Mwst.

Problem:	In einer Tabelle soll die Nettopreis- und Mehrwertsteuerberechnung von Bruttobeträgen vorgenommen werden. Die Mehrwertsteuer soll in Abhängigkeit des Mwst.-Satzes summiert werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beispiel für die Anwendung der SUMMEWENN-FORMEL

Problem:	Wie ist die SUMMEWENN-Formel zu handhaben?
Lösung:	Die Formel: =SUMMEWENN(A:A;D2;B:B)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summe bilden, wenn Kriterium Zahl

Problem:	Ich habe eine Tabelle mit einer Auflistung von Zahlen. Wie summiere ich die Zahlen, die größer als die Zahl in Zelle A1 sind?
Lösung:	Mit der EXCEL-Funktion "SUMMEWENN()". Wenn die Liste der Zahlen in B1:B100 steht, lautet die Formel für das Summieren aller Zahlen, die größer als der Wert in A1 sind: =SUMMEWENN(B1:B100;">"&A1)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summe bilden, wenn Kriterium Zeichenfolge

Problem:	Ich habe eine Tabelle mit den Spalten Artikel und Anzahl. Ist es möglich, die Summe der Anzahl jener Artikel zu berechnen, die ein bestimmtes Wort enthalten?
Lösung:	Ja, mit der EXCEL-Funktion "SUMMEWENN()". Wenn in A1:A100 die Artikel stehen und in B1:B100 die Anzahl steht, lautet die Formel für die Summe der Artikel "Hosen": =SUMMEWENN(A1:A100;"Hosen";B1:B100). Wildcards sind erlaubt.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beispiele für den Einsatz der Funktion SVERWEIS

Problem:	In 10 Tabellenblättern wird der Einsatz der SVERWEIS-Funktion gezeigt und erläutert.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallVlookup() Dim iCriteria As Integer Dim sErgebnis As String iCriteria = 5 sErgebnis = WorksheetFunction.VLookup(_ iCriteria, Range("VBA"), 2, 0) MsgBox "Zum Kriterium "" & iCriteria & _ "" wurde der " & vbLf & "Wert "" & _ sErgebnis & "" gefunden." End Sub Sub TestVlookup() Dim iCriteria As Integer Dim vErgebnis As Variant iCriteria = 0 vErgebnis = Application.VLookup(_ iCriteria, Range("VBA"), 2, 0) If IsError(vErgebnis) Then MsgBox "Der Suchbegriff "" & iCriteria & "" wurde nicht gefunden!" Else MsgBox "Zum Kriterium "" & iCriteria & _ "" wurde der " & vbLf & "Wert "" & _ vErgebnis & "" gefunden." End If End Sub Sub SetVlookup() Range("C19").Value = "Formel:" Range("D19").Formula = "=vlookup(5,vba,2,0)" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Der Einsatz der SVERWEIS-Funktion in einem Ausgabeblatt

Problem: Beispiel für ein Ausgabeblatt. Die Steuerung erfolgt über ScrollBars.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vervollständigung einer Adresse

Problem: Bei Auswahl eines Namens in Zelle A9 soll die übrige Adresse über Formeln vervollständigt werden.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_Calculate()
    Dim iRow As Integer
    iRow = Cells(Rows.Count, 1).End(xlUp).Row
    Range("A2:A" & iRow).Name = "Adressen"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in zweiter Tabelle in anderer Reihenfolge darstellen

Problem: Die Werte aus Spalte A werden in Spalte C in der Reihenfolge gem. Spalte B dargestellt.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert nach zwei Kriterien ermitteln

Problem:	In Zelle D2 soll der Wert aus Spalte C der Zeile geliefert werden, in der die Kriterien aus F1 und F2 beide zutreffen.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Korrespondenzwert einer Gültigkeitslisten-Auswahl abrufen

Problem:	In Spalte A werden Werte über ein Auswahlfeld eingetragen werden, wobei in Spalte B der Korrespondenzwert aus der Datenbank eingetragen werden soll.
Lösung:	Formel in B1: =SVERWEIS(A1;Datenbank!A:B;2;0)
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Position an Bandlaufwerk bestimmen

Problem:	An einem Bandlaufwerk gibt es 26 Bandplätze mit den Bezeichnungen A-Z. Die über das Band laufenden Materialien sind von 1 bis 99 durchnummeriert. Die Position der Materialien soll bestimmt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datenermittlung über SVERWEIS aus zweitem Tabellenblatt

Problem:	In Spalte B sollen die korrespondierenden Zahlenzuordnungen aus Tabelle2 übernommen werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Korrespondenzwert zum Eingabewert anzeigen

Problem:	Bei Eingaben in Zelle A1 sollen in Zelle A2 die jeweiligen Korrespondenzwerte aus der Tabelle Data ausgegeben werden. Formel: =SVERWEIS(A1;Data!A:B;2;0)
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamischer Blattname in SVERWEIS-Formel

Problem:	Der Blattname in einer auf ein anderes Blatt verweisenden SVERWEIS-Formel soll dynamisch - gem. Zelleintrag - gehalten werden.
Lösung:	Folgende Formel: =SVERWEIS(B2;INDIREKT(B1&"!A:B");2;0)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in Abhängigkeit zugeordneter Indizes multiplizieren

Problem:	Jedem Buchstaben in Spalte A ist ein Wert zugeordnet. Die Zuordnung befindet sich in Tabelle2. Entsprechend dieser Zuordnung sollen die jeweiligen Werte mit 3,14 multipliziert und in Spalte B eingetragen werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Integratgion von SVERWEIS in eine Namens-Funktion

Problem:	Der Wert in Spalte A soll in Tabelle1 oder Tabelle2 gesucht werden; wenn die jeweilige Zelle in Spalte B nicht leer ist, soll dieser, verkettet mit dem aus Spalte C, andernfalls nur der aus Spalte C eingetragen werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	<pre>StandardModule: Modul1 Sub NamenEin() Dim nme As Name For Each nme In ThisWorkbook.Names nme.Visible = True Next nme End Sub Sub NamenAus() Dim nme As Name For Each nme In ThisWorkbook.Names nme.Visible = False Next nme End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zweidimensionale Matrix mit der SVERWEIS-Funktion durchsuchen

Problem: Wie kann ich eine nicht in einem Tabellenblatt eingetragene zweidimensionale Matrix nach einem Wert durchsuchen, ohne jeden einzelnen Wert prüfen zu lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub ArrayTest()  
    Dim arr(1 To 100, 1 To 2) As Integer  
    Dim var As Variant  
    Dim iCounter As Integer, iTest As Integer  
    iTest = CInt(InputBox("Kontrollziffer eingeben:", , 5))  
    For iCounter = 1 To 100  
        arr(iCounter, 1) = iCounter  
        arr(iCounter, 2) = iCounter * 10  
    Next iCounter  
    var = Application.VLookup(iTest, arr, 2, 0)  
    If IsError(var) Then  
        Beep  
        MsgBox "Wert nicht gefunden!"  
    Else  
        MsgBox "Wert: " & var  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adresse nach Gültigkeitsfeldauswahl eintragen

Problem:	Wie kann ich nach der Auswahl einer Kundennummer in einem Gültigkeitsfeld die Adresse in Adresszellen eintragen lassen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

SVERWEIS, WVERWEIS, INDEX, VERGLEICH und Zielwertsuche

Problem:	Was bewirken die Funktionen SVERWEIS, WVERWEIS, INDEX, VERGLEICH sowie die Zielwertsuche?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Feststellung der beweglichen Feiertage

Problem:	Wie kann ich feststellen, ob es sich bei einem vorgegebenen Datum um einen Feiertag handelt?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zugriff über SVERWEIS() auf eine Artikelliste

Problem: Wie kann ich mit der SVERWEIS()-Formel auf eine Artikelliste zugreifen? Die Datensätze sollen danach durch Drücken einer Schaltfläche in eine Umsatzliste übernommen werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub Eintragen()  
    Dim var As Variant  
    Dim iRow As Integer, iCol As Integer  
    With Worksheets("Tagesjournal")  
        iRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1  
        For iCol = 1 To 4  
            .Cells(iRow, iCol).Value = Cells(iCol, 2).Value  
        Next iCol  
    End With  
    With Worksheets("Artikelnummer")  
        var = Application.Match(Cells(1, 2).Value, .Columns(1), 0)  
        If IsError(var) Then  
            Beep  
            MsgBox "Artikelnummer wurde nicht gefunden!"  
        Else  
            .Cells(var, 7).Value = .Cells(var, 7).Value - Cells(3, 2).Value  
        End If  
    End With  
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verweise anhand einer Kopfzeile erkennen

Problem:	Wie kann ich in einem Deckblatt Verweise auf Tabellenblätter einfügen, deren Namen in der Kopfzeile stehen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitszeittabelle und SVERWEIS-Formel

Problem:	Ich habe für jeden Mitarbeiter eine Zeittabelle und für jeden Monat eine Zeittabelle angelegt. Wie kann ich von den Mitarbeiterblättern auf die Monatsdaten zugreifen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	<pre>StandardModule: Modul1 Sub Anlegen() Dim i% For i = 1 To 31 Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Name = i Next i End Sub Sub Schutz() Dim i% For i = 1 To 5 Worksheets(i).Unprotect Next i End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in der Zelle neben dem Maximalwert ermitteln

Problem:	Wie kann ich den Wert einer Zelle rechts neber der Zelle mit dem Maximalwert ermitteln?
Lösung:	Folgende Funktion: =SVERWEIS(MAX(A:A);A:B;2;0)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

SVERWEIS auf geschlossene Arbeitsmappe aus Makro aufrufen

Problem:	Wie kann ich über ein VBA-Makro in eine Zelle einen SVERWEIS() auf eine geschlossene Arbeitsmappe eintragen, wenn sich das Suchkriterium, der Dateiname, der Blattname und die Matrix-Adresse im aktiven Blatt befinden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Verweis() Dim sPath As String, sFile As String Dim sWks As String, sRng As String sPath = ThisWorkbook.Path & "\" sFile = Range("B1").Value sWks = Range("B2").Value sRng = Range("B3").Value If Dir(sPath & sFile) = "" Then Beep MsgBox "Quelldatei " & sPath & sFile & _ " wurde nicht gefunden!" Exit Sub End If Range("D1").Formula = _ "=VLOOKUP(B4,'" & sPath & _ "[" & sFile & "]" & sWks & "'!" & _ sRng & ",2,0)" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zinstabelle mit variablen Rückzahlungen

Problem:	Wie erstelle ich eine Tabelle mit Errechnung der Restschuld bei festem Zinssatz und variablen Rückzahlungen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichen als Zahlen darstellen

Problem:	Die Zeichen in Spalte A sollen in den Nebenspalten jeweils mit einem Code dargestellt werden, wobei a=1, b=2 etc. ist. Formel in B2: =WENN(LÄNGE(\$A2)>=SPALTE()-1;CODE(KLEIN(TEIL(\$A2;SPALTE()-1;1)))-96;""")
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wertermittlung in Abhängigkeit eines Zeichens in einem String

Problem:	Bei dem 6. Zeichen in einer Zeichenfolge handelt es sich um eine 1 oder eine 2. In Abhängigkeit davon soll in einer anderen Zelle der Wert "Mann" oder "Frau" angezeigt werden.
Lösung:	Folgende Formel: =WENN(TEIL(A1;6;1)=""1"";""Mann";"Frau")
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Gefilterte Daten summieren

Problem: In Zeile 1 einer Tabelle sollen die Werte einer gefilterten Spalte summiert werden.

Lösung: Formel in E1: =TEILERGEBNIS(9;E3:E65536)

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl der durch Autofilter ermittelten Datensätze anzeigen

Problem:	Wie kann ich die Anzahl der durch Autofilter ermittelten Datensätze auslesen? Es soll die gleiche Meldung ausgegeben werden, die bei Anwendung des Autofilters in der Statusleiste erscheint.
Lösung:	Die Formel: =TEILERGEBNIS(3;A:A)-1&" von "&ANZAHL2(A:A)-1&" Datensätzen gefunden.";
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summe gefilterter Zeilen

Problem:	Wenn ich in einer Tabelle den Autofilter verwende und unten eine Summe einer Spalte anzeigen lassen will, liefert SUMME() immer den Wert aller Zeilen und nicht den der gefilterten. Was mache ich falsch?
Lösung:	Folgende Formel: =TEILERGEBNIS(9;A1:A9) - (Parameter 9 für Summe)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeiten summieren

Problem:	In Spalte A werden Zeiten im Sekunden mit Nachkommastellen eingetragen. In Zelle F2 wird die Summe dieser Zeiten in das Format M:SS,00 umgewandelt.
Lösung:	Zelle B2: =SUMME(A:A), Zelle C2: =GANZZAHL(B2/60), Zelle D2: =GANZZAHL(B2-GANZZAHL(B2/60)*60), Zelle E2: =(B2-GANZZAHL(B2))*100, Zelle F2: =C2";";"&TEXT(D2;"00")";";"&TEXT(E2;"00")
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Umrechnung von Grad in Dezimal und zurück

Problem: Gradangaben mit Min und Sec sollen in dezimal und zurück gerechnet werden.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fußballwette auswerten

Problem:	Wie kann ich eine Tabelle mit Fußballwetten auswerten. Die Vereinbarung ist wie folgt: Spielergebnis und Tore richtig = 3 Punkte Spielergebnis richtig, Tore falsch = 1 Punkt
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung von Datenbank- und Array-Formelergebnissen per VBA

Problem:	Über VBA sollen sowohl die Ergebnisse von Datenbank- wie auch Array-Formeln ermittelt und ausgegeben werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub DSumInVBA() Dim rngDataBase As Range Dim rngCriteria As Range Range("E3").Formula = "=dsum(a:b, ""Wert"", c1:d2)" MsgBox Range("E3").Value Range("E3").ClearContents End Sub Sub DSumInArray() Range("J3").FormulaArray = _ "=sum((a2:a100" & _ Range("F2").Value & _ CStr(CDbl(Range("G2").Value)) & _ ")*(b2:b100" & Range("H2").Value & _ Range("I2").Value & ")*b2:b100)" MsgBox Range("J3").Value Range("J3").ClearContents End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Formeleingabe über ShortCut

Problem: Wie kann ich auf ShortCut-Betätigung den Wert aus Spalte A mit dem Wert aus Spalte B multiplizieren und die Zelle formatieren?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub QuadratMeter()
    ActiveCell.NumberFormat = "#,##0.000 ""m²""
    ActiveCell.Formula = "=RC[-1]*RC[-2]"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Namen löschen ohne Formeln zu zerstören

Problem: Wie kann ich die in einem Tabellenblatt vergebenen Bereichsnamen löschen, ohne daß die Formeln mit den Zellbezügen zu diesen Namen eine Fehlermeldung aufweisen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub NamenLoeschen()
    Dim nme As Name
    Dim rng As Range
    Dim sNew As String
    For Each nme In ThisWorkbook.Names
        For Each rng In ActiveSheet.UsedRange.Cells
            If rng.HasFormula Then
                If InStr(rng.Formula, nme.Name) > 0 Then
                    sNew = nme.Refersto
                    sNew = Right(sNew, Len(sNew) - 1)
                    rng.Replace nme.Name, sNew
                End If
            End If
        Next rng
        nme.Delete
    Next nme
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summenformel unter Zahlenreihe eintragen

Problem:	Wie kann ich über VBA unterhalb der letzten beschriebenen Zelle in Spalte C die Summenformel eintragen lassen? Der erste Wert befindet sich in Zelle C1, die Zahlenreihe weist Lücken auf.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Addieren() Dim rng As Range Set rng = Cells(Rows.Count, 1).End(xlUp).Offset(1, 0) rng.Formula = "=SUM(A1:A" & rng.Row - 1 & ")" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Variable in Formel einbauen

Problem:	Wie kann ich eine als Variable festgelegte Zellposition in eine per VBA einzutragende Formel übernehmen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub VariableInFormel() Dim iCol As Integer iCol = Range("B1").Value Range("B2").Formula = _ "=SUM(R1C" & iCol & ":R100C" & iCol & ")" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die ersten 5 Zeichen nach einem "+" separieren

Problem:	Wie kann ich die jeweils ersten 5 Zeichen nach einem Plus-Zeichen in einer langen Zeichenfolge in Variablen speichern und dann in ein Tabellenblatt eintragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub StringFiltern() Dim arr(1 To 10) Dim iCounter As Integer Dim sTxt As String For iCounter = 1 To 10 sTxt = Cells(iCounter, 1).Value If InStr(sTxt, "+") Then sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, "+")) sTxt = Left(sTxt, 5) arr(iCounter) = sTxt End If Next iCounter For iCounter = 1 To 10 Cells(iCounter, 2).Value = arr(iCounter) Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ab der aktiven Zelle summieren

Problem: Wie kann ich per VBA die aktuelle Cursorposition bestimmen und in die aktive Zelle die Summe aller Zellwerte über bzw. unter dieser Zelle ermitteln?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub SummeBildenOben()
    Dim iCol As Integer
    Dim sFormula As String
    iCol = ActiveCell.Column
    sFormula = "=SUM(R1C" & iCol & ":R[-1]C" & iCol & ")"
    ActiveCell.FormulaR1C1 = sFormula
End Sub

Sub SummeBildenUnten()
    Dim iCol As Integer
    Dim sFormula As String
    iCol = ActiveCell.Column
    sFormula = "=SUM(R[1]C" & iCol & ":R" & Rows.Count & "C" & iCol & ")"
    ActiveCell.FormulaR1C1 = sFormula
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBA-Zugriff auf geschlossene Arbeitsmappe.

Problem:	Wie kann ich über VBA auf eine geschlossene Arbeitsmappe zugreifen?
Lösung:	Starten Sie das VBA-Programm aus dem Arbeitsblatt Data
VBA-Code:	<pre>StandardModule: basMain Sub Zugriff() Dim sPath As String, sFile As String Dim sWks As String, sRng As String sPath = ThisWorkbook.Path & "\" sFile = Range("B1").Value sWks = Range("B2").Value & "'!" sRng = Range("B3").Value If Dir(sPath & sFile) = "" Then Beep MsgBox prompt:="Arbeitsmappe '" & _ sPath & sFile & "' wurde nicht gefunden!" Exit Sub End If With Range("IV1") .Formula = "=" & sPath & "[" & sFile & "]" & sWks & sRng MsgBox prompt:=.Value .ClearContents End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Änderung der Reihenfolge einer Wertereihe

Problem:	Wenn in Zelle B1 der Wert verändert wird, ändern sich alle nachfolgenden Werte. Die Anzahl und die Benennung der Werte ist variabel und richtet sich nach Eintragung in Spalte IV (ohne frei Zeilen).
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abweichungen sichtbar machen

Problem:	In den Spalten C und D sollen die jeweiligen Abweichungen der Spalte A zu Spalte B angezeigt werden. Formel: =WENN(ISTFEHLER(VERGLEICH(A2;B:B;0));A2;""")
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wenn Name zu Telefonnummer vorhanden, eintragen

Problem:	Wenn die Telefonnummer in Spalte A von Sheet2 vorhanden ist, soll in Spalte A eine 1 und in Spalte H der zugehörige Name eingetragen werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilennummern von Fundstellen ermitteln

Problem:	In Spalte C sollen die Fundzeilennummern aus Spalte A der Werte aus Spalte B eingetragen werden.
Lösung:	Folgende Formel: =VERGLEICH(""* "&B1&"* ";A:A;0)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auf- und absteigend sortieren ohne Sortierfunktion

Problem:	Wie kann ich in einer Tabelle mit aufsteigenden Kundennummern die Kundenspalte einem auf- und einmal absteigend sortiert haben, ohne die Sortierfunktion einzusetzen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Multiplikation eines Wertes in Abhängigkeit seiner Position

Problem:	Im Bereich A1:E1 steht jeweils nur ein Wert. Wie kann ich diesen mit einem in Abhängigkeit seiner Spaltenposition stehenden Faktor multiplizieren?
Lösung:	Folgende Formel: =INDEX(A;1;VERGLEICH(WVERWEIS(100;A;1;1);A;0))*WAHL(VERGLEICH(WVERWEIS(100;A;1;1);A;0);1,1;1,2;1,3;1,4;1,5)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Minuszeichen umkehren

Problem:	Beim Import von Textdateien tauchen mehrfach Zellen mit hintenangefügtem Minuszeichen auf. Wie kann ich diese Zeichenfolgen in Minuswerte wandeln?
Lösung:	Folgende Formel: =WERT("-"&WECHSELN(A2;"-";""))
VBA-Code:	<pre>StandardModule: basMain Sub MinusNachVorne() Dim rng As Range For Each rng In ActiveSheet.Range("A2:A4").Cells rng.Offset(0, 2).Formula = "-" & _ WorksheetFunction.Substitute(rng.Value, "-", "") Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolge in Wert wandeln

Problem:	Wie kann ich einen Zahlenstring mit Punkt als Dezimalseparator in eine Zahl umwandeln?
Lösung:	Folgende Formel, wenn sich der Wert in Zelle A1 befindet: =WERT(WECHSELN(A1;".";";","."))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Unterschiedliche Smilies bei Zellvergleich

Problem:	In Abhängigkeit der Werte in B1 zu B2 (kleiner, gleich, größer) werden in B3 unterschiedliche Smilies mit unterschiedlicher Hintergrundfarbe angezeigt.
Lösung:	Formel in B3: =WENN(B2<B1;ZEICHEN(76);WENN(B2=B1;ZEICHEN(75);ZEICHEN(74))) (Zelle ist in der Schriftart WINGDINGS formatiert)
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzeige, wenn Listenwerte gefunden werden

Problem:	Wenn Werte aus Spalte A in den Bereichen F5:F15 oder K5:K10 enthalten sind, soll in Spalte B der Wert AKTIV angezeigt werden.
Lösung:	Formel in Spalte B: =WENN(UND(ISTNV(VERGLEICH(A1;\$F\$5:\$F\$15;0));ISTNV(VERGLEICH(A1;\$K\$5:\$K\$10;0))));"";"AKTIV")
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahlungsfrist prüfen

Problem:	Die Formel in Spalte E soll ermitteln, ob eine Zahlungsfrist überschritten wurden und wenn ja, wie lange.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einen Wert in einem zweiten Blatt einer Kalenderwoche zuordnen

Problem:	Der Wert aus Zelle B2 soll in Tabelle2 in der der Kalenderwoche aus Zelle B1 zugeordneten Spalte erscheinen.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Spielpaarungen von Vorrunde bis Finale durch Formeln ermitteln

Problem:	Wie kann ich die Spielpaarungen eines Turniers von der Vorrunde bis zum Finale ermitteln? Je nach Festlegung des Siegers in einer Paarung sollen die Folgepaarungen angepaßt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tagesliste eines Monats exakt anpassen

Problem:	Wie kann ich das Tagesdatum einer Monatstagesliste exakt an die tatsächlichen Tage unter Berücksichtigung der Schaltjahre anpassen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fehlermeldung #DIV/0! verhindern

Problem:	Wie kann ich die Fehlermeldung #DIV/0! verhindern?
Lösung:	Folgende Funktion: =WENN(E1;D1/E1;"""")
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

SVERWEIS-, WVERWEIS- und Matrixformel-Beispiele

Problem:	Wie sind die SVERWEIS-, WVERWEIS- und Matrixformeln zu handhaben?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorkommen von Werten mit Texten eintragen

Problem:	In Zelle H1 werden die Vorkommen der einzelnen Kurzbegriffe aus Spalte H gezählt und mit zugehörigem Text genannt.
Lösung:	Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beispiele zum Einsatz der ZÄHLENWENN-Funktion

Problem: Es werden Einsatzmöglichkeiten der ZÄHLENWENN-Funktion gezeigt.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Teilungszahlen und Sitzfolge nach dem Höchstzahlverfahren

Problem:	Stimmauszählungen bei Kommunalwahlen sollen nach dem Modus "Teilungszahlen und Sitzfolge nach dem Höchstzahlverfahren" ausgewertet werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zusammenstellung von Tätigkeitsmerkmalen je Mitarbeiter

Problem: In Tabelle2 soll die Anzahl der Tätigkeitsmerkmale pro Mitarbeiter gelistet werden.

Lösung: Nur anhand einer Beispielarbeitsmappe darstellbar.

VBA-Code: StandardModule: Modul1

```
Sub essai()  
    Dim semaine As Integer  
    Dim compt As String  
    Dim dat As String  
    von blatt2 wird die 2 $H$5 festgelegt  
  
    Worksheets("blatt2").Activate  
    If Cells(5, 8) <> "" Then semaine = Cells(5, 8)  
    Worksheets("blatt1").Activate  
    With Worksheets("blatt1").Range("a1:AJ34")  
        'es sind daten von einem Kalender  
        Set c = .Find(semaine, LookAt:=xlWhole)  
        If Not c Is Nothing Then  
            firstAddress = c.Address(ReferenceStyle:=xlA1, _  
                RowAbsolute:=False, _  
                ColumnAbsolute:=False)  
            compt = firstAddress  
            'compt =R[8]C[2]  
            Do  
                c.Interior.Pattern = xlPatternGray50  
                Set c = .FindNext(c)  
            Loop While Not c Is Nothing And c.Address <> firstAddress  
        End If  
    End With  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl der Einträge zu einer bestimmten Kalenderwoche

Problem:	Die Anzahl der Einträge in Spalte A zu einer bestimmten Kalenderwoche sollen ermittelt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zusagen einer Einladung ermitteln

Problem:	Wie kann ich ermitteln, wieviel Einladungen bisher beantwortet bzw. nicht beantwortet wurden und wieviel Zusagen mit wieviel Personen bisher vorliegen?
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beispiele zum Einsatz der ZELLE-Funktion

Problem: In 15 Arbeitsblättern werden Beispiele zum Einsatz der Funktion ZELLE gezeigt.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Application.DisplayCommentIndicator = gvStatus
End Sub

Private Sub Workbook_Open()
    gvStatus = Application.DisplayCommentIndicator
    Application.DisplayCommentIndicator = xlCommentIndicatorOnly
End Sub

StandardModule: basMain

Public gvStatus
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aktueller Arbeitsblattname in Formel

Problem:	Der Name des Arbeitsblattes soll in einer Formel ermittelt werden. Die Formel soll sich bei Blattumbenennung aktualisieren.
Lösung:	Folgende Funktion: =TEIL(ZELLE("dateiname";A1);FINDEN("]";ZELLE("dateiname";A1))+1;255)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blattnamen in Zelle eingeben

Problem:	Wie kann ich Pfad, Datei- und Blattnamen der aktiven Arbeitsmappe ermitteln?
Lösung:	Ausgangsformel in B1: =ZELLE("Dateiname";B1), Pfad: =LINKS(B1;FINDEN("[";B1)-2), Datei: =TEIL(B1;FINDEN("[";B1)+1;FINDEN(";";B1)-FINDEN("[";B1)-1), Blatt: =TEIL(B1;FINDEN(";";B1)+1;LÄNGE(B1))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kapitalverzinsung mit fixen Raten und variablen Zinsen

Problem:	Verzinsungstabelle mit variabler Laufzeit, variablen Zinsen und fixen Monatsraten.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichnungselemente weiß oder schwarz färben

Problem:	Bei Klick auf eines der Shapes-Elemente soll es sich, ist es schwarz, weiß färben und umgekehrt.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn den Shapes zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetColor() With ActiveSheet.Shapes(Application.Caller).Fill.ForeColor If .SchemeColor = 8 Then .SchemeColor = 65 Else .SchemeColor = 8 End If End With End Sub StandardModule: Modul2 Sub Makro1() Makro1 Makro Makro am 31.07.2003 von Hans W. Herber aufgezeichnet ActiveSheet.Shapes("Rectangle 1").Select Selection.ShapeRange.Fill.Visible = msoTrue Selection.ShapeRange.Fill.Solid Selection.ShapeRange.Fill.ForeColor.SchemeColor = 8 Selection.ShapeRange.Fill.Transparency = 0# Selection.ShapeRange.Line.Weight = 0.75 Selection.ShapeRange.Line.DashStyle = msoLineSolid Selection.ShapeRange.Line.Style = msoLineSingle Selection.ShapeRange.Line.Transparency = 0# Selection.ShapeRange.Line.Visible = msoTrue Selection.ShapeRange.Line.ForeColor.SchemeColor = 64 Selection.ShapeRange.Line.BackColor.RGB = RGB(255, 255, 255) Selection.ShapeRange.Fill.Visible = msoTrue Selection.ShapeRange.Fill.Solid Selection.ShapeRange.Fill.ForeColor.SchemeColor = 65 Selection.ShapeRange.Fill.Transparency = 0# Selection.ShapeRange.Line.Weight = 0.75 Selection.ShapeRange.Line.DashStyle = msoLineSolid Selection.ShapeRange.Line.Style = msoLineSingle Selection.ShapeRange.Line.Transparency = 0# Selection.ShapeRange.Line.Visible = msoTrue Selection.ShapeRange.Line.ForeColor.SchemeColor = 64 Selection.ShapeRange.Line.BackColor.RGB = RGB(255, 255, 255) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Animierte TextBox über das Arbeitsblatt bewegen

Problem: Eine animierte TextBox soll über das aktive Blatt bewegt werden.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code:

```
StandardModule: Modul1

Sub Bewegen()
    Dim iLeft As Integer, iTop As Integer
    ActiveSheet.Unprotect
    With ActiveSheet.TextBoxes(1)
        For iLeft = 1 To 400
            Call SetColors(iLeft)
            .Left = iLeft
        Next iLeft
        For iTop = 1 To 260
            Call SetColors(iTop)
            .Top = iTop
        Next iTop
        For iLeft = 400 To 1 Step -1
            Call SetColors(iLeft)
            .Left = iLeft
        Next iLeft
        For iTop = 260 To 1 Step -1
            Call SetColors(iTop)
            .Top = iTop
        Next iTop
        .Interior.ColorIndex = 0
    End With
    ActiveSheet.Protect
End Sub

Private Sub SetColors(iCount As Integer)
    With ActiveSheet.TextBoxes(1)
        If iCount Mod 20 = 0 Then
            If .Interior.ColorIndex = 3 Then
                .Interior.ColorIndex = 6
            Else
                .Interior.ColorIndex = 3
            End If
        End If
        If iCount Mod 100 = 0 Then
            If .Text = "DANGER" Then .Text = "" Else .Text = "DANGER"
        End If
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Animation in Excel

Problem:	Wie kann ich in Excel bewegte Objekte erstellen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn den Zeichnungselementen zu.
VBA-Code:	<pre>StandardModule: mHWH Declare Sub Sleep Lib "kernel32" _ (ByVal dwMilliseconds As Long) Sub Start() Dim oHWH As Shape Dim iCounter As Integer Set oHWH = ActiveSheet.Shapes("HWH") oHWH.Rotation = 0 Do Until iCounter = 360 oHWH.IncrementRotation 2 Sleep 10 DoEvents iCounter = iCounter + 2 Loop oHWH.Rotation = 0 End Sub Sub DM2Euro() Dim oEURO As Shape, oDM As Shape Set oEURO = ActiveSheet.Shapes("Euro") Set oDM = ActiveSheet.Shapes("DM") Dim iCounter As Integer, iCount As Integer oDM.Visible = True With oEURO .Top = 400 .Left = 5 .Height = 41.25 .Width = 36.75 .Rotation = 0 oEURO.Visible = True Do Until .Left >= 500 Or .Top <= 50 .IncrementRotation 15 .Left = .Left + 5 .Top = .Top - 4 Sleep 100 DoEvents Loop .Width = 180 .Height = 180 .Top = 20 .Left = 450 .Rotation = 0 End With ActiveSheet.Shapes("DM").Visible = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Klick auf ein Symbol Farbwechsel

Problem:	Bei Klick auf ein Symbol sollen die Symbole entsprechend Ampelfarben wechseln.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Makro1() Makro1 Makro Makro am 28.04.2004 von Hans W. Herber aufgezeichnet ActiveSheet.Shapes("AutoShape 5").Select Selection.ShapeRange.Fill.ForeColor.SchemeColor = 57 Selection.ShapeRange.Fill.Visible = msoTrue Selection.ShapeRange.Fill.Solid End Sub Sub Makro2() Makro2 Makro Makro am 28.04.2004 von Hans W. Herber aufgezeichnet Selection.ShapeRange.Fill.ForeColor.SchemeColor = 10 Selection.ShapeRange.Fill.Visible = msoTrue Selection.ShapeRange.Fill.Solid End Sub Sub Makro3() Makro3 Makro Makro am 28.04.2004 von Hans W. Herber aufgezeichnet Selection.ShapeRange.Fill.ForeColor.SchemeColor = 53 Selection.ShapeRange.Fill.Visible = msoTrue Selection.ShapeRange.Fill.Solid End Sub StandardModule: Modul2 Sub ChangeShapes() Dim col As New Collection Dim iShp As Integer, iCounter As Integer For iCounter = 1 To 3 col.Add ActiveSheet.Shapes(iCounter) If col(iCounter).Visible Then iShp = iCounter Next iCounter Select Case iShp Case 1 col(1).Visible = False col(3).Visible = False col(2).Visible = True Case 2 col(1).Visible = False col(2).Visible = False col(3).Visible = True Case 3 col(3).Visible = False col(2).Visible = False col(1).Visible = True End Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafik in Abhängigkeit eines Formelwertes ein / ausblenden

Problem: Wenn der Wert der Formel in B1 E ist, soll die Grafik sichtbar, andernfalls nicht sichtbar sein.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Calculate()
    If Range("B1").Value = "E" Then
        ActiveSheet.Pictures(1).Visible = True
    Else
        ActiveSheet.Pictures(1).Visible = False
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattgrafik ein/ausblenden und verschieben

Problem:	Die Grafik soll auf Schaltflächendruck ein/ausgeblendet bzw. verschoben werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub MoveGrafik() Dim pct As Picture Dim iRow As Integer, iCol As Integer Set pct = ActiveSheet.Pictures(1) iRow = pct.TopLeftCell.Row iCol = pct.TopLeftCell.Column If iCol = 4 Then iCol = 1 iRow = iRow + 1 If iRow = 6 Then iRow = 1 Else iCol = iCol + 1 pct.Left = Cells(iRow, iCol).Left Exit Sub End If pct.Left = Cells(iRow, iCol).Left pct.Top = Cells(iRow, iCol).Top End Sub Sub VisibleGrafik() Dim pct As Picture Set pct = ActiveSheet.Pictures(1) pct.Visible = Not pct.Visible End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bild bei Klick vergrößern und zurücksetzen

Problem: Beim Klicken auf eine Grafik soll sich diese vergrößern, bei nochmaligem Klick soll sie wieder zurückgesetzt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub BildGroesse()  
    Dim pct As Shape  
    Dim dblWidthOld As Double, dblHeightOld As Double  
    Dim dblWidthNew As Double, dblHeightNew As Double  
    Dim txt As String, txtTmp As String  
    Set pct = ActiveSheet.Shapes(Application.Caller)  
    txt = pct.TopLeftCell.Value  
    dblWidthOld = Cdbl(Left(txt, InStr(txt, "x") - 1))  
    txt = Right(txt, Len(txt) - InStr(txt, "x"))  
    dblHeightOld = Cdbl(Left(txt, InStr(txt, "/") - 1))  
    txt = Right(txt, Len(txt) - InStr(txt, "/"))  
    dblWidthNew = Cdbl(Left(txt, InStr(txt, "x") - 1))  
    txt = Right(txt, Len(txt) - InStr(txt, "x"))  
    dblHeightNew = Cdbl(txt)  
    With ActiveSheet.Pictures(1)  
        If Abs(dblWidthOld) < pct.Width * 1.5 - pct.Width Then  
            pct.Width = dblWidthOld  
            pct.Height = dblHeightOld  
        Else  
            pct.Width = dblWidthNew  
            pct.Height = dblHeightNew  
        End If  
    End With  
End Sub
```

Elemente in Tabellenblatt Pac-Man-artig steuern

Problem:	Wie kann ich Zeichnungselemente innerhalb eines Tabellenblattes so steuern, daß der Eindruck eines Pac-Man-Spiels entsteht?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdEnde_Click() Set PacMan = Nothing Set Vitamin = Nothing Application.OnKey "{right}" Application.OnKey "{left}" Application.OnKey "{up}" Application.OnKey "{down}" End Sub Private Sub cmdStart_Click() Set PacMan = ActiveSheet.Shapes("PacMan") Set Vitamin = ActiveSheet.Shapes("Vitamin") Schalter = False Tempo = 3 PacMan.Fill.ForeColor.SchemeColor = 7 PacMan.Top = 0 PacMan.Left = 0 Vitamin.Visible = True Application.OnKey "{right}", "ToRight" Application.OnKey "{left}", "ToLeft" Application.OnKey "{up}", "ToTop" Application.OnKey "{down}", "ToDown" End Sub StandardModule: basMain Public PacMan As Shape Public Vitamin As Shape Public Schalter As Boolean Public Tempo As Integer Sub ToRight() With PacMan If .Left >= 300 Then Beep Exit Sub End If .Left = .Left + Tempo End With Call Pille End Sub Sub ToLeft() With PacMan .Left = .Left - Tempo End With Call Pille End Sub Sub ToTop() With PacMan .Top = .Top - Tempo End With Call Pille End Sub Sub ToDown() With PacMan If .Top >= 200 Then Beep Exit Sub End If .Top = .Top + Tempo End With Call Pille End Sub Sub Pille() Dim dblPac As Double Dim dblVit As Double If Schalter = False Then If Abs(PacMan.Left - Vitamin.Left) < 13 And Abs(PacMan.Top - Vitamin.Top) < 13 Then Schalter = True Vitamin.Visible = msoFalse PacMan.Fill.ForeColor.SchemeColor = 12 Tempo = 5 Else Schalter = False End If End If End Sub Sub VitaminZeigen() ActiveSheet.Shapes("Vitamin").Visible = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ein Bild sekundenlang anzeigen lassen

Problem:	Wie kann ich in einem Arbeitsblatt sekundenlang ein Bild anzeigen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Imageinout() Dim pct As Picture Set pct = ActiveSheet.Pictures.Insert(Range("B1").Value) Application.Wait (Now + TimeSerial(0, 0, 1)) pct.Delete End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafiken im Tabellenblatt ordnen

Problem:	Wie kann ich mehrere, sich in einem Arbeitsblatt befindende Grafiken unmittelbar untereinander anordnen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CommandButton1_Click() Call AnOrdnen End Sub StandardModule: basMain Sub AnOrdnen() Dim shp As Shape Dim x As Double, y As Double x = 5: y = 0 For Each shp In Worksheets("Tabelle1").Shapes If shp.Type = msoPicture Then shp.Left = x shp.Top = y End If y = shp.Top + shp.Height Next shp End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Rechtsklick auf Diagramm MessageBox-Aufruf

Problem:	Ein Rechtsklick auf das Diagramm-Objekt soll eine Meldung aufrufen, ohne dass ein Diagramm-Kontextmenü angezeigt wird.
Lösung:	Den Code in die genannten Module eingeben.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Worksheets("Tabelle1").Select Set rng = ActiveCell End Sub ClassModule: Tabelle1 Dim MeinDiagramm As New Klasse1 Private Sub Worksheet_Activate() Set MeinDiagramm.Diagramm = ChartObjects(1).Chart End Sub Private Sub Worksheet_SelectionChange(ByVal Target As Range) Set rng = ActiveCell End Sub ClassModule: Klasse1 Dim MyButton As Long Public WithEvents Diagramm As Chart Private Sub Diagramm_BeforeRightClick(Cancel As Boolean) If MyButton = xlSecondaryButton Then Cancel = True MsgBox "Hallo User!" End If End Sub Private Sub Diagramm_MouseDown(ByVal Button As Long, ByVal Shift As Long, ByVal x As Long, ByVal y As Long) MyButton = Button End Sub Private Sub Diagramm_Select(ByVal ElementID As Long, ByVal Arg1 As Long, ByVal Arg2 As Long) Call Diagramm_BeforeRightClick(True) rng.Select End Sub StandardModule: Modul1 Public rng As Range</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Diagramm dynamisch erstellen

Problem:	Auf Schaltflächendruck soll das bestehende Diagramm - falls vorhanden - gelöscht und ein neues erstellt werden. Die Anzahl der Spalten und Zeilen soll dynamisch sein.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub CreateAndDelete() Dim cht As Chart On Error Resume Next ActiveSheet.ChartObjects(1).Delete On Error GoTo 0 Set cht = Charts.Add cht.ChartType = xlColumnClustered cht.SetSourceData Source:=Sheets("Tabelle1").Range("A1").CurrentRegion cht.Location Where:=xlLocationAsObject, Name:="Tabelle1" Range("A1").Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamisches Diagramm

Problem:	Der Datenbereich eines Diagramms soll der Anzahl der Werte in Zeile 2 angepasst werden.
Lösung:	Lösung über benannte Formeln; nur als Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ändern von Chart-Farben in Abhängigkeit der Daten

Problem:	Beim Wechsel des Wertes in Zelle A2 auf >5 soll sich die SeriesCollection.Interior-Farbe ändern.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_Change(ByVal Target As Range) Dim cht As Chart Set cht = ActiveSheet.ChartObjects(1).Chart If Range("A2").Value > 5 Then cht.SeriesCollection(2).Interior.ColorIndex = 3 Else cht.SeriesCollection(2).Interior.ColorIndex = 6 End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Diagramm erstellen, in UserForm anzeigen und drucken

Problem:	Wie kann ich aus einer Excel-Tabelle ein Diagramm erstellen, dieses als Grafikdatei speichern, in eine UserForm laden und von dort aus drucken lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DiagrammInUserForm() Dim cht As Chart Set cht = Charts.Add Application.ScreenUpdating = False cht.SetSourceData Source:=Sheets("Tabelle1").Range("A1:C3"), _ PlotBy:=xlRows cht.Export "test.gif" With frmChart.imgChart .Picture = LoadPicture("test.gif") .AutoSize = True End With frmChart.Show Kill "test.gif" Application.DisplayAlerts = False cht.Delete Application.DisplayAlerts = True Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Diagrammdatenreihe mit VBA ändern

Problem:	Wie kann ich eine Diagrammdatenreihe mit VBA-Befehl ändern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DatenreiheAendern() Dim cht As Chart Dim rng As Range Set cht = Worksheets(2).ChartObjects(2).Chart With Worksheets(2) Set rng = .Range(.Cells(1, 1), .Cells(100, 1)) End With cht.SeriesCollection(2).Values = rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mit einer Diagramm-Bildlaufleiste durch Tabelle bewegen

Problem:	Wie kann ich mich mit einer in einem Diagramm befindenden Bildlaufleiste durch eine Tabelle bewegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DatenTransfer() Dim wksW As Worksheet, wksWD As Worksheet Dim iCounter As Integer Set wksW = ActiveWorkbook.Sheets("W") Set wksWD = ActiveWorkbook.Sheets("WD") iCounter = wksWD.Cells(345, 1) wksWD.Range(wksWD.Cells(1, 2), wksWD.Cells(342, 2)).Value = _ wksW.Range(wksW.Cells(1, iCounter), wksW.Cells(342, iCounter)).Value End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Diagramm-Datenbereich über Schaltfläche festlegen

Problem:	Wie kann ich den Datenbereich eines Diagramms auf Schaltflächendruck festlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BereichZuweisen() Dim cht As Chart If Selection.Cells.Count = 1 Then Beep MsgBox "Sie müssen einen Bereich auswählen!" Exit Sub End If Set cht = ActiveSheet.ChartObjects(1).Chart cht.SetSourceData Source:=Worksheets("Werte"). _ Range(Selection.Address), PlotBy:=xlColumns End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datenreihenbeschriftungs-Quellbereich ausserhalb des Datenbereiches

Problem:	Wie kann ich bei einem Diagramm als Quellbereich für die Datenreihenbeschriftung einen Bereich ausserhalb des Wertebereiches festlegen?
Lösung:	Nur anhand einer Beipspielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eine Serie von Diagrammen erstellen und Druckansicht zeigen

Problem: Wie kann ich aus einer Tabelle eine Serie von Diagrammen erstellen und diese nacheinander in der Druckansicht anzeigen lassen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub GrafikenErstellen()
    Dim wks As Worksheet
    Dim cht As Chart
    Dim iRow As Integer
    Set wks = ActiveSheet
    iRow = 3
    Do Until IsEmpty(wks.Cells(iRow, 1))
        Set cht = Charts.Add
        With cht
            .ChartType = xlColumnClustered
            .SetSourceData _
                Source:=wks.Range( _
                    wks.Cells(iRow, 1), wks.Cells(iRow, 13)), _
                PlotBy:=xlRows
            .Location Where:=xlLocationAsNewSheet
            .HasTitle = True
            .ChartTitle.Characters.Text = wks.Cells(iRow, 1).Value
            .Axes(xlCategory, xlPrimary).HasTitle = False
            .Axes(xlValue, xlPrimary).HasTitle = False
            .PrintPreview
        End With
        iRow = iRow + 1
    Loop
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Diagramm-Datenreihen blinken lassen und Linie langsam zeichnen

Problem: Wie kann ich in einem Diagramm die Datenreihen blinken lassen und eine Linie so stückweise zeichnen, daß man den Verlauf verfolgen kann?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Blinken()
    Dim cht As Chart
    Dim iCounter As Integer
    Dim bln As Boolean
    Set cht = ActiveSheet.ChartObjects(1).Chart
    For iCounter = 1 To 10
        If bln Then
            cht.ApplyDataLabels (xlShowValue)
            bln = False
        Else
            cht.ApplyDataLabels (xlNone)
            bln = True
        End If
        Application.Wait Now + TimeSerial(0, 0, 1)
    Next iCounter
End Sub

Sub Zeichnen()
    Dim cht As Chart
    Dim iCounter As Integer
    Set cht = ActiveSheet.ChartObjects(1).Chart
    cht.SeriesCollection(2).Border.ColorIndex = 15
    For iCounter = 1 To cht.SeriesCollection(2).Points.Count - 1
        cht.SeriesCollection(2).Points(iCounter).Border.ColorIndex = 0
        Application.Wait Now + TimeSerial(0, 0, 1)
    Next iCounter
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Diagramm aus Daten einer Bandbreite bilden

Problem: Wie kann ich aus einer Tabelle Daten in einem bestimmten Bandreitenbereich separieren und daraus ein Diagramm bilden?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub DiagrammBilden()  
    Dim cht As Chart  
    Dim rng As Range  
    Dim iCounter As Integer, iRow As Integer  
    Range("A1").CurrentRegion.Sort _  
        Key1:=Range("A1"), _  
        Order1:=xlAscending, _  
        Header:=xlNo, _  
        OrderCustom:=1, _  
        MatchCase:=False, _  
        Orientation:=xlTopToBottom  
    iCounter = 1  
    For Each rng In Range("A1").CurrentRegion.Columns(2).Cells  
        If rng.Value > 6.11 And rng.Value < 6.24 Then  
            Cells(iCounter, 5) = rng.Offset(0, -1).Value  
            Cells(iCounter, 6) = rng.Value  
            Cells(iCounter, 7) = rng.Offset(0, 1).Value  
            iCounter = iCounter + 1  
        End If  
    Next rng  
    iRow = Cells(Rows.Count, 5).End(xlUp).Row  
    Set cht = Charts.Add  
    With cht  
        .ChartType = xlColumnClustered  
        .SetSourceData _  
            Source:=Worksheets("Tabelle1").Range("E1:G" & iRow), _  
            PlotBy:=xlColumns  
        .Location Where:=xlLocationAsObject, Name:="Tabelle1"  
    End With  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Diagramm aus Arbeitsblatt in Diagrammblatt übertragen

Problem:	Wie kann ich ein in einem Tabellenblatt erstelltes Diagramm in ein neues Diagrammblatt per VBA verschieben?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Uebertragen() Dim cht As Chart Range("A1:C1").Select Set cht = Charts.Add(before:=Sheets(1)) With Worksheets(2).ChartObjects(1).Activate With ActiveChart.ChartArea .Select .Copy End With End With cht.Paste Worksheets("Tabelle2").ChartObjects(1).Delete End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Die ersten beiden Grafiken in Tabellenblatt löschen

Problem:	Die ersten beiden Bilder im Tabellenblatt sollen gelöscht werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub DeletePictures() Dim pct As Picture Dim iCounter As Integer For iCounter = 1 To 2 ActiveSheet.Pictures(iCounter).Delete Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Flaggen nach dem Zufallsprinzip vertauschen

Problem:	Mit dem ShortCut STRG+a sollen die Bundesländerflaggen aus A1:B2 nach dem Zufallsprinzip ausgetauscht werden
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn dem ShortCut STRG+1 zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Importieren() Dim iCounter As Integer With Application.FileSearch .NewSearch .LookIn = "c:\bilder\flaggen" .Filename = "*.jpg" .Execute For iCounter = 1 To .FoundFiles.Count ActiveSheet.Pictures.Insert .FoundFiles(iCounter) Next iCounter End With End Sub Sub Start() Dim rng As Range, rngAll As Range Dim iRandomize As Integer Set rngAll = Range("A1:B2") Randomize For Each rng In rngAll iRandomize = Int((16 * Rnd) + 1) Do Until WorksheetFunction.CountIf(rngAll, iRandomize) = 0 iRandomize = Int((16 * Rnd) + 1) Loop rng.Value = iRandomize Next rng Call MovePictures End Sub Private Sub MovePictures() Dim pct As Picture Dim rng As Range, rngAll As Range Set rngAll = Range("A1:B2") For Each pct In ActiveSheet.Pictures pct.Left = Range("IV1").Left pct.Top = Range("IV1").Top Next pct For Each rng In rngAll.Cells With ActiveSheet.Pictures(rng.Value) .Left = rng.Left .Top = rng.Top End With Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafiken nach dem Zufallsprinzip anzeigen

Problem: Bei Betätigung von F9 soll nach dem Zufallsprinzip eine von 3 Grafiken angezeigt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle1

