```
import numpy as np
```

1D 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.size
```

```
5
```

```
a.dtype
```

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

```
a1=np.zeros(8)
a1
```

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

```
# create an array of one
a2=np.ones(8)
a2
```

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

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

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

```
a4=np.linspace(0,10,7)
a4
```

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

Arithematic Operators

*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])
```

Comparison

```
a==b
```

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

```
a>b
```

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

Aggregate Functions

```
a.sum()
```

```
15
```

```
a.min()
```

```
1
```

```
a.max()
```

```
5
```

```
a.cumsum()
```

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

```
a.mean()
```

```
3.0
```

```
# correlation coefficient
np.corrcoef(a,b)
```

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

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

```
1.4142135623730951
```

2D Array

```
a2=np.array([[1,2,3],
           [4,5,6],
           [7,8,9]])
a2
```

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

```
a2.shape
```

```
(3, 3)
```

```
len(a2)
```

```
3
```

```
a2.ndim
```

```
2
```

```
a2.size
```

```
9
```

```
a2.dtype
```

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

```
a22=np.zeros(6)
a22
```

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

```
a22=np.ones(6)
a22
```

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

Arithematic Operations

Addition

```
a2=np.array([[1, 2, 3],[4, 5, 6],[7, 8, 9]])
b2=np.array([[11,12,13], [14,15,16], [17,18,19]])
a2+b2
```

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

Subtraction

```
a2-b2
```

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

Multiplication

```
a2*b2
```

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

Division

```
a2/b2
```

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

```
np.exp(b2)
```

```
array([[5.98741417e+04, 1.62754791e+05, 4.42413392e+05],
       [1.20260428e+06, 3.26901737e+06, 8.88611052e+06],
       [2.41549528e+07, 6.56599691e+07, 1.78482301e+08]])
```

```
np.sqrt(b2)
```

```
array([[3.31662479, 3.46410162, 3.60555128],
       [3.74165739, 3.87298335, 4.        ],
       [4.12310563, 4.24264069, 4.35889894]])
```

Comparison

```
a2==b2
```

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

```
a2<b2
```

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

```
np.array_equal(a2,b2)
```

```
False
```

Aggregate Functions

```
a.sum()
```

```
45
```

```
a.min()
```

```
1
```

```
a.max()
```

```
9
```

```
a.min()
```

```
1
```

```
a.cumsum()
```

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

```
a.mean()
```

```
5.0
```

```
# correlation coefficient
np.corrcoef(a2,b2)
```

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

```
np.std(a2)
```

```
2.581988897471611
```

3D Array

```
a3=np.array([[[1, 2, 3, 4],
                      [5, 6, 7, 8],
                      [9, 10, 11, 12]],

                     [[13, 14, 15, 16],
                      [17, 18, 19, 20],
                      [21, 22, 23, 24]]])
a3
```

```
array([[[ 1,  2,  3,  4],
        [ 5,  6,  7,  8],
        [ 9, 10, 11, 12]],

       [[13, 14, 15, 16],
        [17, 18, 19, 20],
        [21, 22, 23, 24]]])
```

```
a3.shape
```

```
(2, 3, 4)
```

```
len(a3)
```

```
2
```

```
a3.ndim
```

```
3
```

```
a3.size
```

```
24
```

```
a3.dtype
```

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

```
a33=np.zeros(6)
```

```
# create an array of three
a333=np.ones(6)
a333
```

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

Arithematic Operators

Addition

```
a3=np.array([[[1, 2, 3, 4],
                     [5, 6, 7, 8],
                     [9, 10, 11, 12]],

                    [[13, 14, 15, 16],
                     [17, 18, 19, 20],
                     [21, 22, 23, 24]]])
b3=np.array([[[11, 22, 33, 44],
                     [55, 66, 77, 88],
                     [99, 110, 111, 122]],

                    [[23, 24, 25, 26],
                     [27, 28, 29, 30],
                     [31, 32, 33, 34]]])
a3+b3
```

```
array([[[ 12,  24,  36,  48],
        [ 60,  72,  84,  96],
        [108, 120, 122, 134]],

       [[ 36,  38,  40,  42],
        [ 44,  46,  48,  50],
        [ 52,  54,  56,  58]]])
```

Double-click (or enter) to edit

Subtraction

```
a3-b3
```

```
array([[[ -10,  -20,  -30,  -40],
        [ -50,  -60,  -70,  -80],
        [ -90, -100, -100, -110]],

       [[ -10,  -10,  -10,  -10],
        [ -10,  -10,  -10,  -10],
        [ -10,  -10,  -10,  -10]]])
```

Multiplication

```
a3*b3
```

```
array([[[  11,   44,   99,  176],
        [ 275,  396,  539,  704],
        [ 891, 1100, 1221, 1464]],

       [[ 299,  336,  375,  416],
        [ 459,  504,  551,  600],
        [ 651,  704,  759,  816]]])
```

Division

```
a3/b3
```

```
array([[[0.09090909, 0.09090909, 0.09090909, 0.09090909],
        [0.09090909, 0.09090909, 0.09090909, 0.09090909],
        [0.09090909, 0.09090909, 0.0990991 , 0.09836066]],

       [[0.56521739, 0.58333333, 0.6       , 0.61538462],
        [0.62962963, 0.64285714, 0.65517241, 0.66666667],
        [0.67741935, 0.6875    , 0.6969697 , 0.70588235]]])
```

```
np.exp(b3)
```

```
array([[[5.98741417e+04, 3.58491285e+09, 2.14643580e+14, 1.28516001e+19],
        [7.69478527e+23, 4.60718663e+28, 2.75851345e+33, 1.65163625e+38],
        [9.88903032e+42, 5.92097203e+47, 1.60948707e+48, 9.63666567e+52]],

       [[9.74480345e+09, 2.64891221e+10, 7.20048993e+10, 1.95729609e+11],
        [5.32048241e+11, 1.44625706e+12, 3.93133430e+12, 1.06864746e+13],
        [2.90488497e+13, 7.89629602e+13, 2.14643580e+14, 5.83461743e+14]]])
```

```
np.sqrt(b3)
```

```
array([[[ 3.31662479,  4.69041576,  5.74456265,  6.63324958],
        [ 7.41619849,  8.1240384 ,  8.77496439,  9.38083152],
        [ 9.94987437, 10.48808848, 10.53565375, 11.04536102]],

       [[ 4.79583152,  4.89897949,  5.        ,  5.09901951],
        [ 5.19615242,  5.29150262,  5.38516481,  5.47722558],
        [ 5.56776436,  5.65685425,  5.74456265,  5.83095189]]])
```

Comparison

```
a3==b3
```

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

       [[False, False, False, False],
        [False, False, False, False],
        [False, False, False, False]]])
```

```
a3<b3
```

```
array([[[ True,  True,  True,  True],
        [ True,  True,  True,  True],
        [ True,  True,  True,  True]],

       [[ True,  True,  True,  True],
        [ True,  True,  True,  True],
        [ True,  True,  True,  True]]])
```

Aggregate Functions

Double-click (or enter) to edit

Double-click (or enter) to edit

```
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
```

```
np.std(a3)
```

```
6.922186552431729
```