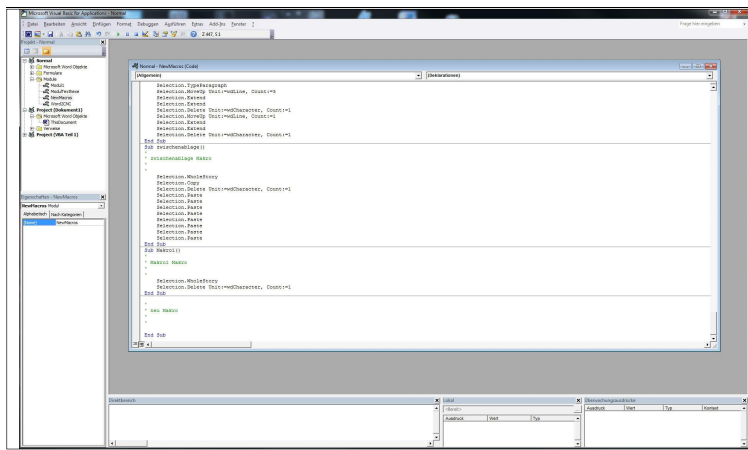
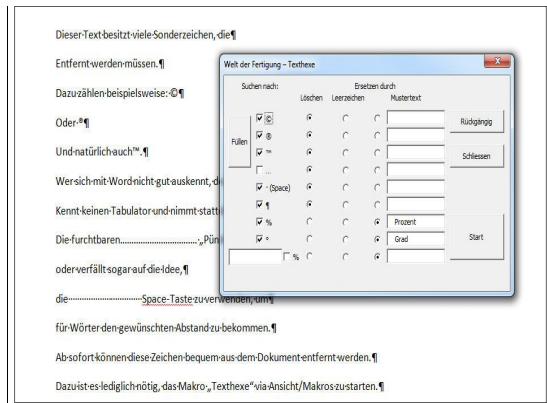




Word mit eigenen Makros noch leistungsfähiger machen

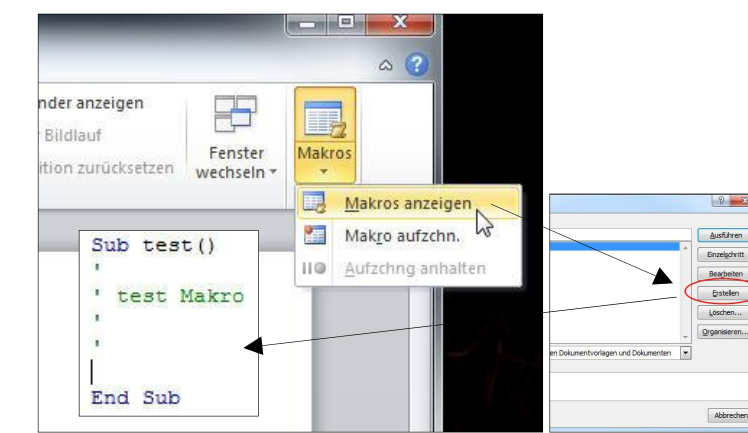
Wer Word oder Excel intensiv nutzt, kann sich über einen Mangel an Befehlen und Funktionalitäten eigentlich nicht beklagen. Dennoch gibt es immer mal wieder Fälle, in denen man gerne einen ganz bestimmten Befehl zusätzlich hätte. Auch bereits vorhandenes findet nicht immer die Zustimmung des Users. Zum Beispiel ist die Suchen und Ersetzen-Funktion von Word alles andere

als einfach handhabbar, wenn es darum geht, etwa bestimmte Sonderzeichen zu entfernen oder Großbuchstaben in Kleinbuchstaben umzuwandeln. In diesen Fällen lohnt ein Griff in die Programmier-Kiste, da via VBA sich rasch entsprechende Funktionen erstellen lassen.



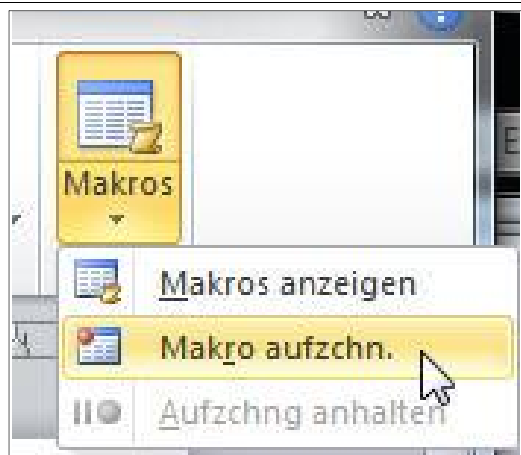
Der VBA-Editor

In den VBA-Editor kommt man nur, indem entweder ein neues Makro erstellt oder ein vorhandenes Makro bearbeitet wird. Die VBA-Entwicklungsumgebung wird zum direkten Erstellen eines Makros per Tastenkombination ALT+F11 aufgerufen.