```
Private Sub Worksheet_Calculate()  
    If Range("Number").Value = 0 Then  
        Pictures(2).Visible = False  
        Pictures(3).Visible = False  
        Pictures(1).Visible = True  
    ElseIf Range("Number").Value = 1 Then  
        Pictures(1).Visible = False  
        Pictures(3).Visible = False  
        Pictures(2).Visible = True  
    ElseIf Range("Number").Value = 2 Then  
        Pictures(1).Visible = False  
        Pictures(2).Visible = False  
        Pictures(3).Visible = True  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafik in Kopfzeile

Problem:	Gibt es In EXCEL eine Möglichkeit, Grafiken in Kopfzeilen anzeigen zu lassen?
Lösung:	Das ist erst ab Version 10.0 (Office XP) möglich. In den Vorgängerversionen tragen Sie unter Menü DATEI / SEITE EINRICHTEN / TABELLE / DRUCKTITEL / WIEDERHOLUNGSZEILEN die Zeile(n) mit der Grafik ein.
VBA-Code:	<pre>StandardModule: basMain Sub PrintView() ActiveSheet.PrintPreview End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Symbolleiste beim Öffnen erstellen und dynamisch halten

Problem:	Beim Öffnen der Arbeitsmappe soll eine Symbolleiste erstellt und beim Schließen wieder gelöscht werden. Die Symbolleistenschaltflächen sollen nach den Einträgen in Spalte A benannt werden. Änderungen werden übernommen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe und der anderen genannten Module ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) On Error Resume Next Application.CommandBars("MyCellBar").Delete On Error GoTo 0 End Sub Private Sub Workbook_Open() Dim oBar As CommandBar Dim oBtn As CommandBarButton Dim iCounter As Integer On Error Resume Next Application.CommandBars("MyCellBar").Delete On Error GoTo 0 Set oBar = Application.CommandBars.Add("MyCellBar", msoBarTop, False, True) iCounter = 1 Do Until IsEmpty(Cells(iCounter, 1)) Set oBtn = oBar.Controls.Add With oBtn .Caption = Cells(iCounter, 1).Value .OnAction = "CntrMsg" .Style = msoButtonCaption End With iCounter = iCounter + 1 Loop oBar.Visible = True End Sub ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) If Intersect(Target, Range("A1").CurrentRegion) _ Is Nothing Then Exit Sub Application.CommandBars("MyCellBar") _ .Controls(Target.Row).Caption = Target.Value End Sub</pre>



Tabellenblattbereich als Grafik speichern

Problem:	Der selektierte Bereich soll als *.gif-Grafik gespeichert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Harold Staff -- see http://www.geocities.com/davemcritchie/excel/xl2gif.htm XL2GIF_module -- GIF_Snapshot Dim container As Chart Dim containerbok As Workbook Dim Obnavn As String Dim Sourcebok As Workbook Function SelectArea() As String Dim Internrange As Range On Error GoTo Brutt Set Internrange = Application.InputBox("Select " _ & "range to be photographed:", "Picture Selection", _ Selection.AddressLocal, Type:=8) SelectArea = Internrange.Address Exit Function Brutt: SelectArea = "A1" End Function Function sShortname(ByVal Orriginal As String) As String Dim iii As Integer sShortname = "" For iii = 1 To Len(Orriginal) If Mid(Orriginal, iii, 1) <> " " Then _ sShortname = sShortname & Mid(Orriginal, iii, 1) Next End Function Private Sub ImageContainer_init() Workbooks.Add (1) ActiveSheet.Name = "GIFcontainer" Charts.Add ActiveChart.ChartType = xlColumnClustered ActiveChart.SetSourceData Source:=Worksheets(1).Range("A1") ActiveChart.Location Where:=xlLocationAsObject, _ Name:="GIFcontainer" ActiveChart.ChartArea.ClearContents Set containerbok = ActiveWorkbook Set container = ActiveChart End Sub Sub MakeAndSizeChart(ih As Integer, iv As Integer) Dim Hincrease As Single Dim Vincrease As Single Obnavn = Mid(ActiveChart.Name, Len(ActiveSheet.Name) + 1) Hincrease = ih / ActiveChart.ChartArea.Height ActiveSheet.Shapes(Obnavn).ScaleHeight Hincrease, _ msoFalse, msoScaleFromTopLeft Vincrease = iv / ActiveChart.ChartArea.Width ActiveSheet.Shapes(Obnavn).ScaleWidth Vincrease, _ msoFalse, msoScaleFromTopLeft End Sub Public Sub GIF_Snapshot() Dim varReturn As Variant Dim MyAddress As String Dim SaveName As Variant Dim MySuggest As String Dim Hi As Integer Dim Wi As Integer Dim Suffiks As Long Set Sourcebok = ActiveWorkbook MySuggest = sShortname(ActiveSheet.Name) ImageContainer_init Sourcebok.Activate MyAddress = SelectArea If MyAddress <> "A1" Then SaveName = Application.GetSaveAsFilename(_ initialfilename:=MySuggest _ & ".gif", fileFilter:="Gif Files (*.gif), *.gif") Range(MyAddress).Select Selection.CopyPicture Appearance:=xlScreen, _ Format:=xlBitmap If SaveName = False Then GoTo Avbryt End If If InStr(SaveName, ".") Then SaveName _ = Left(SaveName, InStr(SaveName, ".") - 1) Selection.CopyPicture Appearance:=xlScreen, _ Format:=xlBitmap Hi = Selection.Height + 4 'adjustment for gridlines Wi = Selection.Width + 6 'adjustment for gridlines containerbok.Activate ActiveSheet.ChartObjects(1).Activate MakeAndSizeChart ih:=Hi, iv:=Wi ActiveChart.Paste ActiveChart.Export Filename:=LCase(SaveName) & _ ".gif", FilterName:="GIF" ActiveChart.Pictures(1).Delete Sourcebok.Activate End If Avbryt: On Error Resume Next Application.StatusBar = False containerbok.Saved = True containerbok.Close End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Export von Diagrammen im Grafikformat

Problem:	Wie kann ich Excel-Diagramm im *.GIF-Format speichern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub GrafikSpeichern() Const csMsg As String = "Die Diagramm-Grafik kann nicht " & _ "auf CD-ROM gespeichert werden. Speichern Sie die " & _ "Arbeitsmappe auf Ihrer Festplatte und versuchen " & _ "Sie es erneut." Dim sFile As String sFile = ThisWorkbook.Path & "\test_dia.gif" On Error GoTo ERRORHANDLER Diagramm1.Export sFile MsgBox prompt:="Das Diagramm wurde gespeichert unter " & _ sFile Exit Sub ERRORHANDLER: MsgBox prompt:=csMsg End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Übernahme einer Grafik gem. SVERWEIS

Problem:	Bei Eingabe einer Artikel-Nummer in Zelle D2 wird ausser der Benennung in Zelle D4 auch das zugehörige Symbol aus Spalte C der Quelltable übernommen.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle3 Private Sub Worksheet_Change(ByVal Target As Range) Dim shp As Shape Dim sAdr As String If Target.Address <> "\$D\$2" Then Exit Sub Application.EnableEvents = False On Error GoTo ERRORHANDLER For Each shp In ActiveSheet.Shapes shp.Delete Next shp If IsEmpty(Target) Then Range("D4").ClearContents GoTo ERRORHANDLER End If With Worksheets("Daten") If IsError(Application.Match(Target.Value, .Range("ArtNr"), 0)) Then Beep Range("D4").ClearContents MsgBox "Artikel-Nr. " & Target.Value & " wurde nicht gefunden!" GoTo ERRORHANDLER End If Range("D4").Value = WorksheetFunction.VLookup(Target.Value, .Range("Artikel"), 2, 0) sAdr = CStr(WorksheetFunction.Match(Target.Value, .Range("ArtNr"), 0) + 1) sAdr = Cells(CInt(sAdr), 3).Address For Each shp In .Shapes If shp.TopLeftCell.Address = sAdr Then shp.Copy Range("D6").Select ActiveSheet.Paste Range("D2").Select End If Next shp End With ERRORHANDLER: Application.EnableEvents = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Doppelklick Grafik in Kommentar einfügen

Problem:	Bei Doppelklick in Spalte A soll das in der aktiven Zelle genannte, im gleichen Verzeichnis wie diese Datei liegende Bild in einen Zellkommentar eingefügt werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Application.DisplayCommentIndicator = xlCommentIndicatorOnly End Sub ClassModule: Tabelle1 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean) Dim pct As Picture Dim cmt As Comment Dim sPath As String If Target.Column <> 1 Then Exit Sub If IsEmpty(Target) Then Exit Sub Application.ScreenUpdating = False Cancel = True If Not Target.Comment Is Nothing Then Target.Comment.Delete Exit Sub End If sPath = ThisWorkbook.Path & "\" If Dir(sPath & Target.Value) = "" Then Beep MsgBox "Bilddatei wurde nicht gefunden!" Exit Sub End If Set pct = ActiveSheet.Pictures.Insert(sPath & Target.Value) Set cmt = Target.AddComment With cmt.Shape .Width = pct.Width .Height = pct.Height With .Line .DashStyle = msoLineSolid .Style = msoLineSingle .Transparency = 0# .Visible = msoTrue .ForeColor.RGB = RGB(0, 0, 0) .BackColor.RGB = RGB(255, 255, 255) End With With .Fill .Visible = msoTrue .ForeColor.RGB = RGB(255, 255, 255) .BackColor.SchemeColor = 80 .Transparency = 0# .UserPicture sPath & Target.Value End With End With pct.Delete Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafiken gem. Hyperlinkadressen öffnen

Problem:	Auf Schaltflächenklick sollen alle mit Hyperlink verbundenen Bilder der ausgewählten Zellen geöffnet werden
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub LoadPictures() Dim rng As Range For Each rng In Selection.Cells Shell "hh " & rng.Hyperlinks(1).Address, vbNormalFocus Next rng End Sub</pre>

Grafiken in Zellkommentaren anzeigen

Problem:	Es sollen alle Bilddateien des in Zelle B1 genannten Verzeichnisses aufgelistet und die Bilder in einem Zellkommentar angezeigt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: Modul1 <pre>Sub PicShow() Dim pct As Picture Dim cmt As Comment Dim arr() As String Dim iFile As Integer, iRow As Integer, iCounter As Integer, iPattern As Integer Dim iAll As Integer, iCol As Integer Dim sPattern As String Dim bln As Boolean Application.ScreenUpdating = False Columns(1).NumberFormat = "@" Range("A3").Value = "Datei" Range("B3").Value = "Datum" Range("C3").Value = "Breite" Range("D3").Value = "Höhe" With Range("A3:D3") .Font.Bold = True .Interior.ColorIndex = 1 .Font.ColorIndex = 2 End With iAll = 1 sPattern = "*.gif" For iPattern = 1 To 2 With Application.FileSearch .LookIn = Range("B1").Value .Filename = sPattern .Execute For iCounter = iAll To .FoundFiles.Count + iAll - 1 ReDim Preserve arr(1 To 5, 1 To iCounter) bln = True Set pct = ActiveSheet.Pictures.Insert(.FoundFiles(iCounter - iAll + 1)) arr(1, iCounter) = .FoundFiles(iCounter - iAll + 1) arr(2, iCounter) = Dir(.FoundFiles(iCounter - iAll + 1)) arr(3, iCounter) = FileDateTime(.FoundFiles(iCounter - iAll + 1)) arr(4, iCounter) = CInt(pct.Width * 1.33333) arr(5, iCounter) = CInt(pct.Height * 1.33333) pct.Delete For iCol = 2 To 5 Cells(iCounter + 3, iCol - 1).Value = arr(iCol, iCounter) Next iCol Set cmt = Cells(iCounter + 3, 1).AddComment With cmt.Shape .Width = CInt(arr(4, iCounter) / 1.33333) .Height = CInt(arr(5, iCounter) / 1.33333) With .Line .DashStyle = msoLineSolid .Style = msoLineSingle .Transparency = 0# .Visible = msoTrue .ForeColor.RGB = RGB(0, 0, 0) .BackColor.RGB = RGB(255, 255, 255) End With With .Fill .Visible = msoTrue .ForeColor.RGB = RGB(255, 255, 255) .BackColor.SchemeColor = 80 .Transparency = 0# .UserPicture arr(1, iCounter) End With End With Next iCounter End With sPattern = "*.jpg" iAll = iCounter Next iPattern If bln = False Then Beep MsgBox "Es wurden keine Bilddateien gefunden -" & vbLf & _ "überprüfen Sie die eingetragenen Verzeichnisse!" End If Columns.AutoFit Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafik importieren und in Zelle zentrieren

Problem: Über die erste Schaltfläche soll die Grafik mit dem in Zelle B1 stehenden Namen in die aktive Zelle eingefügt werden. Über die zweite Schaltfläche soll neu zentriert werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub InsertPicture()
    Dim pct As Picture
    Dim iLeft, iTop, iWidth, iHeight
    Dim sFile As String
    sFile = Range("B1").Value
    If Dir(sFile) = "" Then
        Beep
        MsgBox "Grafikdatei wurde nicht gefunden!"
        Exit Sub
    End If
    With ActiveCell
        iLeft = .Left
        iTop = .Top
        iWidth = .Width
        iHeight = .Height
    End With
    Set pct = ActiveSheet.Pictures.Insert(Range("B1").Value)
    pct.Left = iLeft + iWidth / 2 - pct.Width / 2
    pct.Top = iTop + iHeight / 2 - pct.Height / 2
End Sub

Sub CenterPicture()
    Dim pct As Picture
    Dim iLeft, iTop, iWidth, iHeight
    If ActiveSheet.Pictures.Count = 0 Then
        Beep
        MsgBox "Keine Grafikdatei gefunden!"
        Exit Sub
    End If
    With ActiveCell
        iLeft = .Left
        iTop = .Top
        iWidth = .Width
        iHeight = .Height
    End With
    Set pct = ActiveSheet.Pictures(1)
    pct.Left = iLeft + iWidth / 2 - pct.Width / 2
    pct.Top = iTop + iHeight / 2 - pct.Height / 2
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafiken aus ausgeblendeten Blättern einblenden

Problem:	Wie kann ich in einem Arbeitsblatt ein Pfeilmenü anlegen. Bei Anklicken eines Pfeils soll eine Grafik aus einem ausgeblendeten Arbeitsblatt angezeigt werden und der Pfeil soll seine Farbe ändern.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BildEinfuegen() Dim pct As Shape For Each pct In ActiveSheet.Shapes If pct.Type = msoPicture Then pct.Delete End If Next pct For Each pct In ActiveSheet.Shapes If pct.Type = msoAutoShape Then pct.Fill.ForeColor.SchemeColor = 57 pct.TextFrame.Characters.Font.ColorIndex = 3 End If Next pct With ActiveSheet.Shapes(Application.Caller) .Fill.ForeColor.SchemeColor = 53 .TextFrame.Characters.Font.ColorIndex = 6 End With Worksheets(Right(Application.Caller, 1) + 2).Range("A1:E10") _ .CopyPicture xlScreen, xlBitmap ActiveSheet.Paste Destination:=Range("C1:C10") End Sub Sub Verbergen() Dim iCounter For iCounter = 3 To Worksheets.Count Worksheets(iCounter).Visible = xlVeryHidden 'Worksheets(iCounter).Visible = True Next iCounter End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafik nach Doppelklick laden

Problem: Wie kann ich veranlassen, dass nach einem Doppelklick auf eine Zelle die Grafik, deren Name und Pfad in der Zelle stehen, geladen und in die nebenstehende Zelle eingefügt wird?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle2

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim pic As Picture
    If Target.Column <> 1 Then Exit Sub
    If IsEmpty(Target) Then Exit Sub
    Cancel = True
    If Dir(Target.Value) = "" Then
        Beep
        MsgBox "Grafikdatei wurde nicht gefunden!"
    Else
        Set pic = ActiveSheet.Pictures.Insert(Target.Value)
        With pic
            .Left = Target.Offset(0, 1).Left
            .Top = Target.Offset(0, 1).Top
        End With
    End If
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafik in Abhängigkeit einer Zelleingabe einfügen

Problem: Wie kann ich in Abhängigkeit einer Zelleingabe in Spalte A eine Grafik in Spalte B einfügen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code: ClassModule: Tabelle2

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim pct As Object
    If Target.Column <> 1 Then Exit Sub
    On Error GoTo ErrorHandler
    Select Case Application.Caller
        Case 1
            Set pct = ActiveSheet.Pictures.Insert("Auto.gif")
        Case 2
            Set pct = ActiveSheet.Pictures.Insert("Krad.gif")
        Case 3
            Set pct = ActiveSheet.Pictures.Insert("Bus.gif")
    End Select
End
ErrorHandler:
    MsgBox "Grafik ist nicht vorhanden!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bilder in Videoliste eintragen und löschen

Problem:	Wie kann ich in eine Tabelle mit einer Aufstellung von Video-Filmen Bilder einfügen und löschen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle2 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean) Dim pct As Picture If Target.Column < 2 Then Exit Sub If IsEmpty(Target) Then Exit Sub Cancel = True Target.Offset(0, 1).Select If Dir(Target.Value) = "" Then Beep MsgBox "Grafik wurde nicht gefunden!" Else For Each pct In ActiveSheet.Pictures If pct.Top = Target.Offset(0, 1).Top Then pct.Delete End If Next pct Set pct = ActiveSheet.Pictures.Insert(Target.Value) End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Grafik einfügen, wenn Wert in A1 unter eine Grenze sinkt

Problem: Wie kann ich über VBA automatisch eine Grafik einfügen lassen, wenn der durch eine Formel ermittelte Wert in Zelle A1 unter 10 sinkt?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Calculate()
    Dim pct As Picture
    If Range("A1").Value < 10 Then
        On Error GoTo ERRORHANDLER
        Set pct = ActiveSheet.Pictures.Insert("Warnung.gif")
    End If
    Exit Sub
ERRORHANDLER:
    MsgBox "Grafik wurde nicht gefunden!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bildergruppe in anderes Blatt kopieren

Problem:	Die Bildergruppe im Bereich M1:O8 soll an die gleiche Position in Tabelle2 kopiert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub BilderGruppeKopieren() Dim pctsSource As Object, pctsTarget As Object Set pctsSource = ActiveSheet.Pictures pctsSource.Copy Worksheets(3).Select Range("M1").Select Worksheets(3).Paste End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bild in zweites Blatt an gleiche Position kopieren

Problem:	Ein Bild soll in Tabelle 2 an gleiche Position kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CopyPicture() Dim pct As Picture With ActiveSheet.Pictures("Lxmb") .Copy Worksheets(2).Paste Set pct = Worksheets(2).Pictures(1) pct.Left = .Left pct.Top = .Top End With End Sub</pre>

Einen Tabellenblattbereich als Graf in UserForm einbinden

Problem:	Der Tabellenblattbereich C6:D8 soll in einer UserForm als ScreenShot dargestellt werden.
Lösung:	Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.
VBA-Code:	StandardModule: Modul1 <pre>Dim container As Chart Dim ContainerBook As Workbook Dim Obnavn As String Dim SourceBook As Workbook Public Sub DialogAufruf() Call GIF_Snapshot Application.ScreenUpdating = False frmSnapShot.Image1.Picture = LoadPicture(_ Application.DefaultFilePath & "\img.gif") frmSnapShot.Show Application.ScreenUpdating = False End Sub Private Sub ImageContainer_init() Workbooks.Add 1 ActiveSheet.Name = "GIFcontainer" Charts.Add ActiveChart.ChartType = xlColumnClustered ActiveChart.SetSourceData Source:=Worksheets(1).Range("A1") ActiveChart.Location Where:=xlLocationAsObject, _ Name:="GIFcontainer" ActiveChart.ChartArea.ClearContents Set ContainerBook = ActiveWorkbook Set container = ActiveChart End Sub Private Sub MakeAndSizeChart(ih As Integer, iv As Integer) Dim Hincrease As Single Dim Vincrease As Single Obnavn = Mid(ActiveChart.Name, Len(ActiveSheet.Name) + 1) Hincrease = ih / ActiveChart.ChartArea.Height ActiveSheet.Shapes(Obnavn).ScaleHeight Hincrease, _ msoFalse, msoScaleFromTopLeft Vincrease = iv / ActiveChart.ChartArea.Width ActiveSheet.Shapes(Obnavn).ScaleWidth Vincrease, _ msoFalse, msoScaleFromTopLeft End Sub Private Sub GIF_Snapshot() Dim SaveName As Variant Dim Hi As Integer Dim Wi As Integer Set SourceBook = ActiveWorkbook Call ImageContainer_init SourceBook.Activate SaveName = Application.DefaultFilePath & "\img.gif" If Dir(SaveName) <> "" Then Kill SaveName End If Range("C6:D8").Select Selection.CopyPicture Appearance:=xlScreen, _ Format:=xlBitmap Hi = Selection.Height + 4 Wi = Selection.Width + 6 ContainerBook.Activate ActiveSheet.ChartObjects(1).Activate Call MakeAndSizeChart(ih:=Hi, iv:=Wi) ActiveChart.Paste ActiveChart.Export Filename:=LCase(SaveName), _ FilterName:="GIF" ActiveChart.Pictures(1).Delete SourceBook.Activate On Error Resume Next Application.StatusBar = False ContainerBook.Saved = True ContainerBook.Close End Sub ClassModule: frmSnapShot Private Sub cmdCancel_Click() Unload Me End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

UserForm mit SpinButton zur Datenreiheneinstellung aufrufen

Problem:	Wie kann ich über eine Schaltfläche in einem Diagramm eine UserForm mit einem SpinButton zum Festlegen von Werten in einer Datenreihe aufrufen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DialogAufruf() frmDiagramm.Show End Sub ClassModule: frmDiagramm Private Sub cmdWeiter_Click() Unload Me End Sub Private Sub spnKorrektur_Change() Range("A3").Value = spnKorrektur.Value End Sub Private Sub UserForm_Initialize() spnKorrektur.Value = 1 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bilder in UserForm und Bildgröße anpassen

Problem: Wie kann ich Bilder eines Verzeichnisses in einem UserFormDropDown-Feld auflisten und nach Auswahl in der UserForm anzeigen lassen. Die Bildgröße soll automatisch an die UserForm angepaßt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: frmBilderZeigen

Private Sub cbbBilder_Change()
    imgBild.Picture = LoadPicture(cbobilder.Value)
End Sub

Private Sub cmdWeiter_Click()
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Dim iCounter As Integer
    Dim sPath As String
    sPath = Range("B1").Value
    With Application.FileSearch
        .LookIn = Range("B1").Value
        .Filename = "*.gif"
        .Execute
        For iCounter = 1 To .FoundFiles.Count
            cbobilder.AddItem .FoundFiles(iCounter)
        Next iCounter
    End With
    If cbobilder.ListCount > 0 Then
        cbobilder.ListIndex = 0
    End If
End Sub

StandardModule: basMain

Sub CallForm()
    frmBilderZeigen.Show
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

WordArt-Element hinzufügen und löschen

Problem:	Bei Aktivierung der CheckBox soll ein WordArt-Element hinzugefügt, bei Deaktivierung gelöscht werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub chbWordArt_Click() Dim shp As Shape If chbWordArt.Value = True Then Set shp = ActiveSheet.Shapes.AddTextEffect(_ msoTextEffect15, _ "Mein Text", _ "Arial Black", _ 36#, _ msoFalse, _ msoFalse, _ 289.5, _ 186.75) chbWordArt.TopLeftCell.Value = shp.Name Else ActiveSheet.Shapes(chbWordArt.TopLeftCell.Value).Delete chbWordArt.TopLeftCell.ClearContents End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus geschlossenen Arbeitsmappen einlesen

Problem:	Aus geschlossenen Arbeitsmappen mit den Arbeitsmappennamen 1.xls bis n.xls (gem. den Variablen in den Zellen B2:B3) sollen die in Zelle B5 angegebenen Bereiche aus den in Zelle B4 angegebenen Arbeitsblättern summiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SetSum() Dim dValue As Double Dim iCounter As Integer Dim sFormula As String, sPath As String sPath = Range("B1").Value If Right(sPath, 1) <> "\" Then sPath = sPath & "\" End If For iCounter = Range("B2").Value To Range("B3").Value sFormula = "" & sPath sFormula = sFormula & "[" & iCounter & ".xls" & "]" sFormula = sFormula & Range("B4").Value & "!" sFormula = sFormula & Range("B5").Value Range("C1").Formula = "=" & sFormula dValue = dValue + Range("C1").Value Next iCounter Range("C1").ClearContents Range("B6").Value = dValue End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Details von Verknüpfungen sollen aufgelistet werden

Problem: Alle Verknüpfungen im Bereich A1:E9 sollen mit Detailangaben aufgelistet werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub LinkInfo()
    Dim arrDetails As Variant
    Dim rng As Range, rngSel As Range
    Dim iCounter As Integer, iRow As Integer, sText As String
    Set rngSel = Range("A1:E9")
    Workbooks.Add 1
    Range("A1").Value = "LinkAddress:"
    Range("B1").Value = "Path:"
    Range("C1").Value = "Workbook:"
    Range("D1").Value = "Worksheet:"
    Range("E1").Value = "Range:"
    Range("A1:E1").Font.Bold = True
    iRow = 1
    For Each rng In rngSel
        If rng.HasFormula Then
            If InStr(rng.Formula, "\[") Then
                arrDetails = GetDetails(rng.Formula)
                iRow = iRow + 1
                Cells(iRow, 1).Value = rng.Address
                For iCounter = 1 To 4
                    Cells(iRow, iCounter + 1).Value = arrDetails(iCounter)
                Next iCounter
            End If
        End If
    Next rng
    Columns.AutoFit
End Sub

Private Function GetDetails(sTxt As String) As Variant
    Dim sWkb As String, sWks As String, sRng As String
    Dim sPath As String
    Dim arr(1 To 4) As String
    arr(1) = Mid(sTxt, InStr(sTxt, "'") + 1, _
        InStr(sTxt, "[") - InStr(sTxt, "'") - 2)
    arr(2) = Mid(sTxt, InStr(sTxt, "[") + 1, _
        InStr(sTxt, "]") - InStr(sTxt, "[") - 1)
    arr(3) = Mid(sTxt, InStr(sTxt, "]") + 1, _
        InStr(sTxt, "!!") - InStr(sTxt, "]") - 1)
    arr(4) = Right(sTxt, Len(sTxt) - InStr(sTxt, "!!"))
    GetDetails = arr
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dateiname, Tabelle und Berreich zu einer Verknüpfung verbinden

Problem: Ein über GetOpenFileName ermittelter Dateiname soll in Zelle A2 mit dem Arbeitsblattnamen in Zelle B2 und der Zelladresse in C2 zu einer Verknüpfung verbunden werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub CreateLink()  
    Dim vFile As Variant  
    Dim sPath As String, sWkb As String, sWks As String, sRng As String  
    vFile = Application.GetOpenFilename("Excel-Dateien (*.xls), *.xls")  
    If vFile = False Then Exit Sub  
    sWks = Range("B2").Value  
    sRng = Range("C2").Value  
    sPath = fctPathName(CStr(vFile))  
    sWkb = fctFileName(CStr(vFile))  
    vFile = "'" & sPath & "[" & sWkb & "]" & sWks & "'" & sRng  
    Range("A2").Formula = "=" & vFile  
End Sub  
  
Private Function fctPathName(sFile As String)  
    Dim iCounter As Integer  
    Dim sTmp As String  
    sTmp = sFile  
    Do While InStr(sTmp, "\")  
        sTmp = Right(sTmp, Len(sTmp) - 1)  
        iCounter = iCounter + 1  
    Loop  
    fctPathName = Left(sFile, iCounter)  
End Function  
  
Private Function fctFileName(sFile As String)  
    Dim iCounter As Integer  
    For iCounter = Len(sFile) To 1 Step -1  
        If Mid(sFile, iCounter, 1) = "\" Then Exit For  
    Next iCounter  
    fctFileName = Right(sFile, Len(sFile) - iCounter)  
End Function
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle externen Links löschen und in Werte umwandeln

Problem:	Alle externen Links sollen gelöscht und deren Werte übernommen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub Link2Value() Dim rng As Range For Each rng In ActiveSheet.UsedRange.Cells If rng.HasFormula Then If InStr(rng.Formula, "\[") Then rng.Value = rng.Value End If End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolgen über Doppelklick in externen Bezug umwandeln.

Problem: Wie kann ich eine Zeichenfolge durch Doppelklick in einen Externen Bezug umwandeln?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_BeforeDoubleClick( _
    ByVal Target As Range, Cancel As Boolean)
    If Target.Column > 1 Then Exit Sub
    Cancel = True
    Target.Formula = Target.Value
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Letzte Verknüpfung in einem Tabellenblatt löschen

Problem:	Wie kann ich die letzte, in einem Tabellenblatt enthaltene Verknüpfung löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub LetzteVerknuepfung() Dim lCell As Long With ActiveSheet.UsedRange For lCell = .Cells.Count To 1 Step -1 If InStr(.Cells(lCell).Formula, "[") And _ InStr(.Cells(lCell).Formula, "]") Then MsgBox "Verknüpfung in Zelle " & _ .Cells(lCell).Address(False, False) & " wird gelöscht!" .Cells(lCell).ClearContents End End If Next lCell End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Quellarbeitsmappe einer Verknüpfung wechseln

Problem: Wie kann ich über Eingaben in einer InputBox die Quellarbeitsmappe einer Verknüpfung ändern?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub LinkUmleitung()  
    Dim var As Variant  
    Dim iCounter As Integer  
    Dim sName As String, sPath As String, sExt As String  
    sPath = ThisWorkbook.Path & "\"  
    sExt = ".xls"  
    sName = InputBox( _  
        prompt:="Quellarbeitsmappe:", _  
        Default:="test2")  
    If sName = "" Then Exit Sub  
    sName = sPath & sName & sExt  
    var = ActiveWorkbook.LinkSources(xlExcelLinks)  
    If Not IsEmpty(var) Then  
        For iCounter = 1 To UBound(var)  
            ActiveWorkbook.ChangeLink _  
                Name:=var(iCounter), newname:=sName  
        Next iCounter  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verknüpfung zu einer Mappe durch Eingabe der Zelladresse

Problem:	Wie kann ich durch Eingabe einer Zelladresse (F16) in ein Zelle eine Verknüpfung zu der in der Kopfzeile der Spalte genannten Arbeitsmappe und in der Zelle darunter genannten Tabelle herstellen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim wks As Worksheet Dim iCol As Integer Dim sPath As String, sFormula As String Set wks = Worksheets("Tabelle1") sPath = "=" & ThisWorkbook.Path & "\" For iCol = 1 To 3 sFormula = sPath & "[" & wks.Cells(1, iCol).Value & "]" sFormula = sFormula & wks.Cells(2, iCol).Value & "!A1" wks.Cells(3, iCol).Formula = sFormula sFormula = "" Next iCol End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verknüpfungen der aktiven Mappe listen

Problem:	Wie kann ich in einem Tabellenblatt alle Verknüpfungen der aktiven Arbeitsmappe auflisten lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub LinkList() Dim var As Variant Dim iCounter As Integer var = ActiveWorkbook.LinkSources(xlExcelLinks) If Not IsEmpty(var) Then For iCounter = 1 To UBound(var) Cells(iCounter, 1) = var(iCounter) Next iCounter Else Beep MsgBox "Keine Verknüpfungen enthalten!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Umleitung von Links

