

# VBA-Syntax & Beispiele

**Achtung** “ werden hier falsch angezeigt, sie müssen immer oben stehen!!!!

## 1. Grundsätzliches

In VBA (sowie in vielen anderen Programmiersprachen) wird erst das **Objekt** (zum Beispiel eine **Zelle**) genannt mit dem etwas passieren soll, dann die **Eigenschaft** (zum Beispiel **Farbe**) und dann die konkrete **Ausprägung** (zum Beispiel **pink**).

### Beispiel

`Range(„E5“).Font.Color = 1500`

oder

`Cells(6, 5).Value = „Dies ist eine VBA-Übung“`

## Inhalt von Zellen & Formatierung

Einzelne Zellen können mit den Befehlen

- `Range(„E5“)` -> Spalte E Zeile 5  
oder mit
- `Cells(3,5)` -> 3. Zeile, 5. Spalte (E)  
angesprochen werden.

Mit können allerdings auch Bereiche festgelegt werden:

Bsp: `Range(„B3:E10“).value = „xxx“`

*schreibt in jede Zelle (also in 32 Zellen!!) des Bereiches xxx rein.*

## Eigenschaften von Zellen

Grundsätzlich können dies alle Eigenschaften sein, die Du in Excel festlegen kannst. Dies sind zum Beispiel:

**Schriftfarbe:** `Font.Color = VbColor Name`

- `Range("a1").Font.Color = vbRed`

The easiest way to set colors is with vbColors:

```
Range("a1").Font.Color = vbRed
```

However, you're very limited in terms of colors available. These are the only options available:

	A	B
1	VbColor Name	Color
2	vbBlack	
3	vbBlue	
4	vbCyan	
5	vbGreen	
6	vbMagenta	
7	vbRed	
8	vbWhite	
9	vbYellow	

- vbBlack (schwarz)
- vbBlue
- vbCyan (türkis)
- vbGreen

- vbMagenta (pink)
- vbRed
- Ivbwhite
- vbYellow

oder:

```
Range("a1").Font.Color = RGB(255,255,0)
Cells(zeile, spalte).Font.Color = RGB(0,5,100)
```

RGB steht für rot, grün, blau. Die Zahlen bewegen sich jeweils zwischen 0 und 256. Welche Zahlen eine Farbe ergeben kannst Du hier nachlesen:

<https://www.farb-tabelle.de/de/farbtabelle.htm>

### Hintergrundfarbe der Zelle *Interior.ColorIndex*.

- **Beispiele:**  
`Range("a1").Font.ColorIndex = 36`  
`Cells(zeile, spalte).Interior.ColorIndex = 45`

Der Colorindex ist eine Zahl zwischen **1 und 56**. Welche Hintergrundfarbe sich ergibt kannst Du ausprobieren.

### Umrandungen

- *.BorderAround*  
 Details zur Rahmenformatierung findest Du hier: <https://www.automateexcel.com/de/vba/zellenrahmen/>
- Bsp.:  
`Range(„C4:F15“).BorderAround LineStyle:=xlContinuous, Weight:=xlThick, Color:=vbRed`

---

 2

### Löschen

Inhalt & Formatierung löschen: *clear*

Bsp.: `Range(Cells(1, 1), Cells(5, 4)).Clear`

-> In diesem Bereich werden sowohl Inhalt als auch Formatierung gelöscht.

### Sonstiges

- Fett: *.Font.Bold = True* nicht mehr fett: *.Font.Bold = False*
- Kursiv: *.Font.Italic = True* nicht mehr kursiv: *.Font.Italic = False*
- Schriftgröße: *Font.Size*
- Schriftart: *Font.Name*
- Zeilenhöhe festlegen: `Rows(„1:1“).RowHeight = 30`
- Spaltenbreite setzen: `Columns(„A“).ColumnWidth = 30` oder `Columns(„A:G“).ColumnWidth = 15` oder `Columns(„A:B“).Autofit`

### Beispiele

- `Range("a1").Font.Bold = True`
- `Range("a1").Font.italic = True` (**kursiv**)
- `Range("a1").Font.Size = 26`
- `Range("a1").Font.Name = "Arial"`

