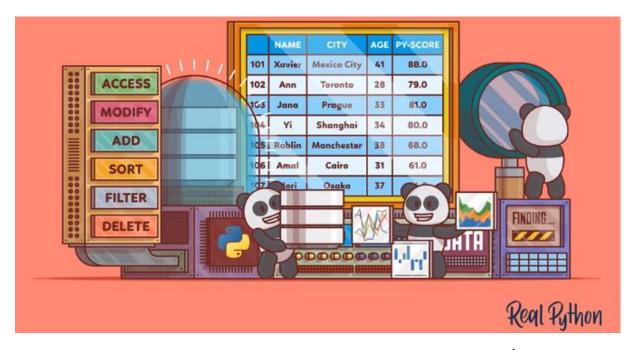
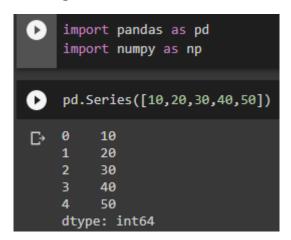
# PANDAS İLE TEMEL İŞLEMLER



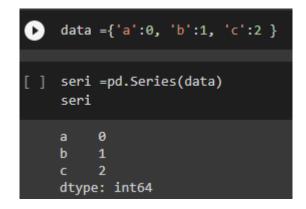
**Bir liste üzerinde Series oluşturma:** içerisine listemizi ekledik. İlk sütundaki sayılar Series'in index'idir. Diğer sütunda listemizin elemanlarını içerir.



Numpy arraylerinden farkı index bulundurmasıdır. Numpy array kullanarak Series oluşturma: index parametresiyle değerlerimizi değiştirdik.

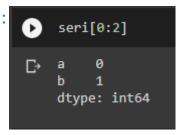
Dikkat edilmesi gereken nokta, data uzunluğu ile index uzunluğunun aynı olması.

Sözlük yapısını kullanarak Series oluşturma:



# Series yapısında verilerimize ulaşma:

İlk iki elemana ulaşma:



Aynı anda birden fazla index'e erişmek:

## Series yapısında verilerimizi güncelleme:

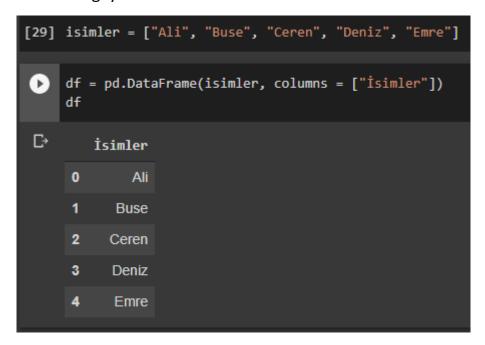
```
[20] data = np.array([10,20,30,40,50])
     seri = pd.Series(data, index = ['a','b','c','d','e'])
     seri
          10
     а
     b
         20
         30
     d
          40
          50
     dtype: int64
    seri['a']
     10
[22] seri['a'] = 90
     seri
          90
          20
          30
          40
          50
     dtype: int64
```

Series yapısında sum(toplama), max(en büyük değer), mean(ortalama), median(medyan) gibi fonksiyonları kullanma:



#### **DataFrame oluşturma:**

**Liste üzerinden DataFrame oluşturma:** columns argümanı değişkenimizi isimlendirmemizi sağlıyor.



#### Sözlük üzerinden DataFrame oluşturma:

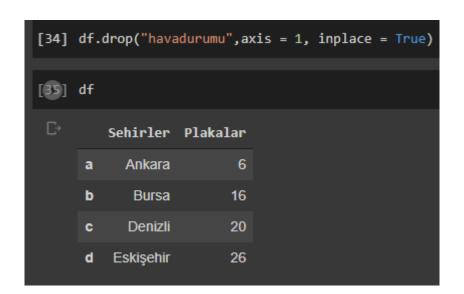
```
data = {'Sehirler': ['Ankara', 'Bursa', 'Denizli', 'Eskişehir'],
         'Plakalar':[6,16,20,26]}
data
{'Plakalar': [6, 16, 20, 26],
'Sehirler': ['Ankara', 'Bursa', 'Denizli', 'Eskişehir']}
df = pd.DataFrame(data, index = ['a', 'b', 'c', 'd'])
df
    Sehirler Plakalar
       Ankara
                       6
 a
 b
        Bursa
                      16
       Denizli
                      20
 C
     Eskişehir
                      26
 d
```

