

Derste yapılan class örnekleri

```
dil='tr'
class meyve:
    def __init__(self,ad,name):
        self.ad=ad
        self.name=name
    def __repr__(self):
        if dil=='tr':
            adi=self.ad
        elif dil=='en':
            adi=self.name
        return adi

elma=meyve('elma','apple')
print(elma)
dil='en'
print(elma)
dil='tr'|
print(elma)

class ders:
    aktif=True # tüm ders nesnelerinde ortak değişken
    def __init__(self,ad):
        self.ad=ad # bir ders nesnesine ait değişken

d1=ders("Matematik")
d2=ders("Fizik")

print(d1.ad,d2.ad)
print(d1.aktif,d1.aktif)

ders.aktif=False #ders class'inin aktif adlı değişkeni False oldu
print(d1.aktif,d2.aktif)

d1.aktif=True # d1 nesnesinin aktif adlı değişkeni True oldu
print(d1.aktif,d2.aktif)

#iki boyutlu koordinat sisteminde caliselim
from math import cos,sin,radians

class nokta:
    def __init__(self,ad="",x=0,y=0):
        self.ad=ad
        self.x=x
        self.y=y

    def donustur(self,ad,alfa):
        c=cos(alfa)
        s=sin(alfa)
        yenix=self.x*c+self.y*s
        yeniy=-self.x*s+self.y*c
        yeninokta=nokta(ad,yenix,yeniy)
        return yeninokta

    def __repr__(self):
        nstr=self.ad+": "+str(format(self.x, '.3f'))+", "+str(format(self.y, '.3f'))
        return nstr

n1=nokta("A",3,4)
n2=nokta("B",6,8)

# Bir n3 noktasi olusturalim
# Bu nokta 30 derece dondurulmus eksende n2 noktasinin koordinatlarina sahip olsun
n3=n2.donustur("C",radians(30))
print(n2,n3)
```

```

class cizgi:
    def __init__(self,n1,n2):
        self.n1=n1
        self.n2=n2
        self.uzunluk=((n2.x-n1.x)**2+(n2.y-n1.y)**2)**0.5
        self.egim=(n2.y-n1.y)/(n2.x-n1.x)
        self.renk='kırmızı'
        self.stil='düz'

    def stil_degistir(self,yenistil):
        self.stil=yenistil

c1=cizgi(n1,n2)

c1=cizgi(n1,n2)

print(c1.n1.ad,"ve",c1.n2.ad,"noktalari arasi uzunluk:",c1.uzunluk,"egim:",
      c1.egim,"renk:",c1.renk,"stil:",c1.stil)

class ders:
    tip="lisans"
    def __init__(self,ad):
        self.ad=ad

d1=ders("Fizik")
print(d1.tip,d1.ad)

d2=ders("Kimya")
print(d2.tip,d2.ad)

class nokta: # 2 boyutlu koordinatlara sahip olsun
    def __init__(self,ad,x,y):
        self.ad=ad
        self.x=x
        self.y=y

class cizgi: # duzlemde bir cizgi
    def __init__(self,n1,n2):
        self.n1=n1
        self.n2=n2
        self.uzunluk=((n2.x-n1.x)**2+(n2.y-n1.y)**2)**0.5
    def __repr__(self):
        yazi=self.n1.ad+'-'+self.n2.ad+": "+str(format(self.uzunluk,'.3f'))
        return yazi

nok1=nokta("A",3,4)
nok2=nokta("B",6,8)
c1=cizgi(nok1,nok2)

print(c1.n1.ad,'noktasindan',c1.n2.ad,'noktasina mesafe:',c1.uzunluk)
p1=nokta('p1',4,6)
p2=nokta('p2',6,5)
p3=nokta('p3',5,3)
p4=nokta('p4',3,4)
l1=cizgi(p4,p3)
l2=cizgi(p2,p3)
l3=cizgi(p1,p2)
l4=cizgi(p1,p4)
l5=cizgi(p2,p4)
l6=cizgi(p1,p3)
cizgiler=[l1,l2,l3,l4,l5,l6]
for cizgi in cizgiler:
    #print(cizgi.n1.ad,'noktasindan',cizgi.n2.ad,'noktasina uzaklik:',cizgi.uzunluk)
    print(cizgi)

```

```

class karmasik:
    def __init__(self, reel, sanal):
        self.reel=reel
        self.sanal=sanal
    def __repr__(self):
        if self.sanal>0:
            isaret='+'
        else:
            isaret=''
        yazi=str(self.reel)+isaret+str(self.sanal)+"i"
        return yazi
    def __add__(self, diger):
        r=self.reel+diger.reel
        s=self.sanal+diger.sanal
        top=karmasik(r,s)
        return top
    def __mul__(self, diger):
        r=self.reel*diger.reel-self.sanal*diger.sanal
        s=self.reel*diger.sanal+self.sanal*diger.reel
        carpim=karmasik(r,s)
        return carpim

a=karmasik(3,-5)
b=karmasik(2,3)
c=a*b
print(c)

