

## 21 Datentyp Dictionary (Assoziatives Array)

### 21.1 Beschreibung

- eine Dictionary (assoziatives Array) ist mit einem Wörterbuch vergleichbar, zu einem Schlüsselbegriff gibt es Information bzw. Werte
- jedes Element (jeder Eintrag) besteht aus einem Schlüssel-Werte-Paar
- der Schlüssel muss eindeutig sein
- der Schlüssel lässt nicht ändern
- der Wert lässt sich ändern
- mit Schlüssel wird auf dem Wert zugegriffen
- die Schlüssel-Werte-Paare sind ungeordnet
- Schlüssel und Wert können verschiedene Typen haben
- ein Dictionary kann unterschiedliche Datentypen enthalten
- ein Dictionary wird durch die Zuweisung von Elementen innerhalb geschweiften Klammern angelegt, wobei Schlüssel und Wert durch : zugewiesen werden: `dict_name = {"schl": "wert"}`
- mit Dictionary-Operatoren und Dictionary-Funktionen lassen sich Dictionary ändern und untersuchen
- ein leeres Dictionary wird definiert durch `dict_name = {}`

- Beispiel:

*# Dictionary anlegen (Schlüssel und Wert vom Datentyp str)*

*# Schlüssel – Autokennzeichen : Wert – Stadt*

```
autokennzeichen = {"HH": "Hamburg",  
                  "B": "Berlin",  
                  "M": "München"}
```

*# weiteres Dictionary anlegen (Schlüssel und Wert vom Datentyp str)*

*# Schlüssel – deutsche Vokabel : Wert – englische Vokabel*

```
vokabeln = {"Hund": "dog",  
            "Katze": "cat",  
            "Vogel": "bird"}
```

*# Dictionary anlegen (Schlüssel Datentyp str – Wert Datentyp int)*

*# Schlüssel – Land : Wert – Einwohner in Mio.*

```
laender = {"Frankreich": 67,  
           "Italien": 60,  
           "Schweiz": 8}
```

## 21.2 Dictionary-Operatoren

- Zugriff auf ein Element per Schlüssel mit **[s]**
- Prüfen, ob ein Schlüssel im Dictionary enthalten ist mit **in**
- Prüfen, ob ein Schlüssel im Dictionary NICHT enthalten ist mit **not in**
- Schlüssel und Wert aus dem Dictionary löschen **del dict[s]**

- Beispiel:

```
# Dictionary anlegen (Schlüssel und Wert vom Datentyp str)
# Schlüssel - Autokennzeichen : Wert - Stadt
autokennzeichen = {"HH": "Hamburg",
                  "B": "Berlin",
                  "M": "München"}

# weiteres Dictionary anlegen (Schlüssel und Wert vom Datentyp str)
# Schlüssel - deutsche Vokabel : Wert - englische Vokabel
vokabeln = {"Hund": "dog",
            "Katze": "cat",
            "Vogel": "bird"}

# Dictionary anlegen (Schlüssel Datentyp str - Wert Datentyp int)
# Schlüssel - Land : Wert - Einwohner in Mio.
laender = {"Frankreich": 67,
           "Italien": 60,
           "Schweiz": 8}

# Dictionary ausgeben
print(autokennzeichen)

{'HH': 'Hamburg', 'B': 'Berlin', 'M': 'München'}
```

```
# Anzahl der Schlüssel-Werte-Paare ausgeben
print("Anzahl der Schlüssel-Werte-Paare:", len(autokennzeichen))

Anzahl der Schlüssel-Werte-Paare: 3

# Element mit dem Schlüssel B ausgeben
print(autokennzeichen["B"])

Berlin

# Element mit dem Schlüssel Katze ausgeben
print(vokabeln["Katze"])

cat

# Element (Einwohner in Mio.) mit dem Schlüssel Schweiz ausgeben
land = "Schweiz"
print(land, laender[land])

Schweiz 8]
```

### 21.3 Länge eines Dictionary (Anzahl der Elemente)

- Länge eines Dictionary (Anzahl der Elemente) **len(dict)**

- Beispiele:

```
laender = {"Frankreich": 67,  
           "Italien": 60,  
           "Schweiz": 8}  
  
# Dictionary-Länge ausgeben  
print("Länge:", len(laender))
```

```
Länge: 3
```

