NUMPY

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

```
In [2]: np.__version__

Out[2]: '2.1.3'
```

```
In [3]: l1 = [1,2,3,4,5,6]
        l1

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

NumPy ndarray

```
In [5]: ar1 = np.array(l1)
        ar1

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

```
In [6]: print(type(ar1))

       <class 'numpy.ndarray'>
```

arange method

```
In [14]: np.arange(10)  #starting value: 0  stepping value: 1 end value: 10-1

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

```
In [15]: np.arange(1, 30, 2) #odd numbers

Out[15]: array([ 1,  3,  5,  7,  9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29])
```

```
In [16]: np.arange(2,30, 2 ) #even numbers

Out[16]: array([ 2,  4,  6,  8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28])
```

```
In [12]: np.arange(30, 10) #starting value: 30  end value: 10 that's why it will g

Out[12]: array([], dtype=int64)
```

```
In [13]: np.arange(30,10,-1) # here stepping value is -1

Out[13]: array([30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 1
         4,
                13, 12, 11])
```

zeros method

```
In [17]: np.zeros(5) #create an array filled with zeros by default it is float val
```

```
Out[17]: array([0., 0., 0., 0., 0.])
```

```
In [18]: np.zeros(5, dtype= 'int')
```

```
Out[18]: array([0, 0, 0, 0, 0])
```

```
In [19]: np.zeros([3,3]) #create a 3x3 array filled with zeros
```

```
Out[19]: array([[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]])
```

```
In [20]: np.zeros([10,5]) #create a 10x5 array filled with zeros
```

```
Out[20]: array([[0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.]])
```

```
In [21]: np.zeros([3,6], dtype= 'int')
```

```
Out[21]: array([[0, 0, 0, 0, 0, 0],
                [0, 0, 0, 0, 0, 0],
                [0, 0, 0, 0, 0, 0]])
```

ones method

```
In [22]: np.ones(5) #array filled with one
```

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

```
In [23]: np.ones([3,5]) #3x 5 dimension array
```

```
Out[23]: array([[1., 1., 1., 1., 1.],
                [1., 1., 1., 1., 1.],
                [1., 1., 1., 1., 1.]])
```

```
In [24]: np.ones([3,5], dtype= int)
```

```
Out[24]: array([[1, 1, 1, 1, 1],
                [1, 1, 1, 1, 1],
                [1, 1, 1, 1, 1]])
```

```
In [ ]:    random method
```

```
In [25]: np.random.rand(3,2) #generate an array with random numbers
```

```
Out[25]: array([[0.26989969, 0.89245172],
                [0.70510701, 0.98309912],
                [0.73088355, 0.96824591]])
```

```
In [26]: np.random.rand(5)

Out[26]: array([0.96436208, 0.3190153 , 0.64793135, 0.48808007, 0.84242821])

In [27]: np.random.rand(5,7)

Out[27]: array([[0.7923851 , 0.84677698, 0.25702073, 0.56434987, 0.51975073,
                 0.09202311, 0.48818138],
                [0.78272154, 0.77463189, 0.58961901, 0.62737049, 0.9568778 ,
                 0.98416205, 0.30160345],
                [0.32122181, 0.61015048, 0.88696536, 0.57240646, 0.24430188,
                 0.85481164, 0.05864836],
                [0.5465196 , 0.03795187, 0.16469408, 0.22151803, 0.59040347,
                 0.72152349, 0.65050451],
                [0.79701743, 0.10065755, 0.35600252, 0.53791282, 0.01006066,
                 0.43343544, 0.87932763]])

In [28]: np.random.randint(2,7) #random integer number

Out[28]: 3

In [29]: np.random.randint([2,7])

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

In [30]: np.random.randint(1,8)

Out[30]: 1

In [31]: np.random.randint(0,10,2) #fist:- starting number (included) Second:-endi

Out[31]: array([0, 0])

In [32]: np.random.randint(0,10,2)

Out[32]: array([4, 4])

In [33]: np.random.randint(0,5, 8)

Out[33]: array([2, 1, 1, 4, 0, 1, 0, 1])

In [34]: n = np.random.randint(10,40,(8,10)) #(8,10):- shape of array 8rows, 10col

In [35]: n

Out[35]: array([[36, 29, 16, 15, 23, 38, 21, 12, 12, 22],
                [20, 17, 20, 23, 29, 33, 12, 38, 32, 18],
                [26, 37, 11, 10, 21, 28, 27, 32, 13, 23],
                [22, 11, 19, 17, 30, 39, 31, 39, 22, 28],
                [25, 24, 20, 10, 12, 33, 21, 17, 30, 15],
                [21, 27, 14, 17, 36, 10, 37, 29, 17, 12],
                [16, 29, 38, 11, 25, 16, 21, 13, 21, 10],
                [19, 26, 23, 20, 23, 17, 28, 38, 18, 22]])

In [36]: n[5] #5th row
```

```
Out[36]: array([21, 27, 14, 17, 36, 10, 37, 29, 17, 12])
```

```
In [37]: n[6]
```

```
Out[37]: array([16, 29, 38, 11, 25, 16, 21, 13, 21, 10])
```

```
In [38]: n[0:6] #0th row to 5th row
```

```
Out[38]: array([[36, 29, 16, 15, 23, 38, 21, 12, 12, 22],
                [20, 17, 20, 23, 29, 33, 12, 38, 32, 18],
                [26, 37, 11, 10, 21, 28, 27, 32, 13, 23],
                [22, 11, 19, 17, 30, 39, 31, 39, 22, 28],
                [25, 24, 20, 10, 12, 33, 21, 17, 30, 15],
                [21, 27, 14, 17, 36, 10, 37, 29, 17, 12]])
```