VBA-Editor starten

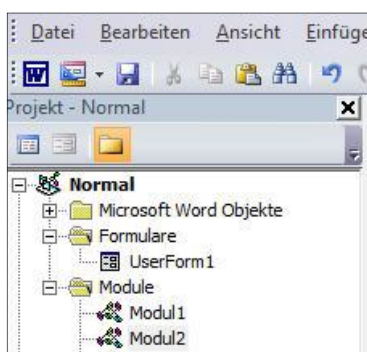
Um ein Makro anzulegen, muss man lediglich über Makros/Makros anzeigen zum Button 'Erstellen' gelangen. Nachdem ein passender Name in das Eingabefeld eingegeben wurde, wird ein noch leeres Makro erzeugt, dem nun Leben eingehaucht werden muss.



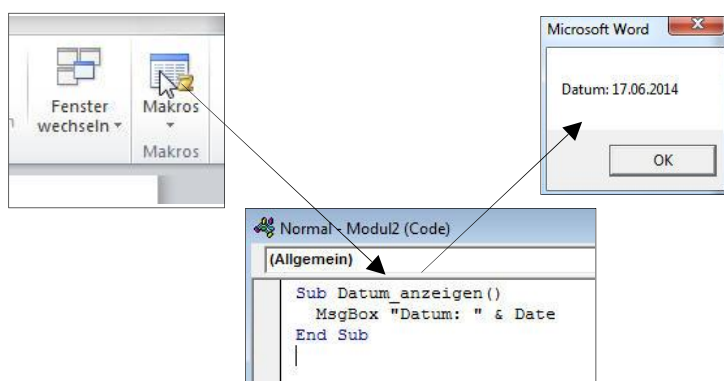
Eine Alternative

Das manuelle Erstellen von VBA-Code ist gerade für Anfänger eine echte Hürde, weshalb dieser Personenkreis gut beraten ist, zunächst vom Makro-Recorder Gebrauch zu machen und von diesem ein Makro aufzeichnen zu lassen, das anschließend manuell verfeinert werden kann. Auf diese Weise lernt man recht schnell den Einsatz der VBA-Befehle kennen.

Der Projektbaum



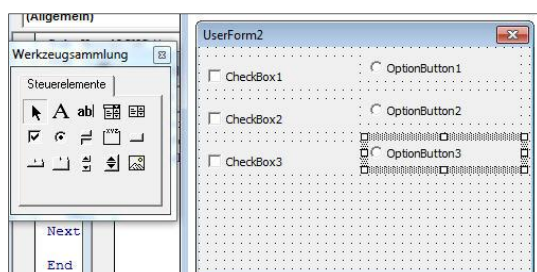
Im Fenster »Projekt« werden alle Makros beziehungsweise die Module und UserForms übersichtlich dargestellt. Auf diese Weise bleibt das Projekt stets handhabbar.



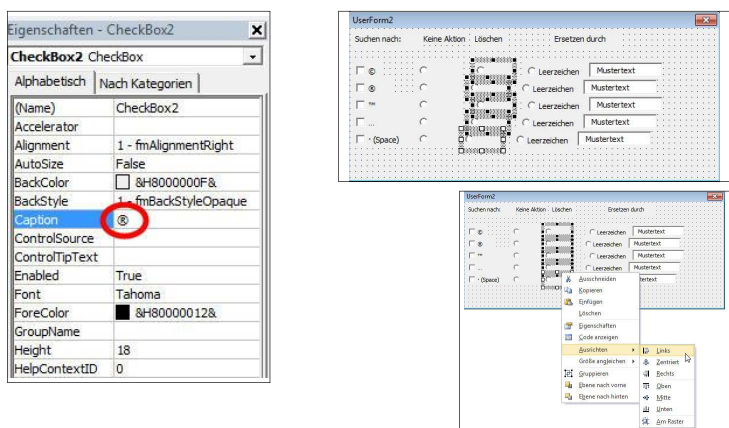
Ein einfaches Makro

Programmieren in VBA ist einfach. Ein Fenster wird beispielsweise mit dem Befehl »MsgBox« aufgerufen. In Anführungszeichen eingefasster Text wird darin angezeigt. Das &-Zeichen sorgt dafür, dass der Datumsbefehl »Date« mit diesem Text verknüpft wird. Gestartet wird das Makro entweder mit der Taste F5 oder durch Betätigen des Buttons »Makro«, Wahl des Makronamens und Druck auf den »Ausführen«-Button.

