

```
In [6]: import numpy as np

In [7]: one_dim= np.array([1,2,3,8,9,10])
         one_dim

Out[7]: array([ 1,  2,  3,  8,  9, 10])

In [8]: type(one_dim)

Out[8]: numpy.ndarray

In [9]: one_dim.dtype

Out[9]: dtype('int32')

In [10]: one_dim.size

Out[10]: 6

In [11]: one_dim.ndim

Out[11]: 1

In [41]: summation =one_dim + one_dim
         summation

Out[41]: array([ 2,  4,  6, 16, 18, 20])

In [42]: one_dim.sum()

Out[42]: 33

In [36]: difference= summation-one_dim
         difference

Out[36]: array([ 1,  2,  3,  8,  9, 10])

In [37]: summation/difference

Out[37]: array([2., 2., 2., 2., 2., 2.])

In [21]: two_dim=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12]])
         two_dim

Out[21]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [ 7,  8,  9],
                [10, 11, 12]])

In [22]: type(two_dim)

Out[22]: numpy.ndarray

In [23]: two_dim.dtype

Out[23]: dtype('int32')

In [24]: two_dim.size

Out[24]: 12

In [25]: two_dim.ndim

Out[25]: 2

In [109]: two_dim.shape

Out[109]: (4, 3)

In [111]: two_dim>2

Out[111]: array([[False, False,  True],
                [ True,  True,  True],
                [ True,  True,  True],
                [ True,  True,  True]])

In [112]: two_dim[two_dim>2]

Out[112]: array([ 3,  4,  5,  6,  7,  8,  9, 10, 11, 12])

In [113]: dim= two_dim.reshape(3,4)
         dim

Out[113]: array([[ 1,  2,  3,  4],
                [ 5,  6,  7,  8],
                [ 9, 10, 11, 12]])

In [122]: dim[0:2,0:2]

Out[122]: array([[1, 2],
                [5, 6]])

In [31]: dim

Out[31]: array([[ 1,  2,  3,  4],
                [ 5,  6,  7,  8],
                [ 9, 10, 11, 12]])

In [40]: dim.sum() #for getting the sum of all the values in the matrices

Out[40]: 78

In [101]: onedimension= np.arange(1,11)
         onedimension

Out[101]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])

In [103]: onedimension.max()

Out[103]: 10

In [105]: onedimension.argmax()

Out[105]: 9

In [54]: onedimension= np.random.rand(3) #will display random numbers within the range [0,1] and can
         form 1-d also 2-d arrays
         onedimension

Out[54]: array([0.82451195, 0.3445223 , 0.32862238])

In [52]: np.random.randn(5,7) #display standard uniform distributed numbers and can form 1-d also 2-d
         arrays

Out[52]: array([[ -0.82766483, -0.20490145,  0.87166805,  1.88398702, -2.36366803,
                -0.71036401,  1.30774142],
                [-0.73744833,  0.35071451, -1.18309602,  0.53774749,  1.03892932,
                1.01684265,  0.88496393],
                [-0.13906342,  0.67308915,  1.67061142, -0.86142052,  0.99564187,
                -0.58458473, -1.18770708],
                [ 0.74264546,  1.75534207,  1.2031388 ,  1.01343905,  0.53530605,
                -1.01752532, -1.10043014],
                [ 0.89780728, -1.38447316,  0.11094169,  1.29145617,  1.49990212,
                0.20685089, -0.36569399]])

In [57]: random_array= np.array([1,2,3,4])
         random_array

Out[57]: array([1, 2, 3, 4])

In [64]: np.random.randint(1,200,10) #display integer selected element from the given range

Out[64]: array([ 85,  54, 181, 166, 186,  98,  83,  82,  82, 133])

In [84]: list1= [8,9,10,22,33]
         list1

Out[84]: [8, 9, 10, 22, 33]

In [94]: list1.reverse()

In [95]: print(list1)

[33, 22, 10, 9, 8]

In [96]: new_list= np.array(list1)
         new_list

Out[96]: array([33, 22, 10,  9,  8])

In [97]: new_list[1:4]

Out[97]: array([22, 10,  9])

In [98]: new_list[-4:-1]

Out[98]: array([22, 10,  9])

In [100]: arr= np.array([1,8,10,22,34,40])
         arr

Out[100]: array([ 1,  8, 10, 22, 34, 40])

In [106]: arr>8

Out[106]: array([False, False,  True,  True,  True,  True])

In [108]: arr[arr>8]

Out[108]: array([10, 22, 34, 40])

In [123]: np.ones(2)

Out[123]: array([1.,  1.])

In [141]: np.zeros((2,3))

Out[141]: array([[0.,  0.,  0.],
                [0.,  0.,  0.]])

In [145]: np.array([5.0]*10)

Out[145]: array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])

In [146]: np.array([5.0,5,5,5,5,5,5,5,5])

Out[146]: array([5., 5., 5., 5., 5., 5., 5., 5., 5.])

