



VBA-Objekte



Objektorientierte Programmierung

- **Objekt** – ein Exemplar eines bestimmten "Datentyps"
- eigene Objekte: z.B. Aktie, Fahrzeug, etc. (hier aus Zeitgründen nicht behandelt)
- in der Office-Anwendung vorhandene Objekte: in Excel z.B. "Tabellenblatt", "Tabellenspalte" etc.
- Ein Objekt hat einen **Zustand** und ein **Verhalten**.



Zustand = Attribute

Der Zustand wird beschrieben durch Attribute



Typ

Kennzeichen

Kilometerstand

Hubraum

	A	B
1	Januar	10,5
2	Februar	8,5
3	März	7,3
4	April	9
5	Mai	-0,3
6	Juni	4

Inhalt

Schriftfarbe

Schriftart

Hintergrundfarbe



Verhalten = Methoden

Methoden sagen, was man mit dem Objekt tun kann.



Halter ändern
Fahrtenbuch drucken
zur Inspektion anmelden

	A	B
1	Januar	10,5
2	Februar	8,5
3	März	7,3
4	April	9
5	Mai	-0,3
6	Juni	4

löschen
kopieren
Inhalte einfügen



	A	B	C	D
1	Januar	10,5		
2	Februar	8,5		
3	März	7,3		
4	April			
5	Mai			
6	Juni			
7	Juli			
8	August			
9	September			
10	Oktober	1		
11	November	1		
12	Dezember	2		
13				
14				

Calibri 11 A A % 000

F *K* [List Icon] [Grid Icon] [Color Picker] [Text Color] [Align Left] [Align Center] [Align Right]

- Ausschneiden
- Kopieren
- Einfügen
- Inhalte einfügen...
- Zellen einfügen...
- Zellen löschen...
- Inhalte löschen
- Filter >
- Sortieren >
- Kommentar einfügen
- Zellen formatieren...
- Dropdown-Auswahlliste...
- Bereich benennen...
- Hyperlink...

