Sözlükler

Sözlükler veya İngilizce ismiyle dictionaryler aynı gerçek hayattaki sözlükler gibi davranan bir veritipidir. Bu veritipi, şimdiye kadar gördüğümüz tüm veritiplerinden yapısı gereği farklıdır. Sözlüğün içindeki her bir eleman indeks ile değil, anahtar (key), değer (value) olarak tutulur. Bu anlamda gerçek hayattaki sözlüklere oldukça benzerler. Örneğin, elimize bir ingilizce-Türkçe sözlük alıp **freedom** kelimesini(key ya da anahtar) aradığımız zaman karşılık değer **özgürlük** (değer ya da value) olarak karşımıza çıkar. Sözlükleri de bu şekilde düşünebiliriz.

Şimdi isterseniz bir sözlük oluşturarak konumuza başlayalım.

Sözlük Oluşturmak

In [1]:
```
# Süslü Parantez ve iki nokta (:) ile anahtar değerlerimizi yerleştirelim.
sözlük1 = {"sıfır":0,"bir":1,"iki":2,"üç":3}
```

In [2]:
```
sözlük1
```

Out[2]:
```
{'bir': 1, 'iki': 2, 'sıfır': 0, 'üç': 3}
```

In [3]:
```
# Boş bir sözlük
sözlük2 = {}
```

In [4]:
```
# Boş bir sözlük - dict() ile
sözlük2 = dict()
```

In [5]:
```
sözlük2
```

Out[5]:
```
{}
```

Sözlük Değerlerine Erişmek ve Sözlüğe Değer Eklemek

Sözlük veritipinin gerçek hayattaki sözlüklere çok benzediğini söylemiştik. Öyleyse, bir değeri (value) elde etmek için, indeksleri değil anahtarları (key) kullanacağız.

In [6]:
```
sözlük1
```

Out[6]:
```
{'bir': 1, 'iki': 2, 'sıfır': 0, 'üç': 3}
```

In [7]:
```
# "bir" anahtarına karşılık gelen değeri buluyoruz.
sözlük1["bir"]
```

Out[7]:
```
1
```

In [8]:
```
# "iki" anahtarına karşılık gelen değeri buluyoruz.
sözlük1["iki"]
```

Out[8]:
```
2
```

In [9]:
```
# Olmayan bir anahtar
sözlük1["beş"]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-9-10d9d9010d5b> in <module>()
      1 # Olmayan bir anahtar
----> 2 sözlük1["beş"]

KeyError: 'beş'
```

In [19]:
```
a = {"bir" : [1,2,3,4], "iki": [[1,2],[3,4],[5,6]],"üç" : 15}
```

In [20]:
```
# "iki" anahtarının değeri
a["iki"]
```

Out[20]:
```
[[1, 2], [3, 4], [5, 6]]
```

In [21]:
```
# İçiçe listeleri biliyoruz.
a["iki"][1][1]
```

Out[21]:
```
4
```

In [22]:
```
a["üç"]
```

Out[22]:
```
15
```

In [23]:
```
a["üç"] += 5
a["üç"]
```

Out[23]:
```
20
```

In [24]:
```
a
```

Out[24]:
```
{'bir': [1, 2, 3, 4], 'iki': [[1, 2], [3, 4], [5, 6]], 'üç': 20}
```

Bir sözlüğe **dinamik** olarak da eleman ekleyebiliriz.

In [32]:
```
# Sözlük oluşturalım.
a = {"bir":1,"iki":2,"üç":3}
```

In [33]:
```
a["dört"] = 4
a
```

Out[33]:
```
{'bir': 1, 'dört': 4, 'iki': 2, 'üç': 3}
```

Dikkat ederseniz yeni eklediğimiz anahtar ve değer sözlüğün sonuna eklenmedi. Burada sözlüklerin bir özelliğini daha görüyoruz. Sözlükler diğer veritiplerinden farklı olarak sıralı olmayan bir veritipidir.

İç içe Sözlükler

Tıpkı listeler gibi, iç içe sözlükler de oluşturulabilir.

In [34]:
```
# İç içe sözlük

a = {"sayılar":{"bir":1,"iki":2,"üç":3},"meyveler":{"kiraz":"yaz","portakal":"kış","erik":"yaz"}}
```

In [35]:
```
a["sayılar"]["bir"]
```

Out[35]:
```
1
```

In [36]:
```
a["meyveler"]["kiraz"]
```

Out[36]:
```
'yaz'
```

Temel Sözlük Metodları

In [2]:
```
yeni_sözlük = {"bir":1,"iki":2,"üç":3}
```

In [3]:
```
# values() metodu sözlüğün değerlerini(value) bir liste olarak döner.
yeni_sözlük.values()
```

Out[3]:
```
dict_values([1, 2, 3])
```

In [4]:
```
# keys() metodu sözlüğün anahtarlarını(key) bir liste olarak döner.
yeni_sözlük.keys()
```

Out[4]:
```
dict_keys(['bir', 'iki', 'üç'])
```

In [5]:
```
# items() metodu sözlüğün anahtar ve değerlerini bir liste içinde demet olarak döner.
yeni_sözlük.items()
```

Out[5]:
```
dict_items([('bir', 1), ('iki', 2), ('üç', 3)])
```

Bu konuda sözlüklerin yapısını biraz anladıysak iyi yol katetmişiz demektir. Zaten sözlükleri ilerde daha derinlemesine göreceğimiz için konumuzu burada bitirebiliriz.

In [ ]:
```

```