Problem: Wie kann ich über ein VBA-Makro alle bestehenden Links einer Arbeitsmappe an eine über eine InputBox abzufragende Datei umleiten?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: basMain

```
Sub LinkUmleitung()  
    Dim var As Variant  
    Dim iCounter As Integer  
    Dim sPath As String, sExt As String, sName As String  
    sPath = "c:\excel\  
    sExt = ".xls"  
    sName = InputBox("Quellarbeitsmappe:")  
    If sName = "" Then Exit Sub  
    sName = sPath & sName & sExt  
    var = ActiveWorkbook.LinkSources(xlExcelLinks)  
    If Not IsEmpty(var) Then  
        For iCounter = 1 To UBound(var)  
            ActiveWorkbook.ChangeLink _  
                Name:=var(iCounter), newname:=sName  
        Next iCounter  
    End If  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle externen Links löschen

Problem:	Wie kann ich über VBA alle externen Links eines Arbeitsblattes löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub LinkLoeschen() Dim rng As Range For Each rng In Worksheets(2).UsedRange.Cells If InStr(rng.Formula, "[") > 0 And _ InStr(rng.Formula, "]") > 0 Then rng.ClearContents End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Links zu einer bestimmten Arbeitsmappe löschen

Problem:	Wie kann ich in einem Zug alle Links zu einer bestimmten Arbeitsmappe in einem Arbeitsblatt entfernen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub LinkLoeschen() Dim rng As Range For Each rng In Worksheets(2).UsedRange.Cells If InStr(LCase(rng.Formula), "[test.xls]") > 0 Then rng.ClearContents End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Externe Verknüpfungen lokal umleiten

Problem:	Wie kann ich Verknüpfungen zu anderen Arbeitsmappen entfernen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DeleteLinks() Dim vLink As Variant Dim iCounter As Integer vLink = ActiveWorkbook.LinkSources(xlExcelLinks) If Not IsEmpty(vLink) Then For iCounter = 1 To UBound(vLink) ActiveWorkbook.ChangeLink Name:=vLink(iCounter), _ newname:=ThisWorkbook.Name Next iCounter End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlink-Formatierung bei Eingabe von Email-Adressen verhindern

Problem:	Es soll verhindert werden, dass die Excel-Autoformatierung einen Email-Hyperlink aus einem Eintrag mit dem Zeichen @ macht. Statt einem @ ist ein µ einzugeben. Die Umwandlung erfolgt später.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) Application.DisplayAlerts = False Application.EnableEvents = False Cells.Replace "µ", "@" Application.EnableEvents = True Application.DisplayAlerts = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamischer Hyperlink

Problem:	Der Hyperlink soll dynamisch - relativ zum Pfad der Arbeitsmappe - gehalten werden.
Lösung:	Folgende Formeln: a) =LINKS(ZELLE("dateiname";A1);FINDEN("[";ZELLE("dateiname";A1))-1) b) =HYPERLINK(A1&"test1.xls")
VBA-Code:	Ohne VBA-Programmierung

Prüfung, ob Datei- und/oder Web-Hyperlink-Adressen existieren

Problem:	In einem neuen Tabellenblatt sollen die Hyperlinks mit der Angabe, ob die Datei bzw. die Webseite existiert, gelistet werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: basLocal Sub CheckHyperlinks() Dim hyps As Hyperlinks Dim hyp As Hyperlink Dim iRow As Integer Dim sAddress As String, sDNS As String Dim bln As Boolean Set hyps = ActiveSheet.Hyperlinks Worksheets.Add For Each hyp In hyps bln = False iRow = iRow + 1 Cells(iRow, 1) = hyp.Address sDNS = hyp.Address If Left(hyp.Address, 4) = "http" Then sDNS = Right(sDNS, Len(sDNS) - 7) If Right(sDNS, 1) = "/" Then sDNS = Left(sDNS, Len(sDNS) - 1) End If sAddress = WwwAddress(sDNS) If sAddress <> "" Then bln = True Else If Dir(hyp.Address) <> "" Then bln = True End If Cells(iRow, 2) = bln Next hyp Columns.AutoFit End Sub StandardModule: basWeb Private Declare Function WSAGetLastError Lib "WSOCK32.DLL" () _ As Long Private Declare Function WSASStartup Lib "WSOCK32.DLL" (ByVal _ wVersionRequired&, lpWSAData As WinSocketDataType) _ As Long Private Declare Function WSACleanup Lib "WSOCK32.DLL" () _ As Long Private Declare Function gethostname Lib "WSOCK32.DLL" (ByVal _ HostName\$, ByVal HostLen%) As Long Private Declare Function gethostbyname Lib "WSOCK32.DLL" _ (ByVal HostName\$) As Long Private Declare Function gethostbyaddr Lib "WSOCK32.DLL" _ (ByVal addr\$, ByVal laenge%, ByVal typ%) As Long Private Declare Sub RtlMoveMemory Lib "kernel32" (hpvDest As _ Any, ByVal hpvSource&, ByVal cbCopy&) Const WS_VERSION_REQD = &H101 Const SOCKET_ERROR = -1 Const WSADESCRIPTION_LEN = 256 Const WSASYS_STATUS_LEN = 128 Private Type HostDeType hName As Long hAliases As Long hAddrType As Integer hLength As Integer hAddrList As Long End Type Private Type WinSocketDataType wversion As Integer wHighVersion As Integer szDescription(0 To WSADESCRIPTION_LEN) As Byte szSystemStatus(0 To WSASYS_STATUS_LEN) As Byte iMaxSockets As Integer iMaxUdpDg As Integer lpszVendorInfo As Long End Type Private Declare Function RasEnumConnections Lib "RasApi32.DLL" _ Alias "RasEnumConnectionsA" (lpRasCon As Any, lpcb As _ Long, lpcConnections As Long) As Long Private Declare Function RasGetConnectStatus Lib "RasApi32.DLL" _ Alias "RasGetConnectStatusA" (ByVal hRasCon As Long, _ lpStatus As Any) As Long Private Declare Function RasEnumEntries Lib "RasApi32.DLL" _ Alias "RasEnumEntriesA" (ByVal reserved\$, ByVal _ lpszPhonebook\$, lprasentryname As Any, lpcb As Long, _ lpcEntries As Long) As Long Private Declare Function RasHangUp Lib "RasApi32.DLL" _ Alias "RasHangUpA" (ByVal hRasConn As Long) As Long Const RAS_MaxEntryName = 256 Const RAS_MaxDeviceType = 16 Const RAS_MaxDeviceName = 32 Private Type RASType dwSize As Long hRasCon As Long szEntryName(RAS_MaxEntryName) As Byte szDeviceType(RAS_MaxDeviceType) As Byte szDeviceName(RAS_MaxDeviceName) As Byte End Type Private Type RASStatusType dwSize As Long RasConnState As Long dwError As Long szDeviceType(RAS_MaxDeviceType) As Byte szDeviceName(RAS_MaxDeviceName) As Byte End Type Private Type RASENTRYNAME95 dwSize As Long szEntryName(RAS_MaxEntryName) As Byte End Type Dim DFÜName\$ Sub Aufruf() Dim sName As String sName = WwwAddress("http://www.herber.de") End Sub Function WwwAddress(strAddress As String) Dim X% Dim IP\$, DNS\$, HOST\$ If Not Online Then Exit Function Call InitSockets WwwAddress = HostByName(strAddress) Call CleanSockets End Function Private Sub InitSockets() Dim Result% Dim LoBy%, HiBy% Dim SocketData As WinSocketDataType Result = WSASStartup(WS_VERSION_REQD, SocketData) If Result <> 0 Then MsgBox ("'winsock.dll' antwortet nicht !") End End If End Sub Private Function HostByName(Name\$, Optional X% = 0) As String Dim MemIp() As Byte Dim Y% Dim HostDeAddress&, HostIp& Dim IPAddress\$ Dim HOST As HostDeType HostDeAddress = gethostbyname(Name) If HostDeAddress = 0 Then HostByName = "" Exit Function End If Call RtlMoveMemory(HOST, HostDeAddress, LenB(HOST)) For Y = 0 To X Call RtlMoveMemory(HostIp, HOST.hAddrList + 4 * Y, 4) If HostIp = 0 Then HostByName = "" Exit Function End If Next Y ReDim MemIp(1 To HOST.hLength) Call RtlMoveMemory(MemIp(1), HostIp, HOST.hLength) IPAddress = "" For Y = 1 To HOST.hLength IPAddress = IPAddress & MemIp(Y) & "." Next Y IPAddress = Left\$(IPAddress, Len(IPAddress) - 1) HostByName = IPAddress End Function Private Function DFÜStatus() As Boolean Dim RAS(255) As RASType, RASStatus As RASStatusType Dim lg&, lpcon&, Result& RAS(0).dwSize = 412 lg = 256 * RAS(0).dwSize Result = RasEnumConnections(RAS(0), lg, lpcon) If lpcon = 0 Then DFÜStatus = False Else RASStatus.dwSize = 160 Result = RasGetConnectStatus(RAS(0).hRasCon, RASStatus) If RASStatus.RasConnState = &H2000 Then DFÜStatus = True Else DFÜStatus = False End If End If End Function Private Function Online() As Boolean Dim Test As Boolean Test = DFÜStatus If Test = False Then MsgBox _ ("Keine Online Verbindung vorhanden ! Bitte einwählen !") Online = Test End Function Private Sub CleanSockets() Dim Result& Result = WSACleanup() If Result <> 0 Then MsgBox ("Socket Error " & Trim\$(Str\$(Result)) & _ " in Prozedur 'CleanSockets' aufgetreten !") End End If End Sub Private Function NextChar(Text\$, Char\$) As String Dim POS% POS = InStr(1, Text, Char) If POS = 0 Then NextChar = Text Text = "" Else NextChar = Left\$(Text, POS - 1) Text = Mid\$(Text, POS + Len(Char)) End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zwei Hyperlinks in einer Zelle

Problem:	In einer Zelle sollen zwei Hyperlinks untergebracht werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlinks zu SVERWEIS-Bezugstabellen anlegen

Problem:	Wie kann ich zu einer Reihe von SVERWEISEN automatisch Hyperlinks zu den Bezugstabellen anlegen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub HyperlinksAnlegen() Dim iRow As Integer, iStart As Integer, iEnd As Integer Dim sTarget As String For iRow = 1 To 9 sTarget = Cells(iRow, 2).FormulaLocal iStart = InStr(sTarget, ";") + 1 iEnd = InStr(iStart, sTarget, ";") sTarget = Mid(sTarget, iStart, iEnd - iStart) ActiveSheet.Hyperlinks.Add Anchor:=Cells(iRow, 2), _ Address:="", SubAddress:=sTarget Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

URL's eines ausgewählten Bereiches in HyperLinks umwandeln

Problem:	In einer Tabelle stehen in verschiedenen Zellen Internetadressen. Wie kann ich diese Zellen in Hyperlinks umwandeln, ohne die Adressen im HyperLink-Dialog neu eintippen zu müssen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SetHyperlinks_Click() Dim rng As Range For Each rng In Range("A1").CurrentRegion.Cells If Left(rng.Value, 7) = "http://" Then rng.Hyperlinks.Add rng, rng.Value End If Next rng End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlink aufteilen

Problem: Wie kann ich die Informationen einer Zelle mit einem Hyperlink so aufteilen, daß in einer Zelle der Text und in einer anderen die Adresse der Hyperlinkzelle stehen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle3

Private Sub CommandButton1_Click()
    Call LinkAufteilen
End Sub

StandardModule: basMain

Sub LinkAufteilen()
    With Range("A1")
        .Offset(0, 1) = .Value
        .Offset(0, 2) = .Hyperlinks(1).Address
    End With
    Columns("A:C").AutoFit
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verknüpfungen in 2. Blatt transponiert anlegen

Problem: Wie kann ich in Tabelle2 Verknüpfungen zu einem Zellbereich in Tabelle2 in transponierter Form anlegen, wobei der Bereich zum Einfügen bei Zelle D1 beginnen soll?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Transponieren()
    Dim iCounter As Integer
    For iCounter = 1 To 4
        With Worksheets("Tabelle2")
            .Cells(1, iCounter + 3).Formula = _
                "=Tabelle1!" & Cells(iCounter, 1).Address
        End With
    Next iCounter
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Handhabung von Arrays in benutzerdefinierten Funktionen

Problem:	Wie kann ich Matrizen in benutzerdefinierten Funktionen handhaben? Wie kann ich einzelne Werte zurückgeben lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Probieren() Dim arrBasis As Variant Dim arrErgebnis As Variant arrBasis = Array(1, 2, 3, 4, 5, 6) arrErgebnis = Berechnen(arrBasis) MsgBox arrErgebnis(2) End Sub Private Function Berechnen(Basis As Variant) As Variant Dim iCounter As Integer, iValue As Integer Dim Ergebnis() For iCounter = 0 To UBound(Basis) - 1 If Basis(iCounter) Mod 2 = 0 Then iValue = iValue + 1 ReDim Preserve Ergebnis(iValue) Ergebnis(iValue) = Basis(iCounter) End If Next iCounter Berechnen = Ergebnis End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ordnernamen eines Verzeichnisses in Zelle einfügen

Problem:	Wie kann ich die Ordnernamen eines in einer Zelle stehenden Verzeichnisnamens über eine benutzerdefinierte Funktion ermitteln?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function DirNames(sPath As String) As Variant Dim arr() As String Dim iCounter As Integer With Application.FileSearch .LookIn = sPath .Execute ReDim arr(1 To .FoundFiles.Count) For iCounter = 1 To .FoundFiles.Count arr(iCounter) = .FoundFiles(iCounter) Next iCounter End With DirNames = arr End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ausgabe einer benutzerdefinierten Funktion in Matrixformel

Problem: Wie kann ich über eine benutzerdefinierte Funktion mehrere Ergebnisse ausgeben lassen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function Berechnung( _
    dLength As Double, _
    dWidth As Double, _
    dHeight As Double) As Variant
    Dim arr(0 To 2) As Double
    arr(0) = dLength * dWidth
    arr(1) = dLength * dWidth * dHeight
    arr(2) = 2 * (dLength * dWidth + dLength _
        * dHeight + dWidth * dHeight)
    Berechnung = arr
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl nach 2 Kriterien ermitteln mit DB-Funktion und MatrixFormel

Problem:	Wie kann ich die Anzahl von Datensätzen nach zwei Kriterien ermitteln.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung, wie oft ein Name in einem Monat vorkommt

Problem: Im Blatt Auswertung wird ermittelt, welcher Name in welchem Monat wie oft vorkommt.

Lösung: Nur anhand der zugehörigen Beispielarbeitsmappe darstellbar.

VBA-Code:

```
ClassModule: UserForm1

Private Sub UserForm_Click()
    ActiveCell.Value = Calendar1.Value
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten in Matrix indizieren

Problem:	Die nach Kalenderwochen gelisteten Werte aus Tabelle 2 sollen in Tabelle1 gelistet werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eine Zusammenstellung von Matrix-Funktionen

Problem: Der Einsatz von Matrixfunktionen wird in 26 Tabellenblättern demonstriert.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl der Monatstage gem. Zeitrahmen suchen

Problem:	Die Matrixformeln im Bereich B2:M2 liefern die Anzahl der Tage des jeweiligen Monats gemäß Zeitrahmen aus den Zellen A1:A2.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kosten nach mehreren Kriterien erfassen

Problem:	Kosten sollen nach mehreren Kriterien erfaßt und analysiert werden. Die Lösung beruht auf Matrixformeln.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Matrixfunktion über VBA eintragen, auslesen und löschen

Problem:	Über VBA wird eine Matrixfunktion in eine Tabelle eingetragen, ausgelesen, das Ergebnis wird gemeldet und die Matrixfunktion wieder gelöscht.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub ComputeArray() Application.ScreenUpdating = False With Range("F1") .FormulaArray = _ "=sum((A1:A18="" & Range("E1").Value & """)*(B1:B18="" & _ Range("E2").Value & """))" If Not IsError(.Value) Then MsgBox "Anzahl der Übereinstimmungen: " & .Value Else MsgBox "Ergebnis konnte nicht berechnet werden!" End If .ClearContents End With Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eine Zeile mit vorgegebenem Interall summieren

Problem: Zeile 1 soll nach dem in Zelle B3 vorgegebenem Intervall summiert werden.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code:



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datensätze zählen, die in einigen Spalten identisch sind

Problem:	Es sollen zeilenweise alle Datensätze gezählt werden, die in den Spalten D, F, G und I identisch sind. Bei den Formeln in Spalte J handelt es sich um Matrixformeln.
Lösung:	Matrixformel: =SUMME((\$D\$1:\$D\$10=D1)*(\$F\$1:\$F\$10=F1)*(\$G\$1:\$G\$10=G1)*(\$I\$1:\$I\$10=I1))
VBA-Code:	<pre>StandardModule: Telefonieren Sub Kopieren() Dim sPath As String sPath = Application.Path & "\" ThisWorkbook.VBProject _ .VBComponents("Telefonieren").Export sPath & "Telefonieren.bas" Workbooks.Add With ActiveWorkbook.VBProject .VBComponents.Import sPath & "Telefonieren.bas" .VBComponents("Telefonieren").Name = "Telefonieren" End With Kill sPath & "\"Telefonieren.bas" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahlenmarkierungen in einer Lottoliste

Problem:	Die Werte im Bereich A3:AW15 sollen bei Übereinstimmung mit den Werten in E19:J19 gelb hinterlegt werden. In Zeile 16 soll die Anzahl der Übereinstimmungen je Spalte ermittelt werden.
Lösung:	Darstellung nur anhand einer Beispielarbeitsmappe möglich.
VBA-Code:	



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Unter Berücksichtigung von Anfang / Ende-Datumsangaben summieren

Problem:	Der Wertebereich in Spalte B gemäß den Anfang/Ende-Vorgaben in Tabelle 2 soll summiert werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeilen zählen, in denen zwei Bedingungen zutreffen

Problem:	In Zelle A1 sollen die Zeilen gezählt werden, in denen beide in der Formel genannten Bedingungen erfüllt sind.
Lösung:	Folgende Matrixformel: =SUMME((E1:E1000=0)*(G1:G1000=2)*1)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werteermittlung unter Berücksichtigung von 2 Kriterien

Problem:	Aus einer Tabelle sollen Werte unter Zugrundelegung von zwei Kriterien ermittelt werden.
Lösung:	Folgende Formel: =SUMME((Verkäufer=Kriterium1)*(Produkt=Kriterium2)*Preis)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung von Anzahl und Summe mit 2 Kriterien

Problem:	Anzahl und Summe sollen unter Berücksichtigung von 2 Kriterien ermittelt werden.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Addition von Werten links und rechts eines Doppelpunktes

Problem:	Die Werte links und rechts eines Doppelpunktes (z.B. Spielergebnisse) sollen addiert werden.
Lösung:	Folgende Matrixformel: =SUMME(WERT(LINKS(A1:A6;FINDEN("";";A1:A6)-1)))&";";&SUMME(WERT(RECHTS(A1:A6;LÄNGE(A1:A6)-FINDEN("";";A1:A6))))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus Liste Anzahl der Januartage eines bestimmten Jahres listen

Problem:	Aus einer Datenreihe einer Tabelle soll die Anzahl der Januartage eines bestimmten Jahres ermittelt werden.
Lösung:	Matrixformel: =SUMME((MONAT(Tabelle1!\$A1:A40)=1)*(JAHR(Tabelle1!A1:A40)=2002)*1)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zahlen unter Berücksichtigung von Monat und Jahr summieren

Problem:	Die Einträge zu einem jeweiligen Monat im Bereich A1:A100 sollen unter Berücksichtigung des Jahres gezählt werden.
Lösung:	Matrixformel: =SUMME((MONAT(A1:A100)=B1)*(JAHR(A1:A100)=C1)*1)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung von Werten aus einer Aufmaßtabelle

Problem:	Gemäß den Vorgaben in den Zellen G2:G3 sollen die Summen aus der Tabelle, Spalte D ermittelt werden.
Lösung:	Matrixformel: =SUMME((A3:A100=G2)*(B3:B100=G3)*D3:D100)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl von Werten nach verschiedenen Bedingungen.

Problem:	Die Anzahl der in der Tabelle A4:C405 mit den Kriterien im Bereich A2:C2 übereinstimmenden Daten soll ausgelesen werden.
Lösung:	Matrixformel: =SUMME((A4:A405=A2)*(B4:B405=B2)*(C4:C405=C2)*1)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Stunden und Minuten getrennt summieren

Problem:	Aus den Sportergebnissen aus Spalte B sollen die ersten und letzten beiden Stellen summiert werden.
Lösung:	Matrixformel: =SUMME(WERT(LINKS(A1:A3;2)))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summen unter Berücksichtigung von Ober- und Untergrenzen

Problem:	Wie kann ich Spalte A auf Werte nach vorgegebenen Ober- und Untergrenzen prüfen und bei Übereinstimmung die Zellen rechts daneben summieren?
Lösung:	Matrixformel: =SUMME((\$A\$2:\$A\$9>=D1)*(\$A\$2:\$A\$9<=D2)*\$B\$2:\$B\$9)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl der Übereinstimmungen in zwei Bereichen ermitteln

Problem:	Wie kann ich ermitteln, welche Werte in zwei Bereichen an gleicher Position erscheinen?
Lösung:	Folgende Matrixformel: =SUMME((A1:A20=C1:C20)*(B1:B20=D1:D20))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabelle nach mehreren Kriterien summieren

Problem:	Wie kann ich einen Tabellenbereich nach mehreren Kriterien summieren. Die Excel-Funktion SUMMEWENN akzeptiert nur ein Kriterium.
Lösung:	Folgende Matrixformel: {=SUMME((A1:A100="K-FF 1655")*(B1:B100="Hespe")*C1:C100)}
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte nach zwei Kriterien auswählen

Problem:	Wie kann ich Werte nach zwei Kriterien summieren? Sowohl die Funktion ZÄHLENWENN als auch SUMMEWENN akzeptieren nur ein Kriterium.
Lösung:	Die Matrixformel: {=SUMME((date=E2)*(Account=F2)*Amount)}
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten eines bestimmten Jahres filtern

Problem:	Wie kann ich die Vereinseintritte eines bestimmten Jahres über eine Funktion zählen lassen? Die Eintrittsdaten stehen in den Zellen A2:A1200.
Lösung:	Die nachfolgende Matrixfunktion zählt die Eintritte des Jahres 98: {=SUMME((JAHR(A2:A1200)=1998)*1)}
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Addition der absoluten Zahlen

Problem:	Wie kann ich die Summe der absoluten Zahlen einer Spalte ermitteln?
Lösung:	Matrixformel: =SUMME(ABS(A2:A9))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte eines bestimmten Monats summieren

Problem:	Wie kann ich, wenn ich in Spalte A die Tage eines Jahres im Datumformat und in Spalte B zu jedem Tag einen Wert eingetragen habe, die Summe der Werte des in Zelle C2 hinterlegten Monats feststellen?
Lösung:	Folgende Matrix-Formel: =SUMME(((MONAT(A1:A365))=C2)*(B1:B365))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summe aus Produkten

Problem:	Aus einer Tabelle mit den Spalten Artikel, Preis und Menge möchte ich die Summe aus den Produkten von Preis und Menge für bestimmte Artikel bilden.
Lösung:	Mit der Arrayformel: =SUMME((Artikel=E4)*(Einzelpreis*Menge))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eindimensionales Array in ein mehrdimensionales splitten

Problem:	Ein eindimensionales Array soll mit den Zahlen von 1 bis 100 gefüllt und dann in ein mehrdimensionales aufgeteilt werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub SplitArray() Dim arrBasis(1 To 100) As Integer Dim arrSplit(1 To 25, 1 To 4) As Integer Dim iCounter As Integer, iAct As Integer For iCounter = 1 To 100 arrBasis(iCounter) = iCounter Next iCounter For iCounter = 1 To 25 For iAct = 1 To 4 arrSplit(iCounter, iAct) = arrBasis(iCounter * 4 - (4 - iAct)) Next iAct Next iCounter Range("A1:D25").Value = arrSplit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eintragen von Matrixformeln in eine Jahres- / Monatsmatrix

Problem:	In eine Jahres-/Monatsmatrix sollen Arrayformeln zur Ermittlung der jeweiligen Werte aus einer Datentabelle eingetragen werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub DatenFiltern() Dim iYear As Integer, iMonth As Integer For iYear = 2001 To 2004 For iMonth = 1 To 12 Cells(iMonth + 1, iYear - 1997).FormulaArray = _ "=sum((year(a1:a100)=" & iYear & ")" & _ "*(month(a1:a100)=" & iMonth & ")*b1:b100)" Next iMonth Next iYear End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Temporärer VBA-Einsatz einer Matrixformel

Problem:	Über den temporären Einsatz einer Matrixformel sollen die Werte aus Spalte C nach zwei Bedingungen summiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub GetArray() Application.ScreenUpdating = False With Range("D1") .FormulaArray = _ "=sum((A1:A100=""a"")*(B1:B100=5)*c1:c100)" MsgBox .Text .ClearContents End With Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Redimensionierung eines Arrays über mehrere Dimensionen

Problem:	Eine Array-Redimensionierung wird nur zugelassen, wenn sich ausschließlich die letzte Datenfelddimension ändert. Wie kann ich das umgehen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub FreeArray() Dim iCol As Integer, iRow As Integer, iCounter As Integer Dim arr() Application.ScreenUpdating = False ReDim arr(1 To 2, 1 To 10) For iRow = 1 To 2 For iCol = 1 To 10 arr(iRow, iCol) = "Spalte " & iCol & " - " & "Zeile " & iRow Next iCol Next iRow MsgBox "Letzter Wert:" & Chr(10) & arr(2, 10) Workbooks.Add Range("A1:J2") = arr For iCounter = 1 To 11 Cells(3, iCounter) = "Spalte " & iCounter & " - " & "Zeile 3" Next iCounter For iCounter = 1 To 3 Cells(iCounter, 11) = "Zeile " & iCounter & " - " & "Spalte 11" Next iCounter arr = Range("A1").CurrentRegion ActiveWorkbook.Close savechanges:=False MsgBox "Letzter Wert:" & Chr(10) & arr(3, 11) Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in Array suchen, ohne jedes Datenfeld abzufragen

Problem:	Wie kann ich in einem Array einen Wert suchen, ohne jedes einzelne Datenfeld abzufragen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Durchsuchen() Dim arr As Variant Dim iRow As Integer arr = Range("A1:A16") iRow = Application.Match("Irgendwas", arr, 0) If Err > 0 Then MsgBox "Nicht gefunden" Else MsgBox "An " & iRow & " .Stelle gefunden" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus Zellbereich in eine Matrix übernehmen und auslesen

Problem: Wie kann ich einen Zellbereich aus einem Arbeitsblatt in eine Matrix ein- und dann einzelne Zellinhalte auslesen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub Einlesen()
    Dim MyValues As Variant
    MyValues = Range("A1:F12")
    MsgBox "Zweite Zeile, dritte Spalte: " & _
        MyValues(2, 3)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Variable in Excel-Formel einbauen.

Problem:	Wie kann ich in eine via VBA zu erstellende Excel-Formel eine Variable einbauen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Summebilden() Dim iRow As Integer, iValue As Integer iValue = 5 For iRow = 2 To 23 Cells(iRow, 4).FormulaArray = _ "=SUM((r2c1:r23c1=1)*(r2c3:r23c3=" & iRow & ")*(r2c[-2]:r23c[-2]))" Next iRow End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Entfernungsmatrix listen

Problem:	Wie kann ich eine Entfernungsmatrix in eine Auflistung umsetzen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MatrixListen() Dim rng As Range Dim iRow As Integer, iCol As Integer, iRowT As Integer Set rng = Range("A1").CurrentRegion With Worksheets("Liste") For iRow = 2 To rng.Rows.Count For iCol = 2 To rng.Rows.Count If Not IsEmpty(Cells(iRow, iCol)) Then iRowT = iRowT + 1 .Cells(iRowT, 1).Value = Cells(iRow, 1).Value .Cells(iRowT, 2).Value = Cells(1, iCol).Value .Cells(iRowT, 3).Value = Cells(iRow, iCol).Value End If Next iCol Next iRow .Select End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl der Wochentage in einem vorgegebenen Zeitraum ermitteln

Problem:	Wie kann ich das Vorkommen eines bestimmten Wochentages in einem vorgegebenen Zeitraum ermitteln?
Lösung:	Matrixformel: =SUMME(WENN(WOCHENTAG(B2-1+ZEILE(INDIREKT(""1:"&KÜRZEN(B3-B2)+1)))=B1;1;0))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte eines bestimmten Wochentages und eines Zeitbereiches addieren

Problem:	Wie kann ich aus einer Tabelle alle Werte eines bestimmten Wochentages und einem bestimmten Zeitbereich addieren?
Lösung:	Folgende Matrixformel: =SUMME((A1:A13>9/24)*(A1:A13<=18/24)*(WOCHENTAG(B1:B13,2)<6)*(C1:C13))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblatt-Menü-, Format- und Standardsymbolleiste einblenden

Problem:	Ein Klick auf die Schaltfläche blendet die Arbeitsblatt-Menüleiste, die Standard- und die Format-Symbolleiste ein, falls diese ausgeblendet sind.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CmdBarEinAus() With Application.CommandBars("Worksheet Menu Bar") .Enabled = True .Visible = True End With Application.CommandBars("Standard").Visible = True Application.CommandBars("Formatting").Visible = True End Sub</pre>

	Herbers Excel-FAQ - Die Texte Bestandteil der Excel-Lehr- und Lernmaterialien
Symbolleiste zum Aufruf externer Anwendungen erstellen	
Problem:	Über ein Schaltfläche soll eine Symbolleiste gem. Tabelle erstellt werden. Bei Anklicken eines der Menüpunkte wird die in Spalte A genannte Anwendung mit der in Spalte C genannten Datei aus dem in Spalte E genannten Verzeichnis aufgerufen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn den Schaltflächen zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) On Error GoTo ERRORHANDLER Application.CommandBars("ApplicationsBar").Delete ERRORHANDLER: End Sub StandardModule: Modul1 Sub CreateCmdBar() Dim oBar As CommandBar Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Dim iRow As Integer, iRowL As Integer On Error Resume Next Application.CommandBars("ApplicationsBar").Delete On Error GoTo 0 Set oBar = Application.CommandBars.Add("ApplicationsBar", msoBarTop) iRowL = Cells(Rows.Count, 4).End(xlUp).Row For iRow = 2 To iRowL If Not IsEmpty(Cells(iRow, 1)) Then Set oPopUp = oBar.Controls.Add(msoControlPopup) oPopUp.Caption = Cells(iRow, 2).Value Do Set oBtn = oPopUp.Controls.Add With oBtn .Style = msoButtonIconAndCaption .Caption = Cells(iRow, 3).Value .FaceId = Cells(iRow, 4).Value .OnAction = "GetFile" .DescriptionText = Cells(iRow, 5).Value .Tag = Cells(iRow, 1).Value End With iRow = iRow + 1 If iRow > iRowL Then Exit Do Loop While IsEmpty(Cells(iRow, 1)) iRow = iRow - 1 End If Next iRow oBar.Visible = True End Sub Sub GetFile() Dim oBtn As CommandBarButton Dim sFile As String Set oBtn = Application.CommandBars.ActionControl sFile = oBtn.DescriptionText & "\" & oBtn.Caption If Dir(sFile) = "" Then Beep MsgBox "Die Datei " & sFile & " wurde nicht gefunden!" Exit Sub End If If Dir(oBtn.Tag) = "" Then Beep MsgBox "Die Anwendung " & oBtn.Tag & " wurde nicht gefunden!" Exit Sub End If Shell oBtn.Tag & " " & sFile, vbMaximizedFocus End Sub Sub DeleteCommandBar() On Error GoTo ERRORHANDLER Application.CommandBars("ApplicationsBar").Delete ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neue Menüleiste mit zwei Menüs und Schaltflächen erstellen

Problem: Es soll eine neue Menüleiste mit zwei Menüs und Menüschtfläche angelegt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub NewMenue()
    Dim oCmdBar As CommandBar
    Dim oPopUp As CommandBarPopup
    Dim oCmdBtn As CommandBarButton
    Dim datDay As Date
    Dim iMonths As Integer
    Call DeleteMenueBar
    Set oCmdBar = Application.CommandBars.Add( _
        "MyCommandBar", msoBarTop, False, True)
    Set oPopUp = oCmdBar.Controls.Add(msoControlPopup)
    oPopUp.Caption = "Prüfung"
    For iMonths = 1 To 12
        Set oCmdBtn = oPopUp.Controls.Add
        With oCmdBtn
            .Caption = Format(DateSerial(1, iMonths, 1), "mmm") & " Druck"
            .OnAction = ""
            .Style = msoButtonCaption
        End With
    Next iMonths
    Set oPopUp = oCmdBar.Controls.Add(msoControlPopup)
    oPopUp.Caption = "Monatsbericht"
    For iMonths = 1 To 12
        Set oCmdBtn = oPopUp.Controls.Add
        With oCmdBtn
            .Caption = Format(DateSerial(1, iMonths, 1), "mmm") & " Druck"
            .OnAction = ""
            .Style = msoButtonCaption
        End With
    Next iMonths
    oCmdBar.Visible = True
End Sub

Sub DeleteMenueBar()
    On Error Resume Next
    Application.CommandBars("MyCommandBar").Delete
    On Error GoTo 0
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benutzerdefinierte Menüleiste vor Veränderungen schützen

Problem: Eine benutzerdefinierte Symbolleiste soll vor Veränderungen durch den Anwender geschützt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error GoTo ERRORHANDLER
    Application.CommandBars("MeineBar").Delete
ERRORHANDLER:
End Sub

