

# Excel – Makros und VBA

## Bücher

- Handbuch des RRZN: Excel 2007 – Automatisierung und Programmierung
- Thomas Theis: Einstieg in VBA mit Excel; Galileo Press
- Can & Schwimmer & Weber: Excel – Programmierung; Microsoft Press
- Breden & Schwimmer: Das Excel-VBA-Codebook; Addison-Wesley

## Informationen im Web

- Makros
  - <http://ms-excel.eu>
  - <http://www.excel-inside.de/>
- V(isual) B(asic) A(pplication)
  - <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/b012.pdf>
  - [http://de.wikibooks.org/wiki/VBA\\_in\\_Excel\\_-\\_Grundlagen](http://de.wikibooks.org/wiki/VBA_in_Excel_-_Grundlagen)
  - <http://www.excel-training.de>
  - <http://www.herber.de/>

## Excel

- Tabellenkalkulationsprogramm.
- ... wird genutzt, um numerische Daten in Tabellenform zu erfassen.
- ... kann Daten automatisch berechnen.
- ... bietet viele vordefinierte Formeln für die Berechnung.

## Makros

- ... automatisieren Aktionen, die häufig wiederholt werden.
- ... führen Routineaufgaben auf Knopfdruck durch.
- ... fassen Arbeitsschritte zusammen.
- ... zeichnen Befehlsfolgen und Arbeitsschritte auf.
- Intern werden die aufgezeichneten Aktionen in die Programmiersprache VBA (Visual Basic for Application) konvertiert.

## Eigenschaften

- Das Makro selber kann beliebig oft aufgerufen werden.
- Ein Makro hat immer die gleichen Startbedingungen. Beispiel: Eine Zelle wird formatiert. Vor dem Start des Makros muss immer eine Zelle markiert sein. Andernfalls wird ein Fehler angezeigt.
- Die Arbeitsschritte werden nach und nach abgearbeitet.
- Eine Wiederholung von einzelnen Arbeitsschritten innerhalb eines Makros ist nicht möglich.
- Arbeitsschritte können nicht in Abhängigkeit eines bestimmten Wertes ausgeführt werden.

## Drehbuch: Zwei Zahlen addieren

In den Zellen A1 : A2 sowie B1 : B2 befinden sich Zahlen.

Makrorecorder starten.

Wechsel in die Zelle C1.

Formular schreiben: = A1 + A2

Makroaufzeichnung beenden.

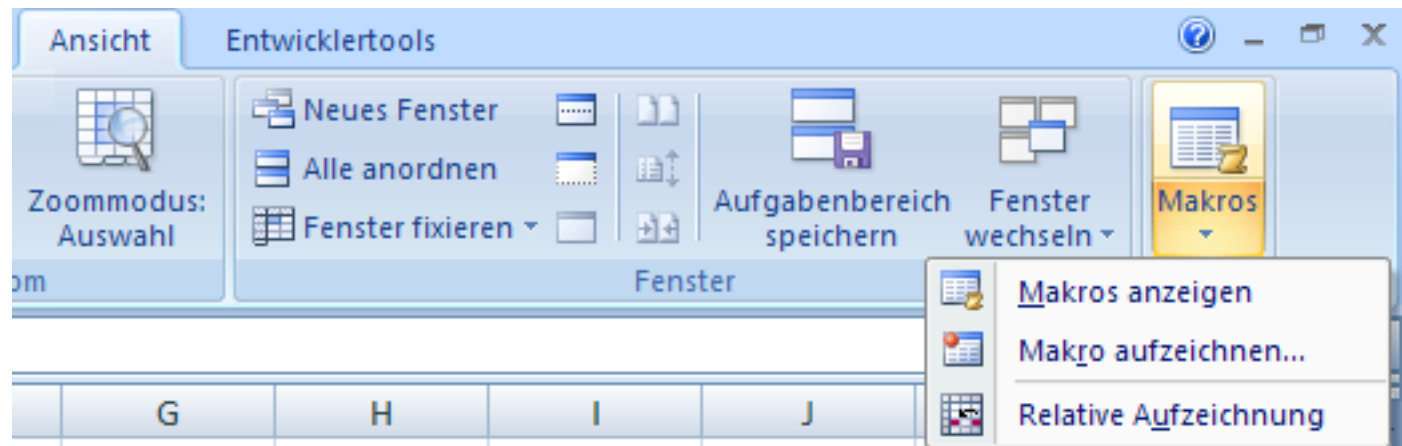
## Arbeitsschritte aufzeichnen

- Führen Sie die Arbeitsschritte nach und nach anhand eines Drehbuchs durch.
- Nicht korrekte Arbeitsschritte können nicht gelöscht werden!



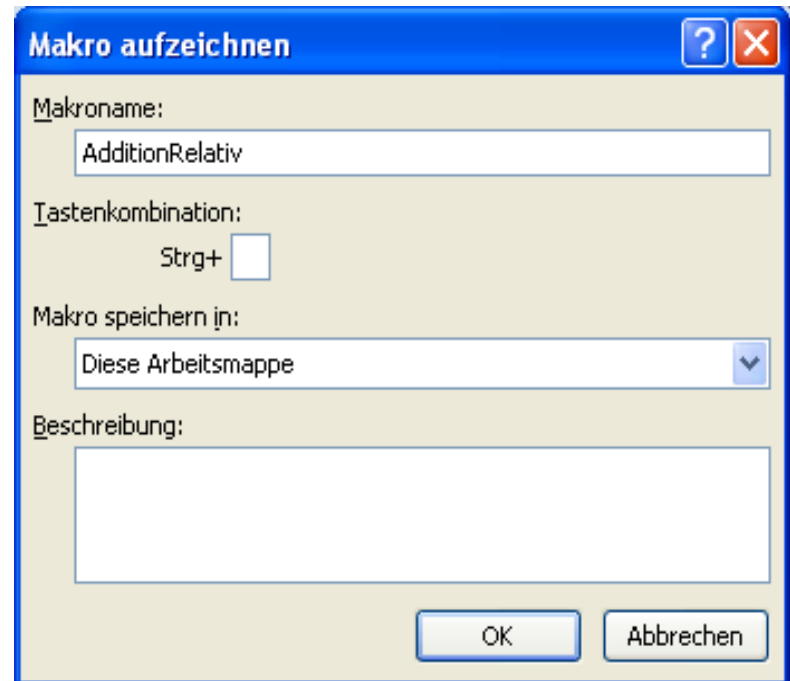
## Makroaufzeichnung starten

- Wählen Sie das Ribbon Ansicht.
- Klicken Sie auf den Pfeil nach unten der Schaltfläche *Makros*.
- Wählen Sie den Befehl *Makros aufzeichnen* aus.
- Es öffnet sich das Dialogfeld Makros aufzeichnen.



## Dialogfeld "Makro aufzeichnen"

- Geben Sie einen Namen für das Makro ein. Der Name ist eindeutig und spiegelt die Aufgabe des Makros wieder.
- Das Makro kann mit Hilfe eines Tastaturkürzels gestartet werden.
- Legen Sie den Speicherort des Makros fest.
- Geben Sie eine Beschreibung für das Makro ein.
- *OK* startet die Aufzeichnung.