```

```

class meyve:
    def __init__(self, ad, renk, tat):
        self.ad=ad
        self.renk=renk
        self.tat=tat
    def __repr__(self):
        yazi=self.ad+"-"+self.renk+"-"+self.tat
        return yazi

m1=meyve('elma', 'kirmizi', 'tatli')
m2=meyve('armut', 'sari', 'tatli')
m3=meyve('kiraz', 'kirmizi', 'tatli')
m4=meyve('limon', 'sari', 'eksi')
m5=meyve('portakal', 'turuncu', 'eksi')
meyveler=[m1,m2,m3,m4,m5]

##for m in meyveler:
##    if m.renk=='kirmizi':
##        print(m)

```

```

class araba:
    def __init__(self,model,yil,renk):
        self.model=model
        self.yil=yil
        self.renk=renk
    def __repr__(self):
        yazi=self.model+"-"+str(self.yil)+"-"+self.renk
        return yazi

```

```

a1=araba("Toyota",2005,'mavi')
a2=araba("Honda",2000,'beyaz')
a3=araba("Ford",1990,'beyaz')
a4=araba("Opel",1996,'kirmizi')
arabalar=[a1,a2,a3,a4]

```

```

##for a in arabalar:
##    if a.renk=='beyaz':
##        print(a)

```

```

from copy import *

```

```

class matris:
    def __init__(self,dizi):
        self.dizi=deepcopy(dizi)

    def __repr__(self):
        yazi=''
        for i in range(len(self.dizi)):
            for j in range(len(self.dizi[i])):
                yazi=yazi+str(self.dizi[i][j])+","
            yazi=yazi+"\n"
        return yazi

    def __add__(self,b):
        top=matris(self.dizi)
        for i in range(len(self.dizi)):
            for j in range(len(self.dizi[i])):
                top.dizi[i][j]+=b.dizi[i][j]
        return top

    def __mul__(self,mb):
        a=self.dizi
        b=mb.dizi
        #Çarpım: a(n,m)*b(m,p)=c(n,p)
        n=len(a)
        m=len(a[0])
        p=len(b[0])
        if m!=len(b):
            print('Çarpım yapılamaz !')
            return None
        c=[[0 for i in range(p)] for j in range(n)]
        for i in range(n):
            for k in range(p):
                for j in range(m):
                    c[i][k]+=a[i][j]*b[j][k]
        mc=matris(c)
        return mc

```

```

a=[[2,3],[5,1]]
b=[[9,2],[3,7]]
a1=matris(a)
a2=matris(b)
a3=a1*a2
print(a3)

```

```

dil='tr'
class kredi:
    def __init__(self,t,u,a):
        self.t=t
        self.u=u
        self.a=a
    def __repr__(self):
        if dil=='tr':
            a='akts'
        elif dil=='en':
            a='ects'
        s="("+str(self.t)+"+"+str(self.u)+") (" +a+": "+str(self.a)+") "
        return s

class ders:
    def __init__(self,kod,ad,name,kredi):
        self.kod=kod
        self.ad=ad
        self.name=name
        self.kredi=kredi
    def __repr__(self):
        if dil=='tr':
            ad=self.ad
        elif dil=='en':
            ad=self.name
        y=self.kod+" "+ad+" "+str(self.kredi)
        return y

```

Aşağıdaki class içinde bazı kısaltmalardan faydalanılmıştır.

```

class puan:
    def __init__(self,ders):
        s = self
        # v:vize f: final b: bütünleme g: geçme notu h: harf notu
        s.v = s.f = s.b = s.g = -1
        s.h = ""
        s.ders = ders

    def hesapla(self):
        s = self
        v = s.v
        # bütünleme sınavına girilmemiş ise final sınavına göre
        # geçme notu hesaplanacaktır.
        if s.b == -1:
            f = s.f
        else:
            f = s.b
        s.g = v * 0.4 + f * 0.6

    def notgir(self,tip,puan):
        s = self
        if tip == 'v': s.v=puan
        elif tip == 'f': s.f=puan
        elif tip == 'b': s.b=puan

    def __repr__(self):
        s = self
        y = str(s.ders) + " v:" + str(s.v) + " f:" + str(s.f) + " b:" + str(s.b)
        y = y + " gn:" + str(s.g) + " hfn:" + s.h
        return y

k = kredi(3,0,5)
d1 = ders('001008', 'Matematik', 'Mathematics', k)
p1 = puan(d1)
p1.notgir('v',65)
p1.notgir('f',75)
p1.hesapla()
print(p1)

```