

# Übung 5: Listenkomprehension, Funktionsdefinitionen und Häufigkeitsverteilung

Bill Bosshard  
12-933-255

Lukas Vollenweider  
13-751-888

## 1 Mengen vs. Listen

Definition von set aus der Python 2 - Documentation:

```
class set([iterable])
```

- a) Da man bei python durch einen String iterieren kann (durch die einzelnen Chars), ist String ein iterable. Da Set das Eingabeobjekt in die einzelnen Elemente zerlegt, durch die man iterieren kann, ist der Rückgabewert von `set('alphabet')` ein Set, welches die einzelnen Chars enthält, während bei `set(['alphabet'])` ein Set zurückgibt, welches die einzelnen Listen enthält (hier gibt's nur eine, nämlich `['alphabet']`.)
- b) Set gibt immer eine Menge zurück. Das bedeutet, dass nicht zwischen mehrfach vorkommenden Elementen unterschieden werden und diese jeweils immer nur ein Mal in dem Set gespeichert werden. Bei Listen werden nicht die einzelnen Elemente unterschieden, sondern die Positionen, welche die Elemente einnehmen. Deshalb können Listen das gleiche Elemente mehrfach enthalten. Das ist auch der Grund, weshalb Listen sortiert sind und Sets nicht.

## 2 Rückgabewerte von Funktionen

a)

---

```
def wc1(textfile):  
    c = 0  
    for line in textfile:  
        for word in line.split():  
            c += 1  
  
    return c
```

---

- Wann stoppt die Funktion `wc1()`?  
`wc1()` stoppt, wenn alle Zeilen des Textfiles gelesen wurden, da `return c` auf gleicher Stufe ist wie `for line in textfile:`.
- Was berechnet sie?  
`c` berechnet die Anzahl Wörter in `textfile`.

b)

---

```
def wc2(textfile):  
    c = 0  
    for line in textfile:  
        for word in line.split():  
            c += 1  
  
    return c
```

---

- Wann stoppt die Funktion `wc2()`?  
`wc2()` stoppt, nachdem eine Zeile gelesen wurde, da `return c` auf gleicher Stufe ist wie `for word in line.split():`.
- Was berechnet sie?  
`c` berechnet die Anzahl Wörter der ersten Linie in `textfile`.

c)

---

```
def wc3(textfile):  
    c = 0  
    for line in textfile:  
        for word in line.split():  
            c += 1  
    return c
```

---

- Wann stoppt die Funktion `wc2()`?  
`wc2()` stoppt, nachdem das erste Wort gelesen wurde, da `return c` auf gleicher Stufe ist wie `c += 1`.
- Was berechnet sie?  
`c` berechnet das erste Wort in `textfile`, also 1.

## 3 Listenkomprehension

### 3.1 Listenkomprehension deuten

```
sorted([word.lower() for word in set(text6) if len(word) >= 4 and word[-3:] == 'ing'])
```

- Gibt eine sortierte Liste zurück, welche alle Wörter von `text6` enthält, die mindestens 4 Buchstaben enthalten und auf 'ing' enden.
- Die Liste ist nicht mehr sortiert, enthält aber noch die gleichen Elemente.
- Siehe `A3_1c.py`

## 4 Reichweite von Variablen

`space1()` definiert eine eigene, lokale Version der Variable `word`. `space1()` speichert dann das 12. Wort von `text6` in `word` ab und gibt dieses zurück. Der Wert von der globalen Variable `word` wird in dieser Methode nie beachtet und deshalb auch nicht verändert oder zurückgegeben. Deshalb gibt `space2()` den Wert der globalen Variable `word` zurück.

## Reflexion/Feedback

**a) Fasse deine Erkenntnisse und Lernfortschritte in zwei Sätzen zusammen.**

Wir haben die Listenkompensation und nltk kennengelernt und können diese nun anwenden. Ausserdem haben wir gelernt, wie man Funktionen in Python richtig kommentiert und werden dies von nun an anwenden.

**b) Wie viel Zeit hast du in diese Übungen investiert?**

3 Stunden