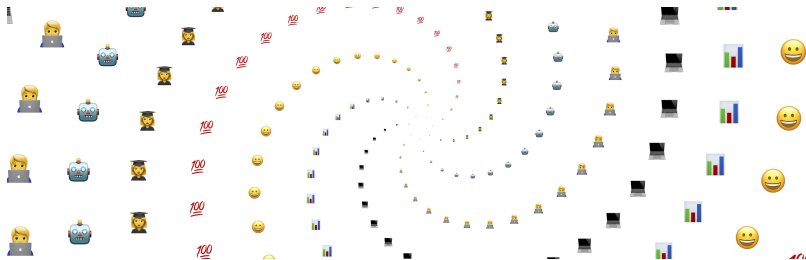


Programmieren: Dictionaries

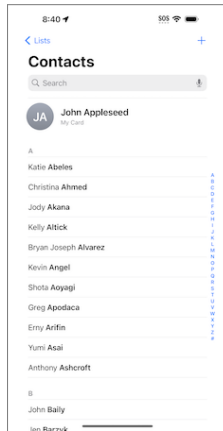
Dictionaries

Cyril Wendl

Fachschaft Informatik
Kantonsschule im Lee

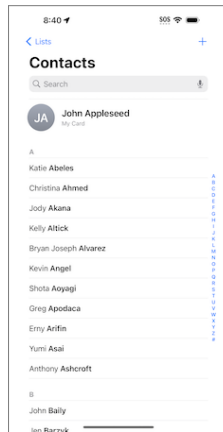


Kontakte-App Programmieren



Womit würden Sie beginnen?

Kontakte-App Programmieren



Womit würden Sie beginnen?
Wie können wir alle Namen und Telefonnummern in Python
speichern?

Listen als Telefonbuch

Namen:

```
namen = ["Anna", "Clara", "Ben"]
```

Nummern:

```
nummern = ["079 123 45 67", "077 555 44 33", "078 111 22 33"]
```

- ▶ Wie findet man die Nummer von "Clara"?



Listen als Telefonbuch

Namen:

```
namen = ["Anna", "Clara", "Ben"]
```

Nummern:

```
nummern = ["079 123 45 67", "077 555 44 33", "078 111 22 33"]
```

- ▶ Wie findet man die Nummer von "Clara"?
- ▶ *Was passiert, wenn die Reihenfolge nicht stimmt?*



Listen als Telefonbuch

Namen:

```
namen = ["Anna", "Clara", "Ben"]
```

Nummern:

```
nummern = ["079 123 45 67", "077 555 44 33", "078 111 22 33"]
```

- ▶ Wie findet man die Nummer von "Clara"?
- ▶ *Was passiert, wenn die Reihenfolge nicht stimmt?*
- ▶ *Was, wenn ein Name fehlt?*



Listen als Telefonbuch

Namen:

```
namen = ["Anna", "Clara", "Ben"]
```

Nummern:

```
nummern = ["079 123 45 67", "077 555 44 33", "078 111 22 33"]
```

- ▶ Wie findet man die Nummer von "Clara"?
- ▶ *Was passiert, wenn die Reihenfolge nicht stimmt?*
- ▶ *Was, wenn ein Name fehlt?*
 - ▶ **Gibt es eine bessere Lösung?**




Was ist ein Dictionary?

- ▶ Eine Sammlung von **Schlüssel-Wert-Paaren**




Was ist ein Dictionary?

- ▶ Eine Sammlung von **Schlüssel-Wert-Paaren**
- ▶ Zugriff nicht über Index, sondern über einen eindeutigen **Schlüssel** 



Was ist ein Dictionary?

- ▶ Eine Sammlung von **Schlüssel-Wert-Paaren**
- ▶ Zugriff nicht über Index, sondern über einen eindeutigen **Schlüssel** 
- ▶ **Beispiel:**

```
telefonbuch = {  
    "Anna": "079 123 45 67",  
    "Clara": "077 555 44 33"  
}  
print(telefonbuch["Anna"]) # druckt 079 123 45 67
```

Schlüssel sind hier Namen "Anna" und "Clara", Werte sind die Telefonnummern.



