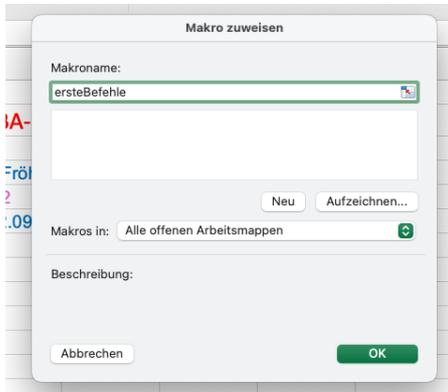


Erste Befehle in VBA

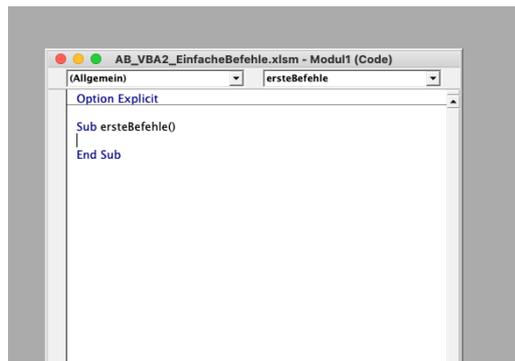
Nach dem Öffnen von Excel muss Du zunächst das Menüband anpassen. (Datei/Optionen/Menüband anpassen / Entwicklertools anklicken).

Erstelle dann als erstes eine Schaltfläche: -> **Entwicklertools/Einfügen/Schaltfläche**



Nenne das Makro **ersteBefehle**.

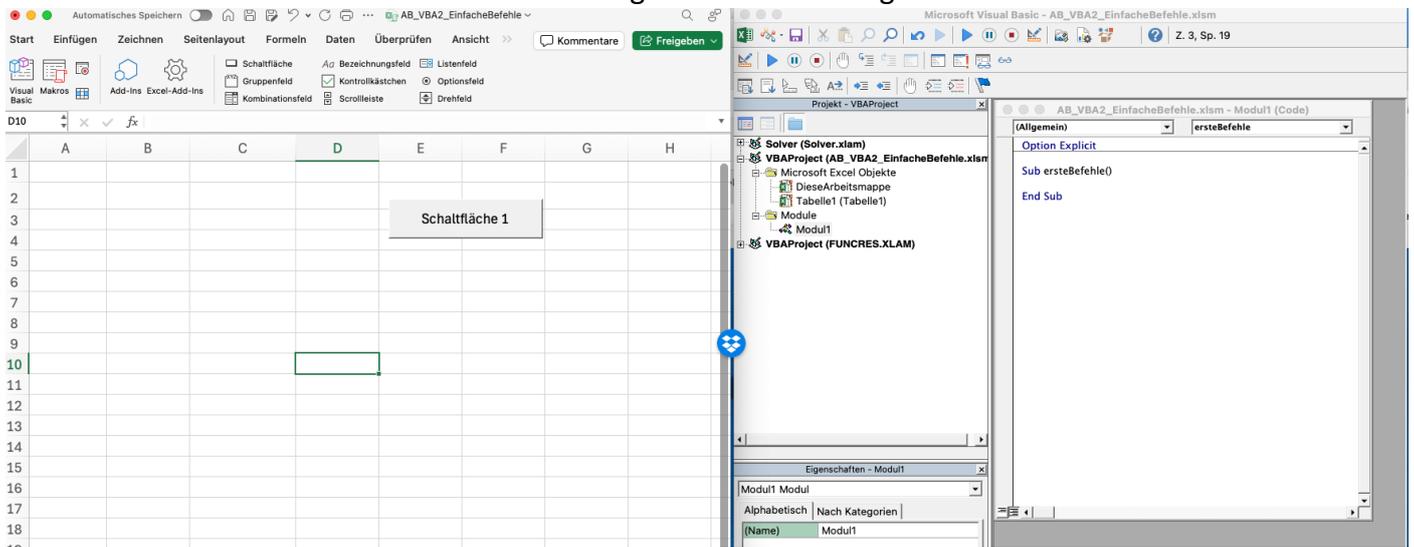
Klicke auf **Neu**.



