

# Makro-Programmierung (nur für Leute, welche einen Text lesen UND verstehen können)

Konventionen für diesen Lehrgang:

- ◇ **Dieses Zeichen verlangt eine Maus- oder Tastatur-Aktion von Dir!**
- **Sätze mit diesem Zeichen musst Du nur verstehen, sie verlangen KEINE Maus- oder Tastatur-Aktion von Dir!**

## Basis-Wissen

- Die Programmiersprache **Visual Basic** wird zum Programmieren von Windows-Anwendungen benutzt
- Die Makrosprache **Visual Basic for Applications (VBA)** basiert auf Visual Basic und ist seit 1995 fester Bestandteil von EXCEL
- Ein **Makro** ist ein kleineres selbständig laufendes Programm, welches stets die gleichen, ihm zugewiesenen Arbeitsschritte ausführt  
Andere Namen für Makro: **Prozedur, Unterprogramm, Subroutine**
- Ein EXCEL-Makro ist in der Regel in der **Arbeitsmappe integriert** und befindet sich in einem sogenannten **MODUL**
- Für das Programmieren und Nachbearbeiten eines Makros wird ein sogenannter **Visual-Basic-EDITOR** benötigt
- Du hast bereits mit anderen Editoren gearbeitet: zB Text-Editoren wie Word oder AppleWorks
- In Excel gibt es zwei Verfahren einen Makro zu erstellen
  - 1) Du befiehst Excel, einen einfachen von Dir ausgeführten Routine-Ablauf **aufzuzeichnen**.
  - 2) Du **programmierst** mit dem VisualBasic-Editor selber ein kleines Programm.

## 1) Aufzeichnen eines Befehlsmakros

- ◇ Erstelle eine neue Excel-Arbeitsmappe und sichere diese z.B. unter dem Titel: **Meine\_Makros**
- Dein erstes Makro ist dein persönliche Adress-Kopf für einen Brief: Vorname, Name, Adresse, PLZ, Ort, Tel., Fax, Handy, E-mail
- Die Darstellung (Anordnung, Grösse und Farbe der Schrift, Zeichensatz) kannst Du selber wählen  
**Meine\_Adresse** ist der Titel (=Makroname) Deines ersten Makros (**Makrotitel dürfen keine Leerschläge enthalten!**)

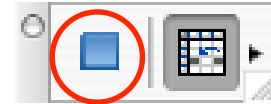
- ◇ 1) Menu **ANSICHT > Symbolleiste > VisualBasic:**

Auf dem Bildschirm erscheint die **VisualBasic-Symbol-Leiste:**



- ◇ 2) Klicke auf die rot eingekreiste Taste: **Makro aufzeichnen**

Auf dem Bildschirm erscheint eine zweite VisualBasic-Symbol-Leiste für das **Beenden** der Aufzeichnung:



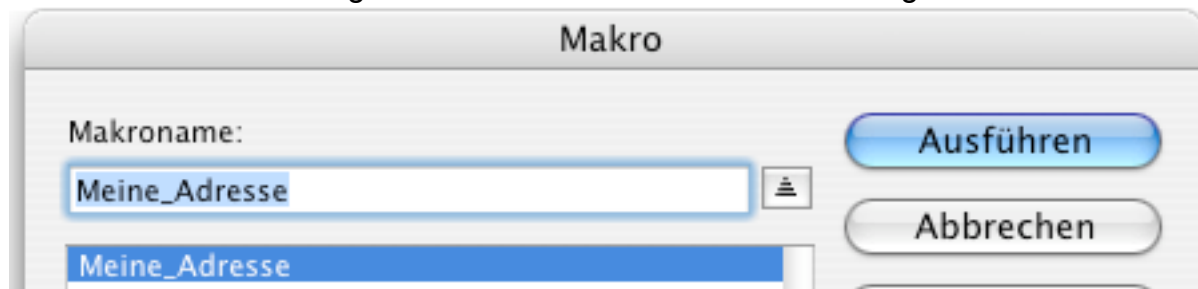
- ◇ 3) Als **Makroname** tippst Du: **Meine\_Adresse** und bestätigst mit der **ok-Taste**
- ◇ 4) Klicke in die **Zelle A1** des geöffneten Excel-Arbeitsblattes
- ◇ 5) Schreibe nun Deine Adresse nach Deinen Vorstellungen (**Excel zeichnet jeden Arbeitsschritt automatisch auf**)
- ◇ 6) Wenn Du Deine Adresse fertig formatiert hast, klickst Du auf die rot eingekreiste **Beenden-Taste**
- Excel hat automatisch ein neues **Modul 1** erstellt und Deine Aktionen in die VisualBasic-Sprache übersetzt und dort abgelegt

### **Abspielen des Befehlsmakros**

- ◇ Damit Du siehst, dass Excel tatsächlich Deine Aktivitäten in einem Makro gespeichert hat, löschst Du auf dem Arbeitsblatt Deine Adress- Eingaben.  
Starte nun das Makroprogramm wie folgt:



- ◇ Klicke in der VisualBasic-Symbolleiste auf die rot eingekreiste Taste: **Makro ausführen**
- Im erscheinenden Dialogfeld ist Dein Adressmakro bereits ausgewählt!



- ◇ Klicke auf die Taste: **Ausführen!**  
Das Makro reproduziert nun Deine Eingaben!

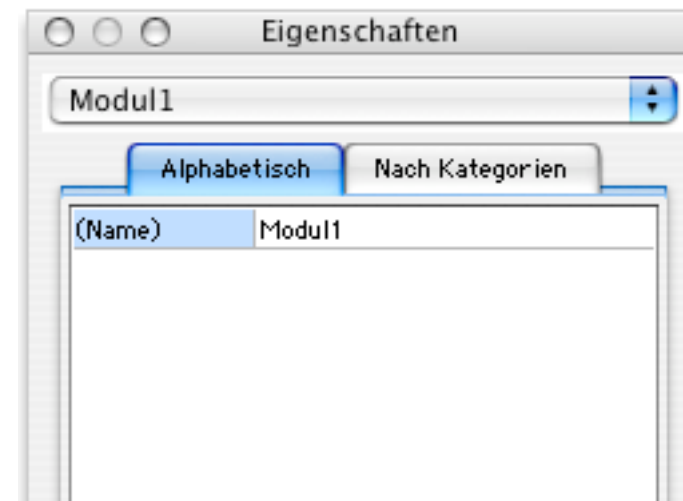
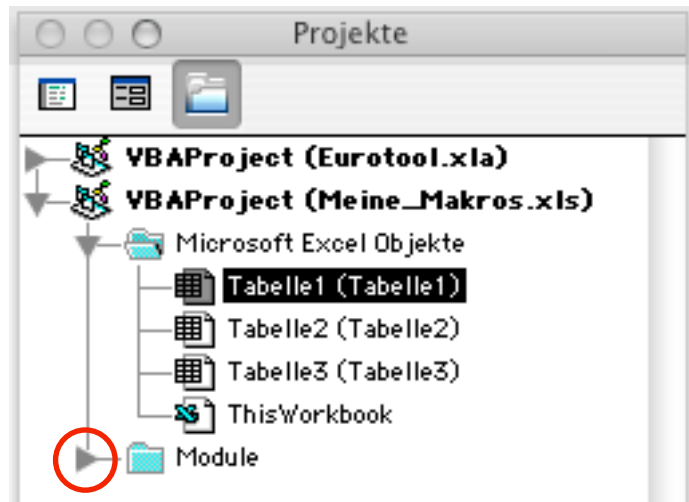
## Lesen und analysieren des Befehlsmakros

- Den Code Deines Makros kannst Du wie folgt, anschauen, analysieren und verändern:

- ◇ Klicke in der VisualBasic-Symbolleiste auf die rot eingekreiste Taste: VisualBasic-Editor



- Auf dem Bildschirm erscheinen zwei Fenster: **Projekte** und **Eigenschaften**



- ◇ Klicke im Projekte-Fenster auf das rot eingekreiste Dreieck links neben dem Ordner: **Module**



- Der Inhalt des Module-Ordners wird nun in einer Liste dargestellt:

- Im **Modul 1** befindet sich der Programmcode Deines Makros!

- ◇ **Öffne** das **Modul 1** mit einem Maus-Doppelklick

- Der Inhalt des Modul 1 wird nun in einem Fenster dargestellt

- Zwischen der Anfangszeile: **Sub Meine\_Adresse()** und der Schlusszeile **End Sub** liegt der ganze Code Deines Makros:

```
Option Explicit
Sub Meine_Adresse()
' Meine_Adresse Makro
' Makro am 6.10.2004 von Feri Kolb aufgezeichnet
  ActiveCell.FormulaR1C1 = "Feri Kolb"
  Range("A2").Select
  ActiveCell.FormulaR1C1 = "HB 38"
  Range("A3").Select
  ActiveCell.FormulaR1C1 = "8405 Winterthur"
  Range("A4").Select
  ActiveCell.FormulaR1C1 = "Tel+Fax: ++41 52 232 47 44"
  Range("A5").Select
  ActiveCell.FormulaR1C1 = "Email: ferikolb@gmx.ch"
  Range("A1:A5").Select
  With Selection.Font
    .Name = "Times"
    .FontStyle = "Fett"
    .Size = 12
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = 11
  End With
  Columns("A:A").EntireColumn.AutoFit
End Sub
```

Titel der Sub-Routine (=Makroname)

Kommentarzeilen sind grün. Sie beginnen mit einem Hochkomma und sind lediglich Erläuterungen zum Programm-Code

In die aktive Zelle wird **HB 38** geschrieben

Die Zelle A4 wird ausgewählt / aktiviert

Die Zellen A1 bis A5 werden ausgewählt / aktiviert

In den ausgewählten Zellen (=Zellbereich) werden die Schrift-Attribute zugeteilt

Die Breite der Spalte A wird der Textlänge angepasst

## 2) Programmieren eines Makros

### 2.1 Programmieren und Aufrufen einer einfachen Dialogbox

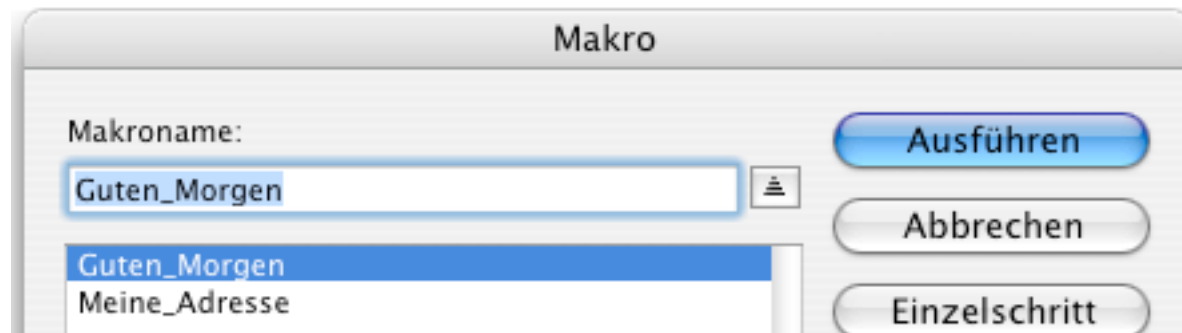
- ◇ Schreibe unter Dein erstes Makro (Sub Meine\_Adresse()) den folgenden Programm-Text (oder kopiere ihn hier und setze ihn dort ein):

```
Sub Guten_Morgen()  
MsgBox "Guten Morgen"  
End Sub
```

- **MsgBox** (= englisch: Message-Box) ist ein einfaches Dialogfeld für Kommentare und Hinweise. **Texte müssen in Anführungs- und Schlusszeichen stehen!**

Starte nun das Makroprogramm wie folgt:

- ◇ Klicke in der VisualBasic-Symbolleiste auf die rot eingekreiste Taste: **Makro ausführen**
- ◇ Wähle mit der Maus den Makronamen: **Guten\_Morgen** und klicke dann auf die Taste: **Ausführen**



## 2.2 Programmieren und Aufrufen einer Dialogbox mit einem Eingabefeld

- ◇ Schreibe unter Dein zweites Makro (Sub Guten\_Morgen()) den folgenden Programm-Text (oder kopiere ihn hier und setze ihn dort ein):

```
Sub WieGehtEsDir()  
Dim Aufforderung1 As String      'eine Variable mit dem Namen Aufforderung1 wird als Text-Variable deklariert  
Dim Aufforderung2 As String      'eine Variable mit dem Namen Aufforderung2 wird als Text-Variable deklariert  
Dim Name As String               'eine Variable mit dem Namen Name wird als Text-Variable deklariert  
Dim Befinden As String           'eine Variable mit dem Namen Befinden wird als Text-Variable deklariert  
  
Aufforderung1 = "Wie heisst Du?"  'der Variablen mit dem Namen Aufforderung1 wird ein Text zugeordnet  
Name = InputBox(Aufforderung1)    'die Antwort aus dem Dialogfeld wird der Variablen Name zugeordnet  
Aufforderung2 = "Wie geht es Dir heute?" 'der Variablen mit dem Namen Aufforderung2 wird ein Text zugeordnet  
Befinden = InputBox(Aufforderung2) 'die Antwort aus dem Dialogfeld wird der Variablen Befinden zugeordnet  
MsgBox Name & "! Das ist aber ein Zufall, mir geht es heute auch " & Befinden & "!" 'die beiden Antworten der Variablen:  
'Name und Befinden werden zusammen mit einem starren Text verkettet und in der MessageBox angezeigt!  
End Sub
```

- Der Inhalt des **Eingabefeldes** (= das, was der Anwender dort eintippt) muss in einer **Variablen** zwischengespeichert werden
- Für **Texte** wird eine sogenannte **String** (-Variable) deklariert
- Für **ganze Zahlen** wird eine sogenannte **integer** (-Variable) deklariert (wird in unserem Beispiel nicht benötigt!)
- **Dim** (= englisch: Dimension) teilt dem Programm mit, dass eine **Variable** deklariert (=festgelegt) und entsprechender Speicherplatz geschaffen wird. Auf das Wort **Dim** folgt ein beliebiger, aussagekräftiger **Name** für die gewünschte **Variable**.  
In obigem Beispiel sind dies: Aufforderung1, Aufforderung2, Name, Befinden
- **InputBox** ist ein Dialogfeld für die Eingabe von Text oder Zahlen
- Der **Klammer-Inhalt** von **InputBox** wird im Dialogfeld der InputBox als Text über dem Eingabefeld angezeigt (zB: Wie geht es Dir?)
- In obigem Beispiel wird der Anwender zweimal aufgefordert, etwas in ein Eingabefeld zu schreiben:  
Beim ersten Aufruf der InputBox wird der Inhalt des **Eingabefeldes** (= das, was der Anwender dort eintippt hat) in der Variablen **Name** zwischengespeichert  
Beim zweiten Aufruf der InputBox wird der Inhalt des **Eingabefeldes** (= das, was der Anwender dort eintippt hat) in der Variablen **Befinden** zwischengespeichert
- Nach dem zweiten Aufruf der InputBox wird eine **MsgBox** mit einem Text, bestehend aus den beiden Antworten des Anwenders eingeblendet!
- Das Zeichen **&** ist ein sogenannter **Text-Operator**, er verbindet die **Variablen** mit dem **starken Text**, welcher in Anführungs- und Schlusszeichen steht

Starte nun das Makroprogramm wie folgt:



- ◇ Klicke in der VisualBasic-Symbolleiste auf die rot eingekreiste Taste: **Makro ausführen**
- ◇ Wähle mit der Maus den Makronamen: **WieGehtEsDir** und klicke dann auf die Taste: **Ausführen**

