

Nesneye Yönelimli Programlama – BLM 205

Hafta 4: Veri Koleksiyonları



Fenerbahçe Üniversitesi

Öğretim Elemanları

Öğretim Üyesi: Dr. Vecdi Emre Levent

Ofis: 311

Email: emre.levent@fbu.edu.tr

Asistan: Arş. Gör. Uğur Özbalkan

Ofis: 307

Email: ugur.ozbalkan@fbu.edu.tr

Asistan: Arş. Gör. Ecenur Alioğulları

Ofis: 307

Email: ecenur.aliogullari@fbu.edu.tr

Ders Planı

- Veri Koleksiyonları
 - Lists
 - Tuples
 - Sets
 - Dictionaries

Veri Koleksiyonları

Python dilinde verileri grup halinde (dizi) tutabilmek için build-in 4 farklı koleksiyon tipi bulunmaktadır.

Bunlar:

- List
- Tuple
- Set
- Dictionary

Veri Koleksiyonları

List

List'ler elemanları grup olarak tutan veri yapılarıdır.

- Yeni bir eleman ekleme
- Değiştirme
- Silme

İşlemleri yapılabilir.

Veri Koleksiyonları

Lists

Liste elemanlarına erişim

Örnek Kod Parçacığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]  
  
print(liste)  
  
print(liste[1])
```

Çıktı

```
['eleman 1', 'eleman 2', 'eleman 3']  
eleman2
```

Veri Koleksiyonları

Lists

Negatif indeks'ler, indeks negatif olması durumunda sondan geriye doğru ilerlemektedir.

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]
```

```
print(liste[-1])  
print(liste[-2])
```

Çıktı

```
eleman 3  
eleman 2
```

Veri Koleksiyonları

Lists

İndeks aralık belirleme, [N-M], N'den M'e kadar (M dahil değil) elemanları döndürür.

Örnek Kod Parçası

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]  
  
print(liste[0:2])
```

Çıktı

```
['eleman 1', 'eleman 2']
```


Veri Koleksiyonları

Lists

Başlangıçtan N'e kadar olan elemanları almak için [:N] syntax'i kullanılabilir.

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3", "eleman 4" , "eleman  
5"]  
  
print(liste[:3])
```

Çıktı

```
['eleman 1', 'eleman 2', 'eleman 3']
```

Veri Koleksiyonları

Lists

N. indeksten son elemana kadar olan elemanların alınması için [N:] syntax'i kullanılabilir.

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3", "eleman 4" , "eleman  
5"]  
  
print(liste[1:])
```

Çıktı

```
['eleman 2', 'eleman 3', 'eleman 4',  
'eleman 5']
```

Veri Koleksiyonları

Lists

Listedeki elemanın değerini değiştirmek için;

Örnek Kod Parçası

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]
```

```
liste[1] = "test"
```

```
print(liste)
```

Çıktı

```
['eleman 1', 'test', 'eleman 3']
```

Veri Koleksiyonları

Lists

Liste üzerinde döngü ile erişim

Örnek Kod Parçası

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]
```

```
for x in liste:  
    print(x)
```

Çıktı

```
eleman 1  
eleman 2  
eleman 3
```

Veri Koleksiyonları

Lists

Liste uzunluğunun alınması için len fonksiyonu kullanılabilir

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]  
  
print(len(liste))
```

Çıktı

3

Veri Koleksiyonları

Lists

Listeye yeni eleman eklemek için append fonksiyonu kullanılabilir.
Append fonksiyonu eklenecek elemanını, listenin sonuna ekler.

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]
```

```
liste.append("Yeni eleman")
```

```
print(liste)
```

Çıktı

```
['eleman 1', 'eleman 2', 'eleman 3', 'Yeni  
eleman']
```

Veri Koleksiyonları

Lists

Listeye istenen lokasyona yeni eleman eklemek için insert fonksiyonu kullanılabilir. Insert fonksiyonu eklenecek elemanı, istedeki istenen yere ekler.

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]
```

```
liste.insert(1, "Yeni eleman")
```

```
print(liste)
```

Çıktı

```
['eleman 1', 'Yeni eleman', 'eleman 2',  
'eleman 3']
```

Veri Koleksiyonları

Lists

Listedeki istenen elemanı silmek için, remove fonksiyonu verilen string ile eşleşen elemanı siler.

