

NUMPY

Dizi ve matrisler üzerinde, üst düzey matematiksel işlemler yapmamızı sağlar. Veri türü ndarray dir. Tek boyutlu olabileceği gibi, çok boyutlu da olabilir. Dizileri oluştururken türünü (int, float) belirlenebilir.

Tek Boyutlu Dizi (Vektör) Oluşturma

Tek boyutlu Numpy dizisi oluşturmak için **np.array()** metodunu kullanırız.

```
In [ ]: import numpy as np
vektor = np.array([1,2,3,4])
print("Vektor :", vektor)
print("Boyutu :", vektor.size)
print("Veri Tipi: ", type(vektor))
print("Array Nesne Tipi : ", vektor.dtype)
```

```
Vektor : [1 2 3 4]
Boyutu : 4
Veri Tipi: <class 'numpy.ndarray'>
Array Nesne Tipi : int32
```

Dizi elemanlarına erişebilir, değişiklik yapabiliriz.

```
In [ ]: print("vektor[3] :",vektor[3])
vektor[0] = 5
print("vektor[0] :",vektor[0])
print("Vektor :", vektor)
```

```
vektor[3] : 4
vektor[0] : 5
Vektor : [5 2 3 4]
```

İki Boyutlu Dizi Oluşturma

İki boyutlu dizileri de **np.array()** metodu ile oluştururuz.

```
In [ ]: #2 boyutlu dizi oluşturma
iki_boyutlu = np.array([[1.2,1.3,3.4],[2.5,6,5.5]])
print("Dizi \n", iki_boyutlu)
print("Boyutu :", iki_boyutlu.size)
print("Veri Tipi : ", type(iki_boyutlu))
print("Array Nesne Tipi : ", iki_boyutlu.dtype)
print("Dizinin Boyutu : ", iki_boyutlu.shape)
print("iki_boyutlu[0,2] :", iki_boyutlu[0,2])
```

```
Dizi
[[1.2 1.3 3.4]
 [2.5 6.  5.5]]
Boyutu : 6
Veri Tipi : <class 'numpy.ndarray'>
Array Nesne Tipi : float64
Dizinin Boyutu : (2, 3)
iki_boyutlu[0,2] : 3.4
```

Özel Numpy Dizileri

np.arange() başlangıç, bitiş ve sayıların tipini belirleyebileceğimiz, artan sayılardan oluşan bir dizi oluşturur.

```
In [ ]: artan_matris = np.arange(1,10,dtype=float)
        print(artan_matris)
```

```
[1.  2.  3.  4.  5.  6.  7.  8.  9.]
```

np.linspace() belirlediğimiz başlangıç ve bitiş aralığında eşit olacak şekilde dilimler oluşturur. Dilim miktarı en fazla 50'dir.

```
In [ ]: dilim_matris = np.linspace(0,50,5)
        print(dilim_matris)
```

```
[ 0.  12.5 25.  37.5 50.]
```

Özel Numpy Matrisleri

np.zeros() tamamı 0 sayısından oluşan bir matris oluşturur.

```
In [ ]: sifir_matris = np.zeros((2,2))
        print(sifir_matris)
```

```
[[0.  0.]
 [0.  0.]]
```

np.ones() tamamı 1 sayısından oluşan bir matris oluşturur.

```
In [ ]: bir_matris = np.ones((3,2))
        print(bir_matris)
```

```
[[1.  1.]
 [1.  1.]
 [1.  1.]]
```

np.full() sabit matristir. Belirleyeceğimiz değer ile, istediğimiz boyutta matris oluşturur.

```
In [ ]: sabit_matris = np.full((2,2),7)
        print(sabit_matris)
```

```
[[7 7]
 [7 7]]
```

np.eye() belirlediğimiz boyutta birim matris oluşturmamızı sağlar.

```
In [ ]: birim_matris = np.eye(2)
        print(birim_matris)
```

```
[[1.  0.]
 [0.  1.]]
```

np.random() istediğimiz boyutta değerleri 0-1 arasında olacak şekilde rastgele değerlerden oluşan bir matris oluşturmamızı sağlar.

```
In [ ]: rastgele_matris = np.random.random((2,2))
print(rastgele_matris)

[[0.43264606 0.05510894]
 [0.34511772 0.47597606]]
```

Özel Numpy Dizileri

Matrisleri istediğimiz şekilde bölümleyerek yeni değerler atayabiliriz. Birbiri ile bağlantılı matrislerde, ilkinde yapacağımız değişikliğin diğerlerini de etkileyeceğini unutmayın.

Tek Boyutta Dilimleme

```
In [ ]: dizi = np.array([[1,2,3], [4,5,6], [7,8,9]])

print(dizi)
print("dizi[1,:] : ", dizi[1,:])
print("dizi[:,2] : ",dizi[:,2])

[[1 2 3]
 [4 5 6]
 [7 8 9]]
dizi[1,:] :  [4 5 6]
dizi[:,2] :  [3 6 9]
```

İki Boyutta Dilimlenme

```
In [ ]: print("dizi[:2, 1:2] : \n",dizi[:2, 1:2])

dizi[:2, 1:2] :
[[2]
 [5]]
```

Matris Fonksiyonları

Sum bütün elemanların toplamını verir. **axis** parametresinde yapacağımız değişiklik ile satır ya da sütun toplamını da alabiliriz.

```
In [ ]: x = np.array([[1,2,3],[4,5,6]])
print("Matris :\n", x)
print("Matris Toplamı : ", np.sum(x))
print("Axis = 0 " , np.sum(x, axis=0))
print("Axis = 1 " , np.sum(x, axis=1))

Matris :
[[1 2 3]
 [4 5 6]]
Matris Toplamı : 21
Axis = 0  [5 7 9]
Axis = 1  [ 6 15]
```

Mean elemanların ortalamasını verir.

```
In [ ]: print("Matrisin Ortalaması : ", np.mean(x))
        print("Axis = 0 " , np.mean(x, axis=0))
        print("Axis = 1 " , np.mean(x, axis=1))
```

```
Matrisin Ortalaması :  3.5
Axis = 0  [2.5 3.5 4.5]
Axis = 1  [2. 5.]
```

Transpose satır ve sütunların yerlerini değiştirir. Vektörlerin devriği alınmaz.

```
In [ ]: transpose = np.array([[1,2], [3,4]])
        print("Matris \n", transpose)
        print("\nTranspose Alınınca \n", (transpose.T))
```

```
Matris
[[1 2]
 [3 4]]
```

```
Transpose Alınınca
[[1 3]
 [2 4]]
```

Reshape matrisin boyutunu değiştirmemizi sağlar. İşlem sırasında matris boyutunun değişiklik için uygun olması gerekmektedir.

```
In [ ]: dizi = np.array([1,2,3,4,5,6,7,8,9,10,11,12])
        print("Matris \n", dizi)
        print("\nBoyut Değişimi \n", dizi.reshape(2,3,2))
```

```
Matris
[ 1  2  3  4  5  6  7  8  9 10 11 12]
```

```
Boyut Değişimi
[[[ 1  2]
  [ 3  4]
  [ 5  6]

  [ 7  8]
  [ 9 10]
  [11 12]]]
```

İletişim



Berkay Yürür