Private Sub Workbook_Open()
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Dim iCounter As Integer
    On Error Resume Next
    Application.CommandBars("MeineBar").Delete
    On Error GoTo 0
    Set oBar = Application.CommandBars.Add( _
        Name:="MeineBar", _
        Position:=msoBarTop, _
        MenuBar:=False, _
        temporary:=True)
    For iCounter = 1 To 10
        Set oBtn = oBar.Controls.Add
        With oBtn
            .Caption = "Menüpunkt " & iCounter
            .Style = msoButtonCaption
        End With
    Next iCounter
    oBar.Protection = msoBarNoCustomize
    oBar.Visible = True
End Sub
```



Jahreskalender als Menü anlegen

Problem:	Es soll eine Menüleiste als Jahreskalender mit Monaten als Menüs und Tagen als Menüpunkten gebildet werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: Modul1 <pre>Sub NewMenue() Dim oCmdBar As CommandBar Dim oPopUp As CommandBarPopup Dim oCmdBtn As CommandBarButton Dim datDay As Date Dim iMonths As Integer, iDays As Integer, iCount As Integer On Error Resume Next Application.CommandBars(CStr(Year(Date))).Delete On Error GoTo 0 Set oCmdBar = Application.CommandBars.Add(_ CStr(Year(Date)), msoBarTop, False, True) For iMonths = 1 To 12 Set oPopUp = oCmdBar.Controls.Add(msoControlPopup) With oPopUp .Caption = Format(DateSerial(1, iMonths, 1), "mmm") If iMonths Mod 3 = 1 And iMonths <> 1 Then .BeginGroup = True iCount = Day(DateSerial(Year(Date), iMonths + 1, 0)) For iDays = 1 To iCount datDay = DateSerial(Year(Date), iMonths, iDays) Set oCmdBtn = oPopUp.Controls.Add With oCmdBtn .Caption = Day(datDay) & " - " & Format(datDay, "ddd") .Style = msoButtonCaption .OnAction = "GetDate" If Weekday(datDay) = 1 And iDays <> 1 Then .BeginGroup = True End With Next iDays End With Next iMonths oCmdBar.Visible = True End Sub Sub DeleteMenue() On Error Resume Next Application.CommandBars(CStr(Year(Date))).Delete On Error GoTo 0 End Sub Sub GetDate() Dim iYear As Integer, iMonth As Integer, iDay As Integer Dim iGroupM As Integer, iGroupD As Integer iYear = Year(Date) iMonth = WorksheetFunction.RoundUp(Application.Caller(2) - _ (Application.Caller(2) / 4), 0) iDay = Application.Caller(1) - GetGroups(iMonth, Application.Caller(1)) MsgBox Format(DateSerial(iYear, iMonth, iDay), "ddd - dd. mmm yyyy") End Sub Private Function GetGroups(iActMonth As Integer, iActDay As Integer) Dim iGroups As Integer, iCounter As Integer iCounter = 1 With Application.CommandBars(CStr(Year(Date))).Controls(iActMonth) Do While iCounter <= .Controls.Count If .Controls(iCounter).BeginGroup = True Then iGroups = iGroups + 1 End If If iCounter = iActDay - iGroups Then Exit Do iCounter = iCounter + 1 Loop GetGroups = iGroups End With End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei Blattwechsel anlegen einer Symbolleiste mit Zellwerten

Problem: Bei Blattwechsel sollen die Eintragungen in den jeweiligen Blattbereichen A1:E1 als Symbolleiste angelegt werden.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der UserForm ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    Application.CommandBars("Tabelle1").Delete
    Application.CommandBars("Tabelle2").Delete
    Application.CommandBars("Tabelle3").Delete
    On Error GoTo 0
End Sub

Private Sub Workbook_Open()
    Call CreateCmdBar
End Sub

Private Sub Workbook_SheetActivate(ByVal Sh As Object)
    On Error Resume Next
    Select Case Sh.Name
        Case "Tabelle1"
            Application.CommandBars("Tabelle2").Delete
            Application.CommandBars("Tabelle3").Delete
            Call CreateCmdBar
        Case "Tabelle2"
            Application.CommandBars("Tabelle1").Delete
            Application.CommandBars("Tabelle3").Delete
            Call CreateCmdBar
        Case "Tabelle3"
            Application.CommandBars("Tabelle1").Delete
            Application.CommandBars("Tabelle2").Delete
            Call CreateCmdBar
    End Select
    On Error GoTo 0
End Sub

Sub CreateCmdBar()
    Dim cmdBar As CommandBar
    Dim cmdBtn As CommandBarButton
    Dim iCounter As Integer
    Set cmdBar = Application.CommandBars.Add(ActiveSheet.Name, _
        msoBarTop, False, True)
    For iCounter = 1 To 5
        Set cmdBtn = cmdBar.Controls.Add
        With cmdBtn
            .Caption = ActiveSheet.Cells(1, iCounter).Value
            .OnAction = ""
            .Style = msoButtonCaption
        End With
    Next iCounter
    cmdBar.Visible = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neue Symbolleiste am rechten Fensterrand anlegen

Problem:	Am rechten Fensterrand soll eine neue Symbolleiste angelegt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub StandardModule: basMain Sub CreateCmdBar() Dim oBar As CommandBar Call CmdDelete Set oBar = Application.CommandBars.Add(_ "test", msoBarRight, False, False) oBar.Visible = True End Sub Sub CmdDelete() On Error GoTo ERRORHANDLER Application.CommandBars("test").Delete ERRORHANDLER: End Sub</pre>



Anzeige einer benutzerdefinierte Menüleiste bei Tabelle2

Problem:	Bei Aktivierung von Tabelle 2 soll eine benutzerdefinierte Menüleiste angezeigt, bei Deaktivierung ausgeblendet werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	ClassModule: DieseArbeitsmappe <pre>Private Sub Workbook_BeforeClose(Cancel As Boolean) On Error GoTo 0 Application.CommandBars("MeineMenueLeiste").Delete On Error GoTo 0 End Sub Private Sub Workbook_Open() Dim objBar As CommandBar Dim objPopUp As CommandBarPopup Dim objBtn As CommandBarButton Dim strBar As String strBar = "MeineMenueLeiste" On Error Resume Next Application.CommandBars(strBar).Delete On Error GoTo 0 Set objBar = Application.CommandBars.Add(_ Name:=strBar, _ Position:=msoBarTop, _ MenuBar:=True, _ Temporary:=True) Set objPopUp = objBar.Controls.Add(Type:=msoControlPopup) objPopUp.Caption = "Meine Dateien" Set objBtn = objPopUp.Controls.Add With objBtn .Caption = "Öffnen" .OnAction = "FilesOpen" .Style = msoButtonCaption End With 'objBar.Visible = True End Sub</pre> StandardModule: Modul1 <pre>Sub FilesOpen() MsgBox "Kein Code hinterlegt!" End Sub</pre> ClassModule: Tabelle2 <pre>Private Sub Worksheet_Activate() With Application .CommandBars("Worksheet Menu Bar").Enabled = False With .CommandBars("MeineMenueLeiste") .Enabled = True .Visible = True End With End With End Sub Private Sub Worksheet_Deactivate() With Application .CommandBars("Worksheet Menu Bar").Enabled = True With .CommandBars("MeineMenueLeiste") .Enabled = False .Visible = False End With End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Menü- und Symbolleisten mit Ihren Schaltflächen auflisten

Problem:	Wie kann ich alle Menü- und Symbolleisten in einem Tabellenblatt auflisten?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub IDFinder() Dim oBar As CommandBar Dim oCtr As CommandBarControl Dim oBtn As CommandBarButton Dim iRow As Integer Range("A1").Value = "Symbolleiste" Range("B1").Value = "Lokaler Name" Range("C1").Value = "Sichtbar" Range("D1").Value = "Schaltfläche" Range("E1").Value = "ID" Range("A1:E1").Font.Bold = True iRow = 1 For Each oBar In CommandBars If oBar.BuiltIn Then iRow = iRow + 1 Cells(iRow, 1) = oBar.Name Cells(iRow, 2) = oBar.NameLocal Cells(iRow, 3) = oBar.Visible For Each oCtr In oBar.Controls If oCtr.BuiltIn Then Cells(iRow, 4) = oCtr.Caption Cells(iRow, 5) = oCtr.ID iRow = iRow + 1 End If Next oCtr End If Next oBar Columns.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen Menüleiste anlegen und auf "ganzen Bildschirm" schalten

Problem:	Wie kann ich beim Öffnen einer Arbeitsmappe eine eigene Menüleiste anlegen und anzeigen lassen. Außerdem soll auf "Ganzer Bildschirm" geschaltet werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub Private Sub Workbook_Open() Dim oBar As CommandBar Dim oBtn As CommandBarButton Call CmdDelete Set oBar = Application.CommandBars.Add(_ Name:="MyFiles", _ Position:=msoBarTop, _ MenuBar:=True) Set oBtn = oBar.Controls.Add With oBtn .Caption = "Laden" .Style = msoButtonCaption End With Set oBtn = oBar.Controls.Add With oBtn .Caption = "Speichern" .Style = msoButtonCaption End With Set oBtn = oBar.Controls.Add With oBtn .Caption = "Erstellen" .Style = msoButtonCaption End With Set oBtn = oBar.Controls.Add With oBtn .Caption = "Zurück" .OnAction = "CmdDelete" .Style = msoButtonCaption End With Set oBtn = oBar.Controls.Add(ID:=178) Application.DisplayFullScreen = True oBar.Visible = True End Sub StandardModule: basMain Sub CmdDelete() On Error GoTo ERRORHANDLER Application.CommandBars("MyFiles").Delete Application.DisplayFullScreen = False ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblattmenüleiste ein- und ausblenden

Problem:	Wie kann ich die Arbeitsblattmenüleiste aus- und einblenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CmdBarEinAus() With Application.CommandBars("Worksheet Menu Bar") .Enabled = Not .Enabled End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eigene Symbolleiste nur beim Schließen löschen

Problem: Wie kann ich veranlassen, daß bei einer Verneinung der Rückfrage, ob eine Arbeitsmappe tatsächlich geschlossen werden soll, die beim Schließen zu löschende benutzerdefinierte Symbolleiste nicht gelöscht wird?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    If MsgBox("Tatsächlich schon wieder schließen?", _
        vbQuestion + vbYesNo) = vbYes Then
        On Error Resume Next
        Application.CommandBars("MeineLeiste").Delete
        On Error GoTo 0
        Cancel = False
    Else
        Cancel = True
    End If
End Sub

Private Sub Workbook_Open()
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Set oBar = Application.CommandBars.Add("MeineLeiste")
    Set oBtn = oBar.Controls.Add
    oBar.Visible = True
    oBtn.Caption = "Es ist nur ein Test"
    oBtn.Style = msoButtonCaption
    ActiveWorkbook.Close
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob eine Symbolleiste existiert

Problem: Wie kann ich beim Öffnen einer Arbeitsmappe über VBA prüfen, ob eine bestimmte Symbolleiste existiert?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Dim oBas As CommandBar
    On Error GoTo ERRORHANDLER
    Set oBas = _
        Application.CommandBars("MeineSymbolleiste")
    MsgBox "Die Symbolleiste" & _
        """"MeineSymbolleiste"" existiert!"
    Exit Sub
ERRORHANDLER:
    MsgBox "Die Symbolleiste" & _
        """"MeineSymbolleiste"" existiert nicht!"
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen neue Symbolleiste

Problem:	Wie kann ich beim Öffnen der Arbeitsmappe eine neue Symbolleiste erstellen, die beim Schließen wieder gelöscht wird?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call DeleteCmdBar End Sub Private Sub Workbook_Open() Dim oBar As CommandBar Dim oBtn As CommandBarButton Dim iCounter As Integer Call DeleteCmdBar Set oBar = Application.CommandBars.Add(_ Name:="NewToolbar", _ temporary:=True, _ Position:=msoBarTop) oBar.Visible = True For iCounter = 1 To 5 Set oBtn = oBar.Controls.Add With oBtn .FaceId = 70 + iCounter .Caption = "&" & iCounter & " Befehl" .OnAction = "Befehl" & iCounter End With Next iCounter End Sub StandardModule: basMain Sub Befehl1() MsgBox "Ich bin Befehl 1" End Sub Sub Befehl2() MsgBox "Ich bin Befehl 2" End Sub Sub Befehl3() MsgBox "Ich bin Befehl 3" End Sub Sub Befehl4() MsgBox "Ich bin Befehl 4" End Sub Sub Befehl5() MsgBox "Ich bin Befehl 5" End Sub Sub DeleteCmdBar() On Error Resume Next Application.CommandBars("NewToolbar").Delete On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Menüs und Symbolleisten ausblenden

Problem:	Wie kann ich alle Menü- und Symbolleisten einschließlich der Excel- und Tabellenblattschaltflächen in der oberech rechten Ecke des Fensters ausblenden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub AusBlenden() CommandBars("Worksheet Menu Bar").Enabled = False Application.DisplayFullScreen = True ActiveWorkbook.Protect Windows:=True End Sub Sub EinBlenden() ActiveWorkbook.Unprotect CommandBars("Worksheet Menu Bar").Enabled = True Application.DisplayFullScreen = False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auflistung aller CommandBars

Problem:	Wie kann ich in einem Arbeitsblatt alle Menü- und Symbolleisten mit ihrem System- und ihrem Namen in der lokalen Sprache auflisten lassen? In einer 3. Spalte soll angegeben werden, ob die CommandBar sichtbar oder nicht sichtbar ist.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub Menueleisten() Dim oBar As CommandBar Dim iRow As Integer Cells(1, 1) = "Name" Cells(1, 2) = "Lokaler Name" Cells(1, 3) = "Sichtbar" iRow = 1 For Each oBar In Application.CommandBars iRow = iRow + 1 Cells(iRow, 1) = oBar.Name Cells(iRow, 2) = oBar.NameLocal Cells(iRow, 3) = oBar.Visible Next oBar Columns("A:C").AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Löschen überflüssiger Symbolleisten

Problem:	Wie kann ich in XL8 überflüssige, benutzerdefinierte Symbolleisten löschen? Die Symbolleisten sollen abgefragt und angezeigt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub DeleteCmdBar() Dim oBar As CommandBar For Each oBar In Application.CommandBars If oBar.Visible And oBar.BuiltIn = False Then If MsgBox(_ prompt:=oBar.Name & " löschen?", _ Buttons:=vbYesNo + vbQuestion) = vbYes Then If MsgBox(_ prompt:"Tatsächlich?", _ Buttons:=vbYesNo + vbQuestion) = vbYes Then oBar.Delete End If End If End If Next oBar End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Position der Symbolleiste festlegen

Problem:	Wie kann ich über VBA-Code die Position einer Symbolleiste in XL8 bestimmen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call Loeschen End Sub ClassModule: Tabelle1 Private Sub CommandButton1_Click() Call NeueSymbolleiste End Sub Private Sub CommandButton2_Click() Call Loeschen End Sub StandardModule: basMain Sub NeueSymbolleiste() Dim oBar As CommandBar Set oBar = Application.CommandBars.Add(_ Name:="MyCommandbar", _ MenuBar:=False, _ temporary:=True) With oBar .Left = 50 .Top = 100 .Visible = True End With End Sub Sub Loeschen() On Error Resume Next Application.CommandBars("MyCommandbar").Delete On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Eigene Menüleiste mit Untermenüs erstellen

Problem:	Wie kann ich in XL8 eine eigene Menüleiste mit Untermenüs erstellen und diese anstelle der Tabellenblattmenüleiste anzeigen lassen.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call MenuLoeschen End Sub StandardModule: basMain Sub MenuErstellen() Dim oBar As CommandBar Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Set oBar = Application.CommandBars.Add(_ Name:="MyCommandBar", _ Position:=msoBarTop, _ MenuBar:=True, _ temporary:=True) Set oPopUp = oBar.Controls.Add(Type:=msoControlPopup) oPopUp.Caption = "Menü1" Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Erster Menüpunkt" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Zweiter Menüpunkt" .Style = msoButtonCaption End With Set oPopUp = oBar.Controls.Add(Type:=msoControlPopup) oPopUp.Caption = "Menü2" Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Erster Menüpunkt" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Zweiter Menüpunkt" .Style = msoButtonCaption End With CommandBars("MyCommandbar").Visible = True End Sub Sub MenuLoeschen() On Error Resume Next Application.CommandBars("MyCommandbar").Delete On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Symbolleisten ein- / ausblenden

Problem: Wie kann ich beim Laden einer Arbeitsmappe alle Symbolleisten ausund eine benutzerdefinierte Leiste einblenden? Nach dem Schließen der Arbeitsmappe soll der alte Zustand wieder hergestellt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Dim oBar As CommandBar
    Dim wks As Worksheet
    Dim iRow As Integer
    Set wks = ThisWorkbook.Worksheets("CmdBars")
    wks.Cells.ClearContents
    For Each oBar In Application.CommandBars
        If oBar.Visible And oBar.Type <> msoBarTypeMenuBar Then
            iRow = iRow + 1
            wks.Cells(iRow, 1).Value = oBar.Name
            oBar.Visible = False
        End If
    Next oBar
End Sub

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Dim oBar As CommandBar
    Dim iRow As Integer
    iRow = 1
    With ThisWorkbook.Worksheets("CmdBars")
        Do Until IsEmpty(.Cells(iRow, 1))
            Application.CommandBars(.Cells(iRow, 1).Value).Visible = True
            iRow = iRow + 1
        Loop
        .Cells.ClearContents
    End With
End Sub

Sub AusEinblenden()
    ThisWorkbook.Worksheets("CmdBars").Visible = xlVeryHidden
    'ThisWorkbook.Worksheets("CmdBars").Visible =true
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Menüleiste wechseln

Problem: Wie kann ich beim Öffnen der Arbeitsmappe eine eigene Menüleiste erstellen, die Arbeitsblattmenüleiste durch diese ersetzen und den Vorgang beim Schließen der Arbeitsmappe wieder rückgängig machen? pe die Tabellenblattmenüleiste ausblenden, eine benutzerdefinierte

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Dim oBar As CommandBar
    Dim oButton As CommandBarButton
    On Error Resume Next
    Application.CommandBars.Add("My Commandbar").Delete
    On Error GoTo 0
    Set oBar = Application.CommandBars.Add("My Commandbar", _
        MenuBar:=True, temporary:=True)
    Set oButton = oBar.Controls.Add
    With oButton
        .Caption = "My Button"
        .Style = msoButtonCaption
    End With
    Application.CommandBars("My Commandbar").Visible = True
End Sub

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    Application.CommandBars("My Commandbar").Delete
    On Error GoTo 0
    Application.CommandBars("Worksheet Menu Bar").Visible = True
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Menü's zurücksetzen

Problem: Ich habe die Menü- und Symbolleisten in verschiedenen Punkten verändert. Wie kann ich diese wieder zurücksetzen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub ResetMenuBars()
    Dim oBar As CommandBar
    For Each oBar In Application.CommandBars
        If oBar.Visible Then
            If oBar.BuiltIn Then
                oBar.Reset
            End If
        End If
    Next oBar
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Symbolleiste sichern

Problem:	Wie kann ich die aktuellen Einstellungen der Menü- und Symbolleisten sichern?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub SaveCommandBars() Dim sPath As String, sFile As String sPath = Range("B1").Value If Right(sPath, 1) <> "\" Then sPath = sPath & "\" End If sFile = Range("B2").Value FileCopy sPath & sFile, sPath & "Backup.xlb" End Sub</pre>



Blattschaltflächen in Menüs Schaltflächen konvertieren.

Problem:	Die ersten beiden Schaltflächen sollen dem Extras-Menü hinzugefügt bzw. aus dem Extras-Menü erstellt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub MeldungA() MsgBox "Aufruf durch erste Schaltfläche" End Sub Sub MeldungB() MsgBox "Aufruf durch zweite Schaltfläche" End Sub Sub ConvertToCommandButton() Dim cnt As CommandBarControl Dim btn As CommandBarButton Dim wksBtn As Button Set cnt = Application.CommandBars.FindControl(ID:=30007) Set btn = cnt.Controls.Add With btn .Caption = ActiveSheet.Buttons("Btn1").Caption .OnAction = ActiveSheet.Buttons("Btn1").OnAction .Style = msoButtonCaption End With Set btn = cnt.Controls.Add With btn .Caption = ActiveSheet.Buttons("Btn2").Caption .OnAction = ActiveSheet.Buttons("Btn2").OnAction .Style = msoButtonCaption End With Set wksBtn = ActiveSheet.Buttons.Add(_ Left:=Range("B11").Left, _ Top:=Range("B11").Top, _ Width:=Range("C12").Left + Range("C12").Width - Range("B12").Left, _ Height:=Range("B12:B13").Height) With wksBtn .Caption = "Buttons zum Blatt" .OnAction = "ConvertToButton" .Name = "Btn4" End With ActiveSheet.Buttons("Btn1").Delete ActiveSheet.Buttons("Btn2").Delete ActiveSheet.Buttons("Btn3").Delete End Sub Sub ConvertToButton() Dim cnt As CommandBarControl Dim btn As CommandBarButton Dim wksBtn As Button Set cnt = Application.CommandBars.FindControl(ID:=30007) Set btn = cnt.Controls("Meldung A") Set wksBtn = ActiveSheet.Buttons.Add(_ Left:=Range("B3").Left, _ Top:=Range("B3").Top, _ Width:=Range("C3").Left + Range("C3").Width - Range("B3").Left, _ Height:=Range("B3:B4").Height) With wksBtn .Caption = btn.Caption .OnAction = btn.OnAction .Name = "Btn1" End With Set btn = cnt.Controls("Meldung B") Set wksBtn = ActiveSheet.Buttons.Add(_ Left:=Range("B6").Left, _ Top:=Range("B6").Top, _ Width:=Range("C6").Left + Range("C6").Width - Range("B6").Left, _ Height:=Range("B6:B7").Height) With wksBtn .Caption = btn.Caption .OnAction = btn.OnAction .Name = "Btn2" End With Set wksBtn = ActiveSheet.Buttons.Add(_ Left:=Range("B9").Left, _ Top:=Range("B9").Top, _ Width:=Range("C9").Left + Range("C9").Width - Range("B9").Left, _ Height:=Range("B9:B10").Height) With wksBtn .Caption = "Buttons zum Extra-Menü" .OnAction = "ConvertToCommandButton" .Name = "Btn3" End With cnt.Controls("Meldung A").Delete cnt.Controls("Meldung B").Delete ActiveSheet.Buttons("Btn4").Delete End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Verknüpfungsdialog aus VBA aufrufen

Problem:	Der Verknüpfungsdialog soll aufgerufen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallLinks() Dim oCntr As CommandBarControl Set oCntr = Application.CommandBars.FindControl(ID:=759) If oCntr.Enabled Then oCntr.Execute End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufruf des Web-Assistenten über VBA

Problem:	Über eine Schaltfläche soll der Web-Assistenten-Dialog aufgerufen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub WebAssistent() Dim cntr As CommandBarControl Set cntr = Application.CommandBars.FindControl(ID:=3823) cntr.Execute End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neuer Menüpunkt zum Einfügen eines benutzerdefinierten Kommentars

Problem:	Im EINFÜGEN-Menü soll beim Öffnen der Arbeitsmappe ein neuer Menüpunkt Mein Kommentar eingefügt werden, der das Erstellen eines Kommentars ohne vorgegebenen Anwendername an einer Position rechts der aktiven Zelle ermöglicht.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Dim oPopUp As CommandBarPopup Set oPopUp = Application.CommandBars("Worksheet Menu Bar") _ .FindControl(ID:=30005) On Error Resume Next oPopUp.Controls("Mein Kommentar").Delete On Error GoTo 0 Application.DisplayCommentIndicator = _ ThisWorkbook.Worksheets(1).Range("IV1") ThisWorkbook.Worksheets(1).Range("IV1").ClearContents End Sub Private Sub Workbook_Open() Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Set oPopUp = Application.CommandBars("Worksheet Menu Bar") _ .FindControl(ID:=30005) On Error Resume Next oPopUp.Controls("Mein Kommentar").Delete On Error GoTo 0 Set oBtn = oPopUp.Controls.Add(before:=11) With oBtn .Caption = "Mein Kommentar" .OnAction = "NewComment" .FaceId = 2056 .Style = msoButtonIconAndCaption End With ThisWorkbook.Worksheets(1).Range("IV1") = _ Application.DisplayCommentIndicator Application.DisplayCommentIndicator = xlCommentIndicatorOnly End Sub ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim cmt As Comment For Each cmt In Comments cmt.Visible = False Next cmt End Sub StandardModule: basMain Sub NewComment() Dim oBtn As CommandBarButton Dim shp As Shape Dim lLeft As Long lLeft = ActiveCell.Left - 50 If lLeft < 0 Then lLeft = 0 Set oBtn = _ Application.CommandBars.FindControl(ID:=1589) oBtn.Execute Set oBtn = _ Application.CommandBars.FindControl(ID:=1593) oBtn.Execute 'On Error Resume Next Set shp = ActiveCell.Comment.Shape shp.Left = lLeft shp.Top = ActiveCell.Top shp.Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblattmenüleiste eine Rückgängig-Schaltfläche hinzufügen

Problem:	Wie kann ich der Arbeitsblattmenüleiste eine Rückgängig-Schaltfläche hinzufügen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Speichern-Schaltfläche aktivieren / deaktivieren

Problem:	Die Speichern-Schaltfläche in der Standard-Symbolleiste soll aktiviert bzw. deaktiviert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub SpeicherAusEin() Dim oBtn As CommandBarButton Set oBtn = Application.CommandBars("Standard").FindControl(ID:=3) oBtn.Enabled = Not oBtn.Enabled End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neuer Menüpunkt mit Kommentar-einfügen-Befehl

Problem: Beim Öffnen der Arbeitsmappe soll dem "Einfügen"-Menü ein neuer Menüpunkt "XKommentar" hinzugefügt werden. Bei Anklicken des Menüpunktes wird ein neuer Kommentar mit vorgegebener Schriftformatierung und selektiertem Anwendernamen erstellt.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Dim oPopUp As CommandBarPopup
    Set oPopUp = Application.CommandBars("Worksheet Menu Bar") _
        .FindControl(ID:=30005)
    On Error GoTo ERRORHANDLER
    oPopUp.Controls("XKommentar").Delete
ERRORHANDLER:
End Sub

Private Sub Workbook_Open()
    Dim oPopUpA As CommandBarPopup
    Dim oPopUpB As CommandBarControl
    Dim oBtn As CommandBarButton
    Set oPopUpA = Application.CommandBars("Worksheet Menu Bar") _
        .FindControl(ID:=30005)
    For Each oPopUpB In oPopUpA.Controls
        If InStr(WorksheetFunction _
            .Substitute(oPopUpB.Caption, "&", ""), _
            "Kommentar") Then Exit For
    Next oPopUpB
    On Error Resume Next
    oPopUpA.Controls("XKommentar").Delete
    On Error GoTo 0
    Set oBtn = oPopUpA.Controls.Add(before:=oPopUpB.Index + 1)
    With oBtn
        .Caption = "XKommentar"
        .OnAction = "SetComment"
        .FaceId = 1591
        .Style = msoButtonIconAndCaption
    End With
End Sub

StandardModule: basMain

Sub SetComment()
    Dim cmt As Comment
    Set cmt = ActiveCell.AddComment
    With cmt.Shape
        .Height = 80
        .Width = 120
        With .TextFrame
            With .Characters.Font
                .Size = 12
                .Bold = False
            End With
        End With
        .Visible = True
        SendKeys Application.UserName & ":+{home}"
        .Select
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Menüpunkt "Zellen einfügen..." im Kontextmenü deaktivieren

Problem:	In jeder aktiven Arbeitsmappe soll im Zell-Kontextmenü der Menüpunkt "Zelle einfügen..." deaktiviert sein.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) On Error Resume Next Application.CommandBars("Cell") _ .Controls("Zellen einfügen...").Enabled = True On Error GoTo 0 End Sub Private Sub Workbook_Open() Set app.xlApp = Application End Sub ClassModule: clsApp Public WithEvents xlApp As Application Private Sub xlApp_WindowActivate(ByVal Wb As Workbook, ByVal Wn As Window) On Error Resume Next Application.CommandBars("Cell") _ .Controls("Zellen einfügen...").Enabled = False On Error GoTo 0 End Sub StandardModule: basMain Public app As New clsApp</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Einfügen eines Menüpunktes zum Erstellen eines Kommentars.

Problem:	Im EINFÜGEN-Menü soll ein neuer Menüpunkt "Mein Kommentar" eingefügt werden, der das Erstellen eines Kommentars ohne vorgegebenen Anwendername ermöglicht.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Dim oPopUp As CommandBarPopup Set oPopUp = Application.CommandBars("Worksheet Menu Bar") _ .FindControl(ID:=30005) On Error Resume Next oPopUp.Controls("Mein Kommentar").Delete On Error GoTo 0 Application.DisplayCommentIndicator = _ ThisWorkbook.Worksheets(1).Range("IV1") ThisWorkbook.Worksheets(1).Range("IV1").ClearContents End Sub Private Sub Workbook_Open() Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Set oPopUp = Application.CommandBars("Worksheet Menu Bar") _ .FindControl(ID:=30005) On Error Resume Next oPopUp.Controls("Mein Kommentar").Delete On Error GoTo 0 Set oBtn = oPopUp.Controls.Add(before:=11) With oBtn .Caption = "Mein Kommentar" .OnAction = "NewComment" .FaceId = 2056 .Style = msoButtonIconAndCaption End With ThisWorkbook.Worksheets(1).Range("IV1") = _ Application.DisplayCommentIndicator Application.DisplayCommentIndicator = xlCommentIndicatorOnly End Sub ClassModule: Tabelle1 Private Sub Worksheet_SelectionChange(ByVal Target As Range) Dim cmt As Comment For Each cmt In Comments cmt.Visible = False Next cmt End Sub StandardModule: basMain Sub NewComment() Dim oBtn As CommandBarButton Set oBtn = _ Application.CommandBars.FindControl(ID:=1589) oBtn.Execute Set oBtn = _ Application.CommandBars.FindControl(ID:=1593) oBtn.Execute On Error Resume Next ActiveCell.Comment.Shape.Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Menüpunkt bei Aktivieren einer Mappe erstellen

Problem: Beim Aktivieren einer Arbeitsmappe mit dem String "KN." im Namen, also beispielsweise "WKN.xls", soll im Arbeitsblattmenü ein neuer Menüpunkt "KN-Befehl" erscheinen, beim Deaktivieren der Mappe wieder gelöscht werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    Application.CommandBars("Worksheet Menu Bar").Controls("KN-Befehl").Delete
    On Error GoTo 0
End Sub

Private Sub Workbook_Open()
    Set app.xlApp = Application
End Sub

ClassModule: clsApp

Public WithEvents xlApp As Application

Private Sub xlApp_WorkbookActivate(ByVal Wb As Workbook)
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Set oBar = Application.CommandBars("Worksheet Menu Bar")
    On Error Resume Next
    oBar.Controls("KN-Befehl").Delete
    On Error GoTo 0
    If InStr(ActiveWorkbook.Name, "KN.") Then
        Set oBtn = oBar.Controls.Add
        With oBtn
            .Caption = "KN-Befehl"
            .OnAction = ThisWorkbook.Name & "!KNBefehl"
            .Style = msoButtonCaption
        End With
    End If
End Sub

StandardModule: basMain

Public app As New clsApp

Sub KNBefehl()
    MsgBox "Ich bin der WKN-Befehl!"
End Sub
```



Benutzerdefiniere Funktion einer Schaltfläche zuweisen

Problem:	Wie kann ich eine benutzerdefinierte Funktion einer Symbolleistenschaltfläche zuweisen? Der Ablauf soll ähnlich sein, wie bei dem Excel-Feature AutoSumme.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub Private Sub Workbook_Open() Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Call CmdDelete With Application.CommandBars("Worksheet Menu Bar") Set oPopUp = .Controls.Add(_ Type:=msoControlPopup, _ before:=.Controls.Count, _ temporary:=True) End With oPopUp.Caption = "&Kubatur" Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Formel Eingabe" .OnAction = "Cbm_Active_Formel" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Wert Eingabe" .OnAction = "Cbm_Active_Wert" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Formel Auswahl" .OnAction = "Cbm_Formel_Select" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Wert Auswahl" .OnAction = "Cbm_Wert_Select" .Style = msoButtonCaption End With End Sub StandardModule: basMain Function Kubatur(rng As Range) As Double Kubatur = rng(1) * rng(2) * rng(3) End Function Sub Cbm_Active_Formel() Dim iLen As Integer, sRng As String With ActiveCell If IsEmpty(.Offset(0, -1)) Or .Column < 4 Then .Formula = "=kubatur()" SendKeys "{F2}{left}" Else sRng = Range(Cells(.Row, .Column - 3), _ Cells(.Row, .Column - 1)).Address iLen = Len(sRng) .Formula = "=kubatur(" & sRng & ")" SendKeys "{F2}{left}+{left " & iLen & "}" End If End With End Sub Sub Cbm_Active_Wert() Dim iLen As Integer, sRng As String With ActiveCell If .Column < 4 Then Beep MsgBox "Die Funktion kann nur ab Spalte D genutzt werden!" Else ActiveCell.Value = Kubatur(Range(ActiveCell.Offset(0, -3), _ ActiveCell.Offset(0, -1))) End If End With End Sub Sub Cbm_Formel_Select() Dim rng As Range Set rng = Application.InputBox("Bereich:", Type:=8) If rng.Cells.Count <> 3 Then Beep MsgBox "Sie benötigen Länge, Breite und Höhe -" & _ vbLf & "also bitte 3 Zellen auswählen!" Exit Sub End If ActiveCell.Formula = "=kubatur(" & rng.Address & ")" End Sub Sub Cbm_Wert_Select() Dim rng As Range Set rng = Application.InputBox("Bereich:", Type:=8) If rng.Cells.Count <> 3 Then Beep MsgBox "Sie benötigen Länge, Breite und Höhe -" & _ vbLf & "also bitte 3 Zellen auswählen!" Exit Sub End If ActiveCell.Value = Kubatur(rng) End Sub Sub CmdDelete() On Error GoTo ERRORHANDLER Application.CommandBars("Worksheet Menu Bar") _ .Controls("&Kubatur").Delete ERRORHANDLER: End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Menüpunkt hinzufügen und bei Aufruf zeitgesteuert UserForm anzeigen

Problem: Wie kann ich beim Öffnen der Arbeitsmappe dem Arbeitsblattmenü einen weiteren Menüpunkt hinzufügen und beim Schliessen wieder löschen? Der Menüpunkt soll für 2 Sekunden den Anwendernamen anzeigen.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Application.CommandBars("Worksheet Menu Bar") _
        .Controls("Dialog aufrufen").Delete
End Sub

Private Sub Workbook_Open()
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Set oBar = Application.CommandBars("Worksheet Menu Bar")
    On Error Resume Next
    oBar.Controls("Dialog aufrufen").Delete
    On Error GoTo 0
    Set oBtn = oBar.Controls.Add
    With oBtn
        .Caption = "Dialog aufrufen"
        .OnAction = "CallForm"
        .Style = msoButtonCaption
    End With
End Sub

ClassModule: frmSplash

Private Sub UserForm_Activate()
    Application.Wait Now + TimeSerial(0, 0, 2)
    Unload Me
End Sub

Private Sub UserForm_Initialize()
    lblUser.Caption = Application.UserName
End Sub

StandardModule: basMain

Sub CallForm()
    frmSplash.Show
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Letztes Zeichen der Werte einer Zellauswahl hoch-/tiefstellen

Problem:	Wie kann ich dem Formatmenü Menüpunkte zum Hoch-/Tiefstellen der jeweils letzten Zeichen der Werte einer Zellauswahl hinzufügen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub Private Sub Workbook_Open() Dim oBar As CommandBar Dim oPopUpA As CommandBarPopup Dim oPopUpB As CommandBarPopup Dim oBtn As CommandBarButton Call CmdDelete Set oBar = Application.CommandBars("Worksheet Menu Bar") Set oPopUpA = oBar.Controls("Format") Set oPopUpB = oPopUpA.Controls.Add(Type:=msoControlPopup) With oPopUpB .Caption = "&HochTief" .BeginGroup = True End With Set oBtn = oPopUpB.Controls.Add With oBtn .Caption = "&Hochstellen" .OnAction = "HochTiefStellen" .Style = msoButtonCaption End With Set oBtn = oPopUpB.Controls.Add With oBtn .Caption = "&Tiefstellen" .OnAction = "HochTiefStellen" .Style = msoButtonCaption End With Set oBtn = oPopUpB.Controls.Add With oBtn .Caption = "&Normal" .OnAction = "HochTiefStellen" .Style = msoButtonCaption End With End Sub StandardModule: modMain Sub HochTiefStellen() Dim rng As Range For Each rng In Selection With rng.Characters(Start:=Len(rng.Text), Length:=1).Font Select Case Application.Caller(1) Case 1 .Superscript = True Case 2 .Subscript = True Case 3 .Superscript = False .Subscript = False End Select End With Next rng End Sub Sub CmdDelete() On Error GoTo ERRORHANDLER Application.CommandBars("Format").Controls("HochTief").Delete ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufrufende Menüleistenschaltfläche ermitteln

Problem:	Wie kann ich feststellen, welche Schaltfläche aus einer benutzerdefinierten Menüleiste ein Makro aufgerufen hat?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub Private Sub Workbook_Open() Dim oBar As CommandBar Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Dim iMonths As Integer, iDays As Integer Call CmdDelete Set oBar = Application.CommandBars.Add(Name:="J" & _ Year(Date), MenuBar:=True) For iMonths = 1 To 12 Set oPopUp = oBar.Controls.Add(Type:=msoControlPopup) oPopUp.Caption = Format(DateSerial(Year(Date), _ iMonths, 1), "mmm") For iDays = 1 To Day(DateSerial(Year(Date), iMonths + 1, 0)) Set oBtn = oPopUp.Controls.Add(Type:=msoControlButton) With oBtn .Caption = iDays .OnAction = "Aufruf" .Style = msoButtonCaption End With Next iDays Next iMonths Set oBtn = oBar.Controls.Add With oBtn .Caption = "Schließen" .OnAction = "Beenden" .Style = msoButtonCaption End With Application.CommandBars("J" & Year(Date)).Visible = True Application.CommandBars("J" & Year(Date)).Enabled = True End Sub StandardModule: modMain Sub Aufruf() MsgBox Format(DateSerial(Year(Date), Application.Caller(2), _ Application.Caller(1)), "dd. mmm yyyy") End Sub Sub CmdDelete() On Error GoTo ERRORHANDLER With Application .CommandBars("J" & Year(Date)).Delete .CommandBars("Worksheets Menu Bar").Visible = True .CommandBars("Worksheets Menu Bar").Enabled = True End With ERRORHANDLER: End Sub Sub Beenden() ThisWorkbook.Close End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen einer Arbeitsmappe neues Symbol in Formatsymbolleiste

Problem: Wie kann ich der Format-Symbolleiste beim Öffnen einer bestimmten Arbeitsmappe eine einem Makrzu zugewiesene neue Schaltfläche hinzufügen und diese beim Schließen der Arbeitsmappe wieder löschen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call BtnDelete
End Sub

Private Sub Workbook_Open()
    Dim oBtn As CommandBarButton
    Call BtnDelete
    Set oBtn = Application.CommandBars("Formatting").Controls.Add
    With oBtn
        .Caption = "Befehl"
        .FaceId = 2
        .OnAction = "MeinBefehl"
        .Visible = True
    End With
End Sub

StandardModule: basMain

Sub BtnDelete()
    On Error GoTo ERRORHANDLER
    Application.CommandBars("Formatting") _
        .Controls("Befehle").Delete
ERRORHANDLER:
End Sub

Sub MeinBefehl()
    MsgBox "Ich bin der Befehl!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Schaltflächen-Id-Nummer finden und Schaltfläche einfügen

Problem:	Wie kann ich die Id-Nummer der Zielwertsuche-Schaltfläche herausfinden und diese Schaltfläche in eine eigene Symbolleiste integrieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call Loeschen End Sub StandardModule: basMain Sub ZielWertSuche() Dim oBar As CommandBar Dim oCntr As CommandBarControl Call Loeschen Set oBar = CommandBars.Add("ZielWert") Set oCntr = oBar.Controls.Add(ID:=856) oBar.Visible = True End Sub Sub Loeschen() On Error GoTo ERRORHANDLER CommandBars("ZielWert").Delete ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Menüpunkt **SPEICHERN** deaktivieren

Problem:	Wie kann ich im Menü DATEI den Menüpunkt "Speichern" deaktivieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) On Error GoTo ERRORHANDLER Application.CommandBars("Worksheet Menu Bar") _ .Controls("Datei").Controls("Speichern").Enabled = True ERRORHANDLER: End Sub StandardModule: basMain Sub DeactivateSave() With Application.CommandBars("Worksheet Menu Bar") _ .Controls("Datei").Controls("Speichern") .Enabled = Not .Enabled End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Menüpunkte zum Einfügen und Löschen von Zeilen hinzufügen

Problem:	Wie kann ich das Einfügen und Löschen von Zeilen zählen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NewBtns() Dim oBtn As CommandBarButton Set oBtn = Application.CommandBars("Cell").Controls.Add On Error Resume Next With oBtn .Caption = "Zeilen einfügen" .OnAction = "Einfuegen" .BeginGroup = True .Style = msoButtonCaption End With Set oBtn = Application.CommandBars("Cell").Controls.Add With oBtn .Caption = "Zeilen löschen" .OnAction = "Loeschen" .Style = msoButtonCaption End With On Error GoTo 0 End Sub Sub ResetBtns() On Error Resume Next With Application.CommandBars("Cell") .Controls("Zeilen einfügen").Delete .Controls("Zeilen löschen").Delete End With On Error GoTo 0 End Sub Sub Einfuegen() Selection.EntireRow.Insert End Sub Sub Loeschen() Selection.EntireRow.Delete End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Menue-Punkt mit Haekchen versehen

Problem:	Wie kann ich in XL8 einen Menüpunkt mit einem Häkchen versehen. Die "Checked"-Eigenschaft aus XL5/7 kann ich nicht finden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call Zurueck End Sub StandardModule: basMain Sub MenuPunkt() Dim oBar As CommandBar Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Set oBar = Application.CommandBars("Worksheet Menu Bar") Call Zurueck Set oPopUp = oBar.Controls.Add(_ Type:=msoControlPopup, Temporary:=False) oPopUp.Caption = "Demo" Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Import" .TooltipText = "Import" .OnAction = "Import" .Style = msoButtonCaption .State = msoButtonDown End With End Sub Sub Zurueck() On Error Resume Next Application.CommandBars("Worksheet Menu Bar") _ .Controls("Demo").Delete On Error GoTo 0 End Sub Sub Import() MsgBox "Import" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benutzerdefiniertes Symbol in neue Symbolleiste einfügen

Problem: Wie kann ich in eine neue Symbolleiste ein benutzerdefiniertes Symbol einfügen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    Application.CommandBars("Help").Controls( _
        "DejaNews-Abfrage").Delete
    On Error GoTo 0
End Sub
```

Bitte anpassen

```
Private Sub Workbook_Open()
    Dim oBtn As CommandBarButton
    On Error GoTo ERRORHANDLER
    Set oBtn = Application.CommandBars("Help").Controls.Add( _
        Type:=msoControlButton)
    With oBtn
        .BeginGroup = True
        .OnAction = "Start"
        .Caption = "DejaNews-Abfrage"
        Application.CommandBars("hwh-standard").Controls(11).CopyFace
        .PasteFace
    End With
ERRORHANDLER:
End Sub
```



Menüpunkte sortieren

Problem:	Wie kann ich in XL8 einzelne Menüpunkte eines benutzerdefinierten Menüs alphabetisch anordnen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call MenuLoeschen End Sub StandardModule: basMain Sub NeuesMenu() Dim oBar As CommandBar Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Set oBar = Application.CommandBars("Worksheet Menu Bar") Set oPopUp = oBar.Controls.Add(msoControlPopup, temporary:=True) Call MenuLoeschen oPopUp.Caption = "MeinMenu" Set oBtn = oPopUp.Controls.Add oBtn.Caption = "B-Befehl" Set oBtn = oPopUp.Controls.Add oBtn.Caption = "A-Befehl" Set oBtn = oPopUp.Controls.Add oBtn.Caption = "C-Befehl" End Sub Sub MenuSortieren() Dim arr() As String Dim oBar As CommandBar Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Dim iCountA As Integer, iCountB As Integer Dim sTxt As String Set oBar = Application.CommandBars("Worksheet Menu Bar") Set oPopUp = oBar.Controls("MeinMenu") For Each oBtn In oPopUp.Controls iCountA = iCountA + 1 ReDim Preserve arr(iCountA) arr(iCountA) = oBtn.Caption Next oBtn For iCountA = 1 To UBound(arr) For iCountB = iCountA + 1 To UBound(arr) If arr(iCountA) > arr(iCountB) Then sTxt = arr(iCountA) arr(iCountA) = arr(iCountB) arr(iCountB) = sTxt End If Next iCountB Next iCountA For Each oBtn In oPopUp.Controls oBtn.Delete Next oBtn For iCountA = 1 To UBound(arr) Set oBtn = oPopUp.Controls.Add oBtn.Caption = arr(iCountA) Next iCountA End Sub Sub MenuLoeschen() On Error Resume Next Application.CommandBars(_ "Worksheet Menu Bar").Controls("MeinMenu").Delete On Error GoTo 0 End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neue Befehle in Menüleiste

Problem: Wie kann ich in XL8 beim Laden einer Arbeitsmappe der Excel-Menüleiste zwei Befehle hinzufügen, die beim Schließen der Mappe wieder gelöscht werden sollen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_Open()  
    Dim oBar As CommandBar  
    Dim oBtn As CommandBarButton  
    Set oBar = Application.CommandBars("Worksheet Menu Bar")  
    On Error Resume Next  
    oBar.Controls("Test1").Delete  
    oBar.Controls("Test2").Delete  
    On Error GoTo 0  
    Set oBtn = oBar.Controls.Add( _  
        Type:=msoControlButton, _  
        before:=CB.Controls.Count, _  
        ID:=59, _  
        temporary:=True)  
    With oBtn  
        .Caption = "Test1"  
        .OnAction = "Makro1"  
    End With  
    Set oBtn = oBar.Controls.Add( _  
        Type:=msoControlButton, _  
        before:=CB.Controls.Count, _  
        ID:=276, _  
        temporary:=True)  
    With oBtn  
        .Caption = "Test2"  
        .OnAction = "Makro2"  
    End With  
End Sub  
  
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
    On Error Resume Next  
    With Application.CommandBars("Worksheet Menu Bar")  
        .Controls("Test1").Delete  
        .Controls("Test2").Delete  
    End With  
    On Error GoTo 0  
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dem Extra-Menü einen Menüpunkt hinzufügen

Problem: Wie kann ich beim Öffnen der Arbeitsmappe in XL8 dem Extra-Menü einen Menüpunkt hinzufügen und diesen beim Schließen wieder entfernen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Open()
    Dim oBar As CommandBar
    Dim oPopUp As CommandBarControl
    Dim oBtn As CommandBarButton
    Set oBar = Application.CommandBars("Worksheet Menu Bar")
    Set oPopUp = oBar.FindControl(ID:=3007)
    On Error Resume Next
    oPopUp.Controls("Befehl").Delete
    On Error GoTo 0
    Set oBtn = oPopUp.Controls.Add
    With oBtn
        .Caption = "Test"
        .OnAction = "Befehl"
        .FaceId = 36
        .Style = msoButtonIconAndCaption
    End With
End Sub

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Dim oBar As CommandBar
    Dim oPopUp As CommandBarControl
    Dim oBtn As CommandBarButton
    Set oBar = Application.CommandBars("Worksheet Menu Bar")
    Set oPopUp = oBar.FindControl(ID:=3007)
    On Error Resume Next
    oPopUp.Controls("Befehl").Delete
    On Error GoTo 0
End Sub

StandardModule: basMain

Sub Befehl()
    MsgBox "Ich bin der Befehl!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Symbolleistenschaltfläche hinzufügen

Problem: Wie kann ich der Standardsymbolleiste beim Laden dieser Arbeitsmappe ein Schaltflächensymbol hinzufügen und beim Schließen der Arbeitsmappe wieder entfernen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    Application.CommandBars("Standard") _
        .Controls("MyButton").Delete
    On Error GoTo 0
End Sub

Private Sub Workbook_Open()
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Set oBar = Application.CommandBars("Standard")
    On Error Resume Next
    oBar.Controls("MyButton").Delete
    On Error GoTo 0
    Set oBtn = oBar.Controls.Add
    With oBtn
        .Caption = "MyButton"
        .Style = msoButtonIcon
        .FaceId = 361
        .OnAction = "Meldung"
    End With
End Sub

StandardModule: Modul1

Sub Meldung()
    MsgBox prompt:="Hallo!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Submenüeinträge

Problem: Wie kann ich einem Menü ein weiteres Submenü hinzufügen? Die Aufschriften sollen aus einem Tabellenblatt übernommen werden. Die Makros zu den Menüpunkten haben die gleichen Namen wie die Aufschriften.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub MenuAdd()
    Dim oPopUpA As CommandBarPopup
    Dim oPopUpB As CommandBarPopup
    Dim oBtn As CommandBarButton
    Dim rng As Range
    Dim iCounter As Integer
    Set oPopUpA = Application.CommandBars( _
        "Worksheet Menu Bar").FindControl(Id:=30007)
    On Error Resume Next
    oPopUpA.Controls("Projekt-Liste").Delete
    On Error GoTo 0
    On Error GoTo ERRORHANDLER
    If ActiveSheet.Buttons(1).Caption = "Menü anlegen" Then
        Set oPopUpB = oPopUpA.Controls.Add(msoControlPopup)
        oPopUpB.Caption = "Projekt-Liste"
        For iCounter = 1 To 10
            Set oBtn = oPopUpB.Controls.Add
            With oBtn
                .Caption = Cells(iCounter, 1).Value
                .OnAction = Cells(iCounter, 1).Value
                .Style = msoButtonCaption
            End With
        Next iCounter
        ActiveSheet.Buttons(1).Caption = "Menü löschen"
    Else
        ActiveSheet.Buttons(1).Caption = "Menü anlegen"
        On Error Resume Next
        oPopUpA.Controls("Projekt-Liste").Delete
        On Error GoTo 0
    End If
    Exit Sub
ERRORHANDLER:
    MsgBox prompt:="Das Menü kann nicht erstellt werden!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Beim Öffnen der Arbeitsmappe Menü-ComboBoxes anlegen

Problem: Beim Öffnen der Arbeitsmappe sollen in der Arbeitsblatt-Menüleiste zwei ComboBoxes mit den Inhalten der Spalten A:B angelegt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    With Application.CommandBars("Worksheet Menu Bar")
        .Controls("Gruppe 1").Delete
        .Controls("Gruppe 2").Delete
    End With
End Sub

Private Sub Workbook_Open()
    Dim oBar As CommandBar
    Dim oCbo As CommandBarComboBox
    Dim iCounter As Integer, iCol As Integer
    Set oBar = Application.CommandBars("Worksheet Menu Bar")
    On Error Resume Next
    oBar.Controls("Gruppe 1").Delete
    oBar.Controls("Gruppe 2").Delete
    On Error GoTo 0
    With oCbo
        For iCol = 1 To 2
            Set oCbo = oBar.Controls.Add(msoControlComboBox)
            oCbo.Caption = "Gruppe " & iCol
            For iCounter = 2 To 11
                oCbo.AddItem Cells(iCounter, iCol).Value
                oCbo.ToolTipText = Cells(1, iCol).Value
            Next iCounter
            oCbo.ListIndex = 1
        Next iCol
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Geöffnete Arbeitsmappen in Menü-ComboBox listen

Problem: Die aktuell geöffneten Arbeitsmappen sollen in einer CommandBar-ComboBox in der Arbeitsblattmenüleiste zum Zweck der Auswahl gelistet werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call CmdDelete
End Sub

Private Sub Workbook_Open()
    Call NeuEinlesen
End Sub

Private Sub Workbook_WindowActivate(ByVal Wn As Window)
    Call NeuEinlesen
End Sub

StandardModule: basMain

Sub NeuEinlesen()
    Dim oCbo As CommandBarComboBox
    Dim wkb As Workbook
    Call CmdDelete
    Set oCbo = Application.CommandBars("Worksheet Menu Bar") _
        .Controls.Add(msoControlComboBox)
    With oCbo
        .Caption = "Mappen"
        For Each wkb In Workbooks
            .AddItem wkb.Name
            .OnAction = "BlattAuswahl"
        Next wkb
        .Text = ActiveWindow.Caption
    End With
End Sub

Sub BlattAuswahl()
    Application.EnableEvents = False
    Windows(Application.CommandBars(1) _
        .Controls("Mappen").Text).Activate
    Application.EnableEvents = True
End Sub

Sub CmdDelete()
    On Error GoTo ERRORHANDLER
    Application.CommandBars("Worksheet Menu Bar") _
        .Controls("Mappen").Delete
ERRORHANDLER:
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Symbolleiste mit ComboBox erstellen

Problem:	Wie kann ich beim Öffnen einer Arbeitsmappe eine Symbolleiste mit einer CommandBarComboBox erstellen und mit den Werten aus Spalte A einer Tabelle bestücken? Es handelt sich bei Dateinamen. Bei Auswahl einer Datei in der ComboBox soll diese geöffnet werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub Private Sub Workbook_Open() Dim wks As Worksheet Dim oBar As CommandBar Dim oCbo As CommandBarComboBox Dim iCounter As Integer Set wks = ThisWorkbook.Worksheets("Tabelle1") Call CmdDelete Set oBar = Application.CommandBars.Add("Dateien", msoBarTop) Set oCbo = oBar.Controls.Add(msoControlComboBox) iCounter = 1 With oCbo Do Until IsEmpty(wks.Cells(iCounter, 1)) .AddItem wks.Cells(iCounter, 1).Value iCounter = iCounter + 1 Loop .OnAction = "DateienLaden" .ListIndex = 1 End With oBar.Visible = True End Sub StandardModule: basMain Sub DateienLaden() Dim wkb As Workbook Dim sFile With CommandBars("Dateien").Controls(1) sFile = .List(.ListIndex) End With On Error Resume Next Set wkb = Workbooks(sFile) If Err > 0 Or wkb Is Nothing Then Err.Clear On Error GoTo 0 Workbooks.Open (sFile) End If End Sub Sub CmdDelete() On Error GoTo ERRORHANDLER Application.CommandBars("Dateien").Delete ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox für Monatsblattauswahl in Menüleiste einfügen

Problem:	Wie kann ich in die Arbeitsblattmenüleiste eine ComboBox mit Monatsnamen zur Auswahl von Monatsblättern einfügen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub Private Sub Workbook_Open() Dim oCbo As CommandBarComboBox Dim wks As Worksheet Dim iCounter As Integer Call CmdDelete With Application.CommandBars("Worksheet Menu Bar") Set oCbo = .Controls.Add(_ Type:=msoControlComboBox, before:=.Controls.Count) End With With oCbo .Caption = "Monate" .OnAction = "BlattAuswahl" For iCounter = 0 To 11 .AddItem Format(DateSerial(1, iCounter + 1, 1), "mmmm") Next iCounter .ListIndex = 1 End With End Sub StandardModule: basMain Sub JahrAnlegen() Dim iMonat As Integer For iMonat = 1 To 12 Worksheets.Add after:=Worksheets(Worksheets.Count) ActiveSheet.Name = Format(DateSerial(1, iMonat, 1), "mmmm") ActiveSheet.Visible = xlSheetVeryHidden Next iMonat End Sub Sub Blattauswahl() Call Verbergen With Worksheets(CommandBars("Worksheet Menu Bar") _ .Controls("Monate").ListIndex + 2) .Visible = True .Select End With End Sub Sub Anzeigen() Dim wks As Worksheet For Each wks In ThisWorkbook.Worksheets wks.Visible = True Next wks End Sub Sub Verbergen() Dim iCounter As Integer For iCounter = 3 To Worksheets.Count Worksheets(iCounter).Visible = xlVeryHidden Next iCounter End Sub Sub CmdDelete() On Error GoTo ERRORHANDLER Application.CommandBars("Worksheet Menu Bar") _ .Controls("Monate").Delete ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert in einer CommandBarComboBox verändern

Problem: Wie kann ich eine Symbolleiste mit einer CommandBarComboBox hinzufügen und dieser als Startwert das aktuelle Datum zuordnen? Der Symbolleiste soll eine weitere Schaltfläche hinzugefügt werden, die auf Mausklick das Datum in der CommandBarComboBox um 1 erhöht.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call Loeschen
End Sub

StandardModule: basMain

Sub DatumInControl()
    Dim oBar As CommandBar
    Dim oCbo As CommandBarComboBox
    Dim oBtn As CommandBarButton
    Set oBar = CommandBars.Add( _
        Name:="MeineSymbolleiste", _
        Position:=msoBarTop, _
        MenuBar:=False, _
        temporary:=True)
    oBar.Visible = True
    Set oCbo = oBar.Controls.Add(Type:=msoControlComboBox)
    With oCbo
        .Caption = "Datum"
        .AddItem Date
    End With
    Set oBtn = oBar.Controls.Add
    With oBtn
        .FaceId = 36
        .OnAction = "Erhoehen"
        oCbo.ListIndex = 1
    End With
End Sub

Sub Erhoehen()
    With Application.CommandBars("MeineSymbolleiste").Controls("Datum")
        .List(1) = Format(DateValue(.List(1)) + 1, "dd.mm.yy")
    End With
End Sub

Sub Loeschen()
    On Error GoTo ERRORHANDLER
    Application.CommandBars("MeineSymbolleiste").Delete
ERRORHANDLER:
End Sub
```


