In [1]:
```
import numpy as np
```
In [2]:
```
array1 = np.array([1,2,3])
```
In [3]:
```
array1
```
Out[3]:
```
array([1, 2, 3])
```

numpy arrays

In [4]:
```
myList = [10,20,30]
```
In [5]:
```
myList
```
Out[5]:
```
[10, 20, 30]
```
In [6]:
```
np.array(myList)
```
Out[6]:
```
array([10, 20, 30])
```
In [8]:
```
type(np.array(myList))
```
Out[8]:
```
numpy.ndarray
```
In [9]:
```
 matrixList=[[10,20,30],[20,30,40],[30,40,50]]
```
In [10]:
```
np.array(matrixList)
```
Out[10]:
```
array([[10, 20, 30],
       [20, 30, 40],
       [30, 40, 50]])
```

arange

In [12]:
```
list(range(0,10))
```
Out[12]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```
In [13]:
```
np.arange(0,10)
```
Out[13]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```
In [14]:
```
np.arange(0,10,2)
```
Out[14]:
```
array([0, 2, 4, 6, 8])
```

zeros

In [15]:
```
np.zeros(5)
```
Out[15]:
```
array([0., 0., 0., 0., 0.])
```
In [16]:
```
np.zeros((5,5))
```
Out[16]:
```
array([[0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.]])
```
In [17]:
```
np.ones(5)
```
Out[17]:
```
array([1., 1., 1., 1., 1.])
```
In [19]:
```
np.ones((5,5))
```
Out[19]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

