

# Programmierung in Python

**Univ.-Prof. Dr. Martin Hepp**, Universität der Bundeswehr München

## Einheit 2: Kontrollflussstrukturen und Algorithmik

**Version: 2019-11-06**

# 1 Sequenz

```
In [12]: print('Erste Anweisung')  
         print('Zweite Anweisung')  
         print('Dritte Anweisung')
```

```
Erste Anweisung  
Zweite Anweisung  
Dritte Anweisung
```

# 2 Schleifen

# 2 Schleifen

## 2.1 Grundlagen: Schleifen mit `for`

```
In [3]: # Schleife für Liste mit Zahlen  
zahlenreihe = [1, 2, 3, 4]  
for zahl in zahlenreihe:  
    print(zahl)
```

```
1  
2  
3  
4
```

```
In [4]: # Schleife über Liste mit Zeichenketten
freunde = ['Peter', 'Paul', 'Mary']
for person in freunde:
    print('Hallo, ' + person + '!')
```

Hallo, Peter!

Hallo, Paul!

Hallo, Mary!

```
In [2]: # Schleife über Zeichen in Zeichenkette  
text = 'UniBwM'  
for char in text:  
    print(char)
```

```
U  
n  
i  
B  
w  
M
```

```
In [17]: # Mehrere Anweisungen innerhalb der Schleife  
for zahl in [1, 2, 3, 4]:  
    print(zahl)  
    print('Hallo')
```

```
1  
Hallo  
2  
Hallo  
3  
Hallo  
4  
Hallo
```

```
In [18]: # Was passiert hier?  
for zahl in [1, 2, 3, 4]:  
    print(zahl)  
print('Hallo')
```

1

2

3

4

Hallo



## 2.2 range als Iterable

Manchmal möchte man eine Schleife auf eine bestimmte Folge an Zahlen anwenden.

```
In [3]: for zahl in range(5):  
        print(zahl)
```

```
0  
1  
2  
3  
4
```

```
In [4]: for zahl in range(4, 10):  
        print(zahl)
```

4

5

6

7

8

9

```
In [9]: for zahl in range(0, 8, 2):  
        print(zahl)
```

0

2

4

6

```
In [11]: for zahl in range(5, 0, -1):  
         print(zahl)
```

```
5  
4  
3  
2  
1
```

## 2.3 Übung

Geben Sie alle Zweierpotenzen von  $2^0$  bis  $2^7$  aus.

## 2.3 Übung

Geben Sie alle Zweierpotenzen von  $2^0$  bis  $2^7$  aus.

```
In [14]: for exponent in range(0, 8):  
         print('2 hoch', exponent, 'ist gleich', 2**exponent)
```

```
2 hoch 0 ist gleich 1  
2 hoch 1 ist gleich 2  
2 hoch 2 ist gleich 4  
2 hoch 3 ist gleich 8  
2 hoch 4 ist gleich 16  
2 hoch 5 ist gleich 32  
2 hoch 6 ist gleich 64  
2 hoch 7 ist gleich 128
```

## 2.4 Schleifen mit `while`

```
In [19]: obere_schranke = 100
         aktueller_wert = 1
         while aktueller_wert < obere_schranke:
             print(aktueller_wert)
             aktueller_wert = aktueller_wert * 2
```

```
1
2
4
8
16
32
64
```

# 3 Verzweigungen

## 3.1 Einfache Verzweigung mit `if`

```
In [39]: wert = input('Geben Sie eine Zahl ein. ')
wert = int(wert)
if wert > 0:
    print(wert, ' ist größer als Null.')
```

```
Geben Sie eine Zahl ein.4
4 ist größer als Null.
```



## 3.2 Verzweigung mit `if` und `else`

```
In [40]: wert = input('Geben Sie eine Zahl ein. ')
wert = int(wert)
if wert > 0:
    print(wert, ' ist größer als Null.')
else:
    print(wert, ' ist kleiner oder gleich Null.')
```

```
Geben Sie eine Zahl ein. 0
0 ist kleiner oder gleich Null.
```

## 3.3 Mehrfachverzweigung mit `if`, `elif` und `else`

```
In [41]: wert = input('Geben Sie eine Zahl ein. ')
wert = int(wert)
if wert > 0:
    print(wert, ' ist größer als Null.')
elif wert == 0:
    print(wert, ' ist gleich Null.')
else:
    print(wert, ' ist kleiner als Null.')
```

```
Geben Sie eine Zahl ein. -3
-3 ist kleiner als Null.
```

## 3.4 Fallstricke bei Mehrfachverzweigungen

Die Bedingungen müssen vom spezielleren Fall zu den allgemeineren Fällen geprüft werden, da nur der erste passende Zweig ausgeführt wird.

```
In [43]: zahl = 4
         if zahl > 0:
             print('Die Zahl ist größer als Null.')
         elif zahl > 2:
             print('Die Zahl größer als 2.')
```

```
Die Zahl ist größer als Null.
```

# 4 Kombinationen

## 4.1 Schleifen in Schleifen

```
In [25]: for faktor_1 in range(1, 11):
          for faktor_2 in range(1, 11):
              # end = '\t' sorgt dafür, dass am Ende des Befehls
              # keine neue Zeile, sondern ein Tabulatorzeichen folgt.
              print(faktor_1 * faktor_2, end = '\t')
          print()
```

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80

## 4.2 Verzweigungen in Schleifen

```
In [26]: for zahl in [0, 1, 2, 3, 4, 5]:
          # zahl % 2 liefert den Divisionsrest bei einer ganzzahligen Division
          if zahl % 2 == 0:
              print(zahl, ' ist gerade')
          else:
              print(zahl, ' ist ungerade')
```

```
0  ist gerade
1  ist ungerade
2  ist gerade
3  ist ungerade
4  ist gerade
5  ist ungerade
```

## 4.3 Übung

Ein Sparkonto soll fünf Jahre lang jeden Monat mit 1 % verzinst werden. Am Ende jedes Jahres wird eine Kontoführungsgebühr von 5 EUR abgezogen.

```
In [ ]: # Hier einfügen
```

# 4.3 Übung

Ein Sparkonto soll fünf Jahre lang jeden Monat mit 1 % verzinst werden. Am Ende jedes Jahres wird eine Kontoführungsgebühr von 5 EUR abgezogen.

```
In [ ]: # Hier einfügen
```

```
In [38]: anzahl_jahre = 5
zinssatz = 0.01 # pro Monat
guthaben = 1200.0
for jahr in range(anzahl_jahre):
    print(jahr, end = ':\t')
    for monat in range(12):
        guthaben = guthaben * (1 + zinssatz)
        print(round(guthaben, 2), end = '\t')
    guthaben = guthaben - 5
    print(jahr, round(guthaben, 2))
```

## Hinweise:

- Die Rundung erfolgt hier nur bei der Ausgabe!
- Für die Rundung bei der Ausgabe gibt es spezielle Mechanismen, die hier aber nicht behandelt werden.
- Man muss festlegen, ob die Gebühr nach oder vor der Zinsgutschrift für den letzten Monat eines Jahres erfolgen soll



# Vielen Dank!

<http://www.ebusiness-unibw.org/wiki/Teaching/PIP>