|  |  |  |  |
| --- | --- | --- | --- |
| **Array olusturma** | Np.array([4,5,9]) | **My\_arrray.dtype** | Data tipini verir |
| or | My\_list=[4,5,9]  Np.array(my\_list) | **X=np.array([1,2], dtype=np.float64)** | Data tipini Float yap demek |
| **np.arange(10)** | 0’dan 9’a kadar sayilarla array | **astype**  **Astype(‘int’)** | arraylerin dtype’ini degistiriyor  mesela int’e ceviriyor. |
| **np.zeros([2,2]** | 2’ye 2, 0'lardan oluşan bir matris döndürür.  (0’in bool type’i False oldugu icin, dtype bool dersem icini False ile doldurur.) | **-arr. size**  **-np.resize(array, (5,5))** | -Eleman sayisini verir.  -Icine size olarak yazarsam, eleman sayisindan bagimsiz, resize yapabiliyor. |
| **np.empty([2,2]** | 2’ye 2, rastgele degerlerle dolu | **size=(3,5)** | Shape’ini belirler |
| **np.ones([2,2]** | 2’ye 2, 1'lerdan oluşan bir matris döndürür. | **np.sort(myarr, axis=None**) | Cok boyutlu arrayin tum elemanlarini siralatmak icin |
| **np.full((2,2),10)** | 2’ye 2, 10'lardan oluşan bir matris döndürür. | **np.sort(a,axis=-1)** | Default=-1, yazmasam da kucukten buyuge siralar |
| **np.linspace(1, 20, num=10)** | 1’den 20’ye kadar linear araliklarla 10 sayi | **np.sort(myarr, axis = 0)** | sutunlari(yukaridan asagi kucukten buyuge ) siraliyor |
| **np.logspace (1,3)** | 10\*\*1 10\*\*3 u hesaplayip arasina 50 tane logorotmik deger dondurur.-Defaultu 50 | **np.sort(myarr, axis = 1)** | satirlari(soldan saga kucukten buyuge) siraliyor |
| **Base=2 dersem** | 10’luk degil, 2’lik tabanda hesaplatir. | **myarray=np.array(mylist, ndmin=4)** | 4 boyutlu hale getirir |
| **np.random.rand(3,3)** | 0’la 1 arasinda uniform dagilimli rastgele sayilardan olusur | **ndim** | kac boyutlu ise verir |
| **np.random.rand(50000),bins=20)** | Bins 20 dersem de 20 aralikla olustur demek | **arr.transpose()** | satirlari sutunlara sutunlari satirlara koyar |
| **np.random.randint(1,100, 10)** | 10 tane random sayi uretir  Size=(2,4) yazarsam da 2’ye 4 array olacak sekilde hazirlar | **Array.split(3)** | Array’i 3’e ayirir |
| **Shape** | Shape(2,5) 2D | **np.split(a,4)** | 4 esit parcaya bol demek |
|  | Shape(2,5,4) 3D | **Np.split(x,[2,6])** | 3’e boler |
| **My\_array.shape** | Shape’i verir | **np.vsplit(x, [1,3])** | axis=0 gibi satir bazinda ayiriyor |
| **My\_array.reshape(x,y)** | **x satir, y sutundan olusan yeni bir array** | **np.hsplit(x, [1,3])** | axis=1 gibi sutun bazinda |
| **Reshape(-1)** | Tek boyutlu hale getirir | **arr[2, :1]** | 2. satira git, onun da 1. columundan basla sonuna kadar git |
| **My\_array.reshape(x,y,-1)** | -1 bilinmeyen bir boyut, verdigimiz satir ve sutuna gore boyutu ayarlar | **arr[::2, ::3]** | 2’ser atlayarak tum satirlari, 3’er atlayarak tum sutunlari al |
| **np.eye(3)** | diagonal sekilde 1’leri doldurup geri kalani 0’la doldurur. | **arr[0:2, :0]** | 0’dan 2’ye kadar git satirlarda, column olarak, 0. columnu al |

|  |  |  |  |
| --- | --- | --- | --- |
| **arr[(arr !=3) & (arr!=4)]** | 3'e ve 4'e esit olmayanlari getir | **arr[1,[1,3]]** | 1. satiri, 1. sutunla kesistir indeksi getir, sonra yine 1. satiri 3. sutunla kesistir indeksi getir demek |
| **a[:5]=77** | 5. indekse kadar olan degeri 77 yapar, geri kalani oldugu gibi getirir | **arr[[0,9],[1]]** | 0.satiri 1. sutunla kesistir, sonra 9. satiri yine 1. sutunla kesistir, indexi getir demek |
| **np.substract(arr,2)=arr-2** | Herbir elemandan 2 cikarir | **arr[1:3, [1,2]]** | satirlarda 1'den 3'e kadar git dedik, bunlarin da 1. ve 2. sutunu getir demek |
| **np.add(arr,2)= arr+2** | Herbir elemana 2 toplar | **arr[1,1]=100** | 1’1e 1’deki elemani, 100e donusturur |
| **np.multiply(arr,2)=arr\*2** | Herbir elemani 2 ile carpar | **arr[0,1]** | 0 satir, 1.sutunla kesisimini getirir.  sadece 0 yazsak ilk satirin hepsini getirecekti |
| **np.divide(arr,2)** | Herbir elemani 2’ye boler | **arr>20** | 20’den buyuk elemanlari icin  **bool** dondurur |
| **np.power(arr,3)** | 3.kuvvetini aliyor herbir elemanin | **bool\_arr=arr>20**  **arr[bool\_arr]** | Coditionimi yeni bir arraye atayip, onu array olarak cagirirsam degerleri verir |
| **np.mod(arr,2)** | 2 ile bolumunden kalanini veriyor |  |  |
| **np.absolute(-9)** | mutlak degeri aliyor |  |  |
| **np.sqrt(arr)** | karakoklerini aliyor. |  |  |
|  |  |  |  |
| **np.max(arr)** | Max deger |  |  |
| **np.min(arr)** | Min deger |  |  |
| **arr.argmax(),** | bir dizinin içindeki en büyük değerin indeksini döndürür |  |  |
| **arr.argmin()** | bir dizinin içindeki en kucuk değerin indeksini döndürür |  |  |
| **np.log(arr) #logunu aliyor** |  |  |  |
| **np.mean(arr)**  **arr.mean()** | ortalamayi verir. |  |  |
| **np.median(arr)** | mediani verir.ortadaki |  |  |
| **arr.std()** | standard sapmayi verir |  |  |
| **arr.ptp()** | bu da en buyuk deger ile en kucuk deger arasindaki farki verir |  |  |
| **xy=np.concatenate((x,y))** | X ve y gibi 2 farkli arrayi birlestiriyor |  |  |
| **np.copy()** | Kopyasini alarak baska bir atama yapabiliriz |  |  |