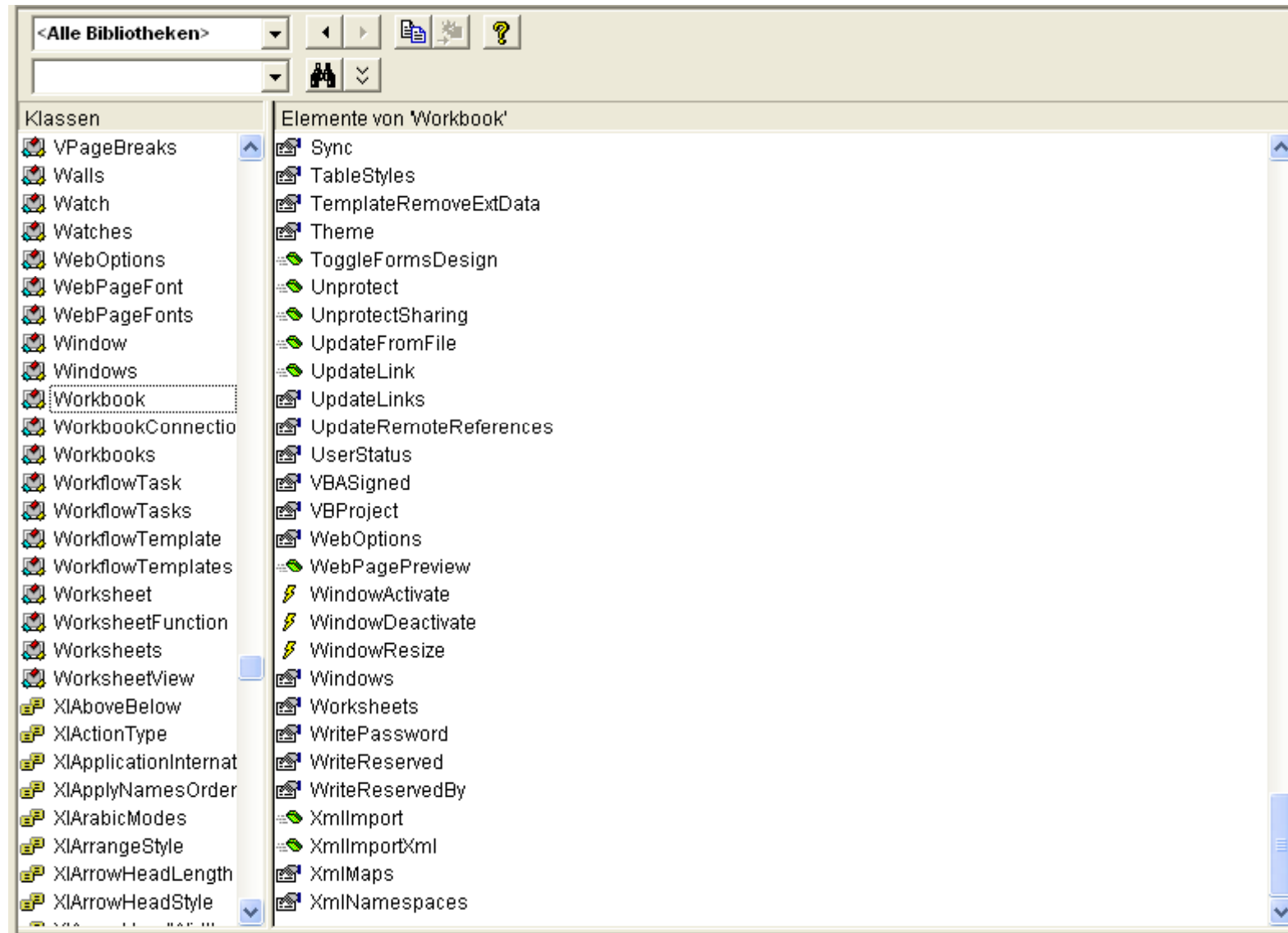


Arbeit mit Objekten

- Debug.Print
Aufruf der Print-Methode des Objekts Debug
("Direktfenster")
- ActiveWorkbook.Path
liefert den Pfad, in dem die aktuelle Excel-
Arbeitsmappe liegt
Zugriff auf das Attribut Path (Wo ist die Excel-
Datei gespeichert?) des Objekts ActiveWorkbook
("die gerade geöffnete Excel-Arbeitsmappe")



Objekte in Excel – Objektkatalog mit F2 öffnen







Objekte in Excel - Workbook



kosten.xls
Microsoft Excel-Arbeitsblatt
27 KB

Objekt: Arbeitsmappe (Workbook), bezeichnet eine Excel-Datei

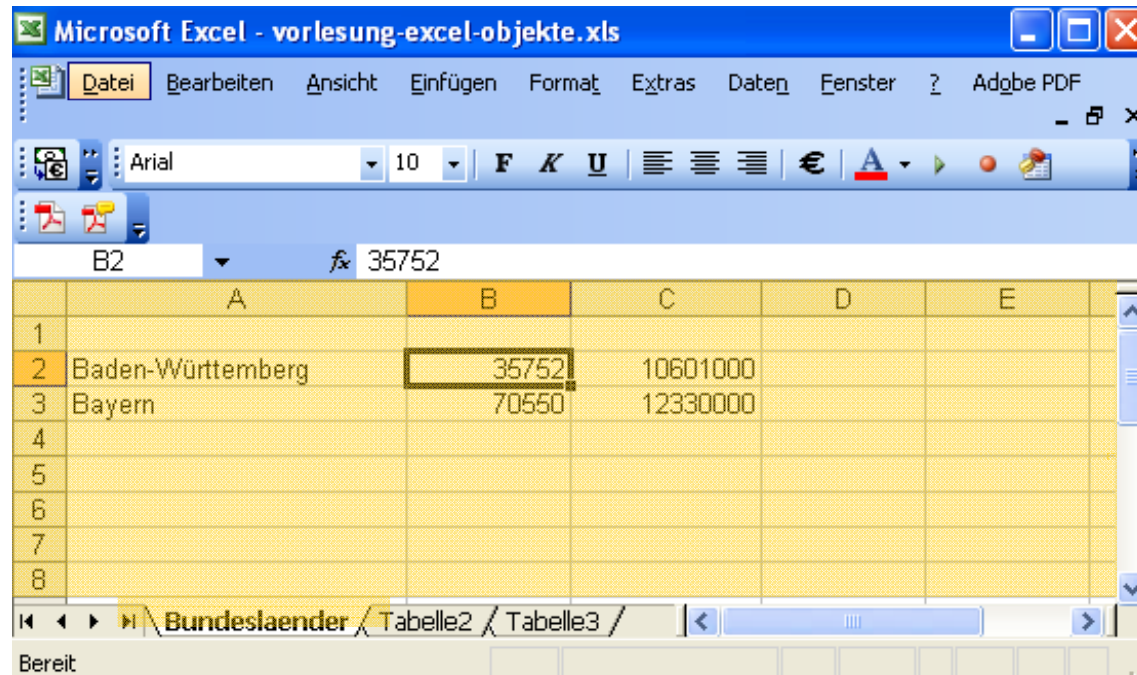
Attribute: z.B. Dateiname

enthaltene Objekte: besteht aus einzelnen Arbeitsblättern und Diagrammen

Methoden: z.B. Öffnen, Speichern



Objekte in Excel - Worksheet



Objekt: Arbeitsblatt (Worksheet)

