

# Merkblatt - Dictionaries

September 22, 2017

## 0.1 Dictionaries in Python

Dictionaries sind unglaublich praktisch, damit kannst du z.B. folgendes machen:

- Du kannst Wertezuordnungen speichern (z.B. Telefonbuch: Ein Nachname hat eine Telefonnummer)
- Du kannst nachträglich Elemente verändern / entfernen / hinzufügen
- Dictionaries brauchst du wirklich immer wieder...

Machen wir mal ein Beispiel...

```
In [3]: d = {"Berlin": "BER", "Helsinki": "HEL", "Saigon": "SGN"}
```

```
In [4]: print(d)
```

```
{'Berlin': 'BER', 'Helsinki': 'HEL', 'Saigon': 'SGN'}
```

Zugriff auf ein einzelnes Element:

```
In [6]: print(d["Helsinki"])
```

```
HEL
```

Hiermit überschreibst du ein einzelnes Element:

```
In [7]: d["Budapest"] = "BUD"
```

```
In [8]: print(d)
```

```
{'Berlin': 'BER', 'Helsinki': 'HEL', 'Saigon': 'SGN', 'Budapest': 'BUD'}
```

## 0.2 Element entfernen

```
In [9]: del d["Budapest"]
```

```
In [10]: print(d)
```

```
{'Berlin': 'BER', 'Helsinki': 'HEL', 'Saigon': 'SGN'}
```

### 0.3 Abfrage: Ist ein Element im Dictionary?

```
In [11]: if "Budapest" in d:
          print("Budapest ist im Dictionary enthalten")
          if "Saigon" in d:
              print("Saigon ist im Dictionary enthalten")
```

Saigon ist im Dictionary enthalten

### 0.4 Auf Elemente zugreifen...

```
In [16]: print(d["Saigon"])
          print(d.get("Saigon"))
```

SGN  
SGN

```
In [18]: print(d["Budapest"])
```

```
-----
KeyError                                Traceback (most recent call last)

<ipython-input-18-9a8194eb0769> in <module>()
----> 1 print(d["Budapest"])

KeyError: 'Budapest'
```

```
In [19]: print(d.get("Budapest"))
```

None

### 0.5 Dictionaries und Schleifen

Du hast bei Dictionaries 2 Möglichkeiten, diese mit einer for - Schleife durchzugehen.  
Entweder direkt, dann gehst du die Schlüssel durch:

```
In [5]: d = {"München": "MUC", "Budapest": "BUD", "Helsinki": "HEL"}

          for key in d:
              value = d[key]
              print(key)
              print(value)
```

München  
MUC  
Budapest  
BUD  
Helsinki  
HEL

Oder über die `.items()` - Methode, damit kannst du Schlüssel + Wert direkt durchgehen.

```
In [8]: for key, value in d.items():  
        print(key + ": " + value)
```

München: MUC  
Budapest: BUD  
Helsinki: HEL

```
In [ ]:
```

### 1.5.1 collections

Stellt u.a. die Struktur defaultdict bereit, mit der man automatisch dictionaries generieren kann.

#### Modul einbinden

```
In [26]: from collections import defaultdict
```

#### Anwendung

```
In [27]: p = defaultdict(int)
         words = ["Hallo", "Hier", "Hallo", "Welt", "Welt", "Welt"]

         for word in words:
             p[word] = p[word] + 3

         print(p)
```

```
defaultdict(<class 'int'>, {'Hallo': 6, 'Hier': 3, 'Welt': 9})
```

**Weitere Infos:** <https://docs.python.org/2/library/collections.html#collections.defaultdict>