IMPORT NUMPY

In [1]:

```
import numpy as np
```

CREATE AN ARRAY OF 10 ZEROS

In [13]:

```
zeros = np.zeros((10))
zeros
```

Out[13]:

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

CREATE AN ARRAY OF 10 ONES

In [14]:

```
ones = np.ones((10))
ones
```

Out[14]:

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

CREATE AN ARRAY OF 10 FIVES

In [20]:

```
fives = np.ones((10))*5
fives
```

Out[20]:

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

CREATE AN ARRAY OF INTEGERS FROM 10 TO 50

In [22]:

```
num = np.arange(10,51)
num
```

Out[22]:

```
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

In [23]:

```
even_num = np.arange(10,51,2)
even_num
```

Out[23]:

```
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 VALUE RANGING FROM 0 TO 8

In [30]:

```
nom = np.arange(0,9)
matrix = nom.reshape(3,3)
matrix
```

Out[30]:

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

CREATE A 3X3 IDENTITY MATRIX

In [31]:

```
id_matrix = np.identity(3)
id_matrix
```

Out[31]:

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

USE NUMPY TO GENERATE RANDOM NUMBER BETWEEN 0 AND 1

In [42]:

```
ran_num = np.random.normal(0,1,1)
ran_num
```

Out[42]:

```
array([-0.63613217, -0.17881572,  1.01319693])
```

USE NUMPY TO GENERATE AN ARRAY OF 25 RANDOM NUMBERS SAMPLED FROM A STANDARD NORMAL DISTRIBUTION

In [43]:

```
rand_num = np.random.normal(0,1,25)
rand_num
```

Out[43]:

```
array([-0.79278025,  0.98537903,  0.89806958,  0.02605532, -1.12511032,
        0.33711473,  0.57586448,  0.10979034, -0.26721891,  0.97729702,
       -2.12418476, -0.3535026 ,  0.73621405, -0.16684042,  1.82420494,
       -0.46522551, -0.81469444, -0.63104   ,  0.24653007,  0.38927486,
       -0.37014155,  0.43909618, -1.27668372,  0.16805337, -0.1022052 ])
```

CREATE THE FOLLOWING MATRIX

In [54]:

```
nums = np.arange(0.01, 1.01, 0.01)
nums
nums_matrix = nums.reshape(10, 10)
nums_matrix
```

Out[54]:

```
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.  ]])
```

NUMPY INDEXING AND SELECTION

CREATE AN ARRAY OF 20 LINEARLY SPACED POINT BETWEEN 0 AND 1

In [56]:

```
line_num = np.linspace(0, 1, 20)
line_num
```

Out[56]:

```
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

In [59]:

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

Out[59]:

```
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 [70]:

```
w = mat[2:5, 1:5]
w
```

Out[70]:

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

In [71]:

```
x = mat[3, 4]
x
```

Out[71]:

```
20
```

In [74]:

```
y = mat[0:3, 1]
y_matrix = y.reshape(3, 1)
y_matrix
```

Out[74]:

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

In [75]:

```
z = mat[4, :5]
z
```

Out[75]:

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

In [76]:

```
a = mat[3:5, :5]
a
```

Out[76]:

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

GET THE SUM OF ALL THE VALUES IN MAT

In [77]:

```
mat_sum = mat.sum()
mat_sum
```

Out[77]:

```
325
```

GET THE STANDARD DEVIATION OF THE VALUE IN SUM

In [78]:

```
mat_sd = mat.std()
mat_sd
```

Out[78]:

```
7.211102550927978
```

GET THE SUM OF ALL COLUMNS IN MAT

In [79]:

```
col_sum = np.sum(mat, axis=0)
col_sum
```

Out[79]:

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

COMPLETED BY EMMANUEL ADEITAN, DATA SCIENCE AND MACHINE LEARNING INTERN, NUPAT TECHNOLOGIES

Thanks