

NumPy (Numerical Python), Python programlama dilinde **sayısal hesaplamalar** yapmak için tasarlanmış **yüksek performanslı** bir dizi işlemleri kütüphanesidir.

Python'un standart listelerinden farklı olarak, NumPy dizileri **homojen** yapıda olmalıdır, yani dizi içindeki tüm elemanlar aynı veri tipinden olmalıdır. Bu sayede NumPy arrayleri, Python listelerine göre **daha hızlı ve verimli** bir şekilde işlem yapar.

NumPy'in temel özellikleri şunları içerir:

**Çok boyutlu diziler (array) ve matrisler:** NumPy, tek boyutlu dizilerin yanı sıra iki veya daha fazla boyuta sahip diziler oluşturmaya ve işlemeye olanak tanır. Bu sayede tablolar, görüntüler ve diğer çok boyutlu verileri kolayca temsil edebilirsiniz.

**Zengin matematiksel işlemler:** NumPy, temel aritmetik işlemler (toplama, çıkarma, çarpma, bölme) yanında trigonometrik fonksiyonlar, lineer cebir işlemleri, rastgele sayı üretme gibi birçok matematiksel işleve sahiptir.

**Verimli bellek yönetimi:** NumPy dizileri, C programlama dilinde yazıldığı için Python listelerine göre bellek kullanımında daha verimlidir. Bu sayede büyük veri kümeleriyle çalışırken performans artışı sağlar.

NumPy, özellikle **veri bilimi**, **yapay zeka** ve **sayısal hesaplama** gerektiren bilimsel alanlarda yaygın olarak kullanılır.

```
# import numpy as np

#np da bir sey yazacaksak np yazmamız yeterlidir

##numpy arrays##

# benimListem=[20,30,40]

# print(type(benimListem))

# np dizisine cevirebiliyoruz np.array(benimListem) seklinde yazıp

# print(np.array(benimListem))

# print(type(np.array(benimListem))) çıktı olarak np.ndarray verir

matrixListesi=[[10,20,30],[20,30,40],[30,40,50]]

# print(type(matrixListesi))

# print(matrixListesi[0][0])

print(np.array(matrixListesi))

##arange##

# result=list(range(0,10))

# print(result)

print(np.arange(0,100))

#başlangıç(dahil) ,bitiş(dahil değil),atlama (step size)

#print(np.arange(0,10,2))

##zeros##
```

```

0 olusturur

#print(np.zeros(5)) #5tane 0 olusturur
#print(np.zeros((2,2))) tuple gibi vermem gerekiyor matris için
##ones##

1 olusturur

# print(np.ones((5,5)))

###inspace##

#baslangıc ve bitis parametrelerini ver kac tane esit aralıga
bolunmus sayı olsun istersen belirt

# print(np.linspace(0,20,5)) 5 sayı olmalı esit aralık olan
aralarında ve 0 dan 20. olan baslangıc bitis

#esit aralıkla dagıtılmıs sayıları yazmak için kullanılır
#eye##

#kosegen olarak 1 lerden olusan kaca kaclık bir kare matris
olusturur

# print(np.eye(10))

#random

#print(np.random.randn(8)) # tek boyutludur

# #random sayılardan olusan dızı verir istedigimiz adet eleman
sayısından olusan

#print(np.random.randn(4,4)) # matris seklinde verir

# print(np.random.randint(1,10)) # 1-10 (10 dahil degıl)arası
rasgele int deger verir

# print(np.random.randint(1,10,5)) # kac tane sayı vermesi
gerektigi en sona yazılır 1-10 arası(10dahıl degıl)

# benimNumpyDizim=np.arange(30) #0-30 arası dızı olusturur 30 dahil
degıl

# print(benimNumpyDizim)

#benimRandomDizim=np.random.randint(0,100,30) # 0-100(100 dahil
degıl) arası 5 tane rasgele deger

# print(benimRandomDizim)

#numpy dizi methodları##

#reshape

#eger bir dizim varsa reshape yaptigımda rasagele rakam veremem

#elimdeki eleman sayısı kadar olmalı carpımları matris olusturmak
icin dizimden