linspace

In [20]:
```
np.linspace(0,20,5)
```
Out[20]:
```
array([ 0.,  5., 10., 15., 20.])
```
In [21]:
```
np.linspace(0,10,6)
```
Out[21]:
```
array([ 0.,  2.,  4.,  6.,  8., 10.])
```
In [22]:
```
np.linspace(0,10,20)
```
Out[22]:
```
array([ 0.        ,  0.52631579,  1.05263158,  1.57894737,  2.10526316,
        2.63157895,  3.15789474,  3.68421053,  4.21052632,  4.73684211,
        5.26315789,  5.78947368,  6.31578947,  6.84210526,  7.36842105,
        7.89473684,  8.42105263,  8.94736842,  9.47368421, 10.        ])
```
In [23]:
```
np.linspace(0,100,100)
```
Out[23]:
```
array([  0.        ,   1.01010101,   2.02020202,   3.03030303,
         4.04040404,   5.05050505,   6.06060606,   7.07070707,
         8.08080808,   9.09090909,  10.1010101 ,  11.11111111,
        12.12121212,  13.13131313,  14.14141414,  15.15151515,
        16.16161616,  17.17171717,  18.18181818,  19.19191919,
        20.2020202 ,  21.21212121,  22.22222222,  23.23232323,
        24.24242424,  25.25252525,  26.26262626,  27.27272727,
        28.28282828,  29.29292929,  30.3030303 ,  31.31313131,
        32.32323232,  33.33333333,  34.34343434,  35.35353535,
        36.36363636,  37.37373737,  38.38383838,  39.39393939,
        40.4040404 ,  41.41414141,  42.42424242,  43.43434343,
        44.44444444,  45.45454545,  46.46464646,  47.47474747,
        48.48484848,  49.49494949,  50.50505051,  51.51515152,
        52.52525253,  53.53535354,  54.54545455,  55.55555556,
        56.56565657,  57.57575758,  58.58585859,  59.5959596 ,
        60.60606061,  61.61616162,  62.62626263,  63.63636364,
        64.64646465,  65.65656566,  66.66666667,  67.67676768,
        68.68686869,  69.6969697 ,  70.70707071,  71.71717172,
        72.72727273,  73.73737374,  74.74747475,  75.75757576,
        76.76767677,  77.77777778,  78.78787879,  79.7979798 ,
        80.80808081,  81.81818182,  82.82828283,  83.83838384,
        84.84848485,  85.85858586,  86.86868687,  87.87878788,
        88.88888889,  89.8989899 ,  90.90909091,  91.91919192,
        92.92929293,  93.93939394,  94.94949495,  95.95959596,
        96.96969697,  97.97979798,  98.98989899, 100.        ])
```
In [25]:
```
np.eye(3)
```
Out[25]:
```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

random

In [26]:
```
np.random.randn(8)
```
Out[26]:
```
array([ 0.15059938,  1.01210271,  0.20818652,  1.28189203,  0.42076249,
       -2.7362678 , -1.81547831,  1.80746059])
```
In [28]:
```
np.random.rand(4,4)
```
Out[28]:
```
array([[0.23632703, 0.34040044, 0.02387875, 0.55236699],
       [0.80247088, 0.34191326, 0.54786621, 0.10135807],
       [0.44161178, 0.91530823, 0.60029383, 0.21935776],
       [0.61899461, 0.25784659, 0.92695548, 0.68196721]])
```
In [34]:
```
np.random.randint(1,10)
```
Out[34]:
```
5
```
In [36]:
```
np.random.randint(0,10,4)
```
Out[36]:
```
array([7, 3, 7, 0])
```
In [37]:
```
np.random.randint(0,300,5)
```
Out[37]:
```
array([ 72,  59, 257,   2, 229])
```
In [38]:
```
myNumpyList=np.arange(30)
```
In [39]:
```
myNumpyList
```
Out[39]:
```
array([ 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, 25, 26, 27, 28, 29])
```
In [40]:
```
myRandomList=np.random.randint(0,100,30)
```
In [41]:
```
myRandomList
```
Out[41]:
```
array([99, 41, 62, 57, 92, 58, 75, 38, 12, 72,  9,  7, 93, 14, 48,  2, 19,
       80, 45,  7,  6, 97, 66, 91, 82, 14, 82, 47, 74, 19])
```

numpy array's methods

In [42]:
```
myNumpyList.reshape(5,6)
```
Out[42]:
```
array([[ 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, 25, 26, 27, 28, 29]])
```
In [43]:
```
myNumpyList.reshape(6,5)
```
Out[43]:
```
array([[ 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],
       [25, 26, 27, 28, 29]])
```
In [44]:
```
myNumpyList.reshape(10,3)
```
Out[44]:
```
array([[ 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, 25, 26],
       [27, 28, 29]])
```
In [47]:
```
myNumpyList.max()
```
Out[47]:
```
29
```
In [48]:
```
myRandomList.max()
```
Out[48]:
```
99
```
In [49]:
```
myNumpyList.min()
```
Out[49]:
```
0
```
In [50]:
```
myRandomList.min()
```
Out[50]:
```
2
```
In [52]:
```
myNumpyList.argmax() #Kaçıncı indiste olduğunu döndürür
```
Out[52]:
```
29
```
In [53]:
```
myRandomList.argmax()
```
Out[53]:
```
0
```
In [54]:
```
myRandomList.argmin()
```
Out[54]:
```
15
```
In [55]:
```
myRandomList
```
Out[55]:
```
array([99, 41, 62, 57, 92, 58, 75, 38, 12, 72,  9,  7, 93, 14, 48,  2, 19,
       80, 45,  7,  6, 97, 66, 91, 82, 14, 82, 47, 74, 19])
```
In [56]:
```
myReshapeArray = myRandomList.reshape(5,6)
```
In [57]:
```
myReshapeArray
```
Out[57]:
```
array([[99, 41, 62, 57, 92, 58],
       [75, 38, 12, 72,  9,  7],
       [93, 14, 48,  2, 19, 80],
       [45,  7,  6, 97, 66, 91],
       [82, 14, 82, 47, 74, 19]])
```
In [58]:
```
myReshapeArray.shape
```
Out[58]:
```
(5, 6)
```

indexs

In [61]:
```
myArray = np.arange(0,15)
```
In [62]:
```
myArray
```
Out[62]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```
In [63]:
```
myArray[5]
```
Out[63]:
```
5
```
In [64]:
```
myArray[3:8]
```
Out[64]:
```
array([3, 4, 5, 6, 7])
```
In [65]:
```
myArray[3:8]=-5
```
In [66]:
```
myArray
```
Out[66]:
```
array([ 0,  1,  2, -5, -5, -5, -5, -5,  8,  9, 10, 11, 12, 13, 14])
```
In [67]:
```
otherArray = np.arange(0,24)
```
In [70]:
```
otherArray
```
Out[70]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23])
```
In [71]:
```
slicingArray = otherArray[4:9]
```
In [73]:
```
slicingArray
```
Out[73]:
```
array([4, 5, 6, 7, 8])
```
In [74]:
```
otherArray
```
Out[74]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23])
```
In [75]:
```
slicingArray[::] = 61
```
In [76]:
```
slicingArray
```
Out[76]:
```
array([61, 61, 61, 61, 61])
```
In [77]:
```
otherArray
```
Out[77]:
```
array([ 0,  1,  2,  3, 61, 61, 61, 61, 61,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23])
```
In [78]:
```
# Kesitin alındığı dizi değişime uğradı bunun olmaması için
```
In [79]:
```
sampleArray = np.arange(0,24)
```
In [80]:
```
sampleArray
```
Out[80]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23])
```
In [81]:
```
sampleArrayCopy = sampleArray.copy()
```
In [82]:
```
sampleArrayCopy
```
Out[82]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23])
```
In [93]:
```
slicedSampleArray = sampleArrayCopy[4:8]
```
In [94]:
```
slicedSampleArray
```
Out[94]:
```
array([4, 5, 6, 7])
```
In [95]:
```
slicedSampleArray[:] = 61
```
In [96]:
```
slicedSampleArray
```
Out[96]:
```
array([61, 61, 61, 61])
```
In [98]:
```
sampleArrayCopy
```
Out[98]:
```
array([ 0,  1,  2,  3, 61, 61, 61, 61,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23])
```
In [99]:
```
sampleArray
```
Out[99]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23])
```

