

Python Dictionaries (dict)

(C) 2020-2021 T.Birnthalder 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 bzw. eigentlich HASHABLE Objekte sein (NoneType, bool, int, float, complex, str, tuple, bytes).

\* 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, (1,2): None} # 3 Paare
d3 = {"a": 1, "b": 2, "c": 1, "d": 2, "e": 3} # 5 Paare
```

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 (da sie das Durchsuchen des gesamten Dictionarys erfordert).

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

ACHTUNG: Nur lesende Operationen, das Dictionary D bleibt unverändert!

len(D)	Anzahl Keys, Values, Paare	
K in D K not in D	Ist Key enthalten? Ist Key nicht enthalten?	
V = D[K] V = D.get(K, DFLT)	Value von Key holen (oder KeyError falls Key fehlt) Value von Key holen (oder DFLT)	
L = D.keys() L = D.values() L = D.items()	Liste aller Keys (unsortiert) Liste aller Values (unsortiert) Liste aller (Key,Value)-Paare (unsortiert)	
L = list(D) L = iter(D) L = reversed(D)	Abkürzung für D.keys() Abkürzung für iter(D.keys()) Abkürzung für reversed(D.keys())	3.8
D2 = D.copy()	Shallow/Flache Kopie	
E = D   D2	D vereinigt mit D2, D2-Val überschreibt D bei gl. Key	3.9

ACHTUNG: Schreibende Operationen, das Dictionary D wird verändert (bleibt aber das gleiche Objekt)!

D[K] = V D.setdefault(K, DFLT)	Value zu Key setzen (oder ersetzen) Value zu Key auf DFLT setzen (falls nicht vorhanden) + Value von Key zurückgeben	
del D[K] V = D.pop(K, DFLT) (K,V) = D.popitem()	Key + Value löschen (oder KeyError falls Key fehlt) Value von Key holen (KeyError/DFLT) + Key löschen Paar holen + entfernen (KeyError falls D leer)	
D = fromkeys(ITBL, V) D.update(D2) D.clear()	Dict mit Keys aus ITBL + Value V (None) erzeugen Paare aus D2 hinzufügen (vorhandene überschreiben) Dictionary leeren (alle Keys + Values entfernen)	
D  = D2	D2 zu D hinzufügen, D2-Val überschreibt D bei gl. Key	3.9

HINWEIS: Bei Zugriff auf ein Dictionary mit einem nicht vorhanden Key wird die Exception "KeyError" ausgelöst.

HINWEIS: Seit Python 3.7 entspricht die Key-Reihenfolge im Dictionary der Einfüge-Reihenfolge und bleibt bei Operationen auf dem Dictionary erhalten. Vorher war die Reihenfolge der Keys ZUFÄLLIG und konnte sich jederzeit ändern.