



WOCHE 6

TUPEL & DICTIONARIES



LERNZIELE

- Tupel und Dictionaries als Datenstrukturen kennenlernen
- Wichtige Begriffe:
 -  Tupel
 -  Dictionary

DEFINITION: TUPEL

- Liste mit Einschränkungen
- Tupel kann *nicht* verändert werden (immutable)
- Notation: **(5,7,3)**

UMGANG MIT TUPELN

TUPEL ANLEGEN

- (<Wert_1> , <Wert_2>, ...)

```
values = ("Foo",1,True)
```

UMGANG MIT TUPELN

WERTE ZÄHLEN: `count()`

```
numbers = (1,3,3,3,5,9,2)
print(numbers.count(3)) # gibt 3 zurück
```

UMGANG MIT TUPELN

WERT FINDEN: `index()`

```
numbers = (1,3,4,9,2)
print(numbers.index(3)) # gibt 1 zurück
```

TUPLE PACKING / UNPACKING

- Tupel = "Liste" zusammenhängender Daten
- Beispiele:
 - Tabelleneintrag (Spalte1, Spalte2, ...)
 - Daten + Metadaten
- *Tuple Packing*: N Werte in Tupel packen
- *Tuple Unpacking*: Aus Tupel N Variablen entpacken

TUPLE UNPACKING

```
fruits = [("Ananas", 50), ("Banane", 89), ("Birne", 57)]  
for fruit in fruits:  
    fruit_name, kcal = fruit
```

- Liste mit 3 Tupeln
- Jedes Tupel enthält Name + kcal/100g
- Unpacking = Tupel wird in 2 Variablen "entpackt"
 - **fruit_name** = "Ananas"
 - **kcal** = 50

TUPLE PACKING

```
person = "Anton" , "Meier", 29  
print(person) # ("Anton", "Meier", 29)  
print(person[0]) # "Anton"
```

- Tuple Packing = Werte werden zu einem Tupel gepackt

TUPEL VS. LISTEN

Tupel

Inhalt unveränderbar
(immutable)

sinnvoll für *heterogene*
Daten

("Anton", "Meier", 29)

Listen

Inhalt veränderbar
(mutable)

sinnvoll für *homogene*
Daten

["Anton", "Meier",
"29"]



LIVE-CODING

AUFGABENBLATT 1



DICTIONARIES

DEFINITION: DICTIONARIES

- Schlüssel-Wert-Paare
- {key: value}
- Zu jedem Schlüssel ein Wert
- Schlüssel sind eindeutig

AUFBAU

```
person = {"name" : "John",  
          "age" : 25,  
          "email" : "john@example.com"}
```

person	
name	"John"
age	25
email	"john@example.com"

- Item
- key
- value

DICTIONARY ERZEUGEN

```
person = {"first_name" : "Anna",  
          "last_name": "Meier",  
          "age": 29}
```

- Eintrag: **key : value**
- Einträge mit Kommata getrennt
- Key muss unveränderbarer Datentyp sein

str **int** **float** **tuple**

- idR. **int** oder **str**

UMGANG MIT DICTIONARIES

ZUGRIFF AUF WERTE

```
capitols = {    "germany":"berlin",  
                "france" : "paris",  
                }
```

- **dict[key]**

```
print(capitols["germany"]) # "berlin"  
print(capitols["hogwarts"]) # KeyError
```

- **get(key, [default])**

```
print(capitols.get("germany", "not found") # "berlin"  
print(capitols.get("hogwarts", "not found")) # "not found"
```

WERTE PRÜFEN

```
capitols = {    "germany" : "berlin",  
                "france"  : "paris"  
            }
```

- Mit Keyword **in** lässt sich überprüfen ob **key** vorhanden ist

```
print("germany" in capitols) # True  
print("narnia" in capitols) # False
```

WERTE HINZUFÜGEN

```
capitols = {  
    "germany" : "berlin",  
    "france" : "paris"  
}  
more_capitols = {  
    "hungary" : "budapest",  
    "netherlands" : "amsterdam",  
    "germany" : "berlin"  
}
```

- **dict[key]**

```
capitols["spain"] = "madrid"
```

- **update(d)**

- Fügt mehrere Dictionaries zusammen

```
capitols.update(more_capitols)
```

WERTE LÖSCHEN

```
word_frequency = {"a" : 235,  
                  "is" : 48,  
                  "python" : 6}
```

- **del**

```
del word_frequency["python"]
```

- **pop(key, [default])**

- Entfernt **key** aus dem Dictionary
- gibt zusätzlich **value** oder **default** zurück

```
print(word_frequency.pop("is")) # 48  
print(word_frequency.pop("Düsseldorf"), 0) # 0
```

DICTIONARY ITERIEREN

- **items()**: Gibt Einträge **(key, value)** zurück

```
for key,value in capitol.items():  
    print(item)
```

- **values()**: Gibt nur **values** zurück

```
for value in capitol.values():  
    print(item)
```

- **keys()**: Gibt nur **keys** zurück

```
for key in capitol.keys():  
    print(item)
```

LIVE-CODING

A man with a mustache and glasses, wearing an orange polo shirt, is seated at a wooden desk. He is looking at a large, vintage CRT computer monitor on the left side of the frame. His hands are on a keyboard, and he appears to be typing. The desk is cluttered with various items, including a black folder or binder. In the background, there is a brick fireplace and a window with white curtains. The overall scene suggests a home office or a personal workspace from the late 20th century.

AUFGABENBLATT 2



VIELEN DANK!