## 21.4 Dictionary-Funktionen (Methoden)

- alle Elemente aus dem Dictionary löschen `.clear()`
- eine Kopie von einem Dictionary anlegen `.copy()`
- Wert zu einem Schlüssel liefern, falls der Schlüssel nicht existiert wird ein Alternativtext zurückgegeben `.get("abc", "nicht da")`
- Wert zu einem Schlüssel liefern und anschließend das Element (Schlüssel-Werte-Paar) löschen `.pop("abc")`
- ein Dictionary zu einem Dictionary hinzufügen, gleiche Elemente werden überschrieben `.update(dict_2)`
- eine dynamische View (als iterierbare Elemente) aller Schlüssel erzeugen `.keys()`
- eine dynamische View (iterierbare Elemente) aller Werte erzeugen `.values()`
- eine dynamische View (iterierbare Elemente) aller Schlüssel-Werte-Paare erzeugen `.items()`

- Beispiele:

```
# Dictionary anlegen (Schlüssel Datentyp str - Wert Datentyp int)
# Schlüssel - Land : Wert - Einwohner in Mio.
laender = {"Frankreich": 67,
           "Italien": 60,
           "Schweiz": 8}
```

```
# Element (Schlüssel-Werte-Paar) aus Dictionary entfernen
print("vor dem Löschen", laender)
del laender["Schweiz"]
print("nach dem Löschen", laender)
```

```
vor dem Löschen {'Frankreich': 67, 'Italien': 60, 'Schweiz': 8}
nach dem Löschen {'Frankreich': 67, 'Italien': 60}
```

```
# prüfen, ob ein Schlüssel enthalten ist
if "Spanien" in laender:
    print("Spanien ist enthalten")
else:
    print("Spanien ist nicht enthalten")
```

```
Spanien ist nicht enthalten
```

```
# Schlüssel-Werte-Paar hinzufügen, falls es noch nicht existiert
laender["Belgien"] = 1
print("nach dem Hinzufügen", laender)
laender["Belgien"] = 10
print("nach dem 2. Hinzufügen", laender)
```

```
nach dem Hinzufügen {'Frankreich': 67, 'Italien': 60, 'Belgien': 1}
nach dem 2. Hinzufügen {'Frankreich': 67, 'Italien': 60, 'Belgien': 10}
```

```

# einem Dictionary ein Dictionary hinzufügen
# gleiche Elemente werden überschrieben
benelux = {"Belgien": 11, "Niederlande": 17, "Luxemburg": 0.6}
laender.update(benelux)
print("nach dem Update", laender)

nach dem Update {'Frankreich': 67, 'Italien': 60, 'Belgien': 11,
'Niederlande': 17, 'Luxemburg': 0.6}

# dynam. Views auf ein Dictionary als iterierbares Objekt
# Schlüssel
for k in laender.keys():
    print(k, type(k))

[Frankreich <class 'str'>
Italien <class 'str'>
Schweiz <class 'str'>

# Werte
for w in laender.values():
    print(w, type(w))

67 <class 'int'>
60 <class 'int'>
8 <class 'int'>

# Schlüssel-Werte-Paare
for i in laender.items():
    print(i, type(i))

('Frankreich', 67) <class 'tuple'>
('Italien', 60) <class 'tuple'>
('Schweiz', 8) <class 'tuple'>

```

## Aufgabe

1. Ein Dictionary mit den Vornamen und dem Alter (Ganzzahl) von 5 Personen ist zu erstellen. Der Vorname ist der Schlüssel und das Alter der Wert in diesem Dictionary. Das Dictionary ist als Ganzes auf der Konsole auszugeben. Anschließend sind in einer Schleife nacheinander alle Elemente als Schlüssel-Werte-Paar auszugeben.
2. Eine Person hatte Geburtstag und das Alter dieser Person soll um 1 erhöht werden. Das Schlüssel-Werte-Paar zu dieser Person soll vor und nach der Änderung ausgegeben werden.
3. Alle Namen der Personen sollen ohne Alter auf der Konsole ausgegeben werden.
4. Für alle Personen im Dictionary soll auch die Telefonnummer ergänzt werden. Dazu kann ein weiteres Dictionary angelegt werden, in dem die Schlüssel Geburtsjahr und Telefon verwendet werden, z.B. {"Geburtsjahr": 1985, "Telefon": "0151 23456"}. Mit diesen Angaben ist die Altersangabe zu überschreiben. Für die weiteren Personen sind diese Änderungen ebenfalls vorzunehmen. Das Personen-Dictionary ist komplett auszugeben.
5. Eine Person hat 2 Telefonnummern. Es ist eine Liste mit den beiden Telefonnummern zu erstellen und anschließend der bestehende Eintrag Telefon mit der Liste der Telefonnummern zu ersetzen. Anschließend sind in einer Schleife nacheinander alle Elemente als Schlüssel-Werte-Paar auszugeben.