

## 7 Kontrollstrukturen

### 7.1 Anweisungen und Folgen

- Was sind Anweisungen?
  - Ein Programm setzt sich aus einer Menge von Anweisungen zusammen, um eine bestimmte Aufgabe zu lösen
  - Anweisungen arbeiten mit den Variablen und führen damit Operationen aus
  - Eine Anweisung ist die kleinste ausführbare Einheit eines Programms
  - Ist eine Anweisungszeile so lang, dass sie nicht in die Zeile passt, wird ein \ eingefügt und in der nächsten Zeile fortgesetzt
- Beispiel:

```
langer_text = "Python unterstützt mehrere Programmierparadigmen," \
              " z. B. "die objektorientierte und die funktionale " \
              " Programmierung."

langer_name_mit_vielen_buchstaben = \
123456789012345678
```
- Strukturierung von Anweisungen zu Anweisungsblöcken werden in Python durch Einrückung vorgenommen.

## 7.2 Bedingungen und Kontrollstrukturen

- Was ist eine Bedingung?
  - Durch eine Bedingung wird der Ablauf eines Programms beeinflusst
  - Ein logischer Ausdruck wird ausgewertet
  - Für die Formulierung solcher Bedingungen stehen Vergleichsoperatoren zur Verfügung

Operator	Bedeutung für den Wertevergleich
>	größer als if x > 5:
>=	größer gleich if x >= 5:
<	kleiner als if x < 5:
<=	kleiner gleich if x <= 5:
==	gleich if x == 5:
!=	ungleich if x != 5:

- Eine Bedingung kann entweder mit „ja“ (True) beantwortet werden oder mit „nein“ (False)
- In Abhängigkeit vom Ergebnis der Bedingungen werden Anweisungen ausgeführt, mehrmals ausgeführt oder gar nicht ausgeführt
- Kontrollstrukturen verwenden Bedingungen, um diese Entscheidungen treffen zu können

Verzweigung (Selektion)	Es werden alternative Anweisungen oder Programmteile angeboten, in die – abhängig von einer Bedingung – beim Programmablauf verzweigt wird
Schleife (Iteration)	Anweisungen oder Programmteile können – abhängig von einer Bedingung – mehrmals durchlaufen werden oder gar nicht

### 7.3 Grundlagen zu Verzweigungen

- Bedingte Anweisung:
  - Je nachdem, ob eine Bedingung erfüllt ist oder nicht, wird ein Programmteil übersprungen oder nicht
  - wird Python mit `if bedingung` -Anweisungen formuliert
- Verzweigung:
  - in Abhängigkeit vom Ergebnis einer Bedingung wird ein Programmteil ausgeführt oder nicht
  - wird Python mit `if bedingung ... else` -Anweisungen formuliert
- Geschachtelte Verzweigung:
  - Innerhalb der alternativen Programmteile einer bedingten Anweisung oder Verzweigung wird eine weitere Verzweigung integriert
  - so lassen sich mehrere Alternativen bereitstellen
  - wird Python mit verschachtelten `if bedingung ... else` -Anweisungen bzw. `if bedingung ... elif ... else` -Anweisungen formuliert
- Mehrfache Verzweigung:
  - Es wird ein Ausdruck ausgewertet – je nachdem, welchem Wert der Ausdruck entspricht, verzweigt das Programm fallweise (case)
  - so lassen sich ebenfalls mehrere Alternativen bereitstellen

## 7.4 Bedingte Anweisung

- Beispiel:  
Wenn ein Kunde einen Auftrag mit mehr als 1000 EUR erteilt, bekommt er 3% Rabatt, bei Aufträgen bis 1000 EUR wird kein Rabatt gewährt

```
preis = 1497.56
if preis > 1000.00:
    preis = preis * (1 - 0.03)
print("Berechneter Preis: ", preis)
```

## 7.5 Verzweigung

- Beispiel:  
Bei einem Kaufpreis von mehr als 1000 EUR wird ein Rabatt von 5% gewährt, ansonsten beträgt der Rabatt 3%

```
preis = 1497.56
if preis > 1000.00:
    preis = preis * (1 - 0.05)
else:
    preis = preis - (1 - 0.03)
print("Berechneter Preis: ", preis)
```

oder

```
preis = 1497.56
if preis <= 1000.00:
    preis = preis * (1 - 0.03)
else:
    preis = preis - (1 - 0.05)
print("Berechneter Preis: ", preis)
```

## 7.6 Geschachtelte Verzweigung

- Beispiel:  
Bei einem Kaufpreis von bis zu 1000 EUR wird ein Rabatt von 3% gewährt. Ist der Preis höher als 1000 EUR, beträgt der 5% Rabatt. Bei mehr als 5000 EUR wird ein Preisnachlass von 8% gewährt.

```
preis = 1497.56
if preis <= 1000.00:
    preis = preis * (1 - 0.03)
else:
    if preis > 5000.00:
        preis = preis - (1 - 0.08)
    else:
        preis = preis - (1 - 0.05)
print("Berechneter Preis: ", preis)
```

oder

```
preis = 1497.56
if preis <= 1000.00:
    preis = preis * (1 - 0.03)
elif preis > 5000.00:
    preis = preis - (1 - 0.08)
else:
    preis = preis - (1 - 0.05)
print("Berechneter Preis: ", preis)
```

