

# Excel Makros Programmieren

<http://paranoia.scienceontheweb.net/paranoia/> - [mailto: paranoia @ hush.com](mailto:paranoia@hush.com)

Wed Feb 3 18:20:05 CET 2010 – 22. August 2011

## Inhaltsverzeichnis

<b>1</b>	<b>Intro</b>	<b>2</b>
<b>2</b>	<b>Mein erstes Makro</b>	<b>3</b>
<b>3</b>	<b>Variablen</b>	<b>4</b>
3.1	Deklariieren . . . . .	4
3.2	Variable ausgeben . . . . .	4
3.3	Ausgewählte Variablentypen . . . . .	4
3.4	Typkonversion . . . . .	4
<b>4</b>	<b>Eingabe und Ausgabe</b>	<b>5</b>
4.1	Eingabe . . . . .	5
4.2	Ausgabe . . . . .	5
<b>5</b>	<b>UserForm</b>	<b>6</b>
5.1	Eine Funktion hinter einen Knopf stecken . . . . .	6
5.2	Aufgabe : Taschenrechner . . . . .	7
<b>6</b>	<b>Bedingte Verzweigung</b>	<b>8</b>
<b>7</b>	<b>Funktionen</b>	<b>8</b>
<b>8</b>	<b>Rekursive Funktionen...</b>	<b>9</b>
8.1	Aufgabe produkt() und differenz() . . . . .	10
8.2	Aufgabe max() und min() . . . . .	10
8.3	Aufgabe Taschenrechner02 . . . . .	10
8.4	Taschenrechner02 – Add-ons . . . . .	11
8.5	Na prima, eine Seite für Musterlösung Taschenrechner02 . . . . .	12
<b>9</b>	<b>Zählschleifen mit for und while..wend [optional]</b>	<b>13</b>

**Disclaimer**

Wissen ist zum Teilen da. Ich teile mein Wissen mit Ihnen, lieber Kollege.  
Ich bin aber nicht perfekt. Unter [paranoia@hush.com](mailto:paranoia@hush.com) nehme ich dankbar Ihre  
Verbesserungsvorschläge entgegen.

\*

**Legal Blurb:** Alle Informationen in diesem Dokument sind falsch, unvollständig,  
irreführend, irrelevant und / oder funktionieren einfach nicht.

Wenn Sie es trotzdem benutzen, und es geht dabei etwas kaputt, ist das Ihr Problem,  
nicht meins.

\*

**Bitte teilen Sie meine Web-Adresse nicht Ihren Schülern mit.**

**1 Intro**

Ich will das gar nicht.

Das ist die unreifste, schlechteste Programmiersprache, die ich je gesehen habe.

Sie heißt Visual Basic for Applications (VBA) und ist eine Untermenge von VisualBasic (VB).

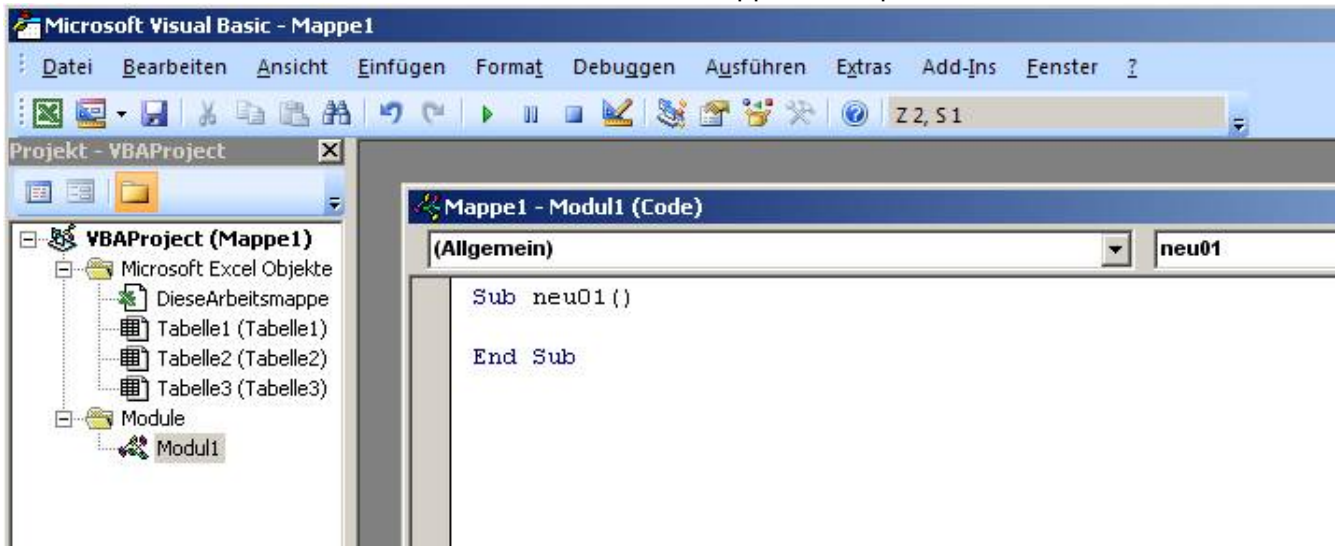
Dennoch scheint es Firmen oder Abteilungen zu geben, die ihre Aufgaben teilweise in lokalen Excel-Versionen mit lokalen Makros zu erledigen versuchen. Vermutlich, weil sie keine andere Programmierumgebung installieren dürfen. Sei's drum.