In [127]: np.eye(3)

Out[127]: array([[1.,  0.,  0.],
                [0.,  1.,  0.],
                [0.,  0.,  1.]])

In [128]: np.linspace(0,2,5)

Out[128]: array([0. , 0.5, 1. , 1.5, 2. ])

In [134]: u=np.array([1,2,3,4])
         z=u+1
         z

Out[134]: array([2, 3, 4, 5])

In [155]: #Like list where the values in the list is changeable so similarly the values in the array a
         re changeable
         arr1=np.arange(1,11)
         arr1

Out[155]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])

In [156]: slice_arr= arr1[0:6]
         slice_arr

Out[156]: array([1, 2, 3, 4, 5, 6])

In [158]: slice_arr[:]=99

In [159]: slice_arr

Out[159]: array([99, 99, 99, 99, 99, 99])

In [161]: arr1[0:6]=99 #you can't directly store the value,for doing so you need to store it in anothe
         r variable

In [162]: arr1

Out[162]: array([99, 99, 99, 99, 99, 99,  7,  8,  9, 10])

In [165]: arr_copy= arr1.copy()

In [166]: arr_copy

Out[166]: array([99, 99, 99, 99, 99, 99,  7,  8,  9, 10])

In [167]: arr_copy[:]=100

In [168]: arr_copy

Out[168]: array([100, 100, 100, 100, 100, 100, 100, 100, 100, 100])

In [169]: arr1

Out[169]: array([99, 99, 99, 99, 99, 99,  7,  8,  9, 10])

In [1]: # Some hands on practice on what we have learnt

In [5]: import numpy as np
         np.zeros(10)

Out[5]: array([0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.])

In [12]: np.ones(10)

Out[12]: array([1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.])

In [13]: np.arange(10,51)

Out[13]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
                27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
                44, 45, 46, 47, 48, 49, 50])

In [20]: np.arange(10,51,2)

Out[20]: array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
                44, 46, 48, 50])

In [22]: np.array([[0,1,2],[3,4,5],[6,7,8]]) #either create array manually or use arange and then res
         hape into 3x3 matrix

Out[22]: array([[0, 1, 2],
                [3, 4, 5],
                [6, 7, 8]])

In [25]: arr2= np.arange(0,9)
         arr2

Out[25]: array([0, 1, 2, 3, 4, 5, 6, 7, 8])

In [27]: arr2.reshape(3,3)

Out[27]: array([[0, 1, 2],
                [3, 4, 5],
                [6, 7, 8]])

In [82]: np.eye(3)

Out[82]: array([[1.,  0.,  0.],
                [0.,  1.,  0.],
                [0.,  0.,  1.]])

In [83]: np.linspace(0.01,1,100).reshape(10,10)

Out[83]: array([[0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1 ],
                [0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2 ],
                [0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3 ],
                [0.31, 0.32, 0.33, 0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4 ],
                [0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47, 0.48, 0.49, 0.5 ],
                [0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59, 0.6 ],
                [0.61, 0.62, 0.63, 0.64, 0.65, 0.66, 0.67, 0.68, 0.69, 0.7 ],
                [0.71, 0.72, 0.73, 0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.8 ],
                [0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.89, 0.9 ],
                [0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1.  ]])

In [42]: np.random.randint(0,2)

Out[42]: 1

In [43]: np.random.randn(25)

Out[43]: array([-1.7263445 ,  0.3306749 ,  0.44670751,  0.7095061 ,  1.36449319,
                -0.70283357, -1.28256349,  0.58635645,  0.50736266, -0.59365604,
                -0.65370078,  1.06180949, -0.51618547,  0.02168307,  0.04810287,
                -0.26883469,  1.15502223, -0.13591166, -2.29402323, -1.72585751,
                -0.7886436 ,  0.54359685,  1.20317272,  0.86392985, -1.04011236])

In [48]: np.linspace(0,1,20)

Out[48]: array([0. , 0.05263158, 0.10526316, 0.15789474, 0.21052632,
                0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,
                0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,
                0.78947368, 0.84210526, 0.89473684, 0.94736842, 1.  ])

In [ ]: #NumPy Indexing and slicing

In [53]: mat= np.arange(1,26).reshape(5,5)
         mat

Out[53]: array([[ 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]])

In [54]: mat[2:,1:]

Out[54]: array([[12, 13, 14, 15],
                [17, 18, 19, 20],
                [22, 23, 24, 25]])

In [55]: mat[1:,3:]

Out[55]: array([[ 9, 10],
                [14, 15],
                [19, 20],
                [24, 25]])

In [58]: mat[3][4]

Out[58]: 20

In [60]: mat[0:3,1:2]

Out[60]: array([[ 2],
                [ 7],
                [12]])

In [61]: mat[4:]

Out[61]: array([[21, 22, 23, 24, 25]])

In [62]: mat[3:]

Out[62]: array([[16, 17, 18, 19, 20],
                [21, 22, 23, 24, 25]])

In [63]: mat.sum()

Out[63]: 325

In [64]: mat.std()

Out[64]: 7.211102550927978
```