

Kommentare

Kommentare beginnt man mit einem #. Es ist wichtig sein Programm zu kommentieren, da man sich sonst nicht auskennt!

Variablen

Zuweisung `x = 5 , y = 10`

Addition `x + y`

Subtraktion `x - y`

Multiplikation `x * y`

Division `x / y`

Modulus `x % y`

Exponentialrechnung `x ** y`

Floor Division `x // y`

- in Python muss man den Datentyp nicht angeben
- müssen mit einem Buchstaben oder `_` beginnen
- Case sensitive
- kein Zeichenabstand
- Name sollte selbsterklärend sein

Datentypen

Integer	Float	Complex	String
<code>x = 1</code>	<code>x = 1,2</code>	<code>x = 5j</code>	<code>x = "Hi"</code>
<code>x = 7</code>	<code>x = 1,6</code>	<code>x = -8j</code>	<code>x = "Auto"</code>
<code>x = -31</code>	<code>x = -6,9</code>	<code>x = 7-9j</code>	<code>x = "gut"</code>

Operatoren

Zuweisungsop.	Logische Op.	Vergleichsop.
<code>=</code>	<code>and</code>	<code>== od. !=</code>
<code>+=</code>	<code>or</code>	<code>< od. <=</code>
<code>-=</code>	<code>not</code>	<code>> od. >=</code>

Das sind nur ein paar Beispiele von Operatoren - weitere kann man auf w3school.com finden.

Funktion

```
def my_function(): # Funktion definieren
```

Eine Funktion ist ein Codeblock, der auf Abruf alle Commands in der Funktion ausführt.

Casting

in Integer	in String	in float
<code>int(7.6) # 7</code>	<code>str(7) # "7"</code>	<code>float(7) -> 7.0</code>
<code>int("3") # 3</code>	<code>str(6.9) # "6.9"</code>	<code>float("2") # 2.0</code>

Casting ist das Umwandeln von einem Datentyp in einen anderen Datentyp

If - else

```
if x > y: # falls x > y, 1 Möglichkeit
    print("x ist größer als y")
elif: x == y: # 2 Möglichkeit
    print("x ist gleich groß wie y") .
else: # alle anderen Möglichkeiten
    print("y ist größer als x")
```

Listen

```
liste = ["Apfel", "Curare", "Effekte"]
```

Listen - Commands

<code>list.append("x")</code>	x hinzufügen
<code>list.remove("y")</code>	y entfernen
<code>list.insert(0,"x")</code>	x an einer stelle einfügen
<code>del list()</code>	die liste löschen

der Index einer Liste beginnt bei 0.

For-Schleife

```
staedte = ["Wien","Graz", "Salzburg"]
for x in staedte:
    print(x)
#Ausgabe: Wien
           Graz
           Salzburg

for n in "Hey":
    print(n)
#Ausgabe: H
           e
           y
```

Mit der Vorschleife kann man für jedes Element einer Liste ein Command durchführen.

Klassen

```
class MeineKlasse:
    x = 5
```

Klasse: init-Funktion

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

p1 = Person("John", 36)
print(p1.name) # --> "John"
print(p1.age) # --> 36
```

range()

```
for x in range(6):
    print(x)

#Ausgabe: 0
1
2
3
4
5
```

For- else

```
for x in range(5):
    print(x)
else:
    print("Fertig")

#Ausgabe: 0
1
2
3
4
Fertig
```

Verschachtelte For-Schleife

```
adj = ["red", "green"]
obj = ["car", "stone"]
for x in adj:
    for y in obj:
        print(x,y)

#Ausgabe: red car
           red stone
           green car
           green stone
```

While-Schleifen

```
while x < 5: # abfrage
    print(x)
    if x == 4: # abfrage
        break # abbruch
    x = x + 1

while y < 5:
    print(x)
    if y == 3: # abfrage
        continue # neustart
    y = y + 1
```

Nützlich, falls man Loops im Programm braucht.