Eine UserForm

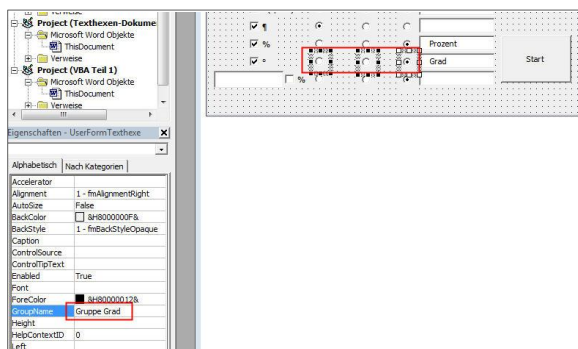


Als UserForm bezeichnet man das vom Programmierer selbst gestaltete Fenster. In dieses lassen sich aus einer Werkzeugsammlung Buttons, Checkboxes und Textfelder einfügen. Auf diese Weise entsteht ein ganz persönliches Bedienfeld, das für eine ganz bestimmte Aufgabe maßgeschneidert ist.



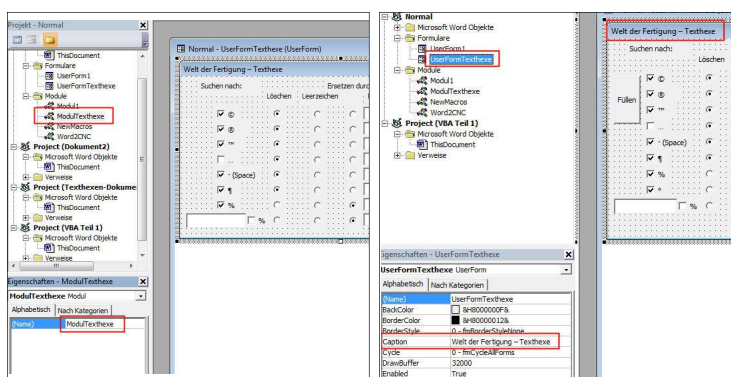
Eigenschaften und Ausrichten

Damit die Elemente in der UserForm sauber ausgerichtet sind, genügt es, diese zu markieren (STRG-Taste gedrückt halten) und per Rechtsklick die gewünschte Ausrichtungsart zu wählen. Eigenschaften, wie Farbe, Sichtbarkeit oder Gruppenzugehörigkeit werden im Eigenschaftenfenster festgelegt.



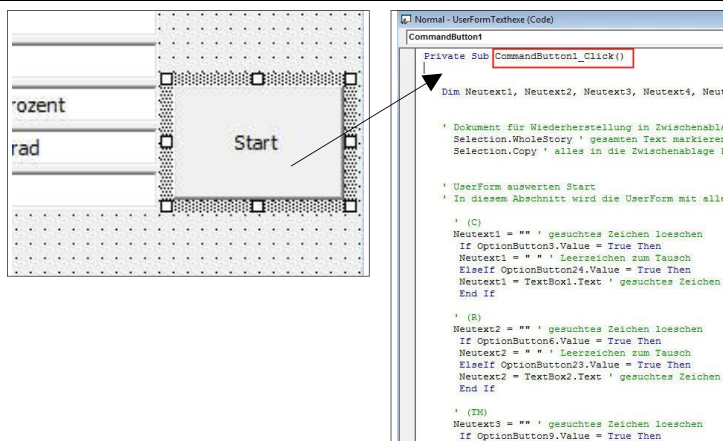
Gruppen bilden

OptionButtons müssen nach Gruppen sortiert werden, wenn nicht alle Buttons zusammengehören. Dazu werden die zusammengehörenden OptionButtons markiert und diesen im Feld Group-Name ein gemeinsamer, nur für diese Gruppe geltender Name zugewiesen. Auf diese Weise wird verhindert, dass ein unberechtigter Button den Fokus bekommt. Derjenige OptionButton, der bei Start des Makros zuerst aktiv sein soll, muss unter ›Value‹ den Wert ›True‹ erhalten. Die anderen Buttons dieser Gruppe bekommen ›False‹.



