

NUMPY

- Numerical Python, popularly known as numpy has been designed to carry out mathematical computations at faster and easier rate.
- In command line
 - pip install numpy
- Anaconda distribution
 - conda install numpy

```
In [1]: pip install numpy
```

Requirement already satisfied: numpy in c:\users\alekhya\anaconda3\lib\site-packages (1.16.2)
Note: you may need to restart the kernel to use updated packages.

```
In [3]: import numpy as np
```

```
In [4]: np.__version__
```

```
Out[4]: '1.16.2'
```

- 1d-array - vector(single row)
- 2d-array -m*n
- 3d-array -m*n

```
In [5]: arr_1d = np.array([1,2,3])  
arr_1d
```

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

```
In [6]: arr_1d.ndim
```

```
Out[6]: 1
```

```
In [7]: arr_1d.dtype
```

```
Out[7]: dtype('int32')
```

```
In [8]: arr1 = np.array([1,2.3,"apssdc"])
arr1
```

```
Out[8]: array(['1', '2.3', 'apssdc'], dtype='<U32')
```

```
In [9]: arr2 = np.array("15-09-2020")
arr2
```

```
Out[9]: array('15-09-2020', dtype='<U10')
```

- "b" - boolean
- "i" - (signed)integer
- "u" - (unsigned)integer
- "f" - floating-point
- "c" - complex-floating point
- "m" - timedelta
- "M" - datetime
- "O" - objects
- "U" - unicode
- "V" - rawdata(void)

```
In [10]: arr3 = np.array(3+2j)
arr3.dtype
```

```
Out[10]: dtype('complex128')
```

```
In [12]: np.arange(1,10,2)
```

```
Out[12]: array([1, 3, 5, 7, 9])
```

```
In [13]: range(1,10.5)
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-13-83f1062f942f> in <module>  
----> 1 range(1,10.5)
```

```
TypeError: 'float' object cannot be interpreted as an integer
```

```
In [15]: np.arange(1,10.5,1.5) # start,stop,stepvalue
```

```
Out[15]: array([ 1. ,  2.5,  4. ,  5.5,  7. ,  8.5, 10. ])
```

```
In [17]: np.linspace(1,5,10) #start,stop,number of parts
```

```
Out[17]: array([1.          , 1.44444444, 1.88888889, 2.33333333, 2.77777778,  
                3.22222222, 3.66666667, 4.11111111, 4.55555556, 5.          ])
```

```
In [20]: arr1.shape
```

```
Out[20]: (3,)
```

```
In [21]: arr1.size
```

```
Out[21]: 3
```

```
In [22]: f = np.arange(1,51)  
f
```

```
Out[22]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,  
                18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,  
                35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50])
```

```
In [24]: f1 = f.reshape(5,10)
f1
```

```
Out[24]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10],
               [11, 12, 13, 14, 15, 16, 17, 18, 19, 20],
               [21, 22, 23, 24, 25, 26, 27, 28, 29, 30],
               [31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
               [41, 42, 43, 44, 45, 46, 47, 48, 49, 50]])
```

```
In [25]: f1.ndim
```

```
Out[25]: 2
```

```
In [26]: f1.shape
```

```
Out[26]: (5, 10)
```

```
In [27]: f1.size
```

```
Out[27]: 50
```

```
In [29]: f.reshape(10,5)
```

```
Out[29]: array([[ 1,  2,  3,  4,  5],
               [ 6,  7,  8,  9, 10],
               [11, 12, 13, 14, 15],
               [16, 17, 18, 19, 20],
               [21, 22, 23, 24, 25],
               [26, 27, 28, 29, 30],
               [31, 32, 33, 34, 35],
               [36, 37, 38, 39, 40],
               [41, 42, 43, 44, 45],
               [46, 47, 48, 49, 50]])
```

```
In [31]: f.reshape(2,25)
```

```
Out[31]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                17, 18, 19, 20, 21, 22, 23, 24, 25],
               [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
                42, 43, 44, 45, 46, 47, 48, 49, 50]])
```

```
In [32]: f.reshape(2,-1)
```

```
Out[32]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                17, 18, 19, 20, 21, 22, 23, 24, 25],
               [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
                42, 43, 44, 45, 46, 47, 48, 49, 50]])
```

```
In [33]: f.reshape(-1,5)
```

```
Out[33]: array([[ 1,  2,  3,  4,  5],
               [ 6,  7,  8,  9, 10],
               [11, 12, 13, 14, 15],
               [16, 17, 18, 19, 20],
               [21, 22, 23, 24, 25],
               [26, 27, 28, 29, 30],
               [31, 32, 33, 34, 35],
               [36, 37, 38, 39, 40],
               [41, 42, 43, 44, 45],
               [46, 47, 48, 49, 50]])
```

```
In [34]: f.reshape(-1,-1)
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-34-af6f5b90af7b> in <module>
----> 1 f.reshape(-1,-1)
```