Beispiele zum Handling des Zell-Kontextmenüs

Problem: In 10 Tabellenblättern werden Beispiele geliefert, wie das Zell-Kontextmenü eingesetzt werden kann.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn Schaltflächen zu.

```
VBA-Code:

ClassModule: Tabelle1

Private Sub Worksheet_Activate()
    Dim oBtn As CommandBarButton
    With Application.CommandBars("Cell")
        Set oBtn = .FindControl(ID:=21)
        oBtn.Enabled = False
    End With
End Sub

Private Sub Worksheet_Deactivate()
    Application.CommandBars("Cell").Reset
End Sub

StandardModule: Modul1

Sub ReadID()
    Dim sMsg As String
    sMsg = "Erstes Element im Zell-Kontextmenü:" & vbCrLf
    With Application.CommandBars("Cell").Controls(1)
        sMsg = sMsg & "Caption: " & .Caption & vbCrLf
        sMsg = sMsg & "ID: " & .ID
    End With
    MsgBox sMsg
End Sub

Sub ShowPosition()
    Dim iLeft As Integer, iTop As Integer
    iLeft = Application.InputBox("Linke Position:", , , 200, Type:=1)
    iTop = Application.InputBox("Obere Position:", , , 200, Type:=1)
    Application.CommandBars("Cell").ShowPopup iLeft, iTop
End Sub

Sub MonthMessage()
    MsgBox "Aufruf kam von " & _
        Application.CommandBars("Months") _
        .Controls(Application.Caller(1)).Caption
End Sub

Sub SetFilter()
    Dim iRow As Integer
    Application.ScreenUpdating = False
    Rows.Hidden = False
    iRow = 1
    Do Until IsEmpty(Cells(iRow, 1))
        If Weekday(Cells(iRow, 1).Value) <> Application.Caller(1) - 2 Then
            Rows(iRow).Hidden = True
        End If
        iRow = iRow + 1
    Loop
    Application.ScreenUpdating = True
End Sub

Sub ResetFilter()
    Rows.Hidden = False
End Sub

Sub SpaltenAus()
    Selection.EntireColumn.Hidden = True
End Sub

Sub SpaltenEin()
    Selection.EntireColumn.Hidden = False
End Sub

Sub InsertValue()
    ActiveCell.Value = Application.CommandBars("RowValues") _
        .Controls(Application.Caller(1)).Caption
End Sub

Sub SelectSheet()
    Worksheets(Application.CommandBars("SheetSelect") _
        .Controls(Application.Caller(1)).Caption).Select
End Sub

Sub DialogShortCutMenus()
    frmKontext.Show
End Sub

Sub ListShortCutMenus()
    Dim oBar As CommandBar
    Dim oCntr As CommandBarControl
    Dim iRow As Integer
    Application.ScreenUpdating = False
    Workbooks.Add
    Range("A1").Value = "Name"
    Range("B1").Value = "Lokaler Name"
    Range("C1").Value = "Eingebaut"
    Range("D1").Value = "Schaltflächen"
    Range("E1").Value = "ID"
    Rows(1).Font.Bold = True
    iRow = 1
    For Each oBar In Application.CommandBars
        If oBar.Type = msoBarTypePopup Then
            iRow = iRow + 1
            Cells(iRow, 1) = oBar.Name
            Cells(iRow, 2) = oBar.NameLocal
            Cells(iRow, 3) = oBar.BuiltIn
            For Each oCntr In oBar.Controls
                Cells(iRow, 4) = oCntr.Caption
                Cells(iRow, 5) = oCntr.ID
            Next oCntr
        End If
    Next oBar
    Columns.AutoFit
    Application.ScreenUpdating = False
End Sub

ClassModule: frmKontext

Private Sub cboControl_Change()
    If cboControl.Text = "" Then Exit Sub
    lblIndexControl.Caption = _
        Application.CommandBars(cboKontext.Text) _
        .Controls(cboControl.Text).ID
End Sub

Private Sub cboKontext_Change()
    Dim objBar As CommandBar
    Dim objCntr As CommandBarControl
    Set objBar = Application.CommandBars(cboKontext.Text)
    lblIndexMenu.Caption = objBar.Index
    lblNameLocal.Caption = objBar.NameLocal
    cboControl.Clear
    For Each objCntr In objBar.Controls
        cboControl.AddItem objCntr.Caption
    Next objCntr
    cboControl.ListIndex = 0
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Unload Me
    Call ListShortCutMenus
End Sub

Private Sub UserForm_Initialize()
    Dim objBar As CommandBar
    For Each objBar In CommandBars
        If objBar.Type = msoBarTypePopup Then
            cboKontext.AddItem objBar.Name
        End If
    Next objBar
    cboKontext.ListIndex = 0
End Sub

ClassModule: Tabelle3

Private Sub Worksheet_BeforeRightClick( _
    ByVal Target As Range, Cancel As Boolean)
    If Not Intersect( _
        Target, Range("A1:F16")) Is Nothing Then
        Cancel = True
    End If
End Sub

ClassModule: Tabelle4

Private Sub Worksheet_Activate()
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Dim iCounter As Integer
    On Error Resume Next
    Application.CommandBars("Months").Delete
    On Error GoTo 0
    Set oBar = Application.CommandBars.Add( _
        "Months", msoBarTypePopup, False, True)
    For iCounter = 1 To 12
        Set oBtn = oBar.Controls.Add
        With oBtn
            .Caption = Format( _
                DateSerial(1, iCounter, 1), "mmmm")
            .Style = msoButtonCaption
            .OnAction = "MonthMessage"
        End With
    Next iCounter
End Sub

Private Sub Worksheet_Deactivate()
    On Error Resume Next
    Application.CommandBars("Months").Delete
    On Error GoTo 0
End Sub

Private Sub Worksheet_BeforeRightClick( _
    ByVal Target As Range, Cancel As Boolean)
    Cancel = True
    Application.CommandBars("Months").ShowPopup
End Sub

ClassModule: Tabelle5

Private Sub Worksheet_Activate()
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Dim iCounter As Integer
    On Error Resume Next
    Application.CommandBars("Days").Delete
    On Error GoTo 0
    Set oBar = Application.CommandBars.Add( _
        "Days", msoBarTypePopup, False, True)
    Set oBtn = oBar.Controls.Add
    With oBtn
        .Caption = "Wochentage:"
        .Style = msoButtonCaption
    End With
    For iCounter = 1 To 7
        Set oBtn = oBar.Controls.Add
        With oBtn
            .Caption = Cells(5 + iCounter, 1).Text
            .Style = msoButtonCaption
            .OnAction = "SetFilter"
            If iCounter = 1 Then
                .BeginGroup = True
            End If
        End With
    Next iCounter
    Set oBtn = oBar.Controls.Add
    With oBtn
        .Caption = "Zurücksetzen"
        .Style = msoButtonCaption
        .OnAction = "ResetFilter"
        .BeginGroup = True
    End With
End Sub

Private Sub Worksheet_Deactivate()
    On Error Resume Next
    Application.CommandBars("Days").Delete
    On Error GoTo 0
End Sub

Private Sub Worksheet_BeforeRightClick(ByVal Target As Range, Cancel As Boolean)
    Cancel = True
    Application.CommandBars("Days").ShowPopup
End Sub

ClassModule: Tabelle6

Private Sub Worksheet_Activate()
    Dim oBar As CommandBar
    Dim oPopUp As CommandBarPopup
    Dim oCntr As CommandBarControl
    Dim oBtn As CommandBarButton
    On Error Resume Next
    Application.CommandBars("KonNames").Delete
    On Error GoTo 0
    Set oBar = Application.CommandBars.Add( _
        "KonNames", msoBarTypePopup, False, True)
    Set oPopUp = Application.CommandBars.FindControl(ID:=30023)
    For Each oCntr In oPopUp.Controls
        Set oBtn = oBar.Controls.Add(ID:=oCntr.ID)
    Next oCntr
End Sub

Private Sub Worksheet_Deactivate()
    On Error Resume Next
    Application.CommandBars("KonNames").Delete
    On Error GoTo 0
End Sub

Private Sub Worksheet_BeforeRightClick( _
    ByVal Target As Range, Cancel As Boolean)
    Cancel = True
    Application.CommandBars("KonNames").ShowPopup
End Sub

ClassModule: Tabelle7

Private Sub Worksheet_Activate()
    Dim oBtn As CommandBarButton
    With Application.CommandBars("Cell")
        On Error Resume Next
        .Controls("Spalten aus").Delete
        .Controls("Spalten ein").Delete
    End With
    On Error GoTo 0
    Set oBtn = .Controls.Add
    oBtn.Caption = "Spalten aus"
    oBtn.Style = msoButtonCaption
    oBtn.OnAction = "SpaltenAus"
    oBtn.BeginGroup = True
    Set oBtn = .Controls.Add
    oBtn.Caption = "Spalten ein"
    oBtn.Style = msoButtonCaption
    oBtn.OnAction = "SpaltenEin"
End Sub

Private Sub Worksheet_Deactivate()
    On Error Resume Next
    With Application.CommandBars("Cell")
        .Controls("Spalten aus").Delete
        .Controls("Spalten ein").Delete
    End With
    On Error GoTo 0
End Sub

ClassModule: Tabelle8

Private Sub Worksheet_Activate()
    Dim oPopUp As CommandBarControl
    Dim oBtn As CommandBarButton
    On Error Resume Next
    Application.CommandBars("Cell") _
        .Controls("Spalten").Delete
    On Error GoTo 0
    Set oPopUp = Application.CommandBars( _
        "Cell").Controls.Add(msoControlPopup)
    With oPopUp
        .Caption = "Spalten"
        .BeginGroup = True
    End With
    Set oBtn = oPopUp.Controls.Add
    With oBtn
        .Caption = "Spalten aus"
        .Style = msoButtonCaption
        .OnAction = "SpaltenAus"
    End With
    Set oBtn = oPopUp.Controls.Add
    With oBtn
        .Caption = "Spalten ein"
        .Style = msoButtonCaption
        .OnAction = "SpaltenEin"
    End With
End Sub

Private Sub Worksheet_Deactivate()
    On Error Resume Next
    Application.CommandBars("Cell") _
        .Controls("Spalten").Delete
    On Error GoTo 0
End Sub

ClassModule: Tabelle9

Private Sub Worksheet_BeforeRightClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Dim iCol As Integer
    If WorksheetFunction.CountA( _
        Range(Cells(Target.Row, 3), _
        Cells(Target.Row, 256))) = 0 Then Exit Sub
    Cancel = True
    On Error Resume Next
    Application.CommandBars("RowValues").Delete
    On Error GoTo 0
    Set oBar = Application.CommandBars.Add( _
        "RowValues", msoBarTypePopup)
    For iCol = 3 To 256
        If Not IsEmpty(Cells(Target.Row, iCol)) Then
            Set oBtn = oBar.Controls.Add
            With oBtn
                .Caption = Cells(Target.Row, iCol).Value
                .Style = msoButtonCaption
                .OnAction = "InsertValue"
            End With
        End If
    Next iCol
    Application.CommandBars("RowValues").ShowPopup
End Sub

Private Sub Worksheet_Deactivate()
    On Error Resume Next
    Application.CommandBars("RowValues").Delete
    On Error GoTo 0
End Sub

ClassModule: Tabelle10

Private Sub Worksheet_BeforeRightClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Dim wks As Worksheet
    Cancel = True
    On Error Resume Next
    Application.CommandBars("SheetSelect").Delete
    On Error GoTo 0
    Set oBar = Application.CommandBars.Add( _
        "SheetSelect", msoBarTypePopup)
    For Each wks In Worksheets
        If wks.Index <> Me.Index Then
            Set oBtn = oBar.Controls.Add
            With oBtn
                .Caption = wks.Name
                .Style = msoButtonCaption
                .OnAction = "SelectSheet"
            End With
        End If
    Next wks
    Application.CommandBars("SheetSelect").ShowPopup
End Sub

Private Sub Worksheet_Deactivate()
    On Error Resume Next
    Application.CommandBars("SheetSelect").Delete
    On Error GoTo 0
End Sub
```


Dem Zell-Kontextmenü ein neues Menü hinzufügen

Problem:	Solange die Arbeitsmappe geöffnet ist, wird dem Zellkontextmenü ein Menü "Auswahl" hinzugefügt, in dem Summe, Zählen, Mittelwert, Anzahl, Min und Max der Zellauswahl angezeigt werden.
Lösung:	Geben Sie den Ereigniscode in die genannten Module der Arbeitsmappe ein.
VBA-Code:	ClassModule: DieseArbeitsmappe Dim xlApplication As New clsApp Private Sub Workbook_BeforeClose(Cancel As Boolean) Set xlApplication.xlApp = Nothing On Error Resume Next Application.CommandBars("Cell").Controls("Auswahl").Delete On Error GoTo 0 End Sub Private Sub Workbook_Open() Set xlApplication.xlApp = Application End Sub ClassModule: clsApp Public WithEvents xlApp As Excel.Application Private Sub xlApp_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean) Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Dim oFct As WorksheetFunction Set oFct = WorksheetFunction With Application.CommandBars("Cell") On Error Resume Next .Controls("Auswahl").Delete On Error GoTo 0 If Selection.Cells.Count = 1 Then Exit Sub If WorksheetFunction.Count(Selection) = 0 Then Exit Sub Set oPopUp = .Controls.Add(msoControlPopup) oPopUp.BeginGroup = True Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Summe: " & oFct.Sum(Selection) .Style = msoButtonCaption .OnAction = "CmdCopy" End With oPopUp.Caption = "Auswahl" Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Zählen: " & oFct.Count(Selection) .Style = msoButtonCaption .OnAction = "CmdCopy" End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Anzahl: " & oFct.CountA(Selection) .Style = msoButtonCaption .OnAction = "CmdCopy" End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Mittelwert: " & Format(oFct.Average(Selection), "0.00") .Style = msoButtonCaption .OnAction = "CmdCopy" End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Maximalwert: " & oFct.Max(Selection) .Style = msoButtonCaption .OnAction = "CmdCopy" End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Minimalwert: " & oFct.Min(Selection) .Style = msoButtonCaption .OnAction = "CmdCopy" End With End With End Sub StandardModule: Modul1 Sub CmdCopy() Dim ClipAblage As DataObject Dim sTxt As String Set ClipAblage = New DataObject sTxt = Application.CommandBars.ActionControl.Caption sTxt = Right(sTxt, Len(sTxt) - InStr(sTxt, ": ") - 1) ClipAblage.SetText sTxt ClipAblage.PutInClipboard End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neues Zellkontextmenü bei Auswahl mehrerer Kompletzeilen

Problem:	Es wird nur dann ein eigenes Zell-Kontextmenü angezeigt, wenn eine oder mehrere Zeilen markiert sind.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_BeforeRightClick(ByVal Target As Range, Cancel As Boolean) Dim oBar As CommandBar Dim oBtn As CommandBarButton If Selection.Columns.Count = 256 Then Cancel = True On Error GoTo ERRORHANDLER Set oBar = Application.CommandBars.Add("MyContextBar", msoBarPopup) Set oBtn = oBar.Controls.Add With oBtn .Caption = "Meldung" .OnAction = "Meldung" .Style = msoButtonCaption End With oBar.ShowPopup End If ERRORHANDLER: On Error Resume Next Application.CommandBars("MyContextBar").Delete Resume End Sub StandardModule: Modul1 Sub Meldung() If Selection.Rows.Count = 1 Then MsgBox "Zeile " & Selection.Address(False, False) & " markiert!" Else MsgBox "Zeilen " & Selection.Address(False, False) & " markiert!" End If Application.CommandBars("MyContextBar").Delete End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neue Menüschaftfläche mit benutzerdefiniertem Icon

Problem: Dem Zellkontextmenü wird eine Schaltfläche mit benutzerdefiniertem Icon hinzugefügt. Bitte beachten: Der CommandBar-Name des Quell-Icons muss angepasst werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn zwei Schaltflächen zu.

VBA-Code:

```
StandardModule: Modul1

Sub GetIconFace()
    Dim oSource As CommandBarButton
    Dim oBtn As CommandBarButton
    On Error Resume Next
    Application.CommandBars("Cell").Controls("Test").Delete
    On Error GoTo 0
    On Error GoTo ERRORHANDLER
    Set oBtn = Application.CommandBars("Cell").Controls.Add
    With oBtn
        .Caption = "Test"
        .OnAction = "Irgendwas"
        .Style = msoButtonIconAndCaption
        Set oSource = Application.CommandBars("MyCmdBar").Controls(14)
        oSource.CopyFace
        oBtn.PasteFace
    End With
    Exit Sub
ERRORHANDLER:
    MsgBox "Das Quell-Icon konnte nicht gefunden werden!"
End Sub

Sub Irgendwas()
    MsgBox "Ich bin nur ein Platzhalter"
End Sub

Sub Loeschen()
    On Error Resume Next
    Application.CommandBars("Cell").Controls("Test").Delete
    On Error GoTo 0
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Feste Fenster- und Cursorpositionen bei Blattwechsel

Problem:	Bei einem Blattwechsel über das Menü GoTo im Zellkontextmenü (rechte Maustaste) sollen nach Blattauswahl die Cursor- und Fensterpositionen beibehalten werden. Dies gilt für alle Arbeitsblätter.
Lösung:	Geben Sie den nachfolgenden Code in die genannten Module ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) On Error Resume Next Application.CommandBars("Cell").Controls("GoTo").Delete On Error GoTo 0 End Sub Private Sub Workbook_Open() Call Workbook_SheetBeforeRightClick(ActiveSheet, ActiveCell, False) End Sub Private Sub Workbook_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean) Dim wks As Worksheet Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton On Error Resume Next Application.CommandBars("Cell").Controls("GoTo").Delete On Error GoTo 0 Set oPopUp = Application.CommandBars("Cell").Controls.Add(msoControlPopup) oPopUp.Caption = "GoTo" For Each wks In Worksheets If wks.Name <> ActiveSheet.Name Then Set oBtn = oPopUp.Controls.Add With oBtn .Caption = wks.Name .OnAction = "GoToWks" .Style = msoButtonCaption End With End If Next wks End Sub StandardModule: Modul1 Sub GoToWks() Dim iRow As Integer, iCol As Integer Dim sRange As String iRow = ActiveWindow.ScrollRow iCol = ActiveWindow.ScrollColumn sRange = ActiveCell.Address Worksheets(CommandBars.ActionControl.Caption).Select ActiveWindow.ScrollRow = iRow ActiveWindow.ScrollColumn = iCol Range(sRange).Select End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dem Zellkontextmenü ein Menü "Zahlungsbedingungen" hinzufügen

Problem: Dem Zell-Kontextmenü wurde ein Menü "Zahlungsbedingungen" mit den Schaltflächen "14 Tage" und "30 Tage" hinzugefügt. Bei Anklicken wird die Zahlungsweise eingetragen.

Lösung: Geben Sie den nachfolgenden Code in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    Application.CommandBars("Cell") _
        .Controls("Zahlungsbedingungen").Delete
    On Error GoTo 0
End Sub

Private Sub Workbook_Open()
    Dim oPopUp As CommandBarControl
    Dim oBtn As CommandBarButton
    With Application.CommandBars("Cell")
        On Error Resume Next
        .Controls("Zahlungsbedingungen").Delete
        On Error GoTo 0
        Set oPopUp = .Controls.Add(msoControlPopup)
    End With
    oPopUp.Caption = "Zahlungsbedingungen"
    Set oBtn = oPopUp.Controls.Add
    With oBtn
        .Caption = "14 Tage"
        .OnAction = "SetPayment"
        .Style = msoButtonCaption
    End With
    Set oBtn = oPopUp.Controls.Add
    With oBtn
        .Caption = "30 Tage"
        .OnAction = "SetPayment"
        .Style = msoButtonCaption
    End With
End Sub

StandardModule: Modul1

Sub SetPayment()
    ActiveCell.Value = _
        "Zahlungsbedingungen: " & _
        Application.CommandBars("Cell") _
            .Controls("Zahlungsbedingungen") _
            .Controls(Application.Caller(1)).Caption
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Auflisten von OptionButtons in Kontextmenü

Problem: Im Zellkontextmenü (rechte Maustaste) sollen alle Optionsfelder dieses Blattes aufgelistet und die Zelle unterhalb des ausgewählten Optionsfeldes selektiert werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call CmdDelete
End Sub

ClassModule: Tabelle1

Private Sub Worksheet_BeforeRightClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim oBar As CommandBar
    Dim oButton As CommandBarButton
    Dim chb As Shape
    Cancel = True
    Call CmdDelete
    Set oBar = Application.CommandBars.Add("CheckBoxes", msoBarPopup)
    For Each chb In ActiveSheet.Shapes
        If Left(chb.Name, 12) = "OptionButton" Then
            Set oButton = oBar.Controls.Add
            With oButton
                .Caption = chb.Name
                .OnAction = "Positionieren"
                .Style = msoButtonCaption
            End With
        End If
    Next chb
    oBar.ShowPopup
End Sub

StandardModule: basMain

Sub Positionieren()
    Tabelle1.Shapes(Application.Caller(1)) _
        .TopLeftCell.Select
End Sub

Sub CmdDelete()
    On Error GoTo ERRORHANDLER
    Application.CommandBars("CheckBoxes").Delete
ERRORHANDLER:
End Sub
```



Kontextmenü mit Buchungsvorgaben erstellen

Problem:	Wie kann ich beim Wechsel in ein Buchungsblatt ein Kontextmenü mit Einträgen aus einem Datenblatt erstellen lassen? Nach Auswahl eines Menüpunktes soll der entsprechende Datensatz eingetragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub ClassModule: Tabelle1 Private Sub Worksheet_Activate() Dim oCmdBar As CommandBar Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Dim iRow As Integer Call CmdDelete Set oCmdBar = Application.CommandBars.Add(_ Name:="Buchen", Position:=msoBarPopup) With Worksheets("Vorgaben") For iRow = 1 To WorksheetFunction.CountA(.Columns("B")) If Not IsEmpty(.Cells(iRow, 1)) Then Set oPopUp = oCmdBar.Controls.Add(msoControlPopup) oPopUp.Caption = .Cells(iRow, 1) End If Set oBtn = oPopUp.Controls.Add oBtn.Caption = .Cells(iRow, 2) oBtn.OnAction = "Buchen" oBtn.Style = msoButtonCaption Next iRow End With End Sub Private Sub Worksheet_BeforeRightClick(ByVal Target As Range, Cancel As Boolean) Cancel = True Application.CommandBars("Buchen").ShowPopup End Sub Private Sub Worksheet_Deactivate() Call CmdDelete End Sub StandardModule: basMain Sub Buchen() Dim rngFind As Range Dim intCol As Integer If ActiveCell.Column > 3 Then Beep MsgBox "Sie müssen sich im Spaltenbereich ""A:C"" befinden!" Exit Sub End If With Worksheets("Vorgaben") Set rngFind = .Columns(2).Find(CommandBars("Buchen"). _ Controls(Application.Caller(2)). _ Controls(Application.Caller(1)).Caption, _ lookat:=xlWhole, LookIn:=xlValues) For intCol = 1 To 3 Cells(ActiveCell.Row, intCol) = rngFind.Offset(0, intCol - 1) Next intCol End With End Sub Sub CmdDelete() On Error GoTo ERRORHANDLER Application.CommandBars("Buchen").Delete ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bei geteiltem Fenster Kontextmenü im jeweils anderen Fenster

Problem:	Wie kann ich bei einem senkrecht geteilten Fenster beim Drücken der rechten Maustaste ein benutzerdefiniertes Kontextmenü im jeweils anderen Fenster aufklappen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub Private Sub Workbook_Open() Dim oBar As CommandBar Dim oBtn As CommandBarButton Call CmdDelete Set oBar = Application.CommandBars.Add("Links", msoBarPopup) Set oBtn = oBar.Controls.Add With oBtn .Caption = "Links" .OnAction = "MeldungLinks" End With Set oBar = Application.CommandBars.Add("Rechts", msoBarPopup) Set oBtn = oBar.Controls.Add With oBtn .Caption = "Rechts" .OnAction = "MeldungRechts" End With End Sub ClassModule: Tabelle1 Private Sub Worksheet_BeforeRightClick(_ ByVal Target As Range, Cancel As Boolean) Cancel = True If ActiveCell.Column < 6 Then CommandBars("Links").ShowPopup 500, 200 Else CommandBars("Rechts").ShowPopup 200, 200 End If End Sub StandardModule: basMain Sub MeldungLinks() MsgBox "Aufruf aus der linken Abteilung!" End Sub Sub MeldungRechts() MsgBox "Aufruf aus der rechten Abteilung!" End Sub Sub CmdDelete() On Error Resume Next With Application .CommandBars("Links").Delete .CommandBars("Rechts").Delete End With On Error GoTo 0 End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kürzel über Kontextmenü auswählen und markieren

Problem: Wie kann ich in einem Tabellenblatt festgelegte Kürzel in ein Kontextmenü einlesen lassen? Nach Aufruf des Kontextmenüs soll die Zelle mit dem ausgewählten Kürzel mit einem roten Hintergrund versehen werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call CmdDelete
End Sub

ClassModule: Tabelle1

Private Sub Worksheet_BeforeRightClick( _
    ByVal Target As Range, Cancel As Boolean)
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Dim iRow As Integer
    Cancel = True
    On Error Resume Next
    Application.CommandBars("KuerzelRot").Delete
    On Error GoTo 0
    Set oBar = Application.CommandBars.Add("KuerzelRot", msoBarPopup)
    iRow = 1
    Set oBtn = oBar.Controls.Add
    oBtn.Caption = "Kürzel:"
    Do Until IsEmpty(Worksheets("Kürzel").Cells(iRow, 1))
        Set oBtn = oBar.Controls.Add
        With oBtn
            .Caption = Worksheets("Kürzel").Cells(iRow, 1)
            .OnAction = "Faerben"
            .Style = msoButtonCaption
            If oBtn.Index = 2 Then .BeginGroup = True
        End With
        iRow = iRow + 1
    Loop
    oBar.ShowPopup
End Sub

StandardModule: basMain

Sub CmdDelete()
    On Error GoTo ERRORHANDLER
    Application.CommandBars("KuerzelRot").Delete
ERRORHANDLER:
End Sub

Sub Faerben()
    Dim rng As Range
    Dim sTxt As String
    sTxt = Application.CommandBars("KuerzelRot") _
        .Controls(Application.Caller(1) - 1).Caption
    For Each rng In Range("A1:F16")
        If rng.Value = sTxt Then
            rng.Interior.ColorIndex = 3
        End If
    Next rng
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten der aktuellen Zeile im Kontextmenü anzeigen

Problem: Wie kann ich mir die Daten der aktuellen Zeile im Zellkontextmenü anzeigen lassen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call CmdDelete
End Sub

StandardModule: basMain

Sub CmdDelete()
    On Error GoTo ERRORHANDLER
    Application.CommandBars("DatenMaske").Delete
ERRORHANDLER:
End Sub

ClassModule: Tabelle1

Private Sub Worksheet_BeforeRightClick(ByVal _
    Target As Range, Cancel As Boolean)
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Dim iCol As Integer
    Cancel = True
    Call CmdDelete
    Set oBar = Application.CommandBars.Add("DatenMaske", msoBarPopup)
    Do Until IsEmpty(Cells(1, iCol + 1))
        Set oBtn = oBar.Controls.Add
        With oBtn
            .Caption = Cells(1, iCol + 1).Value & ": " & _
                Cells(ActiveCell.Row, iCol + 1).Value
        End With
        iCol = iCol + 1
    Loop
    oBar.ShowPopup
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellkontextmenü durch Jahreskalender ersetzen

Problem:	Wie kann ich das Zellkontextmenü durch den Kalender des aktuellen Jahres ersetzen? Bei Anklicken eines Tages soll das ausgewählte Datum in die aktive Zelle eingetragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub Private Sub Workbook_Open() Dim oBar As CommandBar Dim oCntr As CommandBarControl Dim oBtn As CommandBarButton Dim iMonth As Integer, iDay As Integer Call CmdDelete Set oBar = Application.CommandBars.Add("Datum", msoBarPopup) Set oBtn = oBar.Controls.Add With oBtn .Caption = Year(Date) .Style = msoButtonCaption End With For iMonth = 1 To 12 Set oCntr = oBar.Controls.Add(msoControlPopup) oCntr.Caption = Format(DateSerial(Year(Date), iMonth, 1), "mmm") If iMonth = 1 Then oCntr.BeginGroup = True For iDay = 1 To Day(DateSerial(Year(Date), iMonth + 1, 0)) Set oBtn = oCntr.Controls.Add With oBtn .Caption = iDay .Style = msoButtonCaption .OnAction = "Kalender" End With Next iDay Next iMonth End Sub StandardModule: basMain Sub Kalender() ActiveCell.NumberFormat = "dd.mm.yyyy" With CommandBars("Datum") ActiveCell.Value = DateSerial(.Controls(1).Caption, _ Application.Caller(2) - 2, Application.Caller(1)) End With End Sub Sub CmdDelete() On Error GoTo ERRORHANDLER Application.CommandBars("Datum").Delete ERRORHANDLER: End Sub ClassModule: Tabelle1 Private Sub Worksheet_BeforeRightClick(_ ByVal Target As Range, Cancel As Boolean) Cancel = True Application.CommandBars("Datum").ShowPopup End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellkontextmenü ein Neues Menü mit Datum & Zeit hinzufügen

Problem: Wie kann ich dem Zellkontextmenü ein neues Menü mit einer Abrufmöglichkeit für Datum und Zeit hinzufügen? Das Menü soll nur in der aktuellen Arbeitsmappe wirksam sein.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_Activate()
    Dim oPopUp As CommandBarControl
    Dim oBtn As CommandBarButton
    Call CmdDelete
    Set oPopUp = Application.CommandBars("Cell"). _
        Controls.Add(msoControlPopup, before:=1)
    oPopUp.Caption = "Datum und Zeit"
    Set oBtn = oPopUp.Controls.Add
    With oBtn
        .Caption = "Datum"
        .OnAction = "DatumErmitteln"
        .Style = msoButtonIconAndCaption
        .FaceId = 1094
    End With
    Set oBtn = oPopUp.Controls.Add
    With oBtn
        .Caption = "Zeit"
        .BeginGroup = True
        .OnAction = "ZeitErmitteln"
        .Style = msoButtonIconAndCaption
        .FaceId = 33
    End With
End Sub

Private Sub Workbook_Deactivate()
    Call CmdDelete
End Sub

StandardModule: basMain

Sub DatumErmitteln()
    MsgBox Date
End Sub

Sub ZeitErmitteln()
    MsgBox Time
End Sub

Sub CmdDelete()
    On Error GoTo ERRORHANDLER
    Application.CommandBars("Cell").Controls("Datum und Zeit").Delete
ERRORHANDLER:
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kontextmenü zur Auswahl von Arbeitsblättern aufrufen

Problem: Wie kann ich mir bei Rechtsklick in einer Tabelle ein Kontextmenü mit der aktuellen Arbeitsblattauswahl anzeigen lassen und in das jeweils ausgewählte Arbeitsblatt springen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_SheetBeforeRightClick( _
    ByVal Sh As Object, ByVal Target As Excel.Range, _
    Cancel As Boolean)
    Dim oBar As CommandBar
    Dim oBtn As CommandBarButton
    Dim wks As Worksheet
    Cancel = True
    Call CmdDelete
    Set oBar = Application.CommandBars.Add("Register", msoBarPopup)
    For Each wks In ThisWorkbook.Worksheets
        Set oBtn = oBar.Controls.Add
        With oBtn
            .Caption = wks.Name
            .Style = msoButtonCaption
            .OnAction = "mnuAuswahl"
        End With
    Next wks
    oBar.ShowPopup
End Sub

StandardModule: basMain

Sub mnuAuswahl()
    Worksheets(Application.Caller(1)).Select
End Sub

Sub CmdDelete()
    On Error GoTo ERRORHANDLER
    Application.CommandBars("Register").Delete
ERRORHANDLER:
End Sub
```

Tabellenblattbereich sortiert als Zellkontextmenü

Problem:	Wie kann ich einen Tabellenblattbereich sortiert in ein in ein Zell-Kontextmenü übernehmen und dieses bei Klick mit der rechten Maustaste ausschließlich in Zelle B2 anzeiten lassen? Der jeweil ausgewählte Menüpunkt soll in Zelle B2 eingetragen werden.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul des Arbeitsblattes ein.
VBA-Code:	ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub ClassModule: Tabelle1 Private Sub Worksheet_BeforeRightClick(_ ByVal Target As Excel.Range, Cancel As Boolean) Dim oBar As CommandBar Dim oBtn As CommandBarButton Dim arrItems() As String Dim arrMenu As Variant Dim iRow As Integer, iRowM As Integer If ActiveCell.Address <> "\$B\$1" Then Exit Sub Cancel = True Call CmdDelete Set oBar = CommandBars.Add _ (Name:="MyContext", Position:=msoBarPopup) iRow = 5 Do Until IsEmpty(Cells(iRow, 5)) iRowM = iRowM + 1 ReDim Preserve arrItems(iRowM) arrItems(iRowM) = Cells(iRow, 5) iRow = iRow + 1 Loop arrMenu = SortMenu(arrItems) For iRow = 1 To UBound(arrItems) Set oBtn = oBar.Controls.Add With oBtn .Caption = arrItems(iRow) .OnAction = "Eingabe" .Style = msoButtonCaption End With Next iRow oBar.ShowPopup End Sub Private Function SortMenu(arrItems) Dim iCount As Integer, iA As Integer, iB As Integer Dim sTmp As String iCount = UBound(arrItems) For iA = 1 To iCount For iB = iA + 1 To iCount If arrItems(iA) > arrItems(iB) Then sTmp = arrItems(iA) arrItems(iA) = arrItems(iB) arrItems(iB) = sTmp End If Next iB Next iA SortMenu = arrItems End Function StandardModule: basMain Sub CmdDelete() On Error GoTo ERRORHANDLER Application.CommandBars("MyContext").Delete ERRORHANDLER: End Sub Sub Eingabe() ActiveCell.Value = Application.CommandBars("MyContext").Controls(Application.Caller(1)).Caption End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Berechnung des ausgewählten Bereiches in der Statusleiste

Problem: Über das Zellkontextmenü (rechte Maustaste) soll in der Statusleiste eine Multiplikation der ausgewählten Zellwerte mit 3 angezeigt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call CmdDelete
    Application.StatusBar = False
End Sub

Private Sub Workbook_Open()
    Dim oBtn As CommandBarButton
    Application.DisplayStatusBar = True
    Call CmdDelete
    Set oBtn = Application.CommandBars("Cell").Controls.Add
    With oBtn
        .Caption = "DreiMal"
        .BeginGroup = True
        .OnAction = "Multiplizieren"
    End With
End Sub

StandardModule: basMain

Sub Multiplizieren()
    Application.StatusBar = "Auswahl * 3 = " & _
        WorksheetFunction.Sum(Selection) * 3
End Sub

Sub CmdDelete()
    On Error GoTo ERRORHANDLER
    Application.CommandBars("Cell").Controls("DreiMal").Delete
ERRORHANDLER:
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kontextmenüs und ihre Elemente auflisten

Problem:	Wie kann ich alle Kontextmenüs mit ihren Elementen in einem Arbeitsblatt auflisten lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub ListShortCutMenus() Dim oBar As CommandBar Dim iRow As Integer, iCounter As Integer Application.ScreenUpdating = False For Each oBar In Application.CommandBars If oBar.Type = msoBarTypePopup Then iRow = iRow + 1 Cells(iRow, 1) = oBar.Index Cells(iRow, 2) = oBar.Name For iCounter = 1 To oBar.Controls.Count Cells(iRow, iCounter + 2) = _ oBar.Controls(iCounter).Caption Next iCounter End If Next oBar Cells.EntireColumn.AutoFit End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kontextmenü in der Entwicklungsumgebung Menüpunkt hinzufügen

Problem: Wie kann ich dem Kontextmenü in der Entwicklungsumgebung einen Menüpunkt hinzufügen und diesem eine Ereignisprozedur zuweisen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Dim mclsAddMenu As New clsAddMenu

Private Sub Workbook_Open()
    Call Zurueck
    mclsAddMenu.AddMenuItem
End Sub

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Set mclsAddMenu = Nothing
    Call Zurueck
End Sub

StandardModule: basMain

Sub Zurueck()
    On Error Resume Next
    Application.VBE.CommandBars("Code Window") _
        .Controls("Meldung").Delete
    On Error GoTo 0
End Sub

ClassModule: clsAddMenu

Public WithEvents Meldung As VBIDE.CommandBarEvents

Public Sub AddMenuItem()
    Dim ctlTopMenu As CommandBarButton
    Set ctlTopMenu = Application.VBE.CommandBars("Code Window") _
        .Controls.Add(Type:=msoControlButton)
    ctlTopMenu.Caption = "Meldung"
    ctlTopMenu.Enabled = True
    Set Meldung = Application.VBE.Events.CommandBarEvents(ctlTopMenu)
End Sub

Private Sub Meldung_Click( _
    ByVal cmdBar As Object, handled As Boolean, Cancel As Boolean)
    MsgBox "Hallo Welt!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Trennlinie in Kontextmenü einfügen

Problem:	Wie kann ich im Zellkontextmenü eine Trennlinie einfügen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CommandButton1_Click() Call Trennlinie End Sub StandardModule: basMain Sub Trennlinie() With CommandBars("Cell") .Controls(.Controls.Count - 1).BeginGroup = True End With End Sub Sub Zurueck() With CommandBars("Cell") .Controls(.Controls.Count - 1).BeginGroup = False End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellkontextmenü löschen und aufbauen

Problem: Wie kann ich die Menüpunkte des eingebauten Zellkontextmenüs komplett entfernen und eigene Menüpunkte einfügen.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call ResetContext
End Sub

StandardModule: basMain

Sub EditContext()
    Dim oBtn As CommandBarButton
    With Application.CommandBars("Cell")
        Do While .Controls.Count > 0
            .Controls(1).Delete
        Loop
        Set oBtn = .Controls.Add
    End With
    With oBtn
        .Caption = "MeinBefehl"
        .OnAction = "MeinMakro"
    End With
End Sub

Sub ResetContext()
    Application.CommandBars("Cell").Reset
End Sub

Sub MeinMakro()
    MsgBox "Makroaufruf aus dem Zellkontextmenü!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellkontextmenü Befehle hinzufügen

Problem:	Wie kann ich dem Kontextmenü eigene Schaltflächen hinzufügen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub NewMenuItem() Dim oBtn As CommandBarButton On Error Resume Next Application.CommandBars("Cell") _ .Controls("Notiz bearbeiten").Delete On Error GoTo 0 Set oBtn = CommandBars("Cell").Controls.Add With oBtn .Caption = "Notiz bearbeiten" .OnAction = "Notiz" .Style = msoButtonCaption End With End Sub Sub Notiz() MsgBox "Ich bin die Notiz!" End Sub Sub CmdBarReset() On Error Resume Next Application.CommandBars("Cell") _ .Controls("Notiz bearbeiten").Delete End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Daten der Schaltflächen eines Menüs ermitteln

Problem:	Die Daten eines jeden Menüpunktes eines vorgegebenen Menüs werden aufgelistet.
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetMenu() Dim cmd As Object Dim iCol As Integer Dim sCntr As String sCntr = "Excel-Hilfen" On Error GoTo ERRORHANDLER Set cmd = Application.CommandBars("Worksheet Menu Bar").Controls(sCntr) iCol = 1 Range("A1").Value = "Übergeordnet" Range("A2").Value = "Typ" Range("A3").Value = "Aufschrift" Range("A4").Value = "Makro" Columns(1).Font.Bold = True Call GetSubMenu(cmd, iCol) Exit Sub ERRORHANDLER: MsgBox "Das Menü " & sCntr & " wurde nicht gefunden!" End Sub Private Sub GetSubMenu(cmd As Object, iCol As Integer) Dim cnt As CommandBarControl With cmd For Each cnt In .Controls iCol = iCol + 1 Cells(1, iCol).Value = cnt.Parent.Name Cells(2, iCol).Value = TypeName(cnt) Cells(3, iCol).Value = cnt.Caption If TypeName(cnt) <> "CommandBarPopup" Then Cells(4, iCol).Value = cnt.OnAction Else Call GetSubMenu(cnt, iCol) End If Next cnt End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Format-Menü löschen und ShortCut umbenennen

Problem:	In der Arbeitsblatt-Menüleiste soll das Format-Menü gelöscht und der ShortCut für das Daten-Menü auf t gesetzt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub EditCmdBar() On Error GoTo ERRORHANDLER With Application.CommandBars("Worksheet Menu Bar") .Controls("Daten").Caption = "Da&ten" .Controls("Format").Delete End With Exit Sub ERRORHANDLER: MsgBox "Ein Element wurde nicht gefunden!" End Sub Sub Zurueck() Application.CommandBars("Worksheet Menu Bar").Reset End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anpassen-Dialog über VBA aufrufen

Problem:	Der Anpassen-Dialog aus dem Extras-Menü soll über VBA aufgerufen werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub CallAnpassen() Dim oCntr As CommandBarControl Set oCntr = Application.CommandBars.FindControl(ID:=797) oCntr.Execute End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfen, ob ein Menü vorhanden ist

Problem:	Wie kann ich überprüfen, ob in der Arbeitsblattmenüleiste ein bestimmtes Menü enthalten ist und wo es sich, wenn ja, befindet?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub MenuTest() Dim oCntr As CommandBarControl Dim bln As Boolean With Application.CommandBars("Worksheet Menu Bar") For Each oCntr In .Controls If oCntr.Type = msoControlButtonPopup And _ oCntr.Caption = "MeinMenue" Then bln = True Exit For End If Next oCntr End With If bln = True Then MsgBox "Das Menü ""MeinMenue"" existiert!" Else MsgBox "Das Menü ""MeinMenue"" existiert nicht!" End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Das Extras-Menü aktivieren / deaktivieren

Problem:	Das Extras-Menü soll aktiviert bzw. deaktiviert werden.
Lösung:	Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.
VBA-Code:	<pre>StandardModule: Modul1 Sub ExtrasEinAus() Dim ctr As CommandBarControl Set ctr = Application.CommandBars("Worksheet Menu Bar").Controls("Extras") With ctr .Enabled = Not .Enabled End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

In der Arbeitsblattmenüleiste ein Menü "Dateiliste" anlegen

Problem:	Beim Öffnen der Arbeitsmappe soll ein Menü Dateiliste mit einer Dateiauswahl angelegt und beim Schließen der Arbeitsmappe wieder gelöscht werden.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) With Application.CommandBars("Worksheet Menu Bar") On Error Resume Next .Controls("Dateiliste").Delete On Error GoTo 0 End With End Sub Private Sub Workbook_Open() Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton With Application.CommandBars("Worksheet Menu Bar") On Error Resume Next .Controls("Dateiliste").Delete On Error GoTo 0 Set oPopUp = .Controls.Add(_ Type:=msoControlPopup, _ before:=.Controls.Count, _ temporary:=True) End With oPopUp.Caption = "Dateiliste" Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Datei A" .OnAction = "DateiA" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Datei B" .OnAction = "DateiB" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Datei C" .OnAction = "DateiC" .Style = msoButtonCaption End With End Sub StandardModule: basMain Sub DateiA() Workbooks.Open "c:\test\test1.xls" End Sub Sub DateiB() Workbooks.Open "c:\test\test2.xls" End Sub Sub DateiC() Workbooks.Open "c:\test\test3.xls" End Sub</pre>

Neues Menü im Extras-Menü

Problem:	Wie kann ich im Extras-Menü ein neues Menü mit 3 Menüpunkten anlegen? Den Menüpunkten soll Code zum Berechnen von Quadratmeter Außenfläche und Kubikmeter hinterlegt sein.
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Dim oPopUp As CommandBarPopup Set oPopUp = Application.CommandBars("Worksheet Menu Bar").FindControl(ID:=30007) On Error Resume Next oPopUp.Controls("Berechnungen").Delete On Error GoTo 0 End Sub Private Sub Workbook_Open() Dim oPopUpA As CommandBarPopup Dim oPopUpB As CommandBarPopup Dim oButton As CommandBarButton Set oPopUpA = Application.CommandBars(_ "Worksheet Menu Bar").FindControl(ID:=30007) On Error Resume Next oPopUpA.Controls("Berechnungen").Delete On Error GoTo 0 Set oPopUpB = oPopUpA.Controls.Add(msoControlPopup) With oPopUpB .Caption = "Berechnungen" .BeginGroup = True End With Set oButton = oPopUpB.Controls.Add With oButton .Caption = "Daten anlegen" .OnAction = "AnlegenDaten" End With Set oButton = oPopUpB.Controls.Add With oButton .Caption = "Cbm errechnen" .OnAction = "ErrechnenCbm" End With Set oButton = oPopUpB.Controls.Add With oButton .Caption = "Qm Errechnen" .OnAction = "ErrechnenQm" End With oPopUpB.Visible = True End Sub StandardModule: basMain Sub AnlegenDaten() Dim dFactor As Double Dim iRow As Integer, iCol As Integer dFactor = 1 Range("A1") = "Länge" Range("B1") = "Breite" Range("C1") = "Höhe" Range("D1") = "Cbm" Range("E1") = "qmAfl" Rows(1).Font.Bold = True For iRow = 2 To 10 For iCol = 1 To 3 Cells(iRow, iCol) = 1.1 + dFactor dFactor = dFactor + dFactor * 0.1 Next iCol Next iRow Columns("A:C").NumberFormat = "#,##0.00" Columns("D").NumberFormat = "#,##0.000" Columns("E").NumberFormat = "#,##0.00" Columns("A:E").HorizontalAlignment = xlRight End Sub Sub ErrechnenCbm() Dim iRow As Integer For iRow = 2 To 10 Cells(iRow, 4).FormulaR1C1 = "=RC[-3]*RC[-2]*RC[-1]" Next iRow End Sub Sub ErrechnenQm() Dim iRow As Integer For iRow = 2 To 10 Cells(iRow, 5).FormulaR1C1 = _ "=2*(RC[-3]*RC[-2]+RC[-3]*RC[-1]+RC[-2]*RC[-1])" Next iRow End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblattmenüleiste neues Menü mit Befehlen hinzufügen

Problem:	Wie kann ich der Standardmenüleiste ein neues Menü mit Befehlen hinzufügen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Dim oBar As CommandBar Dim oPopUp As CommandBarPopup Dim oBtn As CommandBarButton Call CmdDelete Set oBar = Application.CommandBars("Worksheet Menu Bar") Set oPopUp = oBar.Controls.Add(msoControlPopup, before:=oBar.Controls.Count) oPopUp.Caption = "Projektstatus" Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Budget Doppelblatt" .OnAction = "a_bud_doppelblatt" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Budget Einzelblatt" .OnAction = "a_bud_einzelblatt" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Termin Doppelblatt" .OnAction = "a_doppelblatt" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Termin Einzelblatt" .OnAction = "a_einzelblatt" .Style = msoButtonCaption End With Set oBtn = oPopUp.Controls.Add With oBtn .Caption = "Datei löschen" .OnAction = "Datei_platt_machen" .Style = msoButtonCaption End With End Sub Private Sub Workbook_BeforeClose(Cancel As Boolean) Call CmdDelete End Sub StandardModule: basMain Sub CmdDelete() On Error GoTo ERRORHANDLER Application.CommandBars("Worksheet Menu Bar") _ .Controls("Projektstatus").Delete ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Menü nur in der zugehörigen Arbeitsmappe sichtbar

Problem:	Wie erreiche ich, daß ein zu einer bestimmten Arbeitsmappe gehöriges Menü nur dann sichtbar ist, wenn diese Arbeitsmappe die aktive ist?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) Call Zurueck End Sub Private Sub Workbook_Open() Dim oPopUp As CommandBarPopup Call Zurueck Set oPopUp = Application.CommandBars(_ "Worksheet Menu Bar").Controls.Add(msocontrolpopup) oPopUp.Caption = "MeinMenue" oPopUp.Visible = True End Sub Private Sub Workbook_WindowActivate(ByVal Wn As Excel.Window) On Error Resume Next Application.CommandBars(1).Controls("MeinMenue").Visible = True End Sub Private Sub Workbook_WindowDeactivate(ByVal Wn As Excel.Window) On Error Resume Next Application.CommandBars(1).Controls("MeinMenue").Visible = False End Sub StandardModule: basMain Sub Zurueck() On Error GoTo ERRORHANDLER Application.CommandBars("Worksheet Menu Bar") _ .Controls("MeinMenue").Delete ERRORHANDLER: End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Menü hinzufügen

Problem: Wie kann ich der aktiven Menüleiste ein neues Menü mit einem Menüpunkt hinzufügen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CommandButton1_Click()
    Call NewMenu
End Sub

Private Sub CommandButton2_Click()
    Call Zurueck
End Sub

ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Call Zurueck
End Sub

StandardModule: basMain

Sub NewMenu()
    Dim oBar As CommandBar
    Dim oPopUp As CommandBarPopup
    Dim oBtn As CommandBarControl
    Set oBar = Application.CommandBars("Worksheet Menu Bar")
    Call Zurueck
    Set oPopUp = oBar.Controls.Add( _
        Type:=msoControlPopup, _
        Temporary:=False)
    oPopUp.Caption = "MeinMenü"
    Set oBtn = oPopUp.Controls.Add
    With oBtn
        .Caption = "Import"
        .TooltipText = "Import"
        .Style = msoButtonCaption
    End With
End Sub

Sub Zurueck()
    On Error Resume Next
    Application.CommandBars( _
        "Worksheet Menu Bar").Controls("MeinMenü").Delete
    On Error GoTo 0
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Abfrage, ob ein Menü in einer Menüleiste vorhanden ist

Problem: Wie kann ich über VBA abfragen, ob ein bestimmtes Menü in einer bestimmten Menüleiste vorhanden ist?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub IstVorhanden()
    Dim oPopUp As CommandBarControl
    Dim sPopUp As String
    Dim bln As Boolean
    sPopUp = "E&xtras"
    For Each oPopUp In Application.CommandBars _
        ("Worksheet Menu Bar").Controls
        If oPopUp.Caption = sPopUp Then
            bln = True
            Exit For
        End If
    Next oPopUp
    If bln = True Then
        MsgBox "Menü " & sPopUp & " gefunden"
    Else
        MsgBox "Menü " & sPopUp & " nicht gefunden"
    End If
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufruf des Symbolleisten-Anpassen-Dialogs verhindern

Problem: Beim Öffnen der Arbeitsmappe soll sowohl das Doppelklicken wie auch der Rechtsklick im Symbolleistenbereich zum Aufruf des Anpassen-Dialogs unterbunden werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Private Sub Workbook_BeforeClose(Cancel As Boolean)
    With Application
        .OnDoubleClick = ""
        .CommandBars("Toolbar List").Enabled = True
    End With
End Sub

Private Sub Workbook_Open()
    With Application
        .OnDoubleClick = "Ausschalten"
        .CommandBars("Toolbar List").Enabled = False
    End With
End Sub

StandardModule: Modul1

Sub Ausschalten()
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe nur über Schaltfläche schließen

Problem:	Wie kann ich verhindern, dass eine Arbeitsmappe über Menü- oder Symbolleistenschaltflächen geschlossen werden kann?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_Open() Worksheets("Tabelle1").cmdSchliessen.PrintObject = True Call AusEinSchalten(False) End Sub Private Sub Workbook_BeforeClose(Cancel As Boolean) If Worksheets("Tabelle1") _ .cmdSchliessen.PrintObject = True Then Cancel = True Exit Sub End If Call AusEinSchalten(True) End Sub StandardModule: basMain Sub NoAction() End Sub Sub AusEinSchalten(blnSchalter As Boolean) With Application.CommandBars("Worksheet Menu Bar") .Controls("E&xtras").Controls("&Anpassen...").Enabled = blnSchalter .Controls("&Ansicht").Controls("&Symbolleisten").Enabled = blnSchalter End With Application.CommandBars("Toolbar List").Enabled = blnSchalter If blnSchalter Then Application.OnDoubleClick = "" Else Application.OnDoubleClick = "NoAction" End If End Sub ClassModule: Tabelle1 Private Sub cmdSchliessen_Click() cmdSchliessen.PrintObject = False ThisWorkbook.Close savechanges:=False End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benennung und Wert von Umgebungsvariablen ermitteln

Problem: Die benutzerdefinierten Funktionen in den Zellen B1 und B2 sollen die Benennung und den Wert der jeweiligen Umgebungsvariablen ermitteln.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: Modul1

