

# 3. Zahlen

## 3.1 Ganze Zahlen - int

Die ganzen Zahlen haben in Python den Datentyp int (=integer = ganzzahlig)

Für ganze Zahlen gibt es keine Begrenzung, sie können beliebig groß sein.

Es gibt die bekannten Rechenoperationen wie Addition, Subtraktion, Multiplikation und Division.

```
1 2
```

2

```
1 2+3
```

5

```
1 5-13
```

-8

```
1 12*324
```

3888

```
1 6/3
```

2.0

## 3.2 Kommazahlen - float

Der Wert der Rechnung  $6/3$  ist 2.0. Es gibt folglich auch Kommawerte in python. Diese Zahlen haben den Datentyp float. Die Datentypen sind nicht begrenzt. Auch die Gleitkommazahlen kennen die bekannten Rechenoperationen wie Addition, Subtraktion, Multiplikation und Division.

```
1 8/4
```

2.0

```
1 5/4
```

1.25

```
1 7/3
```

2.333333333333335

```
1 1.2 +3.34
```

4.54

```
1 -3.45 - 7.4
```

-10.850000000000001

```
1 4.6/4.3
```

1.069767441860465

```
1 -2.3*-3.6
```

8.28

```
1 1+2.3
```

3.3

### 3.3 Komplexe Zahlen - complex

Komplexe Zahlen haben in Python den Datentyp `complex`. Sie bestehen aus einem Realteil und einem Imaginärteil, die durch ein `j` getrennt sind. Zum Beispiel:

```
1 3+4j
```

$(3+4j)$

In Python können komplexe Zahlen direkt mit den Operatoren `+`, `-`, `*` und `/` addiert, subtrahiert, multipliziert und dividiert werden.

- **Addition:**  $(a+bj) + (c+dj)$
- **Subtraktion:**  $(a+bj) - (c+dj)$
- **Multiplikation:**  $(a+bj) * (c+dj)$
- **Division:**  $(a+bj) / (c+dj)$

Komplexe Zahlen werden in der Klassenstufe 13.2 eingeführt und im Studium der Mathematik im Bereich der Funktionentheorie ausführlich untersucht.

```
1 (3+4j) + (-2-5j)
```

$(1-1j)$

```
1 (3+4j) - (-2-5j)
```

$(5+9j)$

```
1 (3+4j) * (-2-5j)
```

$(14-23j)$

```
1 (3+4j) / (-2-5j)
```

$(-0.896551724137931+0.24137931034482757j)$

### 3.4 Bits - bool

Computer kennt zwei Zustände: Strom an und Strom aus. Diese ist also die kleinste Informationseinheit, die in einem Computer gespeichert werden kann und wird als binary digit (binär Ziffer), oder kurz Bit, bezeichnet. In Python gibt es den für ein Bit auch den Datentyp `bool`, der zwei mögliche Werte hat: `True` (wahr, 1) und `False` (falsch, 0).

```
1 True
2
```

`True`

```
1 False
```

`False`

```
1 5>10
```

`False`

1 **5==5**

True

Auch auf Bits lassen sich die Grundrechenarten anwenden.

### Addition

$$\begin{array}{r} + \quad 0 \quad 1 \\ \hline \mathbf{0} \quad 0 \quad 1 \\ \mathbf{1} \quad 1 \quad 0 \text{ mit Übertrag } 1 \\ \hline \end{array}$$

### Subtraktion

$$\begin{array}{r} - \quad 0 \quad 1 \\ \hline \mathbf{0} \quad 0 \quad 1 \text{ mit Übertrag } 1 \\ \mathbf{1} \quad 1 \quad 0 \\ \hline \end{array}$$

### Multiplikation

$$\begin{array}{r} * \quad 0 \quad 1 \\ \hline \mathbf{0} \quad 0 \quad 0 \\ \mathbf{1} \quad 0 \quad 1 \\ \hline \end{array}$$

### Division

$$\begin{array}{r} / \quad 0 \quad 1 \\ \hline \mathbf{0} \quad \text{not defined} \quad 0 \\ \mathbf{1} \quad \text{not defined} \quad 1 \\ \hline \end{array}$$