```
import numpy as np
```

1D ARRAY

```
x=np.array([3,5,7,8,9])
x
```

```
array([3, 5, 7, 8, 9])
```

```
x.shape
```

```
(5,)
```

```
len(x)
```

```
5
```

```
x.ndim
```

```
1
```

```
x.size
```

```
5
```

```
x.dtype
```

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

```
x1=np.zeros(4)
x1
```

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

Create an Array of one

```
x2=np.ones(5)
x2
```

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

```
x3=np.arange(10,50,5)
x3
```

```
array([10, 15, 20, 25, 30, 35, 40, 45])
```

```
x4=np.linspace(0,10,7)
x4
```

```
array([ 0.        ,  1.66666667,  3.33333333,  5.        ,  6.66666667,
        8.33333333, 10.        ])
```

Arithmetic operators

Addition

```
x1=np.array([3,5,1,0,1])
x2=np.array([0,1,3,6,1])
x1+x2
```

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

Substraction

```
x1-x2
```

```
array([ 3,  4, -2, -6,  0])
```

Multiplication

```
x1*x2
```

```
array([0, 5, 3, 0, 1])
```

Division

```
x1/x2
```

```
<ipython-input-25-325bdae1640d>:1: RuntimeWarning: divide by zero encountered in true_divide
  x1/x2
array([       inf, 5.        , 0.33333333, 0.        , 1.        ])
```

```
np.exp(x2)
```

```
array([  1.        ,   2.71828183,  20.08553692, 403.42879349,
         2.71828183])
```

```
np.sqrt(x2)
```

```
array([0.        , 1.        , 1.73205081, 2.44948974, 1.        ])
```

COMPARISON

```
x1==x2
```

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

```
x1>3
```

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

AGGREGATE FUNCTION

```
x1.sum()
```

```
10
```

```
x1.min()
```

```
0
```

```
x1.max()
```

```
5
```

```
x1.cumsum()
```

```
array([ 3,  8,  9,  9, 10])
```

```
x1.mean()
```

```
2.0
```

```
#corelation coefficient
np.corrcoef(x1,x2)
```

```
array([[ 1.       , -0.6806376],
       [-0.6806376,  1.       ]])
```

```
np.std(x1)
```

```
1.7888543819998317
```

2D ARRAY

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

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

```
a.shape
```

```
(2, 3)
```

```
a.ndim
```

```
2
```

```
len(a)
```

```
2
```

```
a.size
```

```
6
```

```
a.dtype
```

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

```
a1=np.ones((2,3))
a1
```

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

```
a1=np.zeros((2,3))
a1
```

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

```
a1=np.arange(2*5)
a2=a1.reshape((2,5))
a2
```

```
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
```

```
a2=np.linspace(1,10,12)
a3=a2.reshape((3,4))
a3
```

```
array([[ 1.        ,  1.81818182,  2.63636364,  3.45454545],
       [ 4.27272727,  5.09090909,  5.90909091,  6.72727273],
       [ 7.54545455,  8.36363636,  9.18181818, 10.        ]])
```

ARITHMETIC OPERATORS FOR 2D

Addition

```
a1=np.array([[1,2,3,],[3,4,5]])
a2=np.array([[4,5,0],[2,1,1]])
```

```
a2=np.array([[4,5,6],[2,1,1]])
a1+a2
```

```
array([[5, 7, 3],
       [5, 5, 6]])
```

Substraction

```
a1-a2
```

```
array([[-3, -3,  3],
       [ 1,  3,  4]])
```

Multiplication

```
a1*a2
```

```
array([[ 4, 10,  0],
       [ 6,  4,  5]])
```

Division

```
a1/a2
```

```
<ipython-input-79-3ccc78f0d2eb>:1: RuntimeWarning: divide by zero encountered in true_divide
  a1/a2
array([[0.25, 0.4 ,  inf],
       [1.5 , 4.  , 5.  ]])
```

```
np.exp(a2)
```

```
array([[ 54.59815003, 148.4131591 ,   1.        ],
       [  7.3890561 ,   2.71828183,   2.71828183]])
```

```
np.sqrt(a2)
```

```
array([[2.        , 2.23606798, 0.        ],
       [1.41421356, 1.        , 1.        ]])
```

