

R | R | Z | N |

Regionales Rechenzentrum für Niedersachsen



Leibniz  
Universität  
Hannover

# Excel – VBA Dialogfelder

## Dialogfelder

- Funktionen
  - Mit Hilfe der `MsgBox()` werden kurze Nachrichten an den Benutzer ausgegeben.
  - Mit Hilfe der `InputBox()` kann der Benutzer einen Wert eingegeben.
- Integrierte Dialogfelder
  - Alle in der Anwendung vorhandenen Dialogfelder.
  - ... für die Auswahl und Speicherung von Dateien etc.
- UserForms
  - ... steuern die Eingabe von Daten.
  - ... stellen Informationen in einer bestimmten Form dar.
  - ... werden vom Entwickler erstellt.

## Integrierte Dialogfelder

- ... können nur innerhalb von Excel verwendet werden.
- ... werden mit Hilfe von `Application.Dialogs([Konstante])` definiert. Die Konstante legt die Art des Dialogfeldes fest.
- ... werden mit Hilfe von `Application.Dialogs([Konstante]).Show(arg1, arg2, ...)` angezeigt. Es können bis zu 30 Parameter in Abhängigkeit des Typs des Dialogfeldes übergeben werden.
- Die Konstanten und deren Parameter können Sie unter <http://msdn.microsoft.com/en-us/library/bb211087.aspx> nachschlagen.
- ... liefern `True` zurück, wenn das Dialogfeld ordnungsgemäß beendet wird. Andernfalls wird `False` zurückgeliefert.

## Dialogfeld "Datei öffnen"

```
Dim dialogOeffnen As Dialog
Dim abbruch As Boolean
Dim datei As String

datei = "*.xlsm"

Set dialogOeffnen = Application.Dialogs(xlDialogOpen)
abbruch = dialogOeffnen.Show(datei)

If abbruch = False Then
    MsgBox ("Es wurde keine Datei ausgewählt")
End If
```

## Welche Schaltfläche wurde gedrückt?

```
Sub mappeOpen()
```

```
...
```

```
Set dialogOeffnen = Application.Dialogs(xlDialogOpen)  
abbruch = dialogOeffnen.Show(datei)
```

```
Select Case abbruch
```

```
Case 0
```

```
    MsgBox ("Abbrechen-Schaltfläche")
```

```
Case -1
```

```
    MsgBox ("OK-Schaltfläche")
```

```
Case -2
```

```
    MsgBox ("Schließen-Schaltfläche")
```

```
End Select
```

```
End Sub
```

## Methode der Anwendung nutzen

```
Dim pfad As String
Dim filename As String

pfad = Application.ActiveWorkbook.Path
If pfad <> "" Then
    pfad = pfad & Application.PathSeparator
Else
    pfad = "C:\Dokumente und Einstellungen\...\Eigene Dateien\"
End If

filename = Application.GetOpenFilename(, ,
    "Auswahl einer Arbeitsmappe", , False)
```

## Dialogfeld "Speichern unter"

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Dim speichernUnter As Dialog
    Dim abbruch As Boolean
    Dim pfad As String
    ...

    Set speichernUnter = Application.Dialogs(xlDialogSaveAs)
    abbruch = speichernUnter.Show(pfad)

    If abbruch = False Then
        MsgBox ("Die Datei ist nicht gespeichert.")
    End If
End Sub
```

## Andere Möglichkeit

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Dim pfad As String
    Dim fileName As String
    ...

    pfad = pfad & "myDialog.xlsx"
    fileName = Application.GetSaveAsFilename(pfad,
                                             "Excel-Arbeitsmappe (*.xlsx), *.xlsx", ,
                                             "Arbeitsmappe speichern")

    ActiveWorkbook.SaveAs fileName
End Sub
```

## Eigene Dialogfelder (UserForms)

- ... bieten eine einfache Schnittstelle zwischen Mensch und Maschine an.
- ... stellen Elemente wie Icons, Buttons etc. für die Interaktion mit dem Computersystem bereit.
- ... bieten den Benutzer die Möglichkeit Aktionen mit Hilfe der Maus anzustoßen und zu steuern.
- ... bieten die Möglichkeit, Informationen an den Benutzer auszugeben. Zum Beispiel: Der Benutzer gibt Daten in einem Textfeld ein. In einem Bezeichnungsfeld wird eine Hilfe zum Ausfüllen gegeben.

## UserForms importieren und exportieren

- *Datei – Datei importieren.*
- *Datei – Datei exportieren.*
- Das Layout der UserForm wird vom Code des Dialogfeldes getrennt.
- Die Datei ".frm" enthält die Ereignisprozeduren etc.
  - Diese Datei kann mit einem Texteditor gelesen werden.
- Die Datei ".frx" enthält das Layout und die Steuerelemente auf dem Dialogfeld.
  - Die Informationen werden binär gespeichert.
  - Die Daten können mit Hilfe des VBA-Editors und dem Eigenschaftenfenster verändert werden.
  - Die Datei kann nur mit einem Hex-Editor gelesen werden.

## ... bestehen aus

- ... einem Fenster (window). Merkmale:
  - Rechteckige Form.
  - Begrenzung durch einen Rahmen.
  - Titelleiste mit Informationen am oberen Rand.
- In dem Fenster werden Steuerelemente (controls) abgelegt. Steuerelemente benötigen immer ein Fenster als Container.
- Steuerelemente sowohl als auch das Fenster können mit dem Benutzer interagieren.

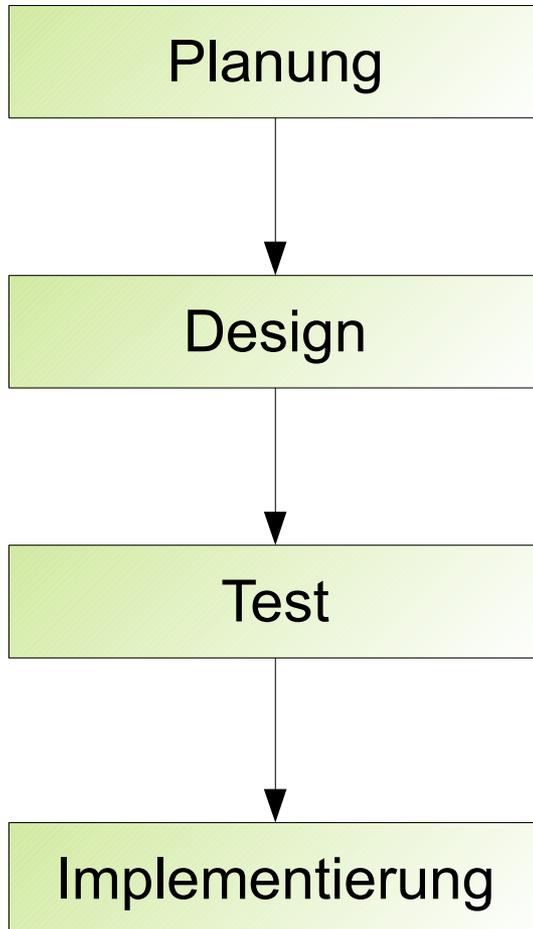
## Steuerelemente sind zum Beispiel

- ... Textfelder. Der Benutzer kann Zahlen und Zeichen eingeben.
- ... Kontrollkästchen oder Optionsfelder. Der Benutzer wählt aus einer bestimmten Anzahl von Möglichkeiten aus.
- ... Listenfelder. Die Möglichkeiten werden als Liste dargestellt.
- ... Schaltflächen. Schaltflächen starten Aktionen.

# Fragekatalog für den Entwurf von Dialogfeldern

- Welches Vorwissen hat der Benutzer?
- Welche Arbeitsabläufe sollen abgebildet werden? Welche Aufgaben sollen mit Hilfe des Dialogfeldes erledigt werden?
- Welche Informationen müssen wie angezeigt werden, um die Aufgabe ohne Fehler oder Nachfrage erledigen zu können?
- Wie interagiert der Nutzer mit dem Dialogfeld?
- Welches Layout wird genutzt? Ist ein Corporate Design vorhanden? Welche Style Guides sind vorhanden?