matrix indexs

In [108]:
```
myList = [[10,20,30],[20,30,40],[40,50,60]]
```
In [109]:
```
myMatrixArray = np.array(myList)
```
In [110]:
```
myMatrixArray
```
Out[110]:
```
array([[10, 20, 30],
       [20, 30, 40],
       [40, 50, 60]])
```
In [111]:
```
myMatrixArray[0]
```
Out[111]:
```
array([10, 20, 30])
```
In [112]:
```
myMatrixArray[1][2]
```
Out[112]:
```
40
```
In [113]:
```
myMatrixArray[1,2]
```
Out[113]:
```
40
```
In [114]:
```
myMatrixArray[1:,2]
```
Out[114]:
```
array([40, 60])
```
In [115]:
```
myMatrixArray[2:,2]
```
Out[115]:
```
array([60])
```
In [116]:
```
myMatrixArray[0:,2]
```
Out[116]:
```
array([30, 40, 60])
```
In [117]:
```
myMatrixArray[0:,0]
```
Out[117]:
```
array([10, 20, 40])
```
In [123]:
```
myNewMatrix = np.arange(0,25)
```
In [124]:
```
myNewMatrix
```
Out[124]:
```
array([ 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])
```
In [133]:
```
myReshapeMatrix = myNewMatrix.reshape(5,5)
```
In [138]:
```
myReshapeMatrix
```
Out[138]:
```
array([[ 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]])
```
In [140]:
```
myReshapeMatrix[0]
```
Out[140]:
```
array([0, 1, 2, 3, 4])
```
In [141]:
```
myReshapeMatrix[[0,2,4]]
```
Out[141]:
```
array([[ 0,  1,  2,  3,  4],
       [10, 11, 12, 13, 14],
       [20, 21, 22, 23, 24]])
```
In [143]:
```
myReshapeMatrix[[4,3,0]]
```
Out[143]:
```
array([[20, 21, 22, 23, 24],
       [15, 16, 17, 18, 19],
       [ 0,  1,  2,  3,  4]])
```

Numpy Operations

In [153]:
```
newArray = np.random.randint(1,100,20)
```
In [154]:
```
newArray
```
Out[154]:
```
array([ 8, 72, 34, 93, 52, 40, 67, 70, 56, 42, 94, 11, 58, 26, 21, 45, 21,
       55, 89, 48])
```
In [155]:
```
newArray > 24
```
Out[155]:
```
array([False,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True, False,  True,  True, False,  True, False,  True,
        True,  True])
```
In [156]:
```
newArray
```
Out[156]:
```
array([ 8, 72, 34, 93, 52, 40, 67, 70, 56, 42, 94, 11, 58, 26, 21, 45, 21,
       55, 89, 48])
```
In [157]:
```
resultArray = newArray > 24
```
In [158]:
```
resultArray
```
Out[158]:
```
array([False,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True, False,  True,  True, False,  True, False,  True,
        True,  True])
```
In [159]:
```
newArray[resultArray]
```
Out[159]:
```
array([72, 34, 93, 52, 40, 67, 70, 56, 42, 94, 58, 26, 45, 55, 89, 48])
```
In [160]:
```
newArray[newArray > 24]
```
Out[160]:
```
array([72, 34, 93, 52, 40, 67, 70, 56, 42, 94, 58, 26, 45, 55, 89, 48])
```
In [161]:
```
endArray = np.arange(0,24)
```
In [162]:
```
endArray
```
Out[162]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23])
```
In [163]:
```
endArray + endArray
```
Out[163]:
```
array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32,
       34, 36, 38, 40, 42, 44, 46])
```
In [164]:
```
endArray - endArray
```
Out[164]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0])
```
In [165]:
```
endArray / endArray
```
```
<ipython-input-165-9a854cacbf13>:1: RuntimeWarning: invalid value encountered in true_divide
  endArray / endArray
```
Out[165]:
```
array([nan,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.])
```
In [166]:
```
np.sqrt(endArray)
```
Out[166]:
```
array([0.        , 1.        , 1.41421356, 1.73205081, 2.        ,
       2.23606798, 2.44948974, 2.64575131, 2.82842712, 3.        ,
       3.16227766, 3.31662479, 3.46410162, 3.60555128, 3.74165739,
       3.87298335, 4.        , 4.12310563, 4.24264069, 4.35889894,
       4.47213595, 4.58257569, 4.69041576, 4.79583152])
```
In [167]:
```
np.max(endArray)
```
Out[167]:
```
23
```
In [168]:
```
np.min(endArray)
```
Out[168]:
```
0
```
In [ ]:
```

```