NUMPY

```
In [3]:  import numpy as np
         a = np.arange(15).reshape(3,5)
         a

Out[3]:  array([[ 0,  1,  2,  3,  4],
                [ 5,  6,  7,  8,  9],
                [10, 11, 12, 13, 14]])
```

```
In [4]:  a.min()        #finding the minimum value in the array

Out[4]:  0
```

```
In [5]:  a.max()       #finding the maximum value in the array

Out[5]:  14
```

```
In [7]:  a.ndim         #gives the dimensions of the array

Out[7]:  2
```

```
In [8]:  a.itemsize    #gives the size of an element in the array

Out[8]:  4
```

```
In [9]:  a.size       #gives the total no.of elements present in the array

Out[9]:  15
```

```
In [10]: a.shape       #gives the dimension of the array

Out[10]: (3, 5)
```

In [13]:

```
np.zeros((3,4))   #gives all the elemenets as zeros as per our conditio
n given
```

Out[13]:

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

In [14]:

```
np.ones((3,4))    #gives all the elements as ones as per our condition
 given
```

Out[14]:

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

In [16]:

```
np.linspace(1,5,20)    #linspace is used to print the exact noof elemen
ts in a particuar of 2 numbers
```

Out[16]:

```
array([1.        , 1.21052632, 1.42105263, 1.63157895, 1.84210526,
       2.05263158, 2.26315789, 2.47368421, 2.68421053, 2.89473684,
       3.10526316, 3.31578947, 3.52631579, 3.73684211, 3.94736842,
       4.15789474, 4.36842105, 4.57894737, 4.78947368, 5.        ])
```

In [17]:

```
a.sum(axis=0)      #gives the sum of all elements in each column
```

Out[17]:

```
array([15, 18, 21, 24, 27])
```

In [18]:

```
a.sum(axis=1)    #gives the sum of all the elements in each row
```

Out[18]:

```
array([10, 35, 60])
```

In [19]:

```
for row in a.flatten():     #flatten is used to print all the elements
 in the array in a flat way('c' order)
    print(row)
```

```
0
1
2
```

```
3
4
5
6
7
8
9
10
11
12
13
14
```

In [23]:

```
for sai in np.nditer(a,order='C'):        #nditer is used to print all the elements in C-ORDER & FORTRAN ORDER
    print(sai)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
```

In [24]:

```
for shiridi in np.nditer(a,order='F'):         #fortran order
    print(shiridi)
```

```
0
5
10
```

```
1
6
11
2
7
12
3
8
13
4
9
14
```

In [25]:

```
a
```

Out[25]:

```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])
```

In [29]:

```
for cor in a:
    print(cor)
```

```
[0 1 2 3 4]
[5 6 7 8 9]
[10 11 12 13 14]
```

In [33]:

```
b= np.arange(15,30,1).reshape(3,5)
b
```

Out[33]:

```
array([[15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24],
       [25, 26, 27, 28, 29]])
```

In [36]:

```
np.vstack((a,b))       #stacking is arranging two arrays in a condition
al order   here we used verical
```

Out[36]:

```
array([[ 0,  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, 26, 27, 28, 29]])
```

```
In [38]: c = np.hstack((a,b))
         c
```

```
Out[38]: array([[ 0,  1,  2,  3,  4, 15, 16, 17, 18, 19],
                [ 5,  6,  7,  8,  9, 20, 21, 22, 23, 24],
                [10, 11, 12, 13, 14, 25, 26, 27, 28, 29]])
```

```
In [41]: d = c[1:2,0:]
         d                    #INDEXING
```

```
Out[41]: array([[ 5,  6,  7,  8,  9, 20, 21, 22, 23, 24]])
```

```
In [42]: d.sum()
```

```
Out[42]: 145
```

```
In [43]: c[0:,9:]
```

```
Out[43]: array([[19],
                [24],
                [29]])
```

```
In [44]: c
```

```
Out[44]: array([[ 0,  1,  2,  3,  4, 15, 16, 17, 18, 19],
                [ 5,  6,  7,  8,  9, 20, 21, 22, 23, 24],
                [10, 11, 12, 13, 14, 25, 26, 27, 28, 29]])
```

```
In [45]: d = c>10       #here we are giving an argument that d is avariable whic
         h store all the values which are greater tahn 10 in c
         d
```

```
Out[45]: array([[False, False, False, False, False,  True,  True,  True,  True,
          True],
        [False, False, False, False, False,  True,  True,  True,  True,
          True],
        [False,  True,  True,  True,  True,  True,  True,  True,  True,
          True]])
```

In [46]:

```
c[d]     #here we accesing the all the true values in d with c
```

Out[46]:

```
array([15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 11, 12, 13, 14, 25, 26,
27,
       28, 29])
```

In [49]:

```
c[d] = 0    #replacing the elements which are greater than 10 with '0'
c
```

Out[49]:

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

In [ ]: