

# Merkblatt - Listen in Python

May 7, 2020

## 0.1 Listen

Listen ermöglichen es dir, direkt mehrere Einträge auf einmal zu speichern. Hier z. B. erfassen wir eine Liste von 4 Studierenden:

```
[ ]: students = ["Max", "Monika", "Erik", "Franziska"]

last_student = students.pop()
print(last_student)
print(students)
```

Über den + - Operator kannst du 2 Listen miteinander verknüpfen!

```
[1]: students = ["Max", "Monika", "Erik", "Franziska"] + ["ABCDEF"]
print(students)
```

```
['Max', 'Monika', 'Erik', 'Franziska', 'ABCDEF']
```

Der `del` - Befehl entfernt einen Eintrag aus einer Liste. Hierbei wird ein Element mit einem bestimmten Index (hier: 4. Element, weil das 4. Element wird über `[3]` angesprochen, weil ja bei 0 angefangen wird zu zählen)

```
[3]: students = ["Max", "Monika", "Erik", "Franziska", "ABCDEF"]
del students[3]
print(students)
```

```
['Max', 'Monika', 'Erik', 'ABCDEF']
```

Der `.remove()` - Befehl entfernt einen Eintrag nach Wert. Sprich, hier wird dann der Eintrag "Monika" aus der Liste entfernt.

```
[12]: students = ["Max", "Monika", "Erik", "Franziska", "ABCDEF"]
students.remove("Monika")
print(students)
```

## 0.2 List Comprehensions

Mit Hilfe von List Comprehensions kannst du recht einfach eine Liste in eine andere Liste umwandeln:

```
[2]: xs = [1, 2, 3, 4, 5, 6, 7, 8]
```

```
ys = [x * x for x in xs]
# ys = []
# for x in xs:
#     ys.append(x * x)

print(xs)
print(ys)
```

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

```
[1, 4, 9, 16, 25, 36, 49, 64]
```

### 0.2.1 List Comprehensions können aber noch viel mehr!

```
[4]: students = ["Max", "Monika", "Erik", "Franziska"]
```

```
lengths = [len(student) for student in students]

#lengths = []
#for student in students:
#    lengths.append(len(student))

print(lengths)
```

```
[3, 6, 4, 9]
```

### 0.2.2 Praktisch auch für's Zeichnen von Grafiken!

```
[2]: %matplotlib inline
import matplotlib.pyplot as plt
```

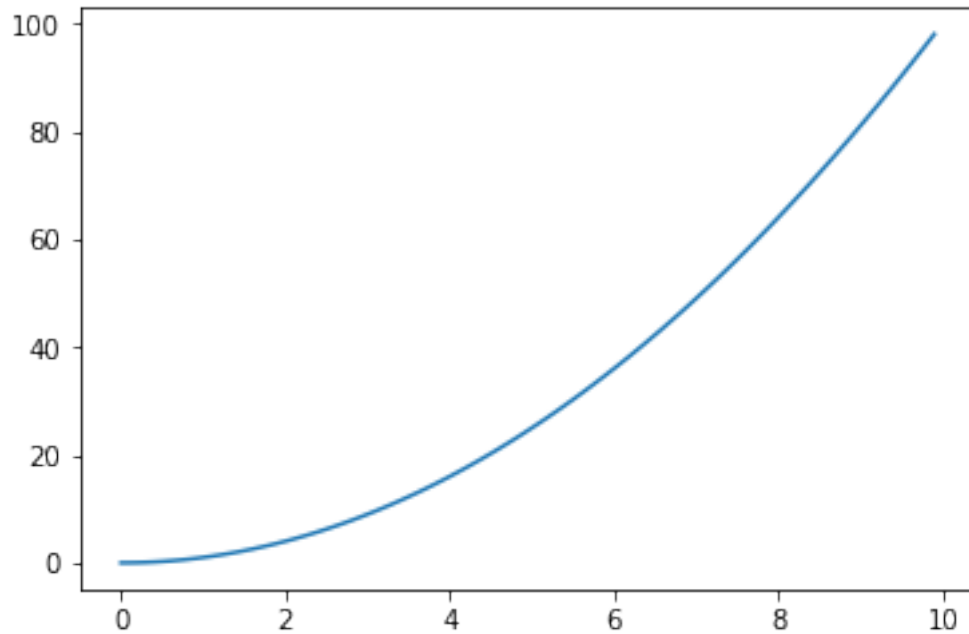
```
[4]: xs = [x / 10 for x in range(0, 100)]
ys = [x * x for x in xs]

print(len(xs))
print(len(ys))

plt.plot(xs, ys)
plt.show()
```

```
100
```

```
100
```



### 0.3 Listen verschachteln

In Python ist es erlaubt, Listen ineinander zu verschachteln. Das erlaubt uns z. B. eine Matrix zu modellieren:

```
[1]: liste = [  
    ["Berlin", "München", "Köln"],  
    ["Budapest", "Pécs", "Sopron"]  
]
```

```
[5]: liste[0][0]
```

```
[5]: 'Berlin'
```

```
[ ]:
```