## Zellen zentrieren und ausrichten

Erst werden die Zellen selektiert:

*Range(„a15:F16“).Select*

Dann werden die selektierten Zellen ausgerichtet zum Beispiel:

*With Selection*

*.HorizontalAlignment = xlCenter*

*.VerticalAlignment = xlBottom*

*End With*

Im Anschluss werden die selektierten Zellen verbunden:

*Selection.Merge*

## Tabellenblätter benennen

Das aktive *Tabellenblatt* wird mit *ActiveSheet.* angesprochen. Mit *ActiveSheet.Name = “Mein Name“* wird dann der Name festgelegt.

## Rechnen in VBA

Wie Du sicher weißt, geeignet Excel (und somit auch VBA) sehr gut zum rechnen. Dazu musst Du lediglich einer Zelle den Wert (Ergebnis) zuweisen.

Addition:

Beispiel: In *D10* soll nun das Ergebnis der Addition von *B10 + C10*

*Cells(10,4).value = Cells(10,3).value + Cells(10,2).value*

*Cells(10,4) = Cells(10,3) + Cells(10,2)* funktioniert auch.

Subtraktion:

Beispiel: In *D10* soll nun das Ergebnis der Addition von *B10 – C10*

*Cells(10,4) = Cells(10,3) – Cells(10,2)*

*Multiplikation/Division* funktioniert mit den Rechenzeichen *\** bzw. */*.

*Potenzieren* funktioniert mit den Rechenzeichen *^* ;

Bsp.:  $2^3 = 8$  (2 hoch 3)

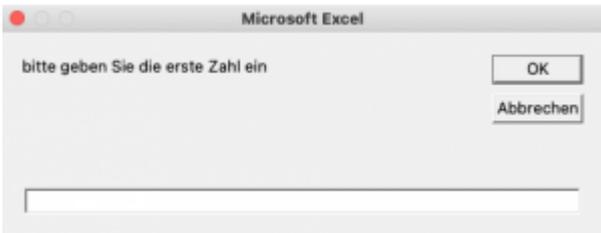
*Wurzel ziehen* funktioniert mit den Rechenzeichen *^* ;

Bsp.:  $8^{(1/3)} = 2$  -> hier die 3. Wurzel aus 8;

->  $^ 1/n$ -> n-te Wurzel

## Eingabe der Werte durch eine Inputbox

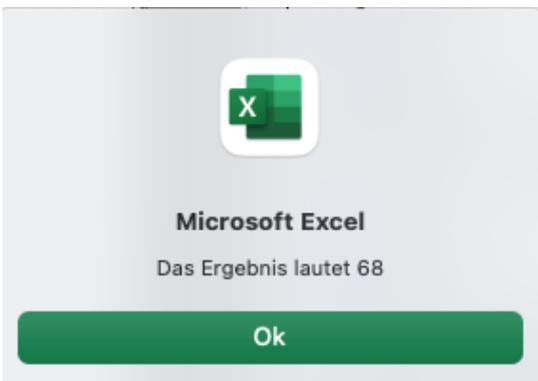
Durch den Befehl *InputBox(„bitte geben Sie die erste Zahl ein“)* erscheint diese Box.



Durch den Befehl `cells(10,2) = InputBox(„bitte geben Sie die erste Zahl ein“)` wird die Benutzereingabe (also die eingegebene Zahl) zusätzlich in die Zelle B10 geschrieben.

## Ausgabe der Werte durch eine MessageBox

Durch den Befehl `MsgBox „Das Ergebnis lautet “ & Cells(10, 4)` wird der Wert der Zelle D10 in einer MessageBox ausgegeben.



### `MsgBox „Heute ist der “ & Date“`

–> Es wird eine MessageBox mit dem heutigen Datum erzeugt

## Deklaration (=Festlegen) von Variablen

Bisher haben wir immer wenn wir Daten, die der Benutzer eingegeben hat (zum Bsp.: Schriftgröße, Schriftfarbe, etc. ) verarbeiten wollten, in eine Zelle geschrieben und haben dann durch den Cells bzw. Range-Befehl darauf zugegriffen.

