

# Sözlükler

Prof.Dr. Bahadır AKTUĞ  
JFM212 Python ile Mühendislik Uygulamaları

*\*Kaynakça bölümünde verilen kaynaklardan derlenmiştir.*

# Sözlükler (Dictionaries)

---

- ▶ Python'un sağladığı güçlü ve esnek programlama olanaklarının en önemlilerinden biri de sözlüklerdir.
- ▶ Sözlükler listeler gibi "değiştirilebilir" türde veri tipleridir.
- ▶ Sözlükler genişletilebilir veya daraltılabilir.
- ▶ Sözlükler içerisinde listeler bulunabildiği gibi, listeler içerisinde de sözlükler bulunabilir.
- ▶ Sözlüklerin en önemli farkı:
  - ▶ Ardışık veri tipi olmamalarıdır.
  - ▶ Sözlük elemanlarına "indisler" ile değil, "anahtarlar" ile erişilir.
  - ▶ Genel olarak bilgisayar programlamada "associative arrays" adı verilen kavramın, Python'da uygulamasıdır. Bu tür veri tiplerinde elemanlar (anahtar,değer) yapısındadır.
  - ▶ Diğer dillerde farklı adlar ile de anılır (hash table vb.)
  - ▶ Birden fazla aynı anahtar bulunamaz.

# Sözlükler (Dictionaries)

---

```
▶ iller = {"Ankara": "06", "Adana": "01", "Samsun": "55"}
```

```
▶ notlar = {"Ahmet KILIÇ": 76, "Veli Demir": 64, "Kazım Gök": 87}
```

```
>>> iller = {"Ankara": "06", "Adana": "01", "Samsun": "55"}
```

```
>>> iller["Ankara"]
```

```
'06'
```

```
>>> iller[0]
```



Ardışık veri tipi değil!  
İndis ile erişilemez!

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
KeyError: 0
```

```
>>> iller["Bursa"] = 16
```



Yeni bir eleman, anahtar ve  
değer ikilisi ile eklenebilir

```
>>> print(iller)
```

```
{'Ankara': '06', 'Samsun': '55', 'Bursa': 16, 'Adana': '01'}
```

# Sözlükler (Dictionaries)

---

```
>>> iller["Kocaeli"]
```



Tanımsız anahtar!

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

KeyError: 'K'

```
>>> ilçeler = {}
```



Boş sözlük tanımlaması!

```
>>> print(ilçeler)
```

```
{}
```

```
>>> sozluk = {"kırmızı":"red","yeşil":"green","mavi":"blue"}
```

```
>>> sozluk["kırmızı"]
```

```
'red'
```

```
>>> sozluk["mavi"]
```

```
'blue'
```

# Sözlükler (Dictionaries)

```
>>> sozluk = {[1,2,3]:"JFM212"}
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: unhashable type: 'list'
```

Değiştirilebilir  
(mutable) tipler  
anahtar olamaz!

```
>>> sozluk = {(1,2,3):"JFM212"}
```

```
>>> sozluk[(1,2,3)]
```

```
'JFM212'
```

Demetler anahtar olabilir.

```
>>> sozluk = {1:2,3:5,9:10}
```

```
>>> sozluk = {"1":"2", "3":"5", "9":"10"}
```

```
>>> sozluk = {"1":2, "3":5, "9":10}
```

```
>>> sozluk = {1:"2", 3:" 5" 9:"10"}
```

Hiçbiri aynı sözlük  
tanımlaması değil!

# Sözlükler (Dictionaries)

---

## Sözlükler için kullanılacak komutlar:

Operatör	Açıklama
<code>len(d)</code>	Toplam (anahtar, değer) çifti sayısı
<code>del d[k]</code>	"k" isimli anahtarı ve bununla ilişkili değeri siler
<code>k in d</code>	"d" isimli sözlükte "k" anahtarı olup olmadığını kontrol eder varsa "True" döndürür
<code>k not in d</code>	"d" isimli sözlükte "k" anahtarı olup olmadığını kontrol eder yoksa "True" döndürür

# Sözlükler (Dictionaries)

---

## Sözlükler için kullanılacak komutlar:

### pop():

"Sözlükler", "listeler" den farklı olarak sıralı olmadıklarından "pop" komutu en sondaki yerine seçilen anahtarın değerini döndürür ve ilgili (anahtar,değer) ikilisini siler.