**Makro aufzeichnen**

Makroname:  
AdditionRelativ

Tastenkombination:  
Strg+

Makro speichern in:  
Diese Arbeitsmappe

Beschreibung:

OK Abbrechen

## Der Makroname

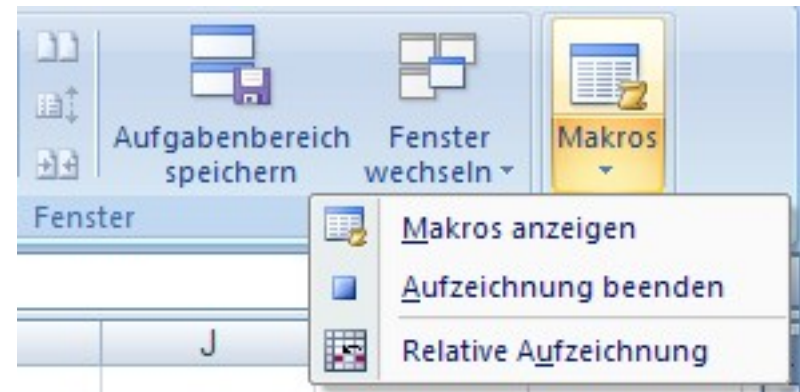
- ... muss mit einem Buchstaben beginnen.
- ... darf nur die Groß- und Kleinbuchstaben des Alphabets, Zahlen und den Unterstrich enthalten.
- Leerzeichen, Umlaute, Satzzeichen etc. sind nicht erlaubt.

## Mögliche Speicherorte

- Diese Arbeitsmappe
  - Das Makro wird in der aktuellen Arbeitsmappe hinterlegt.
  - Das Makro steht nur in dieser Arbeitsmappe zur Verfügung
- Neue Arbeitsmappe
  - Das Makro wird in einer neuen leeren Arbeitsmappe gespeichert, die als Vorlage für andere Mappen dienen kann.
  - Das Makro wird aber in der aktuellen Arbeitsmappe aufgezeichnet.
- Persönliche Makro-Arbeitsmappe
  - ... wird immer zusammen mit Excel geöffnet.
  - Das Makro kann universell in verschiedenen Arbeitsmappen eingesetzt werden.
  - ... ist standardmäßig ausgeblendet.

## Makroaufzeichnung beenden

- Wählen Sie das Ribbon Ansicht.
- Klicken Sie auf den Pfeil nach unten der Schaltfläche *Makros*.
- Wählen Sie den Befehl *Makros beenden* aus.



## Drehbuch: Zelle kopieren

Zwei Zahlen befinden sich in Zellen A1 und A2.

Makrorecorder starten.

Die Zelle A1 wird markiert.

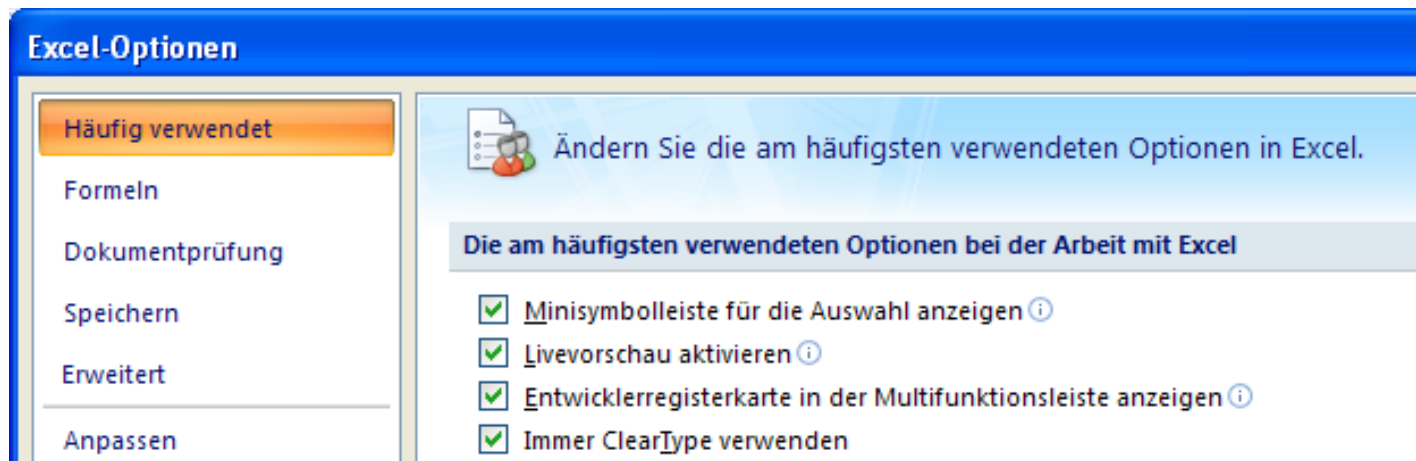
Bearbeiten - Kopieren.

Wechsel in die Zelle B1.

Bearbeiten - Einfügen.

Makroaufzeichnung beenden.

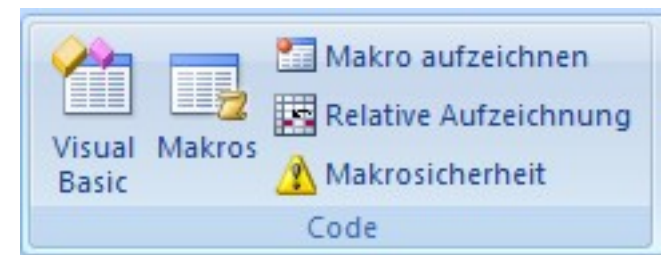
## Entwickler-Tools einblenden



- Klicken Sie auf die Office-Schaltfläche (runder Button am oberen rechten Rand).
- Klicken Sie auf die Schaltfläche *Excel-Optionen* am unteren rechten Rand des Menüs.
- Die Option *Entwicklerregisterkarte in der Multifunktionsleiste anzeigen* in der Kategorie *Häufig verwendet* wird mit einem Klick aktiviert.

## Makroaufzeichnung starten

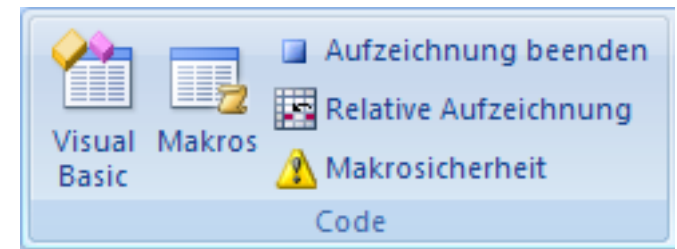
- Wählen Sie das Ribbon Entwicklertools.
- Am linken Rand befinden sich in der Kategorie Code alle Befehle zur Codegenerierung.
- Wählen Sie den Befehl *Makro aufzeichnen* aus.
- Es öffnet sich das Dialogfeld Makros aufzeichnen. Geben Sie einen Makronamen ein und wählen den Speicherort aus. Starten Sie die Aufzeichnung mit Hilfe der Schaltfläche *OK*.





## Aufzeichnung beenden

- Das Ribbon Entwicklungstools wird angezeigt.
- Der Befehl *Aufzeichnung beenden* in der Kategorie Code beendet die Aufzeichnung.



## Absolute Aufzeichnung

- Standard-Aufzeichnung.
- Die Aufzeichnung erfolgt unabhängig von der aktiven Zelle.
- Es werden immer die gleichen Zellen markiert, egal welche Zellen momentan markiert ist.
- Beispiel:
  - Während der Aufzeichnung wird die Zelle A1 markiert.
  - Egal welche Zelle nach dem Start des Makros markiert ist, die Zelle A1 wird automatisch markiert.
- Hinweis: Zellbezüge in Formeln sind standardmäßig immer relativ, egal welche Aufzeichnung man wählt.

