

Aufgabenblatt 0

Einführung in die Bildverarbeitung

Christian Wilms und Simone Frintrop
SoSe 2020

Ausgabe: 20. April 2020 - **Abgabe bis:** muss nicht abgegeben werden

abgegeben am 5. Mai 2020 von:

Abdulssatar Khateb (6976879), Merle Hoffmann (7031673), Felix Swimmer (7162123)

Aufgabe 1 — Python Tutorial

1. Was bedeuten die drei Punkte (...) im interaktiven Terminal?

Die drei Punkte sind die Darstellung der sekundären Eingabeforderung und werden bei Fortsetzungszeilen angezeigt.

2. Welches Zeichen entspricht dem Modulo-Operator?

Mit % lässt sich der Rest berechnen. Beispiel:

```
>>> 17 % 3
2
```

3. Wie kann man den letzten Wert einer Liste austauschen?

Mit negativen Indizes lässt sich auf die Elemente einer Liste von hinten beginnend zugreifen. -1 ist beispielsweise der Index, des letzten Elements.

```
>>> liste = ['wert1', 'wert2', 'wert3', 'wert4']
>>> liste[-1] = 'neuerWert'           #tauscht letzten Wert aus
>>> liste
['wert1', 'wert2', 'wert3', 'neuerWert']
```

4. Warum wird die Funktion range() gerne im Zusammenspiel mit Listen verwendet?

Mit range() lassen sich Listen ganzer Zahlen mit einer speziellen Start- und Endzahl und die Schrittzahl erstellen. In Kombination mit for-Schleifen ist dies sehr praktisch, wenn man über eine bestimmte Folge iterieren will.

5. Wie kann man in einer for-Schleife gleichzeitig über zwei Listen iterieren? Müssen die Listen dafür gleich lang sein?

Mit zip() lässt sich über zwei Listen gleichzeitig iterieren. Die Listen dürfen dabei unterschiedlich lang sein, allerdings stoppt der Iterator, wenn alle Elemente der kürzeren Liste abgearbeitet sind. Die übrigen Elemente der längeren Liste werden daher ignoriert. Beispiel:

```
>>> liste1 = [1, 2, 3]
>>> liste2 = ['a', 'b', 'c', 'd']
>>> for i, a in zip(liste1, liste2):
...     print(i, a)
1 a
2 b
3 c
```

Aufgabe 2 — Direkteingabe in Spyder

1. Erzeugt zwei Listen: eine Liste mit fünf Nullen und eine Liste mit sechs Einsen.

```
>>> a = [0,0,0,0,0]
>>> b = [1]*6
```

2. Verbindet beide Listen zu einer.

```
>>> c = a + b
```

3. Erzeugt eine weitere Liste mit den Zahlen von 1 bis 10 und verbindet beide Listen.

```
>>> d = list(range(1,11))
>>> c += d          # oder: c = c + d oder: c.extend(d)
```

4. Filtert die Duplikate heraus. Tipp: Welche Datenstruktur erlaubt keine Duplikate?

```
>>> e = list(set(e))
```

5. Entfernt nun die Zahl 2 aus der Liste.

```
>>> e.remove(2)
```

6. Fügt den String "2" (Typ `str`) und einen Wert vom Typ `bool` in die Liste ein. Ist dies möglich?

Ja, denn in Python darf eine Liste Elemente verschiedener Typen enthalten. Einen Wert vom Typ `bool` hinzufügen ist möglich, da eine Liste doppelte Elemente haben darf. Hätte man bei Punkt 4. die Menge nicht wieder in eine Liste umgewandelt, wäre dies nicht möglich, da eine Menge keine doppelten Elemente besitzen darf. 0 und 1 wäre schon enthalten und es gilt `True == 1` und `False == 0`.

```
>>> e.append(str(2))
>>> e.append(True)
```

7. Ist die Zahl 2 nun enthalten?

Nein, die Zahl 2 ist nicht enthalten, aber der String 2.

```
>>> e
[0, 1, 3, 4, 5, 6, 7, 8, 9, 10, '2', True]
```

Aufgabe 3 — Listen drehen

Siehe *blatt00_3.py*

Aufgabe 4 — Scrabble

Siehe *blatt00_4.py*

Aufgabe 5 — 5er-Listen

Siehe *blatt00_5.py*