In [ ]:

```
#NAME : SK.SHAREEF
#REG NO : 21BCE9384
```

In [1]:

```
import numpy as np
```

In [2]:

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

Out[2]:

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

In [3]:

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

Out[3]:

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

In [4]:

```
#create an arrays of 10 fives
np.full(10,5)
```

Out[4]:

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

In [6]:

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

Out[6]:

```
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])
```

In [7]:

```
#Create an array of all the even integers from 10 to 50
np.arange(10,51,2)
```

Out[7]:

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

In [8]:

```
# Create a 3x3 matrix with values ranging from 0 to 8
np.arange(9).reshape(3, 3)
```

Out[8]:

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

In [11]:

```
#Create a 3x3 identity matrix
np.identity(3)
```

Out[11]:

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

In [53]:

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

Out[53]:

```
0.6524016727423597
```

In [54]:

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

Out[54]:

```
array([ 0.46724736, -0.09203087, -0.10766245, -0.37508573, -0.49456482,
        0.57506431, -1.72519651, -0.4024901 , -0.4812431 ,  0.60626725,
       -0.6751873 , -0.39248999,  1.72578103, -0.45292473,  0.75308999,
       -0.3206239 ,  1.25017725, -0.97657888, -1.06494777,  0.30572058,
        0.16769221, -0.47667594,  0.86154832,  0.05470819,  1.72797858])
```

In [56]:

```
#create the following matrix
arr =np.linspace(0,1,100).reshape((10,10))
np.round(arr,2)
```

Out[56]:

```
array([[0.  , 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],
       [0.51, 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.  ]])
```

In [57]:

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

Out[57]:

```
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

Now you will be given a few matrices, and be asked to replicate the resulting matrix outputs:

In [58]:

```
mat = np.arange(1,26).reshape(5,5)
mat
```

Out[58]:

```
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]])
```

In [ ]:

```
# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
#array([[12, 13, 14, 15],
      # [17, 18, 19, 20],
       #[22, 23, 24, 25]])
```

In [60]:

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

Out[60]:

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

In [ ]:

```
# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE

#20
```

In [61]:

```
mat[3,4]
```

Out[61]:

```
20
```

In [ ]:

```
# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
#array([[ 2],
 #      [ 7],
  #     [12]])
```

In [63]:

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

Out[63]:

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

In [64]:

```
# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
# array([21, 22, 23, 24, 25])
mat[4:]
```

Out[64]:

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

In [ ]:

```
# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
#array([[16, 17, 18, 19, 20],
       #[21, 22, 23, 24, 25]])
```

In [66]:

```
mat[3:]
```

Out[66]:

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

In [67]:

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

Out[67]:

```
325
```

In [69]:

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

Out[69]:

```
7.211102550927978
```

In [70]:

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

Out[70]:

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