Python Numpy Mathematical Operations

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

In [ ]:
```
arr1 = np.arange(1,10).reshape(3,3)
arr1
```

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

In [ ]:
```
arr2 = np.arange(1,10).reshape(3,3)
arr2
```

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

In [ ]:
```
arr1,arr2
```

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

In [ ]:
```
arr1 + arr2
```

Out[ ]:
```
array([[ 2,  4,  6],
       [ 8, 10, 12],
       [14, 16, 18]])
```

In [ ]:
```
np.add(arr1, arr2)
```

Out[ ]:
```
array([[ 2,  4,  6],
       [ 8, 10, 12],
       [14, 16, 18]])
```

In [ ]:
```
arr1 - arr2
```

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

In [ ]:
```
np.subtract(arr1, arr2)
```

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

In [ ]:
```
arr1 / arr2
```

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

In [ ]:
```
np.divide(arr1, arr2)
```

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

In [ ]:
```
arr1 * arr2
```

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

In [ ]:
```
arr1, arr2
```

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

In [ ]:
```
np.multiply(arr1, arr2)
```

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

In [ ]:
```
arr1, arr2
```

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

In [ ]:
```
## Matrix Product
arr1 @ arr2
```

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

In [ ]:
```
## mATRIX PRODUCT
np.dot(arr1, arr2)
```

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

In [ ]:
```
arr1
```

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

Find Maximum & Minimum Value

In [ ]:
```
## find max value in matrics
```

In [ ]:
```
arr1.max()
```

Out[ ]:
```
9
```

In [ ]:
```
## find max value index number
```

In [ ]:
```
arr1.argmax()
```

Out[ ]:
```
8
```

In [ ]:
```
arr1
```

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

Find Mini & Max Value in ROW & COL

In [ ]:
```
## 0 represent Column & 1 Represent Row in Axis value
arr1.max(axis = 0)
```

Out[ ]:
```
array([7, 8, 9])
```

In [ ]:
```
## max value in Row
arr1.max(axis = 1)
```

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

In [ ]:
```
## Minimum Value
```

In [ ]:
```
arr1.min()
```

Out[ ]:
```
1
```

In [ ]:
```
## Minimum value Index
arr1.argmin()
```

Out[ ]:
```
0
```

In [ ]:
```
## find Max value in every Row and Column
```

In [ ]:
```
arr1.min(axis = 0)
```

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

In [ ]:
```
arr1.min(axis = 1)
```

Out[ ]:
```
array([1, 4, 7])
```

In [ ]:
```
arr1
```

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

In [ ]:
```
## Total Sum of Matrics arr1
np.sum(arr1)
```

Out[ ]:
```
45
```

In [ ]:
```
## Sum of each Column
np.sum(arr1, axis = 0)
```

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

In [ ]:
```
## Sum of each Row value
np.sum(arr1, axis=1)
```

Out[ ]:
```
array([ 6, 15, 24])
```

Mean ,Standard Devi, squrt ,Log

In [ ]:
```
arr1
```

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

In [ ]:
```
## Mean value of Matrics
np.mean(arr1)
```

Out[ ]:
```
5.0
```

In [ ]:
```
## squre root of each value in matrics
np.sqrt(arr1)
```

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

In [ ]:
```
## Standard Deviation
np.std(arr1)
```

Out[ ]:
```
2.581988897471611
```

In [ ]:
```
## Exponent e^x
np. exp(arr1)
```

Out[ ]:
```
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 [ ]:
```
## Log
np.log(arr1)
```

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

In [ ]:
```
np.log10(arr1)
```

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

In [ ]:
```

```