

Dictionaries

Foliensatz 3

Jasmin Noll
wi20086@lehre.dhbw-stuttgart.de

21.04.2022

Was euch heute erwartet

- Dictionaries
 - Konventionen
 - Auf Elemente zugreifen
 - Elemente ändern
 - Elemente hinzufügen
 - Elemente löschen
 - Weitere Nice-to-Know-Methoden

Dictionaries

Konventionen

- Ein Dictionary besteht aus *key-value*-Paaren
 - Dabei ist ein *key*-Element eindeutig, während *value*-Elemente mehrfach auftauchen können
- Dictionaries können alle Datentypen enthalten
- Dictionaries werden mit { } erstellt und ausgegeben
 - Beispiel: {
 "key 0": "Element 0",
 "key 1": "Element 1",
 "key 2": "Element 2"
}

Auf Elemente zugreifen

- Zugriff auf ein Element mit `[]` und *key*
 - Beispiel: `alien["key 0"]` → Ergebnis: "Element 0"
- Zugriff auf ein Element mit `.get()` und *key*
 - Beispiel: `alien.get("key 0")` → Ergebnis: "Element 0"
- Alle *keys* eines Dictionaries mit `.keys()` erhalten
 - Beispiel: `alien.keys()` → Ergebnis: ["key 0", "key 1", ..., "key n"]
- Alle *values* eines Dictionaries mit `.values()` erhalten
 - Beispiel: `alien.values()` → Ergebnis: ["Element 0", "Element 1", ..., "Element n"]
- Alle *key-value*-Paare mit `.items()` erhalten
 - Beispiel: `alien.items()` → Ergebnis: [("key 0", "Element 0"), ("key 1", "Element 1"), ..., ("key n", "Element n")]

Elemente ändern

- Um ein Element zu ändern wird es (wie bei einem Zugriff) ausgewählt und der neue Wert wird übergeben
 - Beispiele:
 - `alien["key 0"] = "neuer Wert"`
- Ebenfalls kann dafür die `.update()`-Funktion verwendet werden
 - Dafür verwendet man den key für das zuänderte Element und den neuen Wert
 - Beispiel: `alien.update({"key 0": "neuer Wert"})`

Elemente hinzufügen

- Dem Dictionary mit neuem key in [] angeben und neuen Wert zuweisen
 - Beispiel: `alien["neuer key"] = "neuer Wert"`
- Mit der `.update()`-Funktion
 - Dabei wird in { } der neue key und der neue Wert übergeben
 - Beispiel: `alien.update({"neuer key": "neuer Wert"})`

Elemente löschen

- Mit `.pop()` kann man ein Element nach einem bestimmten *key* entfernen
 - Wird kein *key* angegeben, so wirft die Funktionen einen Fehler
 - Das *value* zu dem *key* wird parallel zum entfernen nochmal ausgegeben
 - Beispiel: `aliens.pop("key 0")` → "Element 0"
- Mit `.popitem()` kann das letzte Element aus dem Dictionary entfernt werden
 - Beispiel: `aliens.popitem()` → Element n
- Mit `del` kann ebenfalls ein bestimmtes Element mittels *key* oder die gesamte Liste an sich löschen
 - Beispiele:
 - `del aliens`
 - `del aliens["key 0"]`

Weitere Nice-to-know-Methoden

- Mit `.clear()` kann man alle Elemente aus dem Dictionary entfernen
 - Beispiel: `aliens.clear()`
- Die Methode `.items()` gibt eine Liste mit den key-value-Paaren (als Tuple) von einem Dictionary zurück
 - Beispiel: `aliens.items()`
→ `[("key 0", "Element 0"), ("key 1", "Element 1"), ..., ("key n", "Element n")]`

Übungen