## Absolute Adressierung im Code

```
Sub MakroCopyAbsolut()
```

```
    Range("D2").Select  
    Selection.Copy  
    Range("D3").Select  
    ActiveSheet.Paste
```

```
End Sub
```

Mit Hilfe von Range("SpalteZeile") wird eine bestimmte Zelle ausgewählt. Egal wo sich momentan die Einfügemarke befindet, wird diese Zelle kopiert.

## Relative Aufzeichnung

- Es werden die Zellen immer relativ zur aktuellen Zelle markiert.
- Beispiel:
  - Vor der Aufzeichnung ist die Zelle A1 markiert.
  - Während der Aufzeichnung wird die Zelle B2 markiert.
  - Durch das Makro wird immer die Zelle rechts von der aktiven Zelle markiert.
- Die relative Aufzeichnung kann während der Aufzeichnung gestartet oder beendet werden.

## Relative Adressierung im Code

```
Sub MakroCopyRealtiv()
```

```
    ActiveCell.Offset(-1, 0).Range("A1").Select  
    Selection.Copy
```

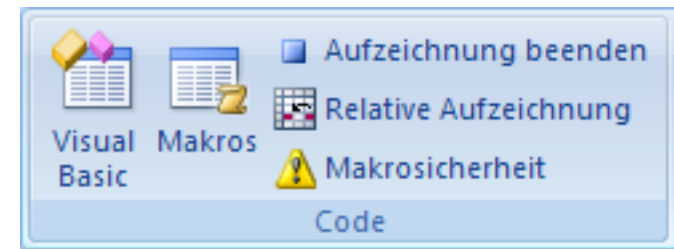
```
    ActiveCell.Offset(2, 0).Range("A1").Select  
    ActiveSheet.Paste
```

```
End Sub
```

Mit Hilfe von ActiveCell wird die momentan ausgewählte Zelle genutzt. Von der aktiven Zelle wird eine Zeile in der selben Spalte nach oben gegangen (Offset(-1, 0))

## Relative Aufzeichnung beenden oder starten

- Das Ribbon Entwicklungstools wird angezeigt.
- Klicken Sie auf den Befehl *Relative Aufzeichnung* in der Kategorie Code.
- Der Befehl ist ein Kippschalter. Wenn der Befehl aktiv ist, wird dieser durch ein Mausklick deaktiviert und umgekehrt.



## Makros in der aktuellen Arbeitsmappe speichern

- Excel 2007: Standardmäßig enthalten Arbeitsmappen keine Makros.
- Klicken Sie auf die Office-Schaltfläche (runder Button am oberen rechten Rand).
- Klicken Sie auf die Schaltfläche *Speichern unter – Excel-Arbeitsmappe mit Makros*. Die vorhandenen Makros werden automatisch mit der Arbeitsmappe gespeichert.

## Makros in einer Excel-Vorlage speichern

- Klicken Sie auf die Office-Schaltfläche (runder Button am oberen rechten Rand).
- Klicken Sie mit der linken Maustaste auf die Schaltfläche *Speichern unter*.
- Im Dialogfeld *Speichern unter* wählen Sie als Dateityp *Excel-Vorlage mit Makros (\*.xltn)* aus.
- Geben Sie in das Textfeld *Dateiname* eine selbsterklärende Bezeichnung für die Vorlage ein.
- Klicken Sie auf *Speichern*. Die Vorlage wird standardmäßig im Verzeichnis *Dokumente und Einstellungen \ [benutzer] \ Anwendungsdaten \ Microsoft \ Templates* abgelegt.



## ... in der persönliche Makroarbeitsmappe speichern

- Schließen Sie Excel.
- Mit Hilfe eines Dialogfeld wird nachgefragt, ob Änderungen an der persönlichen Makroarbeitsmappe gespeichert werden sollen. Durch Klick auf die Schaltfläche *Ja* werden Änderungen gespeichert.

## Die persönliche Makroarbeitsmappe

- ... ist standardmäßig ausgeblendet. Mit Hilfe des Ribbons *Ansicht* und dem Befehl *Fenster einblenden* in der Kategorie *Fenster* kann die Datei eingeblendet werden.
- Makros in der ausgeblendeten Arbeitsmappe können ausgeführt, aber nicht bearbeitet oder gelöscht werden.
- ... wird als binäre Datei unter dem Namen `personal.xlsb` im Verzeichnis `Dokumente und Einstellungen \ [benutzer] \ Anwendungsdaten \ Microsoft \ Excel \ XLSTART` gespeichert.

## Makros anzeigen

- Das Ribbon Entwicklungstools wird angezeigt.
- Klicken Sie auf den Befehl *Makros* in der Kategorie Code.
- In der Liste werden alle Makros der aktuell aktiven Arbeitsmappe angezeigt.
- Mit einem Mausklick wird ein Makro ausgewählt
- Das gewählte Makro wird farbig hinterlegt und in dem Textfeld darüber angezeigt.



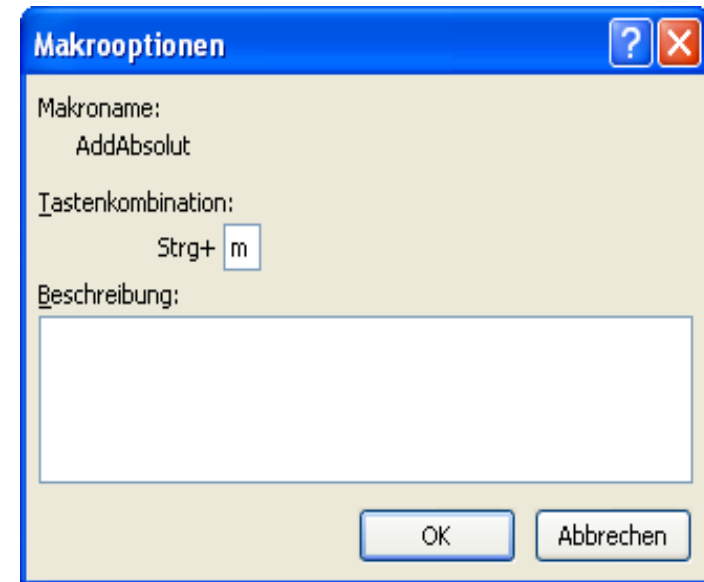
## Makro starten

- Das Ribbon Entwicklungstools wird angezeigt.
- Klicken Sie auf den Befehl *Makros* in der Kategorie Code.
- Wählen Sie ein Makro aus.
- Die Schaltfläche *Ausführen* startet das gewählte Makro.



## Tastaturkürzel nutzen

- Das Ribbon Entwicklungstools wird angezeigt.
- Klicken Sie auf den Befehl *Makros* in der Kategorie Code.
- Wählen Sie ein Makro aus.
- *Optionen* zeigt alle Einstellungen zum aktiven Makro an.
- Geben Sie ein Tastaturzeichen in das Textfeld Str+ ein.
- Bestätigen Sie die Angaben mit *OK*.



## Hinweise

- In der Liste Excel "Tastaturkombinationen und Funktionstasten" werden alle vorbelegten Kombinationen angezeigt. Wählen eine Tastenkombination, die dort nicht vorhanden ist.
- Für Tastenkombinationen in der Form `<STRG>+<UMSCHALT>+<m>` halten Sie die Umschalttaste während der Eingabe des Tastaturzeichens gedrückt.