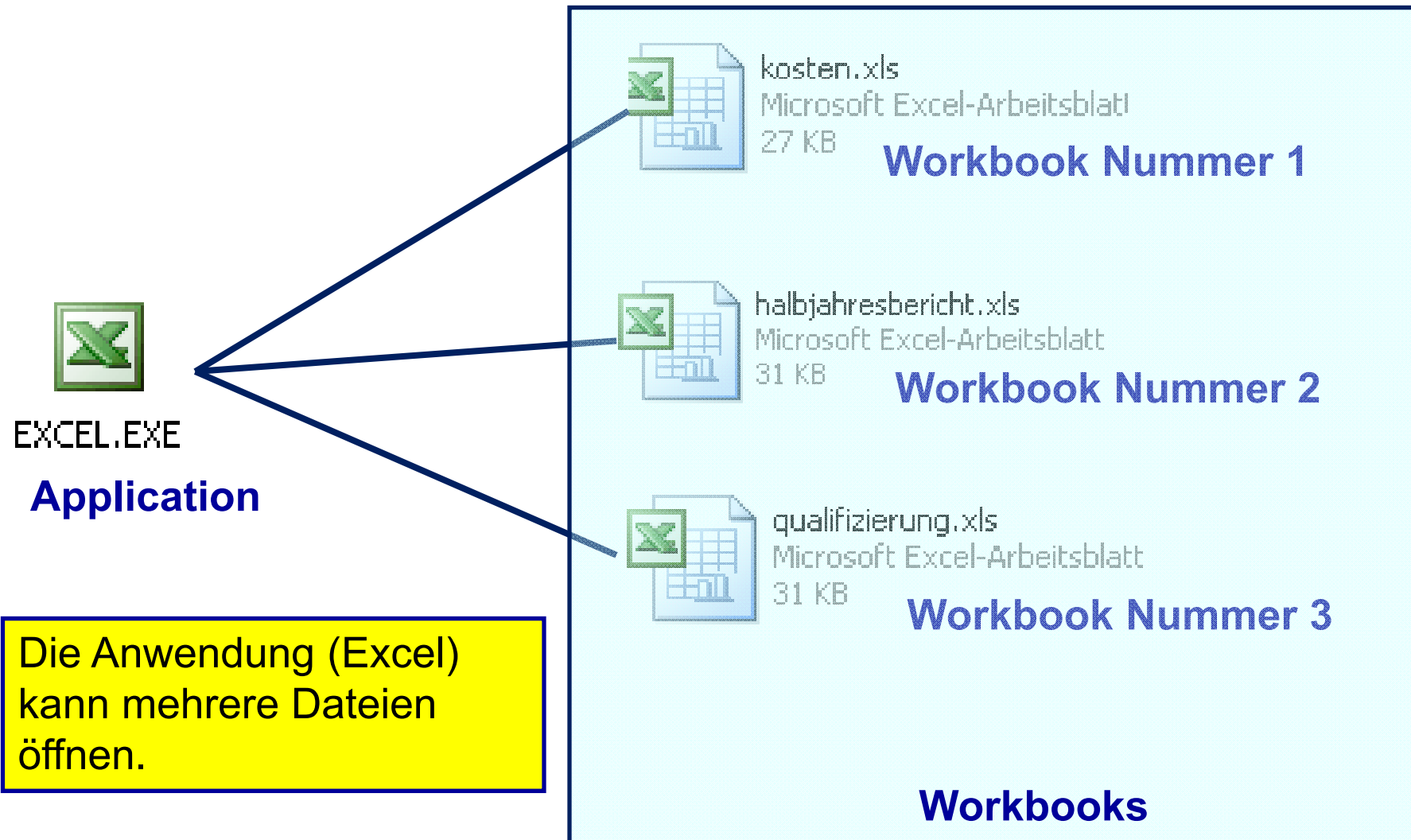
Attribute: hat einen Namen

enthaltene Objekte: besteht aus einzelnen Zellen

Methoden: z.B. Arbeitsblatt kopieren



Objekthierarchie





kosten.xls
Microsoft Excel
27 KB

Workbook

Eine Excel-Datei
kann mehrere
Arbeitsblätter
enthalten.

