

## for-Schleifen

Eine `for`-Schleife durchläuft in Python die Elemente einer Liste oder eines Tupels oder einer anderen Elementsammlung und führt für jedes Element den eingerückten Anweisungsblock aus:

```
for i in 1, 2, 5, 6.0:    # durchlaufe alle Elemente des Tupels (ohne Klammern)
    print(i)
```

```
for i in x:              # durchlaufe alle Elemente von x
    print(i)
```

Auch in `for`-Schleifen darf man `break` und `continue` verwenden!

## Zählschleifen mittels range-Objekten

Möchte man einen Programmblock öfters durchlaufen (z.B. 8 mal), so verwendet man obige `for`-Konstruktion mit einem `range`-Objekt. Dieses wird wie ein Tupel durchlaufen, ohne dass dessen Werte wirklich gespeichert werden (spart also Speicherplatz und Rechenzeit):

```
for i in range(8):      # entspricht (0, 1, 2, 3, 4, 5, 6, 7)
    print(i)
```

```
range(8)                # (0, 1, 2, 3, 4, 5, 6, 7) (beginnt bei 0, endet VOR 8)
range(2,8)              # (2, 3, 4, 5, 6, 7) (beginnt bei 2, endet VOR 8)
range(2,8,2)            # (2, 4, 6)           Inkrement ist 2
range(8,4,-1)          # (8, 7, 6, 5)       absteigender range
```

`range`-Argumente müssen vom Typ `int` sein. Folgendes geht daher nicht:

```
range(1.0, 2.0, 0.1)  # Fehler, range akzeptiert nur int
```