## 2 Mein erstes Makro

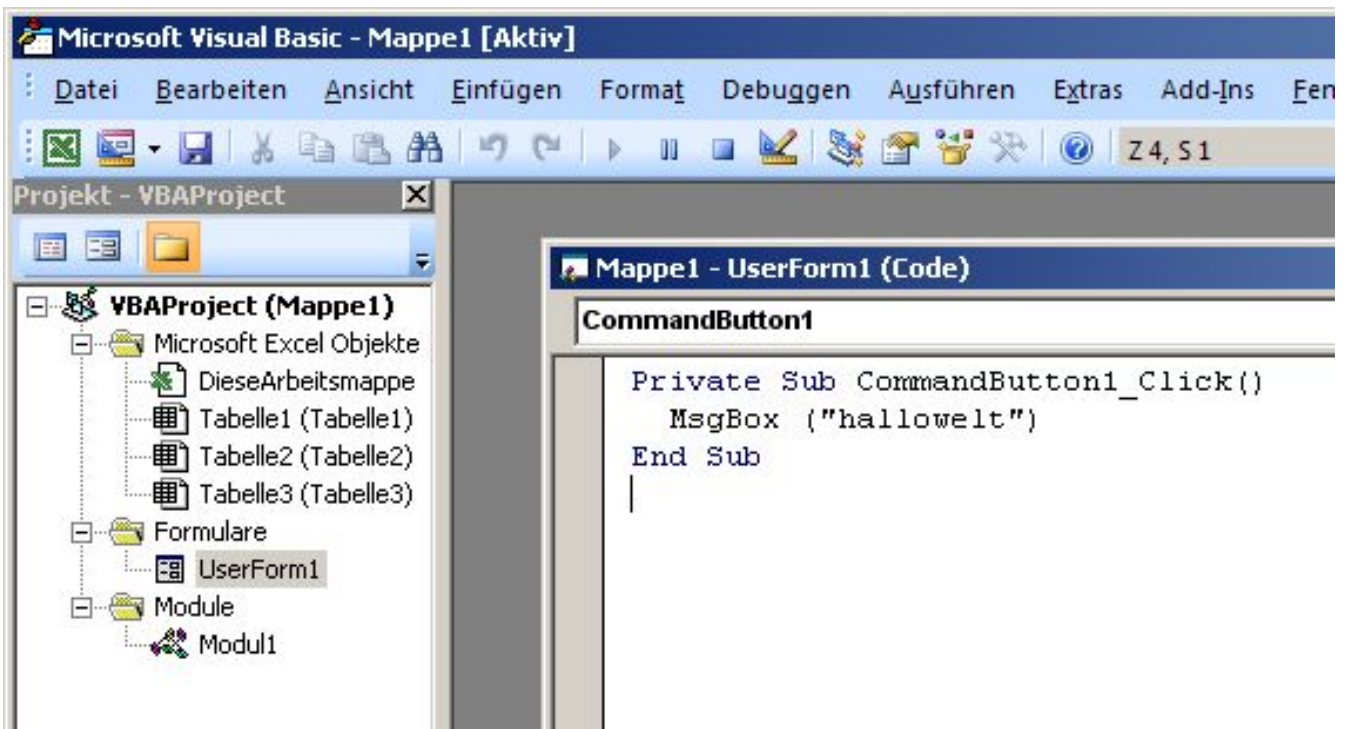
Gehe excel. Öffne Ansicht-Makro-Neu. Nenne es Makro01 oder Test01 oder neu01. Tippe

```
Sub HaloWelt()  
    MsgBox ("Hallo Welt!")  
End Sub
```

