Name : Aariz Zafar
Reg no : 21BCE7868

Assignment 1

Import NumPy as np

In [2]:
```
import numpy as np
```

Create an array of 10 zeros

In [6]:
```
zero_array = np.zeros(10)
print(zero_array)
```

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

Create an array of 10 ones

In [7]:
```
one_array = np.ones(10)
print(one_array)
```

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

Create an array of 10 fives

In [8]:
```
five_array = np.repeat(5,10)
print(five_array)
```

```
[5 5 5 5 5 5 5 5 5 5]
```

Create an array of the integers from 10 to 50

In [10]:
```
array = np.arange(10,50)
print(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]
```

Create an array of all the even integers from 10 to 50

In [11]:
```
even_array = np.arange(10,50,2)
print(even_array)
```

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

Create a 3x3 matrix with values ranging from 0 to 8

In [14]:
```
mat = np.arange(0,9).reshape((3,3))
print(mat)
```

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

Create a 3x3 identity matrix

In [15]:
```
identity = np.identity(5)
print(identity)
```

```
[[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.]]
```

Use NumPy to generate a random number between 0 and 1

In [21]:
```
rand_num = np.random.rand(1)
print(rand_num)
```

```
[0.28036782]
```

Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution

In [25]:
```
norm_array = np.random.normal(1000,size=25)
print(norm_array)
```

```
[1000.39053787 1000.88736788 1000.21293868 1000.11202733  999.49865718
  998.66454609 1000.85452996 1000.02910614  999.92073252  998.20498609
 1000.60408947  998.90611206 1000.10020211  999.74947494 1000.79602737
 1001.27533895  999.60120893  999.76907515 1000.24049044  998.83542112
  998.47031879 1000.60009212  998.09022682  996.65838637 1000.14522975]
```

Create the following matrix:

In [30]:
```
mat1 = np.linspace(0,1,num=100)
np.set_printoptions(precision=2)
print(mat1)
```

```
[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.  ]
```

Create an array of 20 linearly spaced points between 0 and 1:

In [31]:
```
mat2 = np.linspace(0,1,num=20)
print(mat2)
```

```
[0.   0.05 0.11 0.16 0.21 0.26 0.32 0.37 0.42 0.47 0.53 0.58 0.63 0.68
 0.74 0.79 0.84 0.89 0.95 1.  ]
```

Numpy Indexing and Selection

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

In [ ]:
```
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]])
```

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
```

In [39]:
```
mat2 = np.arange(12,24).reshape(3,4)
print(mat2)
```

```
[[12 13 14 15]
 [16 17 18 19]
 [20 21 22 23]]
```

,

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
```

In [40]:
```
print(mat2[2][0])
```

```
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
```

In [42]:
```
mat3 = np.array([
    [1,2,3,4,5],
    [6,7,8,9,10],
    [11,12,13,14,15]])
print(mat3[:,1])
```

```
[ 2  7 12]
```

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
```

In [45]:
```
mat4 = np.arange(21,26)
print(mat4)
```

```
[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
```

In [47]:
```
mat5 = np.arange(16,26).reshape(2,5)
print(mat5)
```

```
[[16 17 18 19 20]
 [21 22 23 24 25]]
```

Now do the following

Get the sum of all the values in mat

In [48]:
```
print(np.sum(mat5))
```

```
205
```

Get the standard deviation of the values in mat

In [49]:
```
print(np.std(mat5))
```

```
2.8722813232690143
```

Get the sum of all the columns in mat

In [50]:
```
print(np.sum(mat5,axis=0))
```

```
[37 39 41 43 45]
```

Type *Markdown* and LaTeX: α^2