Es erscheint dann ein Fenster.

Ordne zunächst die Fenster so an:

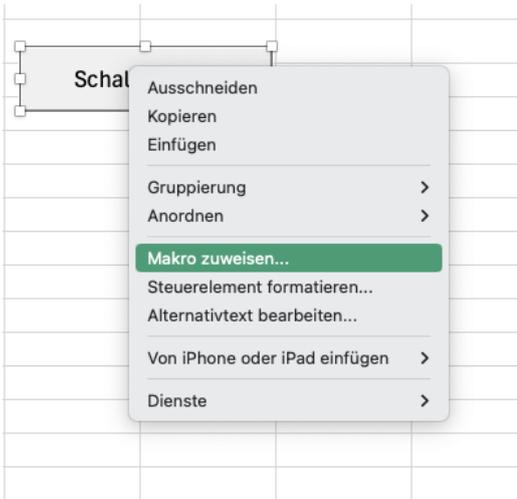
->Das Excel-Arbeitsblatt und das VBA-Fenster liegen überschneidungsfrei nebeneinander.



Durch Rechtsklick auf die Schaltfläche



kannst Du Sie umbenennen:



und auch das Makro zuweisen:

Nun fängst Du an zu programmieren ...

Schreibe die Befehle nun zwischen

Sub ... und End sub.



hier findest Du die einzelnen Befehle:

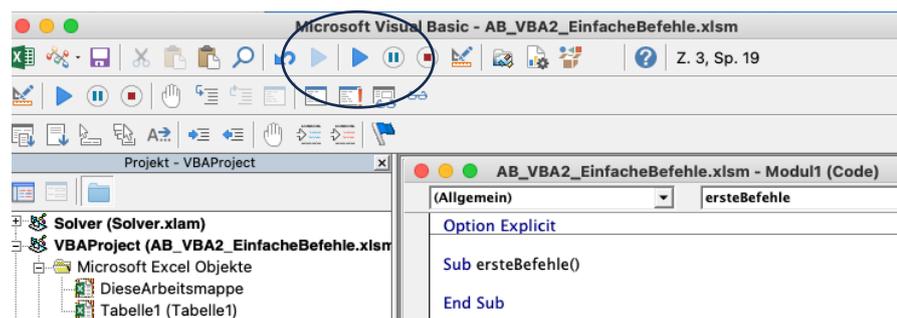
<http://bkoverretung.bplaced.net/wordpress/programmieren-in-vba/vba-syntax-beispiele/>

Zelle B2:

- Inhalt: **Meine ersten VBA-Befehle**
- Schrifttyp: **Calibri**
- Schriftgröße: **16**
- **Fett**
- Schriftfarbe: **Rot**

In die anderen Zeilen schreibst Du Deinen Namen (in Arial und die übrigen Informationen (s. Bild) und gestaltest die Schrift farbig.

Meine ersten VBA-Befehle	
Name	Kerstin Fröhlig
Klasse	W FHI12
Datum	02.09.24



Zum Testen Deiner Befehle kannst Du entweder auf die Schaltfläche klicken oder auf das Dreieck in VBA.

Speichere Deine Datei: VBA2-DeinName.xlsm (**mit MAKROS!!!!!!**)

Erstellen eines einfachen VBA-Programmes – Forts.