## Von der Planung zur Fertigstellung



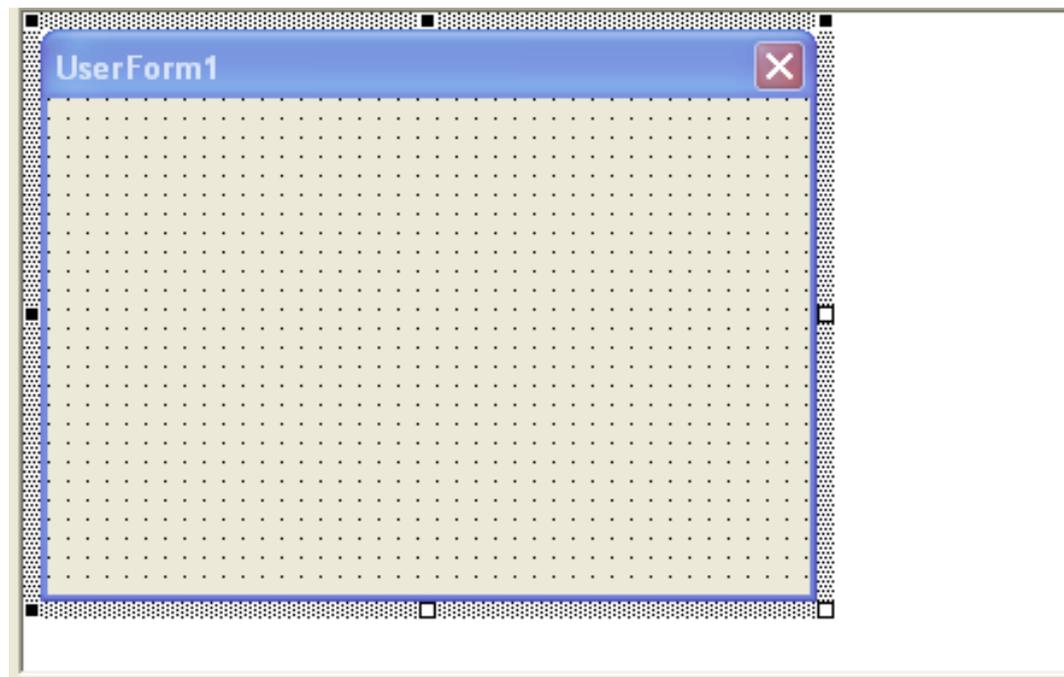
Objekte und Aktionen definieren.  
Entwicklung von Wortlisten und Icons.

Passend zur Aufgabe werden die  
Steuerelemente ausgewählt.  
Layout und Aufbau des Dialogfeldes.  
Navigation zwischen Dialogfeld und Benutzer.

Prototyp erstellen.  
Usability-Test.

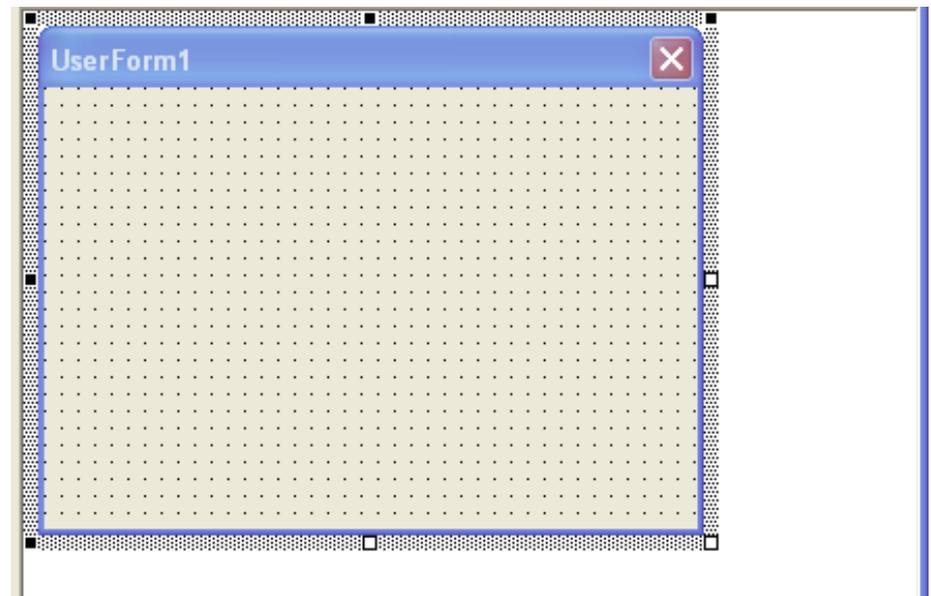
## UserForm einfügen

- Öffnen Sie mit <ALT>+<F11> den Visual Basic Editor.
- Klicken Sie auf *Einfügen – UserForm*. Es wird ein leeres Dialogfeld und die Werkzeugsammlung angezeigt.



## Dialogfeld aktivieren

- Mit einem Klick auf die Titelleiste wird das Dialogfeld aktiviert.
- Ein aktives Dialogfeld
  - ... besitzt einen dicken schraffierten Rand.
  - ... hat acht Anfasser (schwarze Rechtecke). Auf jede Ecke und jeweils in der Mitte einer Kante wird ein Anfasser platziert.

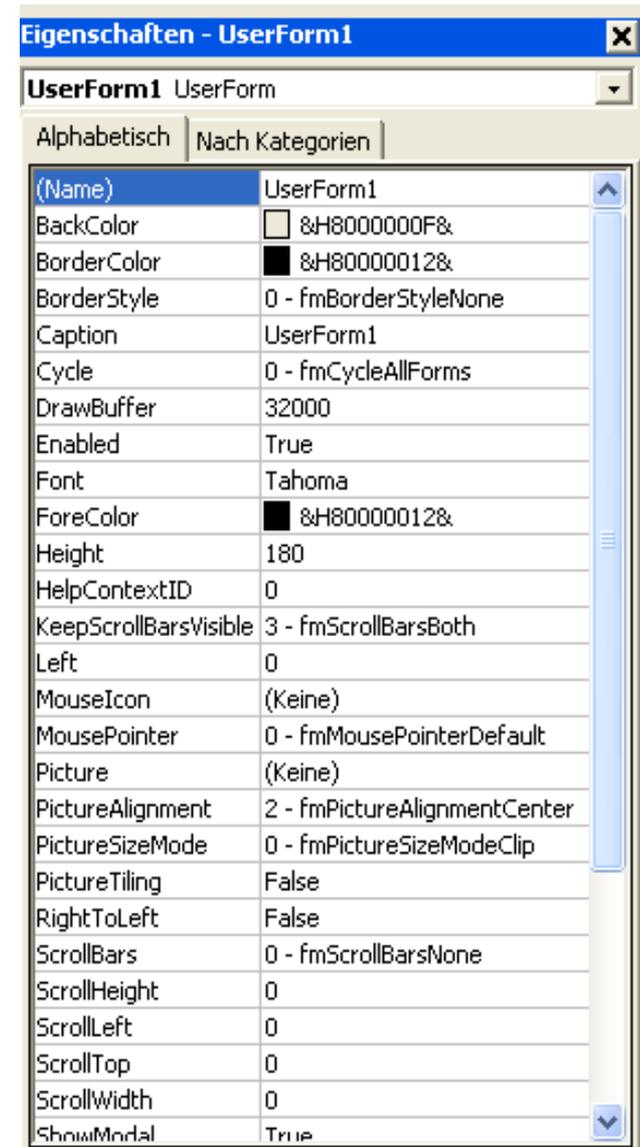


## Größe verändern

- Aktivieren Sie das Dialogfeld.
- Legen Sie den Mauszeiger über einen der acht Anfasser. Der Mauszeiger verwandelt sich in eine Linie, die an den Enden einen Pfeil hat.
- Halten Sie die linke Maustaste gedrückt und ziehen Sie die Maus
  - ... nach innen, um das Dialogfeld zu verkleinern.
  - ... nach außen, um das Dialogfeld zu vergrößern.

