

# Programmieren mit Python

## Teil 7: Comprehensions

---

Dr. Aaron Kunert

*[aaron.kunert@salemkolleg.de](mailto:aaron.kunert@salemkolleg.de)*

16. November 2022

# Comprehensions

---

## **Typische Manipulationen**

Sehr häufig möchte man eine Datenstruktur (d.h. eine Liste oder ein Dictionary) basierend auf den Werten manipulieren. Dabei werden vor allem zwei Aspekte immer wieder gebraucht: Transformationen und Filter.

## Transformation

Ersetzt man jedes Element einer Liste durch ein aus dem ursprünglich berechnetem Element, so spricht man von einer *Transformation* (engl. map/mapping).

### Beispiel

Gegeben ist die Liste `my_list = [2, 5, 3, 12, 7]`. Die Liste soll so manipuliert werden, dass alle Einträge durch ihren doppelten Wert ersetzt werden.

## Traditionelle Lösung

---

```
my_list = [2, 5, 3, 12, 7]
result = []
for k in my_list:
    result.append(2 * k)
print(result)
```

---

## The Pythonian Way

---

```
my_list = [2, 5, 3, 12, 7]
result = [2 * k for k in my_list]
print(result)
```

---

## Filter

Streicht man Elemente entsprechend ihres Wertes aus einer Liste, so spricht man von einem *Filter*.

## Beispiel

Gegeben ist die Liste `my_list = [2, 5, 3, 12, 7]`. Aus der Liste sollen alle ungeraden Einträge gestrichen werden.

## Traditionelle Lösung

---

```
my_list = [2, 5, 3, 12, 7]
result = []
for k in my_list:
    if k % 2 == 0:
        result.append(k)
print(result)
```

---

## The Pythonian Way

---

```
my_list = [2, 5, 3, 12, 7]
result = [k for k in my_list if k % 2 == 0]
print(result)
```

---

## Kombination aus Transformation und Filter

Selbstverständlich können Transformationen und Filter auch kombiniert werden.

### Beispiel

Lösche alle ungeraden Zahlen und verdopple dann alle Zahlen:

---

```
my_list = [2, 5, 3, 12, 7]
result = [2 * k for k in my_list if k % 2 == 0]
print(result)
```

---



## Struktur List Comprehension

```
result = [ Wunschetrug(k) for k in my_list if Bedingung(k) ]
```

## Dictionary Comprehension

Man kann das gleiche Verfahren auch auf Dictionaries anwenden. Dabei können jeweils key und value für die Transformationen und Filter verwendet werden.

### Beispiel

---

```
my_dict = {"a": 2, "b": 3}
result = {key: value for (key, value) in my_dict.items()}
print(result)
```

---

## Struktur Dictionary Comprehension

```
result = { Wunsch-schlüssel(k,v) : Wunsch-Wert(k,v)  
          for (k,v) in my_dict.items() if Bedingung(k,v) }
```

## List Comprehension

Gegeben sei eine beliebige Liste von ganzen Zahlen. Streiche alle Zahlen, die ungerade oder negativ sind. Ersetze die übrigen Zahlen durch ihre Hälfte.

## Lösung

---

```
my_list = [2, -3, 40, 15, 7, 8]
result = [k//2 for k in my_list if k % 2 == 0 and k >= 0]
```

---

## Dictionary Comprehension

Gegeben sei das Dictionary `{"Mathe": 9, "Sport": 13, "Physik": 4, "Bio": 12}`.  
Lösche nun daraus alle Noten unter 5 Punkte sowie die Sportnote. Zusätzlich soll das Dictionary danach wie folgt aussehen: `{"In Mathe": "9 Punkte", "In Bio": "12 Punkte"}`.

## Lösung

---

```
grades = {"Mathe": 9, "Sport": 13, "Physik": 4, "Bio": 12}
result = {
    "In " + key: f"{value} Punkte"
    for (key, value) in grades.items()
    if value >= 5 and key != "Sport"
}
```

---