

```
In [1]: mat=[2,4,6,8],[3,5,7,8],[12,14,18,19],[11,12,13,45]
        for row in mat:
            for col in range(len(mat)):
                row[col]=row[col]-2
        print(mat)
```

```
[0, 2, 4, 6], [1, 3, 5, 6], [10, 12, 16, 17], [9, 10, 11, 43]]
```

```
In [4]: import numpy as np
        a=np.array([[1,2],[3,4],[9,10]])
        a=np.array([1,4,8])
        print(type(a))
        print(type(a1))
```

```
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
```

```
In [7]: np.shape(a1)
```

```
Out[7]: (3, 2)
```

```
In [8]: np.shape(a)
```

```
Out[8]: (3,)
```

```
In [9]: print(a1[2])
```

```
[ 9 10]
```

```
In [10]: print(a1.ndim)
```

```
2
```

```
In [11]: print(a.ndim)
```

```
1
```

```
In [12]: print(a1[-1])
```

```
[ 9 10]
```

```
In [13]: print(a1[1:2:1])
```

```
[[3 4]]
```

```
In [14]: print(a1[2:2:1])
```

```
[]
```

```
In [15]: Nums=np.array([11,12,21])
        Nums<20
```

```
Out[15]: array([ True,  True, False])
```

```
In [16]: Nums=np.array([11,12,21])
        Nums>20
```

```
Out[16]: array([False, False,  True])
```

```
In [21]: np.random.randint(1,10,60).reshape(2,3,10)
```

```
Out[21]: array([[6, 2, 8, 2, 6, 4, 8, 3, 4, 8],
               [6, 3, 5, 9, 9, 5, 8, 6, 5, 8],
               [3, 3, 8, 9, 6, 7, 2, 6, 6, 3]],

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

```
In [23]: TWOD=py.array([[2,4,6,8,10],[1,3,5,7,9],[1,2,3,4,5]])
print(TWOD)
print("\n AFter Slicing:")
# TWOD=[1:3,1:3]
```

```
[[ 2  4  6  8 10]
 [ 1  3  5  7  9]
 [ 1  2  3  4  5]]
```

AFter Slicing:

```
In [18]: import numpy as np
matrix=np.array([
    [[1,2,3],[4,5,6]],
    [[7,8,9],[10,11,12]],
    [[13,14,15],[16,17,18]]
])
print(matrix)
print('After Reshape function')
print(matrix.reshape(2,3,3))
print(matrix[2][0][1])
print(matrix[2][0][0])
```

```
[[[ 1  2  3]
   [ 4  5  6]]
```

```
[[ 7  8  9]
 [10 11 12]]
```

```
[[13 14 15]
 [16 17 18]]]
```

After Reshape function

```
[[[ 1  2  3]
   [ 4  5  6]
   [ 7  8  9]]
```

```
[[10 11 12]
 [13 14 15]
 [16 17 18]]]
```

14
13

Array Operations

```
In [21]: m1=np.array([[11,22,11,5],[33,4,55,5]])
print(m1.shape)
print(m1)
```

```
(2, 4)
[[11 22 11  5]
 [33  4 55  5]]
```

```
In [28]: m1=np.array([[1,1,1,1],[1,1,1,1],[1,1,1,1]])
          m2=np.array([[1,1,1,1],[1,1,1,1],[1,1,1,1]])
          2*m1
          m1-m1
```

```
Out[28]: array([[0, 0, 0, 0],
                [0, 0, 0, 0],
                [0, 0, 0, 0]])
```

```
In [37]: m1=np.array([[1,2,3],[4,5,6],[7,8,9]])
          print(m1)
          m1.diagonal()
          m1.flatten()
          print(m1.min())
          print(m1.max())
          print(m1.mean())
          print(m1.sum())
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
1
9
5.0
45
```

```
In [39]: m1=np.array([[1,2,3],[4,5,6],[7,8,9]])
          m2=np.array([[10,11,12],[13,14,15],[16,17,18]])
          print(m1)
          print(m2)
          np.hstack((m1,m2))
          np.vstack((m1,m2))
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[10 11 12]
 [13 14 15]
 [16 17 18]]
```

```
Out[39]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [ 7,  8,  9],
                [10, 11, 12],
                [13, 14, 15],
                [16, 17, 18]])
```

```
In [42]: import numpy as np
          seqmat=np.arange(0,5)
          print(seqmat)
```

```
[0 1 2 3 4]
```

```
In [44]: np.arange(1,13).reshape(2,3,2)
```

```
Out[44]: array([[[ 1,  2],
                 [ 3,  4],
                 [ 5,  6]],

                [[ 7,  8],
                 [ 9, 10],
                 [11, 12]]])
```

```
In [47]: np.random.randint(2,10,6)
```

```
Out[47]: array([4, 6, 6, 8, 2, 6])
```

```
In [60]: f_m=np.array([[1,2,3,9],[4,5,6,9],[7,8,9,10]])
         # print(f_m.reshape(3,2,2))
         s_m=np.arange(3,12)
         print(s_m.reshape(1,3,3))
```

```
[[[ 3  4  5]
   [ 6  7  8]
   [ 9 10 11]]]
```

```
In [ ]:
```