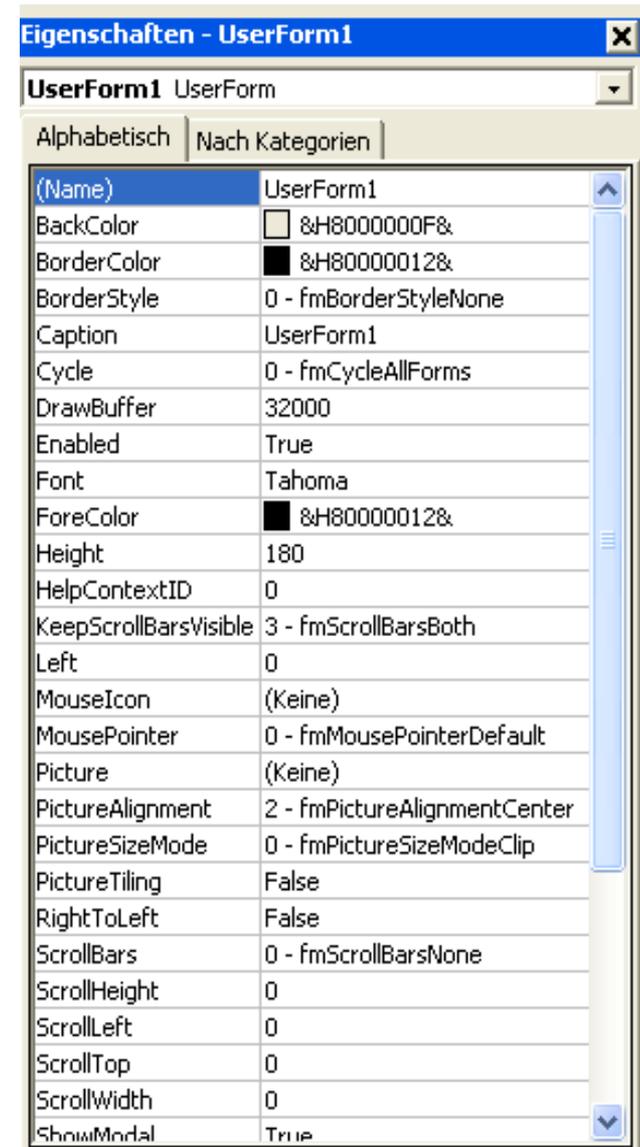
# Eigenschaften

- Blenden Sie das Eigenschaftenfenster mit Hilfe von *Ansicht – Eigenschaftenfenster* oder <F4> ein.
- Es ist möglich die Eigenschaften alphabetisch sortiert oder nach Kategorien zusammengefasst anzuzeigen.



## Aufbau

- In der Titelleiste und in dem Kombinationsfeld darunter wird das aktive Objekt angezeigt. In Abhängigkeit des ausgewählten Objekts werden die Eigenschaften angezeigt.
- Mit Hilfe der Reiter können die Eigenschaften alphabetisch sortiert oder nach Kategorien zusammengefasst werden.
- In der linken Spalte werden alle Eigenschaften aufgelistet und in der rechten Spalte der dazugehörige Wert angezeigt.



## Eigenschaftswerte verändern.

- Mit einem Klick in der linken Spalte wählen Sie eine Eigenschaft aus.
- Wechseln Sie mit der Maus in die rechte Spalte.
- Eigenschaftswerte können
  - ... mit Hilfe der Tastatur eingegeben werden.
  - ... aus einer Liste ausgewählt werden. Am rechten Rand der Zeile wird ein Pfeil nach unten eingeblendet. Durch einen Klick auf den Pfeil wird die Liste geöffnet.
  - ... mit Hilfe eines Assistenten oder Dialogfelds eingestellt werden. Mit einem Klick auf die Schaltfläche *3 Punkte* wird der passende Assistent geöffnet.

## Name eines Objekts

- Die Eigenschaft Name enthält eine Bezeichnung für eine UserForm oder ein Steuerelement.
- Der Eigenschaftswert ist ein Platzhalter für das Objekt. Der Name wird genutzt, um
  - ... eine UserForm zu öffnen oder zu schließen.
  - ... Eigenschaften mit Hilfe von Code einzustellen.
- ... identifiziert das Objekt eindeutig.
- ... ist vergleichbar mit einem Personalausweis.
- ... sollte die Aufgabe der UserForm widerspiegeln.
- ... sollte die enthaltenen Daten / Informationen widerspiegeln.

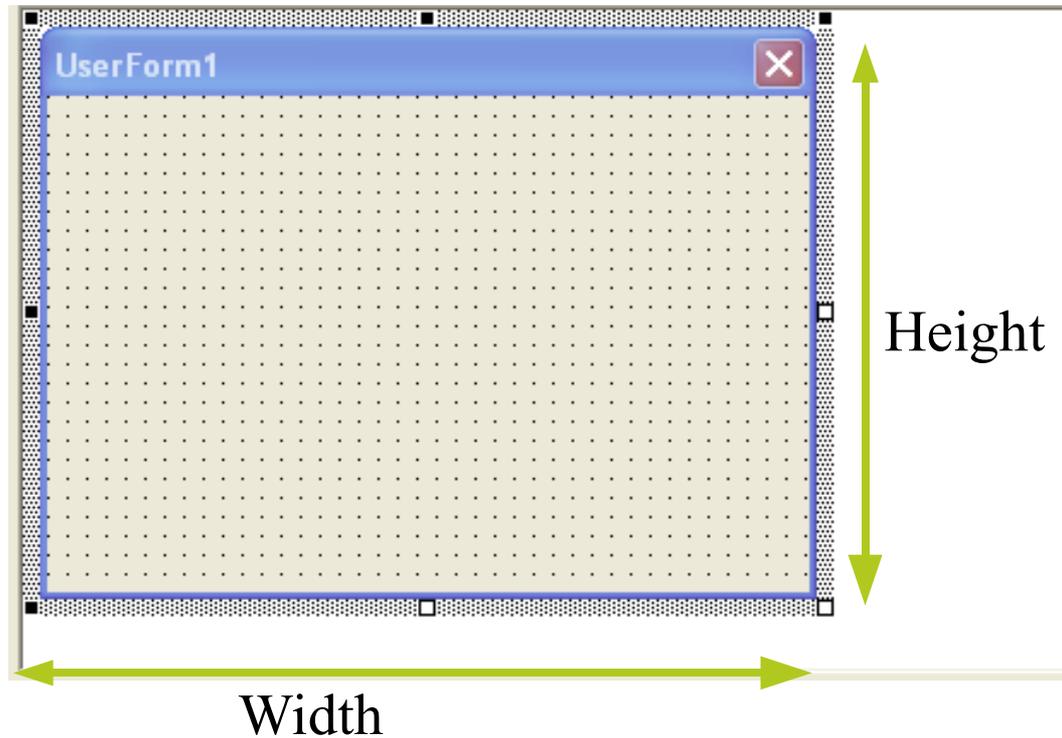
## Hinweise

- Der Name sollte für sich selber sprechen und die Aufgabe des Steuerelements beschreiben.
- Standardmäßig wird die Bezeichnung „UserForms“ plus eine Nummer genutzt. Diese Bezeichnung
  - ... ist nicht aussagekräftig.
  - ... dokumentiert nicht die Verwendung des Dialogfeldes.
- Zusammengesetzte Namen:
  - Jedes Wort in der Bezeichnung beginnt mit einem Großbuchstaben. Zum Beispiel: UmsatzzahlenBerechnen.
  - Die einzelnen Wörter werden durch Unterstriche getrennt. Zum Beispiel: Umsatzzahlen\_Berechnen.

## Präfixe zur Kennzeichnung des Objekts

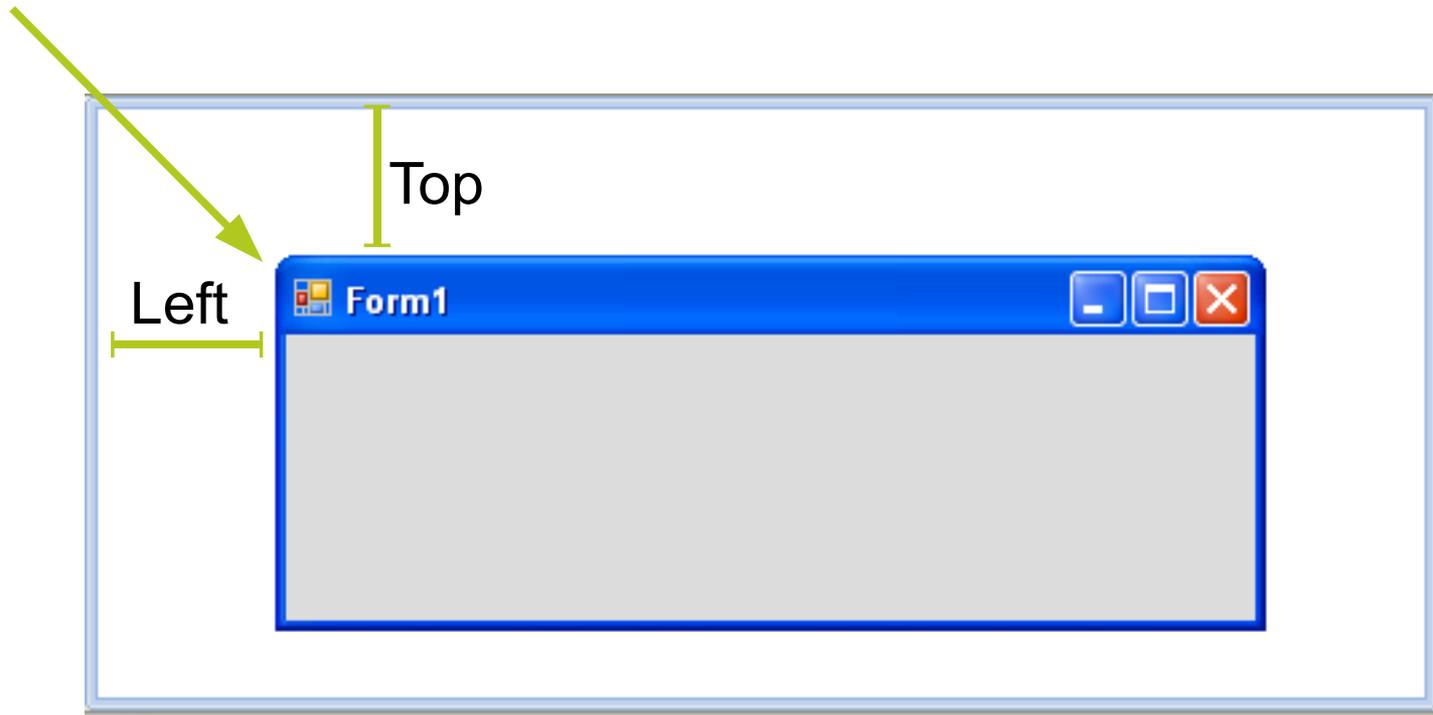
Präfix	Objekt	Beschreibung
frm	UserForm	Container für Steuerelemente.
lbl	Label	Informationen für den Benutzer anzeigen.
opt	Radiobutton	Auswahl eines Elements aus vielen.
chk	Checkbox	Auswahl von mehreren Möglichkeiten aus vielen.
cmd	Button	Schaltfläche zum Starten von Aktionen.
lst	Listbox	Liste von möglichen Elementen.
cbo	Combobox	Liste, kombiniert mit einem Textfeld.