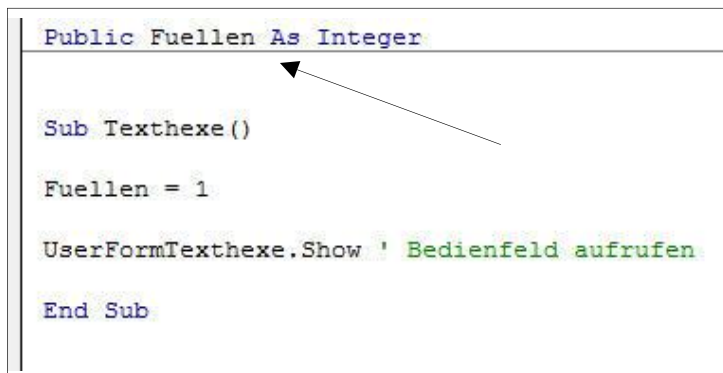
Namen ändern

Damit stets deutlich wird, welches Fenster (User-Form) gerade aktiv ist, muss dieses mit einem aussagekräftigen Text beschriftet werden. Dies geschieht, indem unter ›Formulare‹ die entsprechende UserForm gewählt und anschließend unter ›Caption‹ der gewünschte Text eingegeben wird. Ebenso einfach ist der Modulname zu ändern, was unter ›(Name)‹ vorzunehmen ist.



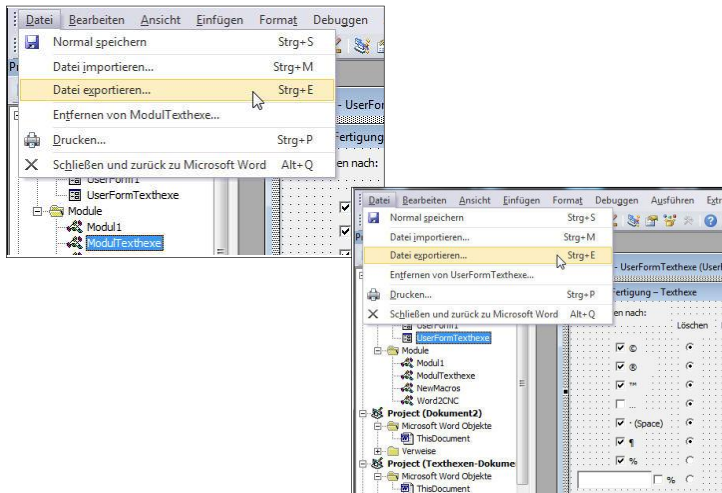
Makro an Button binden

Wenn ein Button betätigt wird, soll in der Regel irgendetwas passieren. Die dafür nötigen Befehle kann man an den Button hängen, indem dieser doppelt angeklickt wird. Dadurch wird ein Makro-Ruf erzeugt, in den die noch fehlenden VBA-Befehle geschrieben werden.



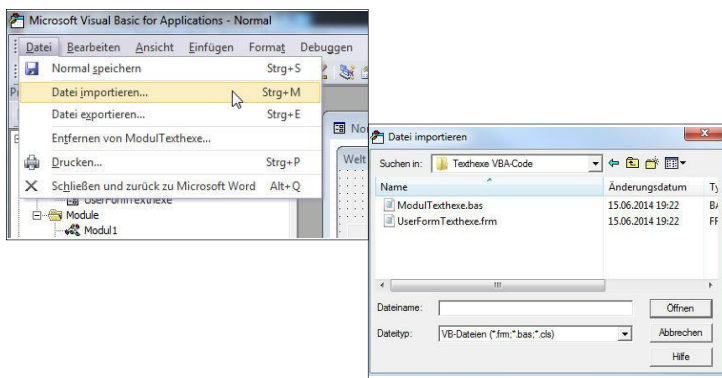
Globale Variablen

Variablen können lokal oder global wirken. In der Regel sollten Variablen stets lokal wirken, damit Seiteneffekte vermieden werden. In manchen Fällen ist dies nicht möglich, etwa bei Zustandsmerkern, die von jeder Stelle des Makros abgefragt werden müssen. In diesem Fall sind globale Variablen die richtige Wahl. Die globale Variable steht oberhalb des Hauptmakros des jeweiligen Moduls.



Makro sichern

Damit das Makro auf verschiedenen Rechnern eingesetzt werden kann, muss es exportiert werden. Dies kann direkt auf einen USB-Stick erfolgen. Zum Exportieren ist darauf zu achten, dass sowohl das Modul, als auch die dazugehörige UserForm jeweils markiert werden müssen, ehe sie exportiert werden. Wird dies nicht beachtet, findet womöglich eine falsche Datei den Weg auf den Stick.



