

Python

Operatoren

Vergleichsoperatoren

Operator	Bedeutung	Beispiel
==	ist gleich	a == b
!=	ist ungleich	a != b
>	größer	a > b
<	kleiner	a < b
>=	größer gleich	a >= b
≤	kleiner gleich	a ≤ b

Logische Operatoren

Operator	Bedeutung	Beispiel
and	und	(a == b) and (z > y)
or	oder	(a == b) or (z > y)
not	nicht	not (a == b)

Rechenoperatoren

Operator	Zeichen	Beispiel	Ergebnis
Addition	+	3 + 4	7
Subtraktion	-	5 - 11	-6
Multiplikation	*	3 * 6	18
Division	/	9 / 2	4.5
Potenz	**	3 ** 4	81

Einseitige Auswahlstruktur

```
if bedingung:
    anweisung(en)
```

Zweiseitige Auswahlstruktur

```
if bedingung:
    anweisung(en)
else:
```

```
anweisung(en)
```

Mehrseitige Auswahlstruktur

```
if bedingung:  
    anweisung(en)  
elif bedingung:  
    anweisung(en)  
else:  
    anweisung(en)
```

Zählschleife

```
for Variable in Sequenz:  
    Anweisung1  
    Anweisung2  
    ...  
    Anweisungn  
  
#range-Funktion: range(start, stop)
```

Eingabe

```
input(prompt)  
#prompt: Zeichenkette, die als Nachricht vor der Eingabe ausgegeben wird  
  
float(value)  
#Konvertiert eine Zeichenkette in eine Gleitkommazahl  
  
int(value)  
#Konvertiert eine Zeichenkette in eine Ganzzahl
```

Ausgabe

```
print(objects)
```

Zufallszahlen

<https://docs.python.org/3/library/random.html>

Eine natürliche Zufallszahl gibt uns folgende Anweisung zurück:

```
random.randint(m, n)
```

Der Aufruf `random.randint(1, 10)` liefert also eine natürliche Zahl zwischen 1 und 10, der Aufruf `random.randint(5, 8)` eine natürliche Zahl zwischen 5 und 8.

```
#Paket importieren
import random
#Anweisung
random.randint(ersteZahl, letzteZahl)
```

Beispiel

```
import random
zahl = random.randint(1, 100)
print(zahl)

turtle.forward(random.randint(1, 10))
```

Funktionen ohne Parameter

```
def bezeichner():
    Anweisungen
```

Turtle Grafik

Bibliothek `turtle graphics` - <https://docs.python.org/3.3/library/turtle.html>

Häufige Anweisungen

```
#turtle.forward(distanz)
turtle.forward(10)

#turtle.right(winkel)
turtle.right(90)

#turtle.left(winkel)
turtle.left(180)

#turtle.back(distanz)
turtle.back(10)

#turtle.goto(x,y)
```

```
turtle.goto(100, -100)

#Ausrichtung der Turtle bestimmen (0, 90, 180 oder 270)
turtle.setheading(0)

#turtle.dot(size=None, *color)
turtle.dot(20, "blue")
```

From: <https://herr-pfeiffer.de/unterrichtswiki/> - **Unterrichtswiki - Herr Pfeiffer**

Permanent link: <https://herr-pfeiffer.de/unterrichtswiki/informatik:computerkunst:python-klassenarbeit>

Last update: **2020/06/23 09:08**