Worksheet Nummer 1

	A	B	C	J	K	L	M
1	16,22	2	32,44				
2	Summe		430,55				
3							

Worksheet Nummer 2

	A	B	C	D	E	F	G
1	77,25						
2	offen:	0					
3							

Worksheet Nummer 3

	A	B	C	D	E	F	G
1	Firma Pfeiffer	760	bezahlt!				
2	MPK	886	bezahlt!				
3							

Worksheet Nummer 4

	A	B	C	D	E	F	G
1	660	(Schätzung)					
2							
3							

Worksheets



Aufzählungen (Collections)

- Beispiel: Eine Arbeitsmappe (Workbook) enthält mehrere Arbeitsblätter (Worksheets).
Also enthält das Objekt Workbook ein Attribut Worksheets.
- Dieses enthält (analog zu Feldern) mehrere Arbeitsblätter:
- `ActiveWorkbook.Worksheets(1)` – erstes Arbeitsblatt
- `ActiveWorkbook.Worksheets(2)` – zweites Arbeitsblatt
- ...



Debug.Print

ActiveWorkbook.Worksheets(2).Name

gibt den Namen des zweiten Arbeitsblatts aus.

Dim blatt As Worksheet

Variable vom Typ
"Worksheet-Objekt"

For Each blatt In ActiveWorkbook.Worksheets

Debug.Print blatt.name

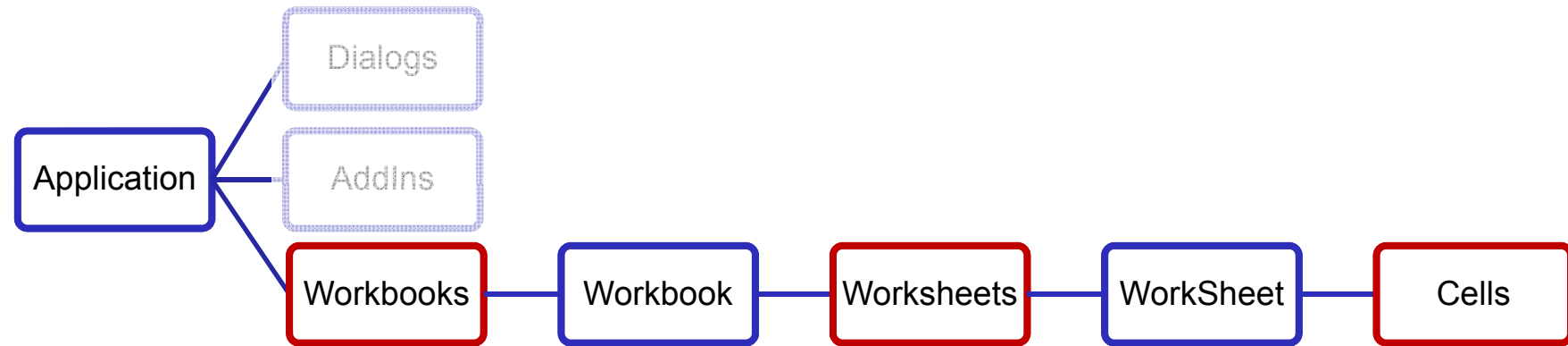
Next

Aufzählung (Collection):
(alle Arbeitsblätter)

gibt die Namen aller Arbeitsblätter aus



Objekthierarchie



Excel-
Anwendung

Arbeitsmappen
(Excel-Dateien)

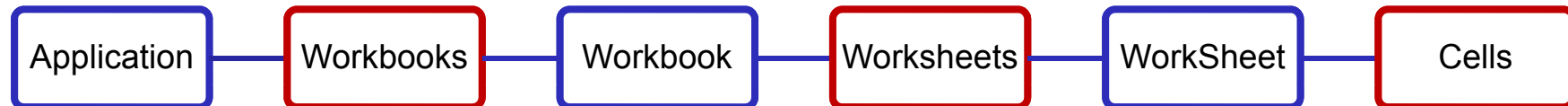
Arbeitsblätter

Tabellen-
Zellen



Zugriff auf ein Objekt

- ausgehend von der "Spitze", also dem Application-Objekt.
- Treffen wir "unterwegs" auf eine Aufzählung in der Abb. (rot), muss in Klammern angegeben werden, welches Objekt dieser Aufzählung genommen werden soll