Starte es mit GrünesDreieck oder Makro-Starten oder Doppelklick... probiers halt.



Speichere die Tabelle als "TabelleMitMakro", Datentyp "Tabelle mit Makro". Standardmäßig speichert Excel ohne Makros. Standardmäßig öffnen schlaue User ohne Makros.



## 3 Variablen

### 3.1 Deklarieren

Die einfache Methode ist : benutzen.

```
s = "irgendein Text wdovbqpowidv qipsjfv piqjsnv"
```

Die zweit-einfachste ist : den alten Basic-DIM-Befehl verwenden.

```
dim s as String
s = "irgendein Text wdovbqpowidv qipsjfv piqjsnv"
```

### 3.2 Variable ausgeben

Ist gelegentlich nützlich, wenn man erkennen will, wer da wo amokläuft.

```
s = "irgendein Text wdovbqpowidv qipsjfv piqjsnv"
msgbox ("s = " & s)
```

Das in Gänsefüßchen wird wörtlich ausgegeben. Das einzelne s wird als Variable verstanden und ihr Wert ausgegeben.

Das & addiert die beiden, so daß EIN String ausgegeben wird.

### 3.3 Ausgewählte Variablentypen

**Integer** Ganzzahl, d.h. ohne Nachkommastellen.

Der Wertebereich ist von -32768 bis +32767.

**Double** Zahlen mit Nachkommastellen.

-1,79769313486232E308 bis -4,94065645841247E-324 für negative Werte;

**String** Zeichenkette. In Gänsefüßchen. ZB "", "Halli hallo", "db238@.-".

Maximal ungefähr 2 Billionen Zeichen.

Boolean – Wahr oder Falsch.

Es gibt noch mehr; bei Interesse frage g : visual basic variablentypen.

### 3.4 Typkonversion

Manchmal hat man einen String und möchte aber, daß er sich verhält wie eine Zahl. **Lösung:** Multipliziere mit 1.

Manchmal hat man eine Zahl und möchte aber, daß sie sich verhält wie eine Zahl. **Lösung:** Addiere ein Nichts ("").

## 4 Eingabe und Ausgabe

### 4.1 Eingabe

Wir können unsere Eingabewerte aus Zellen holen:

```
dim s as String
s = tabelle1.range("b5").value
```

Oder in unserem Formular (das kommt später) aus der TextBox1:

```
dim i as integer
i = userform1.textbox1.value
dim d as double
d = userform1.textbox2.value
```

### 4.2 Ausgabe

ist wie Eingabe, nur rückwärts:

```
userform1.textbox2.value = a * b      ' gibt das Ergebnis von a*b nach TextBox2
tabelle1.range("b5").value = a + b    ' rate mal
msgbox (a*b)                          ' klappt auch
```

## 5 UserForm

Rechtsklicke Neu-Userform, es erscheint ein häßliches Fenster und ein häßlicher Werkzeugkasten. Klicke auf Label (im Werkzeugkasten) und ziehe ein Label auf (in dem häßlichen Fenster).

Erzeuge auch noch eine Textbox und einen Knopf.

Verändere die Eigenschaften jedes Objekts : markiere es – stelle im Eigenschaften-Fenster links unten etwas um. Du könntest mit Farben und Schriften anfangen.

Erzeuge letztlich ein Label "Namenabfrage", ein Label "Sag mir deinen Namen:", ein Label namens Label3 mit Nix drin, eine Textbox namens TextBox1 und einen Knopf mit der Beschriftung (caption) "los". Hups – Knopf heißt in WinzigweichSprech "Befehlsschaltfläche".