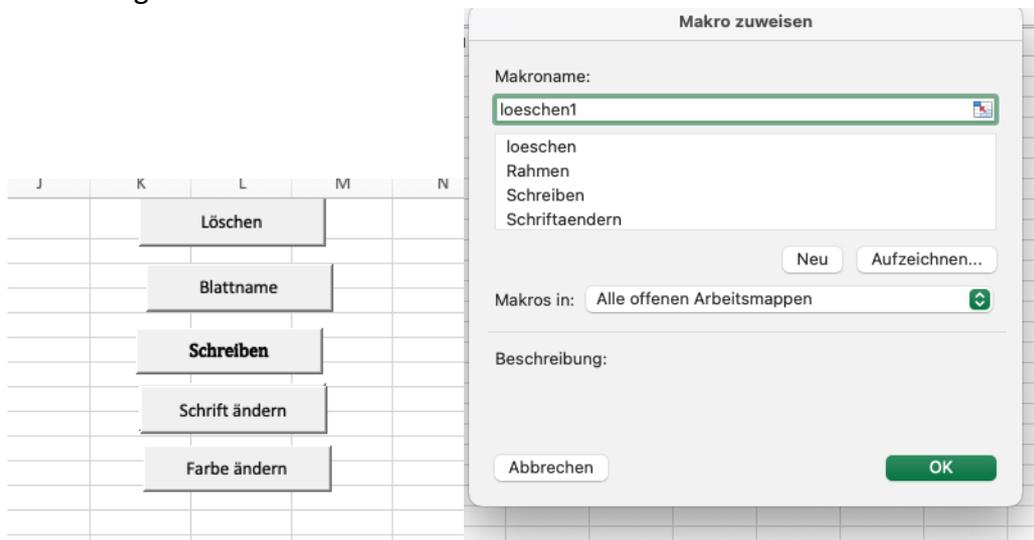
Wir wollen nun weitere Befehle kennenlernen und sie natürlich auch ausprobieren.

Ziel ist es, per Makro in eine bestimmte Zelle etwas reinschreiben, es entsprechend zu formatieren und es dann alles wieder zu löschen.

Informationen zu den erforderlichen Befehlen findest Du hier:

<http://bkoverletzung.bplaced.net/wordpress/wirtschaftsinformatik-ueberblick/programmieren-in-vba/vba-syntax-beispiele/>

1. Aktiviere als erstes die Entwicklertools.
2. Füge dann in Dein Arbeitsblatt 5 Schaltflächen ein und benenne sie entsprechend.



Füge für jede Schaltfläche ein Makro ein, d.h. klicke auf **Neu**.

3. Versuche nun die Syntax (Befehlsaufbau) für jede einzelnen Befehl rauszufinden: Dies kannst Du entweder googeln oder auf der Homepage (s.o.) suchen.

a. Zelleninhalt löschen:

b. Etwas in eine Zelle schreiben:

c. Schriftfarbe anpassen:

d. Schriftgröße anpassen:



e. Schriftart anpassen:

f. Schrift fett

g. Schrift kursiv

h. Zellenhintergrundfarbe festlegen

i. Aktuelles Blatt benennen:

4. Erstellen nun die folgenden Makros und verbinde sie mit den Schaltflächen falls noch nicht geschehen:

a. **Löschen:** Inhalt & Formatierung der Zelle C5 wird gelöscht.

b. **Blattname:** Dein Tabellenblatt bekommt einen Namen.



c. **Schreiben:** Schreibe in die Zelle C5 einen kurzen Text
(Beispiel Deinen Namen).

(zum

d. **Schrift ändern:**

i. Fett & kursiv

ii. Schriftgröße: 24 Pkt.

iii. Schrifttyp: Calibri (oder was anderes)

e. **Farbe ändern:**

i. Schriftfarbe: Grün

ii. Zellenhintergrund festlegen (wähle selbst eine Farbe, Bsp.: 7 ist pink, 8 ist türkis)

Teste Deine Makros und speichere Deine Datei.

Erstellen eines einfachen VBA-Programmes – Schriftgröße

Öffne nun Deine exceldatei von der vergangenen Stunde und ergänze dazu Deine Tabellenblatt von Aufgabe 3. Speichere Deine Datei unter dem Namen **VBA4DeinName** als Datei mit Makros.

Die Befehle findest Du hier:

<http://bkoververtretung.bplaced.net/wordpress/programmieren-in-vba/vba-syntax-beispiele/>

Nun soll der Benutzer die Schriftgröße und Hintergrundfarbe bestimmen, d.h. drückt der Benutzer auf die Schaltfläche **anpassen**, so wird die Schriftgröße der Zelle C7 entnommen und die Hintergrundfarbe der Zelle C8.

