

```
In [1]: # sınıf tanımlama
class Kişi:
    # sınıf özellikleri
    isim = ""
    soyisim = ""
    yas = 0

    # sınıf metodları
    def info(self):
        print("İsim:", self.isim)
        print("Soyisim:", self.soyisim)
        print("Yaş:", self.yas)

class araba:
    marka=''
    model=''
    renk=''
    yil=0

    def metot(self):
        print("Marka:", self.marka)
        print("Model:", self.model)
        print("Renk:", self.renk)
        print("Yıl:", self.yil)
```

```
In [2]: # nesne oluşturma
ali = Kişi()
veli = Kişi()
```

```
In [3]: ali.isim='Ali'
ali.soyisim='Yılmaz'
ali.yas=30

veli.isim='Veli'
veli.soyisim='Kalkan'
veli.yas=25
```

```
In [4]: ali.info()
```

```
İsim: Ali
Soyisim: Yılmaz
Yaş: 30
```

```
In [5]: veli.info()
```

```
İsim: Veli
Soyisim: Kalkan
Yaş: 25
```

```
In [6]: mercedes=araba()
mercedes.marka='Mercedes'
mercedes.model='C200'
mercedes.renk='Beyaz'
mercedes.yil='20'
```

```
In [7]: mercedes.metot()
```

```
Marka: Mercedes
Model: C200
Renk: Beyaz
Yıl: 20
```

```
In [8]: mercedes.marka
```

```
Out[8]: 'Mercedes'
```

```
In [9]: type(mercedes)
```

```
Out[9]: __main__.araba
```

```
In [1]: # Bir Araba sınıfı oluşturalım
class Araba:
    # constructor (yapıcı)
    # nesnelere ilk defa oluşturulduğunda çalışır
    def __init__(self, marka, model, yil, renk):
        self.marka = marka
        self.model = model
        self.yil = yil
        self.renk = renk
        self.durum = 'Durdurulmuş' # Araba başlangıçta durdurulmuş durumda

    def calistir(self):
        self.durum = 'Çalışıyor'
        print(f"{self.marka} {self.model} çalıştırıldı.")

    def durdur(self):
        self.durum = 'Durdurulmuş'
        print(f"{self.marka} {self.model} durduruldu.")

    def durumu_goster(self):
        print(f"{self.marka} {self.model}'in durumu: {self.durum}")
```

```
In [2]: # Araba nesnelere oluşturalım
araba1 = Araba('Toyota', 'Corolla', 2020, 'Beyaz')
araba2 = Araba('Honda', 'Civic', 2021, 'Mavi')
```

```
In [3]: # Arabaları çalıştırmadan durumlarını gösterelim
araba1.durumu_goster()
```

Toyota Corolla'in durumu: Durdurulmuş

```
In [4]: # Arabaları çalıştırmadan durumlarını gösterelim
araba2.durumu_goster()
```

Honda Civic'in durumu: Durdurulmuş

```
In [5]: # Arabaları çalıştıralım ve durumlarını gösterelim
araba1.calistir()
araba1.durumu_goster()
```

Toyota Corolla çalıştırıldı.  
Toyota Corolla'in durumu: Çalışıyor

```
In [6]: araba2.calistir()
araba2.durumu_goster()
```

Honda Civic çalıştırıldı.  
Honda Civic'in durumu: Çalışıyor

```
In [7]: # Arabaları durduralım ve durumlarını gösterelim
araba1.durdur()
araba1.durumu_goster()
```

Toyota Corolla durduruldu.  
Toyota Corolla'in durumu: Durdurulmuş

```
In [8]: araba2.durdur()
araba2.durumu_goster()
```