## Start über die Schnellzugriffsleiste

- Klicken Sie mit der rechten Maustaste auf die Schnellzugriffsleiste. Wählen Sie den Befehl Symbolleiste für den Schnellzugriff anpassen im Kontextmenü aus.
- Die Excel-Optionen werden geöffnet. Die Kategorie Anpassen ist eingeblendet.
- Markieren Sie im Kombinationslistenfeld Befehle auswählen Makros. In der Liste darunter werden alle zur Verfügung stehenden Makros angezeigt. Wählen Sie ein Makro aus.
- Wählen Sie im Kombinationsfeld Symbolleiste für den Schnellzugriff anpassen die gewünschte Arbeitsmappe aus.
- Klicken Sie auf die Schaltfläche *Hinzufügen*. Das Makro wird mit dem entsprechenden Symbol in der Schnellzugriffsleiste angezeigt.

## Mit einer Schaltfläche starten

- In der Arbeitsmappe oder Vorlage wird eine Schaltfläche eingefügt.
- Die Schaltfläche wird mit einem Makro verbunden.
- Sobald der Nutzer auf die Schaltfläche klickt, wird das Makro gestartet.



## Schaltflächen einfügen

- Das Ribbon Entwicklungstools wird angezeigt.
- Klicken Sie auf den Befehl *Einfügen* in der Kategorie Steuerelemente.
- Wählen Sie das Steuerelement Schaltfläche aus. Der Mauszeiger verwandelt sich ein Kreuz.
- Ziehen Sie mit der gedrückt gehaltenen linken Maustaste ein Auswahlrahmen auf. Sobald Sie die Maustaste loslassen, wird das Steuerelement in der gewünschten Größe angezeigt.

## ... und mit Makro verbunden

- Legen Sie eine Schaltfläche auf dem Arbeitsblatt ab.
- Sobald Sie die gedrückt gehaltene Maustaste loslassen, wird automatisch das Dialogfeld *Makro zuweisen* geöffnet.
- Wählen Sie in diesem Dialogfeld ein Makro aus. Das Dialogfeld wird mit *OK* geschlossen.
- Verbindung nachträglich ändern:
  - Klicken Sie mit der rechten Maustaste auf die Schaltfläche.
  - Wählen Sie den Befehl *Makro zuweisen* im Kontextmenü aus.
  - Es öffnet sich das dazugehörige Dialogfeld.

## Text einer Schaltfläche

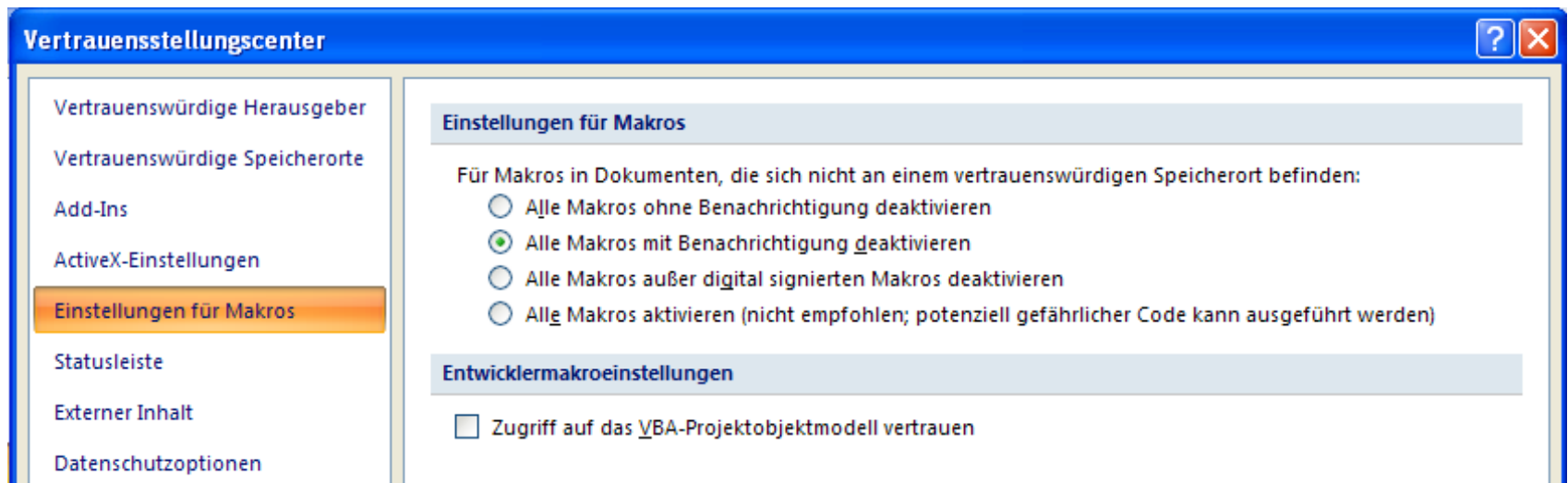
- Markieren Sie den Text auf der Schaltfläche mit Hilfe der Maus.
- Ersetzen Sie den Text durch ein Verb, welches die Aktion beschreibt.
- Andere Möglichkeit:
  - Klicken Sie mit der rechten Maustaste auf die Schaltfläche.
  - Wählen Sie den Befehl *Text bearbeiten* im Kontextmenü aus.
  - Es öffnet sich das dazugehörige Dialogfeld.

## Makros löschen

- Das Ribbon Entwicklungstools wird angezeigt.
- Klicken Sie auf den Befehl *Makros* in der Kategorie Code.
- Wählen Sie ein Makro aus.
- Klicken Sie auf die Schaltfläche *Löschen*.
- Hinweise zum Löschen:
  - Es werden Symbole in der Schnellzugriffsleiste oder Schaltflächen nicht automatisch mit dem Makro gelöscht. Falls ein Makro bei einem Klick auf eine Schaltfläche oder einem Icon in der Schnellzugriffsleiste fehlt, wird eine Warnmeldung ausgegeben.
  - Eine Löschung kann nicht rückgängig gemacht werden.

## Sicherheit

- Das Ribbon Entwicklungstools wird angezeigt.
- Klicken Sie auf den Befehl *Makrosicherheit* in der Kategorie Code.
- Es wird das Vertrauensstellungscenter geöffnet. Die Kategorie Einstellungen für Makros ist eingeblendet.



## Möglichkeiten

- Alle Makros ohne Benachrichtigung deaktivieren  
Alle Makros mit Benachrichtigung deaktivieren
  - Es werden alle Makros mit einem Hinweis deaktiviert.
  - Ausnahme: Makros an vertrauenswürdigen Speicherorten können ausgeführt werden.
- Alle Makros außer digital signierten deaktivieren
  - Makros von vertrauenswürdigen Herausgebern oder Speicherorten können ausgeführt werden.
  - Alle anderen Makros werden deaktiviert.
- Alle Makros aktivieren
  - Alle Makros werden ohne Hinweis aktiviert.
  - Vorsicht: Viren können eingeschleust werden.

## Vertrauenswürdige Speicherorte

- Das Ribbon Entwicklungstools wird angezeigt.
- Klicken Sie auf den Befehl *Makrosicherheit* in der Kategorie Code.
- Es wird das Vertrauensstellungscenter geöffnet. Wechseln Sie in die Kategorie Vertrauenswürdige Speicherorte.
- Vorhandene Speicherorte können gelöscht oder deren Pfad geändert werden.

