# Kontrollstrukturen mit Wiederholungen

## Vorbereitung (range)

In Python kann mit der Funktion *range(endwert)* eine Liste mit Zahlenwerten angelegt werden. Dabei wird eine Liste erzeugt, die bei 0 beginnt und bis zur letzten Zahl **vor** *endwert* reicht, also von 0 bis *endwert-1*. Die im folgenden Beispiel mit range(*10*) erzeugte Zahlenliste beginnt also mit 0 und geht bis 9, nämlich *endwert - 1*, d. h. 10 - 1 = 9:

|  |  |
| --- | --- |
| 1  2  3 | zahlenliste = range(10)  # Ausgabe der Liste mit ihren Zahlenwerten  print(list(zahlenliste)) |

**Ausgabe des Programms:**

|  |
| --- |
| **[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]** |

Die Ausgabe des Programms zeigt, dass die erzeugte Liste mit 0 beginnt und dann bis zur letzten Zahl vor 10, also bis 9, reicht (denn 9 = 10 – 1).

Der *range*-Operator kann in drei Ausprägungen verwendet werden:

|  |  |
| --- | --- |
| range*(endwert)* | Es werden alle Zahlen von 0 bis *endwert-1* durchlaufen |
| range*(startwert, endwert)* | Es werden alle Zahlen vom *startwert* bis zum *endwert-1* durchlaufen. |
| range*(startwert, endwert, schrittweite)* | Es werden alle Zahlen vom *startwert* bis zum *endwert-1* in einer s*chrittweite* durchlaufen. |

## Zählerschleife (for-Schleife)

In manchen Fällen muss ein Programm immer wieder die gleichen Anweisungen hintereinander ausführen bis ein bestimmtes Ergebnis eintritt. Wenn der Programmierer schon vorher weiß, wie oft ein Programm wiederholt werden muss, wird häufig die for-Schleife benutzt.

In Python besteht die for-Schleife aus folgenden Elementen:

* Dem Schlüsselwort „for“, mit dem die Schleife eingeleitet wird,
* einer Variablen (hier: betrag), die der Reihe nach alle Werte der folgenden Liste annimmt,
* einer Liste, in der die Werte stehen, die die angegebene Variable der Reihe nach annimmt. Nach der Angabe der Liste folgt ein Doppelpunkt als Zeichen dafür, dass hier der Inhalt der Schleife beginnt. In den folgenden Zeilen stehen die Anweisungen, die im Rahmen der Schleife ausgeführt werden sollen, wobei die Zeilen, die zur Schleife gehören, eingerückt sein müssen.

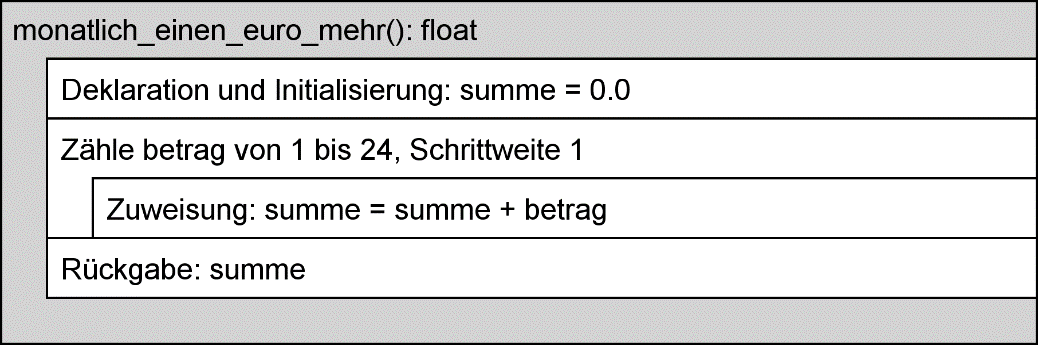
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | class Sparkonto:  def \_\_init\_\_(self, p\_inhaber, p\_kontostand, p\_zinssatz):  self.inhaber = str(p\_inhaber)  self.kontostand = float(p\_kontostand)  self.zinssatz = float(p\_zinssatz)    def monatlich\_einen\_euro\_mehr(self):  summe = 0  for betrag in range(1,25):  summe = summe + betrag  return summe  # Hauptprogramm  k1 = Sparkonto("Simon Amsel",0,0.01)  print("Sparbetrag:", k1.monatlich\_einen\_euro\_mehr(), "EUR") |

Das Programm zeigt ein Sparkonto. Die Methode *monatlich*\_einen\_euro\_mehr(self) berechnet den Sparbetrag, wenn ein Kunde 24 (=25-1) Monate immer einen Euro mehr einzahlt/spart als im Vormonat und im ersten Monat mit einem Euro startet. Die Variable betrag nimmt dabei die Werte von 1 – 24 (24 = 25(*endwert*) – 1) an.

**Ausgabe des Programms:**

|  |
| --- |
| **300 EUR** |

**Struktogramm:**



|  |  |  |
| --- | --- | --- |
| **Schleifendurchlauf** | **betrag** | **summe** |
| - | - | 0 |
| 1 | 1 | 1 |
| 2 | 2 | 3 |
| 3 | 3 | 6 |
| 4 | 4 | 10 |
| 5 | 5 | 15 |
| 6 | 6 | 21 |
| 7 | 7 | 28 |
| 8 | 8 | 36 |
| 9 | 9 | 45 |
| 10 | 10 | 55 |
| 11 | 11 | 66 |
| 12 | 12 | 78 |
| 13 | 13 | 91 |
| 14 | 14 | 105 |
| 15 | 15 | 120 |
| 16 | 16 | 136 |
| 17 | 17 | 153 |
| 18 | 18 | 171 |
| 19 | 19 | 190 |
| 20 | 20 | 210 |
| 21 | 21 | 231 |
| 22 | 22 | 253 |
| 23 | 23 | 276 |
| 24 | 24 | 300 |