COMPARISON

```
a1==a2
```

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

```
a1>a2
```

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

AGGREGATE FUNCTIONS

```
a1.sum()
```

```
18
```

```
a1.min()
```

```
1
```

```
a1.max()
```

```
5
```

```
a1.cumsum()
```

```
array([ 1,  3,  6,  9, 13, 18])
```

```
a1.mean()
```

```
3.0
```

```
#corelation coefficient
np.corrcoef(a1,a2)
```

```
array([[ 1.        ,  1.        , -0.75592895, -0.8660254 ],
       [ 1.        ,  1.        , -0.75592895, -0.8660254 ],
       [-0.75592895, -0.75592895,  1.        ,  0.32732684],
       [-0.8660254 , -0.8660254 ,  0.32732684,  1.        ]])
```

```
np.std(a1)
```

```
1.2909944487358056
```

3D ARRAY

```
y=np.array([[[1,2,3],[2,3,4],[4,5,6]]])
y
```

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

```
y.ndim
```

```
3
```

```
y.shape
```

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

```
len(y)
```

```
1
```

```
y.size
```

```
9
```

```
y.dtype
```

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

```
np.ones((2,3,4))
```

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

       [[1., 1., 1., 1.],
        [1., 1., 1., 1.],
        [1., 1., 1., 1.]]])
```

```
np.zeros((2,3,4))
```

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

       [[0., 0., 0., 0.],
        [0., 0., 0., 0.],
        [0., 0., 0., 0.]]])
```

```
z2=np.linspace(1,10,24)
n=z2.reshape((2,3,4))
n
```

```
array([[[ 1.        ,  1.39130435,  1.7826087 ,  2.17391304],
        [ 2.56521739,  2.95652174,  3.34782609,  3.73913043],
        [ 4.13043478,  4.52173913,  4.91304348,  5.30434783]],

       [[ 5.69565217,  6.08695652,  6.47826087,  6.86956522],
        [ 7.26086957,  7.65217391,  8.04347826,  8.43478261],
        [ 8.82608696,  9.2173913 ,  9.60869565, 10.        ]]])
```

```
z=np.arange(2*3*4)
z1=z.reshape((2,3,4))
z1
```

```
array([[[ 0,  1,  2,  3],
        [ 4,  5,  6,  7],
        [ 8,  9, 10, 11]],

       [[12, 13, 14, 15],
        [16, 17, 18, 19],
        [20, 21, 22, 23]]])
```

ARITHMETIC OPERATION FOR 3D

Addition

```
m=np.array([[[1,2,3],[3,4,5],[5,6,7]]])
c=np.array([[[4,3,2],[1,6,2],[1,5,2]]])
m+c
```

```
array([[[ 5,  5,  5],
        [ 4, 10,  7],
        [ 6, 11,  9]]])
```

Substraction

```
m-c
```

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

Multiplication

```
m*c
```

```
array([[[ 4,  6,  6],
        [ 3, 24, 10],
        [ 5, 30, 14]]])
```

Division

```
m/c
```

```
array([[[0.25      , 0.66666667, 1.5       ],
        [3.        , 0.66666667, 2.5       ],
        [5.        , 1.2       , 3.5       ]]])
```

```
np.exp(c)
```

```
array([[[ 54.59815003,  20.08553692,   7.3890561 ],
        [  2.71828183, 403.42879349,   7.3890561 ],
        [  2.71828183, 148.4131591 ,   7.3890561 ]]])
```

```
np.sqrt(c)
```

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

COMPARISON FOR 3D

```
m==c
```

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

```
m>c
```

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

AGGREGATE FUNCTION FOR 3D

```
m.sum()
```

```
36
```

```
m.min()
```

```
1
```

```
m.max()
```

```
7
```

```
m.cumsum()
```

```
array([ 1,  3,  6,  9, 13, 18, 23, 29, 36])
```

```
m.mean()
```

```
4.0
```

```
np.std(m)
```

```
1.8257418583505538
```

✓ 0s completed at 9:55 AM