## Vertrauenswürdige Speicherorte hinzufügen

- Klicken Sie auf die Schaltfläche *Neuen Speicherort hinzufügen*.
- Mit Hilfe der Schaltfläche *Durchsuchen* wird der gewünschte Pfad ausgewählt. Alle dort vorhandenen Makros sind vertrauenswürdig.
- Mit Hilfe des Kontrollkästchens werden Unterordner des angegebenen Speicherortes automatisch als vertrauenswürdig erklärt.





## Makros bearbeiten

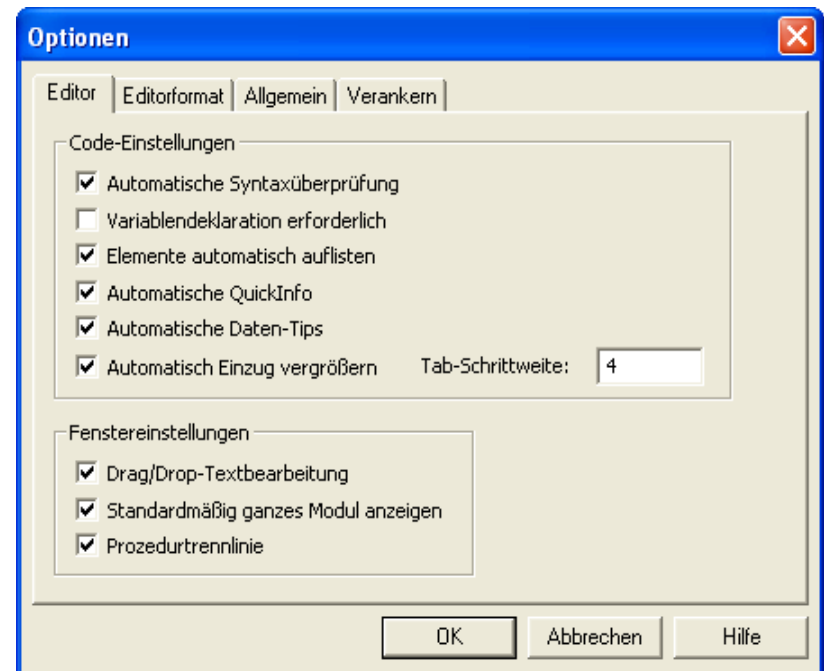
- Das Ribbon Entwicklungstools wird angezeigt.
- Klicken Sie auf den Befehl *Makros* in der Kategorie Code.
- Wählen Sie ein Makro aus.
- Klicken Sie auf die Schaltfläche *Bearbeiten*.
- Es wird automatisch der VBA-Editor geöffnet.

## VBA-Editor

- ... ist die Entwicklungsumgebung für VBA-Programme.
- ... ist ein eigenständiges Programm, welches mit jeder Office-Anwendung geliefert wird.
- ... kann mit <ALT>+<F11> innerhalb einer Office-Anwendung geöffnet werden.
- ... schließen:
  - *Datei - Schließen und zurück zu Microsoft Excel* in der Menüleiste des VBA-Editor.
  - Schließen-Schaltfläche in der Titelleiste des Editors.

## Einstellungen für den VBA-Editor

- Öffnen Sie das Menü *Extras – Optionen*.
- Mit Hilfe von Registerkarten werden die verschiedenen Optionen zusammengefasst.
- *OK* speichert die eingestellten Optionen.



## Registerkarten

- Die Registerkarte Editor ist eine Sammlung von Einstellungen, die den Code sowie das Codefenster selber betreffen.
- Die Registerkarte Editorformat formatiert die verschiedenen Code-Anweisungen.
- Die Registerkarte Allgemein bietet Einstellungen
  - ... für das Raster in UserForms.
  - ... für die Behandlung von Fehlern.
  - ... für die Art der Kompilierung von Code.
- Die Registerkarte Verankerung verankert die Fenster innerhalb des Editors.

## Einstellungen auf der Registerkarte Editor

- Aktivieren Sie das Kontrollkästchen Automatische Syntaxüberprüfung. Sobald Sie mit Hilfe von <Return> in eine neue Zeile wechseln, werden Syntaxfehler farbig dargestellt. Eine Hilfe zur Behebung des Fehlers wird eingeblendet.
- Variablendeklaration erforderlich setzt am Anfang eines VBA-Moduls die Anweisung `Option Explicit` automatisch. Variablen in VBA müssen deklariert werden. Es werden unnötige Fehler vermieden.
- Elemente automatisch auflisten blenden eine Ergänzungsmöglichkeiten innerhalb von Objekthierarchien ein.
- Automatische Quickinfos zeigen gelbe Erklärfenster zu Funktionen etc. an.

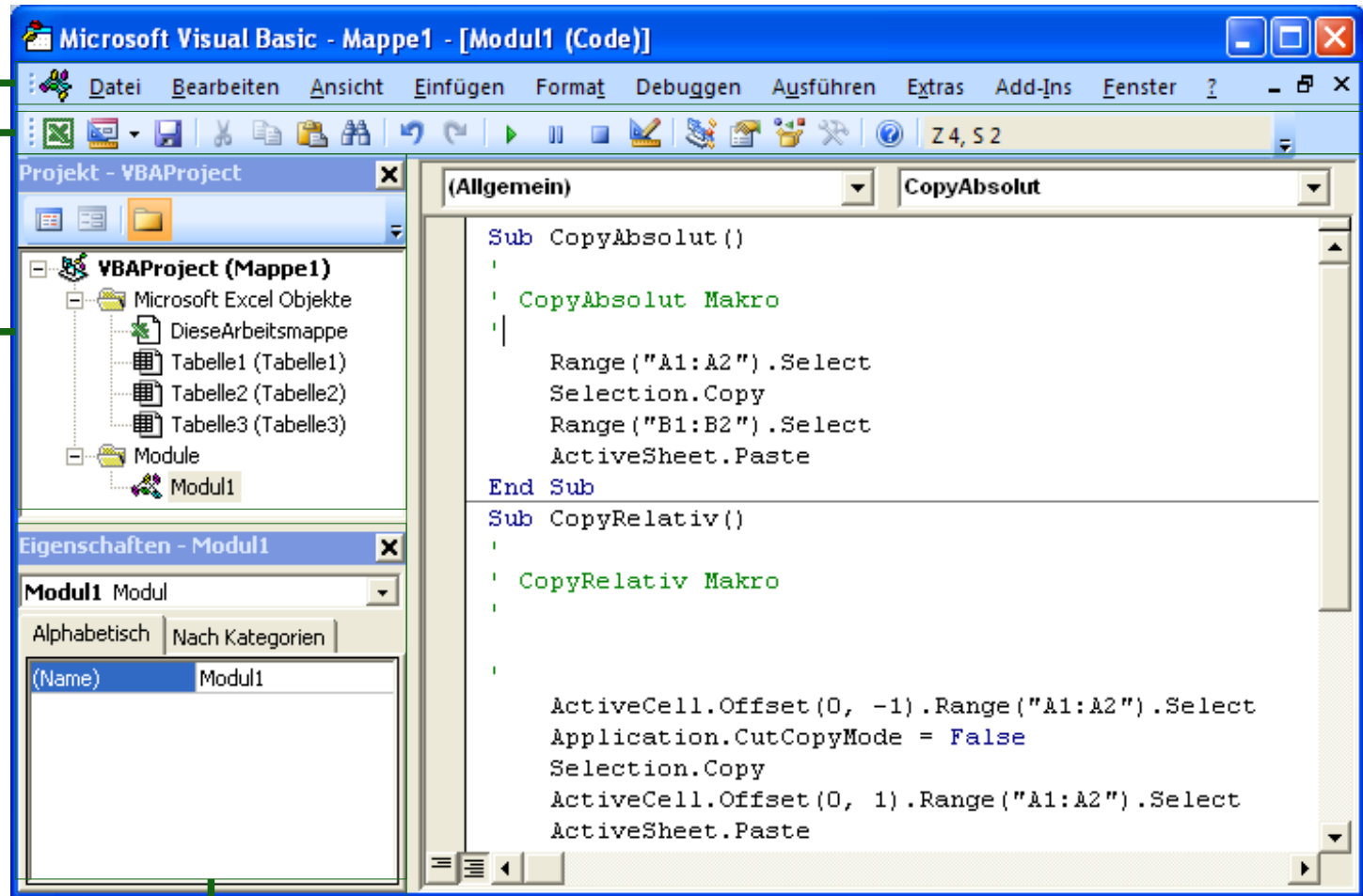
# Aufbau des VBA-Editors

Menü-  
leiste

Symbol-  
leiste

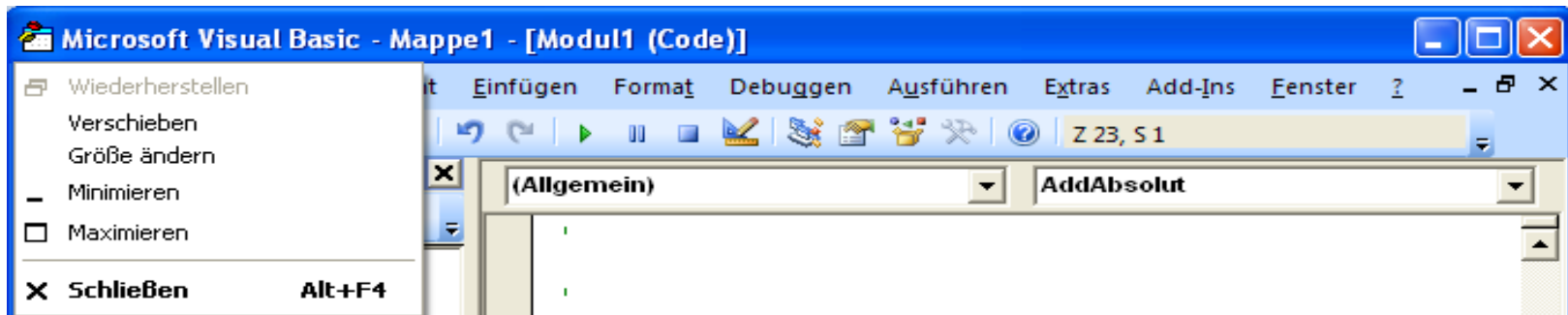
Projekt-  
Explorer

Eigen-  
schaften-  
fenster



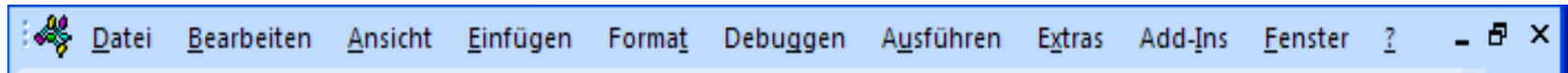
Codefenster

## Titelleiste



- Mit Hilfe des Icons am linken Rand wird das Systemmenü geöffnet. Das Systemmenü enthält Befehle zum Vergrößern und Verkleinern sowie Schließen des Editors.
- Als Titel wird der Name der Datenbank und in eckigen Klammern der Name des sichtbaren Codes angezeigt.
- Am linken Rand befindet sich die Schaltfläche zum Minimieren, Verkleinern und Schließen des Editors.

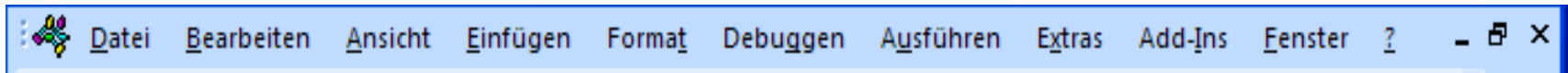
## Menüleiste



- Das Icon am rechten Rand enthält Befehle für die Ansicht des Codefensters.
- *Datei* enthält Befehle zum Speichern, Exportieren usw. von Code.
- *Ansicht* enthält Befehle zum Anzeigen von verschiedenen Elementen des Editors.
- *Extras* enthält Verweise und Optionen des VBA-Editors sowie der Datenbank.
- *Add-Ins* verbindet vorgefertigte Programme mit dem eigenen Code.
- *Fenster* richtet die Elemente innerhalb des Editors aus.



## Menüs für das Arbeiten mit VBA



- *Bearbeiten* enthält Befehle zum Durchsuchen, Einrücken usw. von Code sowie Hilfestellungen für einzelne Codeelemente.
- *Einfügen* erzeugt neue Module oder Prozeduren.
- *Debuggen* kompiliert den Code. Es können Variablen überwacht werden. Das Menü enthält Hilfsmittel zum Testen des Codes.
- *Ausführen* startet das Programm.

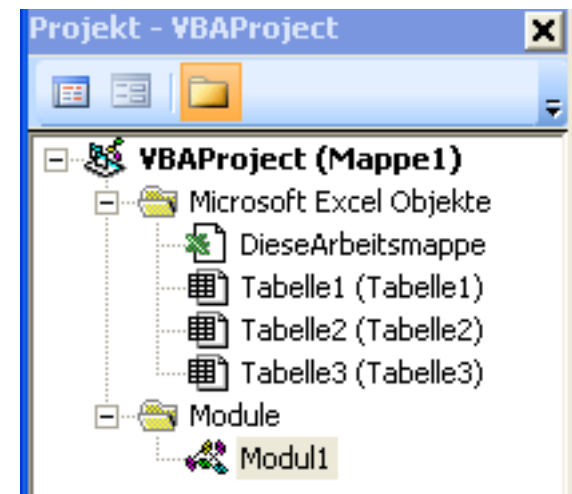
## Symbolleiste



- Wichtige Befehle werden mit Hilfe eines Symbols dargestellt.
- Die Symbole werden je nach Aufgabe deaktiviert oder aktiviert.
- Mit Hilfe der drei Punkte am linken Rand kann die Symbolleiste verschoben werden.
- Das Menü *Ansicht – Symbolleisten* blendet Symbolleisten ein oder aus.
  - Voreinstellungen enthält die wichtigsten Befehle wie zum Beispiel Speichern.
  - Es können die Symbolleisten Debuggen und Bearbeiten eingeblendet werden.
  - UserForm wird für die Entwicklung von Formularen genutzt.