# Größe eines Objekts



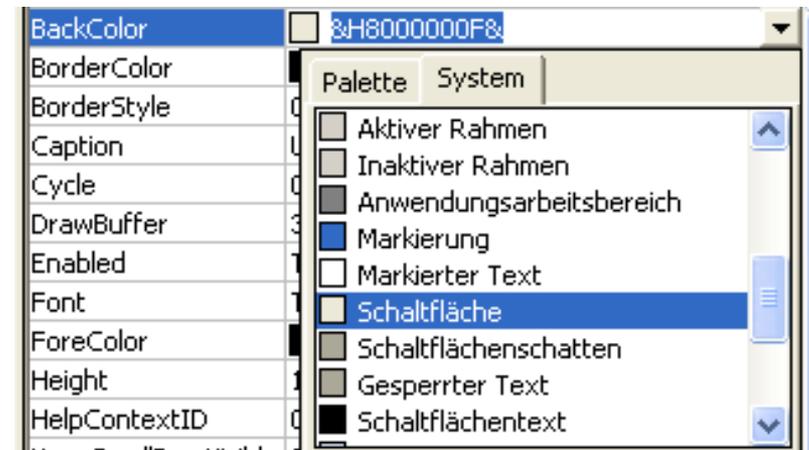
# Position eines Objekts

[Left], [Top]



## BackColor und ForeColor

- BackColor gibt die Hintergrundfarbe als RGB-Wert an.
- ForeColor gibt die Vordergrundfarbe (Schriftfarben etc.) als RGB-Wert an.
- Es können vordefinierte Systemfarben genutzt werden.
- Mit Hilfe der Registerkarte Palette wird die Farbpalette angezeigt. Durch einen Klick mit der rechten Maustaste auf eines der weißen Felder am unteren Rand der Palette wird ein Dialogfeld zur Definition von eigenen Farben geöffnet.



## Rahmen (Border)

- `BorderStyle` blendet den Rahmen ein oder aus.
- `BorderColor` legt die Rahmenfarbe fest.

## Schriftformatierung (Font)

- ... legt die Schriftart fest.
- ... bestimmt die Schriftgröße.
- Eine Schrift kann fett oder kursiv dargestellt werden.
- Schriftformatierungen werden an die Steuerelemente innerhalb einer UserForm vererbt.

## Titelleiste der Form

- Caption enthält den Text der Titelleiste
- Das Icon am linken Rand der Titelleiste kann nur mit Hilfe von A(pplication)P(rogramming)I(nterface) - Funktionen geändert werden.

## Dialogfeld in den Speicher laden

- Die Anweisung Load [Name der Form] lädt das Dialogfeld in den Speicher.
- Das Dialogfeld wird aber nicht angezeigt.
- Die Wartezeit für den Benutzer wird durch das Laden von großen Dialogfeldern in den Speicher verkürzt.

## Dialogfeld anzeigen

- Mit Hilfe des Menübefehls *Ausführen – Sub/UserForm ausführen* oder <F5> wird das Dialogfeld zum Testen gestartet.
- Mit Hilfe der Anweisung [Name der Form].Show in einer Prozedur wird die UserForm eingeblendet.
  - Eine UserForm wird standardmäßig modal geöffnet. Der Benutzer muss das Dialogfeld erst schließen, um eine neue Aktion zu starten. Befehle in den Ribbons etc. sind gesperrt.
  - Mit Hilfe der Anweisung [Name der Form].Show vbModeless muss das Fenster nicht geschlossen werden, bevor andere Aktionen zu gestartet werden.

## Dialogfeld schließen

- Mit Hilfe der Schließen-Schaltfläche in der Titelleiste wird das Dialogfeld geschlossen.
- Mit Hilfe der Anweisung [Name der Form].Hide kann ein Dialogfeld ausgeblendet werden. Das Dialogfeld befindet sich aber noch im Speicher.
- Die Anweisung Unload [Name der Form] entfernt das Dialogfeld aus dem Speicher und schließt es.
  - Der Benutzer kann nicht mit dem Dialogfeld interagieren.
  - Unload Me schließt die UserForm, in der der Code gerade läuft.
- Die Anweisung End beendet das Programm vollständig und schließt alle Dialogfelder.

## Steuerelemente

- ... sind zum Beispiel
  - ... Textfelder, um Daten einzugeben.
  - ... sind Kontrollkästchen, Optionsfelder oder Listen. Der Nutzer kann eine oder mehrere Elemente aus einer Liste auswählen.
  - ... Schaltflächen, um bestimmte Aktionen wie zum Beispiel Berechnungen anzustoßen.
  - ... sind Elemente, die andere Steuerelemente zusammenfassen oder das Formular gliedern.
- ... werden immer in einem Container abgelegt. Ein Steuerelement kann nie ohne ein Dialogfeld existieren.
- ... werden in der Werkzeugsammlung angezeigt.

## Werkzeugsammlung

- ... kann mit Hilfe von *Ansicht – Werkzeugsammlung* eingeblendet werden.
- ... enthält alle Steuerelemente, die auf einer UserForm abgelegt werden können.
- Wenn der Mauszeiger über eines der Steuerelemente in der Werkzeugsammlung liegt, wird in einem gelben Erklärfenster der ausgewählte Typ eingeblendet.



## Steuerelement-Typ auswählen und einfügen

- Klicken Sie auf eines der Icons in der Werkzeugsammlung. Das Icon wird vertieft dargestellt. Das Icon ist ausgewählt.
- Der Mauszeiger wird in die UserForm verschoben.
- Mit einem Klick auf eine leere Fläche in der UserForm wird das ausgewählte Steuerelement an der ausgewählten Position eingefügt.
- Anschließend wird das Steuerelement Objekte auswählen automatisch aktiviert. Das Layout des neu eingefügten Steuerelementes kann angepasst werden.

## Ein Steuerelement aktivieren

- Mit einem Klick auf das Steuerelement wird das Element aktiviert.
- Ein aktives Steuerelement
  - ... besitzt einen dicken schraffierten Rand.
  - ... hat acht Anfasser (schwarze Rechtecke). Für jede Ecke und jeweils in der Mitte einer Kante wird ein Anfasser eingeblendet.

## Mehrere Steuerelemente aktivieren

- Klicken Sie auf ein Steuerelement.
- Jedes weitere Steuerelement kann mit Hilfe von <STRG>+<Mausklick> aktiviert werden.
- Mit Hilfe von <UMSCHALT>+Mausklick werden alle zwischen dem vorherigen und dem aktuell ausgewählten Element automatisch markiert.
- Andere Möglichkeit: Mit Hilfe der gedrückt gehaltenen Maus ziehen Sie einen Auswahlrahmen um die zu markierenden Steuerelemente.
- Mit Hilfe von <STRG>+<A> können alle Steuerelemente auf einer Form markiert werden.

## Steuerelemente kopieren

- Es ist eine beliebige Anzahl von Steuerelementen ausgewählt.
- Mit Hilfe von *Bearbeiten – Kopieren* (<STRG>+<C>) werden die markierten Steuerelemente in der Zwischenablage abgelegt.
- Mit Hilfe von *Bearbeiten – Einfügen* (<STRG>+<V>) werden die Elemente aus der Zwischenablage mittig auf der UserForm eingefügt.