### DataFrame üzerinde ekleme, silme ve seçme işlemleri:

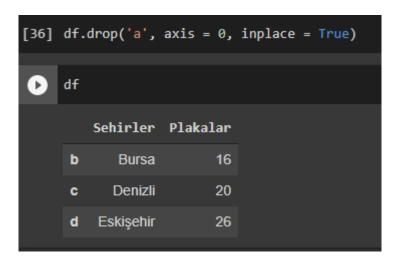
Ekleme yapmak için; df eklemek istediğimiz değişkenin adı ve ardından değişkenin değerlerini giriyoruz. "havadurumu" kolonu eklendi.

| 0 | <pre>df["havadurumu"] = pd.Series( [33,26,40,36], index=['a', 'b','c','d']) df</pre> |           |          |            |  |  |
|---|--|-----------|----------|------------|--|--|
| ₽ |  | Sehirler  | Plakalar | havadurumu |  |  |
|   | a  | Ankara    | 6        | 33         |  |  |
|   | b  | Bursa     | 16       | 26         |  |  |
|   | С  | Denizli   | 20       | 40         |  |  |
|   | d  | Eskişehir | 26       | 36         |  |  |

Silme işlemi için; drop() kullanıyoruz. axis=1 sütun olarak silmemizi sağlıyor. Argüman olarak inplace=True eklersek ana yapısından siliniyor.



Satır silmek: index ismini vererek silebiliriz.



#### Elemanlara erişme işlemleri:

LOC: Tanımlamış olduğumuz şekilde kullanmamızı sağlar. Yani indexlememiz a, b ve c'den oluşuyorsa onların isimlerini kullanarak seçim yapmamızı sağlar. Aynı işlemi sütun bazlıda gerçekleştirebiliriz.



*ILOC:* Normalde kullandığımız indexleme mantığı ile seçim yapar. Yani belirttiğimiz değerden başlayarak belirttiğimiz uzunluğa kadar olan sayılardan oluşan bir indexleme kullanır.



## Bazı fonksiyonların DataFrame üzerinde çalışması:

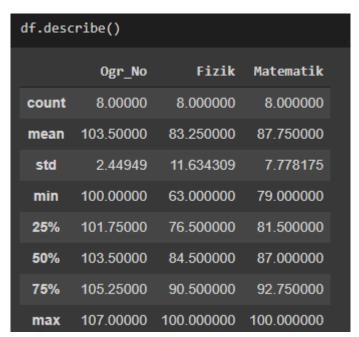
```
[42] data={'isimler': ["Ali", "Buse", "Ceren", "Deniz", "Emre", "Furkan", "Gizem", "Hakan"], 'Ogr_No': pd.Series([100, 101, 102, 103, 104, 105, 106,107]),
             'Fizik': pd.Series([90,88,75,92,100,81,77,63]),
             'Matematik': pd.Series([100,92,82,79,95,91,83,80])}
[43] df = pd.DataFrame(data)
0
 ₽
          isimler Ogr_No Fizik Matematik
               Ali
                                             100
       0
                        100
             Buse
                        101
                                 88
                                              92
            Ceren
                                              82
                                              79
       3
            Deniz
                        103
                                 92
                                              95
       4
             Emre
       5
            Furkan
                        105
                                              91
            Gizem
                                              80
                                 63
            Hakan
```

| [46] | df['Fizik'].sum()      |
|------|------------------------|
|      | 666                    |
| [48] | df['Matematik'].mean() |
|      | 87.75                  |

```
[49] df['Fizik'].min()
63

[50] df['Matematik'].max()
100
```

*describe ()* ile hepsini bir arada hesaplayabiliyoruz. Nümerik kolonların her biri için bu bilgileri hesaplayabiliriz.



### Koşullu eleman işlemleri:

# KAYNAKÇA

Bilgeiş "Herkes için Yapay Zekâ I" eğitimi.