ValueError: can only specify one unknown dimension

```
In [37]: f.reshape(-10,5)
```

```
Out[37]: array([[ 1,  2,  3,  4,  5],
                [ 6,  7,  8,  9, 10],
                [11, 12, 13, 14, 15],
                [16, 17, 18, 19, 20],
                [21, 22, 23, 24, 25],
                [26, 27, 28, 29, 30],
                [31, 32, 33, 34, 35],
                [36, 37, 38, 39, 40],
                [41, 42, 43, 44, 45],
                [46, 47, 48, 49, 50]])
```

```
In [38]: f.ndim
```

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

```
In [39]: f1.ndim
```

```
Out[39]: 2
```

```
In [40]: f
```

```
Out[40]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
                18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
                35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50])
```

```
In [41]: f1
```

```
Out[41]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10],
                [11, 12, 13, 14, 15, 16, 17, 18, 19, 20],
                [21, 22, 23, 24, 25, 26, 27, 28, 29, 30],
                [31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
                [41, 42, 43, 44, 45, 46, 47, 48, 49, 50]])
```

```
In [42]: arr4 = np.arange(0,50).reshape(5,1,10)
arr4
```

```
Out[42]: array([[[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9]],
                [[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]],
                [[20, 21, 22, 23, 24, 25, 26, 27, 28, 29]],
                [[30, 31, 32, 33, 34, 35, 36, 37, 38, 39]],
                [[40, 41, 42, 43, 44, 45, 46, 47, 48, 49]])])
```

```
In [43]: arr4.ndim
```

```
Out[43]: 3
```

```
In [44]: arr4.shape
```

```
Out[44]: (5, 1, 10)
```

```
In [50]: arr5 = np.arange(0,50).reshape(1,1,50)
arr5
```

```
Out[50]: array([[[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
                  16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
                  32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
                  48, 49]])])
```

```
In [51]: arr5.reshape(2,25)
```

```
Out[51]: array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
                  16, 17, 18, 19, 20, 21, 22, 23, 24],
                 [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
                  41, 42, 43, 44, 45, 46, 47, 48, 49]])
```

```
In [52]: arr5.reshape(-1,25)
```

```
Out[52]: array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
                16, 17, 18, 19, 20, 21, 22, 23, 24],
               [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
                41, 42, 43, 44, 45, 46, 47, 48, 49]])
```

```
In [54]: x = np.ones(2,dtype=np.int)
x
```

```
Out[54]: array([1, 1])
```

```
In [55]: x = np.ones((10,2),dtype=np.object)
x
```

```
Out[55]: array([[1, 1],
               [1, 1],
               [1, 1],
               [1, 1],
               [1, 1],
               [1, 1],
               [1, 1],
               [1, 1],
               [1, 1],
               [1, 1]], dtype=object)
```

```
In [56]: y = np.zeros((3,4),dtype=float)
```

```
In [57]: y
```

```
Out[57]: array([[0., 0., 0., 0.],
               [0., 0., 0., 0.],
               [0., 0., 0., 0.]])
```

```
In [68]: b = np.eye(2,2)
b
```

```
Out[68]: array([[1., 0.],
               [0., 1.]])
```


Adding and sorting

```
In [69]: l = [1,2,3]
l2 = [3,4,5]
l3 = []
for i in range(len(l)):
    l3.append(l[i]+l2[i])
print(l3)
```

[4, 6, 8]

```
In [70]: a = np.array([1,2,3])
b = np.array([3,4,5])
a+b
```

Out[70]: array([4, 6, 8])

```
In [71]: arr6 = np.array([2,34,1,5.6,4,5.1])
arr6
```

Out[71]: array([2. , 34. , 1. , 5.6, 4. , 5.1])

```
In [72]: np.sort(arr6)
```

Out[72]: array([1. , 2. , 4. , 5.1, 5.6, 34.])

```
In [76]: ar = np.array([[1,2,3],[6,7,8]])
ar
```

