

## ▼ 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.          ])
```

## ▼ arithmetic 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 coefficient

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

```
2.581988897471611
```