## 7.7 Mehrfache Verzweigung (Fallauswahl)

- Seit Python 3.10 Mehrfachanweisung mit `match ... case ...` -Anweisung möglich
- bei früheren Versionen sind verschachtelte `if`-Anweisungen erforderlich
- Beispiel:  
Schulnote mit dem entsprechenden Text ausgeben

```
note = 2
text = ""
match note:
    case 1:
        text = "sehr gut"
    case 2:
        text = "gut"
    case 3:
        text = "befriedigend"
    case 4:
        text = "ausreichend"
    case 5:
        text = "mangelhaft"
    case 6:
        text = "ungenügend"
    case _:
        text = "FEHLER"
print (note, text)
```

## Aufgabe

- Es ist ein Programm zur Kapitalberechnung zu erstellen.
  - Über die Konsole werden das Anfangskapital und der Zinssatz zur Eingabe angefordert.
  - Es ist das Endkapital nach einem und nach 2 Jahr zu berechnen mit der Formel:  
$$K_1 = K_0 * ((p/100) + 1)$$

## 7.8 Schleifen (Iterationen)

- Eine Schleife besteht aus einer Schleifensteuerung und einem Schleifenrumpf
- Die Schleifensteuerung gibt an, wie oft oder unter welcher Bedingung die Anweisungen abgearbeitet werden
- Innerhalb der Schleifensteuerung befindet sich der Schleifenrumpf, der die zu wiederholenden Anweisungen enthält
- Es drei Arten von Schleifenstrukturen:
  - Zählergesteuerte Schleife
  - Kopfgesteuerte bedingte Schleife
  - Fußgesteuerte bedingte Schleife

### 7.8.1 Zählergesteuerte Schleife (Iteration)

- Die Schleife wird über einen Zähler gesteuert
  - Ein Zähler mit einem Startwert wird für jeden Schleifendurchlauf um einen bestimmten Wert erhöht
  - Vor der Ausführung der Anweisungen im Schleifenrumpf wird überprüft, ob der festgelegte Endwert erreicht ist
  - Ist der Endwert erreicht, wird das Programm hinter der Schleife fortgesetzt
  - Die Schleife läuft solange der Zähler kleiner als der Endwert ist
  - Die Schrittweite gibt die Erhöhung des Zählers nach jedem Schleifendurchlauf an
- 
- Beispiel:  
Ausgabe der Zahlen 10, 30, 50, 70, 90

```
for i in range(10, 101, 20):  
    print(i)
```



### 7.8.2 Kopfgesteuerte bedingte Schleife

- Solange eine Bedingung erfüllt ist, werden die in der Schleife eingeschlossenen Anweisungen ausgeführt
- Die Überprüfung der Bedingung erfolgt vor der Ausführung der Anweisungen im Schleifenrumpf
- das Programm wird hinter der Schleife fortgesetzt, sobald die Bedingung nicht mehr erfüllt ist
- Beispiel:  
Die Summe der Zahlen von 1 bis 20 soll berechnet werden

```
summe = 0
i = 1
while i <= 20:
    summe = summe + i
    i = i + 1
print (summe)
```

### 7.8.3 Fußgesteuerte bedingte Schleife

- Die in der Schleife eingeschlossenen Anweisungen werden ausgeführt
- Anschließend erfolgt die Überprüfung der Bedingung
- Ist die Bedingung erfüllt, werden Anweisungen im Schleifenrumpf erneut ausgeführt
- Erst wenn die Bedingung nicht mehr erfüllt ist, wird das Programm hinter der Schleife fortgesetzt
- wird in Python mit while, if und break abgebildet
- Beispiel:  
„Hello World“ soll ausgegeben werden, bis diese Zeichenkette 10-mal in der Konsole steht

```
i = 0
while True:
    print("Hello World")
    i+=1
    if i == 10:
        break
```

## Aufgabe

- Berechne die Fakultät einer Zahl, die über die Konsole eingelesen wird.
- Fakultät, in der Mathematik geschrieben mit ! hinter der Zahl, ist definiert als
  - $0! = 1$
  - $1! = 1$
  - $2! = 2 * 1 = 2$
  - $3! = 3 * 2 * 1 = 6$
  - $n! = 1 * 2 * 3 * \dots * (n-1) * n$

## Aufgabe

- Für eine Nachricht soll von einem Programm automatisch die Anrede formuliert werden.
- Von der Konsole sind Eingaben für Name, Geschlecht und die Uhrzeit als Stundenangabe einzulesen
- Die Anrede soll je nach Tageszeit mit „Guten Morgen“ (0–9 Uhr), „Guten Tag“ (10–17), „Guten Abend (18–0 Uhr) beginnen und anschließend mit „Herr xxx“ bzw. „Frau xxx“ fortgesetzt werden. Für xxx soll der entsprechende Name eingesetzt werden