



# Unterprogramme und Funktionen





# Modularisierung

Statt ein großes Programm zu schreiben, sollte das Programm in kleinere Teile aufgeteilt werden.

Vorteile:

- Mehrfach verwendete Programmabschnitte werden nur noch einmal geschrieben.
- Übersichtlichkeit, Ändern einfacher
- Arbeitsteilung möglich



## Gegenseitiger Aufruf von Programmen

```
Sub programm1()
```

```
    Debug.Print "Unterprogramm 1 wird abgearbeitet."
```

```
End Sub
```

```
Sub programm2()
```

```
    Debug.Print "Unterprogramm 2 wird abgearbeitet."
```

```
End Sub
```

```
Sub hauptprogramm()
```

```
    programm1
```

```
    call programm2
```

```
End Sub
```



genutzt von  
hauptprogramm

Name	Inhalt
a	21
y	7

genutzt von  
programm1



## Geltungsbereich von Variablen

Sub oben()

Dim x As Integer

call unten

...

End Sub

Sub unten()

Dim y As Integer

...

End Sub

Die Variable x gibt es nur im Programm *oben*, die Variable y nur im Programm *unten*.

Außerhalb des jeweiligen Programms sind diese Variablen unbekannt und können nicht benutzt werden.



## Geltungsbereich von Variablen

Sub oben()

Dim **x** As Integer

call unten

...

End Sub

Sub unten()

Dim **x** As Integer

...

End Sub

Die Variable x im Programm *oben* hat nichts mit der Variablen x im Programm *unten* zu tun!



gemeinsam von allen  
Programmen des  
Moduls genutzt

Name	Inhalt
a	21
y	7

genutzt von  
hauptprogramm

genutzt von  
programm1



## Globale Variable

Dim g As Integer

```
Sub unterprogrammA()
```

```
    g = 99
```

```
End Sub
```

```
Sub unterprogrammB()
```

```
    g = 21
```

```
End Sub
```

```
Sub demo_globale_variable()
```

```
    g = 3
```

```
    unterprogrammA
```

```
    unterprogrammB
```

```
End Sub
```

Die Variable g gilt global, d.h. in allen Programmen des Moduls.

Sie kann überall genutzt, d.h. gelesen und geändert werden.



\* \* \* \* \*  
\* GUTEN TAG! \*  
\* \* \* \* \*



## Programmaufruf mit Parametern

gerufenes Programm:

```
Sub druck(s As String)
```

```
...
```

```
End Sub
```

rufendes Programm:

```
Sub up_aufrufen()
```

```
    Dim k As String
```

```
    druck("Visual")
```

```
    k = "Basic"
```

```
    druck(k)
```

```
    druck("for" & " App.")
```

```
End Sub
```

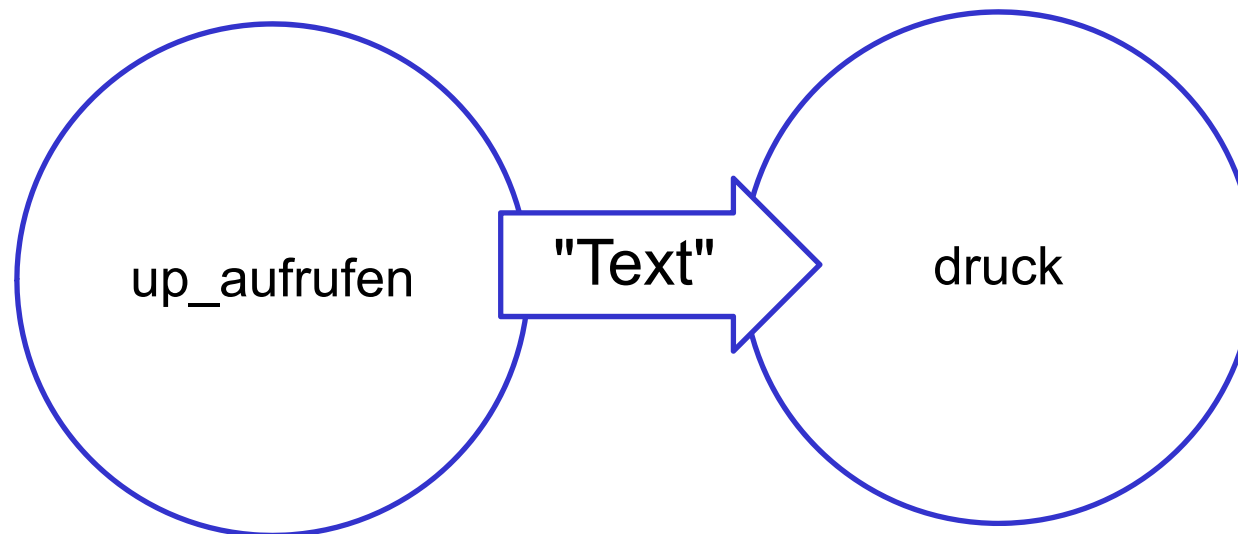
Beim Aufruf wird ein Wert von s vom rufenden Programm up\_aufrufen an das gerufene Programm ausgabe\_mit\_sternchen übergeben.