	A	B	C	D	E	F	G
4							
5			Kerstin Fröhlig			anpassen	
6							
7		Schriftgröße		18			
8		Zellenhintergrund		7	Zahl zwischen 1 und 56)		
9							
10							

➔ Füge die Schaltfläche anpassen hinzu und erstelle ein entsprechendes Makro.

	A	B	C	D
3				
4				
5			Kerstin Fröhlig	
6				
7		Schriftgröße		18
8		Zellenhintergrund		7 Zahl zw
9		Spaltenbreite		30
10		Zellenhöhe		50
11				
12				
13				

Auch die Spaltenbreite und Zellenhöhe der Zelle C5 soll nun automatisch angepasst, d.h. aus der Zelle C9 bzw. C10 entnommen werden.

Ergänze das Makro entsprechend und teste!

Ergänze:

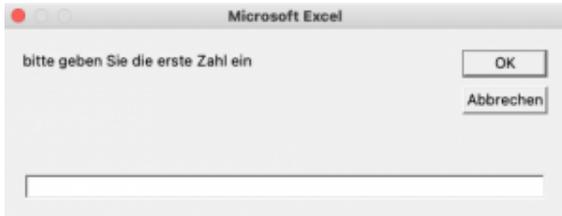
Syntax für die Spaltenbreite:

Syntax für die Zeilenhöhe:

Erstellen eines einfachen VBA-Programmes – Inputboxen

Eingabe der Werte durch eine Inputbox

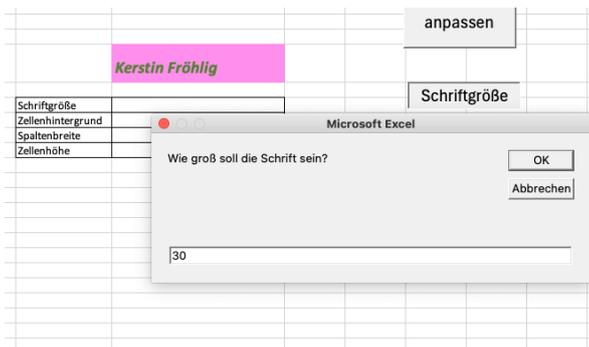
Durch den Befehl `InputBox(„bitte geben Sie die erste Zahl ein“)` erscheint diese Box.



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.

Nun soll der Benutzer die Schridftgröße bestimmen, d.h. erstellt eine Schaltfläche „**Schriftgröße**“ und ein Makro **schriftgroesse**.

Es soll sich nun eine Inputbox öffnen und der Benutzer kann die gewünschte Schriftgröße dort reinschreiben. Die Schriftgröße wird nun in die Zelle C7 geschrieben und die Schriftgröße der Zelle C5 angepasst.



Kerstin Fröhlig	
Schriftgröße	30
Zellenhintergrund	7

Die Zellenhöhe soll nun 6 Punkt größer als die Schriftgröße sein, d.h. die Zellenhöhe (Schriftgröße + 6) wird nun die Zelle C10 geschrieben und die Zellenhöhe von C5 entsprechend angepasst.

VBA-Programme: Befehle Cells und Range

Mit den Befehlen **Cells** und **Range** können Zellen angesprochen werden.
Beide haben eine unterschiedliche **Syntax**:

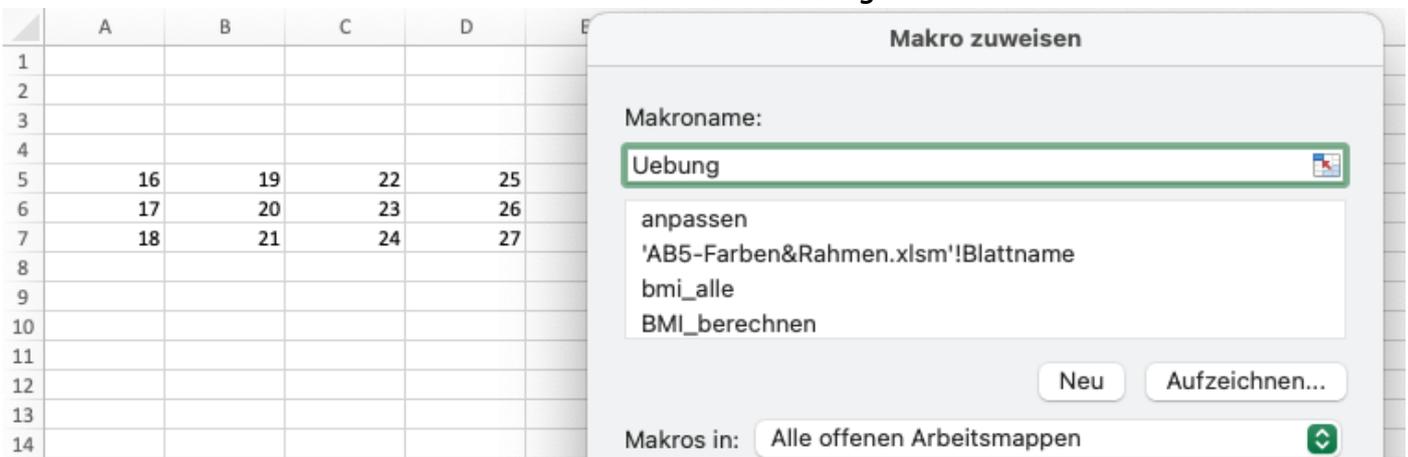
Cells(Zeile, Spalte) beides als Zahl
Range (SpalteZeile) sowie in Excel üblich

Beispiel Zelle B10:

Cells (10,2)
Range(„B10“)

Aufgabe 1:

Öffne Excel und erstelle eine Schaltfläche mit einem Makro **uebung**



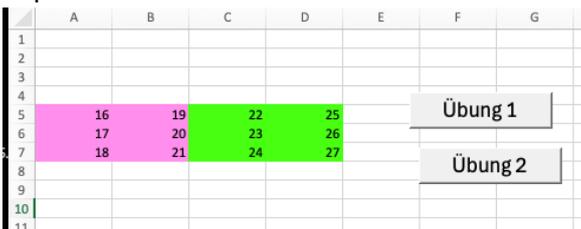
Die Zahlen 16 bis 21 auf dem Excelarbeitsblatt sollen mit dem Befehl **Range** und die anderen Zahlen mit dem Befehl **Cells** in die Zellen geschrieben werden.

Tipp: Das Kopieren von Befehlen (str c und str v) erleichtert das Programmieren erheblich.

Aufgabe 2:

Mit dem Befehl Range können sowohl einzelne Zellen als auch ganze Bereiche angesprochen werden.
Bsp. Range („B2:C6“).

Füge nun eine weitere Schaltfläche hinzu und programmiere ein Makro, das die Zellhintergrundfarbe anpasst.



Befehl:

VBA-Programme: Hintergrundfarbe & Schriftfarbe (1)



Mit dem Befehl `Cells(3, 4).Font.Color = RGB(0,5,100)` wird die Schriftfarbe und mit dem Befehl `Cells(3, 4).Interior.ColorIndex = 45` wird die Hintergrundfarbe der Zelle D3 angepasst.

<http://bkoververtretung.bplaced.net/wordpress/programmieren-in-vba/vba-syntax-beispiele/>

