

J1	BPE 5: Grundlagen der Programmierung Informationsmaterial	Informatik
----	---	------------

L2_3 Rechenoperatoren

Zur Durchführung von Berechnungen stehen verschiedene Rechenoperatoren zur Verfügung, die in der unteren Tabelle aufgelistet sind:

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

Berechnungen können direkt mit Zahlen durchgeführt werden (siehe Beispiel 1 im Programm), genauso gut aber auch mit Variablen (Beispiel 2) oder mit einer Mischung aus Zahlen und Variablen (Beispiel 3). Wichtig ist auch, dass die aus der Mathematik bekannten Regeln „Punkt vor Strich“ und die Klammerregeln gültig sind.

Öffnen Sie das unten abgebildete Beispielprogramm (L2_3_rechenoperatoren.py), lassen Sie es ablaufen und variieren Sie die Beispiele auf verschiedene Arten. So können Sie eine eigene Rechnung wie „was ist drei mal acht geteilt durch 4“ vom Programm berechnen lassen. Anschließend haben Sie das Wissen, die Aufgabe auf dem Aufgabenblatt „L2_3 Aufgabe_Rechenoperatoren“ zu lösen.

Beispielprogramm:

```
# Beispiel 1
ergebnis = 7 * 3
print(ergebnis)

# Beispiel 2
zahl1 = 12
zahl2 = 4
ergebnis = zahl1 - zahl2
print(ergebnis)

# Beispiel 3
wert = 580
halber_wert = wert / 2
print(halber_wert)

# Beispiel 4: 2 hoch 4 ist ...
zahl1 = 2
ergebnis = zahl1 ** 4
print(ergebnis)

# Beispiel 5
ergebnis = (3 + 4) * 7
print(ergebnis)
```

Datei: L2_3_rechenoperatoren.py

J1	BPE 5: Grundlagen der Programmierung Informationsmaterial	Informatik
----	---	------------

Struktogramm:

L2_3_Rechenoperatoren
Deklaration und Initialisierung: ergebnis als Dezimalzahl = 7 * 3
Ausgabe: ergebnis
Deklaration und Initialisierung: zahl1 als Dezimalzahl = 12
Deklaration und Initialisierung: zahl2 als Dezimalzahl = 4
Zuweisung: ergebnis = zahl1 - zahl2
Ausgabe: ergebnis
Deklaration und Initialisierung: wert als Dezimalzahl = 580
Deklaration und Initialisierung: halber_wert als Dezimalzahl = wert / 2
Ausgabe: halber_wert
Zuweisung: zahl1 = 2
Zuweisung: ergebnis = zahl1 ** 4
Ausgabe: ergebnis
Zuweisung: ergebnis = (3 + 4) * 7
Ausgabe: ergebnis

Ausgabe:

21
8
290.0
16
49

Runden von Zahlen in Python

Um Fließkommazahlen zu runden steht in Python die Funktion `round()` zur Verfügung, mit der die üblichen Rundungen durchgeführt werden können:

`round(1.3)` liefert den Wert 1
`round(1.7)` liefert den Wert 2
`round(-1.3)` liefert den Wert -1
`round(-1.7)` liefert den Wert -2
`round(5.2357, 2)` liefert den Wert 5.24

`zahl = 7.86324`
`round(zahl, 3)` liefert den Wert 7.863