

Numpy

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

In [2]: #single dimension array
Single_Dim_array=np.array([1,2,3,4,5,6,7,8,9])
print(Single_Dim_array)

[1 2 3 4 5 6 7 8 9]

In [3]: type(Single_Dim_array)

Out[3]: numpy.ndarray

In [4]: SDim=Single_Dim_array.ndim
print("Single_Dim_array is",SDim,"dimesion array ",)

Single_Dim_array is 1 dimesion array

In [5]: #multi dimension array
Multi_Dim_array=np.array([[1,2,3],[4,5,6],[7,8,9]])
print(Multi_Dim_array)

[[1 2 3]
 [4 5 6]
 [7 8 9]]

In [6]: type(Multi_Dim_array)

Out[6]: numpy.ndarray

In [7]: MDim=Multi_Dim_array.ndim
print("Single_Dim_array is",MDim,"dimesion array ",)

Single_Dim_array is 2 dimesion array

In [8]: # numpy array with zeros
numpy_array_with_zeros=np.zeros((1, 3), dtype=int)
numpy_array_with_zeros

Out[8]: array([[0, 0, 0]])

In [9]: # numpy multi dimesion array with zeros
numpy_multi_dimesion_array_with_zeros=np.zeros((2, 5),dtype=int)
numpy_multi_dimesion_array_with_zeros

Out[9]: array([[0, 0, 0, 0, 0],
               [0, 0, 0, 0, 0]])

In [10]: # initilizing numpy array with same number
Same_number=np.full((3,4),50,dtype=int)
Same_number

Out[10]: array([[50, 50, 50, 50],
               [50, 50, 50, 50],
               [50, 50, 50, 50]])

In [11]: # initilizing array with range
Range=np.arange(1,20)
Range_

Out[11]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
                18, 19])

In [12]: # initilizing array with range including Skip
Range_Skip=np.arange(1,20,2)
Range_Skip

Out[12]: array([ 1,  3,  5,  7,  9, 11, 13, 15, 17, 19])

In [13]: # random values in numpy in array
random_values_in_numpy_in_array= np.random.randint(1,100,5)
# here range is one to zero and we select
print(random_values_in_numpy_in_array)

[56 15 20 35 88]

In [14]: # shapeChange
shapeChange1=np.array([[1,2,3],[4,5,6]])
shapeChange1.shape

Out[14]: (2, 3)

In [15]: shapeChange1

Out[15]: array([[1, 2, 3],
               [4, 5, 6]])

In [16]: shapeChange1.shape=(3,2)
shapeChange1.shape

Out[16]: (3, 2)

In [17]: shapeChange1

Out[17]: array([[1, 2],
               [3, 4],
               [5, 6]])

In [18]: #we can joint two numpy array
# 1) Vertical Stack
# 2) Horizontal Stack
# 3) Column Stack

# 1) Vertical Stack
Numpy_array_One=np.array([1,2,3,4,5,6])
Numpy_array_Two=np.array([6,5,4,3,2,1])
np.vstack((Numpy_array_One,Numpy_array_Two))

Out[18]: array([[1, 2, 3, 4, 5, 6],
               [6, 5, 4, 3, 2, 1]])

In [19]: # 2) Horizontal Stack
Numpy_array_one=np.array([1,2,3,4,5,6])
Numpy_array_two=np.array([6,5,4,3,2,1])
np.hstack((Numpy_array_one,Numpy_array_two))

Out[19]: array([1, 2, 3, 4, 5, 6, 6, 5, 4, 3, 2, 1])

In [20]: Numpy_Array_One=np.array([1,2,3,4,5,6])
Numpy_Array_Two=np.array([6,5,4,3,2,1])
np.column_stack((Numpy_Array_One,Numpy_Array_Two))

Out[20]: array([[1, 6],
               [2, 5],
               [3, 4],
               [4, 3],
               [5, 2],
               [6, 1]])

In [21]: # intersection and difference

In [22]: Intersect1=np.array([1,4,5,9,7,6,3,2])
Intersect2=np.array([10,45,2,5,56,7,12,47,4,6,7,78,3,97,9])
np.intersect1d(Intersect1,Intersect2)

Out[22]: array([2, 3, 4, 5, 6, 7, 9])

In [23]: Intersect3=np.array([1,4,5,9,7,6,3,2])
Intersect4=np.array([10,45,2,5,56,7,12,47,4,6,7,78,3,97,9])
np.setdiff1d(Intersect4,Intersect3)

Out[23]: array([10, 12, 45, 47, 56, 78, 97])

In [28]: # adition of numpy array
sum1=np.array([25,35,36,37])
sum2=np.array([25,41,78,53])
Sum=np.sum([sum1,sum2])
print(Sum)

330

In [33]: # addition row
Sum1=np.array([10,20])
Sum2=np.array([30,40])
sumv=np.sum([Sum1,Sum2],axis=0)
print(sumv)

[40 60]

In [35]: # addition col
Sum1=np.array([10,20])
Sum2=np.array([30,40])
sumv=np.sum([Sum1,Sum2],axis=1)
print(sumv)

[30 70]

In [42]: #numpy array mathmetics
nsum=np.array([10,20,30])
# sum
nsum=nsum+2
print(nsum)

[12 22 32]

In [43]: #substraction
nsum=nsum-2
print(nsum)

[10 20 30]

In [44]: # multiplication
# nsum=nsum*2
# print(nsum)

In [45]: # division
# nsum=nsum/2
# print(nsum)

In [47]: M=np.array([10,20,30,40,50,60,70,80])

In [48]: # mean
np.mean(M)

Out[48]: 45.0

In [49]: #median
np.median(M)

Out[49]: 45.0

In [51]: # std Dev
np.std(M)

Out[51]: 22.9128784747792

In [58]: # save Numpy
Table=np.array([2,4,6,8,10,12,14,6,18,20])
np.save('table',Table)

In [59]: # Loading numpy
Table20=np.load('table.npy')
Table20

Out[59]: array([ 2,  4,  6,  8, 10, 12, 14,  6, 18, 20])
```