Numpy-4

December 16, 2022

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

[2]:
```
arr1 = np.random.randint(-50,50,8)
arr2 = np.random.randint(-50,50,8)
print(arr1,"\n")
print(arr2)
```

```
[ 39 -37  39 -20   3   3  17 -30]

[  2   4 -18  -6  45 -15  20 -12]
```

[3]:
```
# numpy ile matematik işlemleri + - * /
print(arr1 + arr2,"\n")
print(arr1 - arr2,"\n")
print(arr1 + 10,"\n")
print(arr1 * arr2,"\n")
print(arr1 * 10,"\n")
print(arr1 / arr2,"\n")
print(arr1 / 10,"\n")
```

```
[ 41 -33  21 -26  48 -12  37 -42]

[ 37 -41  57 -14 -42  18  -3 -18]

[ 49 -27  49 -10  13  13  27 -20]

[  78 -148 -702  120  135  -45  340  360]

[ 390 -370  390 -200   30   30  170 -300]

[19.5        -9.25       -2.16666667  3.33333333  0.06666667 -0.2
  0.85        2.5       ]

[ 3.9 -3.7  3.9 -2.   0.3  0.3  1.7 -3. ]
```

[4]:
```
# numpy ile ileri matematik işlemleri (cos tan kökalma power üs alma gibi␣
↪matematiksel işlemler)
```

```
print(np.sin(arr1),"\n")
print(np.cos(arr1),"\n")
print(np.abs(arr1),"\n")
print(np.sqrt(np.abs(arr1)),"\n")
print(np.log(np.abs(arr1)),"\n")
print(np.power(arr1,2),"\n")
```

```
[ 0.96379539  0.64353813  0.96379539 -0.91294525  0.14112001  0.14112001
 -0.96139749  0.98803162]

[ 0.26664293  0.76541405  0.26664293  0.40808206 -0.9899925  -0.9899925
 -0.27516334  0.15425145]

[39 37 39 20  3  3 17 30]

[6.244998   6.08276253 6.244998   4.47213595 1.73205081 1.73205081
 4.12310563 5.47722558]

[3.66356165 3.61091791 3.66356165 2.99573227 1.09861229 1.09861229
 2.83321334 3.40119738]

[1521 1369 1521  400    9    9  289  900]
```

[5]:
```
print(arr1,"\n")
print(arr2,"\n")
#print("2 diziyi dikey olarak birleştirme işlemi \n",np.concatenate((arr1,␣
 ↪arr2),axis=1),"\n")
print("2 diziyi yatay olarak birleştirme işlemi \n",np.concatenate((arr1,␣
 ↪arr2)),"\n")
print("2 diziyi dikey olarak birleştirme işlemi \n",np.vstack((arr1,arr2)),"\n")
print("2 diziyi yatay olarak birleştirme işlemi \n",np.hstack((arr1,arr2)))

print("2 diziyi karşılık gelen sütunları ile birleştirme işlemi \n",np.
 ↪stack((arr1, arr2), axis=1))
```

```
[ 39 -37  39 -20   3   3  17 -30]

[  2   4 -18  -6  45 -15  20 -12]

2 diziyi yatay olarak birleştirme işlemi
 [ 39 -37  39 -20   3   3  17 -30   2   4 -18  -6  45 -15  20 -12]

2 diziyi dikey olarak birleştirme işlemi
 [[ 39 -37  39 -20   3   3  17 -30]
 [  2   4 -18  -6  45 -15  20 -12]]

2 diziyi yatay olarak birleştirme işlemi
```

```
 [ 39 -37  39 -20   3   3  17 -30   2   4 -18  -6  45 -15  20 -12]
2 diziyi karşılık gelen sütunları ile birleştirme işlemi
 [[ 39   2]
 [-37   4]
 [ 39 -18]
 [-20  -6]
 [  3  45]
 [  3 -15]
 [ 17  20]
 [-30 -12]]
```

0.1

[6]:
```
newarray = np.array([np.random.randint(-20,20,5),np.random.randint(-20,20,5),np.
↪random.randint(-20,20,5)])
newarray.sort() # diziyi sıralayı o şekilde bastırdık
newarray
```

[6]:
```
array([[-12,  -2,   0,  19,  19],
       [-15, -13,   6,  11,  14],
       [  0,   3,   4,   9,  14]])
```

[7]:
```
sonuc = newarray > 0 # dizinin içerisindekilerinin 0 dan büyük olanlarını␣
↪tespit etme
sonuc
```

[7]:
```
array([[False, False, False,  True,  True],
       [False, False,  True,  True,  True],
       [False,  True,  True,  True,  True]])
```

[8]:
```
newarray[newarray > 0] # ve 0 dan büyük olan elemanlar
```

[8]:
```
array([19, 19,  6, 11, 14,  3,  4,  9, 14])
```

[9]:
```
newarray[newarray % 2 == 0] # dizi içerisinde 2 ye tam bölünenler yani çift␣
↪olanlar
```

[9]:
```
array([-12,  -2,   0,   6,  14,   0,   4,  14])
```

[ ]:
```

```