

Datentypen

Numerisch

| | | |
|----------------|------------------|-----------------------|
| int | ganze Zahlen | -3214, 42 |
| float | Gleitkommazahlen | 3.141, -5.76e-2, 2E02 |
| bool | boolesche Werte | True, False |
| complex | komplexe Zahlen | 3.4 + 4e2j, 3J + 7 |

Datenstrukturen

| | | |
|-------------------|--|-------------------------------|
| string | Zeichenkette | "grün", 'Blau' |
| list | allgemeine Liste (Stack) | [value, ...] |
| tuple | unveränderbare Liste | (value, ...) ¹ |
| set | ungeordnete und nicht indizierte Liste | {value, ...} |
| dictionary | Liste mit key als Index | {key:value, ...} ² |

¹ Klammern optional

² key muss **eindeutig** sein

Typumwandlung

| | | |
|------------------|-------------------|-----------------------------------|
| int (a) | a → int | x = int(3.0) , y = int("4") |
| float (a) | a → float | x = float(2) , y = float(" - 42") |
| str (a) | a → string | x = str(2) , y = str(4.2) |
| set (a) | a → set | x = set([1, 2, 3]) , y = set() |
| list (a) | a → list | x = list('a', 'b', 'c') |

Die Funktion **input()** gibt immer einen **String** zurück.

Soll die Eingabe z.B. einem int zugewiesen werden:

```
x = int(input())
```

Arithmetische Operatoren

| | | | |
|--------|-----------------------|-------|-------------|
| x + y | Addition | x - y | Subtraktion |
| x * y | Multiplikation | x / y | Division |
| x // y | Division ohne Rest | x % y | Modulo |
| x ** y | Potenz x ^y | | |

Zuweisungskurzbehele: **x Operator= y**

Bsp: **x += 5** entspricht **x = x + 5**

Vergleichsoperatoren

| | | | |
|--------|---------|--------|---------------------|
| x < y | kleiner | x <= y | kleiner oder gleich |
| x > y | größer | x >= y | größer oder gleich |
| x == y | gleich | x != y | ungleich |

Logische Operatoren

| | | |
|---------|----------------------|-----------------------------|
| x and y | x und y wahr | (3 > 2) and (2 < 4) # False |
| x or y | x oder y wahr | (3 > 2) or (2 < 4) # True |
| not x | x nicht wahr | (3 > 2) # False |

Mitgliedsoperationen

| | | |
|------------|-----------------------------|----------------------------------|
| x in y | x Element von y | 5 in (1, 2, 3, 4, 5) # True |
| x not in y | x kein Element von y | 5 not in (1, 2, 3, 4, 5) # False |

Bedingte Ausführung

if *Bedingung:* Anweisungen werden ausgeführt, falls *Anweisung(en)* **Bedingung** wahr und if-Konstrukt wird verlassen

elif *Bedingung:* (optional, mehrere elif-Blöcke möglich) Anweisungen werden ausgeführt, falls **Bedingung** wahr und if-Konstrukt wird verlassen

else: (optional) Anweisungen werden ausgeführt, falls keine der Bedingungen wahr waren

Schleifen

while *Bedingung:* *Anweisung(en)*

for i **in** range(start, end [, step]): *Anweisung(en)*

for index **in** *Datens truktur* *Anweisung(en)*

for key **in** sorted (dictionary): *Anweisung(en)*

¹ Schleife geht von **start** bis **end-1**

break beendet die Schleife, **continue** den aktuellen Durchlauf

Indizes und Teilsequenzen (string, list, tuple)

| | | | |
|----------------------|-------------|-----------------------|-------------|
| <code>len(a)</code> | 6 | | |
| <code>a[0]</code> | 0 | <code>a[5]</code> | 5 |
| <code>a[-1]</code> | 5 | <code>a[-2]</code> | 4 |
| <code>a[1:]</code> | [1,2,3,4,5] | <code>a[:5]</code> | [0,1,2,3,4] |
| <code>a[:-2]</code> | [0,1,2,3] | <code>a[1:3]</code> | [1,2] |
| <code>a[1:-1]</code> | [1,2,3,4] | <code>b = a[:]</code> | klonen |

Indizes und Teilsequenzen für a = [0,1,2,3,4,5]

String-, List-, Tuple-Operationen

| | |
|------------------------|-----------------------|
| <code>len(s)</code> | Länge von s |
| <code>sorted(s)</code> | Sortierte Kopie von s |

String-Operationen

| | |
|------------------------------------|--------------------------------------|
| <code>str.lower()</code> | umwandeln in Kleinbuchstaben |
| <code>str.upper()</code> | umwandeln in Grossbuchstaben |
| <code>str.replace(alt, neu)</code> | <i>alt</i> durch <i>neu</i> ersetzen |
| <code>str.split()</code> | erzeugt Liste aus den Sub-Strings |

List-Operationen

| | |
|-------------------------------|--------------------------------|
| <code>lst.append(e)</code> | hängt e an |
| <code>lst.insert(i, e)</code> | fügt e for 1-tem Element ein |
| <code>lst.pop(i)</code> | entfernt i-tes Element |
| <code>lst.remove(e)</code> | entfernt 1. Element mit Wert e |
| <code>del lst[i]</code> | entfernt i-tes Element |