```
Function Umgebung(bln As Boolean)
    Dim sTxt As String, sFirst As String, sSecond As String
    sTxt = Environ(27)
    sFirst = Left(sTxt, InStr(sTxt, "=") - 1)
    sSecond = Right(sTxt, Len(sTxt) - InStr(sTxt, "="))
    If bln Then
        Umgebung = sFirst
    Else
        Umgebung = sSecond
    End If
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Im Assistenten anzuzeigende Funktion festlegen

Problem: Über UserForm-OptionButtons soll festgelegt werden, welche von 2 Funktionen in der Kategorie Benutzerdefiniert des Funktionsassistenten angezeigt wird.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
StandardModule: basMain

Private Sub AddCustomProperties()
    ThisWorkbook.CustomDocumentProperties.Add _
        Name:="Functions", _
        LinkToContent:=False, _
        Type:=msoPropertyTypeBoolean, _
        Value:=True
End Sub

Sub DialogAufruf()
    Dim iRow As Integer
    frmFunctions.Show
    With ThisWorkbook.VBProject.VBComponents("basFunctions").CodeModule
        If ThisWorkbook.CustomDocumentProperties("Functions") = True Then
            iRow = .ProcBodyLine("HoleWert", 0)
            .DeleteLines iRow
            .InsertLines iRow, "Function HoleWert()"
            iRow = .ProcBodyLine("GetValue", 0)
            .DeleteLines iRow
            .InsertLines iRow, "Private Function GetValue()"
        Else
            iRow = .ProcBodyLine("GetValue", 0)
            .DeleteLines iRow
            .InsertLines iRow, "Function GetValue()"
            iRow = .ProcBodyLine("HoleWert", 0)
            .DeleteLines iRow
            .InsertLines iRow, "Private Function HoleWert()"
        End If
    End With
End Sub

ClassModule: frmFunctions

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    ThisWorkbook.CustomDocumentProperties("Functions").Value = _
        OptionButton1.Value
    Unload Me
End Sub

StandardModule: basFunctions

Private Function HoleWert()
    HoleWert = 10
End Function

Function GetValue()
    GetValue = 12
End Function
```



Eigene Funktionskategorien erstellen

Problem:	Wie kann ich eigene Funktionskategorien erstellen
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	StandardModule: basMain Die nachfolgende Lösung stammt von Laurent Longre: Const Lib = ""c:\windows\system\user32.dll"" Option Base 1 Private Function Multiply(N1 As Double, N2 As Double) As Double Multiply = N1 * N2 End Function ===== Private Function Divide(N1 As Double, N2 As Double) As Double Divide = N1 / N2 End Function ===== Sub Auto_open() Register "DIVIDE", 3, "Numerator,Divisor", 1, "Division", _ "Divides two numbers", ""Numerator","Divisor """, "CharPrevA" Register "MULTIPLY", 3, "Number1,Number2", 1, "Multiplication", _ "Multiplies two numbers", ""First number","Second number """, _ "CharNextA" End Sub ===== Sub Register(FunctionName As String, NbArgs As Integer, _ Args As String, MacroType As Integer, Category As String, _ Descr As String, DescrArgs As String, FLib As String) Application.ExecuteExcel4Macro _ "REGISTER(" & Lib & ", "" & FLib & """, "" & String(NbArgs, "P") _ & """, "" & FunctionName & """, "" & Args & """, " & MacroType _ & ", "" & Category & """,,, "" & Descr & """, " & DescrArgs & ")" End Sub ===== Sub Auto_close() Dim FName, FLib Dim I As Integer FName = Array("DIVIDE", "MULTIPLY") FLib = Array("CharPrevA", "CharNextA") For I = 1 To 2 With Application .ExecuteExcel4Macro "UNREGISTER(" & FName(I) & ")" .ExecuteExcel4Macro "REGISTER(" & Lib & _ ", ""CharPrevA"", ""P"", "" & FName(I) & """,,,0)" .ExecuteExcel4Macro "UNREGISTER(" & FName(I) & ")" End With Next End Sub



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Funktion ANZAHL2 ohne ausgeblendete Spalten

Problem:	Wie kann ich die Excel-Funktion ANZAHL2 anwenden, ohne daß die ausgeblendeten Spalten mitgezählt werden?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function CountC(iRow As Integer) As Integer Dim iCol As Integer, iCounter As Integer For iCol = 1 To 256 If Application.Caller.Parent _ .Cells(iRow, iCol).Text <> "" Then iCounter = iCounter + 1 End If Next iCol CountC = iCounter End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Buchstabenindex des Arbeitsblattes mit der Fundzelle

Problem:	Die benutzerdefinierte Funktion in Spalte F soll einen Buchstabenindex für das Blatt liefern, in dem der Name aus Spalte A gefunden wird.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function GetGroup(sName As String) As String Dim vRow As Variant Dim iWks As Integer, iCounter As Integer iCounter = Application.Caller.Parent.Index + 1 For iWks = iCounter To Worksheets.Count vRow = Application.Match(sName, Worksheets(iWks).Columns(1), 0) If Not IsError(vRow) Then GetGroup = Chr(65 + iWks - iCounter) Exit Function End If Next iWks GetGroup = "Nicht gefunden" End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblattnamen in Zelle eintragen

Problem: Wie kann ich den Namen des jeweiligen Arbeitsblattes in eine Zelle eintragen lassen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain  
  
Function Blatt() As String  
    Blatt = Application.Caller.Parent.Name  
End Function
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Name des vorhergehenden Blattes

Problem: Wie kann ich den Namen des der aktiven Tabelle vorhergehenden Tabellenblattes ermitteln?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function PrevSheet()
    If Application.Caller.Parent.Index = 1 Then
        PrevSheet = "Sie befinden sich im ersten Blatt!"
    Else
        PrevSheet = Worksheets( _
            Application.Caller.Parent.Index - 1).Name
    End If
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Pfad gem. Pfadhierarchie ermitteln

Problem: Die benutzerdefinierte Funktion liefert das dem Verzeichnis dieser Arbeitsmappe übergeordnete Verzeichnis gem. Parameter.

Lösung: Geben Sie die benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: Modul1

Function GetParentDir(iDir As Integer) As String
    Dim iChar As Integer, iAct As Integer
    Dim sDir As String
    sDir = ThisWorkbook.Path
    For iAct = 1 To iDir
        iChar = Len(sDir)
        Do Until Mid(sDir, iChar, 1) = "\"
            iChar = iChar - 1
        Loop
        sDir = Left(sDir, iChar - 1)
    Next iAct
    GetParentDir = sDir
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappen-Titel über benutzerdefinierte Funktion

Problem:	Wie kann ich mir den Titel der aktiven Arbeitsmappe über eine benutzerdefinierte Funktion anzeigen lassen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Public Function WbTitel() As String WbTitel = ActiveWorkbook.BuiltinDocumentProperties(1) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zugriff auf geschlossene Arbeitsmappen

Problem: Weder über VBA noch mit der Excel-Funktion INDIREKT() kann ich auf geschlossene Arbeitsmappen zugreifen. Läßt sich dieses Problem umgehen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function CLink(strLink As String)
    CLink = ExecuteExcel4Macro(strLink)
End Function

Sub Aufruf()
    Dim sTxt As String
    sTxt = Range("C4").Text
    MsgBox CLink(sTxt)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Den am häufigsten vorkommenden Wert ermitteln

Problem: Die benutzerdefinierte Funktion in Zelle N1 liefert den im Wertebereich am häufigst vorkommenden Wert.

Lösung: Geben Sie die benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: Modul1

```
Function MostValue(rngAll As Range) As Integer
    Dim col As New Collection
    Dim rng As Range
    Dim dValue As Double
    Dim iCounter As Integer
    Dim iMost As Integer
    On Error Resume Next
    For Each rng In rngAll.Cells
        col.Add rng.Value, CStr(rng.Value)
    Next rng
    On Error GoTo 0
    For iCounter = 1 To col.Count
        If WorksheetFunction.CountIf(rngAll, col(iCounter)) > iMost Then
            iMost = WorksheetFunction.CountIf(rngAll, col(iCounter))
            dValue = col(iCounter)
        End If
    Next iCounter
    MostValue = dValue
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bestimmung der Zeilenhöhe

Problem: Durch die benutzerdefinierte Funktion in Spalte A soll die jeweilige Zeilenhöhe bestimmt werden.

Lösung: Geben Sie die benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: Modul1  
  
Function ZeilenHoehe(iRow As Integer) As Double  
    ZeilenHoehe = Rows(iRow).RowHeight  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benutzerdefinierte Funktion zur Ermittlung hinterlegter Kategorien

Problem: Die benutzerdefinierte Funktion in Spalte C liefert die in Tabelle2 hinterlegten Kategorien.

Lösung: Geben Sie die benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: Modul1

```
Function Zuordnung(sTxt As String, rng As Range) As String
    Dim wks As Worksheet
    Dim iRow As Integer
    Set wks = Worksheets("Tabelle2")
    iRow = 1
    Do Until IsEmpty(wks.Cells(iRow, 1))
        If InStr(sTxt, wks.Cells(iRow, 1).Value) Then
            Zuordnung = wks.Cells(iRow, 2).Value
            Exit Function
        End If
        iRow = iRow + 1
    Loop
    Zuordnung = "#NV"
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Telefonnummern den Telefonierern zuordnen

Problem: Über eine benutzerdefinierte Funktion sollen zu jeder Telefonnummer die im Blatt Data hinterlegten Telefonierer ermittelt werden. Jedem Telefonierer ist ein fester Kreis von Telefonnummern zugeordnet.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: Modul1

Function Telefon(sNo As String) As String
    Dim wks As Worksheet
    Dim vRow As Variant
    Dim iCol As Integer
    Set wks = Worksheets("Data")
    iCol = 1
    Do Until IsEmpty(wks.Cells(1, iCol))
        vRow = Application.Match(sNo, wks.Columns(iCol), 0)
        If Not IsError(vRow) Then
            Telefon = wks.Cells(1, iCol).Value
            Exit Function
        End If
        iCol = iCol + 1
    Loop
    Telefon = "Unbekannt"
End Function
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Fundwerte ermitteln

Problem:	Die benutzerdefinierte Funktion sucht je nach Parameter den ersten oder den letzten gesuchten Wert, die Fundadresse, Fundzeile oder Fundspalte.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function GetValue(rng As Range, vValue As Variant, bln As Boolean, iArt As Integer) As Variant Dim vFound As Variant Dim iCounter As Integer For iCounter = 1 To rng.Cells.Count If rng(iCounter).Value = vValue Then If iArt = 0 Then vFound = rng(iCounter).Address(False, False) ElseIf iArt = 1 Then vFound = rng(iCounter).Row Else vFound = rng(iCounter).Column End If If Not IsEmpty(vFound) And bln = False Then GetValue = vFound Exit Function End If End If Next iCounter If Not IsEmpty(vFound) Then GetValue = vFound End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vergleich von zwei Listen und Ausgabe eines Wahrheitswertes

Problem:	Die benutzerdefinierte Funktion in Zelle C1 soll prüfen, ob alle Werte im Bereich A in Bereich B enthalten sind.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function IstKomplett(rngA As Range, rngB As Range) As Boolean Dim rng As Range Dim var As Variant Dim bln As Boolean For Each rng In rngA var = Application.Match(rng.Value, rngB, 0) If IsError(var) Then bln = True Exit Function End If Next rng IstKomplett = Not bln End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Über benutzerdefinierte Funktion Filterkriterium anzeigen

Problem: Über die benutzerdefinierte Funktion in Zelle G2 soll das Filterkriterium aus Spalte A angezeigt werden.

Lösung: Über die nachfolgende benutzerdefinierte Funktion.

VBA-Code: StandardModule: Modul1

```
Function Kriterium()  
    Kriterium = Worksheets("Tabelle1").AutoFilter.Filters(1).Criteria1  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl von Zellen aus einem nicht rechteckigen Bereich ermitteln

Problem: Über eine benutzerdefinierte Funktion soll die Anzahl der Zellen in einem nicht rechteckig umrahmten Bereich ermittelt werden.

Lösung: Über die nachfolgende benutzerdefinierte Funktion.

VBA-Code: StandardModule: Modul1

```
Function CellsCount(rng As Range) As Integer
    Dim lRow As Long
    Dim iCol As Integer, iCounter As Integer
    For lRow = rng.Row To ActiveSheet.Rows.Count
        For iCol = rng.Column To ActiveSheet.Columns.Count
            iCounter = iCounter + 1
            If Cells(lRow, iCol).Borders(xlEdgeRight).Weight = xlMedium Then
                Exit For
            End If
        Next iCol
        iCol = rng.Column
        If Cells(lRow, iCol).Borders(xlEdgeBottom).Weight = xlMedium Then
            Exit For
        End If
    Next lRow
    CellsCount = iCounter
End Function

Sub Test()
    MsgBox Range("H30").Borders(xlEdgeRight).Weight
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zuschläge gem. Tabelle berechnen

Problem:	Es sollen Zuschläge gem. vorgegebener Zuschlagstabelle errechnet werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Zuschlag(dValue As Double) As Double Dim dAct As Double Dim iValue As Integer dAct = dValue Mod 500000 Select Case dAct Case Is < 10000: iValue = 20 Case Is < 25000: iValue = 40 Case Is < 50000: iValue = 80 Case Is < 150000: iValue = 165 Case Is < 300000: iValue = 315 Case Is < 500000: iValue = 540 End Select iValue = iValue + _ ((dValue - (dValue Mod 500000)) / 500000) * 300 Zuschlag = iValue End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Den ersten Wert einer gefilterten Spalte ermitteln

Problem:	Der erste sichtbare Wert aus einer gefilterten Spalte soll angezeigt werden. Die Spalte wird als Parameter vorgegeben.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function FilterValue(iCol As Integer) Dim iRow As Integer Application.Volatile iRow = 2 Do While Rows(iRow).Hidden = True iRow = iRow + 1 Loop FilterValue = Cells(iRow, iCol).Value End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bestimmung der ersten und letzten Zelle eines Filters

Problem: Eine benutzerdefinierte Funktion soll die erste und letzte sichtbare Zeile einer gefilterten Tabelle ermitteln. Der zweite Parameter steht für erste oder letzte Zeile.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function FirstLast(rng As Range, bln As Boolean) As Integer
    Dim iCounter As Integer
    If bln Then
        For iCounter = rng.Row + 1 To rng.Row + _
            rng.Rows.Count - 1
            If Rows(iCounter).Hidden = False Then Exit For
        Next iCounter
    Else
        For iCounter = rng.Row + rng.Rows.Count - 1 _
            To rng.Row + 1 Step -1
            If Rows(iCounter).Hidden = False Then Exit For
        Next iCounter
    End If
    FirstLast = iCounter
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Preisvergleiche

Problem: Es sollen Preisvergleiche nach verschiedenen Kriterien und Ausgabemodi durchgeführt werden.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function Preis(rngAll As Range, sArtikel As String, _
    blnArt As Boolean, blnPreis As Boolean)
    Dim rng As Range
    Dim dValue As Double
    Dim iRow As Integer
    If blnPreis = False Then dValue = _
        WorksheetFunction.Max(rngAll.Columns(rngAll.Columns.Count))
    For Each rng In rngAll.Columns(2).Cells
        If rng.Value = sArtikel Then
            If blnPreis Then
                If rng.Offset(0, 1) > dValue Then
                    dValue = rng.Offset(0, 1).Value
                    iRow = rng.Row
                End If
            Else
                If rng.Offset(0, 1) < dValue Then
                    dValue = rng.Offset(0, 1).Value
                    iRow = rng.Row
                End If
            End If
        End If
    Next rng
    If blnArt Then
        Preis = dValue
    Else
        Preis = rngAll(iRow, 1).Value
    End If
End Function

Sub Aufruf()
    MsgBox Preis(Range("A1:C20"), "Artikel5", True)
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bereiche auf Übereinstimmung vergleichen

Problem:	Wie kann ich feststellen, ob zwei Bereiche identische Werte haben?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function BereichVergleich(rngEins, rngZwei) As Boolean Dim rngCell As Range Dim iCounter As Integer Dim bln As Boolean For Each rngCell In rngEins iCounter = iCounter + 1 If rngCell <> rngZwei.Cells(iCounter) Then bln = True Exit For End If Next rngCell If bln = False Then BereichVergleich = True End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Letzter Wert aus einem Bereich

Problem:	Wie kann ich in einer benutzerdefinierten Funktion den Wert der äußersten rechten, unteren Zelle eines Bereiches ermitteln?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function LastValue(MyArray As Range) As Variant With MyArray LastValue = .Cells(.Cells.Count).Value End With End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorkommen von Zahlenreihen in Spalten

Problem:	Wie kann ich feststellen, in wieviel Spalten bestimmte Zahlenreihen erscheinen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	StandardModule: basMain

```
Function HowMuch(rng As Range, ParamArray Data()) As Integer
    Dim vTest As Variant
    Dim iCol As Integer, iCounter As Integer
    Dim iCount As Integer
    Dim bln As Boolean
    For iCol = 1 To rng.Columns.Count
        For iCounter = 0 To UBound(Data)
            vTest = Application.Match(Data(iCounter), _
                rng.Columns(iCol), 0)
            If IsError(vTest) Then
                bln = True
                Exit For
            End If
        Next iCounter
        If bln = False Then
            iCount = iCount + 1
        Else
            bln = False
        End If
    Next iCol
    HowMuch = iCount
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Letzter Wert in Zeile

Problem:	Wie ermittle ich den letzten Text-Wert im Bereich A1:A500?
Lösung:	Folgende Matrixformel: =INDEX(\$A\$1:\$A\$500;MAX(WENN(ISTTEXT(\$A\$1:\$A\$500);ZEILE(\$A\$1:\$A\$500)));1)
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wahrheitsprüfung einer mit Zellwerten zusammengesetzten Formel

Problem:	Wie kann ich ein in einer Zelle stehendes Ergebnis einer durch Zellwerte zusammengesetzte Formel auf seinen Wahrheitswert prüfen? In Zelle A1 steht "3", in Zelle B1 "+", in Zelle C1 "5", in Zelle D1 "=" und in Zelle E1 "8". In Zelle F1 soll als Wahrheitsprüfung WAHR angezeigt werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Pruefung(_ sA As String, _ sB As String, _ sC As String, _ sD As String, _ sE As String) As Boolean Dim sFormula As String sFormula = "=" & sA & sB & sC & sD & sE Pruefung = Evaluate(sFormula) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Feststellen, welche Datei die ältere ist

Problem:	Die benutzerdefinierte Funktion in Zelle C2 soll den Namen der zuletzt gespeicherten Datei aus dem Bereich B1:B2 liefern.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function LastFile(sFileA As String, sFileB As String) As String Dim datLast As Date datLast = FileDateTime(sFileA) If datLast < FileDateTime(sFileB) Then LastFile = sFileB ElseIf datLast = FileDateTime(sFileB) Then LastFile = "Beide Dateien gespeichert am " & FileDateTime(sFileB) Else LastFile = sFileA End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeitspann in Zeitgruppen aufteilen

Problem: Der im Bereich B2:C3 genannte Zeitraum soll auf die Stunden gem. Zeile 1 aufgeteilt werden.

Lösung: Darstellung nur anhand einer Beispielarbeitsmappe möglich.

VBA-Code: StandardModule: Modul1

```
Function SplitTime( _  
    datDStart As Date, _  
    datTStart As Date, _  
    datDEnd As Date, _  
    datTEnd As Date) As Integer  
    Dim arr(1 To 5) As Integer  
    Dim datStart As Date, datEnd As Date  
    Dim lHour As Long  
    datStart = datDStart + datTStart  
    datEnd = datDEnd + datTEnd  
    For lHour = datStart * 24 To (datEnd * 24) - 1  
        Select Case lHour Mod 24  
            Case Is < Range("D1").Value: arr(1) = arr(1) + 1  
            Case Is < Range("E1").Value: arr(2) = arr(2) + 1  
            Case Is < Range("F1").Value: arr(3) = arr(3) + 1  
            Case Is < Range("G1").Value: arr(4) = arr(4) + 1  
            Case Is < Range("H1").Value: arr(5) = arr(5) + 1  
        End Select  
    Next lHour  
    SplitTime = arr(Application.Caller.Column - 3)  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorkommen eines Namens ab Zeitpunkt und in Gebiet ermitteln

Problem: Über eine benutzerdefinierte Funktion soll ermittelt werden, wie oft ein Name ab einem bestimmten Datum in einem vorgegebenen Gebiet vorkommt.

Lösung: Über die nachfolgende benutzerdefinierte Funktion.

VBA-Code: StandardModule: Modul1

```
Function CountNames(Bereich As Range, Start As Date, Gebiet As String)
    Dim rng As Range
    Dim col As New Collection
    Dim iCounter As Integer
    On Error Resume Next
    For Each rng In Bereich.Columns(1).Cells
        If rng.Value > Start And rng.Offset(0, 2) = Gebiet Then
            col.Add rng.Offset(0, 1).Value, rng.Offset(0, 1).Value
        End If
    Next rng
    On Error GoTo 0
    CountNames = col.Count
End Function
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung der jeweiligen Arbeitsschicht

Problem:	Wie kann ich Schichten nach verschiedenen Kriterien ermitteln?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	3. Woche: Mo+Di, Fr+Sa = Frühschicht Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein. StandardModule: basMain <pre>Function Schicht(datStart As Date, datAct As Date) As String Dim datWeek As Date datWeek = (datAct - datStart) Mod 21 Select Case datWeek Case Is < 7 If WeekDay(datAct) < 6 And WeekDay(datAct) > 1 Then Schicht = "Frühschicht" End If Case Is < 14 If WeekDay(datAct) > 3 And WeekDay(datAct) < 7 Then Schicht = "Spätschicht" End If Case Else If WeekDay(datAct) = 2 Or WeekDay(datAct) = 3 Then Schicht = "Spätschicht" ElseIf WeekDay(datAct) = 6 Or WeekDay(datAct) = 7 Then Schicht = "Frühschicht" End If End Select End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Online-Kosten anhand von Start- und Endzeit ermitteln

Problem:	Aus den Daten in Zeile 2 sollen Online-Kosten errechnet werden. Der Wochenendtarif liegt bei der Hälfte des Onlinetarifes.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Kosten(_ datDay As Date, _ datStart As Date, _ datEnd As Date, _ dblTarif As Double) As Double If WorksheetFunction.WeekDay(datDay, 2) < 6 Then Kosten = (datEnd - datStart) * 24 * 60 * dblTarif Else Kosten = ((datEnd - datStart) * 24 * 60 * dblTarif / 2) End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung des ersten Tages einer Kalenderwoche nach DIN 1355

Problem: Wie kann ich den ersten Tag einer beliebigen Kalenderwoche in einem beliebigen Jahr ermitteln.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: Modul1

Private Function DINWeek(dat As Date) As Integer
    Dim dValue As Double
    dValue = DateSerial(Year(dat + (8 - WeekDay(dat)) Mod 7 - 3), 1, 1)
    DINWeek = (dat - dValue - 3 + (WeekDay(dValue) + 1) Mod 7) \ 7 + 1
End Function

Function DINDay(iYear As Integer, iDIN As Integer)
    Dim iDay As Integer, iWeek As Integer
    If iYear = 0 Then
        DINDay = 0
        Exit Function
    End If
    iDay = 1
    iWeek = DINWeek(DateSerial(iYear, 1, 1))
    If iWeek <> 1 Then
        Do Until DINWeek(DateSerial(iYear, 1, iDay)) = 1
            iDay = iDay + 1
        Loop
    Else
        Do Until DINWeek(DateSerial(iYear, 1, iDay)) <> 1
            iDay = iDay - 1
        Loop
        iDay = iDay + 1
    End If
    DINDay = DateSerial(iYear, 1, iDay) + (iDIN - 1) * 7
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

5-Längen-Zeichenfolgen in Zeit mit Format mss,00 umwandeln

Problem: Wie kann ich eine Zeichenfolge wie "21307" in eine Zeit mit dem Format m:ss,00 (2:13,07) umwandeln. Mit dieser Zeit soll weitergerechnet werden können.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function sTime(dTime As Double) As String
    sTime = Left(dTime, 1) & ":" & _
        Mid(dTime, 2, 2) & "," & Right(dTime, 2)
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeitdifferenz in Schulstunden (45 Minuten) errechnen

Problem:	Wie kann ich die Differenz zwischen einer Start- und einer Endzeit in Schulstunden - 45 Minuten - errechnen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function ScoolHours(datStart As Date, datEnd As Date) As Double Dim dHours As Double dHours = (datEnd - datStart) * 24 / 3 * 4 ScoolHours = dHours End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Werte eines Wochentages aus Monatsblättern addieren

Problem: In einer Arbeitsmappe habe ich 12 Monatsblätter. In Spalte A ist das Datum eingetragen, in Spalte B der Wochentag und in Spalte C der zugehörige Wert. Wie kann ich mir in einem Analyseblatt die Werte aller Montage aus den Monatsblättern anzeigen lassen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function SumIf3D(intShStart As Integer, intShEnd As Integer, _  
    varCriteria) As Double  
    Dim intCounter As Integer  
    Dim dblSum As Double  
    For intCounter = intShStart To intShEnd  
        With Worksheets(intCounter)  
            dblSum = dblSum + WorksheetFunction.SumIf(.Columns(2), _  
                varCriteria, .Columns(3))  
        End With  
    Next intCounter  
    SumIf3D = dblSum  
End Function
```

```
Sub JahrAnlegen()  
    Dim i As Integer, y As Integer, intCounter As Integer  
    For i = 2 To 13  
        Worksheets.Add after:=Worksheets(Worksheets.Count)  
        ActiveSheet.Name = Format(DateSerial(1, i - 1, 1), "mmmm")  
        Columns(1).NumberFormat = "dd.mm.yy"  
        Columns(2).NumberFormat = "dddd"  
        For y = 1 To Day(DateSerial(1999, i, 0))  
            intCounter = intCounter + 1  
            Cells(y, 1) = DateSerial(1999, i - 1, y)  
            Cells(y, 2) = WeekDay(Cells(y, 1))  
            Cells(y, 3) = intCounter  
        Next y  
    Next i  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitstage mit Samstagen

Problem: Bei den Analysefunktionen gibt es die Formel =NETTOARBEITSTAGE(). Hier werden Feiertage, Samstage und Sonntage herausgerechnet. Ich benötige eine Funktion, die das Gleiche macht, allerdings die Samstage inkludiert.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function ATC(datStart As Date, datEnd As Date, _  
    rngHolliday As Object) As Integer  
    Dim rng As Range  
    Dim iCounter As Integer, iStart As Integer  
    Dim iEnd As Integer, iTmp As Integer  
    iStart = 8 - WeekDay(datStart)  
    iEnd = WeekDay(datEnd) - 1  
    iTmp = (datEnd - datStart) - (iStart + iEnd)  
    iTmp = iTmp - (iTmp / 7)  
    iTmp = iTmp + iStart + iEnd  
    If WeekDay(datStart) = 1 Then iTmp = iTmp - 1  
    For iCounter = 1 To rngHolliday.Rows.Count  
        Set rng = rngHolliday.Cells(iCounter, 1)  
        If rng >= datStart And rng <= datEnd Then  
            If WeekDay(rng) <> 1 Then iTmp = iTmp - 1  
        End If  
    Next iCounter  
    ATC = iTmp  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Datumsdifferenz in der Form "12 Jahre, 3 Monate, 4 Tage"

Problem:	Wie kann eine Datumsdifferenz mit Angabe von Jahren, Monaten und Tagen aufgerufen werden?
Lösung:	Folgende Funktion: =DATEDIF(A2;B2;""y"")&"" Jahr(e), """ DATEDIF(A2;B2;""ym"")&"" Monat(e), """ DATEDIF(A2;B2;""md"")&"" Tag(e)"
VBA-Code:	<pre>StandardModule: basMain Function DatDiff(datStart As Date, datEnd As Date) As String Dim intYear As Integer, intMonth As Integer Dim intDay As Integer, intYD As Integer Dim intMD As Integer, intTD As Integer intYear = Year(datStart) intMonth = Month(datStart) intYD = Year(datEnd) - intYear intMD = Month(datEnd) - intMonth intTD = Day(datEnd) - Day(datStart) If intTD < 0 Then intMD = intMD - 1 intTD = intTD + Day(DateSerial(intYear, intMonth + 1, 0)) End If If intMD < 0 Then intYD = intYD - 1 intMD = intMD + 12 End If DatDiff = intYD & " Jahr(e), " & intMD _ & " Monat(e), " & intTD & " Tag(e)" End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wieviele Wochen hat ein Monat?

Problem:	Wie ermittle ich die Wochen eines Monats? Als Woche zählen mindestens 4 Tage am Anfang und am Ende des Monats.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Weeks(iYear As Integer, iMonth As Integer) Dim iDay As Integer, iCounter As Integer, iWeek As Integer, iPart As Integer Dim lDay As Long Dim bln As Boolean For lDay = DateSerial(iYear, iMonth, 1) To DateSerial(iYear, iMonth + 1, 0) iPart = iPart + 1 If WeekDay(lDay) = 1 Then If iPart >= 4 Then iWeek = iWeek + 1 iPart = 0 End If Next lDay If iPart >= 4 Then iWeek = iWeek + 1 Weeks = iWeek End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsstunden in einem vorgegebenen Zeitraum

Problem:	Wie kann ich die Arbeitsstunden in einem vorgegebenen Zeitraum ermitteln? Anfangs-, End- und Pausenzeiten sind vorgegeben.
Lösung:	Nur anhand einer Beispielarbeitsmappe darstellbar.
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Adresse bei mehrfach vorkommenden Namen zuordnen

Problem: Die benutzerdefinierte Funktion B4 prüft, ob in Blatt "Adressen" sowohl der Name wie auch die PLZ übereinstimmen und liefert den zutreffenden Ort.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: Modul1

Function GetAddress(sName As String, Optional sPlz As String)
    Dim lRow As Long
    Dim iCount As Integer
    iCount = WorksheetFunction.CountIf(Worksheets("Adressen").Columns(1), sName)
    lRow = 1
    With Worksheets("Adressen")
        Do Until IsEmpty(.Cells(lRow, 1))
            If iCount > 1 Then
                If .Cells(lRow, 1).Value = sName And _
                    .Cells(lRow, 2).Value = sPlz Then
                    GetAddress = .Cells(lRow, 3).Value
                    Exit Function
                End If
            Else
                If .Cells(lRow, 1).Value = sName Then
                    GetAddress = .Cells(lRow, 3).Value
                    Exit Function
                End If
            End If
            lRow = lRow + 1
        Loop
    End With
    GetAddress = ""
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert eintragen, wenn Bedingte-Formatierung-Bedingung zutrifft

Problem:	Wenn die Bedingung der bedingten Formatierung in A1 zutrifft, soll in Tabelle2, Zelle B1 der Wert von Zelle A1 eingetragen werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function FormCond(rng As Range) As Variant Dim sFormula As String With rng.FormatConditions(1) sFormula = .Formula1 If Evaluate(sFormula) = True Then FormCond = rng.Value Else FormCond = "" End If End With End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Zellen mit einer vorgegebenen Hintergrundfarbe zählen

Problem: In einem anzugebenden Bereich sind alle Zellen mit einer vorgegebenen Hintergrundfarbe zu zählen.