Benutzen / Setzen von Feldinhalten in Excel-Tabellen

```
Application.Workbooks(1).Worksheets(2).Cells(1,2).Value =  
"ABC"
```

die 1.
geöffnete
Datei

das 2.
Arbeitsblatt

die Zelle in
Zeile 1,
Spalte 2

Attribut
"Value" =
Zelleninhalt

analog:

```
x =Application.Workbooks(1).Worksheets(2). Cells(1,2).Value
```

(Anmerkung: .Value kann auch weggelassen werden.)



Zugriff auf "aktuelle" Objekte

einzelne Objekte:

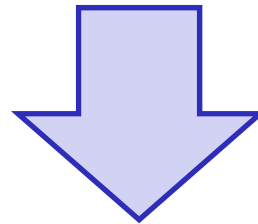
- ActiveWorkbook - die gerade aktive Arbeitsmappe
- ActiveSheet - das gerade aktive Arbeitsblatt
- ActiveCell – die gerade aktive Zelle (die Zelle, in der der Cursor steht)

Aufzählungen von Objekten:

- Workbooks - Aufzählung der geöffneten Arbeitsmappen
- Worksheets - Aufzählung der geöffneten Arbeitsblätter
- Cells - Zellen im aktuellen Arbeitsblatt



```
Application.Workbooks(1).Worksheets(2)  
.Cells(1,2).Value = "ABC"
```



```
Cells(1,2) = "ABC"
```



Debug.Print ActiveWorkbook.Saved

gibt aus, ob schon alle Änderungen an der Excel-
Datei gespeichert wurden



Aktivieren eines Arbeitsblattes

`ActiveWorkbook.Worksheets("Bundeslaender").
Activate`



Das Range-Objekt

Mit Cells(zeile,spalte) können wir auf eine einzelne Zelle zugreifen. Oft möchte man aber mit mehreren Zellen arbeiten, dafür gibt es das Range-Objekt.

einfachstes Beispiel:

`Range("A1") = 89` entspricht

`Cells(1,1) = 89`

bzw. der Langform

`Application.ActiveWorkbook.ActiveSheet
.Cells(1,1).Value = 89`



Range("B1")

	A	B	C	D	E
1	x	x	x	x	x
2	x	x	x	x	x
3	x	x	x	x	x
4	x	x	x	x	x
5	x	x	x	x	x
6	x	x	x	x	x
7	x	x	x	x	x
8	x	x	x	x	x
9	x	x	x	x	x
10	x	x	x	x	x
11					
12					



Range("A1:B4") oder Range("A1","B4")

	A	B	C	D	E
1	x	x	x	x	x
2	x	x	x	x	x
3	x	x	x	x	x
4	x	x	x	x	x
5	x	x	x	x	x
6	x	x	x	x	x
7	x	x	x	x	x
8	x	x	x	x	x
9	x	x	x	x	x
10	x	x	x	x	x
11					
12					



Range("A1,B2,C3")

	A	B	C	D	E	F
1	x	x	x	x	x	
2	x	x	x	x	x	
3	x	x	x	x	x	
4	x	x	x	x	x	
5	x	x	x	x	x	
6	x	x	x	x	x	
7	x	x	x	x	x	
8	x	x	x	x	x	
9	x	x	x	x	x	
10	x	x	x	x	x	
11						
12						



Range("B:B")

	A	B	C	D	E
1	x	x	x	x	x
2	x	x	x	x	x
3	x	x	x	x	x
4	x	x	x	x	x
5	x	x	x	x	x
6	x	x	x	x	x
7	x	x	x	x	x
8	x	x	x	x	x
9	x	x	x	x	x
10	x	x	x	x	x
11					
12					



Range("3:3")

	A	B	C	D	E
1	x	x	x	x	x
2	x	x	x	x	x
3	x	x	x	x	x
4	x	x	x	x	x
5	x	x	x	x	x
6	x	x	x	x	x
7	x	x	x	x	x
8	x	x	x	x	x
9	x	x	x	x	x
10	x	x	x	x	x
11					
12					



