```
import numpy as np
```

```
# 2D Array
a=np.array([[[1,2,3],[4,5,6],[7,8,9]]])
a
```

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

```
a.shape
```

```
(1, 3, 3)
```

```
len(a)
```

```
1
```

```
a.ndim
```

```
3
```

```
a.size
```

```
9
```

```
a.dtype
```

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

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

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

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

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

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

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

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

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

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

```
a=np.array([[[1,2,3],[4,5,6],[7,8,9]]])
b=np.array([[[7,8,9],[10,11,12],[13,14,15]]])
a+b
```

```
array([[[ 8, 10, 12],
        [14, 16, 18],
        [20, 22, 24]]])
```

```
#SUBTRACTION
```

```
a=np.array([[[1,2,3],[4,5,6],[7,8,9]]])
b=np.array([[[7,8,9],[10,11,12],[13,14,15]]])
a-b
```

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

```
# MULTIPLICATION


a=np.array([[[1,2,3],[4,5,6],[7,8,9]]])
b=np.array([[[7,8,9],[10,11,12],[13,14,15]]])
a*b
```

```
array([[[  7,  16,  27],
        [ 40,  55,  72],
        [ 91, 112, 135]]])
```

```
#DIVISION


a=np.array([[[1,2,3],[4,5,6],[7,8,9]]])
b=np.array([[[7,8,9],[10,11,12],[13,14,15]]])
a/b
```

```
array([[[0.14285714, 0.25      , 0.33333333],
        [0.4       , 0.45454545, 0.5       ],
        [0.53846154, 0.57142857, 0.6       ]]])
```

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

```
array([[[1.09663316e+03, 2.98095799e+03, 8.10308393e+03],
        [2.20264658e+04, 5.98741417e+04, 1.62754791e+05],
        [4.42413392e+05, 1.20260428e+06, 3.26901737e+06]]])
```

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

```
array([[[2.64575131, 2.82842712, 3.        ],
        [3.16227766, 3.31662479, 3.46410162],
        [3.60555128, 3.74165739, 3.87298335]]])
```

```
#COMPARSION


a==b
```

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

```
a>2
```

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

```
#AGGREGATE FUNCTION


a.sum()
```

```
45
```

```
a.min()
```

```
1
```

```
a.max()
```

```
9
```

```
a.cumsum()
```

```
array([ 1,  3,  6, 10, 15, 21, 28, 36, 45])
```

```
a.mean()
```

```
5.0
```

```
# CORRELATION COEFFICIENT


np.corrcoef(a,b)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-94-9b39842b5144> in <cell line: 1>()
----> 1 np.corrcoef(a,b)

                                 3 frames
/usr/local/lib/python3.10/dist-packages/numpy/lib/function_base.py in cov(m, y,
rowvar, bias, ddof, fweights, aweights, dtype)
   2591     m = np.asarray(m)
   2592     if m.ndim > 2:
-> 2593         raise ValueError("m has more than 2 dimensions")
   2594
   2595     if y is not None:

ValueError: m has more than 2 dimensions
```

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

```
2.581988897471611
```



0s completed at 12:12 AM