```python
import numpy as np
```

```python
# 2D Array
a=np.array([[1,2,3],[4,5,6]])
a
```

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

```python
a.shape
```

```
(2, 3)
```

```python
len(a)
```

```
2
```

```python
a.ndim
```

```
2
```

```python
a.size
```

```
6
```

```python
a.dtype
```

```
dtype('int64')
```

```python
a1=np.zeros(5)
a1
```

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

```python
# create an array of one
a2=np.ones(5)
a2
```

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

```python
a3=np.arange(10,30,5)
a3
```

```
array([10, 15, 20, 25])
```

```python
a4=np.linspace(0,10,8)
a4
```

```
array([ 0.        ,  1.42857143,  2.85714286,  4.28571429,  5.71428571,
        7.14285714,  8.57142857, 10.        ])
```

```python
#2-ARITHMETIC OPERATION:
#ADDITION
```

```python
a=np.array([[1,2,3],[4,5,6]])
b=np.array([[7,8,9],[10,11,12]])
a+b
```

```
array([[ 8, 10, 12],
       [14, 16, 18]])
```

```python
#SUBTRACTION
```

```python
a=np.array([[1,2,3],[4,5,6]])
b=np.array([[7,8,9],[10,11,12]])
a-b
```

```
array([[-6, -6, -6],
       [-6, -6, -6]])
```

```python
# MULTIPLICATION
```

```
a=np.array([[1,2,3],[4,5,6]])
b=np.array([[7,8,9],[10,11,12]])
a*b
```

```
array([[ 7, 16, 27],
       [40, 55, 72]])
```

```
#DIVISION
```

```
a=np.array([[1,2,3],[4,5,6]])
b=np.array([[7,8,9],[10,11,12]])
a/b
```

```
array([[0.14285714, 0.25      , 0.33333333],
       [0.4       , 0.45454545, 0.5       ]])
```

```
np.exp(b)
```

```
array([[  1096.63315843,   2980.95798704,   8103.08392758],
       [ 22026.46579481,  59874.1417152 , 162754.791419  ]])
```

```
np.sqrt(b)
```

```
array([[2.64575131, 2.82842712, 3.        ],
       [3.16227766, 3.31662479, 3.46410162]])
```

```
#COMPARSION
```

```
a==b
```

```
array([[False, False, False],
       [False, False, False]])
```

```
a>2
```

```
array([[False, False,  True],
       [ True,  True,  True]])
```

```
#AGGREGATE FUNCTION
```

```
a.sum()
```

```
21
```

```
a.min()
```

```
1
```

```
a.max()
```

```
6
```

```
a.cumsum()
```

```
array([ 1,  3,  6, 10, 15, 21])
```

```
a.mean()
```

```
3.5
```

```
# CORRELATION COEFFICIENT
```

```
np.corrcoef(a,b)
```

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

```
np.std(a)
```

```
1.707825127659933
```

Colab paid products - Cancel contracts here

✓ 0s   completed at 11:43 PM   ● ✕