

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

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

[1 2 3 4 5]

```
In [2]: print(type(food))
```

<class 'numpy.ndarray'>

```
In [3]: #food.mean()
```

```
In [4]: #food[0]
```

*Range

```
In [5]: #p=np.arange(2,10,5)
```

Table

```
In [33]: np.arange(5,55,5)
```

```
Out[33]: array([ 5, 10, 15, 20, 25, 30, 35, 40, 45, 50])
```

Line Space

```
In [7]: np.linspace(5,50,num=7)
```

```
Out[7]: array([ 5. , 12.5, 20. , 27.5, 35. , 42.5, 50. ])
```

```
In [8]: # Data Type
np.ones(56, dtype=np.int64)
```

```
Out[8]: 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, 1, 1, 1, 1,  
               1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

Array Functions

```
In [9]: v=np.array([0.2,0.8,0.87,3,8,9,12])
print(v)
```

```
[ 0.2    0.8    0.87   3.     8.     9.    12. ]
```

```
In [10]: v.sort()
          v
```

Out[10]: array([0.2 , 0.8 , 0.87, 3. , 8. , 9. , 12.])

In [11]: `w=np.array([2,6,7,4,33])`
w

Out[11]: array([2, 6, 7, 4, 33])

In [14]: `#concatenation`
`e=np.concatenate((v,w))`
`print(e)`

[0.2 0.8 0.87 3. 8. 9. 12. 2. 6. 7. 4. 33.]

In [16]: `# 2 Dimentional Array`
`t=np.array([[3,2,1],[1,3,7]])`
t

Out[16]: array([[3, 2, 1],
[1, 3, 7]])

In [17]: `y=np.array([[7,6,5],[8,0,45]])`
y

Out[17]: array([[7, 6, 5],
[8, 0, 45]])

In [22]: `f=np.concatenate((t,y),axis=0)`
f

Out[22]: array([[3, 2, 1],
[1, 3, 7],
[7, 6, 5],
[8, 0, 45]])

In [23]: `f.ndim`

Out[23]: 2

Three dimentional array

In [29]: `A=np.array([[[1,2,3],`
`[3,2,1]],`
`[[5,4,3],`
`[3,4,5]],`
`[[8,9,0],`
`[0,9,8]]])`
`print(A)`

[[[1 2 3]
[3 2 1]]

[[5 4 3]
[3 4 5]]

[[8 9 0]
[0 9 8]]]

```
In [30]: A.ndim
```

```
Out[30]: 3
```

Finding the no of elements in Array

```
In [31]: A.size
```

```
Out[31]: 18
```

Finding the Shape of Array

```
In [32]: A.shape
```

```
Out[32]: (3, 2, 3)
```

Reshaping the Arrays

```
In [40]: a=np.arange(1,13)  
a
```

```
Out[40]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

```
In [41]: a.reshape(4,3)
```

```
Out[41]: array([[ 1,  2,  3],  
                [ 4,  5,  6],  
                [ 7,  8,  9],  
                [10, 11, 12]])
```

```
In [43]: np.reshape(a,newshape=(2,6),order="C")
```

```
Out[43]: array([[ 1,  2,  3,  4,  5,  6],  
                [ 7,  8,  9, 10, 11, 12]])
```

Conversion in 2 Dimentional

```
In [44]: a.shape
```

```
Out[44]: (12,)
```

```
In [46]: c=a[np.newaxis,:]  
c
```

```
Out[46]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]])
```

```
In [47]: #Column wise conversion
```

```
c=a[:,np.newaxis]  
c
```

```
Out[47]: array([[ 1],  
              [ 2],  
              [ 3],  
              [ 4],  
              [ 5],  
              [ 6],  
              [ 7],  
              [ 8],  
              [ 9],  
             [10],  
             [11],  
             [12]])
```

```
In [48]: c.shape
```

```
Out[48]: (12, 1)
```

```
In [ ]:
```