

MODUL 13: DICTIONARYS

Ein Dictionary ist mit einem Wörterbuch zu vergleichen. In einem Wörterbuch findet man unter einem Schlüsselbegriff die zugeordnete Information. So steht etwa in einem englisch-deutschen Wörterbuch unter dem Eintrag „house“ der zugeordnete deutsche Begriff „Haus“.

1. Kompetenzen

- Ich kann ein Dictionary erstellen.
- Ich kann Elemente aus einem Dictionary lesen und schreiben.
- Ich kann überprüfen, ob sich ein Element in einem Dictionary befindet.
- Ich kann in einer Schleife über die Elemente eines Dictionarys laufen.

2. Eigenschaften

In Python stellen Dictionarys veränderliche Objekte dar und bestehen aus mehreren Paaren. Jedes Paar besteht aus einem eindeutigen Schlüssel und einen zugeordneten Wert. Über den Schlüssel greift man auf den Wert zu. Als Schlüssel werden meistens Strings verwendet, es können aber auch andere unveränderliche Objekte (Zahlen, Tupel) benutzt werden. Die Schlüssel sind ungeordnet, daher ist auch die Reihenfolge bei der Ausgabe eines gesamten Dictionarys nicht festgelegt.

Im folgenden Beispiel sollen mehrere Personen und ihre Altersangaben in einem Dictionary erfasst und bearbeitet werden. Der Name der jeweiligen Person dient als Schlüssel. Über den Namen kann auf das Alter der Person (auf den Wert des Schlüssels) zugegriffen werden.

```
# Erzeugung eines Dictionarys
alter = {'Peter':31, 'Julia':28, 'Werner':35}
print(alter)

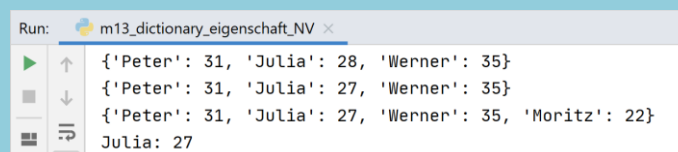
# Ersetzen eines Werts
alter['Julia'] = 27
print(alter)

# Ein Element hinzu
alter['Moritz'] = 22
print(alter)

# Ausgabe
print('Julia:', alter['Julia'])
```

Speichere die Datei unter **m13_dictionary_eigenschaft.py**

Ausgabe:



```
Run: m13_dictionary_eigenschaft_NV x
{'Peter': 31, 'Julia': 28, 'Werner': 35}
{'Peter': 31, 'Julia': 27, 'Werner': 35}
{'Peter': 31, 'Julia': 27, 'Werner': 35, 'Moritz': 22}
Julia: 27
```

Erläuterung:

- Es wird das Dictionary `alter` mit drei Informationspaaren erzeugt und ausgegeben. Dictionarys werden mithilfe von geschweiften Klammern (`{}`) erzeugt. Die Paare werden durch Komma voneinander getrennt, ein Paar wird in der folgenden Form notiert: **Schlüssel: Wert**.
- Auf einem Element greift man über die Angabe des Schlüssels in eckigen Klammern `[]` zu. Dies wird im vorliegenden Beispiel für Zuweisung und Ausgabe genutzt.
- Elemente sind **veränderlich**. Hier wird die Altersangabe von `'Julia'` verändert.
- Elemente können **hinzugefügt** werden. Hier wird Dictionary das Paar `'Moritz': 22` hinzugefügt, da kein Element mit dem Schlüssel `'Moritz'` gefunden wurde.

3. Funktionen

Es gibt eine Reihe von Funktionen zur Bearbeitung von Dictionarys. Einige werden im folgenden Programm verdeutlicht:

```
# Erzeugung
alter = {'Peter':31, 'Julia':28, 'Werner':35}
print(alter)

# Element enthalten?
if 'Julia' in alter:
    print(alter['Julia'])

# Entfernen eines Elements
del alter['Julia']

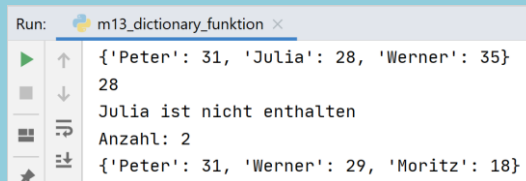
# Element enthalten?
if 'Julia' not in alter:
    print('Julia ist nicht enthalten')

# Anzahl Elemente
print('Anzahl:', len(alter))

# Aktualisierung mit zweitem Dictionary
ualter = {'Moritz': 18, 'Werner': 29}
alter.update(ualter)
print(alter)
```

Speichere die Datei unter **m13_dictionary_funktion.py**

Ausgabe:



Erläuterung:

- Ein einzelnes Element wird mithilfe der Funktion `del()` aus der Liste gelöscht.
- Die Existenz eines Elements wird mithilfe des Operators `in` geprüft.
- Die Anzahl der Elemente wird mit der Funktion `len()` ermittelt.
- Ein Dictionary aktualisiert man mit der Funktion `update()` mit einem anderen Dictionary. Dabei erhalten vorhandene Elemente gegebenenfalls einen neuen Wert, neue Elemente werden angehängt. Die beiden Dictionarys werden also zusammengeführt.