```
>>> baskentler= {"Avusturya":"Viyana", "Almanya":"Berlin",  
"Hollanda":"Amsterdam"}
```

```
>>> baskent = baskentler.pop ("Avusturya")
```

```
>>> print(baskent)
```

```
'Viyana'
```

```
>>> print(baskentler)
```

```
{'Almanya': 'Berlin', 'Hollanda': 'Amsterdam'}
```

# Sözlükler (Dictionaries)

---

## Sözlükler için kullanılacak komutlar:

### **popitem():**

Bu komut "pop" komutundan farklı olarak parametre almaz ve sözlükten rasgele bir (anahtar,değer) ikilisini döndürülür. Döndürülen ikili sözlükten silinir.

```
>>> baskentler= {"Avusturya":"Viyana", "Almanya":"Berlin",  
"Hollanda":"Amsterdam"}
```

```
>>> baskentler.popitem()
```

```
('Almanya', 'Berlin')
```

```
>>> print(baskentler)
```

```
{'Hollanda': 'Amsterdam', 'Avusturya': 'Viyana'}baskent = baskentler.pop  
("Avusturya")
```



# Sözlükler (Dictionaries)

---

## Mevcut olmayan anahtarlar ile çalışma:

Sözlükte olmayan bir anahtar ile değer döndürmek istendiğinde hata mesajı ile karşılaşılır. Bunu önlemek için öncelikle bu anahtarın olup olmadığı kontrol edilmelidir.

```
>>> locations = {"Toronto" : "Ontario", "Vancouver": "British Columbia"}
```

```
>>> locations["Ottawa"]
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

KeyError: 'Ottawa'

### 1. Yol

```
>>> if "Ottawa" in locations: print(locations["Ottawa"])
```

### 2. Yol

```
>>> locations.get("Ottawa")
```

# Sözlükler (Dictionaries)

---

## Sözlüklerin kopyalanması:

Sözlükler için "copy" komutu ile "sığ kopyalama" yapılabilir.

```
>>> sozcukler = { "cat": "Katze", "house": "Haus" }
```

```
>>> w = sozcukler.copy()
```

```
>>> sozcukler["cat"] = "chat"
```

```
>>> print(w)
```

```
{'house': 'Haus', 'cat': 'Katze'}
```

```
>>> print(sozcukler)
```

```
{'house': 'Haus', 'cat': 'chat'}
```

# Sözlükler (Dictionaries)

---

## Sözlüklerin içeriğinin silinmesi (clear):

Sözlüğün kendisinin değil, içeriğinin silindiğine dikkat ediniz.

```
>>> sozcukler = {"cat":"Katze", "house":"Haus"}
>>> sozcukler.clear()
>>> print(sozcukler)
{}
```

## Sözlüklerin güncellenmesi (update)

```
>>> og1= {"Ali ÇELİK":{"JFM211","JFM212"},"Zeynep TAŞ": {"JFM212","JFM230"}}
>>> og2= {"Ali ÇELİK":{"JFM230","JFM212"},"Zeynep TAŞ":
{"JFM211","JFM230"},"Can Demir":{"JFM230"}}
>>> og1.update(og2)
>>> {'Can Demir': {'JFM230'}, 'Ali ÇELİK': {'JFM230', 'JFM212'}, 'Zeynep TAŞ':
{'JFM211', 'JFM230'}}
```

