In [2]:
```
# Import numpy as np

import numpy as np
```

In [17]:
```
# Create an array of 10 zeroes

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

Out[17]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

In [18]:
```
# Create an array of 10 ones

arr1 = np.ones(10)
arr1
```

Out[18]:
```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

In [19]:
```
# Create an array of 10 fives

arr2 = np.ones(10)*5
arr2
```

Out[19]:
```
array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

In [20]:
```
# Create an array of integers from 10 to 50

arr3 = np.arange(10,50)
arr3
```

Out[20]:
```
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])
```

In [21]:
```
# Create an array of all the even integers from 10 to 50

arr4 = np.arange(10,52,2)
arr4
```

Out[21]:
```
array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
       44, 46, 48, 50])
```

In [22]:
```
# Create a 3x3 matrix with values ranging from 0 to 8

matrix1 = np.arange(0,9).reshape(3,3)
matrix1
```

Out[22]:
```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

In [25]:
```
# Create a 3x3 identity matrix

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

Out[25]:
```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

In [26]:
```
# Use NumPy to generate a random number between 0 and 1

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

Out[26]:
```
array([-0.32783812])
```

In [27]:
```
# Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution

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

Out[27]:
```
array([ 1.33715177,  0.3798055 , -1.14974531,  1.4278952 ,  0.13347524,
        0.58876098,  1.25896539,  1.98187822,  0.69645498,  0.10848946,
       -0.38943889, -0.56050995,  1.66924481, -0.67930743,  0.22359733,
        1.12118458, -1.16136531, -0.12403986, -1.46168292, -0.75937564,
       -1.09358361, -1.48898643, -0.3915242 , -2.08850756, -0.1027156 ])
```

In [29]:
```
# Create the following matrix

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

Out[29]:
```
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.  ]])
```

In [30]:
```
# Create an array of 20 linearly spaced points between 0 and 1

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

Out[30]:
```
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 [32]:
```
matrix1 = np.arange(1,26).reshape(5,5)
matrix1
```

Out[32]:
```
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 [33]:
```
# 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

arr5 = np.array([[12,13,14,15],
                 [17,18,19,20],
                 [22,23,24,25]])
arr5
```

Out[33]:
```
array([[12, 13, 14, 15],
       [17, 18, 19, 20],
       [22, 23, 24, 25]])
```

In [34]:
```
# 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

print(20)
```

```
20
```

In [35]:
```
# 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

arr6 = np.array([[2],
                 [7],
                 [12]])
arr6
```

Out[35]:
```
array([[ 2],
       [ 7],
       [12]])
```

In [36]:
```
# 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

arr7 = np.array([21,22,23,24,25])
arr7
```

Out[36]:
```
array([21, 22, 23, 24, 25])
```

In [37]:
```
# 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

arr8 = np.array([[16,17,18,19,20],
                 [21,22,23,24,25]])
arr8
```

Out[37]:
```
array([[16, 17, 18, 19, 20],
       [21, 22, 23, 24, 25]])
```

Now do the following

In [40]:
```
# Get the sum of all the values in mat

import numpy as np
mat = np.arange(1,26).reshape(5,5)
total = np.sum(mat)
print(total)
```

```
325
```

In [41]:
```
# Get the standard deviation of all values in mat

import numpy as np
mat = np.arange(1,26).reshape(5,5)
total_SD = np.std(mat)
print(total_SD)
```

```
7.211102550927978
```

In [42]:
```
# Get the sum of all columns in mat


import numpy as np
mat = np.arange(1,26).reshape(5,5)
total = np.sum(mat, axis=0)
print(total)
```

```
[55 60 65 70 75]
```