Lösung: Über die nachfolgende benutzerdefinierte Funktion. Eingabesyntax: =countcolor(A1:C6;6)

VBA-Code:

```
StandardModule: Modul1

Function CountColor(rng As Range, iColor As Integer)
    Dim rngAct As Range
    Dim iCount As Integer
    For Each rngAct In rng.Cells
        If rngAct.Interior.ColorIndex = iColor And _
            Not IsEmpty(rngAct) Then
            iCount = iCount + 1
        End If
    Next rngAct
    CountColor = iCount
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Multiplikation in Abhängigkeit von der Schriftfarbe

Problem: Je nach Schriftfarbe sollen die Werte aus Spalte A mit einem anderen Faktor multipliziert werden.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function Schrift(rng As Range) As Double
    Select Case rng.Font.ColorIndex
        Case 3: Schrift = rng.Value * 2
        Case 5: Schrift = rng.Value * 3
        Case 10: Schrift = rng.Value * 4
    End Select
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nur sichtbare Zellen addieren

Problem:	Wie kann ich von einem vorgegebenen Bereich nur die sichtbaren Zellen summieren lassen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function NettoSumme(rngAll As Object) Dim rngAct As Range Dim dValue As Double For Each rngAct In rngAll.Cells If Rows(rngAct.Row).Hidden = False Then dValue = dValue + rngAct.Value End If Next rngAct NettoSumme = dValue End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summe aller kursiv geschriebenen Werte ermitteln

Problem:	Wie kann ich die Summe aller Werte eines vorgegebenen Bereiches ermitteln, deren Schriftart auf kursiv gesetzt ist?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function SumItalic(rng As Range) As Double Dim rngAct As Range Dim dAdd As Double For Each rngAct In rng.Cells If IsNumeric(rngAct) Then If rngAct.Font.Italic = True Then dAdd = dAdd + rngAct.Value End If End If Next rngAct SumItalic = dAdd End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl eines Namens in einer bestimmten Schriftfarbe ermitteln

Problem: Wie kann ich aus einer Tabelle alle Namen "Meier" ermitteln, deren Schriftfarbe rot ist?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function NameFarbe(rng As Range, _
    iColor As Integer, sName As String) As Integer
    Dim rngAct As Range
    Dim iCount As Integer
    For Each rngAct In rng.Cells
        If rngAct.Value = sName And _
            rngAct.Font.ColorIndex = iColor Then
            iCount = iCount + 1
        End If
    Next rngAct
    NameFarbe = iCount
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte in Zellen mit vorgegebener Hintergrundfarbe addieren

Problem: Wie kann ich durch eine Formel all Werte einer Tabelle addieren, deren Zelle die gleiche Hintergrundfarbe hat, wie die Zelle mit der Formel?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function FarbenAddieren(rng As Range) As Double
    Dim rngAct As Range
    Dim dAdd As Double
    For Each rngAct In rng.Cells
        If rngAct.Interior.ColorIndex = _
            Application.Caller.Interior.ColorIndex Then
            dAdd = dAdd + rngAct.Value
        End If
    Next rngAct
    FarbenAddieren = dAdd
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl von Zellen mit einer bestimmten Hintergrundfarbe ermitteln

Problem: Wie kann ich die Anzahl der Zellen mit einer bestimmten Hintergrundfarbe ermitteln?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function Farbe(rng As Range, iColor As Integer) As Integer
    Dim iCounter As Integer
    Dim rngAct As Range
    For Each rngAct In rng
        If rngAct.Interior.ColorIndex = iColor Then
            iCounter = iCounter + 1
        End If
    Next rngAct
    Farbe = iCounter
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl der gelb hinterlegten Zellen ermitteln

Problem: Wie kann ich die Anzahl aller mit einer bestimmten Hintergrundfarbe formatierten und einen bestimmten Text beinhaltenden Zellen errechnen lassen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function WertUndFarbe(rng As Object, _
    iColor As Integer, sTxt As String) As Integer
    Dim rngZelle As Range
    Dim iCounter As Integer
    For Each rngZelle In rng.Cells
        If rngZelle.Interior.ColorIndex = iColor And _
            rngZelle.Value = sTxt Then _
            iCounter = iCounter + 1
    Next rngZelle
    WertUndFarbe = iCounter
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellen nach Schriftfarbe summieren

Problem:	Wie kann ich einen Bereich nach Schriftfarben summieren?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Sub CountIfColor() Dim rngAll As Range, rngAct As Range Dim dRed As Double, dBlue As Double, dGreen As Double Set rngAll = Range("A1:F100") For Each rngAct In rngAll.Cells Select Case rngAct.Font.ColorIndex Case Is = 3: dRed = dRed + rngAct.Value Case Is = 5: dBlue = dBlue + rngAct.Value Case Is = 4: dGreen = dGreen + rngAct.Value End Select Next rngAct Range("H1") = dRed Range("H2") = dBlue Range("H3") = dGreen End Sub Function cc(rngAll As Object) Dim rngAct As Range Dim dValue As Double Application.Volatile For Each rngAct In rngAll If rngAct.Font.ColorIndex = _ Application.Caller.Interior.ColorIndex Then dValue = dValue + rngAct.Value End If Next rngAct cc = dValue End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wenn "fett", dann addieren

Problem: Wie kann ich die Werte aller als "fett" formatierter Zellen eines Bereiches addieren?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function BoldAddieren(rngAll As Range) As Double
    Dim rng As Range
    Dim dValue As Double
    Application.Volatile
    For Each rng In rngAll.Cells
        If rng.Font.Bold And IsNumeric(rng.Value) Then
            dValue = dValue + rng.Value
        End If
    Next rng
    BoldAddieren = dValue
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung der Kopfzeile nach der Funktion KGRÖSSTE()

Problem: Wie kann ich die Kopfzeile der mit der Funktion KGRÖSSTE() ermittelten Rangfolge abrufen? Mehrfachtreffer müssen berücksichtigt werden

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function sRang(rngAll As Range, iPos As Integer)
    Dim wks As Worksheet
    Dim rng As Range
    Dim sTxt As String
    Set wks = Application.Caller.Parent
    For Each rng In rngAll.Cells
        If rng.Value = iPos Then
            sTxt = sTxt & Cells(1, rng.Column) & ", "
        End If
    Next rng
    sRang = Left(sTxt, Len(sTxt) - 2)
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benutzerdefinierte Funktion als Matrixformel

Problem: Aus Verpackungsabmessungen sollen über eine gemeinsame benutzerdefinierte Funktion Kubatur und qm Außenfläche errechnet werden.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: Modul1

Function CbmAndQm(dL As Double, dW As Double, dH As Double) As Variant
    Dim arr(1 To 2) As Double
    arr(1) = dL * dW * dH
    arr(2) = ((dL * dW) + (dL * dH) + (dW * dH)) * 2
    CbmAndQm = arr
End Function

Sub Aufruf()
    Dim var As Variant
    var = CbmAndQm(Range("A2").Value, Range("B2").Value, Range("C2").Value)
    Range("D2").Value = var(1)
    Range("E2").Value = var(2)
End Sub
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Minimalzeit je Athlet ermitteln

Problem:	Die benutzerdefinierten Funktionen in Spalte D sollen die Minimalzeiten je Athlet und Disziplin ermitteln.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function GetMinimum(_ sAthlet As String, _ sDisziplin As String, _ rng As Range) As Date Dim datTemp As Date Dim iRow As Integer Application.Volatile datTemp = WorksheetFunction.Max(rng) For iRow = 1 To rng.Rows.Count If Cells(iRow, 1).Value = sAthlet And _ Cells(iRow, 2).Value = sDisziplin Then If datTemp > Cells(iRow, 3).Value Then datTemp = Cells(iRow, 3).Value End If End If Next iRow GetMinimum = datTemp End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Maximalwert für Wochenenden und Feiertage ermitteln

Problem:	Es soll der Maximalwert einer Datenreihe ermittelt werden, bei dessen Werten in der Datumsspalte es sich um Wochenenden oder Feiertage handelt.
Lösung:	Über die nachfolgende benutzerdefinierte Funktion. Eingabesyntax: =specialmax(A1:A31;A40:A50;2)
VBA-Code:	<pre>StandardModule: Modul1 Function SpecialMax(rngA As Range, rngB As Range, iCol As Integer) Dim rng As Range Dim var As Variant Dim dValue As Double For Each rng In rngA.Cells var = Application.Match(CDbl(rng.Value), rngB, 0) If Not IsError(var) Or WorksheetFunction.Weekday(rng.Value, 2) > 5 Then If rng.Offset(0, 2).Value > dValue Then dValue = rng.Offset(0, iCol).Value End If End If Next rng SpecialMax = dValue End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Errechnung der Maximaltemperatur an bestimmtem Datum

Problem:	Es soll die Maximaltemperatur an einem bestimmten Datum errechnet werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Temperatur(dat As Date, rng As Range) Dim rngAct As Range Dim iStart As Integer, iEnd As Integer For Each rngAct In rng.Cells If rngAct = dat Then iStart = rngAct.Row iEnd = iStart Exit For End If Next rngAct Do iEnd = iEnd + 1 Loop While rng.Cells(iEnd - 1, 1) = rng.Cells(iEnd, 1) Temperatur = WorksheetFunction.Max(_ rng.Range(rng.Cells(iStart, 3), rng.Cells(iEnd, 3))) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Median ohne ausgeblendete Zellen berechnen

Problem: Die benutzerdefinierte Funktion in Zelle A17 berechnet den Merian der Zahlenreihe in A1:A16 ohne Berücksichtigung der ausgeblendeten Zellen.

Lösung: Geben Sie die benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: Modul1

```
Function MyMedian(rng As Range) As Double
    Dim rngAct As Range
    Dim arr() As Double
    Dim dAct As Double
    Dim iCountA As Integer, iCountB As Integer, iCount As Integer
    For Each rngAct In rng.Cells
        If Rows(rngAct.Row).Hidden = False Then
            iCountA = iCountA + 1
            ReDim Preserve arr(1 To iCountA)
            arr(iCountA) = rngAct.Value
        End If
    Next rngAct
    iCount = iCountA
    For iCountA = 1 To UBound(arr)
        For iCountB = iCountA + 1 To iCount
            If arr(iCountA) > arr(iCountB) Then
                dAct = arr(iCountA)
                arr(iCountA) = arr(iCountB)
                arr(iCountB) = dAct
            End If
        Next iCountB
    Next iCountA
    dAct = 0
    For iCountA = 1 To iCount
        If iCount Mod 2 = 0 Then
            dAct = dAct + arr(iCountA)
            If iCountA > 2 Then dAct = dAct - arr(iCountA - 2)
            If iCountA > iCount / 2 Then
                MyMedian = dAct / 2
                Exit Function
            End If
        Else
            dAct = arr(iCountA)
            If iCountA > iCount / 2 Then
                MyMedian = dAct
                Exit Function
            End If
        End If
    Next iCountA

End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Bedingter Mittelwert

Problem: Wie kann ich aus einer Wertetabelle den Mittelwert unter Berücksichtigung einer Ober- und Untergrenze errechnen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function IfAverage( _  
    rngAll As Object, _  
    dFirst As Double, _  
    dLast As Double) As Double  
    Dim rng As Range  
    Dim dSum As Double  
    Dim iCounter As Integer  
    For Each rng In rngAll.Cells  
        If IsNumeric(rng.Value) Then  
            If rng.Value >= dFirst And rng.Value <= dLast Then  
                dSum = dSum + rng.Value  
                iCounter = iCounter + 1  
            End If  
        End If  
    Next  
    If iCounter > 0 Then  
        IfAverage = dSum / iCounter  
    Else  
        IfAverage = 0  
    End If  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mittelwert ohne Nullwerte

Problem: Wie ermittle ich den Mittelwert aus einer Liste von Zahlen, ohne daß die Nullwerte berücksichtigt werden?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function AverageSZero(rngAct As Range) As Double
    Dim rng As Range
    Dim dSum As Double
    Dim iCounter As Integer
    For Each rng In rngAct.Cells
        If IsNumeric(rng.Value) Then
            If rng.Value > 0 Then
                dSum = dSum + rng.Value
                iCounter = iCounter + 1
            End If
        End If
    Next rng
    If iCounter > 0 Then
        AverageSZero = dSum / iCounter
    Else
        AverageSZero = 0
    End If
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

NETTOARBEITSTAGE-Funktion sprachunabhängig gestalten

Problem:	Die Nettoarbeitstage-Funktion aus dem Analysepack soll sprachunabhängig gestaltet werden. Das AddIn "Analysefunktionen - VBA" muss aktiviert sein.
Lösung:	Über die nachfolgende benutzerdefinierte Funktion.
VBA-Code:	<pre>StandardModule: Modul1 Function NetArbTage(datStart As Date, datEnd As Date, rngHolidays As Range) NetArbTage = networkdays(datStart, datEnd, rngHolidays) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nachbildung der Funktion NETTOARBEITSTAGE aus dem Analyse-Pack

Problem:	Die Funktion NETTOARBEITSTAGE aus dem Analyse-Pack soll als benutzerdefinierte Funktion (UDF) erstellt werden.
Lösung:	Über die nachfolgende benutzerdefinierte Funktion. Eingabesyntax: =atage(A1;B1;FreieTage!A1:A4)
VBA-Code:	<pre>StandardModule: Modul1 Function ATage(datStart As Date, datEnd As Date, rng As Range) Dim var As Variant Dim lDay As Long, lCount As Long For lDay = datStart To datEnd If WorksheetFunction.Weekday(lDay, 2) < 6 Then var = Application.Match(lDay, rng, 0) If IsError(var) Then lCount = lCount + 1 End If End If Next lDay ATage = lCount End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alternative zur DEC2BIN-Funktion ohne Begrenzung auf 512

Problem:	Die DEC2BIN-Funktion aus dem Analyse-Toolpack rechnet nur bis zum Wert von 512 um. Diese Begrenzung soll umgangen werden.
Lösung:	Über die nachfolgende benutzerdefinierte Funktion. Eingabesyntax: =vdecimaltobinary(A1)
VBA-Code:	<pre>StandardModule: Modul1 Funktion von Rob Bovey Function vDecimalToBinary(_ ByVal lDecimal As Long, _ Optional ByVal bReturnArray As Boolean = False, _ Optional ByVal b32Bit As Boolean = False) As Variant Const l16_BITS As Long = 15 Const l32_BITS As Long = 31 Dim dPlaceArray() As Double Dim lCount As Long Dim lNumBits As Long Dim lReturn() As Long Dim szReturn As String If b32Bit Then lNumBits = l32_BITS Else lNumBits = l16_BITS ReDim dPlaceArray(0 To lNumBits) ReDim lReturn(0 To lNumBits) For lCount = LBound(dPlaceArray) To UBound(dPlaceArray) dPlaceArray(lCount) = 2 ^ lCount Next lCount For lCount = UBound(dPlaceArray) To LBound(dPlaceArray) Step -1 If CBool(dPlaceArray(lCount) And lDecimal) Then If bReturnArray Then lReturn(lCount) = 1 Else szReturn = szReturn & Chr\$(49) End If Else If bReturnArray Then lReturn(lCount) = 0 Else szReturn = szReturn & Chr\$(48) End If End If Next lCount If bReturnArray Then vDecimalToBinary = lReturn() Else vDecimalToBinary = szReturn End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Prüfziffer abprüfen

Problem:	Die Prüfziffer der in Spalte A eingetragenen Nummer soll geprüft werden. Prüfung als Formel und als VBA-Makro.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Module11(strNo As String) As Boolean Dim iCounter As Integer Dim iValue As Integer iValue = CInt(Left(strNo, 1)) * 2 For iCounter = 2 To 7 iValue = iValue + CInt(Mid(strNo, _ iCounter, 1)) * (7 - iCounter + 2) Next iCounter iValue = 11 - (iValue Mod 11) If iValue = 10 Or iValue = 11 Then iValue = 0 If iValue = CInt(Right(strNo, 1)) Then Module11 = True End Function Sub Eintragen() Dim iRow As Integer iRow = 2 Do Until IsEmpty(Cells(iRow, 1)) Cells(iRow, 3) = Module11(Cells(iRow, 1).Text) If Cells(iRow, 3) = False Then Cells(iRow, 1).Interior.ColorIndex = 3 Else Cells(iRow, 1).Interior.ColorIndex = _ xlColorIndexNone End If iRow = iRow + 1 Loop End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Suchbegriff in Tabelle suchen und Fundzeilenindex zurückgeben

Problem: Ein Suchbegriff soll in einer mehrspaltigen Tabelle gesucht und der Fundstellenindex soll als Ergebnis zurückgegeben werden.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function mVergleich(varSearch As Variant, rng As Range) As String
    Dim var As Variant
    Dim iCol As Integer
    For iCol = 1 To rng.Columns.Count
        var = Application.Match(varSearch, rng.Columns(iCol), 0)
        If Not IsError(var) Then
            mVergleich = var & "," & iCol
            Exit Function
        End If
    Next iCol
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zählen von in einer Formel vorkommenden Zahlen

Problem:	In Zelle B2 sollen die in der Formel in A1 enthaltenen Zahlen gezählt werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function NumberCount(rng As Range) Dim sFormula As String Dim iCounter As Integer, iCount As Integer Dim bln As Boolean sFormula = rng.Formula For iCounter = 1 To Len(sFormula) If IsNumeric(Mid(sFormula, iCounter, 1)) Then If bln = False Then iCount = iCount + 1 bln = True End If Else If Mid(sFormula, iCounter, 1) <> "." Then bln = False End If End If Next iCounter NumberCount = iCount End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Text in Formel und Formel in Text

Problem:	Der Text aus Zelle A1 soll in eine Formel und die Formel aus A2 soll in Text umgewandelt werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Text2Formula(sTxt As String) sTxt = Application.Substitute(sTxt, ",", ".") Text2Formula = Evaluate(sTxt) End Function Function Formula2Text(rng As Object) Formula2Text = rng.Formula End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Werte aus Zeichenfolgen filtern und berechnen

Problem:	Aus Zeichenfolgen sollen Werte ausgelesen und berechnet werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Wertgehalt(sTxt As String) Const csChars As String = "0123456789+-()*/*^." Dim iChr As Integer Dim sFormula As String sTxt = WorksheetFunction.Substitute(sTxt, ".", "") sTxt = WorksheetFunction.Substitute(sTxt, ",", ".") For iChr = 1 To Len(sTxt) If InStr(csChars, Mid(sTxt, iChr, 1)) Then sFormula = sFormula & Mid(sTxt, iChr, 1) End If Next iChr Wertgehalt = Evaluate(sFormula) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Prüfung, ob es sich um einen Integer-Wert handelt

Problem:	Wie kann ich über eine benutzerdefinierte Funktion prüfen, ob es sich beim Eingabewert um einen Integer-Wert handelt?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function IntegerTest(dValue As Double) As Boolean If dValue > -32769 And dValue < 32768 And _ dValue - Fix(dValue) = 0 Then IntegerTest = True Else IntegerTest = False End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Textstring in berechenbare Formel umwandeln

Problem: Wie kann ich eine als Text vorliegende Formel über eine benutzerdefinierte Funktion berechnen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function TextAlsWert(sFormula As String) As Double
    Dim var As Variant
    var = Application.Substitute(sFormula, ",", ".")
    TextAlsWert = Application.Evaluate(var)
End Function
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summe aller Kosten einer Kostenart in einem Quartal

Problem:	Wie kann ich die Summe aller Kosten einer bestimmten Kostenart in einem vorgegebenen Quartal ermitteln?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Kosten(rng As Range, iQ As Integer, sK As String) As Double Dim dValue As Double Dim iRow As Integer For iRow = rng.Row To rng.Rows.Count + rng.Row - 1 If CInt(Format(Cells(iRow, 1), "q")) = iQ Then If Cells(iRow, 3) = sK Then dValue = dValue + Cells(iRow, 2) End If End If Next iRow Kosten = dValue End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung von Entfernungen über benutzerdefinierte Funktion

Problem: Wie kann ich die Entfernung von 2 Orten über eine benutzerdefinierte Funktion ermitteln, wenn sich in einem zweiten Tabellenblatt eine Entfernungstabelle befindet.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function Entfernung( _
    swks As String, _
    sStart As String, _
    sZiel As String) As Integer
    Dim wks As Worksheet
    Dim iRow As Integer
    Set wks = Worksheets(swks)
    iRow = 1
    Do Until IsEmpty(wks.Cells(iRow, 1))
        If wks.Cells(iRow, 1).Value = sStart And _
            wks.Cells(iRow, 2).Value = sZiel Then
            Entfernung = wks.Cells(iRow, 3).Value
        End If
        iRow = iRow + 1
    Loop
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl der über ParamArray festgelegten Parameter

Problem:	Wie kann ich die Anzahl der in einer benutzerdefinierten Funktion über "ParamArray" übergebenen Argumente bestimmen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function ArrayCount(_ ParamArray arr() As Variant) As Integer ArrayCount = UBound(arr) + 1 End Function Sub Aufruf() MsgBox ArrayCount(5, 7, 9, 11, 13) End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Filtern einer Zahl aus einer Zeichenfolge

Problem:	Wie kann ich eine in einer Zeichenfolge enthaltene Zahl herausfiltern?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function NumberFilter(strText) As Long Dim iCounter As Integer Dim sNumber As String For iCounter = 1 To Len(strText) If IsNumeric(Mid(strText, iCounter, 1)) Then sNumber = sNumber & Mid(strText, iCounter, 1) End If Next iCounter NumberFilter = CLng(sNumber) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Quersumme berechnen

Problem:	Wie kann ich in Excel die Quersumme einer Zahl berechnen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function CheckSum(lValue As Long) Dim iCounter As Integer, iSum As Integer lValue = Fix(lValue) For iCounter = 1 To Len(CStr(lValue)) iSum = CInt(iSum) + CInt(Mid(lValue, iCounter, 1)) Next iCounter CheckSum = iSum End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Monatsberechnung in der Rentenversicherung

Problem: In der Rentenversicherung errechnen sich die Tage für Teilmonate jeweils bis zum Monatsende, Komplettmonate zählen als 30 Tage. Wie kann diese Berechnung in Excel vorgenommen werden?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function RVTime(datStart As Date, datEnd As Date)
    Dim iMonth, iTime As Integer
    If Day(datStart) = 1 Then
        iTime = 30
    Else
        If Year(datStart) = Year(datEnd) And _
            Month(datStart) = Month(datEnd) Then
            iTime = Day(DateSerial(Year(datStart), _
                Month(datStart) + 1, 0)) - Day(datStart)
        Else
            iTime = Day(DateSerial(Year(datStart), _
                Month(datStart) + 1, 0)) - Day(datStart) + 1
        End If
    End If
    iMonth = (((Year(datEnd) - 1) - _
        Year(datStart) + 1) - 1) * 12
    iMonth = iMonth + (12 - Month(datStart))
    iMonth = iMonth + Month(datEnd) - 1
    iTime = iTime + (iMonth * 30)
    If Day(datEnd) = Day(DateSerial(Year(datStart), _
        Month(datStart) + 1, 0)) Then
        iTime = iTime + 30
    Else
        iTime = iTime + Day(datEnd)
    End If
    RVTime = iTime
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kalenderwoche nach DIN 1355

Problem:	Die Excel-Funktion =KW() liefert nicht die Kalenderwoche nach DIN 1355, wie errechnet sich diese?
Lösung:	Die Formel: =KÜRZEN((A2-WOCHENTAG(A2;2)-DATUM(JAHR(A2+4-WOCHENTAG(A2;2));1;-10))/7) - Nachfolgend als benutzerdefinierte Funktion:
VBA-Code:	<pre>StandardModule: basMain Function ISOWeek(dat As Date) As Integer Dim dbl As Double With WorksheetFunction ISOWeek = Fix((dat - .WeekDay(dat, 2) - _ DateSerial(Year(dat + 4 - _ .WeekDay(dat, 2)), 1, -10)) / 7) End With End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Durchgehende Rangfolge

Problem:	Wenn bei mehrere gleiche Ränge vorhanden sind, besetzt die ExcelRANG()-Funktion den nachfolgenden Rang nicht mehr. Beispiel: 2 erste Plätze, dann folgt der 3. Platz. Wie kann ich die Plätze durchgehend, ohne Lücken, belegen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function XRang(rng As Range, dValue As Double) As Integer Dim iCounter As Integer, iCell As Integer, iLarge As Integer For iCounter = 1 To rng.Rows.Count If iLarge <> Application.Large(rng, iCounter) Then iLarge = Application.Large(rng, iCounter) iCell = iCell + 1 If Application.Large(rng, iCounter) = dValue Then XRang = iCell Exit Function End If End If Next iCounter End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Mehrwertsteuer hinzu- oder abrechnen

Problem:	Eine benutzerdefinierte Funktion soll zu vorhandenen Werten die Mehrwertsteuer hinzurechnen bzw. abziehen. Parameter: Wert, Faktor, Wahrheitswert für hinzubzw. abrechnen.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function NetBru(_ Wert As Double, _ Faktor As Double, _ PlusMin As Boolean) As Double If PlusMin Then NetBru = Wert + Wert * Faktor Else NetBru = Wert / (1 + Faktor) End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Monatsblätter in Abhängigkeit ihres Namens summieren

Problem: Es sollen Monatsblätter in Abhängigkeit ihres Namens zu einem Start- und einem Enddatum summiert werden.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function DreiDSum( _
    swksA As String, _
    swksB As String, _
    sRng As String) _
    As Double
    Dim wks As Worksheet
    Dim dValue As Double
    Dim bln As Boolean
    For Each wks In Worksheets
        If wks.Name = swksA Then bln = True
        If bln Then
            dValue = dValue + WorksheetFunction.Sum(wks.Range(sRng))
        End If
        If wks.Name = swksB Then bln = False
    Next wks
    DreiDSum = dValue
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Addition der 10 höchsten Werte einer Zahlenreihe

Problem: Aus der mittleren Zahlenreihe sollen die 10 höchsten Werte addiert werden. Die jeweiligen Werte in der Zeile drüber und drunter werden dazugerechnet.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function Rank10Plus(rng As Range)
    Dim rngAct As Range
    Dim iValue As Integer
    For Each rngAct In rng.Cells
        If WorksheetFunction.Rank(rngAct.Value, rng) < 11 Then
            iValue = iValue + rngAct.Value
            iValue = iValue + rngAct.Offset(-1, 0).Value
            iValue = iValue + rngAct.Offset(1, 0).Value
        End If
    Next rngAct
    Rank10Plus = iValue
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summe über dynamischen Bereich bilden

Problem:	ber VBA soll eine Summe über einen dynamischen Bereich gebildet werden. Die erste Zelle ist die Zelle über der Summenformel oder bei VBA-Aufruf - die aktive Zelle, die letzte Zelle die Zelle vor der ersten leeren Zelle.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Sub Aufruf() MsgBox DynaSum(ActiveCell) End Sub Function DynaSum(Optional rng As Range) Dim dSum As Double Dim lRow As Long Dim iCol As Integer If rng Is Nothing Then Set rng = Application.Caller End If iCol = rng.Column lRow = rng.Row Do Until IsEmpty(Cells(lRow - 1, iCol)) dSum = dSum + Cells(lRow - 1, iCol) lRow = lRow - 1 If lRow = 1 Then Exit Do Loop DynaSum = dSum End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wahlweise gerade oder ungerade Zeilen summieren

Problem:	Wie kann ich wahlweise alle geraden oder ungeraden Zeilen summieren?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function BlnSum(rng As Range, bln As Boolean) As Double Dim dbl As Double Dim intRow As Integer, rngAct As Range For intRow = 1 To rng.Rows.Count If bln = False Then If rng.Cells(intRow).Row Mod 2 = 0 Then dbl = _ dbl + rng.Cells(intRow).Value Else If rng.Cells(intRow).Row Mod 2 <> 0 Then dbl = _ dbl + rng.Cells(intRow).Value End If Next intRow BlnSum = dbl End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach Währung und Zeitspannen summieren

Problem:	In Spalte B sollen die Werte aus dem Bereich B7:B100 summiert werden, die den Kriterien Währung, Startdatum, Enddatum entsprechen.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function IfSum(rngAll As Range, sValuta As String) As Variant Dim rng As Range Dim var() As Variant Dim iCounter As Integer ReDim var(1 To rngAll.Cells.Count, 1 To 2) For Each rng In rngAll.Cells iCounter = iCounter + 1 var(iCounter, 1) = False If InStr(rng.Text, sValuta) Then var(iCounter, 2) = True End If Next rng IfSum = var End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Summieren von Werten mit gleicher Schriftfarbe

Problem:	Über eine benutzerdefinierte Funktion sollen die Zellen einer vorgegebenen Schriftfarbe summiert werden. Als zweiter Parameter der Funktion wird der Farbindex der Schrift gefordert.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function SumFontColor(rng As Range, iColor As Integer) As Double Dim rngAct As Range Dim dAdd As Double For Each rngAct In rng.Cells If IsNumeric(rngAct.Value) Then If rngAct.Font.ColorIndex = iColor Then dAdd = dAdd + rngAct.Value End If End If Next rngAct SumFontColor = dAdd End Function Sub a() MsgBox ActiveCell.Font.ColorIndex End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach unterschiedlichen Währungen summieren

Problem:	Die Währungen aus Spalte A sollen in Spalte D summiert werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function GetSum(rngAll As Range, rngAct As Range, sValuta As String) Dim rng As Range Dim dValue As Double Dim sValue As String For Each rng In rngAll.Cells sValue = rng.Text If InStr(sValue, sValuta) Then dValue = dValue + rng.Value End If Next rng GetSum = dValue End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

SummeWenn-Summierung über mehrere Blätter

Problem:	In D1 soll die Summe aller Werte aus Spalte D der nachfolgenden Blätter ermittelt werden, sofern der Datumsbereich im jeweiligen Blatt der Bandbreite in den Zellen B1:B2 dieses Blattes entspricht.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function ThreeDSum() Dim dValue As Double Dim iCounter As Integer For iCounter = 2 To Worksheets.Count With Worksheets(iCounter) dValue = dValue + _ WorksheetFunction.SumIf(.Range("C2:C250"), _ ">=" & CDb1(Range("B1")), .Range("D2:D250")) - _ WorksheetFunction.SumIf(.Range("C2:C250"), _ ">" & CDb1(Range("B2")), .Range("D2:D250")) End With Next iCounter ThreeDSum = dValue End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

SUMMEWENN-Funktion über mehrere Tabellenblätter

Problem: Die Funktion SUMMEWENN kann nicht in 3D-Bereichen angewandt werden. Wie kann ich dieses Problem beseitigen.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function SuperSumIf(var As Variant, _  
    sFirst As String, sLast As String, _  
    colSearch As Integer, colValues As Integer)  
    Dim wks As Worksheet  
    Dim dValue As Double  
    Dim intSheet As Integer  
    For intSheet = Worksheets(sFirst).Index To _  
        Worksheets(sLast).Index  
        Set wks = Worksheets(intSheet)  
        dValue = dValue + _  
            WorksheetFunction.SumIf(wks.Columns(colSearch), _  
                var, wks.Columns(colValues))  
    Next intSheet  
    SuperSumIf = dValue  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

SVERWEIS-Funktion ohne Berücksichtigung leerer Zellen

Problem:	Die benutzerdefinierte Funktion in Zelle E2 soll die SVERWEIS-Formel so nachbilden, dass leere Zellen in Spalte B keine Berücksichtigung finden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function NoEmptysvL(vSearch As Variant, rngSearch As Range, iCol As Integer) As Variant Dim iRow As Integer For iRow = 1 To rngSearch.Rows.Count If rngSearch.Cells(iRow, 1).Value = vSearch And Not IsEmpty(rngSearch.Cells(iRow, iCol)) Then NoEmptysvL = rngSearch.Cells(iRow, iCol).Value Exit Function End If Next iRow NoEmptysvL = "#NV" End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus dem vorhergehenden Blatt einen SVERWEIS-Wert liefern

Problem:	Eine benutzerdefinierte Funktion soll aus dem vorhergehenden Blatt den Wert aus der rechts neben der Zelle mit dem höchsten Datum liegenden Zelle liefern.
Lösung:	Über die nachfolgende benutzerdefinierte Funktion.
VBA-Code:	<pre>StandardModule: Modul1 Function LastDateValue() As Variant Dim wks As Worksheet Set wks = Worksheets(Application.Caller.Parent.Index - 1) With WorksheetFunction LastDateValue = .Index(wks.Columns("A:B"), _ .Match(.Max(wks.Columns(1)), wks.Columns(1), 0), 2) End With End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

SVERWEIS unter Verwendung mehrerer Kriterien

Problem:	DieFormel soll mehrere Suchkriterien berücksichtigen und den ersten gefundenen Wert zurückliefern.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function myVLOOKUP(rng As Range, _ iResult As Integer, _ ParamArray varSearch() As Variant) Dim var As Variant Dim iCounter As Integer For iCounter = 0 To UBound(varSearch) var = Application.VLookup(_ varSearch(iCounter), rng, iResult, 0) If Not IsError(var) Then myVLOOKUP = var Exit Function End If Next iCounter myVLOOKUP = "#NV" End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

SVERWEIS-Formel über mehrere Fundstellen

Problem:	Wie kann ich in einer SVERWEIS-Formel mehrere Fundstellen als Zeichenfolge zusammenfassen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function sSVERWEIS(sName As String, rng As Range) As Variant Dim rngAct As Range Dim sResult As String For Each rngAct In rng.Columns(1).Cells If rngAct = sName Then sResult = sResult & rngAct.Offset(0, 1) & Chr(10) End If Next rngAct sSVERWEIS = Left(sResult, Len(sResult) - 1) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Wert aus Spalte C, wenn Spalten A und B Bedingungen erfüllen

Problem:	Ich brauche eine Formel, die mir in eine Zelle ausgibt, was in Spalte C steht, wenn in Spalte A "Müller" und in Spalte B "Spirit" stehen.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function sSVERWEIS(_ varA As Variant, _ varB As Variant, _ rng As Range) Dim rngAct As Range For Each rngAct In rng.Columns(1).Cells If rngAct.Value = varA Then If rngAct.Offset(0, 1).Value = varB Then sSVERWEIS = rngAct.Offset(0, 2).Value Exit Function End If End If Next rngAct End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benutzerdefinierte SVERWEIS-Funktion über mehrere Bereiche

Problem:	Wie muß eine benutzerdefinierte Funktion aussehen, die eine SVERWEIS()-Suche über mehrere Bereiche durchführt?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function SSVerweis(_ var As Variant, _ iCol As Integer, _ ParamArray rng()) Dim iCounter As Integer For iCounter = 0 To UBound(rng) If IsError(Application.VLookup(var, _ rng(iCounter), iCol, 0)) = False Then SSVerweis = Application.VLookup(var, _ rng(iCounter), iCol, 0) Exit Function End If Next iCounter End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus Notenspiegel die am häufigsten vergebene Note ermitteln

Problem:	Aus einem Notenspiegel soll die am häufigsten vergebene Note ermittelt werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function MultiMax(rng As Range) As String Dim iRow As Integer Dim sMax As String With rng For iRow = 1 To .Cells.Count If .Cells(iRow) = WorksheetFunction.Max(rng) Then sMax = sMax & .Cells(iRow).Offset(0, -1) & "," End If Next iRow End With MultiMax = Left(sMax, Len(sMax) - 1) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzahl eins Wertes unter Berücksichtigung der Groß- / Kleinschreibung

Problem:	Wie kann ich feststellen, wie oft eine bestimmte Zeichenfolge in einem vorgegebenen Bereich vorkommt? Groß- und Kleinschreibung sollgen berücksichtigt werden.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function WieOft(rngSource, rngTarget) As Integer Dim rngAct As Range Dim iCounter As Integer For Each rngAct In rngTarget.Cells If rngAct = rngSource Then iCounter = iCounter + 1 Next rngAct WieOft = iCounter End Function</pre>



Zahlwort englisch

Problem:	Wie kann ich eine Zahl in englischer Sprache in Worten darstellen lassen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	StandardModule: basMain Public Function ZWortEn(Number As Double) Dim m As Double m = Abs(Int(Number + 0.5)) m = Fix(Number) Select Case m Case Is < 100 ZWortEn = TextNum10(CInt(m)) Case Is < 1000 ZWortEn = TextNum100(CInt(m)) Case Is < 1000000 ZWortEn = TextNum1000(CInt(m)) Case Else ZWortEn = TextNum1000000(CInt(m)) End Select If Number < 0 Then ZWortEn = "minus " + ZWortEn ZWortEn = UCase(Left(ZWortEn, 1)) + Mid(ZWortEn, 2) End Function Private Function TextNum10(n As Integer) Select Case n Case 0 TextNum10 = "zero" Case Is < 20 TextNum10 = Einer(n) Case Else TextNum10 = Zweier(Int(n / 10)) If n <> Int(n / 10) * 10 Then TextNum10 = TextNum10 + "-" + _ Einer(n - Int(n / 10) * 10) End Select End Function Private Function TextNum100(n As Integer) Dim h#, x\$ h = Int(n / 100) x = TextNum10(n - h * 100) If x = "zero" Then TextNum100 = Einer(CInt(h)) + " hundred" Else TextNum100 = Einer(CInt(h)) + " hundred and " + x End If End Function Private Function TextNum1000(n As Integer) Dim k# If n < 100000 Then TextNum1000 = TextNum10(Int(n / 1000)) + " thousand " Else TextNum1000 = TextNum100(Int(n / 1000)) + " thousand " End If k = Int(n / 1000) Select Case n - k * 1000 Case 0 TextNum1000 = TextNum1000 Case Is < 100 TextNum1000 = TextNum1000 + "and " + TextNum10(n - k * 1000) Case Else TextNum1000 = TextNum1000 + TextNum100(n - k * 1000) End Select End Function Private Function TextNum1000000(n As Integer) Dim k# If n < 100000000 Then TextNum1000000 = TextNum10(Int(n / 1000000)) + " million " Else TextNum1000000 = TextNum100(Int(n / 1000000)) + " million " End If k = Int(n / 1000000) Select Case n - k * 1000000 Case 0 TextNum1000000 = TextNum1000000 Case Is < 100 TextNum1000000 = TextNum1000000 + "and " + TextNum10(n - k * 1000000) Case Is < 1000 TextNum1000000 = TextNum1000000 + TextNum100(n - k * 1000000) Case Else TextNum1000000 = TextNum1000000 + TextNum1000(n - k * 1000000) End Select End Function Private Function Einer(z As Integer) As String Einer = WorksheetFunction.Choose(z, "one", "two", "three", "four", "five", _ "six", "seven", "eight", "nine", "ten", "eleven", "twelve", "thirteen", _ "fourteen", "fifteen", "sixteen", "eighteen", "nineteen") End Function Private Function Zweier(z As Integer) As String Zweier = WorksheetFunction.Choose(z - 1, "twenty", "thirty", _ "forty", "fifty", "sixty", "seventy", "eighty", "ninety") End Function



Zahlwort deutsch

Problem:	Wie kann ich Zahlen in Zahlwörtern darstellen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	StandardModule: basMain Dim BisNeunzehn As Variant Dim Zehner As Variant Dim Tausender As Variant In Anlehnung an eine Klassenprogrammierung von Hans W. Hofmann Function ZWort(dZahl As Double, Optional bln As Boolean) Dim dRest As Double dRest = WorksheetFunction.Round((dZahl - Fix(dZahl)), 2) * 100 dZahl = Fix(dZahl) BisNeunzehn = Array("", "ein", "zwei", "drei", "vier", _ "fünf", "sechs", "sieben", "acht", "neun", "zehn", _ "elf", "zwölf", "dreizehn", "vierzehn", "fünfzehn", _ "sechzehn", "siebzehn", "achtzehn", "neunzehn") Zehner = Array("", "zehn", "zwanzig", "dreißig", _ "vierzig", "fünfzig", "sechzig", "siebzig", _ "achtzig", "neunzig") Tausender = Array("", "tausend", "millionen", "milliarden") If dRest = 0 Then ZWort = Text(dZahl) Else If bln Then ZWort = Text(dZahl) & " " & dRest & "/00" Else ZWort = Text(dZahl) End If End If End Function Private Function Wort(wert As Integer) As String Dim h As Integer h = wert Mod 100 If h < 20 Then Wort = BisNeunzehn(h) Else Wort = BisNeunzehn(h Mod 10) & IIf(h Mod 10 > 0, "und", "") & _ Zehner(Int(h / 10)) End If h = (wert Mod 1000 - h) / 100 If h > 0 Then Wort = BisNeunzehn(h) & "hundert" & Wort End Function Private Function Text(wert As Double) Dim l As Integer, i As Integer, p As Integer If InStr(1, Str(wert), ",") = 0 And InStr(1, Str(wert), ".") = 0 Then For i = 1 To 1 + Int(Len(Str(wert)) / 3) p = Val("0" & Mid("000" + Str(wert), _ Len("000" & Str(wert)) - i * 3 + 1, 3)) If p > 0 Then Text = Wort(p) & Tausender(i - 1) & Text Next Else Text = "#Ganzzahl!" End If If Right(Text, 3) = "ein" Then Text = Text & "s" End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Hyperlinks in Text, Adresse und Sub-Adresse trennen

Problem:	Mit einer benutzerdefinierten Funktion sollen Text, Adresse und SubAdresse von Hyperlinks ermittelt werden. Der Parameter 0 steht für den Text, 1 für die Adresse und 2 für die SubAdresse.
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: Modul1 Function SplitHype(rng As Range, iInfo As Integer) Select Case iInfo Case 0: SplitHype = rng.Value Case 1: SplitHype = rng.Hyperlinks(1).Address Case 2: SplitHype = rng.Hyperlinks(1).SubAddress End Select End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Formeltexte anzeigen

Problem:	Die Formeln aus dem Bereich A2:E18 sollen im Bereich A10:E12 über eine benutzerdefinierte Funktion angezeigt werden.
Lösung:	Über die nachfolgende benutzerdefinierte Funktion.
VBA-Code:	<pre>StandardModule: Modul1 Function fncGetFormula(rng As Range) If rng.HasFormula Then fncGetFormula = rng.FormulaLocal Else fncGetFormula = "" End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus Adressangabe Straße und Hausnummer separieren

Problem:	Aus der Adresse sollen die Strasse und die Hausnummer gefiltert werden.
Lösung:	Über die nachfolgende benutzerdefinierte Funktion. Eingabesyntax: =strasse(a1) und =hausnr(a1)
VBA-Code:	<pre>StandardModule: Modul1 Function HausNr(sAdr As String) As String Dim iCounter As Integer For iCounter = 1 To Len(sAdr) If IsNumeric(Mid(sAdr, iCounter, 1)) Then Exit For End If Next iCounter If iCounter = Len(sAdr) + 1 Then HausNr = "" Else HausNr = Right(sAdr, Len(sAdr) - iCounter + 1) End If End Function Function Strasse(sAdr As String) As String Dim iCounter As Integer For iCounter = 1 To Len(sAdr) If IsNumeric(Mid(sAdr, iCounter, 1)) Then Exit For End If Next iCounter Strasse = Trim(Left(sAdr, iCounter - 1)) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ermittlung einer VANITY-Rufnummer

Problem:	Der Abdruck von Buchstaben auf der Tastatur der Telefonapparate eröffnet die Möglichkeit, Telefonnummern zu beantragen, deren alphanumerische Umsetzung einen bestimmten Namen ergibt. Wie kann ich aus einer Zeichenfolge die Vanity-Nummer ermitteln?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Ph700(sTxt As String) As String Dim iCounter As Integer, iChr As Integer Dim sTmp As String sTxt = UCase(sTxt) For iCounter = 1 To 8 iChr = Asc(Mid(sTxt, iCounter, 1)) - 64 Select Case iChr Case Is < 19 sTmp = sTmp & Fix((iChr - 1) / 3) + 2 Case Is < 26 sTmp = sTmp & Fix((iChr - 2) / 3) + 2 Case Else sTmp = sTmp & Fix((iChr - 3) / 3) + 2 End Select Next iCounter Ph700 = "0700-" & sTmp End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolgen vervollständigen

Problem:	Der Zahlenstring aus Zelle A2 soll in Tabelle1 gesucht werden. Mit dem 2. Paramter ist anzugeben, ob komplette Übereinstimmung oder nicht (0 = komplett, 1 = nicht komplett)
Lösung:	Die gefundenen Zahlen sind jeweils mit dem Stirng "Nummer " zu vervollständig
VBA-Code:	<pre>StandardModule: basMain Function InsertStrings(sTxt As String, bln As Boolean) Dim iRow As Integer, sA As String, sB As String iRow = 1 With Worksheets("Tabelle1") Do Until IsEmpty(.Cells(iRow, 1)) If bln Then If InStr(.Cells(iRow, 1), sTxt) Then Exit Do Else If .Cells(iRow, 1) = sTxt Then Exit Do End If iRow = iRow + 1 Loop If IsEmpty(.Cells(iRow, 1)) Then InsertStrings = "" Else sA = .Cells(iRow, 1) Do While InStr(sA, ",") sB = sB & "Nummer " & _ Left(sA, InStr(sA, ",") - 1) & ", " sA = Right(sA, Len(sA) - InStr(sA, ",")) Loop sB = sB & "Nummer " & sA InsertStrings = sB Exit Function End If End With End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Vorkommen eines Zeichens in einem Bereich ermitteln

Problem: Wie kann ich ermitteln, wie oft ein bestimmtes Zeichen oder auch eine Zeichenfolge in einem bestimmten Bereich vorkommt?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function CharCount( _  
    rng As Range, _  
    sChr As String, _  
    bln As Boolean)  
    Dim rngAct As Range  
    Dim iCounter As Integer, iChr As Integer  
    For Each rngAct In rng.Cells  
        If InStr(rngAct.Value, sChr) Then  
            If bln = False Then  
                iCounter = iCounter + 1  
            Else  
                For iChr = 1 To Len(rngAct.Value)  
                    If Mid(rngAct.Value, iChr, Len(sChr)) = sChr Then  
                        iCounter = iCounter + 1  
                    End If  
                Next iChr  
            End If  
        End If  
    Next rngAct  
    CharCount = iCounter  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolge in Ziffern und Buchstaben aufteilen

Problem: Wie kann ich eine Zeichenfolge wie "12AB3C45D" in die Zahl 12345 und die Zeichenfolge "ABCD" aufteilen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function TrennenA(sTxt As String)
    Dim iCounter As Integer
    Dim sDbl As String
    For iCounter = 1 To Len(sTxt)
        If IsNumeric(Mid(sTxt, iCounter, 1)) Then
            sDbl = sDbl & Mid(sTxt, iCounter, 1)
        End If
    Next iCounter
    TrennenA = CDb1(sDbl)
End Function

Function TrennenB(sTxt As String)
    Dim iCounter As Integer
    Dim sTmp As String
    For iCounter = 1 To Len(sTxt)
        If Not IsNumeric(Mid(sTxt, iCounter, 1)) Then
            sTmp = sTmp & Mid(sTxt, iCounter, 1)
        End If
    Next iCounter
    TrennenB = sTmp
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Formelbereich in Einzelelemente aufschlüsseln

Problem:	Wie kann ich eine Formel wie =SUMME(A1:A3) in ihre Einzelelemente, also =SUMME(A1+A2+A3) aufschlüsseln?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function FormelText(rng As Range) As String Dim rngAct As Range Dim sTxt As String, sPre As String sTxt = rng.Formula sPre = Left(sTxt, InStr(sTxt, "(")) Set rng = Range(Mid(sTxt, InStr(sTxt, "(") + 1, Len(sTxt) - InStr(sTxt, "(") - 1)) sTxt = "" For Each rngAct In rng.Cells sTxt = sTxt & rngAct.Address(False, False) & "+" Next rngAct FormelText = sPre & Left(sTxt, Len(sTxt) - 1) & ")" End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Name des Verfassers in das Dokument eintragen

Problem: Wie kann ich den Namen des Verfassers einer Arbeitsmappe in eine Zelle einer Tabelle in dieser Arbeitsmappe eintragen lassen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain  
  
Function Verfasser() As String  
    Verfasser = ThisWorkbook.BuiltinDocumentProperties(3)  
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Telefonnummern nach Vorwahl und Anschluß aufteilen

Problem: Wie kann ich Telefonnummern nach Vorwahl und Anschluß aufteilen? Die Nummern liegen in den Formaten (02683)949049, 02683/949049 und 02683-949049 vor.