Motivation: Schlüssel-Wert-Paare

- ▶ Wie speichert man zu einem Namen eine Telefonnummer?



Motivation: Schlüssel-Wert-Paare

- ▶ Wie speichert man zu einem Namen eine Telefonnummer?
- ▶ Wie kann man zu einem Produkt den Preis speichern?

Motivation: Schlüssel-Wert-Paare

- ▶ Wie speichert man zu einem Namen eine Telefonnummer?
- ▶ Wie kann man zu einem Produkt den Preis speichern?
- ▶ Wie kann man zu einem Land die Hauptstadt speichern?

Motivation: Schlüssel-Wert-Paare

- ▶ Wie speichert man zu einem Namen eine Telefonnummer?
- ▶ Wie kann man zu einem Produkt den Preis speichern?
- ▶ Wie kann man zu einem Land die Hauptstadt speichern?
- ▶ Mit Listen: möglich, aber unpraktisch (z.B. Index-Suche)



Motivation: Schlüssel-Wert-Paare

- ▶ Wie speichert man zu einem Namen eine Telefonnummer?
- ▶ Wie kann man zu einem Produkt den Preis speichern?
- ▶ Wie kann man zu einem Land die Hauptstadt speichern?
- ▶ Mit Listen: möglich, aber unpraktisch (z.B. Index-Suche)
- ▶ Lösung: **Dictionaries** (Wörterbücher)



Dictionary: Definition

Definition

Ein **Dictionary** ist eine Datenstruktur, die jedem Schlüssel (*key*) einen Wert (*value*) zuordnet. Die Schlüssel müssen eindeutig sein und können z.B. Zahlen oder Zeichenketten sein.



Beispiel: Produktpreise

```
preise = {  
    "Apfel": 0.80,  
    "Brot": 2.50  
}  
print(preise["Brot"]) # gibt 2.5 aus
```



Werte hinzufügen und ändern

```
# Dictionary erstellen
preise = {
    "Apfel": 0.80,
    "Brot": 2.50
}


preise["Milch"] = 1.60    # neues Produkt hinzufügen
preise["Apfel"] = 0.90    # Preis ändern

print(preise)
# gibt {'Apfel': 0.9, 'Brot': 2.5, 'Milch': 1.6} aus
```

Wie bei Listen, aber mit Schlüsseln!

Alle Schlüssel und Werte durchgehen


```
for produkt in preise:  
    print(produkt, "kostet", preise[produkt], "Franken")
```

► `for produkt in preise` geht durch alle **Schlüssel** 



Alle Schlüssel und Werte durchgehen

```
for produkt in preise:  
    print(produkt, "kostet", preise[produkt], "Franken")
```

- ▶ `for produkt in preise` geht durch alle **Schlüssel** 
- ▶ `preise[produkt]` gibt den Wert zum Schlüssel zurück








Überprüfen, ob ein Schlüssel existiert

```
preise = {  
    "Apfel": 0.80,  
    "Brot": 2.50  
}  
  
if "Brot" in preise:  
    print("Brot ist im Dictionary vorhanden.")  
else:  
    print("Brot ist nicht im Dictionary vorhanden.")
```



Aufgaben

Programmier-Skript, Kapitel 6.2

- ▶  Lesen Sie Definition 6.3, Beispiele 6.10-6.14
- ▶  6.21-6.25
- ▶ Aufgaben 6.24 und 6.25 auf Moodle (Lernkontrolle)
- ▶  Challenge:
 - ▶  6.26-6.27
 - ▶  Weitere Moodle-Aufgaben

Zeichen		Windows		MacOS
[Alt Gr + ü		⌘ + 5
]		Alt Gr + !		⌘ + 6
{		Alt Gr + ä		⌘ + 8
}		Alt Gr + \$		⌘ + 9