Aufgabe

1. Öffne Deine VBA-Datei, füge ein weiteres Tabellenblatt „**Farben**“ hinzu. Erstelle dann eine Schaltfläche mit weise dieser das Makro aus Aufgabe 2 zu (hier werden Zahlen in Zellen geschrieben). Ergänze dieses Makro so, dass auch die Zellen A8 bis D9 gefüllt werden.
- | | A | B | C | D | E | F | G |
|----|----|----|----|----|---|-------|---|
| 3 | | | | | | | |
| 4 | | | | | | Übung | |
| 5 | 16 | 19 | 22 | 25 | | | |
| 6 | 17 | 20 | 23 | 26 | | | |
| 7 | 18 | 21 | 24 | 27 | | | |
| 8 | 28 | 30 | 28 | 28 | | | |
| 9 | 29 | 31 | 32 | 33 | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |
| 12 | | | | | | | |
2. Erstelle nun ein Makro und eine Schaltfläche mit dem Namen **Hintergrund**. Ordne jetzt jeder Zelle mit dem Befehl `Cells(...).Interior.ColorIndex = ...` eine unterschiedliche Hintergrundfarbe zu. (Zahl zwischen 1 und 56).
 3. Erstelle nun ein Makro und eine Schaltfläche mit dem Namen **Schriftfarbe**. Jetzt soll jede Zelle eine andere Schriftfarbe erhalten. Benutze dazu den Befehl `Cells(...).Font.Color = RGB(...)`. RGB steht für Red, Green, Blue und muss eine Zahl zwischen 0 und 255 sein. Eine Farbübersicht findest Du hier: <https://www.farb-tabelle.de/de/farbtabelle.htm> Beachte: Die Zahlen sollten noch lesbar sein.
 4. Speichere Deine Arbeit.



Tipp: Kopieren erleichtert das Leben!!!! (Str c und Str v)

VBA-Programme: Hintergrundfarbe & Schriftfarbe (2)

1. Füge eine Schaltfläche (**init**) und ein Makro mit dem Namen **farbenInit** ein.

Dieses Makro soll

- ➔ die Zellen von A5 bis D11 löschen
- ➔ und dann folgende Werte in die Zellen schreiben s. Bild),
- ➔ zudem A8 und B7 fett.
- ➔ In die Zelle C5 euren Namen schreiben in kursiv und Schriftgröße 20.

6				
7		Hintergrundfarbe		Zahl zwischen 1 und 56)
8	Schriftfarbe	Rot		Zahl zwischen 10 und 255
9		Grün		
10		Blau		
11				
12				

2. *Schreibe nun Werte in die Zellen C7 bis C10.*

Hintergrundfarbe	34	Zahl
Rot	0	Zahl
Grün	123	
Blau	10	

3. Erstelle nun ein Makro **Schriftfarbe2** und eine Schaltfläche mit dem Namen **Schriftfarbe2**. Nun soll die Schrift- und die Hintergrundfarbe auf die entsprechenden Werte gesetzt werden.

VBA-Programme: Formatieren von Zellen



Mit dem Befehl **Format** wird eine Zelle formatiert.

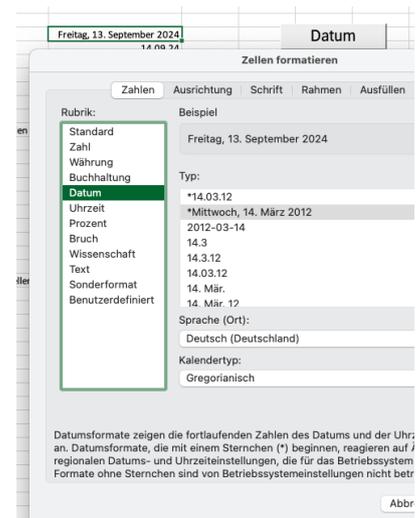
Da wir die Befehle noch nicht kennen und sie nicht ganz so einfach sind, versuchen wir es jetzt mal mit dem Aufzeichnen von Makros.

Aufgabe 1

- Füge eine Schaltfläche (**Datum**) und ein Makro mit dem Namen datum ein.
 - ➔ Die Zelle D3 soll das Datum von heute (geht mit **date**) erhalten.
 - ➔ Die Zelle D4 soll das Datum von morgen (geht mit **date+1**) erhalten.
 - ➔ Die Zelle D5 soll das Datum von übermorgen (geht mit **date+2**) erhalten.

- Jetzt sollen die Daten so formatiert werden:

Freitag, 13. September 2024	Datum lang
14.09.24	Datum kurz
15.9.24 12:00 AM	Datum / Uhrzeit in Englisch



- Da wir leider nicht wissen, wie das geht werden wir diese Befehle von Excel selbst erstellen lassen und sie anschließend in unser Makro kopieren. Dazu sind folgende Schritte notwendig:
 - ➔ Gehe auf Makro aufzeichnen und nenne diese Makro **Datumxy** (das löschen wir nachher wieder)
 - ➔ Formatiere die Felder entsprechend in Excel (Start) oder Zelle selektieren/rechte Maustaste/Zellen formatieren.
 - ➔ Beende die Aufzeichnung des Makros.
 - ➔ Kopiere die Befehle nun in Dein Makro und löschen das Makro **Datumxy**.