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code: StandardModule: basMain

```
Function Vorwahl(txt As String) As String
    Dim sTxt As String
    If InStr(txt, "(") > 0 Then
        sTxt = Right(txt, Len(txt) - 1)
        sTxt = Left(sTxt, InStr(sTxt, ")") - 1)
    ElseIf InStr(txt, "/") > 0 Then
        sTxt = Left(txt, InStr(txt, "/") - 1)
    ElseIf InStr(txt, "-") > 0 Then
        sTxt = Left(txt, InStr(txt, "-") - 1)
    Else
        sTxt = ""
    End If
    Vorwahl = sTxt
End Function

Function Nr(txt As String) As String
    Dim sTxt As String
    If InStr(txt, "(") > 0 Then
        sTxt = Right(txt, Len(txt) - InStr(txt, "("))
    ElseIf InStr(txt, "/") > 0 Then
        sTxt = Right(txt, Len(txt) - InStr(txt, "/"))
    ElseIf InStr(txt, "-") > 0 Then
        sTxt = Right(txt, Len(txt) - InStr(txt, "-"))
    Else
        sTxt = txt
    End If
    Nr = sTxt
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolgen zusammenfassen

Problem:	In Spalte B stehen Zeichenfolgen. Wie kann ich diese immer dann, wenn in Spalte A in der gleichen Zeile der Begriff "Anzahl" steht, in einer Zelle zusammenfassen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Sammeln(rngAll As Range) As String Dim rng As Range Dim sTxt As String For Each rng In rngAll.Columns(1).Cells If rng.Value = "Anzahl" Then sTxt = sTxt & rng.Offset(0, 1) & "," End If Next rng If Len(sTxt) > 1 Then Sammeln = Left(sTxt, Len(sTxt) - 1) End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anwendername in Tabellenblatt eintragen

Problem: Wie kann ich den unter Menü EXTRAS / OPTIONEN / Register "Allgemein" / Benutzername / eingetragenen Anwendernamen in eine Zelle des Arbeitsblattes eintragen lassen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function BenutzerName() As String
    BenutzerName = Application.UserName
End Function
```




Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Ein Zeichen in einer Formel suchen

Problem:	Wie kann ich über eine benutzerdefinierte Funktion feststellen, ob sich ein bestimmtes Zeichen in einer vorgegebenen Formel befindet?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function fSuchen(sTxt As String, rng As Range) As Boolean If InStr(rng.Formula, sTxt) > 1 Then fSuchen = True Else fSuchen = False End If End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Nach jedem 5. Zeichen ein Leerzeichen einfügen

Problem:	Wie kann ich über VBA in eine Zeichenfolge nach jedem 5. Zeichen ein Leerzeichen einfügen lassen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function Leer(sText As String) As String Dim iCounter As Integer Dim sTxt As String For iCounter = 5 To Len(sText) Step 5 sTxt = sTxt & Mid(sText, iCounter, 5) & " " Next iCounter Leer = Left(sTxt, Len(sTxt) - 1) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Häufigkeit des Vorkommens eines Zeichens

Problem: Wie kann ich die Häufigkeit des Vorkommens eines bestimmten Zeichens in einer Zelle oder einem Bereich ermitteln?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function HowMuch(rng As Range, strText As String)
    Dim rngAct As Range
    Dim var As Variant
    Dim iCounter As Integer
    Application.Volatile
    For Each rngAct In rng.Cells
        var = rngAct.Value
        iCounter = iCounter + Len(var) - _
            Len(Application.Substitute(var, strText, ""))
    Next rngAct
    HowMuch = iCounter
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeichenfolge umkehren

Problem: Wie kann ich eine Zeichenfolge umkehren, beispielsweise aus "Weber" "rebeW" machen?

Lösung: Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.

VBA-Code:

```
StandardModule: basMain

Function TurnOver(sSource As String) As String
    Dim iCounter As Integer
    Dim sText As String
    For iCounter = Len(sSource) To 1 Step -1
        sText = sText & Mid(sSource, iCounter, 1)
    Next
    TurnOver = sText
End Function
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Kopfzeile der Zelle mit dem höchsten Wert ermitteln.

Problem:	Wie ermittle ich in einer Matrix die Kopfzeile der Zelle, in der sich der höchste Wert der Matrix befindet?
Lösung:	Folgende Matrixformel: =INDEX(A1:E11;1;SUMME(WENN(A1:E11=MAX(A1:E11); SPALTE(A1:E11))))
VBA-Code:	Ohne VBA-Programmierung



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Anzeigen von Werten und Formeln

Problem:	Wie kann ich in A1 das Ergebnis einer Formel, 98, und in einer anderen Zelle die zugehörige Formel, z.B. "=8*9+26" anzeigen lassen?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function FormulaAsText(rng As Object) FormulaAsText = rng.Formula End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zwischensumme als Text

Problem:	Wie kann ich einer Reihe von Zahlen und Zeichenfolgen mit einer Funktion Zwischensummen bilden, ohne daß diese Zwischensummen in die Endsumme einfließen (z.B. als Übertrag)?
Lösung:	Geben Sie die nachfolgende benutzerdefinierte Funktion in ein Standardmodul ein.
VBA-Code:	<pre>StandardModule: basMain Function TextSum(rngAct As Object) Dim rngCell As Range Dim dblSum As Double For Each rngCell In rngAct.Cells If Application.IsNonText(rngCell) Then dblSum = dblSum + rngCell.Value End If Next rngCell TextSum = CStr(Format(dblSum, "#,##0.00")) End Function</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBA-Code als Zellkommentar anzeigen

Problem:	Der Code eines VBA-Moduls soll in Zelle A1 als Kommentar angezeigt werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub GetCode() Dim cmt As Comment Dim sCode As String With ThisWorkbook.VBProject.VBComponents("Modul1").CodeModule sCode = .Lines(1, .CountOfLines) & vbLf & vbLf & "Aktualisierung: " & Now End With sCode = Replace(sCode, vbCr, "") With Range("A1") If Not .Comment Is Nothing Then .Comment.Delete End If Set cmt = .AddComment(sCode) cmt.Shape.TextFrame.AutoSize = True End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zugriff auf VB-Editor verhindern, solange Arbeitsmappe geöffnet

Problem:	Solange die Arbeitsmappe geöffnet ist, soll auf die Entwicklungsumgebung nicht zugegriffen werden können.
Lösung:	Geben Sie den nachfolgenden Code in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	<pre>ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) LockWindowUpdate 0& End Sub Private Sub Workbook_Open() LockWindowUpdate Application.VBE.MainWindow.HWnd End Sub StandardModule: Modul1 Declare Function LockWindowUpdate Lib "user32" (ByVal hwndLock As Long) As Long</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBE-Komponenten der aktiven Arbeitsmappe auslesen

Problem:	Wie kann ich die VBE-Komponenten der aktiven Arbeitsmappe auslesen und in das aktive Blatt mit Angabe des Typs eintragen lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: modMain Sub VBEKomponenten() Dim oVBE As Object Dim iCounter As Integer iCounter = 1 For Each oVBE In ThisWorkbook.VBProject.VBComponents Select Case oVBE.Type Case 1: Cells(iCounter, 2) = "Standardmodul" Case 100: Cells(iCounter, 2) = "Klassenmodul" Case 3: Cells(iCounter, 2) = "UserForm" End Select Cells(iCounter, 1) = oVBE.Name iCounter = iCounter + 1 Next oVBE End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Cursorposition in der Entwicklungsumgebung auslesen

Problem: Wie kann ich die exakte, aktuelle Cursorposition in der Entwicklungsumgebung auslesen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub CommandButton1_Click()
    Dim lRow As Long, lColumn As Long
    On Error GoTo ERRORHANDLER
    Application.VBE.ActiveCodePane _
        .GetSelection lRow, lColumn, 0, 0
    MsgBox "Zeile: " & lRow & vbCrLf & _
        "Spalte: " & lColumn, _
        vbInformation, _
        "Cursorposition in der Entwicklungsumgebung"
    Exit Sub
ERRORHANDLER:
    MsgBox "Kein Cursor in der Entwicklungsumgebung!"
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Für ein VBE-Projekt mit VBA-Code ein Paßwort festlegen

Problem: Wie kann ich einem VBE-Projekt über einen VBA-Code ein Paßwort zuordnen?

Lösung: Den Code aus der Entwicklungsumgebung aufrufen.

VBA-Code: StandardModule: basMain

```
Sub Passwort()  
    SendKeys "%XI{TAB 9}{RIGHT}{tab} " & _  
        "{tab}Passwort{TAB}Passwort{tab}{enter}"  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Entwicklungsumgebung ein- und ausblenden

Problem:	Wie kann ich die Entwicklungsumgebung ein- und ausblenden?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub VBEEinAus() With Application.VBE.MainWindow .Visible = Not .Visible End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neue Klasse erstellen

Problem: Wie kann ich eine neue Klasse für ein Ereignis erstellen?

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code:

```
ClassModule: DieseArbeitsmappe

Dim MyObject As New Klasse1

Private Sub Workbook_Open()
    Set MyObject.appevent = Application
End Sub

ClassModule: Klasse1

Public WithEvents appevent As Application

Private Sub appevent_WindowResize(ByVal Wb As Excel.Workbook, _
    ByVal Wn As Excel.Window)
    MsgBox "Die Fenstergröße wurde verändert"
End Sub

Private Sub appevent_WorkbookBeforeClose(ByVal Wb As Excel.Workbook, _
    Cancel As Boolean)
    MsgBox "Feierabend!"
End Sub

Private Sub appevent_WorkbookOpen(ByVal Wb As Excel.Workbook)
    MsgBox "Guten Morgen!"
End Sub

StandardModule: basMain

Dim MyObject As New Klasse1
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsblatt mit Standardmodul in neue Arbeitsmappe kopieren

Problem:	Dieses Blatt wird in eine neue Arbeitsmappe kopiert, wobei auch Modul1 in die neue Arbeitsmappe übernommen wird.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub MeinMakro() MsgBox "Ich sollte kopiert werden!" End Sub StandardModule: Modul2 Sub Meldung() MsgBox "Ich bin nur ein Platzhalter!" End Sub StandardModule: basMain Sub CopyWkbVBA() Dim sPath As String sPath = Application.DefaultFilePath & "\temp.bas" ActiveSheet.Copy With ActiveSheet.Buttons(1) .OnAction = ActiveWorkbook.Name & "!MeinMakro" .Caption = "Meldung" End With Range("E8").Value = "Dieses Blatt wurde in eine neue Arbeitsmappe" Range("E10").Value = "Arbeitsmappe übernommen wurde." ThisWorkbook.VBProject.VBComponents("Modul1").Export sPath ActiveWorkbook.VBProject.VBComponents.Import sPath Kill sPath End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makro aus einer Arbeitsmappe in die Personl.xls kopieren

Problem:	Ein Makro aus einer Arbeitsmappe soll in die Personl.xls kopiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMyCode Sub Meldung() MsgBox "Hallo " & Application.UserName & "!" End Sub StandardModule: basMain Sub CopyInPersonlXLS() Dim vbC As Object Dim iRow As Integer Dim sCode As String On Error GoTo ERRORHANDLER With ThisWorkbook.VBProject.VBComponents("basMyCode").codemodule sCode = .Lines(1, .CountOfLines) End With Set vbC = Workbooks("Personl.xls").VBProject.VBComponents.Add(1) vbC.codemodule.deletelines 1, vbC.codemodule.CountOfLines vbC.codemodule.AddFromString sCode Exit Sub ERRORHANDLER: MsgBox "Das Makro konnte nicht kopiert werden!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blatt verschieben und neuer Code für Schaltfläche

Problem:	Das aktive Blatt soll in eine neue Arbeitsmappe verschoben werden, wobei der Schaltfläche ein zu generierender Code zuzuweisen und deren Aufschrift zu ändern ist.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub NewWkb() Dim vbC As Object Dim vbMdl As Object Dim iRow As Integer ActiveSheet.Move With ActiveSheet.Buttons(1) .OnAction = ActiveWorkbook.Name & "!Zurueck" .Caption = "Zurück" End With Set vbC = ActiveWorkbook.VBProject.VBComponents.Add(1) With vbC.codemodule iRow = .CountOfLines + 2 .InsertLines iRow, "Sub Zurueck" .InsertLines iRow + 1, " Worksheets.Add" .InsertLines iRow + 2, " Worksheets(2).Move After:=Workbooks(0119.xls).worksheets(1)" .InsertLines iRow + 3, " ActiveSheet.Buttons(1).Delete" .InsertLines iRow + 4, "End Sub" End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Makros aus einem Standardmodul in ComboBox einlesen

Problem: Über Schaltflächenklick sollen die Makros eines Standardmoduls in die ComboBox eingelesen und bei Auswahl in der ComboBox aufgerufen werden.

Lösung: Geben Sie den Ereigniscode in die nachfolgend genannten Module ein.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cboMacros_Change()
    If cboMacros.Value = "" Then Exit Sub
    Run cboMacros.Value
End Sub

Private Sub cmdFillCbo_Click()
    Dim iRow As Integer
    cboMacros.Clear
    With ThisWorkbook.VBProject.VBComponents("basMain").CodeModule
        For iRow = 1 To .CountOfLines
            If .ProcOfLine(iRow, 0) > "" Then
                If .ProcBodyLine(.ProcOfLine(iRow, 0), 0) = iRow Then
                    cboMacros.AddItem .ProcOfLine(iRow, 0)
                End If
            End If
        Next iRow
    End With
    MsgBox "Makros wurden eingelesen!"
End Sub

StandardModule: basMain

Sub Makro1()
    MsgBox "Ich bin Makro" & 1
End Sub

Sub Makro2()
    MsgBox "Ich bin Makro" & 2
End Sub

Sub Makro3()
    MsgBox "Ich bin Makro" & 3
End Sub

Sub Makro4()
    MsgBox "Ich bin Makro" & 4
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBA-Code auf Schaltflächen-Klick auskommentieren

Problem:	VBA-Code soll bei Schaltflächen-Klick auskommentiert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: Modul1 Sub Meldung() MsgBox "Hallo!" End Sub StandardModule: Modul2 Sub Auskommentieren() Dim iRow As Integer Dim sCode As String With ThisWorkbook.VBProject.VBComponents("Modul1").CodeModule For iRow = 1 To .CountOfLines sCode = .Lines(iRow, 1) .DeleteLines iRow .InsertLines iRow, "'" & sCode Next iRow End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Dynamische Generierung von Programmteilen

Problem: Während der Laufzeit werden Programmteile generiert und gelöscht.

Lösung: Den nachstehenden Code in ein Standardmodul eingeben, einer Schaltfläche zuweisen und starten.

VBA-Code:

```
StandardModule: Modul1

Sub SplitArray()
    Dim mdl As Object
    Dim arr(1 To 2, 1 To 12)
    Dim iCounter As Integer, iAct As Integer
    Dim sCode As String
    Application.VBE.MainWindow.Visible = False
    For iCounter = 1 To 12
        arr(1, iCounter) = iCounter
        arr(2, iCounter) = Format(DateSerial(1, iCounter, 1), "mmm")
    Next iCounter
    sCode = "Sub NewArrays(arrWhole)" & vbCrLf
    sCode = sCode & "    Dim iAct as Integer" & vbCrLf
    For iCounter = 1 To UBound(arr, 1)
        sCode = sCode & "    Dim arr" & iCounter & "(1 to " & _
            UBound(arr, 2) & ")" & vbCrLf
        sCode = sCode & "    For iAct = 1 to " & _
            UBound(arr, 2) & vbCrLf
        sCode = sCode & "        arr" & iCounter & "(iAct) = arrWhole(" & _
            iCounter & ", iAct)" & vbCrLf
        sCode = sCode & "    Next iAct" & vbCrLf
        sCode = sCode & "    For iAct = 1 to " & _
            UBound(arr, 2) & vbCrLf
        sCode = sCode & _
            "        MsgBox ""Datenfeld "" & iAct & "" aus Array arr" & _
            iCounter & ": "" & arr" & iCounter & "(iAct)" & vbCrLf
        sCode = sCode & "    Next iAct" & vbCrLf
    Next iCounter
    sCode = sCode & "End Sub" & vbCrLf & vbCrLf
    Set mdl = ThisWorkbook.VBProject.VBComponents.Add(1)
    mdl.CodeModule.AddFromString sCode
    Call Aufruf(arr)
    ThisWorkbook.VBProject.VBComponents.Remove mdl
End Sub

Sub Aufruf(arrAct As Variant)
    Call NewArrays(arrAct)
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Blatt ohne Code, Steuerelement und Formeln speichern

Problem:	Ein Arbeitsblatt soll als neue Arbeitsmappe ohne den Ereigniscode, die Schaltfläche und die Formel gespeichert werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle3 Private Sub CommandButton1_Click() Worksheets("Tabelle1").Copy ActiveSheet.OLEObjects.Delete With ActiveSheet.UsedRange .Value = .Value End With With ActiveSheet.Parent.VBProject.VBComponents(ActiveSheet.CodeName).CodeModule .DeleteLines 1, .CountOfLines End With On Error GoTo ERRORHANDLER ActiveWorkbook.SaveAs Range("B1").Value & "\test.xls" Exit Sub ERRORHANDLER: MsgBox "Die Datei konnte nicht gespeichert werden!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Löschen von Code aus Klassen- und Standardmodulen

Problem: Der Code soll aus allen Arbeitsblatt-Klassenmodulen der aktiven Arbeitsmappe und aus Modul 2 gelöscht werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub Worksheet_Change(ByVal Target As Range)
    MsgBox Target.Address
End Sub

StandardModule: Modul1

Sub DeleteCode()
    Dim wks As Worksheet
    Dim mdl As Object
    With ActiveWorkbook.VBProject
        For Each wks In Worksheets
            With .VBComponents(wks.CodeName).CodeModule
                .DeleteLines 1, .CountOfLines
            End With
        Next wks
        With .VBComponents("Modul2").CodeModule
            .DeleteLines 1, .CountOfLines
        End With
    End With
    MsgBox "Alles klar!"
End Sub

StandardModule: Modul2

Sub MsgboxCall()
    MsgBox "Hallo"
End Sub

ClassModule: Tabelle2

Private Sub Worksheet_Change(ByVal Target As Range)
    MsgBox Target.Address
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neue Schaltfläche mit zugehörigem Code in Tabellenblatt

Problem:	Wie kann ich auf in ein Tabellenblatt eine neue Schaltfläche mit zugehörigem Code einfügen lassen? Nach betätigung dieser Schaltfläche soll sie einschließlich Code wieder gelöscht werden.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdCreateButton_Click() Dim oBtn As Button Dim sCode As String sCode = "Sub cmdNew_Click" & vbCrLf sCode = sCode & " MsgBox ""Hat funktioniert!"" & vbCrLf sCode = sCode & _ " ActiveSheet.Buttons(Application.Caller).Delete " & vbCrLf sCode = sCode & " With ThisWorkbook.VBProject.VBComponents" & _ " ("basMain").CodeModule" & vbCrLf sCode = sCode & " .DeleteLines 1, .CountOfLines" & vbCrLf sCode = sCode & " End With" & vbCrLf sCode = sCode & "End Sub" Set oBtn = Buttons.Add(100, 100, 70, 20) With oBtn .Caption = "Meldung" .OnAction = "cmdNew_Click" End With With ThisWorkbook.VBProject.VBComponents("basMain").CodeModule .AddFromString sCode End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zellinhalt während der Laufzeit in Prozedur übernehmen

Problem:	Wie kann ich einen Zellinhalt während der Laufzeit in eine Prozedur übernehmen und wirksam werden lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Sub Aufruf() End Sub Sub DialogeStarten() Dim lRow As Long Dim sTxt As String sTxt = " Call " & Range("A1").Value With ThisWorkbook.VBProject.VBComponents("Tabelle1").CodeModule lRow = .ProcBodyLine("Aufruf", 0) + 1 .replaceline lRow, sTxt Call Dialoge Call Aufruf End With End Sub Private Sub Dialoge() frmTest.Show End Sub Private Sub cmdStart_Click() Call DialogeStarten End Sub ClassModule: frmDialog Private Sub cmdWeiter_Click() Unload Me End Sub ClassModule: frmTest Private Sub cmdWeiter_Click() Unload Me End Sub</pre>



CommandBar in Abhängigkeit vorhandener Makros erstellen

Problem:	Wie kann ich beim Öffnen einer Arbeitsmappe automtisch eine Symbolleiste für alle in einem bestimmten Modul vorhandenen Makros erstellen lassen?
Lösung:	Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.
VBA-Code:	ClassModule: DieseArbeitsmappe Private Sub Workbook_BeforeClose(Cancel As Boolean) On Error Resume Next Application.CommandBars("AktuelleMakros").Delete On Error GoTo 0 End Sub Private Sub Workbook_Open() Dim oBar As CommandBar Dim oBtn As CommandBarButton Dim arr() As String Dim iCounter As Integer On Error Resume Next Application.CommandBars("AktuelleMakros").Delete On Error GoTo 0 Set oBar = Application.CommandBars.Add(_ Name:="AktuelleMakros", _ Position:=msoBarTop, _ MenuBar:=False, _ temporary:=True) arr = GetMacros For iCounter = 1 To UBound(arr) Set oBtn = oBar.Controls.Add With oBtn .Caption = "Makro Nr. " & iCounter .OnAction = "MacroNo" & iCounter .Style = msoButtonCaption .FaceId = 70 + iCounter End With Next iCounter oBar.Visible = True End Sub StandardModule: basMain Sub MacroNo1() MsgBox "Ich bin Makro Nr. 1" End Sub Sub MacroNo2() MsgBox "Ich bin Makro Nr. 2" End Sub Sub MacroNo3() MsgBox "Ich bin Makro Nr. 3" End Sub Sub MacroNo4() MsgBox "Ich bin Makro Nr. 4" End Sub Sub MacroNo5() MsgBox "Ich bin Makro Nr. 5" End Sub StandardModule: basFunction Function GetMacros() Dim arrMacros() As String Dim var As Variant Dim iRow As Integer, iCounter As Integer With ThisWorkbook.VBProject _ .VBComponents("basMain").CodeModule For iRow = 1 To .CountOfLines If .ProcOfLine(iRow, 0) > "" Then If iCounter = 0 Then iCounter = iCounter + 1 ReDim Preserve arrMacros(iCounter) arrMacros(iCounter) = .ProcOfLine(iRow, 0) Else var = Application.Match(.ProcOfLine(iRow, 0), arrMacros, 0) If IsError(var) Then iCounter = iCounter + 1 ReDim Preserve arrMacros(iCounter) arrMacros(iCounter) = .ProcOfLine(iRow, 0) End If End If End If Next iRow End With GetMacros = arrMacros End Function



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Zeile in einem Code-Modul austauschen

Problem:	Wie kann ich eine Zeile in einem Code-Modul während der Laufzeit austauschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdCodeAustauschen_Click() Call ModulZeileTauschen End Sub StandardModule: basMain Sub ModulZeileTauschen() Dim iCounter As Integer Call Meldung With ThisWorkbook.VBProject.VBComponents("basMain").CodeModule For iCounter = 1 To .CountOfLines If .Lines(iCounter, 1) = " MsgBox ""Austauschen!"" Then .ReplaceLine iCounter, " MsgBox ""Ausgetauscht!"" Exit Sub End If Next iCounter End With MsgBox "Der Code wurde ausgetauscht!" End Sub Sub Meldung() MsgBox "Austauschen!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aufzurufenden Makronamen während der Laufzeit aus Zelle entnehmen

Problem: Wie kann ich während der Laufzeit zu einem Makro verzweigen, dessen Name in Zelle A1 steht?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub cdmAufruf_Click()
    Call AufrufStarten
End Sub

StandardModule: basMain

Sub AufrufStarten()
    Dim lRow As Long
    Dim sTxt As String
    sTxt = "  Call " & Range("A1").Value
    With ThisWorkbook.VBProject.vbcomponents("basMain").CodeModule
        lRow = .ProcBodyLine("Aufruf", 0) + 1
        MsgBox "Makroname wird im Aufruf-Makro eingesetzt!"
        .replaceline lRow, sTxt
        Call Aufruf
        MsgBox "Makroname wird im Aufruf-Makro gelöscht!"
        .replaceline lRow, "  'Hier wird der Code temporär eingefügt"
    End With
End Sub

Private Sub Aufruf()
    'Hier wird der Code temporär eingefügt
End Sub

Private Sub MeinMakro()
    MsgBox "Ich bin " & Range("A1")
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Benutzerdefinierte Funktion durch Code erstellen und löschen

Problem: Wie kann ich über VBA-Code eine benutzerdefinierte Funktion erstellen, aufrufen und wieder - mit dem Aufruf - löschen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle11

Private Sub cmdNewFunction_Click()
    Call FunktionAnlegen
End Sub

StandardModule: basMain

Sub FunktionAnlegen()
    Dim bas As Object
    Dim iStart As Integer
    With ThisWorkbook.VBProject.VBComponents("basFunction").CodeModule
        .InsertLines 3, "Function Berechnen(dValue As Double) As Double"
        .InsertLines 4, "    Berechnen = dValue * 12"
        .InsertLines 5, "End Function"
    End With
    Call FunktionAufrufen
End Sub

Sub FunktionAufrufen()
    Dim iStart As Integer, iCount As Integer
    MsgBox "Ergebnis der Funktion Berechnen:" & Berechnen(5)
    If MsgBox("Funktion löschen?", vbQuestion + vbYesNo) _
        = vbNo Then Exit Sub
    With ThisWorkbook.VBProject.VBComponents("basFunction").CodeModule
        .DeleteLines 1, .CountOfLines
    End With
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellencode nach Kopieren des Blattes löschen

Problem:	Wenn ich ein Arbeitsblatt als neue Arbeitsmappe kopiere, wird das Klassenmodul des Blattes mitkopiert. Dieses läßt sich nicht löschen. Wie kann ich den Code in diesem Modul per VBA entfernen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle3 Private Sub cmdWksCopy_Click() Dim wks As Worksheet Worksheets("Tabelle1").Copy Set wks = ActiveSheet With ActiveWorkbook.VBProject _ .VBComponents(wks.CodeName).CodeModule .DeleteLines 1, .CountOfLines End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Aus AddIn neue Arbeitsmappe mit Code erstellen

Problem: Nach Abfrage soll über die Workbook_Open-Prozedur eine neue Arbeitsmappe mit Button und Code erstellt werden.

Lösung: Geben Sie den Ereigniscode in das Klassenmodul der Arbeitsmappe ein.

VBA-Code: ClassModule: DieseArbeitsmappe

```
Private Sub Workbook_Open()  
    Dim oBtn As Button  
    Dim vbc As Object  
    Dim sCode As String  
    If MsgBox( _  
        prompt:="Neue Arbeitsmappe erstellen?", _  
        Buttons:=vbYesNo + vbQuestion) = vbNo Then Exit Sub  
    Application.ScreenUpdating = False  
    sCode = "Sub Meldung()" & vbCrLf  
    sCode = sCode & "    MsgBox ""Hallo Welt!"" & vbCrLf  
    sCode = sCode & "End Sub"  
    Workbooks.Add 1  
    Set vbc = ActiveWorkbook.VBProject.VBComponents.Add(1)  
    vbc.codemodule.addfromstring sCode  
    Set oBtn = ActiveWorkbook.ActiveSheet.Buttons.Add(60, 12, 120, 25)  
    oBtn.Caption = "Meldung"  
    oBtn.OnAction = ActiveWorkbook.Name & "!Meldung"  
    Application.ScreenUpdating = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Code aufrufen und löschen

Problem:	Wie kann ich eine aufgerufene Sub-Routine nach dem Aufruf löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub CommandButton1_Click() Call AufrufUndLoeschen End Sub StandardModule: basMain Sub Meldung() MsgBox "Hallo " & Application.UserName End Sub Sub AufrufUndLoeschen() Dim iStart As Integer, iLen As Integer Call Meldung With ThisWorkbook.VBProject.VBComponents("basMain").CodeModule iStart = .ProcBodyLine("Meldung", 0) iLen = .ProcCountLines("Meldung", 0) .DeleteLines iStart, iLen End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Alle Prozeduren und Funktionen auflisten

Problem:	Wie kann ich mir in XL8 alle Prozeduren und benutzerdefinierten Funktionen einer Arbeitsmappe in einem Tabellenblatt auflisten lassen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Activate() 'Dies ist nur ein Platzhalter zu Demonstrationszwecken End Sub StandardModule: basMain Sub MakroListe() Dim vbc As Object Dim iRow As Integer, iCol As Integer, iCounter As Integer Dim sMacro As String Cells.Clear Rows(1).Font.Bold = True For Each vbc In ThisWorkbook.VBProject.VBComponents iRow = 1 iCol = iCol + 1 Cells(iRow, iCol).Value = vbc.Name With vbc.CodeModule For iCounter = 1 To .CountOfLines If .ProcOfLine(iCounter, 0) > "" Then sMacro = .ProcOfLine(iCounter, 0) If sMacro <> Cells(iRow, iCol) Then iRow = iRow + 1 Cells(iRow, iCol).Value = sMacro End If End If Next iCounter End With Next vbc Columns.AutoFit End Sub ClassModule: UserForm1 Private Sub UserForm_Click() 'Dies ist nur ein Platzhalter zu Demonstrationszwecken End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Arbeitsmappe anlegen und Workbook_Open-Prozedur schreiben

Problem:	Wie kann ich in XL8 über VBA-Code eine Arbeitsmappe anlegen, im Klassenmodul der Arbeitsmappe und eine Workbook_Open-Prozedur schreiben?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub OpenProzedurAnlegen() Dim wkb As Workbook Dim sFile As String Application.ScreenUpdating = False sFile = Application.Path & "\testwkb.xls" Set wkb = Workbooks.Add(1) With wkb.VBProject.VBComponents(wkb.CodeName).CodeModule .InsertLines 3, "Private Sub Workbook_Open()" .InsertLines 4, " MsgBox ""Bin jetzt da!"""" .InsertLines 5, " ActiveWorkbook.Close Savechanges:=False" .InsertLines 6, "End Sub" End With Application.DisplayAlerts = False ActiveWorkbook.SaveAs sFile Application.DisplayAlerts = True ActiveWorkbook.Close savechanges:=False Workbooks.Open sFile Application.ScreenUpdating = True End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Während der Laufzeit aktives Modul löschen

Problem:	Wie kann ich während der Laufzeit das Modul mit dem aktiven Makro löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub cmdDelete_Click() Call ModulLoeschen End Sub StandardModule: basMain Sub ModulLoeschen() MsgBox "Noch bin ich da!" With ThisWorkbook.VBProject .VBComponents.Remove .VBComponents("basMain") End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Codeblatt mit Rändern drucken

Problem: Bei XL8 habe ich beim Druck eines Moduls nicht die Möglichkeit der Randeinstellung. Alle Drucke beginnen am äußersten oberern und linken Rand. Wie kann ich das ändern?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub ModulDrucken()
    Dim sTxt As String
    Application.ScreenUpdating = False
    With ThisWorkbook.VBProject.VBComponents("basMain").CodeModule
        sTxt = .Lines(1, .CountOfLines)
    End With
    sTxt = WorksheetFunction.Substitute(sTxt, vbCr, "")
    Workbooks.Add 1
    With Range("A1")
        .EntireColumn.ColumnWidth = 80
        .WrapText = True
        .Value = sTxt
        .Font.Name = "Courier New"
    End With
    ActiveSheet.PageSetup.LeftMargin = Application.InchesToPoints(1)
    ActiveSheet.PrintPreview
    ActiveWorkbook.Close savechanges:=False
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Modul in einer zweiten Arbeitsmappe löschen

Problem: Wie kann ich in XL8 über VBA-Code ein VBA-Modul einer zweiten Arbeitsmappe löschen?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub Loeschen()
    Dim sFile As String
    Application.ScreenUpdating = False
    sFile = ThisWorkbook.Path & "\testvbe.xls"
    If Dir(sFile) = "" Then
        MsgBox "Test-Arbeitsmappe wurde nicht gefunden!"
    Else
        Workbooks.Open sFile
        On Error GoTo ERRORHANDLER
        With ActiveWorkbook.VBProject
            .VBComponents.Remove .VBComponents("basTest")
        End With
    End If
    Application.ScreenUpdating = True
    Exit Sub
ERRORHANDLER:
    MsgBox "Modul wurde nicht gefunden!"
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Module und UserForms austauschen

Problem: Wie kann ich alle Module und UserForms einer Arbeitsmappe löschen und danach die Module und UserForms des aktuellen Unterverzeichnisses laden?

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: basMain

Sub DeleteAndImport()
    Dim col As New Collection
    Dim vbc As VBComponent
    Dim iCounter As Integer
    Dim sPath As String
    sPath = Application.Path & "\"
    For Each vbc In ThisWorkbook.VBProject.VBComponents
        If vbc.Type <> 3 And _
            vbc.Type < 100 And _
            vbc.Name <> "basMain" Then
            vbc.Export sPath & vbc.Name & ".bas"
            col.Add sPath & vbc.Name & ".bas"
            ThisWorkbook.VBProject.VBComponents.Remove vbc
        ElseIf vbc.Type = 3 Then
            vbc.Export sPath & vbc.Name & ".frm"
            col.Add sPath & vbc.Name & ".frm"
            ThisWorkbook.VBProject.VBComponents.Remove vbc
        End If
    Next vbc
    For iCounter = 1 To col.Count
        ThisWorkbook.VBProject.VBComponents.Import col(iCounter)
        Kill col(iCounter)
    Next iCounter
    MsgBox "Ex- und Import abgeschlossen!"
End Sub

ClassModule: Tabelle1

Private Sub Worksheet_SelectionChange(ByVal Target As Range)

End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Module von Mappe zu Mappe kopieren

Problem:	Wie kann ich Modulblätter von der einen zur anderen Arbeitsmappe kopieren und das neue Blatt benennen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub CopyModule() Dim sPath As String sPath = Application.Path & "\" ThisWorkbook.VBProject _ .VBComponents("basMain").Export sPath & "basMain.bas" Workbooks.Add 1 With ActiveWorkbook.VBProject .VBComponents.Import sPath & "basMain.bas" .VBComponents("basMain").Name = "MyModul" End With Kill sPath & "\basMain.bas" MsgBox "Modul wurde kopiert!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

ComboBox mit Ereigniscode in Tabellenblatt erstellen

Problem: Im aktiven Blatt soll eine neue ComboBox mit zugeordnetem Ereigniscode erstellt werden.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
ClassModule: Tabelle1

Private Sub ComboBox1_Change()
    MsgBox "Der Monat " & ComboBox1.Text & " wurde ausgewählt!"
End Sub

StandardModule: Modul1

Sub NewComboBox()
    Dim chb As OLEObject
    Dim iCounter As Integer
    Dim sCode As String
    Set chb = ActiveSheet.OLEObjects.Add( _
        ClassType:="Forms.ComboBox.1", _
        Width:=100, _
        Height:=20, _
        Top:=100, _
        Left:=200)
    For iCounter = 1 To 12
        chb.Object.AddItem Format(DateSerial(1, iCounter, 1), "mmm")
    Next iCounter
    sCode = "Private Sub ComboBox1_Change()" & vbCrLf
    sCode = sCode & "    MsgBox ""Der Monat "" & " & _
        "ComboBox1.Text & "" wurde ausgewählt!"" & vbCrLf
    sCode = sCode & "End Sub" & vbCrLf
    With ThisWorkbook.VBProject.VBComponents(ActiveSheet.CodeName).codemodule
        .AddFromString sCode
    End With
    ActiveCell.Select
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Neues Blatt hinzufügen und Klassenmodul-Code übernehmen

Problem:	Der Arbeitsmappe soll ein neues Blatt hinzugefügt werden, dessen Klassenmodul den Code von Tabelle1 enthält.
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>ClassModule: Tabelle1 Private Sub Worksheet_Change(ByVal Target As Range) MsgBox "Eingabe in Zelle " & Target.Address End Sub StandardModule: basMain Sub NewSheetWithCode() Dim txt As String Worksheets.Add With ThisWorkbook.VBProject With .vbcomponents("Tabelle1").codemodule txt = .Lines(1, .CountOfLines) End With With .vbcomponents(ActiveSheet.CodeName).codemodule If .CountOfLines > 0 Then .DeleteLines 1 End If .AddFromString txt End With End With End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBE-Namen der Blattmodule beim Kopieren festlegen

Problem: Wenn ich das jeweils letzte Blatt einer Arbeitsmappe 50 mal kopiere, wird im VBE-Projektexplorer dem Namen des Klassenmoduls des jeweils kopierten Blattes immer eine 1 hinzugefügt, bis das Programm abbricht. Wie kann ich das ändern.

Lösung: Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code:

```
StandardModule: Modul1

Sub cmdButton_Click()
    Dim iCounter As Integer, iCount As Integer
    Application.ScreenUpdating = False
    For iCounter = 1 To 50
        iCount = Worksheets.Count
        Worksheets(iCount).Visible = True
        Worksheets(iCount).Copy after:=Worksheets(iCount)
        ActiveSheet.Name = "Blatt" & iCounter
        With ThisWorkbook.VBProject.VBComponents(ActiveSheet.CodeName)
            .Properties("_CodeName").Value = ActiveSheet.Name
        End With
    Next iCounter
    Worksheets(1).Select
    Application.ScreenUpdating = True
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Tabellenblattnamen der VBE-Projekte ändern

Problem:	Wenn ich ein Tabellenblatt mit Namen "Tabelle1" kopiere, wird für das kopierte Blatt im Projektexplorer der Entwicklungsumgebung der Name "Tabelle1" angelegt. Wie kann ich diesen Namen per VBA-Makro korrigieren?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub BlattKopieren() Dim wks As Worksheet Dim sName As String Worksheets(1).Copy after:=Worksheets(Worksheets.Count) Set wks = ActiveSheet sName = "NewTable" & Worksheets.Count - 2 ActiveSheet.Name = sName ThisWorkbook.VBProject.VBComponents(wks.CodeName) _ .Properties("_codename") = sName MsgBox "Blatt wurde kopiert!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Module und UserForms einer zu öffnenden Arbeitsmappe exportieren

Problem: Module und UserForms der in Zelle B1 genannten Arbeitsmappe sollen exportiert werden.

Lösung: Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.

VBA-Code: StandardModule: Modul1

```
Sub BasExtract()  
    Dim vbc As Object  
    Dim iCounter As Integer  
    Dim sFile As String, sTarget As String  
    sFile = Range("B1").Value  
    If Dir(sFile) = "" Then  
        Beep  
        MsgBox "Die Datei " & sFile & " wurde nicht gefunden!"  
        Exit Sub  
    End If  
    sTarget = WorksheetFunction.Substitute(sFile, "\" & Dir(sFile), "")  
    Application.EnableEvents = False  
    On Error GoTo ERRORHANDLER  
    Workbooks.Open sFile, False  
    For Each vbc In ActiveWorkbook.VBProject.VBComponents  
        iCounter = iCounter + 1  
        If vbc.Type = 1 Or vbc.Type = 100 Then  
            vbc.Export sTarget & "\backup" & iCounter & ".bas"  
        Else  
            vbc.Export sTarget & "\backup" & iCounter & ".frm"  
        End If  
    Next vbc  
    ActiveWorkbook.Close savechanges:=False  
ERRORHANDLER:  
    Application.EnableEvents = True  
End Sub
```



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

VBE-Verweis löschen

Problem:	Wie kann ich in XL8 einen VBProject-Verweis auf eine andere Arbeitsmappe oder eine DLL per VBA löschen?
Lösung:	Geben Sie den nachfolgenden Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub VerweisLoeschen() Dim sLink As String sLink = "xlVBE" On Error GoTo ERRORHANDLER ActiveWorkbook.VBProject.References.Remove _ ThisWorkbook.VBProject.References(sLink) Exit Sub ERRORHANDLER: MsgBox "Verweis zu " & sLink & " wurde nicht gefunden!" End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Per VBA einen Verweis zu einer *.DLL erstellen

Problem:	Wie kann ich in XL8 per VBA einen Verweis zur mso97.dll erstellen?
Lösung:	Geben Sie den Code in ein Standardmodul ein und weisen Sie ihn einer Schaltfläche zu.
VBA-Code:	<pre>StandardModule: basMain Sub VerweisHinzu() Dim sLink As String sLink = Range("B1").Value If CInt(Left(Application.Version, 1)) = 9 Then Application.VBE.ActiveVBProject _ .References.AddFromFile sLink End If End Sub</pre>



Herbers Excel-FAQ - Die Texte

Bestandteil der Excel-Lehr- und Lernmaterialien

Das Impressum

Verantwortlich für den Inhalt:

Hans W. Herber
Im Schönblick 5
D-53567 Asbach
Germany, Old Europe

☞ **Telefon:** +49 2683 949047

☞ **URL:** <http://www.herber.de>

☞ **Email:** hans@herber.de

☞ **Texte, Gestaltung, Programmierung:** Hans W. Herber

☞ **USt-ID:** DE 149 450 590

 [Visitenkarte in Outlook einfügen](#)

© Das Copyright liegt ausschließlich bei **Hans W. Herber**.
