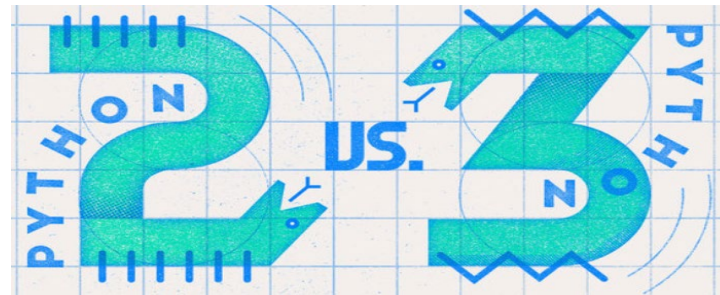




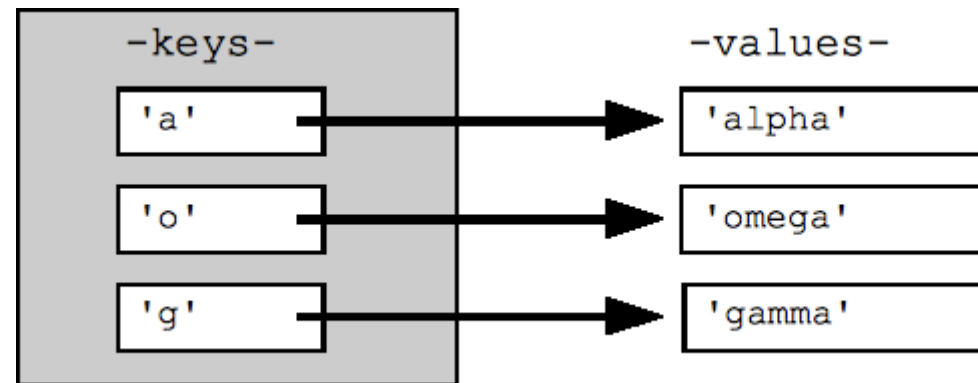
Dictionary

Map / Hashtable



Dictionary

- A dictionary maps keys to values (objects, ...).



- Finding the object is simplified using its key (-> no search required)



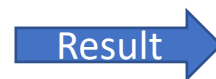
Example: Dictionary

```
# Colortable englisch -> french  
colors = {"red" : "rouge", "green" : "vert", "blue" : "bleu",  
         "yellow":"jaune", "black":"noir", "white":"blanche"}
```

```
print("black ->", colors["black"])  
print("red ->", colors["red"])
```

The keys : values pairs are each connected by a colon.

The value of a key is found by specifying the key in [...]



```
black -> noir  
red -> rouge
```



Example: Morse alphabet

Excerpt from the Morse alphabet

```
# Morse Alphabet  
morse = {"A": ".-.", "B": "-...", "C": "-.-.", "D": "-..", "E": ".",  
         "G": "-.-.", "H": "....", "I": "..", "S": "...", "R": "-.-."}
```

The "encrypting" of the letters

```
print(morse["D"], morse["A"], morse["S"])  
print(morse["D"], morse["E"], morse["R"])
```



- -
- -



Example: Morse alphabet

Excerpt from the Morse alphabet

```
# Morse Alphabet  
morse = {"A": ".-.", "B": "-...", "C": "-.-.", "D": "-..", "E": ".", 'G': "-.-.", "H": "....", "I": "..", "S": "...", "R": ".-."}
```

Adding elements to the Morse alphabet:

```
morse['F'] = "..-."  
morse['K'] = "-.-"
```



The dict() method

A dictionary can also be created by the function `dict(. . .)`

The capitals of Europe

```
# Europe's capitals  
capitals = dict(Italy='Roma', Austria="Vienna", Switzerland='Bern',  
                Germany='Berlin')  
# Insert further capitals  
capitals["France"] = 'Paris'  
capitals['Poland'] = "Warszawa"
```

capitals is a dictionary
The values are mapped to their key with an
equal sign.

The values can be enclosed in single ' ' or in
double " " quotes.

Here, too, the dictionary can be
supplemented later with new values.

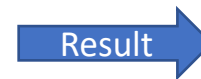


Print out the Dictionary

A dictionary can be iterated in the same way as a list.

Access is gained via the keys using the keys() method.

```
# Print all items
for k in capitals.keys():
    print(k, " -> ", capitals[k])
```



```
Italy -> Roma
Austria -> Vienna
Switzerland -> Bern
Germany -> Berlin
Netherlands -> Amsterdam
France -> Paris
Poland -> Warszawa
```

Print out the values:

```
for v in capitals.values():
    print(v)
```



```
Roma
Vienna
Bern
Berlin
Amsterdam
Paris
Warszawa
```



Further Dictionary Methods

<code>capitals.keys()</code>	returns the list of keys
<code>capitals.values()</code>	returns the list of values
<code>capitals.get(key)</code>	returns the value of this key
<code>capitals.pop(key)</code>	deletes the entry for this key
<code>len(capitals)</code>	returns the number of entries
<code>key in capitals</code>	returns true if the key exists
<code>capitals.clear()</code>	deletes all entries



Examples

```
capitals = dict(Italy='Roma', Austria="Vienna",  
               Switzerland='Bern', Germany='Berlin')
```

```
print(list(capitals.keys()))  
print(list(capitals.values()))
```

```
capitals["Lichtenstein"] = "Vaduz"  
print(capitals)
```

```
print(capitals.get("Italy"))  
print("Italy" in capitals)
```

```
print(len(capitals))  
capitals.pop("Italy")  
print(list(capitals.values()))
```

```
['Italy', 'Austria', 'Switzerland', 'Germany']  
['Roma', 'Vienna', 'Bern', 'Berlin']
```

```
{'Italy': 'Roma', 'Austria': 'Vienna', 'Switzerland': 'Bern',  
 'Germany': 'Berlin', 'Lichtenstein': 'Vaduz'}
```

```
Roma  
True
```

```
5  
['Vienna', 'Bern', 'Berlin', 'Vaduz']
```