## Andere Möglichkeiten

- Das Steuerelement wird mit Hilfe von <STRG>+<Maustaste> an eine neue Position verschoben. Es wird automatisch eine Kopie erzeugt.
- Andere Möglichkeit: Halten die die rechte Maustaste gedrückt und verschieben das Steuerelement. Sobald Sie die Maustaste loslassen, wird das Kontextmenü eingeblendet. Wählen Sie den Befehl *An diese Stelle kopieren* aus. Es wird eine Kopie des Steuerelements an der neuen Position eingefügt.

## Größe verändern

- Aktivieren Sie das Steuerelement.
- Legen Sie den Mauszeiger über einen der acht Anfasser. Der Mauszeiger verwandelt sich in eine Linie, die jeweils an den Enden einen Pfeil hat.
- Halten Sie die linke Maustaste gedrückt und ziehen Sie die Maus
  - ... nach innen, um das Steuerelement zu verkleinern.
  - ... nach außen, um das Steuerelement zu vergrößern.
- Andere Möglichkeit: Einstellung der Eigenschaften Width und Height.
- Falls mehr als ein Steuerelement aktiviert ist, wird jedes Steuerelement entsprechend der Angaben angepasst. Bestehende Abstände zwischen den Steuerelementen werden ignoriert.

## Position verändern

- Der Mauszeiger befindet sich innerhalb des Steuerelements oder des Auswahlrahmens.
- Mit Hilfe der gedrückt gehaltenen linken Maustaste wird das Steuerelement verschoben. Sobald die Maustaste losgelassen wird, wird das Steuerelement an die neue Position verschoben.
- Andere Möglichkeit: Einstellung der Werte Top und Left.

## Raster nutzen

- Innerhalb der UserForm können mit Hilfe von Rasterpunkten die Steuerelemente ausgerichtet werden.
- Mit Hilfe der Registerkarte *Allgemein* des Menübefehls *Extras – Optionen*
  - ... werden die Abstände zwischen den Rasterpunkten eingestellt.
  - ... wird das Raster aus- oder eingeblendet.
  - ... wird die Ausrichtung an den Rasterpunkten aktiviert oder deaktiviert.

## Bezeichnungsfeld (Label)

- ... zeigt einen beliebigen Text an einer bestimmten Stelle an.
- ... gibt eine Hilfestellung zum Ausfüllen von Textfeldern.
- ... bietet Informationen zur Aufgabe des Dialogfeldes an.

## Eigenschaften

- Caption enthält den anzuzeigenden Text.
- AutoSize passt die Größe des Steuerelements dem Text entsprechend an.
- TextAlign legt die Textausrichtung fest. Standardmäßig wird der Text links ausgerichtet.

## Textfeld (TextBox)

- Eingabefeld für den Benutzer.
- Der Benutzer kann Daten eingeben oder verändern.
- Das Ergebnis einer Berechnung wird angezeigt.
- ... stellt Buchstaben, Ziffern etc. dar.
- ... sind standardmäßig einzeilig.

## TipText einblenden

- ControlTipText enthält eine Hilfe zum Ausfüllen des Textfeldes oder zur angezeigten Information.
- Sobald sich die Maus über dem Textfeld befindet, wird der Text in einem gelben Erklärfenster angezeigt.

## Anzahl von Zeichen begrenzen

- `MaxLength` begrenzt die Anzahl der Zeichen in einem Textfeld. Zum Beispiel wird für die Eingabe von Postleitzahlen die Länge des Textes auf fünf Zeichen begrenzt.
- Falls die Eigenschaft `AutoTab` den Wert `True` hat und die maximale Anzahl von Zeichen eingegeben ist, wird automatisch in das nächste Steuerelement gewechselt.

## Steuerelement sperren

- Wenn die Eigenschaft `Enabled`
  - ... den Wert `True` hat, kann das Steuerelement ausgewählt werden. Der Nutzer kann Text eingeben oder vorhandenen Text verändern.
  - ... den Wert `False` hat, wird das Steuerelement leicht grau dargestellt. Das Steuerelement kann nicht ausgewählt werden.
- Wenn die Eigenschaft `Locked`
  - ... den Wert `True` hat, kann der Inhalt des Steuerelements kopiert, aber nicht bearbeitet werden. Das Steuerelement kann den Fokus bekommen. Das Steuerelement ist gesperrt.
  - ... den Wert `False` hat, kann der Inhalt des Steuerelements bearbeitet werden.

## Mehrzeilige Textfeld

- Mit Hilfe der Eigenschaft `MultiLine` kann ein mehrzeiliger Text eingegeben werden.
- Die Eigenschaft `WordWrap` bricht automatisch die Zeilen in Abhängigkeit der Größe des Textfeldes um.

## Zeilenumbruch und Tabulator

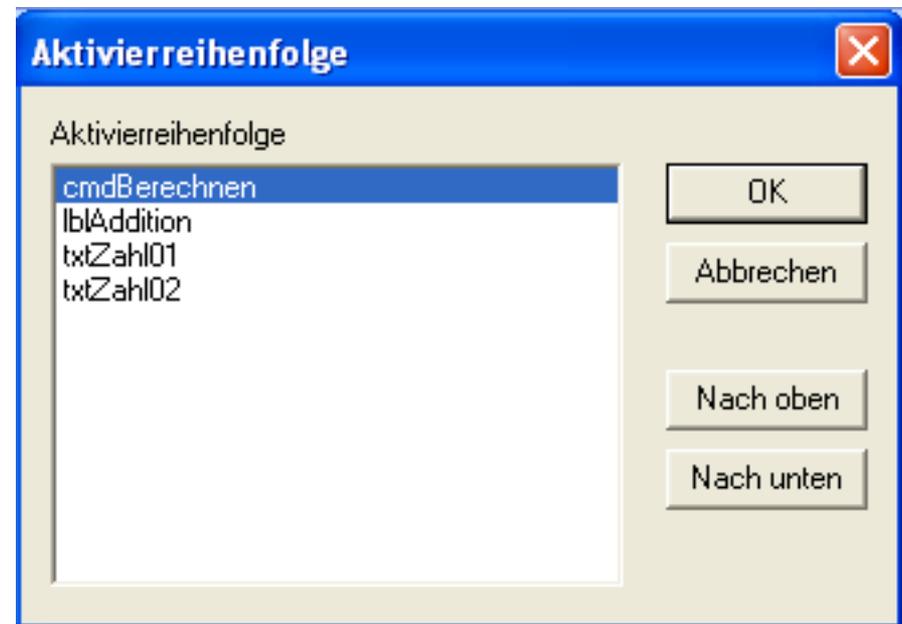
- Standardmäßig wird ein Zeilenumbruch in einem mehrzeiligen Textfeld mit Hilfe von <STRG>+<ENTER> erzeugt. Mit Hilfe der Eigenschaft `EnterKeyBehaviour` ist es möglich nur die <ENTER>-Taste für einen Zeilenumbruch zu nutzen.
- Standardmäßig wird mit Hilfe von <TAB> das nächste Steuerelement aktiviert. Mit Hilfe der Eigenschaft `TabKeyBehaviour` können Tabulatoren in einem mehrzeiligen Textfeld gesetzt werden.

## ... zur Eingabe eines Passwortes nutzen

- Mit Hilfe der Eigenschaft PasswordChar wird ein Zeichen zur Maskierung der vom Benutzer eingegebenen Zeichen festgelegt.
- Die eingegebenen Zeichen werden durch den Wert der Eigenschaft PasswordChar ersetzt.

## Tabulator-Taste zur Auswahl eines Steuerelements

- Mit Hilfe der Tabulator-Taste können die Steuerelemente in einer bestimmten Reihenfolge durchlaufen werden.
- Die Eigenschaft `TabIndex` gibt die Reihenfolge innerhalb der UserForm an.
- Mit Hilfe von *Ansicht – Aktivierreihenfolge* kann die Reihenfolge verändert werden.



## Befehlsschaltflächen (CommandButton)

- ... lösen auf Mausklick immer eine Aktion aus.
- Die Aktion wird mit Hilfe eines Ereignisses abgefangen.
- Beispiele für Aktionen:
  - Berechnung starten.
  - Dialogfeld öffnen oder schließen.
- ... sollten immer eine Beschriftung in Verbform mit großen Anfangsbuchstaben bekommen. Das Verb beschreibt die angestoßene Aktion.

