

```
In [1]: myList = list()
        type(myList)

Out[1]: list

In [2]: # myList değişkeni list sınıfından bir obje
```

instance && attribute

```
In [10]: class Meslekler():
        def __init__(self,dil,araç,sure):
            print("init fonksiyonu çalıştı")
            self.dil = dil
            self.araç = araç
            self.sure = sure

In [11]: yazilim = Meslekler("python","Jupyter Notebook",4)
        init fonksiyonu çalıştı

In [12]: yazilim.araç

Out[12]: 'Jupyter Notebook'

In [13]: yazilim.araç = "visual studio code"

In [14]: yazilim.araç

Out[14]: 'visual studio code'

In [16]: makina = Meslekler("Teknik Resim","solid edge",4)
        init fonksiyonu çalıştı

In [17]: makina.araç

Out[17]: 'solid edge'

In [18]: makina.sure

Out[18]: '4'

In [19]: class Meslekler():
        sıfat = "Mühendis"

        def __init__(self,dil,araç,sure):
            print("init fonksiyonu çalıştı")
            self.dil = dil
            self.araç = araç
            self.sure = sure

        def ortakOzellik(self):
            print("Ben mühendisim")

In [20]: makina = Meslekler("Teknik Resim","solid edge",4)
        init fonksiyonu çalıştı

In [21]: makina.ortakOzellik

Out[21]: <bound method Meslekler.ortakOzellik of <_main__.Meslekler object at 0x0000019C7917FCD0>>

In [22]: makina.ortakOzellik()

Ben mühendisim

In [23]: class Meslekler():
        sıfat = "Mühendis"

        def __init__(self,dil,araç,sure):
            print("init fonksiyonu çalıştı")
            self.dil = dil
            self.araç = araç
            self.sure = sure

        def ortakOzellik(self):
            print(f"Ben {self.araç} kullanıyorum")

In [24]: makina = Meslekler("Teknik Resim","solid edge",4)
        init fonksiyonu çalıştı

In [25]: makina.ortakOzellik()

Ben solid edge kullanıyorum
```

Default Value

```
In [31]: class Ulke():
        def __init__(self,isim="Dünya"):
            self.isim = isim

In [32]: ulkem = Ulke("Türkiye")

In [33]: ulkem.isim

Out[33]: 'Türkiye'

In [34]: ulkem2 = Ulke()

In [35]: ulkem2.isim

Out[35]: 'Dünya'

In [96]: class Nozzle():
        baskiSicakligi = 210

        def __init__(self,sicaklik):
            self.sicaklik = sicaklik
            self.artacakSicaklik = Nozzle.baskiSicakligi - self.sicaklik

        def gerekliSicaklik(self):
            print(f"sicaklık {self.sicaklik} derece {self.baskiSicakligi-self.sicaklik} derece a

In [97]: nozzle1 = Nozzle(50)

In [98]: nozzle1.sicaklik

Out[98]: 50

In [99]: nozzle1.gerekliSicaklik()

sicaklık 50 derece 160 derece artmalı

In [100]: nozzle1.artacakSicaklik

Out[100]: 160
```

inheritance

```
In [118]: class Yazilim():
        def __init__(self):
            print("yazilim sınıfı init fonksiyonu çalıştı")

        def metodi(self):
            print("yazilim sınıfı metod 1")
        def metod2(self):
            print("yazilim sınıfı metod 2")
        def metod3(self):
            print("yazilim sınıfı metod 3")

In [160]: class Mekatronik(Yazilim):
        def __init__(self):
            Yazilim.__init__(self)
            print("mekatronik sınıfı init fonksiyonu çalıştı")

        #ovrride
        def metodi(self):
            print("mekatronik sınıfı metod 1")

In [161]: degiskenMekatronik = Mekatronik()
        yazilim sınıfı init fonksiyonu çalıştı
        mekatronik sınıfı init fonksiyonu çalıştı

In [162]: degiskenYazilim = Yazilim()
        yazilim sınıfı init fonksiyonu çalıştı

In [163]: degiskenYazilim.metodi()
        yazilim sınıfı metod 1

In [164]: degiskenYazilim.metod2()
        yazilim sınıfı metod 2

In [165]: degiskenMekatronik.metodi()
        mekatronik sınıfı metod 1

In [166]: ## init fonksiyonu içindekileri devraldığı gibi metodları da alır.Aynı isimde bir metod kend
        ine de varsa
```

