Numpy

1. NumPy is an open-source NUMerical PYthon library.
2. main data structure is Array
3. NumPy can be used to perform a wide variety of mathematical operations & scientific computing on arrays in Python.

NumPy indexing is used for accessing an element from an array by giving it an index value that starts from 0. Slicing NumPy arrays means extracting elements from an array in a specific range. It obtains a substring, subtuple, or sublist from a string, tuple, or list.

In [1]:

```
import numpy as np  #short form for np so that we dont have to write pandas each time.
a=[[1,2,3],[4,5,6]] #list declaration.
b=np.array(a)  #convert list 'a' as numpy 2D array and assign it into 'b'.
print(b) #'b' will become numpy 2D array.
print()
print(b.shape)#will return number of rows and number of columns in 2D array 'b'
print()
print(b.ndim) #will return dimension of numpy array.
```

```
[[1 2 3]
 [4 5 6]]

(2, 3)

2
```

In [2]:

```
b=np.zeros(5)
print(b)
```

```
[0. 0. 0. 0. 0.]
```

In [3]:

```
b=np.zeros((5,5))
print(b)
```

```
[[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 [4]:

```
b=np.ones(5)
print(b)
```

```
[1. 1. 1. 1. 1.]
```

In [5]:

```
b=np.ones((2,3))
print(b)
```

```
[[1. 1. 1.]
 [1. 1. 1.]]
```

In [6]:

```
b=np.eye(5) #make diagonal elements as 1 upto 5 rows and 5 columns
print(b)
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

In [7]:

```
b=np.eye(3,4) #make diagonal elements as 1 upto 3 rows and 4 columns
print(b)
```

```
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]]
```

In [8]:

```
b=np.diag([5,2,6,9])
print(b)
```

```
[[5 0 0 0]
 [0 2 0 0]
 [0 0 6 0]
 [0 0 0 9]]
```

In [9]:

```
b= np.diag(b) #return diagonal elements of b.
print(b)
```

```
[5 2 6 9]
```

In [10]:

```
b=np.random.randint(1,10,3)
# return any 3 numbers between 1 to 10. Each time on running ,these values will change.
print(b)
```

```
[1 8 6]
```

In [11]:

```
b=np.random.rand(5) # return any 5 numbers between 0 to 1. Each time on running ,these values will change.
print(b)
```

```
[0.03734085 0.68391768 0.60585426 0.18776148 0.39431987]
```

In [12]:

```
b=np.random.rand(5,2) # return any 10 numbers between 0 to 1.(5 rows, 2 columns) Each time on running ,
#these values will change.
print(b)
```

```
[[0.7683896  0.07765085]
 [0.12126901 0.54438397]
 [0.20537347 0.50153547]
 [0.07302354 0.35979818]
 [0.59174767 0.02378912]]
```

In [13]:

```
a=[0,1,2,3,4,5,6,7,8,9,10]
arr=np.array(a)
print(arr)
b=arr[2:9:2] #print from index 2 - every 2nd number upto index 8(9-1).
print(b)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10]
[2 4 6 8]
```

In [14]:

```
c=arr[:-6] #print from index 0 upto index -7(-6-1) (from right to left indexing are -1,-2,-3,....).
print(c)
```

```
[0 1 2 3 4]
```

In [15]:

```
c=arr[7:] #print from index 7 upto last
print(c)
```

```
[ 7  8  9 10]
```

In [16]:

```
c=arr[:] #print from index start upto last
print(c)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10]
```

In [17]:

```
c=arr[1:7:3] #print from index 1 - every 3rd number upto index 6(7-1).
print(c)
```

```
[1 4]
```

In [18]:

```
c=arr[-5:]  #print from index -5 upto last
print(c)
```

```
[ 6  7  8  9 10]
```