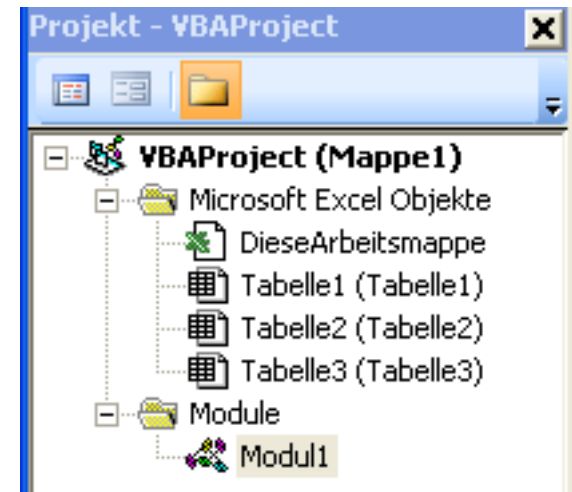
## Projekt-Explorer

- ... gibt einen Überblick über die programmierten Komponenten der Arbeitsmappe.
- ... ist die Schaltzentrale eines VBA-Projekts.
- ... erleichtert die Navigation innerhalb eines Projekts.
- ... ist frei platzierbar.
- ... enthält alle zu dem Projekt gehörenden Elemente, die Programmcode enthalten.



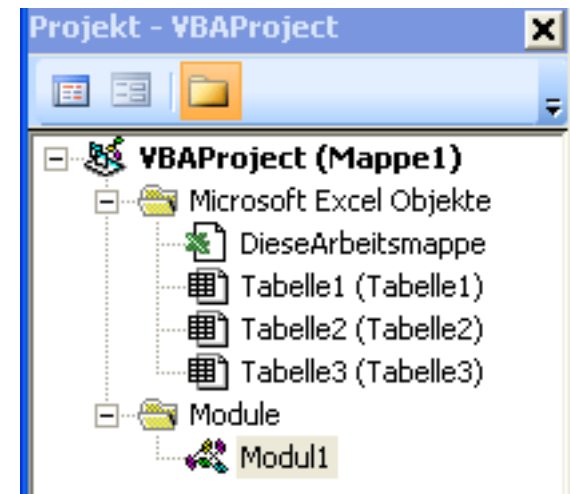
## Aufbau

- Mit Hilfe des Kreuzes in der Titelleiste kann der Projekt-Explorer geschlossen werden. Über das Menü *Ansicht* kann der Explorer eingeblendet werden.
- Darunter befinden sich von links nach rechts folgende Schaltflächen:
  - Code zum ausgewählten Element anzeigen.
  - Ausgewähltes Objekt (zum Beispiel Arbeitsmappe) anzeigen.
  - Ordnerstruktur anzeigen.



## Anzeige der Elemente

- Die Elemente werden in einer hierarchischen Struktur abgelegt.
- Als Wurzel wird immer der Name der aktuellen Arbeitsmappe angezeigt.
- Die Elemente werden in Abhängigkeit ihres Typs in Ordner zusammengefasst. Ein Minuszeichen kennzeichnet einen geöffneten Ordner. Ein Pluszeichen kennzeichnet einen geschlossenen Ordner.
- Die einzelnen Module oder Objekte entsprechen Dateien im Windows Explorer.



## VBA-Projekte

- ... enthalten Makros, Dialogfelder und die Arbeitsmappe.
- ... sind der äußere Rahmen für ein VBA-Programm.
- ... werden automatisch von einer Office-Anwendung angelegt.
- ... können nicht vom Entwickler angelegt werden.
- ... werden gemeinsam mit der Arbeitsmappe in einer Datei gespeichert.

## Objekte

- ... stellen eine Abstraktion eines realen Gegenstandes dar.
- ... sind zum Beispiel Arbeitsmappen und Arbeitsblätter
- ... haben bestimmte Eigenschaften (Attribute) und Methoden (Funktionen).
- ... reagieren auf Aktionen des Benutzers.

## Module

- ... enthalten Code, der nicht an eine Arbeitsmappe oder Arbeitsblatt gebunden ist.
- ... werden automatisch beim Erstellen von Makros angelegt. Alle Makros einer Sitzung werden in einem Modul abgelegt. Für jede neue Sitzung wird ein neues Modul angelegt.
- ... sind nicht an ein bestimmtes Objekt gebunden.
- ... zeigen Code an, der in der gesamten Arbeitsmappe genutzt wird.



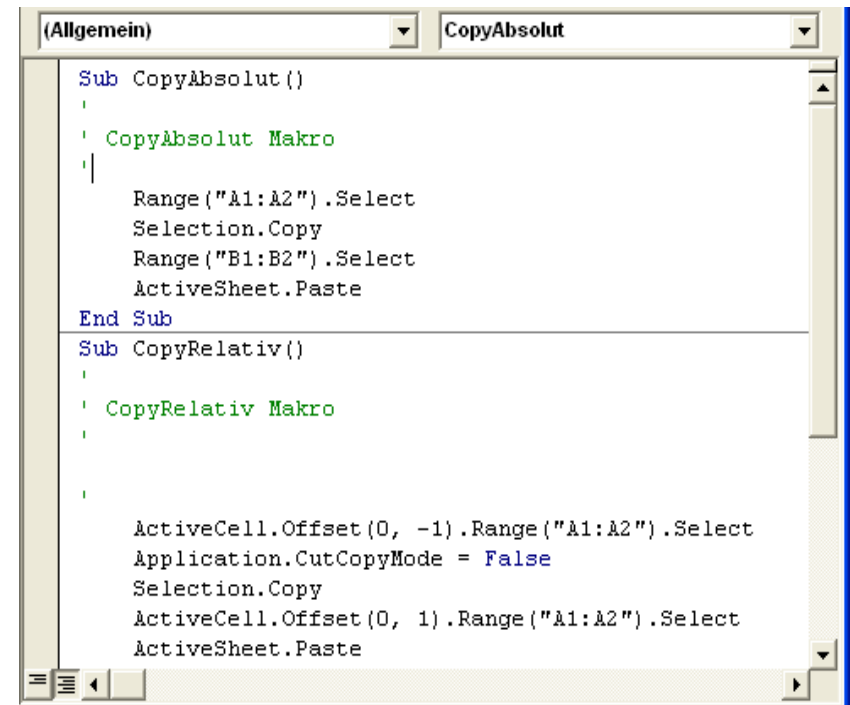
## Eigenschaftensfenster

- ... zeigt die Merkmale des aktiven Moduls oder Objekts an.
- Die gezeigten Merkmale beziehen sich immer auf das ausgewählte Element im Projekt-Explorer.
- Die Eigenschaft *Name* ist immer vorhanden. Die Klammerung des Namens zeigt an, dass ein Wert für diese Eigenschaft gesetzt werden muss.



## Codefenster

- ... zeigt den Code zu dem, im Projekt-Explorer ausgewählten Element an.
- ... zeigt ein Programm in VBA an. Der Code ist wie jeder andere Text lesbar.
- ... zeigt Befehle und Anweisungen an, die eine bestimmte Aufgabe lösen.



```
(Allgemein) CopyAbsolut  
Sub CopyAbsolut()  
    ' CopyAbsolut Makro  
    Range("A1:A2").Select  
    Selection.Copy  
    Range("B1:B2").Select  
    ActiveSheet.Paste  
End Sub  
Sub CopyRelativ()  
    ' CopyRelativ Makro  
    '   
    '   
    ActiveCell.Offset(0, -1).Range("A1:A2").Select  
    Application.CutCopyMode = False  
    Selection.Copy  
    ActiveCell.Offset(0, 1).Range("A1:A2").Select  
    ActiveSheet.Paste
```

## V(isual) B(asic for) A(pplication)

- ... ist speziell für die Programmierung in Office-Anwendungen (Word, Access, PowerPoint, Outlook) entwickelt wurden.
- ... ist an Visual Basic angelehnt.
- Merkmale:
  - VBA-Programme laufen nur innerhalb einer Office-Anwendung.
  - In welcher Office-Anwendung der Code läuft, ist dem Programm unbekannt.
  - Die Anwendungen werden als Objektmodelle implementiert.
  - Strukturierte, prozedurale Programmierung.