## 2.3 Programmieren eines einfachen Zahlen-Ratespiels

- **Das Programm beinhaltet:**
  - Eingabe- und Ausgabefenster (**InputBox** und **MsgBox**)
  - Ganzzahl-Funktion: **Int(Zahl)**
  - Zufallszahl-Funktion: **rnd()**
  - Schleife mit Prüfung am Ende: **Do Until Bedingung Loop**
- ◇ Schreibe unter Dein drittes Makro (WieGehtEsDir()) den folgenden Programm-Text (oder kopiere ihn hier und setze ihn dort ein):

### **Sub ZahlenRaten1()**

'---Eingabefeld, Es muss zwingend eine ZUFALLSZAHL 1-5 eingegeben werden---

```
Dim Zufall As Integer
Dim Spielerzahl As Integer
Zufall = Int((Rnd * 5) + 1)
MsgBox "Tippe so lange Zahlen zwischen 1 und 5 ein, bis Du meine Zufallszahl erraten hast!"
Do Until Spielerzahl = Zufall
    Spielerzahl = Application.InputBox(prompt:="Zahl zwischen 1 und 5 eintippen", Type:=1)
Loop
MsgBox "Deine Zahl: " & Spielerzahl & " ist richtig! Meine Zufallszahl war: " & Zufall
End Sub
```

- Die in VisualBasic für Excel - eingebaute Funktion: **Int(Dezimalzahl)** liefert eine **Ganzzahl**.  
Beispiel: Int(10.75) liefert 10, **der Dezimalteil wird abgeschnitten!**
- Die in VisualBasic für Excel - eingebaute Funktion: **Rnd** liefert eine **Zufallszahl (Dezimalzahl) zwischen Null und Eins** :  $0 \leq \text{Zufallszahl} < 1$   
In obigem Beispiel ergibt **(Rnd \* 5)** Null oder eine **Dezimalzahl**  $< 5$   
**Int(Rnd \* 5)** ergibt Null oder eine **Ganzzahl**  $< 5$

$\text{Int}((\text{Rnd} * 5) + 1)$  ergibt eine Ganzzahl zwischen 1 und 5

### 2.3.1 Verbessern des obigen Zahlen-Ratespiels

- **Das Programm beinhaltet zusätzlich:**

- eine Zählvariable für das Zählen der Rate-Versuche des Anwenders
- **If-Then**-Funktion
- **Chr(Zeichennummer)**-Funktion

◇ Schreibe unter Dein drittes Makro (WieGehtEsDir()) den folgenden Programm-Text (oder kopiere ihn hier und setze ihn dort ein):

#### **Sub ZahlenRaten2()**

```
' ---Eingabefeld, Es muss zwingend eine ZUFALLSZAHL 1-5 eingegeben werden---  
' verbesserte Version von Sub ZahlenRaten1() mit Kommentar und Statistik
```

```
Dim Zufall As Integer  
Dim Spielerzahl As Integer  
Dim AnzVersuche As Integer  
Zufall = Int((Rnd * 5) + 1)  
AnzVersuche = 1
```

```
Do Until Spielerzahl = Zufall  
    Spielerzahl = Application.InputBox(prompt:="Zahl zwischen 1 und 5 eintippen", Type:=1)  
    If Spielerzahl <> Zufall Then  
        MsgBox "Deine Zahl war leider falsch"  
        AnzVersuche = AnzVersuche + 1  
    End If  
Loop  
MsgBox "Anzahl Deiner Rate-Versuche: " & AnzVersuche & Chr(13) & Chr(13) & _  
"Meine Zufallszahl war: " & Zufall
```

#### **End Sub**

- Die **If-Then**-Funktion ist eine der häufigsten Funktionen in Programmen. Im If-Teil wird eine **Bedingung** gestellt. Wenn diese **erfüllt** (=stimmt, richtig ist) wird, wird der Then-Teil ausgeführt; sonst geschieht **nichts**!
- ◇ Starte nun das Makroprogramm

*viel Spass*