4. Views

Die Funktionen `keys()`, `items()` und `values()` erzeugen sogenannte Views eines Dictionarys. Diese Views verändern sich seit Python 3 unmittelbar, falls sich das zugeordnete Dictionary verändert. Dictionarys sind also dynamisch.

Beispiel:

```
# Erzeugung
alter = {'Peter':31, 'Julia':28, 'Werner':35}

# Werte
w = alter.values()
print('Anzahl Werte:', len(w))
for x in w:
    print(x)
if 31 in w:
    print('31 ist enthalten')
alter['Peter'] = 41
if 31 not in w:
```

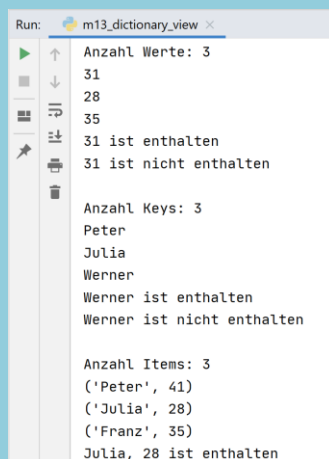
```
print('31 ist nicht enthalten')
print()

# Keys
k = alter.keys()
print('Anzahl Keys:', len(k))
for x in k:
    print(x)
if 'Werner' in k:
    print('Werner ist enthalten')
del alter['Werner']
if 'Werner' not in k:
    print('Werner ist nicht enthalten')
print()

# Items
i = alter.items()
alter['Franz'] = 35
print('Anzahl Items:', len(i))
for x in i:
    print(x)
if ('Julia', 28) in i:
    print('Julia, 28 ist enthalten')
```

Speichere die Datei unter `m13_dictionary_view.py`

Ausgabe:



Erläuterung der Werte-View:

- Mithilfe der Funktion `values()` wurde eine View (w) der Werte des Dictionarys erzeugt.
- Den Inhalt der View kann mithilfe einer `for-Schleife` und des Operators `in` ausgegeben werden.
- Wiederum mithilfe des Operators `in` kann geprüft werden, ob ein bestimmter Wert in der View existiert.
- Der Wert eines Dictionary-Elements wird verändert. Dies hat auch Auswirkungen auf die zugehörige View. Diese muss also nicht mehr neu erzeugt werden. Der ursprüngliche Wert wird nach der Änderung nicht mehr gefunden.

Erläuterung der Schlüssel-View:

- Mithilfe der Funktion `keys()` wurde eine View (k) der Keys des Dictionarys erzeugt.
- Den Inhalt der View kann mithilfe einer `for-Schleife` und des Operators `in` ausgegeben werden.
- Wiederum mithilfe des Operators `in` kann geprüft werden, ob ein bestimmter Key in der View existiert.
- Ein Dictionary-Element wird gelöscht. Dies hat auch Auswirkungen auf die zugehörige View. Das ursprünglich vorhandene Dictionary-Element wird nach dem Löschen nicht mehr gefunden.

Erläuterung der Elemente-View:

- Mithilfe der Funktion `items()` wurde eine View (i) der Elemente des Dictionarys erzeugt.
- Den Inhalt der View kann mithilfe einer `for-Schleife` und des Operators `in` ausgegeben werden.
- Wiederum mithilfe des Operators `in` kann geprüft werden, ob ein bestimmtes Element, also eine Schlüssel-Wert-Kombination, in der View existiert.

Übung

1. Erstelle ein Dictionary, das deutsche Wörter ihrer englischen Übersetzung zuordnet.
 - a) Versuchen, die Einträge deines Dictionary alphabetisch sortiert auszugeben.
 - b) Schreibe ein Programm, das nach einem Wort in Deutsch fragt und die Übersetzung liefert.
 - c) Schreibe ein Programm, das sowohl Wörter von Deutsch nach Englisch als auch umgekehrt übersetzt.

Speichere die Datei unter `m13_woerterbuch.py`

Abgaben:

- `m13_dictionary_eigenschaft.py`
- `m13_dictionary_funktion.py`
- `m13_dictionary_view.py`
- `m13_woerterbuch.py`