# Sözlükler (Dictionaries)

---

## Sözlük elemanları üzerinde döngüler:

```
>>> d = {"a":123, "b":34, "c":304, "d":99}
>>> for key in d:
...     print(key)
...
b
c
a
D
```

```
>>> for key in d.keys():
...     print(key)
...
b
c
a
d
```

## Sözlük elemanları üzerinde döngüler:

```
>>> d = {"a":123, "b":34, "c":304, "d":99}
>>> for k in d.items():
...     print(k)
...
('b', 34)
('c', 304)
('a', 123)
('d', 99)
```

```
>>> for k in d.values():
...     print(k)
...
34
304
123
99
```

# Sözlükler (Dictionaries)

---

## Sözlüklerin listeye dönüştürülmesi:

```
>>> d = {"a":123,"b":547,"c":878}
```

```
>>> list(d)
```

```
['b', 'c', 'a']
```

```
>>> list(d.items())
```

```
[('b', 547), ('c', 878), ('a', 123)]
```

```
>>> list(d.keys())
```

```
['b', 'c', 'a']
```

```
>>> list(d.values())
```

```
[547, 878, 123]
```

# Sözlükler (Dictionaries)

---

## Listelerin sözlüklere dönüştürülmesi:

```
>>> sehirler = ["Bursa","Kayseri","Gaziantep","Konya","Urfa"]
>>> yemekler = ["İskender","Mantı","Lahmacun","Etli Ekmek","Kebap"]
>>> list(zip(sehirler,yemekler))
[('Bursa', 'İskender'), ('Kayseri', 'Mantı'), ('Gaziantep', 'Lahmacun'),
 ('Konya', 'Etli Ekmek'), ('Urfa', 'Kebap')]
>>> dict(list(zip(sehirler,yemekler)))
{'Gaziantep': 'Lahmacun', 'Urfa': 'Kebap', 'Kayseri': 'Mantı', 'Bursa':
 'İskender', 'Konya': 'Etli Ekmek'}
```

## VEYA

```
>>> dict(zip(sehirler,yemekler))
{'Gaziantep': 'Lahmacun', 'Urfa': 'Kebap', 'Kayseri': 'Mantı', 'Bursa':
 'İskender', 'Konya': 'Etli Ekmek'}
```

## ► Kaynakça

- 1 Wentworth, P., Elkner, J., Downey, A.B., Meyers, C. (2014). *How to Think Like a Computer Scientist: Learning with Python (3rd edition)*.
- 2 Pilgrim, M. (2014). *Dive into Python 3 by*. Free online version: [DiveIntoPython3.org](http://DiveIntoPython3.org) ISBN: 978-1430224150.
- 3 Summerfield, M. (2014) *Programming in Python 3 2nd ed (PIP3) :- Addison Wesley* ISBN: 0-321-68056-1.
- 4 Summerfield, M. (2014) *Programming in Python 3 2nd ed (PIP3) :- Addison Wesley* ISBN: 0-321-68056-1.
- 5 Jones E, Oliphant E, Peterson P, et al. *SciPy: Open Source Scientific Tools for Python, 2001-*, <http://www.scipy.org/>.
- 6 Millman, K.J., Aivazis, M. (2011). *Python for Scientists and Engineers, Computing in Science & Engineering*, 13, 9-12.
- 7 John D. Hunter (2007). *Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering*, 9, 90-95.
- 8 Travis E. Oliphant (2007). *Python for Scientific Computing, Computing in Science & Engineering*, 9, 10-20.
- 9 Goodrich, M.T., Tamassia, R., Goldwasser, M.H. (2013). *Data Structures and Algorithms in Python*, Wiley.
- 10 <http://www.diveintopython.net/>
- 11 <https://docs.python.org/3/tutorial/>
- 12 <http://www.python-course.eu>
- 13 <https://developers.google.com/edu/python/>
- 14 <http://learnpythonthehardway.org/book/>