

Dictionaries

Python'da dictler sırasız item listesidir. Built-in methodlarıyla değiştirilebilir(mutable). Dictler değerlerin anahtar kelimelerinin haritasını oluşturmak için kullanılır.

4.1. DICT OLUSTURMAK & INDEXLEMEK

Python'da dict oluşturmak için {} kullanırız, onu indexlemek için de [] kullanılır. Anahtar kelimeler ile değerleri ayırmak için : kullanılır.

```
mdict = {} # Boş dict
d = {
    <key>: <value>,
    <key>: <value>,
    <key>: <value>
}
mdict = {"key1": "value1", "key2": "value2"} # tek satırda da tanımlayabiliriz
print(mdict["key1"]) # value1 (Index diğer tiplerin aksine anahtar kelime ile yapılır.)
print(mdict) # {'key1': 'value1', 'key2': 'value2'}
print(type(mdict)) # <class 'dict'>
d = {0: 'a', 1: 'b', 2: 'c', 3: 'd'} # anahtar kelime olarak integer da kullanılabilir
dict() fonksiyonu ile dict oluşturmak
dict() fonksiyonunun içinde key,value pairlerinden oluşan bir dizi(sequence) olmalıdır.
mdict2 = dict([("key1", "value1"), ("key2", "value2")])
# [] gerekli, key ve value arasına virgül ile tanımlanır
print(mdict2) # {'key1': 'value1', 'key2': 'value2'}
```

Dict comprehension ile dict oluşturmak

```
mdict = {x*x: x for x in range(8)} # {0: 0, 1: 1, 4: 2, 9: 3, 16: 4, 25: 5, 36: 6, 49: 7}
```

Yeni eleman tanımlamak

Dict içinde key olarak kullanılmayan bir değeri [] içine koyup ona bir değer atarız.

```
mdict = {'a': "val1", 'b': "val2"}
mdict['c'] = "val3" # yeni eleman eklemek
print(mdict) # {'a': 'val1', 'b': 'val2', 'c': 'val3'}
```

4.2. DICT ELEMANLARINI DEĞİŞTİRMEK

Anahtar kelimelerle indexlediğimiz değeri değiştirebiliriz(yeni değer atamak).

```
video = {"baslik": "Python Eğitim Serisi", "konu": "Setler"}
video["konu"] = "Dictler"
print(video) # {'baslik': 'Python Eğitim Serisi', 'konu': 'Dictler'}
```

4.3. DICT İCİNDE ELEMAN ARAMA

```
mdict = {'a': "val1", 'b': "val2"}
# sadece anahtar kelimelerin varlığını kontrol edebiliriz
'a' in mdict # True
'c' not in mdict # True

if 'b' in mdict:
    print(mdict['b']) # val2
```

`del dict_ismi[key]` methodu o anahtar kelimeyi ve onun gösterdiği değeri siler

`len(dict_ismi)` fonksiyonu key-value çiftlerinin sayısını döndürür.

`dict_ismi.get(key)` methodu o anahtar kelimenin gösterdiği değeri döndürür, o anahtar kelime yoksa None döndürür.

`dict_ismi.pop(key)` methodu o anahtar kelimeyi ve onun gösterdiği değeri siler ve silinen değeri döndürür. (Yani bir key-value çiftini silerken aynı zamanda value'yu return eder.)

`dict_ismi.popitem()` methodu herhangi bir(random) key-value çiftini siler ve aynı zamanda o çifti döndürür.

`dict_ismi.clear()` methodu tüm dict'i siler.

`dict_ismi.copy()` methodu kopya bir dict döndürür. Kullanım ex. `newDict = oldDict.copy()`

`any(dict_ismi)` & `all(dict_ismi)` tuple'lardaki gibi davranır fakat sadece key'lere bakar.

