# LU11a - Dictionary

Ein Dictionary ist eine Datenstruktur in Python. Die Einträge im Dictionary bestehen aus einem Schlüssel (Key) und einem Wert (Value).

## Einleitung

Dictionaries in Python sind ähnliche Datenstrukturen wie Listen. Der wichtigste Unterschied ist die Art, wie die Elemente in einer Liste bzw. einem Dictionary identifiziert wird.

- Bei einer Liste werden die einzelnen Elemente über eine fortlaufende Nummer (Index, Position) identifiziert.
- Im Dictionary werden die Elemente über einen von uns definierten Schlüssel identifiziert.

### Listen

Listen sind am besten geeignet, wenn wir viele gleiche Informationen wie z.B. Obstsorten speichern wollen.

| Index | Wert |
|---|---|
| 0 | Kiwi |
| 1 | Papaya |
| 2 | Birne |

### Dictionary

Ein Dictionary kann verschiedene Informationen zu einem Objekt speichern. Zum Beispiel können alle Informationen zu einer Person als Dictionary zusammengefasst werden.

| Schlüssel | Wert |
|---|---|
| 'name' | 'Peter' |
| 'age' | 32 |
| 'eyecolor' | 'brown' |

## Schlüssel : Wert (engl. key:value)

### Schlüssel

Der Schlüssel in einem Dictionary kann ein String oder eine Zahl sein, wobei Strings häufiger verwendet werden. Der Schlüssel muss eindeutig sein, es können als nicht zwei oder mehr Elemente

den gleichen Schlüssel haben.

### Wert

Die Werte in einem Dictionary können fast beliebige Datentypen sein. Wir können Zeichenketten, Zahlen, Objekte und sogar Listen und Dictionaries als Werte speichern.

### Beispiel für einen Dictionary

```
animal = {
  'species': 'Lion',
  'name': 'Leon',
  'age': 12
}
```

| Schlüssel | Wert |
|---|---|
| species | Lion |
| name | Leon |
| age | 12 |

## Dictionary anlegen

Wir definieren einen Dictionary indem wir die geschweiften Klammern **{ }** verwenden. Jedes Element wird mit `Schlüssel: Wert` definiert.

Um die Übersicht zu vereinfachen, **können** wir jedes Schlüssel/Wert-Element auf eine eigene Zeile schreiben.

```
colors = {
          'red': '#ff0000',
          'green': '#00ff00',
          'blue': '#0000ff'
         }
print(colors)
```

#### Output

```
{'red': '#ff0000', 'green': '#00ff00', 'blue': '#0000ff'}
```

Wenn der Schlüssel ein einzelnes Wort ist, können wir den Befehl **dict** verwenden.

```
colors = dict(
          red='#ff0000',
          green='#00ff00',
          blue='#0000ff'
```

```
        )
print(colors)
```

**Output**

```
{'red': '#ff0000', 'green': '#00ff00', 'blue': '#0000ff'}
```

# I want it all ...

Möchten Sie in einem Programm **alle** Informationen zu **vielen** Objekten speichern? Also vielleicht eine Mitgliederliste eines Vereins. Dann kombinieren Sie Listen und Dictionary; Sie erstellen eine Liste von Dictionaries.

| Index | Wert |
|---|---|
| 0 | {'name': 'Peter', 'age': 32, 'eyecolor': 'brown'} |
| 1 | {'name': 'Anna', 'age': 27, 'eyecolor': 'green'} |

```
# Dictionaries erstellen
member0 = {'name': 'Peter', 'age': 32, 'eyecolor': 'brown'}
member1 = {'name': 'Anna', 'age': 27, 'eyecolor': 'green'}

# Dictionaries in eine Liste einfügen
member_list = [member0, member1]

# ODER:
# Direkt eine Liste von Dictionaries erstellen
member_list = [
    {'name': 'Peter', 'age': 32, 'eyecolor': 'brown'},
    {'name': 'Anna', 'age': 27, 'eyecolor': 'green'}
]
```

---

M319-C1G, M319-C1F, M319-C1E

Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
**https://wiki.bzz.ch/de/modul/m319/learningunits/lu11/dictionary**

Last update: **2025/06/23 07:45**