Out[76]: array([[1, 2, 3],
 [6, 7, 8]])

```
In [77]: ar.T
```

Out[77]: array([[1, 6],
 [2, 7],
 [3, 8]])

```
In [83]: surnames = ("sdc","apssdc")
first_names = ("telengana","andhrapradesh")
ind = np.lexsort((first_names,surnames))
ind
```

```
Out[83]: array([1, 0], dtype=int64)
```

```
In [84]: [surnames[i]+" "+first_names[i] for i in ind]
```

```
Out[84]: ['apssdc andhrapradesh', 'sdc telengana']
```

```
In [86]: a = np.array([1,2,3])
b = np.array([7,23,45])
np.concatenate((b,a))
```

```
Out[86]: array([ 7, 23, 45,  1,  2,  3])
```

```
In [90]: x = np.array([[1,2,3],[89,23,45]])
y = np.array([[23,12,34],[56,67,89]])
x
```

```
Out[90]: array([[ 1,  2,  3],
               [89, 23, 45]])
```

```
In [91]: y
```

```
Out[91]: array([[23, 12, 34],
               [56, 67, 89]])
```

```
In [89]: np.concatenate((x,y),axis=0)
```

```
Out[89]: array([[ 1,  2,  3],
               [89, 23, 45],
               [23, 12, 34],
               [56, 67, 89]])
```

```
In [92]: np.concatenate((x,y),axis=1)
```

```
Out[92]: array([[ 1,  2,  3, 23, 12, 34],
               [89, 23, 45, 56, 67, 89]])
```

Indexing

```
In [94]: f = np.arange(1,100,2)
         f
```

```
Out[94]: array([ 1,  3,  5,  7,  9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33,
                35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67,
                69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99])
```

```
In [95]: f[1:20]
```

```
Out[95]: array([ 3,  5,  7,  9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35,
                37, 39])
```

```
In [96]: f[5]
```

```
Out[96]: 11
```

```
In [97]: f[-1]
```

```
Out[97]: 99
```

```
In [98]: f[::-1]# start:stop:step
```

```
Out[98]: array([99, 97, 95, 93, 91, 89, 87, 85, 83, 81, 79, 77, 75, 73, 71, 69, 67,
                65, 63, 61, 59, 57, 55, 53, 51, 49, 47, 45, 43, 41, 39, 37, 35, 33,
                31, 29, 27, 25, 23, 21, 19, 17, 15, 13, 11,  9,  7,  5,  3,  1])
```

```
In [ ]: # 2d and 3d
```

```
In [99]: a1 = np.array([[1,1],[2,2]])
         a2 = np.array([[3,3],[4,4]])
         a1,a2
```

```
Out[99]: (array([[1, 1],
                [2, 2]]), array([[3, 3],
                [4, 4]]))
```

```
In [100]: np.vstack((a1,a2)) #axis=0-rows
```

```
Out[100]: array([[1, 1],  
                [2, 2],  
                [3, 3],  
                [4, 4]])
```

```
In [101]: np.hstack((a1,a2)) #axis=1-columns
```

```
Out[101]: array([[1, 1, 3, 3],  
                [2, 2, 4, 4]])
```

```
In [156]: x = np.arange(1,25).reshape(2,12)  
x
```

```
Out[156]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12],  
                [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]])
```

```
In [157]: np.hsplit(x,6)
```

```
Out[157]: [array([[ 1,  2],  
                [13, 14]]), array([[ 3,  4],  
                [15, 16]]), array([[ 5,  6],  
                [17, 18]]), array([[ 7,  8],  
                [19, 20]]), array([[ 9, 10],  
                [21, 22]]), array([[11, 12],  
                [23, 24]])]
```

```
In [159]: np.vsplit(x,2)
```

```
Out[159]: [array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]]),  
          array([[13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]])]
```

Mathematical operations

```
In [118]: a = np.array([1,2,3])  
b = np.array([4,5,6])  
print(a+b)  
print(a-b)  
print(a/b)  
print(a*b)  
print(a//b)
```

```
[5 7 9]  
[-3 -3 -3]  
[0.25 0.4 0.5 ]  
[ 4 10 18]  
[0 0 0]
```

```
In [111]: np.divide(a,b)
```

```
Out[111]: array([0.25, 0.4 , 0.5 ])
```

```
In [112]: np.floor_divide(a,b)
```

```
Out[112]: array([0, 0, 0], dtype=int32)
```

```
In [113]: np.divmod(a,b)
```

```
Out[113]: (array([0, 0, 0], dtype=int32), array([1, 2, 3], dtype=int32))
```

```
In [114]: a%b
```

```
Out[114]: array([1, 2, 3], dtype=int32)
```

```
In [119]: a/b
```

```
Out[119]: array([0.25, 0.4 , 0.5 ])
```

```
In [121]: a = np.array([[1,2],[3,4]])
```

```
In [122]: a.sum(axis=0)
```

```
Out[122]: array([4, 6])
```

```
In [123]: a.sum(axis=1)
```

```
Out[123]: array([3, 7])
```

