

Merkblatt Dictionaries

1 Dictionaries

Python kennt einen Datentyp, der Beziehungen zwischen Objekten beschreibt. Das sind sogenannte Dictionaries. Es wird auch vom Datentyp “Mapping” und in anderen Programmiersprachen von *Hashes* gesprochen.

```
[1]: user = {"name": "Hans", "age": "36", "phone": "555-2134"}
```

Anders als Tupel sind Dictionaries veränderbar. Einzelne Elemente können geändert werden.

Elemente können hinzugefügt werden.

1.1 Element hinzufügen

```
[2]: user["email"] = "orakel@olymp.org"
print(user)
```

```
{'name': 'Hans', 'age': '36', 'phone': '555-2134', 'email': 'orakel@olymp.org'}
```

1.2 Element löschen

```
[3]: del user["phone"]
print(user)
```

```
{'name': 'Hans', 'age': '36', 'email': 'orakel@olymp.org'}
```

2 .items()

```
[4]: pizzas = {
    "Margherita" : "Tomate, Basilikum und Käse",
    "Prosciutto e funghi" : "Schinken und Pilze",
    "Quattro Formaggi" : "Vier Käse",
    "Diavolo" : "Salami und Pepperoni",
    "Tonno" : "Thunfisch und Zwiebel",
    "Hawaii" : "Schinken und Ananas"
}
```

```
[5]: pizzas.items()
```

```
[5]: dict_items([('Marghertia', 'Tomate, Basilikum und Käse'), ('Prosciutto e  
funghi', 'Schinken und Pilze'), ('Quattro Formaggi', 'Vier Käse'), ('Diavolo',  
'Salami und Pepperoni'), ('Tonno', 'Thunfisch und Zwiebel'), ('Hawaii',  
'Schinken und Ananas')])
```

`items()` ist von Bedeutung für den Aufruf in Schleifen. Mit nachfolgendem Code kann auf Schlüssel (*key*) und Wert (*value*) von `pizzas` zugegriffen werden:

```
[6]: for key, value in pizzas.items():  
      print(key + "\n" + value + "\n\n")
```

```
Marghertia  
Tomate, Basilikum und Käse
```

```
Prosciutto e funghi  
Schinken und Pilze
```

```
Quattro Formaggi  
Vier Käse
```

```
Diavolo  
Salami und Pepperoni
```

```
Tonno  
Thunfisch und Zwiebel
```

```
Hawaii  
Schinken und Ananas
```

3 .keys()

Auf die Schlüssel (*keys*) kann separat mit `.keys()` zugegriffen werden.

```
[7]: print(pizzas.keys())
```

```
dict_keys(['Marghertia', 'Prosciutto e funghi', 'Quattro Formaggi', 'Diavolo',  
'Tonno', 'Hawaii'])
```

4 .values()

Auf die Werte (*values*) kann separat mit `.values()` zugegriffen werden.

```
[8]: print(pizzas.values())
```

```
dict_values(['Tomate, Basilikum und Käse', 'Schinken und Pilze', 'Vier Käse',  
'Salami und Pepperoni', 'Thunfisch und Zwiebel', 'Schinken und Ananas'])
```