Dies geschieht in der Informatik i.d.R. einfacher. Es wird eine Variable definiert, in die dieser Wert dann reingeschrieben wird.

**VBA-Syntax:** `Dim VariablenName as <Datentyp>`

### Beispiele:

`Dim schriftgroesse as integer` -> Variable wird definiert

`schriftgroesse= InputBox(„bitte geben Sie die Schriftgröße ein!“)`

-> Schriftgröße wird durch den Benutzer eingegeben und in die Variable geschrieben

`cells(1,1).Font.Size = schriftgroesse`

-> Die Schrift in der Zelle A1 wird auf diese Größe gesetzt.

Zur Festlegung (Deklaration) von Variablen muss der **Name** und der **Datentyp** festgelegt werden.

Gängige Datentypen sind u.a. **Integer** (ganze Zahl), **Double** (Zahl mit Nachkommastellen), **Date** (im Format MM/TT/YYYY) und **String** (Text, d.h. beliebige Zeichen)

**Beispiele:**

Dim KerstinsZahl as Integer

Dim KerstinsName as String

Dim KerstinsNachkommaZahl as Double

Dim KerstinsGeburtsdatum as Date

**Zuordnen von Werten in Zellen**

Dim laufzeit as Integer

laufzeit = InputBox(„Wie lange möchten Sie den Kredit abbezahlen?“)

Range(„C5“).Value = laufzeit

-> Es wird durch eine *Inputbox* die *laufzeit* abgefragt, in der Variablen *laufzeit* gespeichert und in die Zelle C5 geschrieben.

... oder mit Variablen

Dim Zeile, Spalte as Integer

Zeile = 5

Spalte=3

Cells(Zeile, Spalte ).Value = „Laufzeit“

-&gt; in C5 wird der Wert reingeschrieben.

**Zuordnen von Werten in Variablen**

Spalte = Spalte +1

-&gt; Die Variable Spalte wird um eins hochgezählt.

Jahr=Year(Date)

-&gt; Die Variable Jahr wird auf das Jahr des aktuellen Datums gesetzt.

Dim ergebnis, zahl As Double

Dim exponent As Integer

exponent = 3

zahl =2

ergebnis = zahl ^exponent

-&gt; In der Variablen ergebnis steht nun 8 (2 hoch 3)

**Selektieren von Zellen / Bereichen**

Oft müssen Zellen bzw. Bereiche selektiert (=markiert) werden, um sie zu formatieren:

**Beispiele:**

Range(„C6“).Select

Cells(2,2).Select

**Beispiel für das Selektieren von Bereichen mit Variablen:**

Dim zeileA, zeileE, spalteA, spalteE As Integer

zeileA = 5

zeileE = 10

spalteA = 2

spalteE = 4

Range(Cells(zeileA, spalteA), Cells(zeileE, spalteE)).Select  
 -> Der Bereich von B5 bis D10 wird selektiert

## Formatieren – Beispiele

Range(Cells(zeileA, spalteA), Cells(zeileE, spalteE)).Clear  
 -> In diesem Bereich werden sowohl Inhalt als auch Formatierung gelöscht.

Range(Cells(zeileA, spalteA), Cells(zeileE, spalteE)).Select  
 Selection.Font.Bold = True  
 Selection.NumberFormat = „0\_;-0 „

-> Es wird ein Bereich selektiert und dann fett und als Zahl ohne Nachkommastellen formatiert.

oder Bereich selektieren, fett und zentriert

Range(„B9:F9“).Select  
 Selection.Font.Bold = True  
 With Selection .HorizontalAlignment = xlCenter  
 End With

... als **Euro-Wert**:

Cells(zeile, spalte).Select I selektieren des Bereichs  
 Selection.NumberFormat = „\_,-\* ###0.00 [€-de-DE]\_-;\* ###0.00 [€-de-DE]\_-;\* „-,,“?? [€-de-DE]\_-;\_-@\_-“

oder als **%-Wert**

Range(„C6“).Select  
 Selection.Style = „Percent“  
 Selection.NumberFormat = „0.00%“

oder als ganze **Zahl ohne Tausenderpunkt**

Selection.NumberFormat = „0\_;-0 „