

# Numpy(Numerical Python) Eğitimi

Burada nümerik python için geliştirilen Numpy kütüphanesini göreceğiz. Birlikte kodlar üzerinde çalışmamızı gerçekleştireceğiz.

Numpy nümerik hesaplamalar yapmak için kullanılan pakettir. Python'daki listelerin yetersiz kaldığı durumlarda kullanılır. Özellikle çok boyutlu dizilerde işlem yapmak için kolaylıklar sağlamaktadır. Numpy aynı zamanda veri bilimi, veri madenciliği ya da makine öğrenmesinde kullanılan birçok kütüphanenin de temellerinde yer alır.

## Numpy'ın çağırılması

Genellikle np kısaltması ile kullanılır. Bu nedenle `import numpy as np` komutu ile çağırılır.

```
In [1]: import numpy as np
```

Numpy'da diziler array olarak çağırılır. Tek boyutlu basit bir dizi tanımlamak için;

```
In [ ]: dizi = np.array([1,2,3,4,5])
```

ic\_ice dizi yazmak için;

```
In [ ]: ic_ice = np.array([1,2,[3,4,5, [6,7,8]]])
```

```
In [ ]: ic_ice[2][3][1]
```

```
In [ ]: x1 = np.random.randint(10, size=6) # Tek boyutlu dizi
x2 = np.random.randint(10, size=(3, 4)) # iki boyutlu dizi
x3 = np.random.randint(10, size=(3, 4, 5)) # üç boyutlu dizi
```

```
In [ ]: print("x3 boyut: ", x3.ndim)
print("x3 shape:", x3.shape)
print("x3 size: ", x3.size)
print("dtype:", x3.dtype)
```

```
In [ ]: x3[1][2][1]
```

Dizi oluşturmak için

```
In [ ]: np.zeros(10, dtype=int)
```

```
In [ ]: np.ones((3, 5), dtype=float)
```

```
In [ ]: np.full((3, 5), 3.14)
```

```
In [ ]: # 0 ile 1 arasında 5 adet eşit aralıklı dizi oluştur.
np.linspace(0, 1, 5)
```

```
In [ ]: # 3e3 rastgele sayılardan oluşan bir dizi oluştur.
np.random.random((3, 3))
```

```
In [ ]: # normal dağılıma uyan ortalaması 0 standart sapması 1 olan 3e3 bir dizi oluştur.
np.random.normal(0, 1, (3, 3))
```

```
In [ ]: # 0 ile 10 arasında rastgele tam sayılardan oluşan 3e3 bir dizi
np.random.randint(0, 10, (3, 3))
```

```
In [ ]: # Birim matris
np.eye(3)
```

```
In [ ]: np.random.seed(0) # her zaman aynı rassal değerleri almak için sabitlenir.
```

## Numpy Dizisindeki Verilere Ulaşmak

```
In [ ]: print(x1[0]) # tek boyutlu
        print(x1[-1]) # sondan saymak istersek
        print(x1[-2]) # sondan ikinci
```

```
In [ ]: x2 # iki boyutlu dizi
```

```
In [ ]: print(x2[0,0]) # ilk satır, ilk sütun
        print(x2[2, 0]) # ikinci satır, ilk sütun
```

```
In [ ]: # x[start:stop:step] ile dizi içerisinde dilimleme yapılabilir.
```

```
In [ ]: x = np.arange(10) # 0-10 arasındaki sayılar
```

```
In [ ]: x
```

```
In [ ]: print(x[:5]) # ilk 5 eleman
        print(x[5:]) # 5 elemandan sonrası
        print(x[4:7]) # 4. ve 7 elemanlar arası
        print(x[::2]) # tüm elemanlar iki iki artır
        print(x[::-1]) # elemanları ters çevir
        print(x[5::-2]) # 5 ten sonrasını bir sonraki olacak şekilde ters çevir
```

```
In [ ]: # çok boyutlu dizilerde
        x2
```

```
In [ ]: x2[:2, :3] # 2 satır, 3 sütun
```

```
In [ ]: print(x2[:3, ::2]) # ilk 3 satır, sonraki olacak şekilde sütunlar
        print(x2[::-1, ::-1]) # satırları ve sütunları ters çevir
```

```
In [ ]: print(x2[:, 0]) # ilk sütundaki tüm satırlar
        print(x2[0, :]) # ilk satırdaki tüm sütunlar
```

## Dizilerin kopyalanması

```
In [ ]: x2_kopya = x2[:2, :2].copy() # x2deki ilk 2 satır, ilk 2 sütun
```

```
In [ ]: x2_kopya[0, 0] = 42 # x2 etkilenmez.
```

```
In [ ]: x2
```

```
In [ ]: ## Dizilerin yeniden şekillendirilmesi
```

```
In [ ]: np.arange(1, 10).reshape((3, 3)) # 1-10 arasındaki değerleri 3e3 matris şeklinde kaydet
```

```
In [ ]: x = np.array([1, 2, 3])
        x.reshape((3, 1)) # x i 3 satır 1 sütun şeklinde
```

## Dizilerde aritmetik işlemler

```
In [ ]: x = np.arange(4)
print("x =", x)
print("x + 5 =", x + 5)
print("x - 5 =", x - 5)
print("x * 2 =", x * 2)
print("x / 2 =", x / 2)
print("x // 2 =", x // 2) # kaç sefer var
print("-x = ", -x)
print("x ** 2 = ", x ** 2) # karesi
print("x % 2 = ", x % 2) # mod bölümünden kalan
```

```
In [ ]: -(0.5*x + 1) ** 2
```

```
In [ ]: x = np.array([-2, -1, 0, 1, 2])
abs(x) # mutlak değer
```

```
In [ ]: print(np.sum(x)) # dizideki elemanların toplamı
print(np.min(x))
print(np.max(x))
print(np.std(x)) # standart sapma
print(np.var(x)) # varyans
print(np.argmin(x)) # minimum değer indexi
print(np.argmax(x))
print(np.median(x))
print(np.percentile(x,q=25))
```

## Numpy Örnek Uygulama

transactions.csv dosyasını alıp temel istatistik hesaplamalarını yapalım.

```
In [ ]: import pandas as pd
data = pd.read_csv("transactions.csv")
costs = np.array(data['Cost'])
```

```
In [ ]: data.head()
```

```
In [ ]: costs
```

```
In [ ]: print("ort. maliyet: ", costs.mean())
print("standart sapma:", costs.std())
print("Minimum: ", costs.min())
print("Maximum: ", costs.max())
```

```
In [ ]: print("1. kartil: ", np.percentile(costs, 25))
print("Medyan: ", np.median(costs))
print("3. kartil: ", np.percentile(costs, 75))
```

```
In [ ]: print(np.any(costs > 100)) # Maliyeti 100'den fazla olan var mı?
print(np.any(costs < 0)) # 0'dan küçük olan var mı
print(np.all(costs < 1000)) # hepsi 1000'den küçük mü
print(np.sum((costs > 20) & (costs < 100))) # 20'den büyük 100'den küçük olanları topla
```

## Maskeleme

```
In [ ]: x = np.array([[5, 0, 3, 3],
[7, 9, 3, 5],
[2, 4, 7, 6]])
```

```
In [ ]: x<5 # 5'ten küçük olanlara True yazar
```

```
In [ ]: x[x<5] # 5 ten küçük olanları getir
```

## Sıralama

```
In [ ]: np.sort(x)
```

```
In [ ]: np.sort(x, axis=1) # sütuna göre sırala
```