```
#Import NumPy as np__rachanavarsha
import numpy as np
```

```
#Create an array of 10 zeros__rachanavarsha
z1=np.zeros(10)
z1
```

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

```
#Create an array of 10 ones__rachanavarsha
z2=np.ones(10)
z2
```

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

```
#Create an array of 10 fives__rachanavarsha
z=np.full(10,5.0)
z
```

```
array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

```
#Create an array of the integers from 10 to 50__rachanavarsha
i=np.arange(10,51)
```

```
i
```

```
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])
```

```
#Create an array of all the even integers from 10 to 50__rachanavarsha
'''
e1=[]
for j in i:
  if i%2==0:
    e1.append(j)
e1_arr=np.array(e1)
e1_arr
'''
ev_arr=np.arange(10,51,2)
ev_arr
```

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

```
#Create a 3x3 matrix with values ranging from 0 to 8__rachanavarsha
a1=np.array([[0,1,2],[3,4,5],[6,7,8]])
```

```
a1
```

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

```
#Create a 3x3 identity matrix__rachanavarsha
a2=np.eye(3)
a2
```

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

```
#Use NumPy to generate a random number between 0 and 1__rachanavarsha
ran_num=np.random.rand()
ran_num
```

```
0.6464024276155605
```

```
#Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution__rachanavarsha
c=np.random.randn(25)
c
```

```
array([-1.39765765, -2.96252371,  0.66280222, -0.82082562, -0.34234319,
       -0.47571051,  3.2514936 , -0.30716788,  1.21567691,  0.33269017,
       -0.14532567, -0.07562363, -0.99218526,  1.03052571, -1.03361821,
       -1.06330386,  1.04272025,  1.04113522,  0.18424914, -1.09563515,
       -3.36506315, -0.88317701, -0.93364365, -0.41329918, -1.25748937])
```

```
#__rachanavarsha
ar=np.arange(0.01,1.01,0.01)
ar
```

```
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.  ])
```

```
#Create an array of 20 linearly spaced points between 0 and 1:__rachanavarsha
la=np.linspace(0,1,20)
la
```

```
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.        ])
```

```
#Numpy Indexing and Selection
mat = np.arange(1,26).reshape(5,5)
mat
```

```
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]])
```

```
mat[2:6,1:6]
```

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

```
mat[3:4,4:6]
```

```
array([[20]])
```

```
mat[0:3,1:2]
```

```
array([[ 2],
       [ 7],
       [12]])
```

```
mat[4:6,0:6]
```

```
array([[21, 22, 23, 24, 25]])
```

```
mat[3:6,0:6]
```

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

```
#Get the sum of all the values in mat__rachanavarsha
sum1=np.sum(mat)
sum1
```

```
325
```

```
#Get the standard deviation of the values in mat__rachanavarsha
sd=np.std(mat)
sd
```

```
7.211102550927978
```

```
#Get the sum of all the columns in mat__rachanavarsha
col_sum=np.sum(mat,axis=0)
col_sum
```

```
array([55, 60, 65, 70, 75])
```