## ... nutzen

- Zellen automatisiert aktualisieren und verändern.
- Berechnungen automatisch ausführen.
- Makros verändern und anpassen.
- Verarbeitung von großen Datenmengen. Die Menge ist nicht bekannt, nur der Aufbau der Daten.
- Daten müssen in zeitlichen Abständen immer auf die selbe Weise verarbeitet werden.
- Andere Office-Anwendungen integrieren und nutzen.
- Kommunikation mit dem Nutzer.

## Beispiel: Zellen kopieren

```
Sub CopyAbsolut()  
,  
' CopyAbsolut Makro  
,  
    Range("A1:A2").Select  
    Selection.Copy  
    Range("B1:B2").Select  
    ActiveSheet.Paste  
End Sub
```

## Beginn und Ende des Codes für ein Makro

```
Sub CopyAbsolut()  
,  
' CopyAbsolut Makro  
,  
    Range("A1:A2").Select  
    Selection.Copy  
    Range("B1:B2").Select  
    ActiveSheet.Paste  
End Sub
```

Der Code für ein Makro beginnt mit dem Schlüsselwort Sub und endet mit der Anweisung End Sub.

Dem Schlüsselwort Sub folgt der Name des Makros. Mit Hilfe der runden Klammern können Werte an den Code übergeben werden. Die runden Klammern sind bei Makros immer leer.

## Kommentare im VBA-Code

```
Sub CopyAbsolut()  
'  
' CopyAbsolut Makro  
'  
    Range("A1:A2").Select  
    Selection.Copy  
    Range("B1:B2").Select  
    ActiveSheet.Paste  
End Sub
```

Eine Zeile wird mit Hilfe des Apostrophs auskommentiert. Die Zeile wird nicht ausgeführt. Mehrere Zeilen können mit den entsprechenden Symbolen in der Symbolleiste Bearbeiten auskommentiert werden.

## Kommentare

- ... erleichtern das Verstehen eines Programms.
- ... bieten Informationen für den Entwickler.
- ... müssen bei Codeänderungen oder Programm-Erweiterungen angepasst werden.
- ... beziehen sich auf das Warum und nicht so sehr auf das Wie.  
Beispiele:
  - Warum wird an dieser Stelle diese Berechnung durchgeführt?  
Nicht: Wie werden die Werte berechnet?
  - Warum wird der String an diesem Zeichen getrennt? Nicht:  
Wie wird der String getrennt?



## ... platzieren

- Kommentare am Anfang eines Moduls platzieren:
  - Informationen zum Entwickler.
  - Informationen zur Historie des Moduls.
- Kommentare oberhalb oder unterhalb des Prozedurkopfes (Sub... ) platzieren:
  - Welche Funktionalität wird abgebildet?
  - Ein- und Ausgabeparameter der Prozedur.
- Kommentare oberhalb oder rechts von einer Anweisung platzieren:
  - Warum wird diese Anweisung gerade ausgeführt?
  - Welches Resultat wird erwartet?

## Überflüssige Kommentare

- Kommentare sollten offensichtliche Dinge nicht in Prosa wiederholen.  
Ein negatives Beispiel:  $a = c + b$ ; ' a ist die Summe von c und b
- Kommentare, die dem Programmcode widersprechen, sind redundant.

## Zellen selektieren

```
Sub CopyAbsolut()  
,  
' CopyAbsolut Makro  
,  
    Range("A1:A2").Select  
    Selection.Copy  
    Range("B1:B2").Select  
    ActiveSheet.Paste  
End Sub
```

Die Zellen A1 und A2 werden selektiert. Range beschreibt eine Zelle, eine Zeile, eine Spalte oder einen Zellbereich auf einem Arbeitsblatt. Das Objekt bietet die Methode Select, um Zellen etc. zu markieren. Objekt und Methode werden mit einem Punkt getrennt.

## Inhalt kopieren

```
Sub CopyAbsolut()  
,  
' CopyAbsolut Makro  
,  
    Range("A1:A2").Select  
    Selection.Copy  
    Range("B1:B2").Select  
    ActiveSheet.Paste  
End Sub
```

Selection beschreibt den ausgewählten Bereich eines Arbeitsblattes. Mit Hilfe der Methode Copy wird der Inhalt des selektierten Bereichs in die Zwischenablage kopiert.

## Inhalt einfügen

```
Sub CopyAbsolut()  
,  
' CopyAbsolut Makro  
,  
    Range("A1:A2").Select  
    Selection.Copy  
    Range("B1:B2").Select  
    ActiveSheet.Paste  
End Sub
```

ActiveSheet beschreibt das aktive Arbeitsblatt. Mit Hilfe der Methode Paste wird der Inhalt der Zwischenablage in das aktuell aktive Arbeitsblatt eingefügt.

## Beispiel: Addition zweier Zellen

```
Sub AddRelativ()
```

```
'
```

```
' AddRelativ Makro
```

```
'
```

```
'
```

```
ActiveCell.FormulaR1C1 = "= R[-2]C[-2] + R[-2]C[-1]"
```

```
ActiveCell.Offset(1, 0).Range("A1").Select
```

```
End Sub
```

Eine Formel (FormulaR1C1)  
wird in die aktive Zelle  
(ActiveCell) eingegeben.

## Zellbezug für die Methode FormulaR1C1 angeben

ActiveCell.FormulaR1C1 = "= R[-2]C[-2] + R[-2]C[-1]"

- R (Row, Zeile).
  - Von der aktuellen Zelle aus wird mit Hilfe des Minuszeichen eine bestimmte Anzahl von Zeilen nach oben gegangen.
  - Von der aktuellen Zelle aus wird mit Hilfe des Pluszeichens eine bestimmte Anzahl von Zeilen nach unten gegangen.
- C (Column, Spalte).
  - Von der aktuellen Zelle aus wird mit Hilfe des Minuszeichens eine bestimmte Anzahl von Spalten nach links gegangen.
  - Von der aktuellen Zelle aus wird mit Hilfe des Pluszeichens eine bestimmte Anzahl von Spalten nach rechts gegangen.

## Versatz angeben

`ActiveCell.Offset(1, 0).Range("A1").Select`

- Ausgangspunkt: Aktive Zelle (ActiveCell).
- `Offset(1, 0)`:
  - Es wird ein Versatz von einer Zeile angegeben.
  - Die Spalte bleibt gleich.
- `Range("A1")` bezeichnet die oberste, linke Zelle des Versatz. Diese Zelle wird ausgewählt.



## Makro im VBA-Editor starten

- Die Einfügemarke befindet sich zwischen den Anweisungen Sub ... End Sub.
- Wählen Sie den Befehl *Ausführen – Sub/UserForm ausführen* im VBA-Editor aus. Andere Möglichkeiten:
  - Drücken Sie die Taste F5.
  - Grüner Pfeil in der Symbolleiste Voreinstellungen auswählen.
- Die Zeilen zwischen den Anweisungen Sub ... End Sub werden ausgeführt.