Polymorphism

```
In [168]: class Ogrenci():
        def __init__(self,isim):
            self.isim = isim

        def bilgiVer(self):
            print(f"{self.isim} okula gider")

In [170]: class Calisan():
        def __init__(self,isim):
            self.isim = isim

        def bilgiVer(self):
            print(f"{self.isim} işe gider")

In [171]: ogrenci = Ogrenci("Ömer")

In [172]: ogrenci.bilgiVer()
        Ömer okula gider

In [173]: calisan = Calisan("Can")

In [174]: calisan.bilgiVer()
        Can işe gider

In [176]: listem =[ogrenci,calisan]

In [177]: for kişi in listem:
        kişi.bilgiVer()

        Ömer okula gider
        Can işe gider

In [180]: def aciklama(kisi):
        kisi.bilgiVer()

In [181]: aciklama(ogrenci)
        Ömer okula gider

In [183]: aciklama(calisan)
        Can işe gider
```

Special Methods

```
In [197]: class Ogrenci():
        def __init__(self,isim,bolum,sure):
            self.isim = isim
            self.bolum = bolum
            self.sure = sure

In [199]: ogrenci1 = Ogrenci("Ömer", "mekatronik",4)

In [200]: print(ogrenci1)

<_main__.Ogrenci object at 0x0000019C7761EF70>

In [201]: class Ogrenci():
        def __init__(self,isim,bolum,sure):
            self.isim = isim
            self.bolum = bolum
            self.sure = sure
        def __str__(self):
            return f"{self.isim} {self.bolum} okuyor"

In [203]: ogrenci1 = Ogrenci("Ömer", "mekatronik",4)

In [204]: print(ogrenci1)

Ömer mekatronik okuyor

In [205]: class Ogrenci():
        def __init__(self,isim,bolum,sure):
            self.isim = isim
            self.bolum = bolum
            self.sure = sure
        def __str__(self):
            return f"{self.isim} {self.bolum} okuyor"

        def __len__(self):
            return self.sure

In [206]: ogrenci1 = Ogrenci("Ömer", "mekatronik",4)

In [207]: print(ogrenci1)

Ömer mekatronik okuyor

In [ ]: len(ogrenci1)
```

Error Handling

try && except && else &&finally

```
In [ ]: while 1 :

        try:
            deger = int(input("bir sayı girin.))

        except:
            print("lütfen sayı girin.\n")
            continue

        else:
            print("teşekkürler")
            break

        finally:
            print("finally çalıştı\n")
```

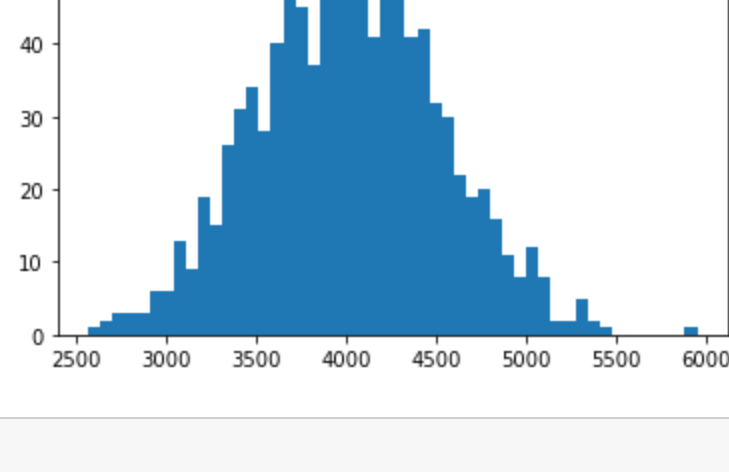
Modules && Packets

```
In [8]: import numpy as np
        import matplotlib.pyplot as matplotlib

In [10]: maasListesi = np.random.normal(4000,500,1000)
        numpy.mean(maasListesi)

Out[10]: 4021.729969144526

In [11]: matplotlib.hist(maasListesi,50)
        matplotlib.show()



In [ ]:
```