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]
```

```
liste.remove("eleman 2")
```

```
print(liste)
```

Çıktı

```
['eleman 1', 'eleman 3']
```


Veri Koleksiyonları

Lists

Listedeki istenen elemanı silmek için, pop fonksiyonu son elemanı siler.

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]
```

```
liste.pop()
```

```
print(liste)
```

Çıktı

```
['eleman 1', 'eleman 2']
```

Veri Koleksiyonları

Lists

Listedeki istenen elemanı silmek için, del kullanarak istenen indeksteki elemanı siler.

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]
```

```
del liste[1]
```

```
print(liste)
```

Çıktı

```
['eleman 1', 'eleman 3']
```

Veri Koleksiyonları

Lists

Listedeki istenen tüm elemanı silmek için, clear fonksiyonu kullanılabilir

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]
```

```
liste.clear()
```

```
print(liste)
```

Çıktı

```
[]
```

Veri Koleksiyonları

Lists

İki listeyi birleştirmek için,

Örnek Kod Parçasığı

```
liste = ["eleman 1", "eleman 2",  
"eleman 3"]  
liste2 = ["eleman X", "eleman Y",  
"eleman Z"]  
  
liste3 = liste + liste2  
print(liste3)
```

Çıktı

```
['eleman 1', 'eleman 2', 'eleman 3',  
'eleman X', 'eleman Y', 'eleman Z']
```

Veri Koleksiyonları

Lists

Listeyi sıralamak için, sort fonksiyonu kullanılabilir.

Örnek Kod Parçasığı

```
liste = ["Xyz", "Abc", "Def"]  
  
liste.sort()  
  
print(liste)
```

Çıktı

```
['Abc', 'Def', 'Xyz']
```

Veri Koleksiyonları

Lists

Listedeki elemanları ters sıralamak için, reverse fonksiyonu kullanılabilir.

Örnek Kod Parçasığı

```
liste = ["Xyz", "Abc", "Def"]  
  
liste.reverse()  
  
print(liste)
```

Çıktı

```
['Def', 'Abc', 'Xyz']
```

Veri Koleksiyonları

Tuple

Tuple, listelerdeki gibi elemanları grup halinde tutmaktadır. Ancak tuple oluşturulduktan sonra

- Yeni bir eleman ekleme
- Değiştirme
- Silme

İşlemleri yapılamaz. Oluşturma yapıldıktan sonra değişiklik yapılamaz bir yapıdadır.

Veri Koleksiyonları

Tuple

Tuple oluşturmak için parantezler arasında elemanlar yazılmalıdır.

Örnek Kod Parçasığı

```
tuple = ("Xyz", "Abc", "Def")  
print(tuple)
```

Çıktı

```
('Xyz', 'Abc', 'Def')
```


Veri Koleksiyonları

Tuple

Tuple daki elemanlara indeks'ler üzerinden erişim, listeler ile aynıdır.

Örnek Kod Parçasığı

```
tuple = ("Xyz", "Abc", "Def")  
print(tuple[0])
```

Çıktı

Xyz

Veri Koleksiyonları

Sets

Setler, sıralı olmayan, her elemanı eşsiz (unique) olan veri koleksiyonlarıdır.

Her bir eleman değiştirilemezdir ancak set'in kendisine yeni eleman eklenebilir ve silinebilir.

İndeks'leri yoktur. Yani set'in içindeki elemanlara indeks'ler ile erişilemez.

Genellikle küme işlemleri olan birleşim, kesişim gibi işlemler için kullanılırlar.

