```
pip install numpy # is used to install numpy library in Python

import numpy as np # to import numpy in python program
```

# Creating NUMPY ARRAYS

```
a1=np.array([1,2,3,4]) # we convert a list into an array/ or create an array using this syntax

print(a1) # Printing of the array. Compare the difference between print of an array and that of the list

[1 2 3 4]

type(a1) # Displays the type of the Variable a1: array

numpy.ndarray

import os # To check our Current Working Directory
os.getcwd()

a2=np.array([1.1,10.3,4.6,7,8],dtype='float')# dtype can be int/float etc. dtype helps define the data type of an array

a2 # This is how float data type array is displayed

array([ 1.1, 10.3,  4.6,  7. ,  8. ])
```

## arange() function

```
a3=np.arange(1,10)# just like range function, but "arange" in NUMPY-ARRAY

a3 # due to arange function we created an array

array([1, 2, 3, 4, 5, 6, 7, 8, 9])

a4=np.arange(11) # this creates an array from 0 to 10

a4

array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10])

a5=np.arange(1,11,2) # hopper:2 creates an array of odds from 1 to 9

a5

array([1, 3, 5, 7, 9])

a6=np.arange(10,0,-1) # this way, we create an array of descending order from 10 to 1
```

```
a6
```

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

## random Simulations

```
random_array=np.random.randn(5,4) # it create an array shape(5,4) with random values
print(random_array)
```

```
[[-0.7359764  -2.05357788 -0.39476282  0.28642994]
 [ 0.3199846  -0.31390054  0.49812315 -1.52074411]
 [ 0.90676041  0.76795887  0.47207263 -0.34854122]
 [-0.48364529 -0.41234633  0.47301638  0.079715  ]
 [-1.21102337 -0.4342686  -1.82303383 -0.42477508]]
```

```
a = np.array([1,2,3,'4.0',5]) # presence of string element in an array ,converts all the element in string too(Homogemous array)
print("Array a: ",a)

b = np.array([1,2,3.98,4,5]) # presence of float element in an array , converts all the element in float too(Homogemous array)
print("Array b: ",b)
```

```
Array a:  ['1' '2' '3' '4.0' '5']
Array b:  [1.   2.   3.98 4.   5.  ]
```

```
type(a)
```

```
numpy.ndarray
```

```
arr_zeros= np.zeros(6,dtype=int).reshape(2,3) # Created array of zeros with shape of (2,3)
arr_zeros
```

```
array([[0, 0, 0],
       [0, 0, 0]])
```

```
arr_ones= np.ones(6,dtype=int).reshape(3,2) # Created array of zeros with shape of (2,3)
arr_ones
```

```
array([[1, 1],
       [1, 1],
       [1, 1]])
```

```
arr_2d= np.array([[2,3,6],[6,8,9],[5,7,1]]) # 2-D Array
# we want to get 8 from above array
num=arr_2d[1][1]
print(num)
```

```
8
```

```
arr_2d= np.array([[2,3,6],[6,8,9],[5,7,1]]) # 2-D Array
# we want to get 3rd array-item from above array, i.e.: array- [5,7,1]
arr_3=arr_2d[2]
print(arr_3)
```

```
[5 7 1]
```

```
arr_2d.ndim  # Gets the Dimension of an array
```

```
2
```

```
arr_2d.size # Gets total number of data in an array
```

```
9
```

```
len(arr_2d) # Gets the items in an array
```

```
3
```

```
arr_2d_T = arr_2d.T
print("Original Array is: \n",arr_2d)
print("Transpose of  an Array is: \n",arr_2d_T)
```

```
Original Array is:
 [[2 3 6]
 [6 8 9]
 [5 7 1]]
Transpose of  an Array is:
 [[2 6 5]
 [3 8 7]
 [6 9 1]]
```

## INDEXING & SLICING

```
arr1=np.array([[33,17,12,55],[65,32,87,43],[54,31,77,28]])
print("Arr1 is: \n",arr1) # please note there are 3 rows and 4 columns
```

```
Arr1 is:
 [[33 17 12 55]
 [65 32 87 43]
 [54 31 77 28]]
```

```
arr1[1,1] # arr[row,column], we wanted 32 from above array arr1
```

```
32
```

```
arr1[:,:] # gets the whole array output
```

```
array([[33, 17, 12, 55],
       [65, 32, 87, 43],
       [54, 31, 77, 28]])
```

```
arr1[1:,1:] # gets the output array from row index1 and column index 1
```

```
array([[32, 87, 43],
       [31, 77, 28]])
```

```
arr1[1:,0:2] # we wanted 2nd/3rd row and 1st two columns
```

```
array([[65, 32],
       [54, 31]])
```

```
arr12= np.array([12,13,14,15,16,29,43,32,21])
print(arr12)
print(arr12>17) # Boolean Indexing
```

```
[12 13 14 15 16 29 43 32 21]
[False False False False False  True  True  True  True]
```

## Broadcasting

```
arr3=np.array([10,12,14,16]) # Broadcasting the changes in all
elements of an array at once
arr4=arr3+100
arr4
```

```
array([110, 112, 114, 116])
```

```
arr5=arr4/2 #
print(arr5)
```

```
[55. 56. 57. 58.]
```

## Re Shaping

```
a=np.array([1,2,3,4,5,6])
a.reshape(3,2)
```

```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

```
ar = np.arange(10,20).reshape(2,5)
ar
```

```
array([[10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19]])
```

```
ar.sum()
```

```
145
```

```
ar.sum(axis=0) # axis=0 for vertical summation
```

```
array([25, 27, 29, 31, 33])
```

```
ar.sum(axis=1) # axis=1 for Horizontal summation
```

```
array([60, 85])
```

```
A1=np.arange(1,7).reshape(2,3)
A2=np.arange(7,13).reshape(2,3)
print("Array A1: \n",A1)
print("Array A2: \n",A2)
```

```
Array A1:
 [[1 2 3]
 [4 5 6]]
Array A2:
 [[ 7  8  9]
 [10 11 12]]
```

```
#Concatenation
A3=np.concatenate((A1,A2))
A3
```

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

---