4.4. DICT LİSTELERİ & NESTED DICT

2d bir liste gibi düşünebiliriz, dictleri tek eleman olarak görüp ilk [] içine ona göre index veririz, ikinci [] içine de dict indexi yani keyword yazarız.

```
myList = [
    {
        'foo':12,
        'bar':14
    },
    {
        'moo':52,
        'car':641
    },
    {
        'doo':6,
        'tar':84
    }
]
```

Diagram illustrating the structure of `myList`:

- `myList[0]` points to the first dictionary: `{'foo':12, 'bar':14}`
- `myList[1]` points to the second dictionary: `{'moo':52, 'car':641}`
- `myList[2]` points to the third dictionary: `{'doo':6, 'tar':84}`

Nested dictionary de aynı şekilde işliyor, ilk olarak dışardan bakıp {} içindekileri tek eleman olarak düşünüp ilk indexi ona göre yapıyoruz, daha sonra tek eleman olarak düşündüğümüz {} içindeki elemanların indexine bakıyoruz.

```
myDict = {'Key1': {'Key2': 'Val1'},
          'Key3': {'Key4': 'Val2'}}
```

Diagram illustrating the structure of `myDict`:

- `myDict['Key1']` points to `{'Key2': 'Val1'}`
- `myDict['Key1']['Key2']` points to `'Val1'`
- `myDict['Key3']` points to `{'Key4': 'Val2'}`
- `myDict['Key3']['Key4']` points to `'Val2'`

`sorted(dict_ismi)` fonksiyonu anahtar kelimelerin sıralı listesini döndürür.

`dict_ismi.values()` methodu dict'in value listesini syntax içinde döndürür.

Print yaptığımızda çıktısı şöyle olur -> `dict_values(['val1', 'val2'])`

`dict_ismi.keys()` methodu dict'in key listesini syntax içinde döndürür.

Print yaptığımızda çıktısı şöyle olur -> `dict_keys(['key1', 'key2'])`

`dict_ismi.items()` methodu dict'i syntax içinde döndürür.

Print yaptığımızda çıktısı şöyle olur -> `dict_items([('key1', 'val1'), ('key2', 'val2')])`

`dict_ismi = dict.fromkeys(seq, val)` veya `dict_ismi = {}.fromkeys(seq, val)` methodu verilen bir sequence(ex. set) ile o sequence elemanlarını key olarak kullanan bir dict oluşturur. val değeri ise her anahtar kelimenin aynı değeri göstermesini sağlamak için kullanılır, boş bırakılırsa (sadece seq yazılırsa) her key'in göstereceği değer `None` olacaktır. val olarak liste vb. de verilebilir.

`dict1.update(dict2)` methodu dict2'de olup da dict1'de olmayan key-value çiftlerini dict1'e ekler. Eğer aynı key fakat farklı value varsa dict2'deki değeri kullanır.

`dict_ismi.setdefault(key, default_value)` methodu parametre olarak verilen key dictte varsa onun değerini döndürür, key dictte yoksa parametre olarak verilen key-value çiftinden dict elemanı oluşturur ayrıca parametre olarak verilen değeri döndürür. (parametrelerden `default_value` opsiyoneldir)

`str(dict_ismi)` fonksiyonu dicti string olarak döndürür.

EXTRA KAYNAKLAR:

(EN) PYTHON DICTS:

1. <https://data-flair.training/blogs/python-dictionary/>
2. <https://www.programiz.com/python-programming/dictionary>

(FR) LES DICTIONNAIRES:

1. <https://www.tresfacile.net/les-dictionnaires-en-langage-python/>

(EN) SORTED & STR FUNCTIONS:

1. <https://www.programiz.com/python-programming/methods/built-in/sorted>
2. <https://appdividend.com/2019/11/13/how-to-convert-python-dictionary-to-string-example/>

(EN) COPY & VALUES & KEYS & ITEMS & FROMKEYS & UPDATE & SETDEFAULT METHODS:

1. <https://www.geeksforgeeks.org/python-dictionary-copy/>
2. <https://www.geeksforgeeks.org/python-dictionary-values/>
3. <https://www.geeksforgeeks.org/python-dictionary-keys-method/>
4. <https://www.geeksforgeeks.org/python-dictionary-items-method/>
5. <https://www.geeksforgeeks.org/python-dictionary-fromkeys-method/>
6. <https://www.geeksforgeeks.org/python-dictionary-update-method/>
7. <https://www.geeksforgeeks.org/python-dictionary-setdefault-method/>