- trigonometric Functions

```
In [124]: np.sin(np.pi/2)
```

```
Out[124]: 1.0
```

```
In [125]: np.sin(np.pi/3)
```

```
Out[125]: 0.8660254037844386
```

```
In [126]: np.sin(np.pi/4)
```

```
Out[126]: 0.7071067811865476
```

- exponential and log

```
In [127]: a = np.array([1,2,3])  
np.log(a)
```

```
Out[127]: array([0.          , 0.69314718, 1.09861229])
```

```
In [128]: np.exp(a)
```

```
Out[128]: array([ 2.71828183,  7.3890561 , 20.08553692])
```

```
In [130]: np.log10(a)#  $\log_2(a)$ 
```

```
Out[130]: array([0.          , 0.30103   , 0.47712125])
```

```
In [140]: c = np.array([4.6,4.678,4.2,5.9,1.01])  
          np.round_(c)
```

```
Out[140]: array([5., 5., 4., 6., 1.])
```

```
In [141]: np.floor(c)
```

```
Out[141]: array([4., 4., 4., 5., 1.])
```

```
In [142]: np.ceil(c)
```

```
Out[142]: array([5., 5., 5., 6., 2.])
```

Random

```
In [138]: from numpy import random
```

```
In [145]: random.randint(100,size=(5))
```

```
Out[145]: array([ 6, 97, 70, 31, 19])
```

```
In [146]: random.randint(100,size=(5,4))
```

```
Out[146]: array([[41, 81, 23,  3],  
                 [51, 68, 97, 24],  
                 [40, 53, 86, 39],  
                 [57, 79, 22, 41],  
                 [76, 58, 97, 72]])
```

```
In [148]: random.randint(1,100,(5))
```

```
Out[148]: array([96, 98, 70, 52, 52])
```

```
In [149]: random.rand(5) # 0 to 1
```

```
Out[149]: array([0.59911609, 0.57586355, 0.99377898, 0.88268805, 0.46131524])
```

```
In [150]: random.rand(5,5) # 0 to 1
```

```
Out[150]: array([[0.93389973, 0.82492524, 0.75717449, 0.14254909, 0.18261763],
 [0.27893147, 0.26956043, 0.49728998, 0.63252403, 0.71317433],
 [0.07715539, 0.81731406, 0.16919288, 0.28686757, 0.35935828],
 [0.98638432, 0.18977658, 0.30049893, 0.78824519, 0.72929521],
 [0.18355132, 0.4232984 , 0.33147094, 0.68155206, 0.08539143]])
```

```
In [152]: random.choice([3,7,5],size=(3,5))
```

```
Out[152]: array([[3, 3, 3, 3, 5],
 [5, 3, 3, 3, 7],
 [7, 7, 7, 7, 5]])
```

```
In [153]: random.choice([3,7,5],p=[0.1,0.6,0.3],size=(100))
```

```
Out[153]: array([7, 7, 5, 7, 7, 7, 7, 7, 7, 7, 5, 7, 7, 5, 5, 5, 7, 7, 5, 5, 5, 5,
 5, 5, 3, 7, 7, 7, 7, 5, 3, 7, 7, 7, 7, 7, 5, 5, 5, 5, 7, 7, 7, 7,
 7, 5, 7, 7, 5, 7, 5, 7, 3, 7, 5, 7, 7, 7, 3, 7, 7, 7, 7, 7, 5, 7,
 7, 5, 5, 3, 5, 5, 7, 7, 7, 7, 7, 5, 5, 7, 7, 7, 5, 7, 7, 7, 7, 5,
 3, 7, 7, 5, 7, 7, 7, 7, 7, 7, 7, 7])
```

```
In [155]: random.choice([3,7,5,6],p=[0.1,0.5,0.4,0.0],size=(2,4))
```

```
Out[155]: array([[7, 5, 5, 7],
 [5, 5, 7, 5]])
```

```
In [ ]:
```