1D ARRAY

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

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

```
a.shape
```

```
(6,)
```

```
len(a)
```

```
6
```

```
a.ndim
```

```
1
```

```
a.size
```

```
6
```

```
a.dtype
```

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

array of 0

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

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

array of 1

```
c=np.ones(4)
c
```

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

```
d=np.arange(10,20,2)
d
```

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

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

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

arthmetic operation

addition

```
a=np.array([1,2,3,4,5])
a1=np.array([6,7,8,9,10])
a+a1
```

```
array([ 7,  9, 11, 13, 15])
```

subtraction

```
a-a1
```

```
array([-5, -5, -5, -5, -5])
```

multiplication

```
a*a1
```

```
array([ 6, 14, 24, 36, 50])
```

division

```
a/a1
```

```
array([0.16666667, 0.28571429, 0.375     , 0.44444444, 0.5       ])
```

```
np.exp(a1)
```

```
array([  403.42879349,  1096.63315843,  2980.95798704,  8103.08392758,
       22026.46579481])
```

```
np.sqrt(a)
```

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

comparsion

```
a==a1
```

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

```
a>3
```

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

aggregate function

```
a.sum()
```

```
15
```

```
a.min()
```

```
1
```

```
a.max()
```

```
5
```

```
a.cumsum()
```

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

```
a.mean()
```

```
3.0
```

corre;ation cofficient

```
np.corrcoef(a,a1)
```

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

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

```
1.4142135623730951
```

2D ARRAY

```
import numpy as np
a=np.array([[1,2,3,4,5],[6,7,8,9,10]])
a
```

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

```
a.shape
```

```
     (2, 5)
```

```
len(a)
```

```
     2
```

```
a.ndim
```

```
     2
```

```
a.size
```

```
     10
```

```
a.dtype
```

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

creating an array of zero

```
b=np.zeros(6)
b
```

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

creating an array of one

```
c=np.ones(6)
c
```

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

```
d=np.arange(10,20,2)
d
```

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

```
e=np.linspace(0,10,6)
e
```

```
     array([ 0.,  2.,  4.,  6.,  8., 10.])
```

arthmetic operation

addition

```
a=np.array([[0,1,2,3,4],[5,6,7,8,9]])
a1=np.array([[10,11,12,13,14],[15,16,17,18,19]])
a+a1
```

```
     array([[10, 12, 14, 16, 18],
            [20, 22, 24, 26, 28]])
```

subtraction

```
a-a1
```

```
     array([[-10, -10, -10, -10, -10],
            [-10, -10, -10, -10, -10]])
```

multiplication

```
a*a1
```

```
     array([[  0,  11,  24,  39,  56],
            [ 75,  96, 119, 144, 171]])
```

division

```
a/a1
```

```
array([[0.        , 0.09090909, 0.16666667, 0.23076923, 0.28571429],
       [0.33333333, 0.375     , 0.41176471, 0.44444444, 0.47368421]])
```

```
np.exp(a)
```

```
array([[1.00000000e+00, 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]])
```

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

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

comparsion

```
a==a1
```

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

```
a>4
```

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

aggregate function

```
a.sum()
```

```
45
```

```
a.min()
```

```
0
```

```
a.max()
```

```
9
```

```
a.cumsum()
```

```
array([ 0,  1,  3,  6, 10, 15, 21, 28, 36, 45])
```

```
a.mean()
```

```
4.5
```

correlation function

```
np.corrcoef(a,a1)
```

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

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

```
2.8722813232690143
```

✓ 0s completed at 9:34 AM ● ×

3D ARRAY

```
import numpy as np
a=np.array([[[0,1,2,3],[4,5,6,7],[8,9,10,11]]])
a
```

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

```
a.shape
```

```
     (1, 3, 4)
```

```
len(a)
```

```
     1
```

```
a.ndim
```

```
     3
```

```
a.size
```

```
     12
```

```
a.dtype
```

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

creating array of zero

```
b=np.zeros(6)
b
```

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

creating array of one

```
c=np.ones(6)
c
```

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

```
d=np.arange(10,20,2)
d
```

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

```
e=np.linspace(0,10,4)
e
```

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

arthmetic operation

addition

```
a=np.array([[[0,1,2],[3,4,5],[6,7,8]]])
a1=np.array([[[9,10,11],[12,13,14],[15,16,17]]])
a+a1
```

```
     array([[[ 9, 11, 13],
             [15, 17, 19],
             [21, 23, 25]]])
```

subtraction

```
a-a1
```

```
     array([[[-9, -9, -9],
             [-9, -9, -9],
             [-9, -9, -9]]])
```

multiplication

```
a*a1
```

```
     array([[[  0,  10,  22],
             [ 36,  52,  70],
             [ 90, 112, 136]]])
```

division

```
a/a1
```

```
     array([[[0.        , 0.1       , 0.18181818],
             [0.25      , 0.30769231, 0.35714286],
             [0.4       , 0.4375    , 0.47058824]]])
```

```
np.exp(a)
```

```
     array([[[1.00000000e+00, 2.71828183e+00, 7.38905610e+00],
             [2.00855369e+01, 5.45981500e+01, 1.48413159e+02],
             [4.03428793e+02, 1.09663316e+03, 2.98095799e+03]]])
```

```
np.sqrt(a1)
```

```
     array([[[3.        , 3.16227766, 3.31662479],
             [3.46410162, 3.60555128, 3.74165739],
             [3.87298335, 4.        , 4.12310563]]])
```

comparsion

```
a==a1
```

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

```
a<2
```

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

```
a>a1
```

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

aggregate function

```
a.sum()
```

```
36
```

```
a.min()
```

```
0
```

```
a.max()
```

```
8
```

```
a.cumsum()
```

```
array([ 0,  1,  3,  6, 10, 15, 21, 28, 36])
```

```
a.mean()
```

```
4.0
```

correlation cofficient

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

```
2.581988897471611
```