```

```
#result=benimRandomDizim.reshape(6,5) #6 satır 5 sütun
# print(result)
#max ve min
# result=benimNumpyDizim.max()
# print(result)
#argmax max. elemanın kaçınclı indekste olduğunu verir
#argmin min. elemanın kaçınclı indekste olduğunu verir
# print(benimNumpyDizim.argmax())
#shape deger donduru guncellemez guncel dızımın seklını bize verir
# print(benimNumpyDizim.shape)
#indeksleme
# benimDizim=np.arange(0,15)
# print(benimDizim)
# #numpy dizilerinde indeksler 0 dan baslar
# print(benimDizim[0])
# #slicing
#print(benimDizim[3:5]) # 5 dahil degıl (stoping indeks)
#benimDizim[3:8]=-5 #3-8.indeksler arası degerler -5 oldu
# print(benimDizim) # ayrı dızı olusturmadı dızımdeki elemanları
degıstırdı
#baskaDizi=np.arange(0,24)
# slicingDizisi=baskaDizi[4:9]
# print(slicingDizisi)
#slicingDizisi=700 #type ne bakarsak int olmus olur direk int dgere
esıtlemlıs oluruz
# slicingDizisi[:]=700
# #tum elemanları 700 e esıtlendı
# print(slicingDizisi) #baska dızıyı cagırırsak 4 ve 9 indeks
aralıgındaki degerlerde degıstı ve 700 e esıtlendı
# print(baskaDizi) # orijinalini degıstirmeden yapamk ıstersek
#ornekDizi=np.arange(0,24)
# ornekDiziKopyasi=ornekDizi.copy()
# #ornek dızı ve ornek dızı kopyası ayrı bır dızı olmus oldu
# ornekDiziKopyaSlicing=ornekDiziKopyasi[3:6]
```

```

# ornekDiziKopyaSlicing[:]=800
# print(ornekDiziKopyaSlicing)
# print(ornekDiziKopyasi)
# print(ornekDizi)

#matrix
# benimListem=[[10,20,30],[20,30,40],[40,50,60]]
# benimMatrixDizim=np.array(benimListem)
# print(benimMatrixDizim)
# print(benimMatrixDizim[0])
# print(benimMatrixDizim[1][2])
# print(benimMatrixDizim[1,2]) #yukarki satırla aynı işleve sahip
# #slicing
print(benimMatrixDizim[1:,2]) # 1.indeksdeki satır başlayıp geri
kalan sütunlardaki 2. indeksdeki sütun elemanını verir
#print(benimMatrixDizim[2:,1:]) 2.indeksdeki satırdan başlar sona
kadar 2.indeksdeki sütundaki elemanları verir
# yeniListe=[[0,1,2,3,4],[5,6,7,8,9],[10,11,12,13,14],
[15,16,17,18,19],[20,21,22,23,24]]
# yeniMatrix=np.array(yeniListe)
# print(yeniMatrix[0,2,4]) #0.2.ve 4. indeks satırlarını bana verir
matrix şeklinde

#sıra vermek zorundada değilim fancy matrix demektir bu

#operasyonlar
# yeniBirDizi=np.random.randint(1,100,20)
# print(yeniBirDizi)
# print(yeniBirDizi>24)
# sonucDizisi=yeniBirDizi>24
# print(sonucDizisi)
# print(yeniBirDizi[sonucDizisi]) #filtereleme işlemi yapmış oluruz
(sadece true'leri alır)
#print(yeniBirDizi[yeniBirDizi>24]) yukarıdakiyle aynı işlemi yapar
# sonDizi=np.arange(0,24)
# print(sonDizi+sonDizi) # 2 diziyi toplar
# sonuc=sonDizi*sonDizi

```

```
# print(sonuc)
# sonuc2=sonDizi-sonDizi
# print(sonuc2)
# print(sonDizi/sonDizi)
#0/0 belırsızlıđına nan deyp dıgerlerine 1 der
# print(np.sqrt(sonDizi)) #dizinin karekokunu alır
```

## **PANDAS**