Dieser steht im gerufenen Programm in der Variablen s zur Verfügung.

Für s muss es keine Dim-Anweisung geben.



\* \* \* \* \*  
\* GUTEN TAG! \*  
\* \* \* \* \*





## Funktionen

$$f(x) = \sin x$$

Definitionsbereich:  $\mathbb{R}$  , Wertebereich:  $[-1,1]$

Len(x)

Definitionsbereich: String, Wertebereich: Integer

Chr(x)

Definitionsbereich: Integer, Wertebereich: String

Round(x,g)

Definitionsbereich: (Single, Integer)

Wertebereich: Single



## Rückgabe gerufenes Programm → rufendes Programm

Function speed(x As Double, t As Double) As Double

    speed = x / t

End Function

Aufruf:

weg = 120

zeit = 8

geschwindigkeit = speed(weg, zeit)



Welcher Datentyp wird zurückgeliefert?



## Arbeit mit dem von der Funktion gelieferten Ergebnis

geschwindigkeit = speed(weg, zeit)

MsgBox("Geschwindigkeit: " & (speed(weg,zeit)))

MsgBox("Geschwindigkeit" &  
Str(Round(speed(weg,zeit),2)))

If speed(weg,zeit) < 100 then

    MsgBox("langsam")

Else

    MsgBox("schnell")

End If



## Arbeit mit dem von der Funktion gelieferten Ergebnis

```
MsgBox("Geschwindigkeit" & Str(speed(weg,zeit) ) )
```

```
MsgBox("Geschwindigkeit" & Str(Round(speed(weg,zeit),2) ) )
```