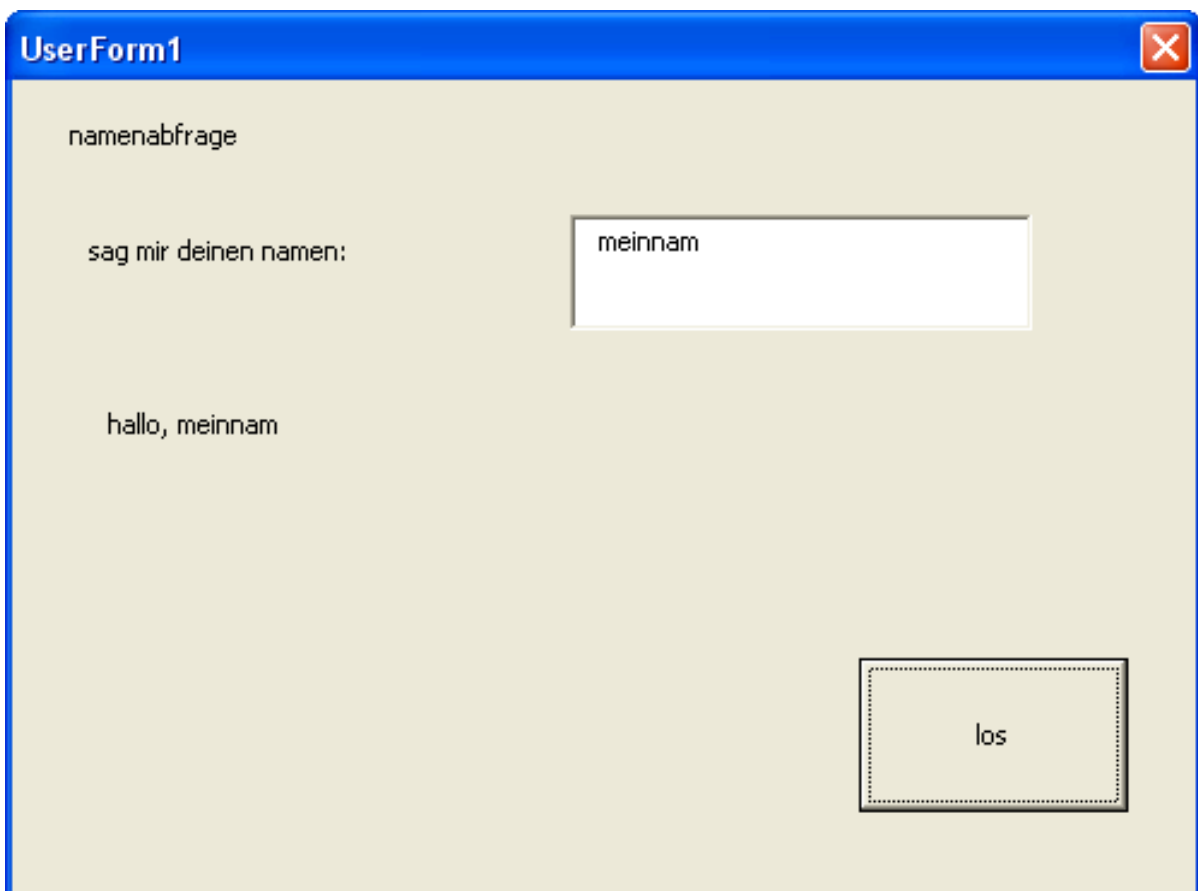
Führe es aus. Alles geht – bis auf den Knopf.

### 5.1 Eine Funktion hinter einen Knopf stecken

Doppelklicke den Knopf und sage:

```
Private Sub CommandButton1_Click()  
    Dim name As String  
    name = TextBox1  
    Label3 = "hallo, " & name  
End Sub
```