Veri Koleksiyonları

Sets

Set oluşturma örneği

Örnek Kod Parçası

```
set = {1, 2, 3}  
print(set)
```

```
set = {1.0, "Hello", (1, 2, 3)}
```

```
print(set)
```

Çıktı

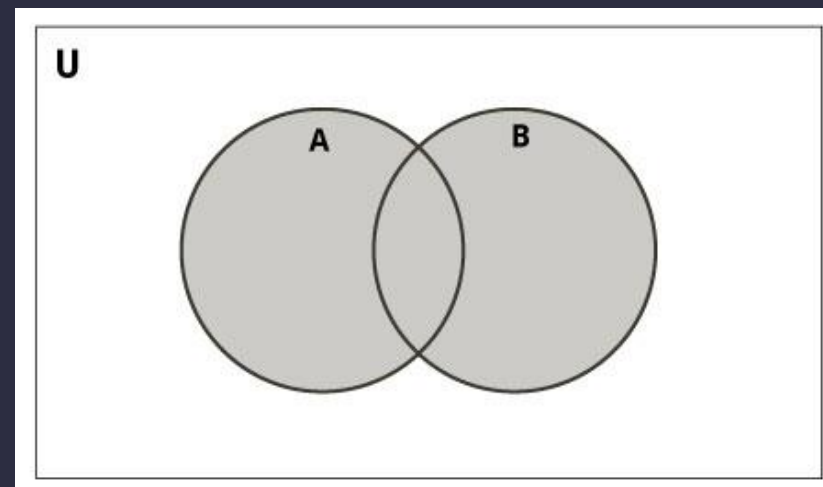
```
{1, 2, 3}
```

```
{1.0, 'Hello', (1, 2, 3)}
```

Veri Koleksiyonları

Sets

Birşelim işlemi | operatörü ile yapılmaktadır.



Örnek Kod Parçasığı

```
A = {1, 2, 3, 4, 5, 9}
```

```
B = {4, 5, 6, 7, 8}
```

```
print(A | B)
```

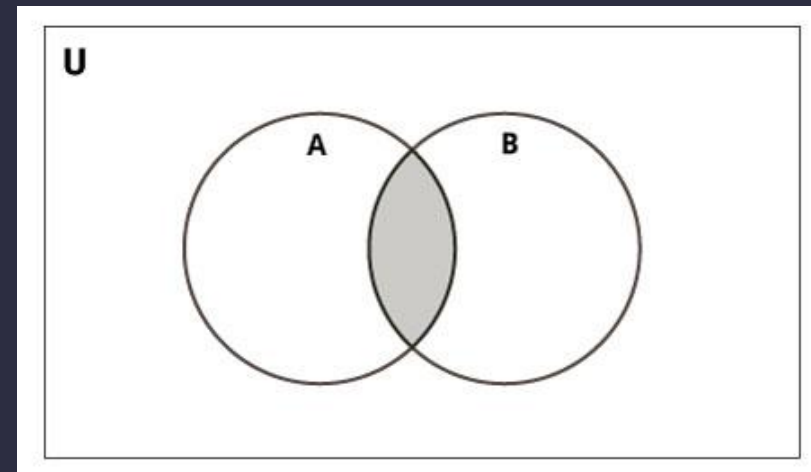
Çıktı

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

Veri Koleksiyonları

Sets

Kesişim işlemi | operatörü ile yapılmaktadır.



Örnek Kod Parçasığı

```
A = {1, 2, 3, 4, 5, 9}
```

```
B = {4, 5, 6, 7, 8}
```

```
print(A & B)
```

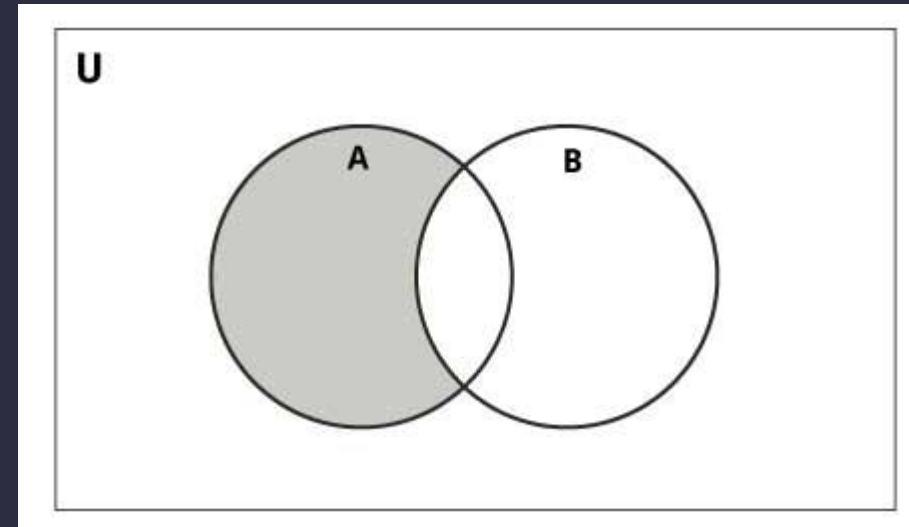
Çıktı

```
{4, 5}
```

Veri Koleksiyonları

Sets

Fark işlemi - operatörü ile yapılmaktadır.