Range("A1").CurrentRegion

alle nichtleeren Zellen in der "Nachbarschaft" von A1

	A	B	C	D	E	F
1	x	x	x	x	x	
2	x	x	x	x	x	
3	x	x	x	x	x	
4	x	x	x	x	x	
5						
6						
7	x	x	x	x	x	
8	x	x	x	x	x	
9	x	x	x	x	x	
10	x	x	x	x	x	
11						
12						



Range("A1:A4").Offset(2,3)

	A	B	C	D	E
1	x	x	x	x	x
2	x	x	x	x	x
3	x	x	x	x	x
4	x	x	x	x	x
5	x	x	x	x	x
6	x	x	x	x	x
7	x	x	x	x	x
8	x	x	x	x	x
9	x	x	x	x	x
10	x	x	x	x	x
11					
12					

Verschiebung um 2 Zeilen und 3 Spalten



Schreiben in mehrere Zellen

`Application.ActiveWorkbook.ActiveSheet.Range("A1:B4") = 89`

`Application.ActiveWorkbook.ActiveSheet.Range("C:C") = "*"`

	A	B	C	D	E	F
1	89	89 *		1	1	
2	89	89 *		1	1	
3	89	89 *		1	1	
4	89	89 *		1	1	
5	1	1 *		1	1	
6	1	1 *		1	1	
7	1	1 *		1	1	
8	1	1 *		1	1	
9	1	1 *		1	1	
10	1	1 *		1	1	
11		*				
12		*				
13		*				
14		*				
15		*				

Range("C:C") = ganze Spalte



Schreiben in mehrere Zellen

`Application.ActiveWorkbook.ActiveSheet.Range("A1").
CurrentRegion = "x"`

	A	B	C	D	E
1	x	x	x	x	x
2	x	x	x	x	x
3	x	x	x	x	x
4	x	x	x	x	x
5	x	x	x	x	x
6	x	x	x	x	x
7	x	x	x	x	x
8	x	x	x	x	x
9	x	x	x	x	x
10	x	x	x	x	x
11					
12					

`Range("A1").CurrentRegion`
= zusammenhängend
gefüllter Bereich



Excel-Tabelle → VBA-Variablen

```
ActiveWorkbook.Worksheets("Bundeslaender").Activate
For zaehler = 1 To Range("A2").CurrentRegion.Rows.Count
    land(zaehler).name = Cells(zaehler, 1)
    land(zaehler).flaeche = Cells(zaehler, 2)
    land(zaehler).einwohner = Cells(zaehler, 3)
Next zaehler
```



Methoden eines Range-Objekts (Auswahl)

- Select – wählt den Bereich des Range-Objektes
- Copy – kopiert den Inhalt des Range-Objekts in die Zwischenablage
- Cut – entfernt das Range-Objekt aus der Tabelle und fügt es in die Zwischenablage ein
- Delete – löscht Inhalt des Range-Objekts
- PasteSpecial – Fügt den Inhalt der Zwischenablage ein
- Sort – Sortiere
z. B. `Range("A1:B7").Sort Key1:=ActiveSheet.Range("B:B")`
sortiert angegebenen Bereich nach Spalte B



Attribute eines Range-Objekts (Auswahl)

- Interior – Zellenhintergrund
Beispiel: Hintergrund rot färben
`Range("A1:B3").Interior.Color = vbRed`
oder
`Range("A1:B3").Interior.Color = RGB(255, 0, 0)`
- Borders – Zelumrandung
- Value – Zellinhalt (Wert) wie schon besprochen
- Font - Schriftart

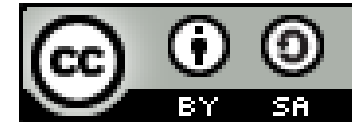


Nutzung von Excel-Funktionen

Application.WorksheetFunction.Min

Application.WorksheetFunction.Max

Application.WorksheetFunction.Average



Dieses Werk bzw. Inhalt steht unter einer "Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz".

Sie dürfen:

- das Werk vervielfältigen, verbreiten und öffentlich zugänglich machen
- Bearbeitungen des Werkes anfertigen

zu den folgenden Bedingungen:

- Namensnennung — Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.
- Weitergabe unter gleichen Bedingungen — Wenn Sie das lizenzierte Werk bzw. den lizenzierten Inhalt bearbeiten oder in anderer Weise erkennbar als Grundlage für eigenes Schaffen verwenden, dürfen Sie die daraufhin neu entstandenen Werke bzw. Inhalte nur unter Verwendung von Lizenzbedingungen weitergeben, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind.
- Weiteres dazu: <http://creativecommons.org/licenses/by-sa/3.0/de/>