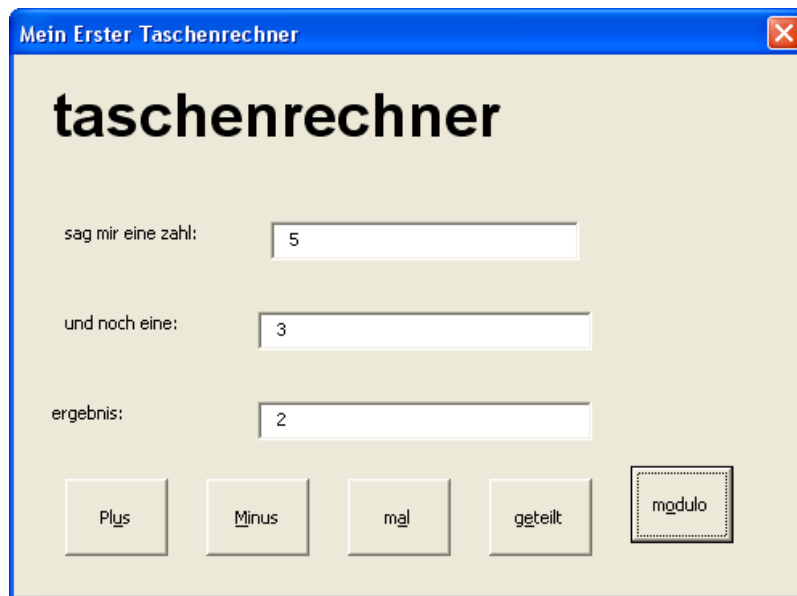
Nun geht's.



Namentliche Begrüßung

## 5.2 Aufgabe : Taschenrechner

Bastele in VBA einen Taschenrechner:



Taschenrechner Ver. 01

Hinweise:

1. Schriftgrößen und alberne Titel und Beschriftungen stellt man im Eigenschaftenfenster ein. Dabei darf das Skript nicht laufen.
2. Für den Plus-Knopf : doppelklicke ihn – hole die Werte und schreibe das Ergebnis.
3. Die anderen Knöpfe sind sehr, sehr ähnlich.
4. modulo gibt den Rest bei der ganzzahligen Division.  
Bsp:  $17 : 4 = 4$  Rest 1. Genau diese 1.  
Bsp:  $5 : 3 = 1$  Rest 2. Genau diese 2.  
Ach so: schreibt man in VBA `mod`.
5. Unterstrichenes U: Eigenschaft `accelerator` (engl: Beschleuniger, auch: shortcut.) Beschleunigt während der Ausführung durch Alt-U.
6. Beim Teilen durch 0 bricht das Script ab. Abhilfe: Statt

```
c = a/b
TextBox15 = c
```

nimm

```
if b=0 then
  c = "teile nicht durch null du loser"
else:
  c = a/b
endif
```

Hilft auch bei Modulo.

## 6 Bedingte Verzweigung

```

if eineBestimmteBedingungWahrIst then           ' kurz
    tue dies
endif

if nochEineBestimmteBedingungWahrIst then      ' lang
    tue dies
    und dies
    und dieses noch
    und dies auch
    und dies nochmal
else:                                           ' sonst
    tue das
    und das
    und das auch noch
endif

```

## 7 Funktionen

Ein Sub kann keine Parameter übernehmen; eine Funktion schon.  
 Ein Sub kann man an einen Knopf binden; eine Funktion nicht.

```

Function summe (a, b)
    summe = a + b
End Function

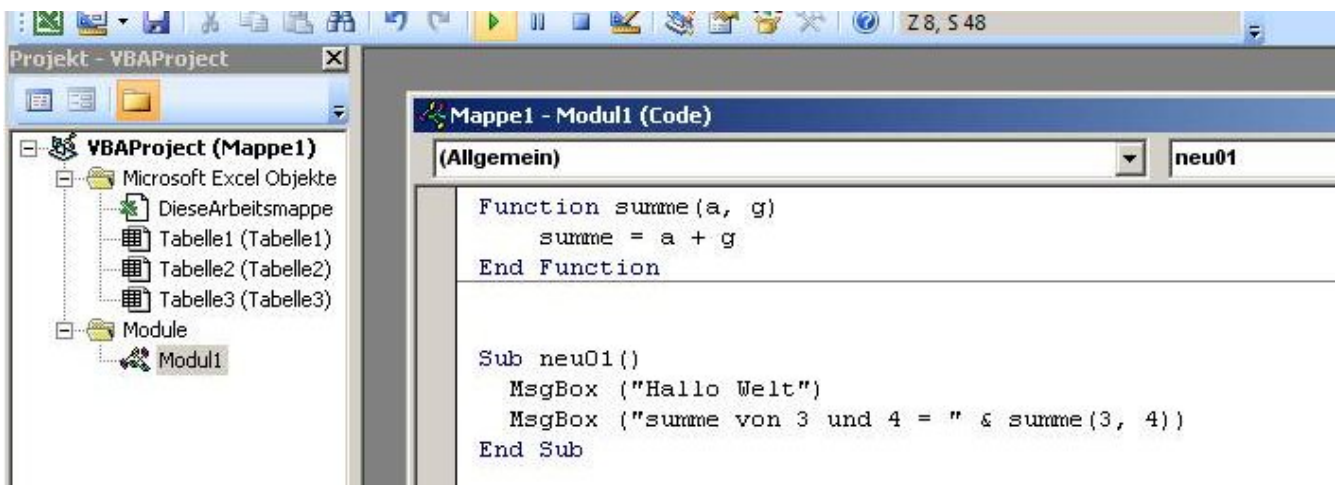
```

```

Sub SummeTest()
    MsgBox ("summe (8, 3) = " & ( summe (8, 3)))
End Sub

```

Die Funktion summe bekommt zwei Parameter, die hier a und b genannt werden – Sie können sie auch Hinz und Kunz nennen – dann müssen Sie aber auch dabei bleiben.



summe() berechnet die Summe und gibt sie zurück. Zurückgeben geht, indem einer nicht existenten Variablen, die so wie die Funktion heißt, das Ergebnis zugewiesen wird.

## 8 Rekursive Funktionen...

... rufen sich selber auf.

Deshalb muß man höllisch aufpassen, daß sie das nicht unendlich oft tun, und eine funktionierende Abbruchbedingung einbauen.

Hier ein Beispiel für Fakultät:

```
Function fact (i)
  ' neugierig ? msgbox("berechne fact(" & i & ").")
  if (i<2) then
    fact = 1
  endif
  fact = i * fact(i-1)
End Function
```

### 8.1 Aufgabe produkt() und differenz()

Schreiben Sie die Funktionen produkt (p1, p2) und differenz (kiks, keks).

### 8.2 Aufgabe max() und min()

max() gibt die größere von beiden Zahlen zurück.

min() gibt die kleinere.

max(max(a,b),c) gibt – was?

### 8.3 Aufgabe Taschenrechner02

Schreiben Sie den verbesserten Taschenrechner02.

TODO : realistischer Screenshot. Ohne hex, bin, etc.

Hinweise:

```
Private Sub TextBox1_Change()
    MsgBox ("textbox1 wurde geändert.")
End Sub
```

```
Private Sub UserForm_Click()
    MsgBox ("User hat Form geklickt.")
    ' hier kommt der Code für Hintergrund-Farbwechsel rein
End Sub
```

Mehr Hinweise:

Ähm – also – ein Sub() an die Textfelder zu binden ist leicht. Es dann zum Laufen zu bringen, zB während die Textfelder leer sind, ist schwer.

Aber möglich, geht so: Wenn meine Textbox leer ist (textbox1.value = ""), setze ich a=0, sonst setze ich a auf textbox1.value \* 1. Für beide Textboxen, versteht sich.

Beim Dividieren muß ich aufpassen, ob b=0 ist.

Beim Potenzieren sollte b kleiner 5 sein.

Bei Fakultät sollte a kleiner 200 sein. Der letzte lesbare Wert wird bei 170! ausgegeben.

Radizieren:

a hoch b schreibt man  $a ^ b$ .

wurzel aus a schreibt man  $a ^ 0.5$ .

dritte wurzel aus a schreibt man  $a ^ 0.3333333$ .

vierte oder 17te wurzel aus a verrate ich nicht.

**\* Ist alles nur definiert wenn  $a \geq 0$ . \***

```
Sub f()
    ' 2 werte holen und labels vollschreiben
End Sub
```

```
Private Sub TextBox1_Change()
    MsgBox ("textbox1 wurde geändert.")
    call f()
End Sub
```

```
Private Sub TextBox2_Change()
    MsgBox ("textbox2 wurde geändert.")
    call f()
End Sub
```

#### 8.4 Taschenrechner02 – Add-ons

Add-On: Schreiben Sie ihn so, daß er beim Klick aufs Formular die Formularfarbe ändert.

Add-On: Geben Sie das Ergebnis der Division aus als GanzZahl Rest Ganzzahl.

Add-On : Geben Sie auch den gerundeten Wert `round ( zahl, nks)` aus. `zahl` ist dabei die Zahl, die gerundet wird; `nks` ist die zahl der Nachkommastellen. – Testen Sie sorgfältig, u.a. mit 2,5.

Add-On : Schreiben Sie selbst die Funktion runden. Die Funktion `conversion.int(double)` könnte dabei hilfreich sein.

### **8.5 Na prima, eine Seite für Musterlösung Taschenrechner02**

... und die ist noch ganz leer. Montag nochmal gucken.

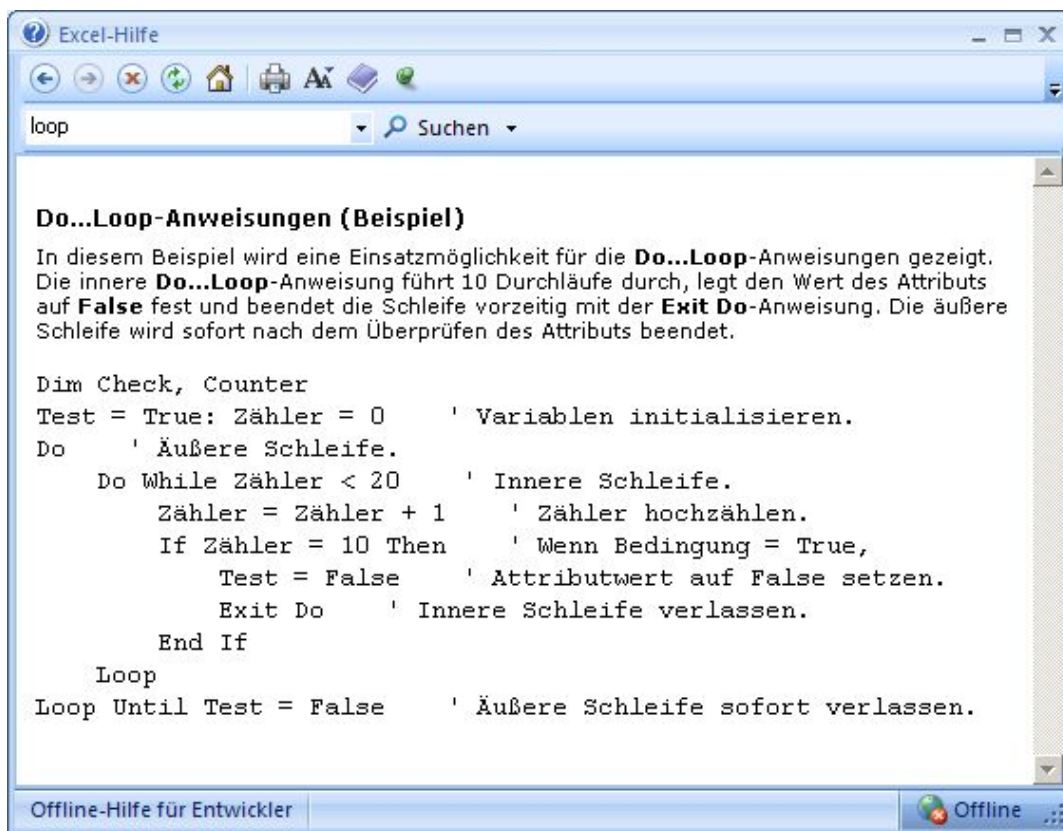
## 9 Zählschleifen mit for und while..wend [optional]

```
'zählschleife mit if drin
  For i = 1 To 20
  '   MsgBox (i)
    Range("d" & (i + 6)).Value = i * i
    If (i = 9) Then
      Range("d" & (i + 6)).Value = "i ist 9, hier mach ich nix. "
    End If
  Next i

'while..wend
  Dim v As Integer
  v = 20
  While (v > 0)
    Range("f" & (v + 6)).Value = v
    v = v - 1
  Wend

' do ... loop while bed. / until bed.
' do while bed. ... loop

MsgBox ("sag ok um ford zu fahren")
```



hilfe do..loop