Örnek Kod Parçasığı

```
A = {1, 2, 3, 4, 5, 9}
```

```
B = {4, 5, 6, 7, 8}
```

```
print(A - B)
```

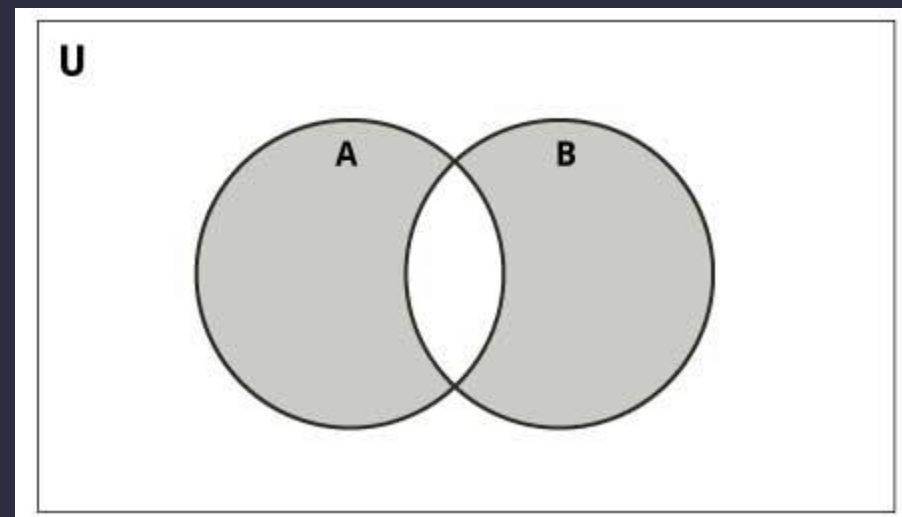
Çıktı

```
{1, 2, 3, 9}
```

Veri Koleksiyonları

Sets

Simetrik fark işlemi - operatörü ile yapılmaktadır.



Örnek Kod Parçasığı

```
A = {1, 2, 3, 4, 5, 9}
```

```
B = {4, 5, 6, 7, 8}
```

```
print(A ^ B)
```

Çıktı

```
{1, 2, 3, 6, 7, 8, 9}
```

Veri Koleksiyonları

Dictionary

Sözlükler, anahtar ve değeri olan verileri grup olarak tutmaya yarayan koleksiyonlardır.

Örnek Kod Parçasığı

```
sozluk = {  
    "Marka": "Ford",  
    "Model": "Mustang",  
    "Sene": 1964  
}  
print(sozluk)
```

Çıktı

```
{'Marka': 'Ford', 'Model': 'Mustang',  
'Sene': 1964}
```


Veri Koleksiyonları

Dictionary

Sözlüklerde elemana erişmek için anahtar kullanılır.

Örnek Kod Parçasığı

```
sozluk = {  
    "Marka": "Ford",  
    "Model": "Mustang",  
    "Sene": 1964  
}  
  
x = sozluk["Model"]  
  
print(x)
```

Çıktı

Mustang

Veri Koleksiyonları

Dictionary

Sözlüklerde eleman değerini değiştirme

Örnek Kod Parçasığı

```
sozluk = {  
    "Marka": "Ford",  
    "Model": "Mustang",  
    "Sene": 1964  
}  
  
sozluk["Sene"] = 2020  
  
print(sozluk["Sene"])
```

Çıktı

2020

Veri Koleksiyonları

Dictionary

- Listelerde olan özelliklerin çoğu sözlüklerde mevcuttur.
- Sözlüklerde bir anahtardan birden çok olamaz.