In [1]:

```
import numpy as np
```

In [2]:

```
# numeric python
# numpy is 50 times faster than list of python
# it is faster b'cz most of numpy codes are written in C++, c.
```

In [3]:

```
# you can create multidimensional array or matrix using numpy
```

In [4]:

```
# to create multidimensional array numpy uses a ndarray object.
```

In [5]:

```
# how to craete Arrays in numpy
```

In [6]:

```
# 1D array
```

In [7]:

```
arr = np.array([10,20,25,20,56])
arr
```

Out[7]:

```
array([10, 20, 25, 20, 56])
```

In [8]:

```
arr2 = np.array((10,20,25,20,56))
arr2
```

Out[8]:

```
array([10, 20, 25, 20, 56])
```

In [9]:

```
arr.ndim
```

Out[9]:

```
1
```

In [10]:

```
arr.size   # no. of elements in your array
```

Out[10]:

```
5
```

In [11]:

```
# 2D array
```

In [12]:

```
arr3 =np.array([[10,20,30],[40,50,60]])
arr3
```

Out[12]:

```
array([[10, 20, 30],
       [40, 50, 60]])
```

In [13]:

```
arr3.shape
```

Out[13]:

```
(2, 3)
```

In [14]:

```
arr3.ndim
```

Out[14]:

```
2
```

In [15]:

```
arr4 =np.array([[10,20,30,56],[40,50,60,79], [12,87,46,99],[67,42,73,15]])
arr4
```

Out[15]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 87, 46, 99],
       [67, 42, 73, 15]])
```

In [16]:

```
arr4.ndim
```

Out[16]:

```
2
```

In [17]:

```
arr4.shape
```

Out[17]:

```
(4, 4)
```

In [18]:

```
#3D Array-: combination of 2D arrays
```

In [19]:

```
arr5 = np.array([[[10,20,30,56],[40,50,60,79],[12,78,90,80]],[[12,87,46,99],[67,42,73,15],[
arr5
```

Out[19]:

```
array([[[  10,   20,   30,   56],
        [  40,   50,   60,   79],
        [  12,   78,   90,   80]],

       [[  12,   87,   46,   99],
        [  67,   42,   73,   15],
        [1200,  500, 6000, 9000]]])
```

In [20]:

```
arr5.ndim
```

Out[20]:

```
3
```

In [21]:

```
arr5.shape
```

Out[21]:

```
(2, 3, 4)
```

In [22]:

```
# here 2 is showing no. of planes or no. of 2D arrays, 3 is showing number of rows in each
# 4 is showing no. of columns in each 2D array
```

In [23]:

```
arr5.size
```

Out[23]:

```
24
```

In [24]:

```
arr4.size
```

Out[24]:

```
16
```

In [25]:

```
arr6 = np.linspace(30, 100, num=16, endpoint=True)
arr6
```

Out[25]:

```
array([ 30.        ,  34.66666667,  39.33333333,  44.        ,
        48.66666667,  53.33333333,  58.        ,  62.66666667,
        67.33333333,  72.        ,  76.66666667,  81.33333333,
        86.        ,  90.66666667,  95.33333333, 100.        ])
```

In [26]:

```
arr6 = np.linspace(30, 100, num=16, endpoint=False, dtype=int)
arr6
```

Out[26]:

```
array([30, 34, 38, 43, 47, 51, 56, 60, 65, 69, 73, 78, 82, 86, 91, 95])
```

In [27]:

```
arr6 = np.linspace(30, 100, num=16, endpoint=False, dtype=str)
arr6
```

Out[27]:

```
array(['30.0', '34.375', '38.75', '43.125', '47.5', '51.875', '56.25',
       '60.625', '65.0', '69.375', '73.75', '78.125', '82.5', '86.875',
       '91.25', '95.625'], dtype='<U32')
```

In [28]:

```
arr7 = np.random.random()
arr7
```

Out[28]:

```
0.6250023936713537
```

In [29]:

```
arr8 = np.random.randint(20,80,size=15,dtype=int)
arr8
```

Out[29]:

```
array([34, 61, 33, 50, 40, 20, 35, 51, 57, 45, 41, 25, 37, 41, 39])
```

In [30]:

```
arr9 = np.random.randint(20,80,size=(3,4),dtype=int)
arr9
```

Out[30]:

```
array([[79, 36, 49, 51],
       [39, 59, 23, 29],
       [68, 31, 67, 35]])
```

In [31]:

```
arr10 = np.random.randint(20,80,size=(3,4,4),dtype=int)
arr10
```

Out[31]:

```
array([[[24, 72, 59, 64],
        [27, 58, 73, 61],
        [45, 41, 60, 66],
        [72, 76, 28, 72]],

       [[23, 78, 60, 28],
        [77, 33, 40, 44],
        [65, 67, 63, 69],
        [68, 38, 67, 58]],

       [[73, 38, 22, 49],
        [77, 49, 48, 24],
        [29, 49, 64, 47],
        [77, 41, 53, 66]]])
```

In [32]:

```
arr11 = np.arange(10,26).reshape(4,4)
arr11
```

Out[32]:

```
array([[10, 11, 12, 13],
       [14, 15, 16, 17],
       [18, 19, 20, 21],
       [22, 23, 24, 25]])
```

In [33]:

```
arr12 =np.arange(100,130,2).reshape(5,3)
arr12
```

Out[33]:

```
array([[100, 102, 104],
       [106, 108, 110],
       [112, 114, 116],
       [118, 120, 122],
       [124, 126, 128]])
```

slicing in numpy array

In [34]:

```
# 2d array slicing-:
```

[start:stop:step for row,start:stop:step for column, step]

In [35]:

```
arr12[2:4, 0:2]
```

Out[35]:

```
array([[112, 114],
       [118, 120]])
```

In [36]:

```
arr4
```

Out[36]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 87, 46, 99],
       [67, 42, 73, 15]])
```

In [37]:

```
arr4[:3, 1:]
```

Out[37]:

```
array([[20, 30, 56],
       [50, 60, 79],
       [87, 46, 99]])
```

In [38]:

```
arr5
```

Out[38]:

```
array([[[  10,   20,   30,   56],
        [  40,   50,   60,   79],
        [  12,   78,   90,   80]],

       [[  12,   87,   46,   99],
        [  67,   42,   73,   15],
        [1200,  500, 6000, 9000]]])
```

In [39]:

```
# 3d array slicing
```

[start:stop:step for planes,start:stop:step for row,start:stop:step for column]

In [40]:

```
arr5[1:2,:,:2]
```

Out[40]:

```
array([[[  12,   87],
        [  67,   42],
        [1200,  500]]])
```

In [41]:

```
arr10
```

Out[41]:

```
array([[[24, 72, 59, 64],
        [27, 58, 73, 61],
        [45, 41, 60, 66],
        [72, 76, 28, 72]],

       [[23, 78, 60, 28],
        [77, 33, 40, 44],
        [65, 67, 63, 69],
        [68, 38, 67, 58]],

       [[73, 38, 22, 49],
        [77, 49, 48, 24],
        [29, 49, 64, 47],
        [77, 41, 53, 66]]])
```

In [42]:

```
arr10[1:, 1:,:]
```

Out[42]:

```
array([[[77, 33, 40, 44],
        [65, 67, 63, 69],
        [68, 38, 67, 58]],

       [[77, 49, 48, 24],
        [29, 49, 64, 47],
        [77, 41, 53, 66]]])
```

In [43]:

```
arr4
```

Out[43]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 87, 46, 99],
       [67, 42, 73, 15]])
```

In [44]:

```
arr4
```

Out[44]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 87, 46, 99],
       [67, 42, 73, 15]])
```

```

```

In [45]:

```
arr10
```

Out[45]:

```
array([[[24, 72, 59, 64],
        [27, 58, 73, 61],
        [45, 41, 60, 66],
        [72, 76, 28, 72]],

       [[23, 78, 60, 28],
        [77, 33, 40, 44],
        [65, 67, 63, 69],
        [68, 38, 67, 58]],

       [[73, 38, 22, 49],
        [77, 49, 48, 24],
        [29, 49, 64, 47],
        [77, 41, 53, 66]]])
```