Honda Civic durduruldu.  
Honda Civic'in durumu: Durdurulmuş

```
In [1]: # sınıf tanımlama
class Araba():
    # constructor (yapıcı)
    # nesnelere ilk defa oluşturulduğunda çalışır
    def __init__(self, model, renk, beygir_gücü, silindir):
        # sınıf özellikleri
        self.model = model
        self.renk = renk
        self.beygir_gücü = beygir_gücü
        self.silindir = silindir
```

```
In [2]: araba1=Araba('Renault Megane','Gümüş',110,4)
type(araba1)
```

```
Out[2]: __main__.Araba
```

```
In [3]: araba1.model
```

```
Out[3]: 'Renault Megane'
```

```
In [4]: araba1.renk
```

```
Out[4]: 'Gümüş'
```

```
In [5]: araba1.beygir_gücü
```

```
Out[5]: 110
```

```
In [6]: araba1.silindir
```

```
Out[6]: 4
```

```
In [7]: araba2=Araba('Renault Clio','Beyaz',75,3)
```

```
In [8]: araba2.model
```

```
Out[8]: 'Renault Clio'
```

```
In [9]: araba2.renk
```

```
Out[9]: 'Beyaz'
```

```
In [10]: araba2.beygir_gücü
```

```
Out[10]: 75
```

```
In [11]: araba2.silindir
```

```
Out[11]: 3
```

```
In [12]: # araba1 nesnesinin kullanılabilir bileşenleri
dir(araba1)
```

```
Out[12]: ['__class__',
          '__delattr__',
          '__dict__',
          '__dir__',
          '__doc__',
          '__eq__',
          '__format__',
          '__ge__',
          '__getattr__',
          '__getstate__',
          '__gt__',
          '__hash__',
          '__init__',
          '__init_subclass__',
          '__le__',
          '__lt__',
          '__module__',
          '__ne__',
          '__new__',
          '__reduce__',
          '__reduce_ex__',
          '__repr__',
          '__setattr__',
          '__sizeof__',
          '__str__',
          '__subclasshook__',
          '__weakref__',
          'beygir_gücü',
          'model',
          'renk',
          'silindir']
```

```
In [1]: class OyunKarakteri:
# constructor (yapıcı)
# nesnelere ilk defa oluşturulduğunda çalışır
def __init__(self, ad, max_saglik=100, max_enerji=100):
self.ad = ad
self.max_saglik = max_saglik
self.saglik = max_saglik # Başlangıçta maksimum sağlık
self.max_enerji = max_enerji
self.enerji = max_enerji # Başlangıçta maksimum enerji
self.silah = None # Karakterin başlangıçta silahı yok

def atak(self):
if self.enerji >= 10:
print(f"{self.ad} saldırı yapıyor!")
self.enerji -= 10
else:
print(f"{self.ad}, enerjisi yetersiz. Daha fazla enerji topla!")

def savunma(self):
print(f"{self.ad} savunma yapıyor!")

def saglik_kontrol(self):
print(f"{self.ad}'in sağlık durumu: {self.saglik}/{self.max_saglik}")

def enerji_kontrol(self):
print(f"{self.ad}'in enerji durumu: {self.enerji}/{self.max_enerji}")

def silah_tak(self, silah_ad):
self.silah = silah_ad
print(f"{self.ad}, {silah_ad} silahını taktı!")
```

```
In [2]: # OyunKarakteri sınıfından nesne oluşturalım
oyuncu = OyunKarakteri("Kahraman")
```

```
In [3]: # Oyun karakteriyle ilgili işlemler gerçekleştirelim
oyuncu.atak()
oyuncu.savunma()
oyuncu.saglik_kontrol()
oyuncu.enerji_kontrol()
```

```
Kahraman saldırı yapıyor!
Kahraman savunma yapıyor!
Kahraman'in sağlık durumu: 100/100
Kahraman'in enerji durumu: 90/100
```

```
In [4]: # Yeni bir silah ekleyelim
oyuncu.silah_tak("Kılıç")
```

```
Kahraman, Kılıç silahını taktı!
```

```
In [5]: # Silah takıldıktan sonra tekrar saldırı yapalım
oyuncu.atak()
oyuncu.enerji_kontrol()
```

```
Kahraman saldırı yapıyor!
Kahraman'in enerji durumu: 80/100
```