Makro auf Fremdrechner übertragen

Nachdem das Makro auf einen Stick oder eine tragbare Festplatte gesichert ist, kann es von dort direkt in den VBA-Editor eines Fremdrechners importiert werden. Der Import ist genauso einfach, wie das Öffnen einer Word-Datei. Man muss nicht einmal auf die Endungen achten, da der VBA-Editor die einzelnen Dateien anhand ihrer Endung erkennt und richtig zuordnet.

```
Private Sub CommandButton4_Click()  
    If Fuellen = 0 Then  
        CheckBox1.Value = False  
        CheckBox2.Value = False  
        CheckBox3.Value = False  
        CheckBox4.Value = False  
        CheckBox5.Value = False  
        CheckBox6.Value = False  
        CheckBox7.Value = False  
        CheckBox8.Value = False  
        CheckBox9.Value = False  
        Fuellen = 1  
    ElseIf Fuellen = 1 Then  
        CheckBox1.Value = True  
        CheckBox2.Value = True  
        CheckBox3.Value = True  
        CheckBox4.Value = True  
        CheckBox5.Value = True  
        CheckBox6.Value = True  
        CheckBox7.Value = True  
        CheckBox8.Value = True  
        CheckBox9.Value = True  
        Fuellen = 0  
    End If  
End Sub
```

```
Public Fuellen As Integer ' Globale Variable  
  
Sub Texthexe()  
    Fuellen = 0 ' Startwert fuer Fuellen-Button  
    UserFormTexthexe.Show ' Bedienfeld aufrufen  
End Sub
```

```
Private Sub CommandButton2_Click()  
    Selection.Paste ' Inhalt aus Zwischenablage  
    Selection.Delete ' Zwischenablage leeren  
    Set Leerdaten = New DataObject  
    Variable = ""  
    Leerdaten.SetText Variable  
    Leerdaten.PutInClipboard  
End Sub
```

```
Private Sub CommandButton3_Click()  
    Unload Me  
End Sub
```

Welt der Fertigung - Texthexe

Suchen nach:	Löschen	Leerzeichen	Ersetzen durch	Mustertext
<input checked="" type="checkbox"/> Fuellen	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/> %	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/> \	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/> . (Space)	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/> ¶	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/> %	<input type="checkbox"/>	<input type="checkbox"/>	Prozent	
<input checked="" type="checkbox"/> °	<input type="checkbox"/>	<input type="checkbox"/>	Grad	

Normal - UserFormTexthexe (Code)

```
Private Sub CommandButton1_Click()  
    Dim Neutext1, Neutext2, Neutext3, Neutext4, Neutext5  
  
    ' Dokument für Wiedherstellung in Zwischenablage  
    Selection.WholeStory ' gesamten Text markieren  
    Selection.Copy ' alles in die Zwischenablage k...  
  
    ' UserForm auswerten Start  
    ' In diesem Abschnitt wird die UserForm mit alle...  
    (C)  
    Neutext1 = "" ' gesuchtes Zeichen loeschen  
    If OptionButton3.Value = True Then  
        Neutext1 = "" ' Leerzeichen zum Tausch  
    ElseIf OptionButton24.Value = True Then  
        Neutext1 = TextBox1.Text ' gesuchtes Zeichen  
    End If  
  
    (R)  
    Neutext2 = "" ' gesuchtes Zeichen loeschen  
    If OptionButton6.Value = True Then  
        Neutext2 = "" ' Leerzeichen zum Tausch  
    ElseIf OptionButton23.Value = True Then  
        Neutext2 = TextBox2.Text ' gesuchtes Zeichen  
    End If  
  
    (ES)  
    Neutext3 = "" ' gesuchtes Zeichen loeschen  
    If OptionButton9.Value = True Then
```

- Dieser-Text-besitzt-viele-Sonderz
- Entfernt-werden-müssen.¶
- Dazu-zählen-beispielsweise:Ⓢ¶
- Oder-¶¶
- Und-natürlich-auch™.¶
- Wers-ich-mit-Word-nicht-gut-aus
- Kennt-keinen-Tabulator-und-nim
- Die-fürchtbaren.....
- oder-verfällt-sogar-auf-die-Idée,¶
- die.....Space-Tast
- für-Wörter-den-gewünschten-Ab
- Ab-sofort-können-diese-Zeichen-
- Dazu-ist-es-lediglich-nötig,-das-M