## Beschriftung

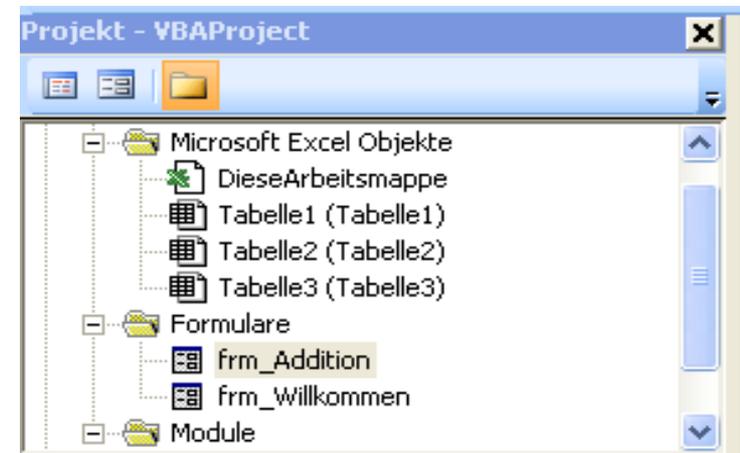
- Caption definiert die Beschriftung auf der Schaltfläche.
- Picture nutzt ein Icon für die Erläuterung der Aktion. Mit Hilfe von PicturePosition kann das Bild auf der Schaltfläche positioniert werden.

## Standard-Schaltflächen

- Die Eigenschaft `Default` wird auf `True` gesetzt. Das Ereignis `[button]_Click()` wird auch durch die `<RETURN>`-Taste ausgelöst.
- Die Eigenschaft `Cancel` wird auf `True` gesetzt. Das Ereignis `[button]_Click()` wird auch durch die `<ESC>`-Taste ausgelöst
- Ok- und Abbrechen-Schaltflächen sind jeweils nur einmal in einem Dialogfeld vorhanden.

## UserForm im Projekt-Explorer

- Im Ordner Formulare befinden sich alle benutzerdefinierten UserForms.
- Die Layout-Ansicht wird durch einen Doppelklick auf die Bezeichnung geöffnet. In der Layout-Ansicht wird das Aussehen des Dialogfeldes, die Positionierung der Steuerelemente etc. festgelegt.

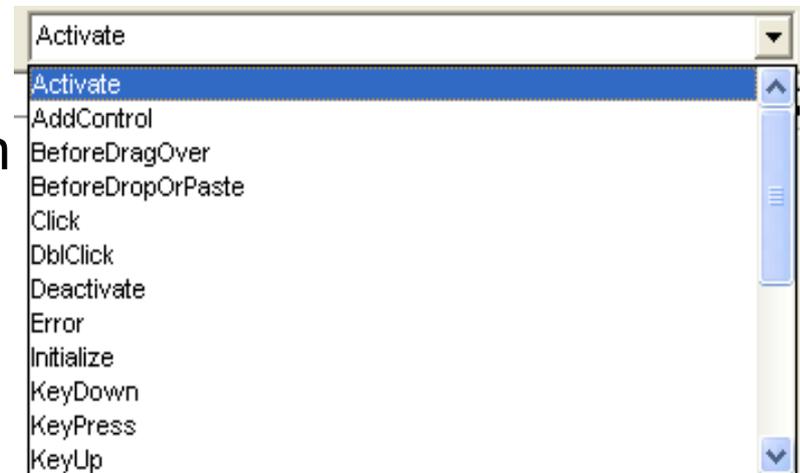


## Codefenster öffnen

- Voraussetzung: Ein Objekt ist im Projekt-Explorer ausgewählt
- 1. Möglichkeit:
  - Mit Hilfe der rechten Maustaste wird das dazugehörige Kontextmenü geöffnet.
  - Der Befehl *Code anzeigen* im Kontextmenü wird angeklickt. Das passende Codefenster wird geöffnet.
- 2. Möglichkeit:
  - Mit Hilfe von *Ansicht – Code* oder <F7> wird das dazugehörige Codefenster angezeigt.

## Codefenster einer UserForm

- Das Codefenster einer UserForm entspricht dem Codefenster eines Standardmoduls.
- Am oberen Rand befinden sich zwei Kombinationsfelder.
  - Im rechten Kombinationsfenster werden alle Steuerelemente und die UserForm aufgelistet.
  - In Abhängigkeit des ausgewählten Elements werden im linken Kombinationsfeld die dazugehörigen Ereignisse angezeigt.



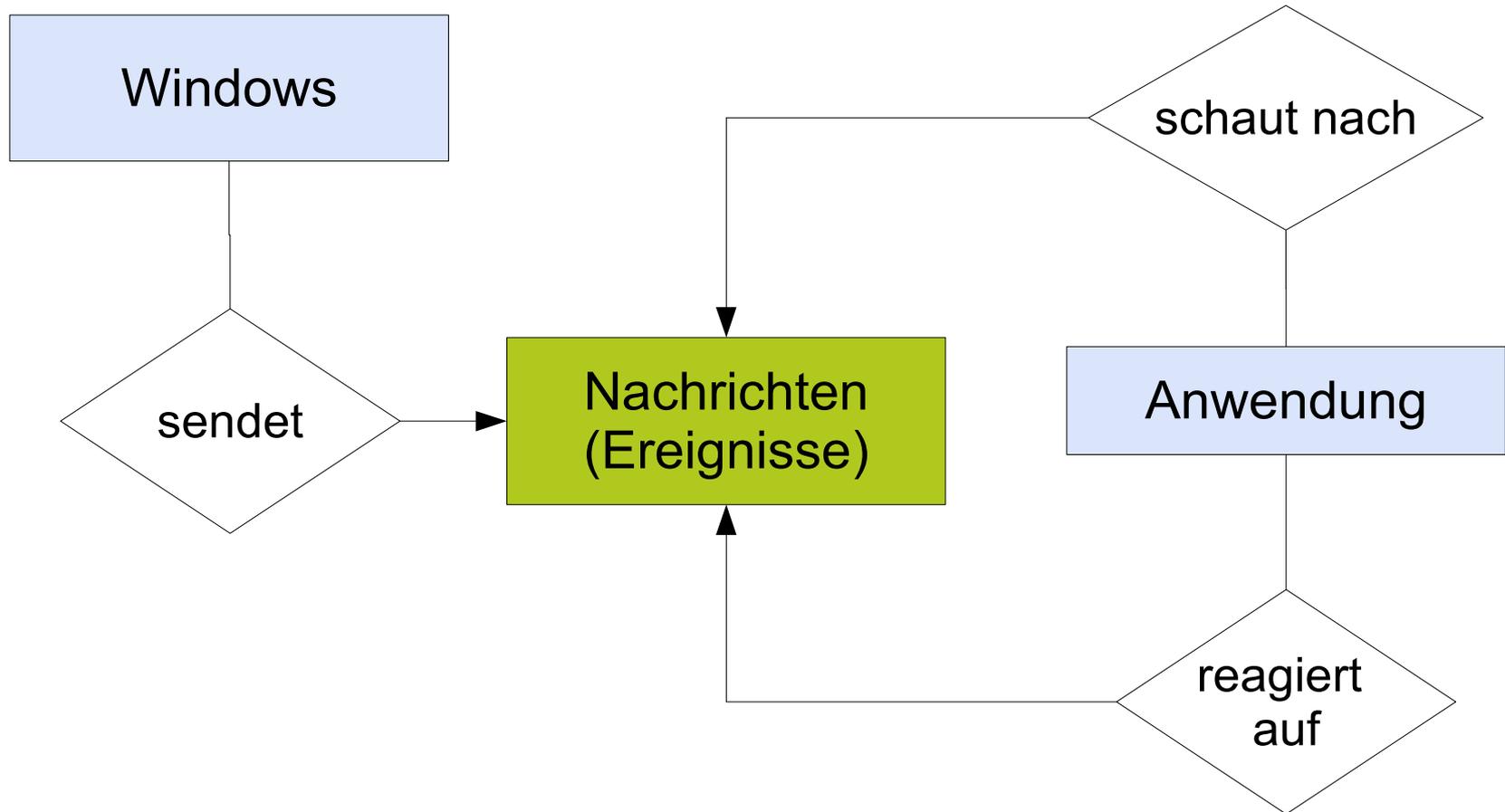
## Ereignis (Event)

- ... tritt immer in Verbindung mit einem Objekt auf.
- ... wird vom Nutzer des Dialogfeldes oder des Steuerelements ausgelöst.
- ... ist eine Reaktion auf eine Benutzeraktion.
- ... ist eine von einem Objekt gesendete Nachricht.

## Beispiele

- Die Größe des Dialogfeldes wird verändert.
- Nach der Eingabe einer bestimmten Anzahl von Zeichen wird das Textfeld automatisiert verlassen
- In ein Textfeld werden Zeichen mit Hilfe der Tastatur eingegeben und auf Richtigkeit überprüft.
- Mit Hilfe eines Klicks auf eine Schaltfläche wird eine Berechnung gestartet.

