

```
In [1]: import numpy as np
```

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

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

```
In [3]: r.shape
```

```
Out[3]: (12,)
```

```
In [5]: r.reshape(1,12)
```

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

```
In [ ]: 12 = 1 x 12  
         2 x 6  
         3 x 4  
         4 x 3  
         6 x 2  
        12 x 1
```

```
In [6]: r.reshape(2,6)
```

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

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

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

```
In [8]: r.reshape(4,3)
```

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

```
In [9]: r.reshape(6,2)
```

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

```
In [10]: r.reshape(12,1)
```

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

```
In [11]: r1 = np.array([[1,2,3,4]])
         r1
```

```
Out[11]: array([[1, 2, 3, 4]])
```

```
In [12]: r1.shape
```

```
Out[12]: (1, 4)
```

```
In [ ]: 1 x 4
        4 x 1
        2 x 2
```

```
In [13]: r1.reshape(4,1)
```

```
Out[13]: array([[1],
                [2],
                [3],
                [4]])
```

```
In [14]: r1.reshape(2,2)
```

```
Out[14]: array([[1, 2],
                [3, 4]])
```

```
In [15]: r2 = np.array([[1,2,3,4,5,6]])
         r2
```

```
Out[15]: array([[1, 2, 3, 4, 5, 6]])
```

```
In [17]: r2.shape
```

```
Out[17]: (1, 6)
```

```
In [19]: r2.reshape(2,3)
```

```
Out[19]: array([[1, 2, 3],
               [4, 5, 6]])
```

```
In [20]: r2.reshape(3,2)
```

```
Out[20]: array([[1, 2],
               [3, 4],
               [5, 6]])
```

```
In [ ]: 1 x 6
        6 x 1
        2 x 3
        3 x 2
```

```
In [22]: ## Convert Ndim to 1 D
a = np.array([[[[1,2]]]])
a
```

```
Out[22]: array([[[[1, 2]]]])
```

```
In [23]: a.ndim
```

```
Out[23]: 4
```

```
In [27]: b = a.flatten()
b
```

```
Out[27]: array([1, 2])
```

```
In [26]: b.ndim
```

```
Out[26]: 1
```

```
In [28]: a.ravel()
```

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

```
In [29]: a.T
```

```
Out[29]: array([[[[1]]],
               [[2]]])
```

```
In [ ]: a = 1 / a==> a-1 = 1/a
```

asarray

```
In [30]: lst = [1,2,3,4]
s = np.asarray(lst) # other data type ----> ndarray
s
```

```
Out[30]: array([1, 2, 3, 4])
```

```
In [31]: s.dtype
```

```
Out[31]: dtype('int32')
```

```
In [32]: type(s)
```

```
Out[32]: numpy.ndarray
```

```
In [33]: lst = (1,2,3,4)
s = np.asarray(lst) # other data type ----> ndarray
s
```

```
Out[33]: array([1, 2, 3, 4])
```

hstack

vstack

```
In [34]: a = np.array([1,2,3,4])
b = np.array([5,6,7,8])
c = np.hstack((a,b))
c
```

```
Out[34]: array([1, 2, 3, 4, 5, 6, 7, 8])
```

```
In [35]: d = np.vstack((a,b))
d
```

```
Out[35]: array([[1, 2, 3, 4],
               [5, 6, 7, 8]])
```

```
In [37]: a.max()
```

```
Out[37]: 4
```

```
In [38]: a.min()
```

```
Out[38]: 1
```

```
In [39]: a.mean()
```

```
Out[39]: 2.5
```

```
In [40]: a.std()
```

```
Out[40]: 1.118033988749895
```

```
In [42]: np.median(a)
```

```
Out[42]: 2.5
```

Load data from file

```
In [45]: filedata = np.genfromtxt("test.csv",delimiter=",",dtype='int32')
filedata
```

```
Out[45]: array([[ -1,  -1,  -1],
                [ -1, 25, 65],
                [ -1, 20, 60],
                [ -1, 15, 55]])
```

```
In [44]: filedata.dtype
```

```
Out[44]: dtype('float64')
```

```
In [49]: filedata[filedata > 50]
```

```
Out[49]: array([65, 60, 55])
```

```
In [50]: filedata[filedata < 50]
```

```
Out[50]: array([-1, -1, -1, -1, 25, -1, 20, -1, 15])
```

```
In [52]: filedata[(filedata > 60) & (filedata < 100) ]
```

```
Out[52]: array([65])
```

```
In [53]: filedata.ndim
```

```
Out[53]: 2
```

```
In [56]: np.any(filedata > 50 ,axis=0)
```

```
Out[56]: array([False, False,  True])
```

```
In [ ]:
```