Mathematic Operation Using Numpy

In [1]:

```
import numpy as np
```

In [5]:

```
arr1 = np.arange(1,10).reshape(3,3)
arr2 = np.arange(1,10).reshape(3,3)
print(arr1)
print(arr2)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

In [9]:

```
add_arr = np.add(arr1, arr2)
print(add_arr)
```

```
[[ 2  4  6]
 [ 8 10 12]
 [14 16 18]]
```

In [11]:

```
sub_arr = np.subtract(arr1, arr2)
print(sub_arr)
```

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

In [13]:

```
arr1 / arr2
```

Out[13]:

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

In [15]:

```
div_arr = np.divide(arr1,arr2)
print(div_arr)
```

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

In [17]:

```
arr1 * arr2
```

Out[17]:

```
array([[ 1,  4,  9],
       [16, 25, 36],
       [49, 64, 81]])
```

In [20]:

```
mult_arr = np.multiply(arr1,arr2)#Multiply element wise
print(mult_arr)
```

```
[[ 1  4  9]
 [16 25 36]
 [49 64 81]]
```

In [22]:

```
#Matric Multiplicaton first row with first column
arr1 @ arr2
```

Out[22]:

```
array([[ 30,  36,  42],
       [ 66,  81,  96],
       [102, 126, 150]])
```

In [25]:

```
matr_multiply = arr1.dot(arr2)
print(matr_multiply)
```

```
[[ 30  36  42]
 [ 66  81  96]
 [102 126 150]]
```

In [27]:

```
arr1
```

Out[27]:

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [28]:

```
#maximum value
arr1.max()
```

Out[28]:

```
9
```

In [29]:

```
#find index
arr1.argmax()
```

Out[29]:

```
8
```

In [32]:

```
# every row and every column maximum value
# zero mean columns
# one means Rows
arr1.max(axis = 0)
arr1.max(axis = 1)
```

Out[32]:

```
array([3, 6, 9])
```

In [34]:

```
#Find Minimum Value
arr1.min()
```

Out[34]:

```
1
```

In [36]:

```
#Find Index of minimum
arr1.argmin()
```

Out[36]:

```
0
```

In [38]:

```
arr1.min(axis = 0)
```

Out[38]:

```
array([1, 2, 3])
```

In [39]:

```
arr1
```

Out[39]:

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [40]:

```
np.sum(arr1)
```

Out[40]:

```
45
```

In [41]:

```
np.sum(arr1, axis = 0) #Every Columns Sum
```

Out[41]:

```
array([12, 15, 18])
```

In [42]:

```
#Average function
np.mean(arr1)
```

Out[42]:

```
5.0
```

In [43]:

```
#Square Root
np.sqrt(arr1)
```

Out[43]:

```
array([[1.        , 1.41421356, 1.73205081],
       [2.        , 2.23606798, 2.44948974],
       [2.64575131, 2.82842712, 3.        ]])
```

In [44]:

```
#Standard Divisoin Function
np.std(arr1)
```

Out[44]:

```
2.581988897471611
```

In [45]:

```
#Exponent Function
np.exp(arr1)
```

Out[45]:

```
array([[2.71828183e+00, 7.38905610e+00, 2.00855369e+01],
       [5.45981500e+01, 1.48413159e+02, 4.03428793e+02],
       [1.09663316e+03, 2.98095799e+03, 8.10308393e+03]])
```

In [46]:

```
#Log Function Natural Log
np.log(arr1)
```

Out[46]:

```
array([[0.        , 0.69314718, 1.09861229],
       [1.38629436, 1.60943791, 1.79175947],
       [1.94591015, 2.07944154, 2.19722458]])
```

In [47]:

```
#log Base 10 Function
np.log10(arr1)
```

Out[47]:

```
array([[0.        , 0.30103   , 0.47712125],
       [0.60205999, 0.69897   , 0.77815125],
       [0.84509804, 0.90308999, 0.95424251]])
```