# Nachrichtenschleife



## Ereignis auswählen

- Mit einem Doppelklick wird
  - ... automatisch die Prozedur `Click()` für die UserForm, Schaltflächen, Optionsfelder, Kontrollkästchen und Listenfelder erstellt. Die Objekte reagieren standardmäßig auf einen Mausklick.
  - ... automatisch die Prozedur `Change()` für Textfelder erstellt.
  - ... in implementierte Ereignisse gesprungen, die mit dem ausgewählten Objekt verbunden sind.
- Im Codefenster wählen Sie aus dem linken Kombinationsfeld das gewünschte Objekt und im rechten Kombinationsfeld das gewünschte Ereignis aus. Auf das Ereignis wird mit Hilfe von Code reagiert.

## Ereignisprozeduren (Event-Handler)

Private Sub UserForm\_Initialize()

- ... fangen ein bestimmtes Ereignis ab.
- ... reagieren mit Hilfe von Anweisungen auf das ausgelöste Ereignis.
- ... enthalten häufig keine Anweisungen. Der Aufruf des Ereignisses bleibt ohne Folgen.

## Informationen im Namen des Event-Handlers

Private Sub UserForm\_Initialize()

- Der Name eines Event-Handlers besteht
  - ... aus dem Objekt, welches das Ereignis ausgelöst hat
  - ... und dem Ereignis, welches von der Maus oder Tastatur ausgelöst wurde.
  - Objekt und Ereignis werden durch ein Unterstrich getrennt.

## Aufbau des Event-Handlers

```
Private Sub UserForm_Initialize()
```

```
End Sub
```

- Ein Event-Handler
  - ... beginnt mit Sub und endet mit End Sub.
  - ... ist eine Prozedur, die keinen Wert zurück gibt.
- Als Zugriffsmodifizierer wird immer Private angegeben. Die Prozedur ist nur innerhalb des Moduls bekannt, in der die Prozedur definiert ist. Die Prozedur ist an ein Objekt gebunden.
- In Abhängigkeit des Ereignisses können der Prozedur Werte übergeben werden.

## Steuerelemente initialisieren

```
Private Sub UserForm_Initialize()  
    txtArtikelNr.Value = ""  
    txtArtikelName.Value = ""  
    txtArtikelPreis.Value = ""  
End Sub
```

Das Ereignis wird nach dem Laden eines Formulars, aber noch vor der Anzeige ausgelöst.

## Steuerelementname

- [controlName].[Eigenschaft] oder [controlName].[Methode]
  - Das Steuerelement wird direkt über den Eigenschaftswert Name angesprochen.
  - Jedes Steuerelement hat Eigenschaften oder Methoden. Das Objekt und die Eigenschaften / Methoden werden durch ein Punkt getrennt.
- Me.Controls([controlName]).[Eigenschaft] oder Me.Controls([controlName]).[Methode]
  - Me bezeichnet die zu dem Code gehörende UserForm.
  - Controls ist eine Auflistung aller Steuerelemente auf der UserForm.
  - In runden Klammern wird der Name als String übergeben.

## Inhalt eines Textfeldes

- `.Value` enthält den gespeicherten Wert des Textfeldes. Die Eigenschaft kann nur im Code, der zur UserForm gehört, genutzt werden.
- `.Text` enthält den Text eines Steuerelements. Die Eigenschaft kann nur genutzt werden, wenn das Textfeld den Fokus hat. Die Eigenschaft kann einen anderen Wert besitzen als `.Value`.

## Klick auf eine Schaltfläche

```
Private Sub cmdSpeichern_Click()  
    Dim zeile As Integer  
    Dim wareSheet As Worksheet  
    Dim zelle As Range  
  
    zeile = freeSpalte  
    Set wareSheet = ActiveWorkbook.Worksheets(1)  
    With wareSheet.Cells(zeile, 1)  
        .Font.Size = 16  
        .Font.Name = "Arial"  
        .Font.Color = vbRed  
        .Value = txtArtikelNr.Value  
    End With  
End Sub
```

Mausklick auf eine  
Schaltfläche,  
UserForm etc..

## Tastatureingaben abfangen

```
Private Sub txtArtikelNr_KeyPress  
    (ByVal KeyAscii As MSForms.ReturnInteger)  
  
    If KeyAscii > 47 And KeyAscii < 58 Then  
        KeyAscii = 0  
    End If  
  
    If KeyAscii > 96 And KeyAscii < 123 Then  
        KeyAscii = KeyAscii - 32  
    End If  
  
    If KeyAscii > 64 And KeyAscii < 91 Then  
        SendKeys ("-")  
    End If  
  
End Sub
```

## Event KeyPress

- Die Taste ist gedrückt.
- Als Parameter wird der ASCII-Code eines Zeichens an das Ereignis übergeben.
- Mit Hilfe von `KeyAscii = 0` kann das Ereignis unterdrückt werden. Das Zeichen wird nicht am Bildschirm angezeigt.
- ASCII-Zeichentabellen:
  - <http://www.torsten-horn.de/techdocs/ascii.htm>
  - [http://de.wikipedia.org/wiki/American\\_Standard\\_Code\\_for\\_Information\\_Interchange](http://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange)
  - [http://de.wikibooks.org/wiki/C-Programmierung:\\_ASCII-Tabelle](http://de.wikibooks.org/wiki/C-Programmierung:_ASCII-Tabelle)

## Tastatur-Events

- `KeyDown()`. Die Taste wird gedrückt.
  - Die gedrückte Taste wird als Windows-Spezifischer Code übergeben.
  - Es kann überprüft werden, ob die `<UMSCHALT>`, `<STRG>` oder die `<ALT>`- Taste gleichzeitig gedrückt wurde.
- `KeyPress()`. Die Taste ist gedrückt.
- `KeyUp()`. Die Taste wird losgelassen.
  - Die gedrückte Taste wird als Windows-Spezifischer Code übergeben.

## SendKeys

- ... simuliert Eingaben des Benutzers in das aktive Steuerelement.
- ... schreibt ein Zeichen in das aktive Dialogfeld.
- ... bekommt als Parameter eine Zeichenkette übergeben.
- Optional kann ein Wartemodus eingestellt werden. Nach dem Senden der Taste wird standardmäßig die Prozedur fortgesetzt.
- <UMSCHALT>, <ALT> und <STRG> werden häufig gleichzeitig mit einer Taste gedrückt.
  - SendKeys("+{a}"). <UMSCHALT>+<a>.
  - SendKeys("%{F1}"). <ALT>+<F1>.
  - SendKeys("^ {c}"). <STRG>+<c>.

## Validierung nach dem Verlassen

```
Private Sub txtArtikelPreis_Exit  
    (ByVal Cancel As MSForms.ReturnBoolean)  
    Dim tmpPreis As String  
    Dim pos As Integer  
  
    tmpPreis = txtArtikelPreis.Value  
    pos = InStr(tmpPreis, ".")  
  
    If pos > 0 Then  
        tmpPreis = Replace(tmpPreis, ".", ",")  
    End If  
  
    txtArtikelPreis.Value = tmpPreis  
End Sub
```

## Tastatureingaben abfangen

```
Private Sub txtArtikelPreis_Exit(  
    ByVal Cancel As MSForms.ReturnBoolean)  
    Dim tmpPreis As String  
    Dim pos As Integer  
  
    tmpPreis = txtArtikelPreis.Value  
    pos = InStr(tmpPreis, ".")  
  
    If pos > 0 Then  
        tmpPreis = Replace(tmpPreis, ".", "")  
    End If  
  
    txtArtikelPreis.Value = tmpPreis  
End Sub
```

Cancel = True unterbricht das Ereignis. Der Fokus bleibt in dem Steuerelement txtArtikel. Andernfalls wird zum nächsten Steuerelement gewechselt. txtArtikel verliert den Fokus.

## Fokus

- Das Steuerelement oder das Fenster sind aktiv.
- In einem Textfeld ist die Einfügemarke sichtbar.
- Mit Hilfe der Methode `SetFocus` wird der Fokus auf ein bestimmtes Element gesetzt.

## Änderungen des Inhalts

- Change()
  - Die Eigenschaft Value eines Steuerelements verändert sich.
- BeforeUpdate()
  - Bevor Daten geändert werden.
- AfterUpdate()
  - Nachdem Daten geändert wurden.

## Optionsfelder (RadioButton)

- ... bieten eine Auswahl von Elementen an.
- Der Benutzer kann nur ein Element aus einer Gruppe auswählen.
- Beispiel:
  - Der Kunde bestellt ein Pullover. Der Pullover wird in verschiedenen Größen angeboten. Der Kunde entscheidet sich aber für die passende Größe.
  - Ein Fahrrad kann in vielen verschiedenen Farben geliefert werden. Das Fahrrad des Kunden hat aber nur exakt eine Farbe.

