

Numpy Reshape

- The shape of an array can be reshape i.e ; we can convert 1D to 2D , 1D to 3D , 2D to 3D etc
- This is done by calling a function reshape available in numpy module
- Let's see this example in Jupiter notebook

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

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

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

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

// This change is temporary

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

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

```
In [ ]:
```

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

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

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

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

// This change is permanent

```
In [6]: ar = ar.reshape(3,4)
```

```
In [7]: ar
```

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

```
In [ ]:
```

- Now we will **reshape** this array

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

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

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

// Reshaping into 4 rows and 3 columns

```
In [10]: ar = ar.reshape(4,3)

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

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

// converting the above array into a 3D array

```
In [12]: ar = ar.reshape(3,2,2)

In [13]: ar
Out[13]: array([[[ 1,  2],
                 [ 3,  4]],
                [[ 5,  6],
                 [ 7,  8]],
                [[ 9, 10],
                 [11, 12]])
```

- There a function called **flatten** array that convert any array into a 1D array

```
In [20]: ar.flatten()

Out[20]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