Aufgabe 2

- ➔ Erstelle nun **drei** weitere Schaltflächen und Makros zur Formatierung von Zahlen, Eurobeträge und Prozent.
- ➔ Schreibe in die Zellen C10 bis C12: 99999,99
- ➔ Schreibe in die Zellen C17 bis C18: 999,99
- ➔ Schreibe in die Zellen C22 bis C23: 0,89
- ➔ Zeichne jetzt wieder ein Makro auf und formatiere die Zellen in Excel.

	A	B	C	D	E	F	G
1							
2							
3		Datum					
4		Datum lang	Freitag, 13. September 2024				Datum
5		Datum kurz	14.09.24				
6		Datum englisch	15.9.24 12:00 AM				
7							
8		Zahlen					
9		Ganze Zahl	100000				Zahlen
10		mit zwei Nachkommastellen	99999,99				
11		mit Tausenderpunkten	99.999,99				
12							
13							
14		Euro					
15		Voller Eurobetrag	1.000 €				Euro
16		mit Cent	999,99 €				
17							
18		Prozent					
19		mit zwei Nachkommastellen	89%				Prozent
20		Ganze Zahl	89,00%				
21							
22							
23							
24							

- ➔ Kopiere nun die Befehle in die entsprechenden Makros. Unnötige Befehle lässt Du dabei einfach weg.

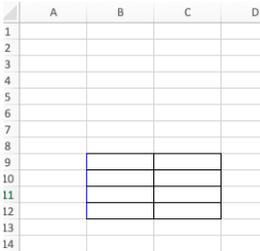
VBA-Programme: Umranden von Zellen und Zellbereichen



Umrandungen

Details zur Rahmenformatierung findest Du

hier: <https://www.automateexcel.com/de/vba/zellenrahmen/>



Möchte man mit VBA Umrandungen erstellen, so ist jede Linie der Zelle (oben, unten, rechts, links) einzeln festzulegen. Die Zellen werden zunächst selektiert und bekommen dann die entsprechende Eigenschaft.

Zeichnest Du ein Makro auf, während Du ein paar Zellen markierst und sie umrandest bekommst du folgenden Code:

Code	Was bewirkt das?
<pre>Range("B9:C12").Select With Selection.Borders(xlEdgeLeft) .LineStyle = xlContinuous .ColorIndex = 0 .TintAndShade = 0 .Weight = xlThin End With With Selection.Borders(xlEdgeTop) .LineStyle = xlContinuous .ColorIndex = 5 .TintAndShade = 0 .Weight = xlThin End With With Selection.Borders(xlEdgeBottom) (...) End With With Selection.Borders(xlEdgeRight) ... End With With Selection.Borders(xlInsideVertical) ... End With With Selection.Borders(xlInsideHorizontal) ... End With</pre>	<p>Es wurden die Zellen B9 bis C12 selektiert</p> <p>With Selection -> bezieht sich immer auf die selektierten Zellen.</p> <p>xlEdgeLeft: in Klammern steht die Position der Linie, um welche es hier handelt. (s. unten)</p> <p>.LineStyle: Linienart</p> <p>.TintAndShade: Farbe heller oder dunkler</p> <p>.ColorIndex: Farbe der Linie</p> <p>.Weight: Stärke der Linie</p> <p>End With: Ein With endet immer mit end with</p> <p>Für jede Linienposition werden nun diese Eigenschaften festgelegt. Welche Werte diese annehmen können, findest du unten.</p>

Linienposition:

- xlEdgeLeft (links senkrecht)
- xlEdgeTop (oben)
- xlEdgeBottom (unten)
- xlEdgeRight (rechts senkrecht)
- xlInsideVertical & xlInsideHorizontal (diagonal Linien)

Linienart:

Eigenschaft `LineStyle` kann in `xlContinuous`, `xlDot`, `xlDash`, `xlDashDot`, `xlDashDotDot`, `xlDouble`, `xlSlantDashDot` und `xlLineStyleNone` geändert werden.