In [46]:

```
arr4
```

Out[46]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 87, 46, 99],
       [67, 42, 73, 15]])
```

In [47]:

```
np.sort(arr4, axis=1)  # sorts an array row wise
```

Out[47]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 46, 87, 99],
       [15, 42, 67, 73]])
```

In [48]:

```
np.sort(arr4, axis=0)  # sorts an array column wise
```

Out[48]:

```
array([[10, 20, 30, 15],
       [12, 42, 46, 56],
       [40, 50, 60, 79],
       [67, 87, 73, 99]])
```

In [49]:

```
arr4
```

Out[49]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 87, 46, 99],
       [67, 42, 73, 15]])
```

In [50]:

```
np.sort(arr4, axis=-1)
```

Out[50]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 46, 87, 99],
       [15, 42, 67, 73]])
```

In [52]:

```
%%time
np.sort(arr4, axis=0, kind='quicksort')
```

```
Wall time: 0 ns
```

Out[52]:

```
array([[10, 20, 30, 15],
       [12, 42, 46, 56],
       [40, 50, 60, 79],
       [67, 87, 73, 99]])
```

In [55]:

```
%%time
np.sort(arr4, axis=0)  # sorts an array column wise
```

```
7.96 µs ± 580 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)
Wall time: 6.46 s
```

In [56]:

```
arr10
```

Out[56]:

```
array([[[24, 72, 59, 64],
        [27, 58, 73, 61],
        [45, 41, 60, 66],
        [72, 76, 28, 72]],

       [[23, 78, 60, 28],
        [77, 33, 40, 44],
        [65, 67, 63, 69],
        [68, 38, 67, 58]],

       [[73, 38, 22, 49],
        [77, 49, 48, 24],
        [29, 49, 64, 47],
        [77, 41, 53, 66]]])
```

In [57]:

```
arr10.max()
```

Out[57]:

```
78
```

In [58]:

```
arr10.min()
```

Out[58]:

```
22
```

In [59]:

```
arr10.mean()
```

Out[59]:

```
53.791666666666664
```

In [60]:

```
arr10.sum()
```

Out[60]:

```
2582
```

In [62]:

```
arr10
```

Out[62]:

```
array([[[24, 72, 59, 64],
        [27, 58, 73, 61],
        [45, 41, 60, 66],
        [72, 76, 28, 72]],

       [[23, 78, 60, 28],
        [77, 33, 40, 44],
        [65, 67, 63, 69],
        [68, 38, 67, 58]],

       [[73, 38, 22, 49],
        [77, 49, 48, 24],
        [29, 49, 64, 47],
        [77, 41, 53, 66]]])
```

In [63]:

```
arr10.sum(axis=1)  # column wise
```

Out[63]:

```
array([[168, 247, 220, 263],
       [233, 216, 230, 199],
       [256, 177, 187, 186]])
```

In [64]:

```
arr10.sum(axis=0)
```

Out[64]:

```
array([[120, 188, 141, 141],
       [181, 140, 161, 129],
       [139, 157, 187, 182],
       [217, 155, 148, 196]])
```

In [65]:

```
arr4
```

Out[65]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 87, 46, 99],
       [67, 42, 73, 15]])
```

In [67]:

```
arr4.sum(axis=-1)  # row wise sum
```

Out[67]:

```
array([116, 229, 244, 197])
```

In [69]:

```
arr4.sum(axis=0)  # column wise sum
```

Out[69]:

```
array([129, 199, 209, 249])
```

In [70]:

```
arr4.mean(axis=-1)  # row wise mean
```

Out[70]:

```
array([29.  , 57.25, 61.  , 49.25])
```

In [71]:

```
arr4.mean(axis=0) # column wise mean
```

Out[71]:

```
array([32.25, 49.75, 52.25, 62.25])
```

In [72]:

```
arr4
```

Out[72]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 87, 46, 99],
       [67, 42, 73, 15]])
```

In [73]:

```
arr11
```

Out[73]:

```
array([[10, 11, 12, 13],
       [14, 15, 16, 17],
       [18, 19, 20, 21],
       [22, 23, 24, 25]])
```

In [74]:

```
arr4+arr11
```

Out[74]:

```
array([[ 20,  31,  42,  69],
       [ 54,  65,  76,  96],
       [ 30, 106,  66, 120],
       [ 89,  65,  97,  40]])
```

In [75]:

```
arr4*arr11
```

Out[75]:

```
array([[ 100,  220,  360,  728],
       [ 560,  750,  960, 1343],
       [ 216, 1653,  920, 2079],
       [1474,  966, 1752,  375]])
```

In [76]:

```
arr4-arr11
```

Out[76]:

```
array([[  0,   9,  18,  43],
       [ 26,  35,  44,  62],
       [ -6,  68,  26,  78],
       [ 45,  19,  49, -10]])
```

In [77]:

```
arr11
```

Out[77]:

```
array([[10, 11, 12, 13],
       [14, 15, 16, 17],
       [18, 19, 20, 21],
       [22, 23, 24, 25]])
```

conditinal syntax-: np.where(condition, True_Result, False_Result)

In [78]:

```
res = np.where(arr11>=20,arr11,0)
```

In [79]:

```
res
```

Out[79]:

```
array([[ 0,  0,  0,  0],
       [ 0,  0,  0,  0],
       [ 0,  0, 20, 21],
       [22, 23, 24, 25]])
```

In [80]:

```
arr11
```

Out[80]:

```
array([[10, 11, 12, 13],
       [14, 15, 16, 17],
       [18, 19, 20, 21],
       [22, 23, 24, 25]])
```

In [81]:

```
res = np.where(arr11%2==0,"Even","Odd" )
res
```

Out[81]:

```
array([['Even', 'Odd', 'Even', 'Odd'],
       ['Even', 'Odd', 'Even', 'Odd'],
       ['Even', 'Odd', 'Even', 'Odd'],
       ['Even', 'Odd', 'Even', 'Odd']], dtype='<U4')
```

In [89]:

```
arr11.transpose()
```

Out[89]:

```
array([[10, 14, 18, 22],
       [11, 15, 19, 23],
       [12, 16, 20, 24],
       [13, 17, 21, 25]])
```

In [90]:

```
# hstack
# vstack
```

In [91]:

```
arr4
```

Out[91]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 87, 46, 99],
       [67, 42, 73, 15]])
```

In [92]:

```
arr11
```

Out[92]:

```
array([[10, 11, 12, 13],
       [14, 15, 16, 17],
       [18, 19, 20, 21],
       [22, 23, 24, 25]])
```

In [94]:

```
arr16 = np.hstack((arr4,arr11))
arr16
```

Out[94]:

```
array([[10, 20, 30, 56, 10, 11, 12, 13],
       [40, 50, 60, 79, 14, 15, 16, 17],
       [12, 87, 46, 99, 18, 19, 20, 21],
       [67, 42, 73, 15, 22, 23, 24, 25]])
```

In [95]:

```
arr17 = np.vstack((arr4,arr11))
arr17
```

Out[95]:

```
array([[10, 20, 30, 56],
       [40, 50, 60, 79],
       [12, 87, 46, 99],
       [67, 42, 73, 15],
       [10, 11, 12, 13],
       [14, 15, 16, 17],
       [18, 19, 20, 21],
       [22, 23, 24, 25]])
```

In [96]:

```
arr18 = np.append(arr17, [100,200,300,400])
arr18
```

Out[96]:

```
array([ 10,  20,  30,  56,  40,  50,  60,  79,  12,  87,  46,  99,  67,
        42,  73,  15,  10,  11,  12,  13,  14,  15,  16,  17,  18,  19,
        20,  21,  22,  23,  24,  25, 100, 200, 300, 400])
```

In [ ]:

```

```