

Python Dictionaries (dict)

(C) 2020 T.Birnthaler OSTC GmbH

Doku --> <http://docs.python.org/3/library/stdtypes.html#dict>

Dictionaries (dict) (auch Map, Hash, Zuordnung oder Assoziatives Array genannt) speichern Paare (Key, Value) von Werten und damit eine "Beziehung" zwischen zwei Datenwerten (Key --> Value).

\* Keys müssen IMMUTABLE Objekte (None, bool, int, float, complex, str, tuple) bzw. eigentlich "HASHABLE" Objekte sein.

\* Keys müssen UNIQUE sein (d.h. der gleiche Key darf nur 1x vorkommen).

\* Als Value sind beliebige Python-Objekte erlaubt.

\* Values dürfen NON-UNIQUE sein, d.h. der gleiche Value darf mehrfach vorkommen.

```
d1 = {} # Leeres Dictionary
d2 = {"abc": 123, True: 4.0, "efg": False}
d3 = {"a": 1, "b": 2, "c": 3}
```

Die Zugriffsgeschwindigkeit auf einen bestimmten Key (und damit seinen Value) ist unabhängig von der Dictionary-Größe immer gleich schnell ("RANDOM ACCESS").

Die Zugriffsgeschwindigkeit auf einen bestimmten Value ist abhängig von der Dictionary-Größe und immer langsam (erfordert das Durchsuchen des gesamten Dictionaries)

Folgende Operation werden von Python Dictionaries unterstützt (K ist ein Key = immutable Objekt, V und DFLT sind Values = beliebige Objekte, D und D2 sind Dictionaries, ITBL ist ein Iterable):

len(D)	Anzahl Paare
K in D K not in D	Ist Key enthalten? Ist Key nicht enthalten?
V = D[K] V = D.get(K, DFLT)	Wert von Key holen (oder KeyError falls Key fehlt) Wert von Key holen (oder DFLT)
D[K] = V D.setdefault(K, DFLT)	Wert zu Key setzen (oder ersetzen) Wert von Key setzen (falls vorhanden)
del D[K]	Key + Wert löschen (oder KeyError falls Key fehlt)
L = list(D) L = D.keys() L = D.values() L = D.items() L = iter(D) L = reversed(D)	Liste aller Keys (unsortiert) Liste aller Keys (unsortiert) Liste alle Werte (unsortiert) Liste aller Paare (unsortiert) Abkürzung für iter(D.keys()) Abkürzung für reversed(D.keys())
V = D.pop(K, DFLT) (K,V) = D.popitem()	Wert von Key (KeyError oder DFLT) + Key löschen Paar holen + entfernen (KeyError falls D leer)
D = fromkeys(ITBL, V) D.update(D2) D.clear() D2 = D.copy()	Dict mit Keys aus ITBL + Value V (None) erzeugen Paare aus D2 hinzufügen (vorhandene überschreiben) Dictionary leeren (alle Keys + Values entfernen) Shallow/Flache Kopie

HINWEIS: Seit Python 3.7 entspricht die Keyreihenfolge im Dictionary der Einfügereihenfolge, vorher war die Reihenfolge der Keys zufällig und konnte sich jederzeit ändern.