Linienstärke:

Die Eigenschaft `Weight` kann entweder `xlHariline`, `xlThin`, `xlMedium` oder `xlThick` entsprechen.

TintAndShade (eher unwichtig)

Es kann eine Zahl von -1 (dunkelster Wert) bis 1 (hellster Wert) für die **TintAndShade**-Eigenschaft eingegeben werden. Der Wert 0 (null) ist neutral.

Linienfarbe

Die Linienfarbe kann auf unterschiedliche Weise angegeben werden auch zum Beispiel so:

`Color = RGB(128, 0, 0)`

Umrandungen außen:

Mit `BorderAround` kann auch eine Umrandung außen erstellt werden.

Bsp.: `Range("B5:C7").BorderAround LineStyle:=xlContinuous, Weight:=xlThick, Color:=vbRed`

Aufgaben

1. Öffne Deine Excel-Arbeitsmappe, füge ein neues Tabellenblatt **Umrandung** hinzu.
2. Erstelle nun eine Schaltfläche (Umrandung 1) und zeichne ein Makro auf, während Du die Zellen B9 bis C11 umrandest.
3. Erstellen nun 4 weitere Schaltflächen und Makros (Umrandung 2 bis 5) und kopiere den Code aus dem Makro Umrandung 1 dort hinein.
4. Passe die Makros nun entsprechend an:
 - a. **Umrandung 2:** Diese beziehen sich auf die Zellen A2 bis D4, hier sollen die Linien allerdings dicker und in rot sein.
 - b. **Umrandung 3:** Diese beziehen sich auf die Zellen A2 bis D4, hier sollen die Linien oben und unten allerdings gepunktet und in grün auch etwas dicker, die senkrechten Linien durchgehend, etwas dünner und in blau sein.
 - c. **Umrandung 4:** Diese beziehen sich auf die Zellen A14 bis D8, hier kannst Du Dir selber etwas ausdenken, allerdings soll der Zellbereich außen Dick umrandet sein.
 - d. **Umrandung 5:** Diese beziehen sich auf die Zellen F1 bis J5, hier kannst Du Dir selbst etwas ausdenken.
5. Speichere **Deine Arbeit!**

VBA-Programme: Taschenrechner mit Variablen



Programmiere einen Taschenrechner in Excel mit genau diesen Makros /Schaltflächen

1. Clear

Es werden alle Felder im Bereich A1 bis G25 gelöscht.

2. Init

Mein Taschenrechner		
Erste Zahl	Zweite Zahl	Ergebnis
<input type="text"/>	<input type="text"/>	<input type="text"/>

Nach dem Ausführen sieht die Tabelle so aus!
Gerne auch in bunt und mit Hintergrundfarben!

3. Zahlen eingeben (Makro: number)

Es öffnen sich Inputboxen und der Benutzer gibt die entsprechenden Zahlen ein. Diese werden in die Tabelle geschrieben.

4. Add

Es werden drei Variablen (zahl1, zahl2, eergebnis) definiert.

DIM zahl1, zahl2, ergebnis as Double

Die Variable zahl1 wird auf den Wert der eingegebenen Zahl 1 gesetzt, Zahl 2 analog.

zahl1 = Range("b5").Value (...)

ergebnis = zahl1 + zahl2

Es wird ein + in c5 geschrieben.

Es wird das Ergebnis in die Zelle f5 geschrieben.

Es wird eine Meldung ausgegeben. ->MsgBox "Das Ergebnis lautet " & Range("f5").Value



5. minus

analog zu Add

6. multiply

analog zu Add

7. divide

analog zu Add

8. potenz

(ergebnis = zahl1^zahl2)

9. Wurzel

(ergebnis = zahl1^(1/zahl2))

Tipp: Kopiere für die Makros 5 bis 9 das Makro Add und passe es entsprechend an.





Scheifen