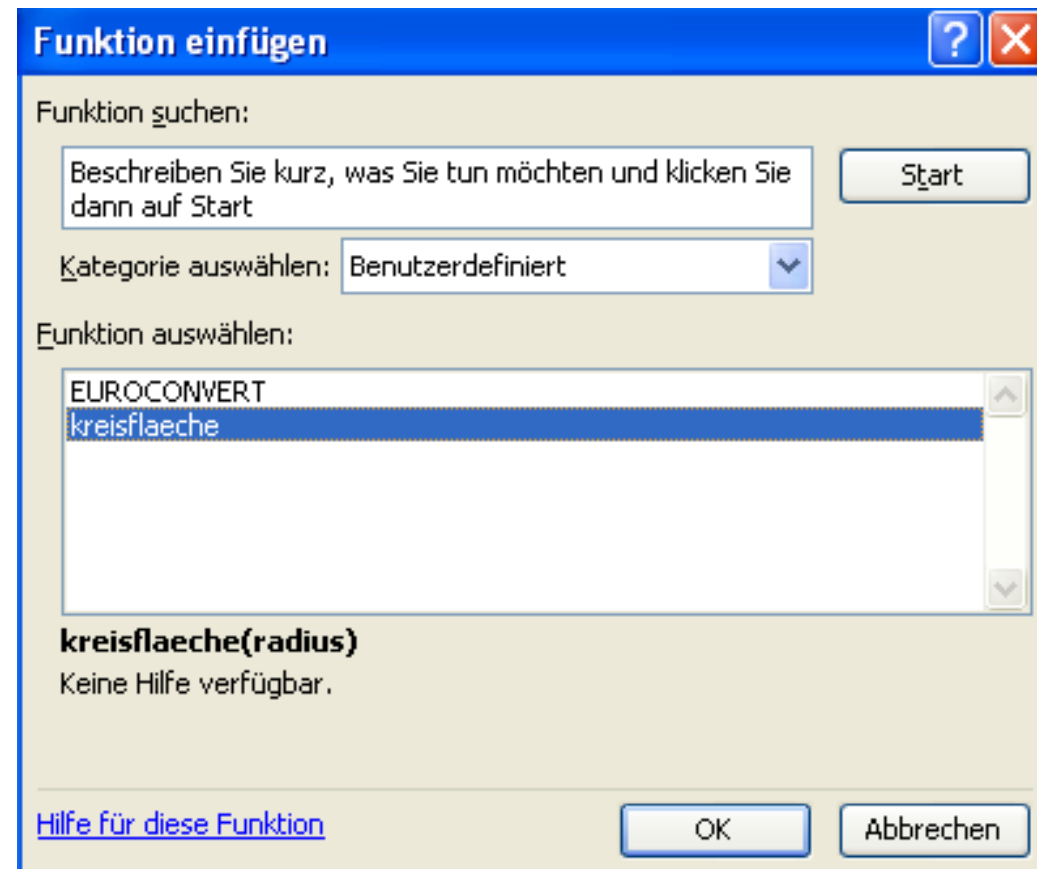
# Funktion steht in Excel zur Verfügung

Function kreisflaeche(r As Double) As Double

Const PI As Single = 3.1415926535

kreisflaeche = PI \* r \* r

End Function





## Function oder Sub?

- Function verwenden wir...
- ...wenn wir wie gerade gesehen eine neue Funktion brauchen, die in Excel-Tabellenblättern verwendet werden kann
- ... wenn ein Rückgabewert zurückzugeben ist.
- Andernfalls verwenden wir Sub.



## Struktogramm-Symbol für Unterprogrammaufruf

Unterprogramm

Unterprogramm (übergebene Werte)

Variable = Funktion (übergebene Werte)



## Modulare Programmierung

- Extrem Wichtig: Gute Dokumentation beim Funktionsaufruf:

\*\*\*\*\*

'\* Berechnung des Volumens eines Kreiskegels

'\* übergebene Parameter: r - Radius des Grundkreises

'\* h - Höhe

'\* Rückgabewert: Volumen

\*\*\*\*\*

- Funktion kann genutzt werden, ohne die interne Funktionsweise dieser Funktion zu kennen.
- Ergebnis:  
Arbeitsteilung beim Programmieren  
Wiedernutzung gut funktionierender Lösungen  
Man kann die Funktion ändern, ohne das rufende Programm zu ändern.



## Optionale Parameter

Function speed(x As Double, t As Double, **Optional unit As String**) As Double

speed = x / t

If unit = "mph" Then speed = speed \* 0.621371192

End Function

kann aufgerufen werden als:

speed (120,2) oder

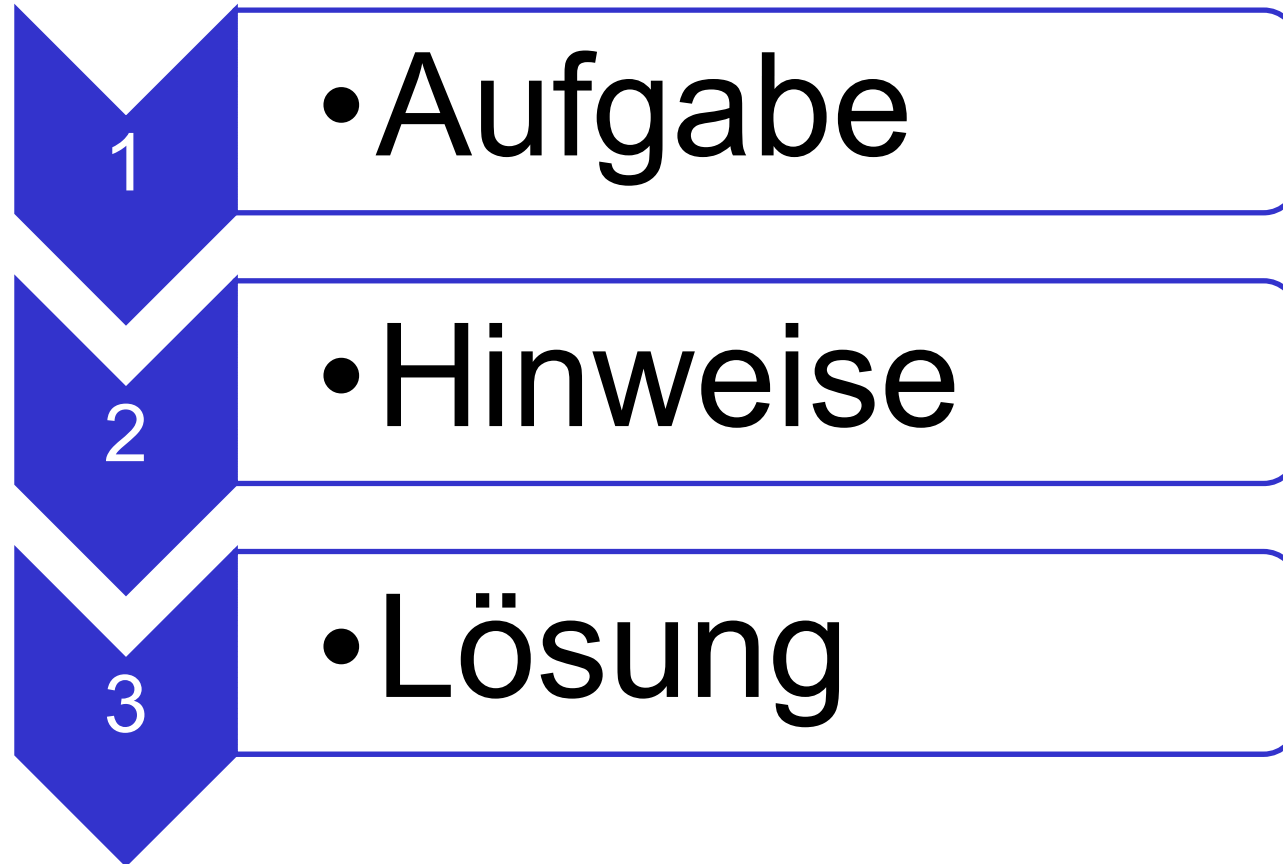
speed(120,2, "mph")



## Optionale Parameter mit Vorgabewert

bedeutet: Wenn der optionale Parameter unit nicht angegeben wurde, wird unit auf "km/h" gesetzt.

Aufruf `speed_as_text(120,8)` bewirkt dasselbe wie  
Aufruf `speed_as_text(120,8,"km/h")`





## Übung

- Schreiben Sie eine Funktion, die die Eingabe einer ganzen Zahl realisiert.
- Es ist zu prüfen, ob für den Wert dieser Zahl gilt  $\min \leq \text{Eingabe} \leq \max$  gilt.
- Solange eine Zahl außerhalb dieses Bereichs eingegeben wurde, ist die Eingabe zu wiederholen.

So könnte der Aufruf aussehen:

`x = zahleneingabe(1,10)`

Name der  
aufgerufenen  
Funktion

beim Funktionsaufruf  
übergebene Werte für  
min und max



Function zahleneingabe(min As Integer, max As Integer) As Integer

zahleneingabe = ...



## Übung

- Ändern Sie die Eingaben aus der "Türsteher-Aufgabe" aus Woche 3 so ab, dass die folgenden Zeilen nur einmal geschrieben werden müssen:

```
eingabe = InputBox("...")
```

```
If eingabe = "ja" Or eingabe = "j" Then
```

```
    bezahlt = True
```

```
Else
```

```
    bezahlt = False
```

```
End If
```

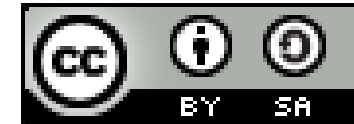


Informationen zu Urhebern und Lizenzen der verwendeten Bilder finden Sie unter den angegebenen Links:

Foto Türsteher:

Urheber: Hans-Petter Fjeld

<https://commons.wikimedia.org/wiki/File:Doorman.JPG>



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/>