```
In [39]: n
```

```
Out[39]: array([[36, 29, 16, 15, 23, 38, 21, 12, 12, 22],
                [20, 17, 20, 23, 29, 33, 12, 38, 32, 18],
                [26, 37, 11, 10, 21, 28, 27, 32, 13, 23],
                [22, 11, 19, 17, 30, 39, 31, 39, 22, 28],
                [25, 24, 20, 10, 12, 33, 21, 17, 30, 15],
                [21, 27, 14, 17, 36, 10, 37, 29, 17, 12],
                [16, 29, 38, 11, 25, 16, 21, 13, 21, 10],
                [19, 26, 23, 20, 23, 17, 28, 38, 18, 22]])
```

```
In [40]: n[::-1] #rows reverse upper to lower and lower to upper
```

```
Out[40]: array([[19, 26, 23, 20, 23, 17, 28, 38, 18, 22],
                [16, 29, 38, 11, 25, 16, 21, 13, 21, 10],
                [21, 27, 14, 17, 36, 10, 37, 29, 17, 12],
                [25, 24, 20, 10, 12, 33, 21, 17, 30, 15],
                [22, 11, 19, 17, 30, 39, 31, 39, 22, 28],
                [26, 37, 11, 10, 21, 28, 27, 32, 13, 23],
                [20, 17, 20, 23, 29, 33, 12, 38, 32, 18],
                [36, 29, 16, 15, 23, 38, 21, 12, 12, 22]])
```

```
In [41]: n
```

```
Out[41]: array([[36, 29, 16, 15, 23, 38, 21, 12, 12, 22],
                [20, 17, 20, 23, 29, 33, 12, 38, 32, 18],
                [26, 37, 11, 10, 21, 28, 27, 32, 13, 23],
                [22, 11, 19, 17, 30, 39, 31, 39, 22, 28],
                [25, 24, 20, 10, 12, 33, 21, 17, 30, 15],
                [21, 27, 14, 17, 36, 10, 37, 29, 17, 12],
                [16, 29, 38, 11, 25, 16, 21, 13, 21, 10],
                [19, 26, 23, 20, 23, 17, 28, 38, 18, 22]])
```

```
In [42]: n[::2] #take every 2nd row from the 1st row
```

```
Out[42]: array([[36, 29, 16, 15, 23, 38, 21, 12, 12, 22],
                [26, 37, 11, 10, 21, 28, 27, 32, 13, 23],
                [25, 24, 20, 10, 12, 33, 21, 17, 30, 15],
                [16, 29, 38, 11, 25, 16, 21, 13, 21, 10]])
```

```
In [43]: n
```

Out[43]:
```
array([[36, 29, 16, 15, 23, 38, 21, 12, 12, 22],
       [20, 17, 20, 23, 29, 33, 12, 38, 32, 18],
       [26, 37, 11, 10, 21, 28, 27, 32, 13, 23],
       [22, 11, 19, 17, 30, 39, 31, 39, 22, 28],
       [25, 24, 20, 10, 12, 33, 21, 17, 30, 15],
       [21, 27, 14, 17, 36, 10, 37, 29, 17, 12],
       [16, 29, 38, 11, 25, 16, 21, 13, 21, 10],
       [19, 26, 23, 20, 23, 17, 28, 38, 18, 22]])
```

In [45]:
```
n[0:5]
```

Out[45]:
```
array([[36, 29, 16, 15, 23, 38, 21, 12, 12, 22],
       [20, 17, 20, 23, 29, 33, 12, 38, 32, 18],
       [26, 37, 11, 10, 21, 28, 27, 32, 13, 23],
       [22, 11, 19, 17, 30, 39, 31, 39, 22, 28],
       [25, 24, 20, 10, 12, 33, 21, 17, 30, 15]])
```

In [46]:
```
n
```

Out[46]:
```
array([[36, 29, 16, 15, 23, 38, 21, 12, 12, 22],
       [20, 17, 20, 23, 29, 33, 12, 38, 32, 18],
       [26, 37, 11, 10, 21, 28, 27, 32, 13, 23],
       [22, 11, 19, 17, 30, 39, 31, 39, 22, 28],
       [25, 24, 20, 10, 12, 33, 21, 17, 30, 15],
       [21, 27, 14, 17, 36, 10, 37, 29, 17, 12],
       [16, 29, 38, 11, 25, 16, 21, 13, 21, 10],
       [19, 26, 23, 20, 23, 17, 28, 38, 18, 22]])
```

In [47]:
```
n[5,-3]
```

Out[47]:
```
np.int64(29)
```

In [48]:
```
v = int(n[5,-3])
```

In [49]:
```
v
```

Out[49]:
```
29
```

In [50]:
```
n[2,3]
```

Out[50]:
```
np.int64(10)
```

In [54]:
```
import numpy as np
x = np.int64(12)   #for storing integer 64 bits space is required
print(x)
```

```
12
```

In [55]:
```
import numpy as np
x = np.int32(7) #for storing integer 32 bit space is required
print(x)
```

```
7
```

reshape method

In [57]:
```
np.arange(1,13).reshape(6,2) #reshape(6, 2):- s 1D array ko 6 rows × 2 co
```

```
Out[57]: array([[ 1,  2],
               [ 3,  4],
               [ 5,  6],
               [ 7,  8],
               [ 9, 10],
               [11, 12]])
```

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```