

```
import numpy as np
```

▼ 1 D Array

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

```
a.shape
(5,)
```

```
len(a)
5
```

```
a.ndim
1
```

```
a=np.array([1,2,3,4,5])
a.dtype
dtype('int64')
```

```
a.size
5
```

```
a1=np.zeros(6)
a1
array([0., 0., 0., 0., 0., 0.])
```

▼ Create an array of one

```
a2=np.ones(6)
a2
array([1., 1., 1., 1., 1., 1.])
```

```
a3=np.arange(10,30,5)
a3
array([10, 15, 20, 25])
```

```
a4=np.linspace(0,10,5)
a4
array([ 0. , 2.5, 5. , 7.5, 10. ])
```

▼ Array Functions

Addition

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

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

▼ Subtraction

a-b

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

▼ Multiplication

a*b

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

▼ Division

a/b

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

np.exp(b)

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

np.sqrt(b)

```
array([2.44948974, 2.64575131, 2.82842712, 3.      , 3.16227766])
```

np.sin(a)

```
array([ 0.84147098,  0.90929743,  0.14112001, -0.7568025 , -0.95892427])
```

np.cos(b)

```
array([ 0.96017029,  0.75390225, -0.14550003, -0.91113026, -0.83907153])
```

np.log(b)

```
array([1.79175947, 1.94591015, 2.07944154, 2.19722458, 2.30258509])
```

a==b

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

a>2

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

▼ Aggregate Function

a.sum()

```
15
```

a.max()

```
5

a.min()

1

a.cumsum()

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

a.mean()

3.0

np.std(a)

1.4142135623730951

import numpy as np

np.corrcoef(a,b)

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

▼ 2D ARRAY

```
import numpy as np

b=np.array([[5,5,5],[5,5,5],[5,5,5]])
b

array([[5, 5, 5],
       [5, 5, 5],
       [5, 5, 5]])

b.shape

(3, 3)

len(b)

3

b.ndim

2

b.dtype

dtype('int64')

b.size

9

b1=np.zeros(9)
b1

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

▼ create an array of one

```
b2=np.ones(9)
b2

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

b3=np.arange(10,30,3)
b3

array([10, 13, 16, 19, 22, 25, 28])

b4=np.linspace(0,10,5)
b4

array([ 0. ,  2.5,  5. ,  7.5, 10. ])
```

▼ Array functions

▼ addition

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

array([[ 6,  7,  8],
       [ 9, 10, 11],
       [12, 13, 14]])
```

▼ Subtraction

```
a-b

array([[ -4,  -3,  -2],
       [ -1,   0,   1],
       [  2,   3,   4]])
```

▼ Multiplication

```
a*b

array([[ 5, 10, 15],
       [20, 25, 30],
       [35, 40, 45]])
```

▼ Division

```
a/b

array([[0.2, 0.4, 0.6],
       [0.8, 1. , 1.2],
       [1.4, 1.6, 1.8]])

np.exp(b)

array([[148.4131591, 148.4131591, 148.4131591],
       [148.4131591, 148.4131591, 148.4131591],
       [148.4131591, 148.4131591, 148.4131591]])

np.sqrt(b)
```

```
array([[2.23606798, 2.23606798, 2.23606798],
       [2.23606798, 2.23606798, 2.23606798],
       [2.23606798, 2.23606798, 2.23606798]])
```

```
np.sin(a)
```

```
array([[ 0.84147098,  0.90929743,  0.14112001],
       [-0.7568025 , -0.95892427, -0.2794155 ],
       [ 0.6569866 ,  0.98935825,  0.41211849]])
```

```
np.cos(b)
```

```
array([[0.28366219, 0.28366219, 0.28366219],
       [0.28366219, 0.28366219, 0.28366219],
       [0.28366219, 0.28366219, 0.28366219]])
```

```
np.log(b)
```

```
array([[1.60943791, 1.60943791, 1.60943791],
       [1.60943791, 1.60943791, 1.60943791],
       [1.60943791, 1.60943791, 1.60943791]])
```

▼ Comparison operations

```
a==b
```

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

```
a>2
```

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

▼ Aggregate function

```
a.sum()
```

```
45
```

```
a.max()
```

```
9
```

```
a.min()
```

```
1
```

```
a.cumsum()
```

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

```
a.mean()
```

```
5.0
```

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

```
2.581988897471611
```

```
c=np.corrcoef(a)
```

```
print(c)
[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]
```

▼ 3D ARRAY

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

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

```
c.shape
```

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

```
len(c)
```

```
1
```

```
c.ndim
```

```
3
```

```
c.dtype
```

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

```
c.size
```

```
9
```

```
c1=np.zeros(10)
```

```
c1
```

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

▼ create an array of one

```
c2=np.ones(6)
```

```
c2
```

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

```
c3=np.arange(10,30,5)
```

```
c3
```

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

```
c4=np.linspace(0,12,3)
```

```
c4
```

```
array([ 0.,  6., 12.])
```

▼ array functions

▼ Addition

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

```
array([[ 3,  5,  5],
       [ 6,  9,  8],
       [ 9, 13, 11]])
```

▼ Subtraction

```
b-c
```

```
array([[ 1,  1, -1],
       [-2, -1, -4],
       [-5, -3, -7]])
```

▼ Multiplication

```
b*c
```

```
array([[ 2,  6,  6],
       [ 8, 20, 12],
       [14, 40, 18]])
```

▼ Division

```
b/c
```

```
array([[2.         , 1.5         , 0.66666667],
       [0.5         , 0.8         , 0.33333333],
       [0.28571429, 0.625        , 0.22222222]])
```

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

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

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

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

```
np.sin(c)
```

```
array([[ 0.84147098,  0.90929743,  0.14112001],
       [-0.7568025 , -0.95892427, -0.2794155 ],
       [ 0.6569866 ,  0.98935825,  0.41211849]])
```

```
np.cos(b)
```

```
array([[ -0.41614684, -0.98999925 , -0.41614684],
       [-0.41614684, -0.65364362, -0.41614684],
       [-0.41614684,  0.28366219, -0.41614684]])
```

```
np.log(b)
```

```
array([[0.69314718, 1.09861229, 0.69314718],
       [0.69314718, 1.38629436, 0.69314718],
       [0.69314718, 1.60943791, 0.69314718]])
```

▼ Comparison operations

```
b==c
```

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

```
b>5
```

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

▼ Aggregate functions

```
b.sum()
```

```
24
```

```
b.max()
```

```
5
```

```
b.min()
```

```
2
```

```
b.cumsum()
```

```
array([ 2,  5,  7,  9, 13, 15, 17, 22, 24])
```

```
c.mean()
```

```
5.0
```

```
np.std(c)
```

```
2.581988897471611
```