LU11b - Verarbeitungen mit Dictionary

Dieses Kapitel zeigt Ihnen die Verarbeitungen (Lesen, Schreiben, Löschen) von Dictionaries

Elemente einfügen / ändern

Nachdem der Dictionary erstellt ist, können wir Elemente einfügen bzw. ändern. Dazu geben wir den Schlüssel in eckigen Klammern **[ ]** an.

- Ist der Schlüssel bereits vorhanden, so wird der bisherige Wert überschrieben.
- Andernfalls wird ein neuer Schlüssel und Wert eingetragen.

```
colors = {'red': '#ff0000', 'green': '#00ff00', 'blue': '#0000ff'}
colors['indigo'] = '#4B0082'  # adds a new key/value
colors['blue']   = '#000080'  # replaces the value for 'blue'
print(colors)
```

Output

```
{'red': '#ff0000', 'green': '#00ff00', 'blue': '#000080', 'indigo':
'#4B0082'}
```

Elemente löschen

Der Befehl `del` löscht ein Element aus dem Dictionary.

```
colors = {'red': '#ff0000', 'green': '#00ff00', 'blue': '#0000ff'}
del colors['green']   # Removes the element with the key 'green'
print(colors)
```

Output

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

Einzelne Elemente lesen

Wir können die Element über ihren Schlüssel in den eckigen Klammern lesen.

Wollen wir wissen, ob ein bestimmter Schlüssel exisitert, so verwenden wir key **`in`** `dictionary`.

```
colors = {'red': '#ff0000', 'green': '#00ff00', 'blue': '#0000ff'}
print(colors['green'])  # prints the value #00ff00
if 'pink' in colors:
    print ('Pink!!')
else:
    print ('No pink found')
```

Output

```
#00ff00
No pink found
```

Lesen anhand des Werts

Es existiert keine Funktion um ein Element anhand seines Werts zu lesen. Wir können uns behelfen, indem wir die Schlüssel und Werte in separate Listen umwandeln.

Die Funktion `values()` liest alle Werte eines Dictionaries und erzeugt ein View-Objekt als Returnwert. In Kombination mit `list(…)` können wir dieses View-Objekt in eine Liste umwandeln.

Die Funktion `keys()` funktioniert ähnlich und liefert eine View-Objekt aller Schlüssel.

```
colors = {'red': '#ff0000', 'green': '#00ff00', 'blue': '#0000ff'}
hex_values = list(colors.values())  # make a list of all values
color_names = list(colors.keys())   # make a list of all keys
index = hex_values.index('#00ff00') # search the index of the value
print(color_names[index])           # print the key by using the index
```

Output

```
green
```

Alle Elemente verarbeiten

Mit einem **for**-Loop können wir alle Elemente eines Dictionarys verarbeiten.

```
colors = {'red': '#ff0000', 'green': '#00ff00', 'blue': '#0000ff'}
for color_name in colors:    # loops through the keys
```

```
    print(color_name)

for key, value in colors.items():    # loops through the keys and values
    print(f'The hexcode for {key} is {value}')
print(colors)
```

Output

```
red
green
blue
The hexcode for red is #ff0000
The hexcode for green is #00ff00
The hexcode for blue is #0000ff
```

Sortieren

Mit dem **sorted**-Befehl können wir die Elemente nach ihrem Schlüssel sortieren.

```
colors = {'red': '#ff0000', 'green': '#00ff00', 'blue': '#0000ff'}
for name in sorted(colors.items()):
    print(f'The hexcode for {name} is {colors[name]}')
```

Output

```
The hexcode for blue is #0000ff
The hexcode for green is #00ff00
The hexcode for red is #ff0000
```

M319-C1G, M319-C1F, M319-C1E