```
In [2]: import numpy as np
```

identity matrix

```
 In [3]: np.eye(3,3)

Out[3]: array([[1., 0., 0.],
               [0., 1., 0.],
               [0., 0., 1.]])
```

convert one 1d array into 2d array

```
 In [6]: a1 = np.arange(12)
         a1

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

```
 In [7]: np.reshape(a1,(3,4))

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

```
 In [11]: a1.reshape(3,4)

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

```
 In [12]: a1

Out[12]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

Resize existing array

```
 In [14]: np.resize(a1, (3,4))

Out[14]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11]])
```

```
 In [15]: a1

Out[15]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

```
 In [18]: a1.resize(3,4)
          a1

Out[18]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11]])
```

```
 In [19]: np.sum(a1)

Out[19]: 66
```

```
In [21]: np.sum(a1, 0)

Out[21]: array([12, 15, 18, 21])
```

```
In [22]: np.sum(a1, 1)

Out[22]: array([ 6, 22, 38])
```

```
In [23]: np.mean(a1)

Out[23]: 5.5
```

```
In [24]: a1

Out[24]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11]])
```

```
In [25]: np.mean(a1, 0)

Out[25]: array([4., 5., 6., 7.])
```

```
In [26]: np.mean(a1, 1)

Out[26]: array([1.5, 5.5, 9.5])
```

```
In [27]: a1.min()

Out[27]: 0
```

```
In [28]: a1.min(0)

Out[28]: array([0, 1, 2, 3])
```

```
In [29]: a1.min(1)

Out[29]: array([0, 4, 8])
```

```
In [30]: a1

Out[30]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11]])
```

```
In [51]:  np.max(a1)

Out[51]: 11
```

```
In [32]: np.max(a1,0)

Out[32]: array([ 8,  9, 10, 11])
```

```
In [34]: a2 = np.arange(1,5)
         a3 = np.array([1,5,3,2])
         print(a2, a3)

         [1 2 3 4] [1 5 3 2]
```

```
In [35]: np.where([a2 == a3])

Out[35]: (array([0, 0]), array([0, 2]))
```

Matrix

Dot Product

```
In [36]: a1
```

```
Out[36]: array([[ 0,  1,  2,  3],
               [ 4,  5,  6,  7],
               [ 8,  9, 10, 11]])
```

```
In [37]: a2
```

```
Out[37]: array([1, 2, 3, 4])
```

```
In [38]: a3
```

```
Out[38]: array([1, 5, 3, 2])
```

```
In [44]: arr2 = np.array([[1,4],[3,4]])
         arr3 = np.array([[5,7],[2,1]])
         arr2.dot(arr3)
```

```
Out[44]: array([[13, 11],
               [23, 25]])
```

3D Array

```
In [47]: arr3d = np.arange(64)
         arr3d.resize([3,3,3])
         arr3d
```

```
Out[47]: 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],
                [24, 25, 26]]])
```

```
In [49]: arr3d.ndim
```

```
Out[49]: 3
```

3d array shape return depth, row, coloums

2d array shape return row, coloums

```
In [53]: arr3d.shape
```

```
Out[53]: (3, 3, 3)
```

flatten() Return a copy of the array collapsed into one dimension.

In [58]:
```
arr3d.flatten()
```

Out[58]:
```
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, 24, 25, 26])
```