

Python Syntax

Vorkurs Bauingenieurwesen - Informatik

Rechnen

Zeichen	Beschreibung	Beispiel	Ergebnis
+	Plus	4 + 5	9
-	Minus	12 - 7	5
*	Mal	3 * 6	18
/	Geteilt (exakt)	5 / 2	2.5
//	Geteilt (abgerundet)	5 // 2	2
%	Rest	5 % 2	1
**	Hoch	10 ** 3	1000

Variablen

Eine Variable ist ein Name für ein Speicherort.

```
# lege einen neuen Speicherort mit name "friedhelm" an,  
# speichere dort vorerst die Zahl 3.
```

```
friedhelm = 3
```

```
# überschreibe was an Speicherort "friedhelm" steht  
# mit dem Ergebnis der Rechnung (also der Zahl 26)
```

```
friedhelm = 4 * 5 + 6
```

```
# speichere in "friedhelm" das was  
# vorher dort gespeichert war plus 4 (also 30)
```

```
friedhelm = friedhelm + 4
```

Typen

Typ	Beschreibung	Beispiel
int	ganze Zahl	3
float	Kommazahl	3.0
bool	Wahr oder Falsch	False
str	Zeichenkette	"Hallo Freunde!"

Entscheiden

Zeichen	Beschreibung	Beispiel	Ergebnis
==	gleicher Wert	"hallo" == "hallo"	True
!=	ungleicher Wert	"hello" != "Hallo"	True
<	kleiner als	4 < 5	True
>	größer als	4 > 5	False
<=	kleiner oder gleich	7 <= 7	True
>=	größer oder gleich	4 >= 5	False
not	Gegenteil	not True	False
and	Und	True and False	False
or	Oder	True or False	True

Kontrollfluss

Bedingte Ausführung

Nur, wenn eine Bedingung erfüllt ist, werden die weiter eingerückten Anweisungen darunter ausgeführt.

```
if 3 > 4:
    print("Hilfe, die Mathematik ist kaputt!")
    print("ruft die Feuerwehr!")
print("Diesen Text sehen wir immer.")
```

Ausgabe:
Diesen Text sehen wir immer.

Wenn die Bedingung Wahr ist mache das eine, wenn nicht das andere:

```
if 3 > 4:
    print("Hilfe, die Mathematik ist kaputt!")
    print("ruft die Feuerwehr!")
else:
    print("Puuh, die Mathematik ist noch heil.")
print("Diesen Text sehen wir immer.")
```

Ausgabe:
Puuh, die Mathematik ist noch heil.
Diesen Text sehen wir immer.

Unterscheide mehrere Einzelfälle:

```
if 3 == 4:
    print("Ganze Zahlen sind kaputt!")
elif 3.0 > 4.0:
    print("Ganze Zahlen sind heil.")
    print("Kommazahlen sind kaputt!")
elif "hallo" == "tschüss":
    print("Ganze Zahlen sind heil.")
    print("Kommazahlen sind heil.")
    print("Zeichenketten sind kaputt!")
else:
    print("Entwarnung")
print("Diesen Text sehen wir immer.")
```

Ausgabe:
Entwarnung
Diesen Text sehen wir immer.

Wiederholen von Anweisungen

Will man *fast* die selben Anweisungen mehrfach ausführen, nutzt man Schleifen.

Version 1:	Version 2:	Ausgabe:
<pre>x = 2 while x < 6: print(x) x = x + 1 print("fertig!")</pre>	<pre>for x in range(2, 6): print(x) print("fertig!")</pre>	<pre>2 3 4 5 fertig!</pre>