## Wert eines Optionsfeldes

- Ein Optionsfeld ist entweder True (schwarzer Kreis im weißen Kreis) oder False.
- Value speichert den boolschen Wert.
- Es ist immer nur ein Optionsfeld innerhalb des aktiven Dialogfeldes oder der Optionsgruppe aktiv.

## Optionsgruppen

- Mit Hilfe der Eigenschaft `GroupName` können Optionsgruppen zusammengefasst werden. Alle Optionsfelder, die den gleichen Gruppennamen besitzen, gehören zu einer Gruppe. Innerhalb der Gruppe ist immer nur ein Optionsfeld aktiv.
- Andere Möglichkeit:
  - Das Steuerelement "Rahmen" wird in die UserForm eingefügt. Hinweis: Ein Rahmen hat keinen transparenten Hintergrund!
  - Innerhalb des Rahmens werden die gewünschten Optionsfelder abgelegt. Der Rahmen ist ein Container für die Optionsfelder. Die Optionsfelder werden zusammengefasst.
  - Vorteil: Optische Zuordnung der Optionsfelder zu einer Gruppe.

## Kontrollkästchen (CheckBox)

- ... bietet eine Auswahl von Elementen zu einem Thema.
- Der Benutzer kann beliebig viele oder keines aktivieren.
- Beispiel: "Bestellung von Sonderzubehör"
  - Der Kunde bestellt kein Sonderzubehör. Das Auto ist trotzdem betriebsbereit.
  - Ein weiterer Kunde bestellt ein Schiebedach. Der Kunde nutzt nur eine Teil aus der Liste "Sonderzubehör".
  - Ein Kunde bestellt eine Sonderlackierung und getönte Scheiben. In Abhängigkeit der Wünsche des Kunden werden die Sonderteile bestellt.

## Steuerelemente ein- oder ausblenden

```
Private Sub chkBestellen02_Click()  
    If chkBestellen02 = True Then  
        lblBestellAnzahl02.Visible = True  
        lblBestellGesamt02.Visible = True  
        txtBestellAnzahl02.Visible = True  
        txtBestellGesamt02.Visible = True  
    Else  
        lblBestellAnzahl02.Visible = False  
        lblBestellGesamt02.Visible = False  
        txtBestellAnzahl02.Visible = False  
        txtBestellGesamt02.Visible = False  
        txtBestellAnzahl02.Value = ""  
        txtBestellGesamt02.Value = ""  
    End If  
End Sub
```

Das Steuerelement wird  
eingebildet (True)  
oder ausgeblendet  
(False).

## Listenfeld (ListBox)

- ... ist eine Auflistung von Elementen.
- ... stellt verschiedene Möglichkeiten in Form einer Liste dar.
- ... bietet die Möglichkeit mehr als ein Element aus der Liste auszuwählen. Eine Mehrfachauswahl ist aber für den Benutzer sehr verwirrend.
- ... in Kombination mit einem Textfeld wird es als Kombinationsfeld (ComboBox) bezeichnet.

## Spalten eines Listenfeldes

- `ColumnCount`. Anzahl Spalten in einem Listenfeldes.
- `ColumnWidths`. Spaltenbreite in Form eines Strings. Für jede Spalte wird eine Breite angegeben. Mit Hilfe von Null wird eine Spalte ausgeblendet.
- `ColumnHeads`. Überschriften der Spalten in einem Listenfeld.

## Mehrspaltiges Listenfeld

```
Private Sub UserForm_Initialize()
```

```
    With lstArtikelAuswahl
```

```
        .ColumnCount = 2
```

```
        .ColumnWidths = "4cm;1 cm"
```

```
        .BoundColumn = 0
```

```
    End With
```

```
End Sub
```

Für jede Spalte wird eine Breite angegeben. Die Angaben werden durch ein Komma getrennt.

Welche Spalte wird für die Eigenschaft `.Value` genutzt?  
`.Value` enthält das ausgewählte Element der Liste. Hier wird die erste Spalte genutzt.

## Elemente der Liste hinzufügen

- `[objekt].AddItem [element], [index]`
  - Ein Element wird am Ende der Liste eingefügt.
  - Mit Hilfe des optionalen Parameters `[index]` kann die Position innerhalb der Liste angegeben werden.
  - In einer mehrspaltigen Liste wird der einzufügende Wert in der ersten Spalte abgelegt.
- `[objekt].Column([spalte], [zeile]) = [element]`
  - ... liest oder schreibt ein Element in eine bestimmte Zeile und Spalte.
  - Die Angaben für die Zeile und Spalte sind optional.
- `[objekt].List() = [range].Value`
  - ... wird genutzt, um ein zweidimensionales Feld in die Liste einzufügen.

## Mehrspaltiges Listenfeld füllen

```
Private Sub UserForm_Initialize()
```

```
With lstArtikelAuswahl
```

```
    .ColumnCount = 2
```

```
    .ColumnWidths = "4cm"
```

```
    .BoundColumn = 0
```

```
for index = 0 To Ubound(artikelname)
```

```
    .AddItem
```

```
    .Column(0, index) = index
```

```
    .Column(1, index) = artikelname(index)
```

```
next index
```

```
End With
```

```
End Sub
```

Es muss immer erst eine neue Zeile im Listenfeld erzeugt werden und ...

... und anschließend können die Spalten der neuen Zeile gefüllt werden.

## Ausgewähltes Element

```
Private Sub lstArtikelAuswahl_Click()  
    Dim zeile As Integer  
    Dim wareSheet As Worksheet  
  
    Set wareSheet = ActiveWorkbook.Worksheets("Ware")  
  
    With lstArtikelAuswahl  
        zeile = .Column(1, .ListIndex)  
        txtArtikelNr.Value = wareSheet.Cells(zeile, 1).Value  
        txtArtikelName.Value = .Value  
        txtArtikelPreis.Value = wareSheet.Cells(zeile, 3).Value  
    End With  
End Sub
```

.ListIndex gibt die ausgewählte Zeile eines Listenfeldes an. .ListIndex = -1 hebt die Auswahl auf. Falls eine Mehrfachauswahl möglich ist, muss die Liste vollständig durchlaufen werden.

## Andere Möglichkeit

```
Private Sub lstArtikelAuswahl_Click()  
...  
With lstArtikelAuswahl  
    For count = 0 To .ListCount - 1  
        If .Selected(count) Then  
            zeile = .Column(1, count)  
            txtArtikelNr.Value = wareSl  
            txtArtikelName.Value = .Colu  
            txtArtikelPreis.Value = wareSl  
        End If  
    Next count  
End With  
End Sub
```

Anzahl der  
Elemente in der  
Liste.

Ist das Element  
ausgewählt?

## Elemente löschen

- `[objekt].RemoveItem ([zeile])`
  - `[objekt].RemoveItem (0)` löscht das erste Element in einer Liste.
  - `[objekt].RemoveItem ([objekt].ListCount - 1)` löscht das letzte Element in einer Liste.
  - `[objekt].RemoveItem ([objekt].ListIndex)` löscht das ausgewählte Element in einer Liste. Eine Mehrfachauswahl ist in dieser Liste nicht möglich.
- `[objekt].Clear`
  - ... löscht alle Elemente der Liste.

## Zusätzliche Steuerelemente

- Mit einem Klick der rechten Maustaste auf eine freie Fläche in der Werkzeugsammlung wird das dazugehörige Kontextmenü geöffnet. Wählen Sie den Befehl *Zusätzliche Steuerelemente...* aus.
- Es wird eine Liste mit allen sonstigen Steuerelementen geöffnet.
- Mit einem Klick in das Kontrollkästchen links von der Bezeichnung wird zum Beispiel das Steuerelement Kalender-Steuerelement 12.0 aktiviert.
- *OK* schließt das Fenster und das ausgewählte Steuerelement wird in die Werkzeugsammlung übernommen.

## Beispiel: Kalender

```
Private Sub UserForm_Initialize()  
    myKalender.Value = Date  
End Sub
```

Das aktuelle  
Datum ist im  
Kalender aktiv.

```
Private Sub myKalender_Click()  
    Dim count As Integer
```

```
    For count = 0 To UserForms.count - 1
```

```
        If UserForms(count).Name = "frmBestellung" Then
```

```
            frmBestellung.txtBestelldatum.Value = myKalender.Value
```

```
            frmBestellung.Show
```

```
            Exit For
```

```
        End If
```

```
    Next count
```

```
End Sub
```

Das ausgewählte Datum  
wird in ein Steuerelement  
auf einer anderen  
UserForm geschrieben.