

Creating a basic array

```
In [1]: # creating a List and converting to array  
my_list = [1,2,3,4,5,6,7,8,9,10]  
my_list
```

```
Out[1]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
In [2]: import numpy as np
```

```
In [3]: np.array(my_list) # converted list to 1d array
```

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

create 1d array

```
In [4]: np.array(my_list)
```

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

create 2d array

```
In [5]: np.array(my_list).reshape(2,5)
```

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

create nd array

```
In [6]: np.array(my_list).reshape(5,2)
```

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

arange() --> [start,stop,step] , dtype=None

```
In [7]: np.arange(10,30,2)
```

```
Out[7]: array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28])
```

```
In [8]: np.arange(10)
```

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

```
In [9]: np.arange(-10,20)
```

```
Out[9]: array([-10, -9, -8, -7, -6, -5, -4, -3, -2, -1,  0,  1,  2,
   3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
  16, 17, 18, 19])
```

zeros() ---> it creates an array with 0's

```
In [10]: np.zeros(3) # default it takes float datatype --> parameter tuning
```

```
Out[10]: array([0., 0., 0.])
```

```
In [11]: np.zeros((4),dtype=int) # default value changed to user required parameter --> hyperparameter tuning
```

```
Out[11]: array([0, 0, 0, 0])
```

ones() ----> an array with 1's

```
In [12]: np.ones(3)      # default will give float values
```

```
Out[12]: array([1., 1., 1.])
```

```
In [13]: np.ones((4),dtype=int)
```

```
Out[13]: array([1, 1, 1, 1])
```

np.sort() & np.concatenate()

```
In [14]: a=[2,3,15,24,6,7,8,9,39,7]  
a
```

```
Out[14]: [2, 3, 15, 24, 6, 7, 8, 9, 39, 7]
```

```
In [15]: a.sort()
```

```
In [16]: a
```

```
Out[16]: [2, 3, 6, 7, 7, 8, 9, 15, 24, 39]
```

concatenate()

```
In [17]: a=np.array([2,3,4,5,7,5])  
b=np.array([3,4,7,8,9])
```

```
In [18]: np.concatenate((a,b))
```

```
Out[18]: array([2, 3, 4, 5, 7, 5, 3, 4, 7, 8, 9])
```

```
In [19]: x = np.array([[1, 2], [3, 4]])  
y = np.array([[5, 6]])
```

```
In [20]: np.concatenate((x,y))
```

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

ndarray.ndim --will tell you the number of axes, or dimensions, of the array.

ndarray.size --will tell you the total number of elements of the array.

ndarray.shape -- will give dimension of the array.

```
In [21]: a=np.array([[[0, 1, 2, 3],  
                   [4, 5, 6, 7]],  
  
                   [[[0, 1, 2, 3],  
                     [4, 5, 6, 7]],  
  
                   [[[0 ,1 ,2, 3],  
                     [4, 5, 6, 7]]])
```

```
In [22]: a
```

```
Out[22]: array([[[0, 1, 2, 3],  
                  [4, 5, 6, 7]],  
  
                  [[[0, 1, 2, 3],  
                    [4, 5, 6, 7]],  
  
                  [[[0, 1, 2, 3],  
                    [4, 5, 6, 7]]])
```

```
In [23]: a.ndim      # dimension of array its 3d array
```

```
Out[23]: 3
```

```
In [24]: a.size      # total number of elements in an array
```

```
Out[24]: 24
```

```
In [25]: a.shape     # shape of an array
```

```
Out[25]: (3, 2, 4)
```

reshape() --> to reshape an array.ex: you can reshape an array with rows and columns

```
In [26]: a=np.arange(12)  
print(a)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11]
```

```
In [27]: a.reshape(3,4)
```

```
Out[27]: array([[ 0,  1,  2,  3],  
                 [ 4,  5,  6,  7],  
                 [ 8,  9, 10, 11]])
```

```
In [28]: a.reshape(4,3)
```

```
Out[28]: array([[ 0,  1,  2],  
                 [ 3,  4,  5],  
                 [ 6,  7,  8],  
                 [ 9, 10, 11]])
```

```
In [29]: a.reshape(6,2)
```

```
Out[29]: array([[ 0,  1],
   [ 2,  3],
   [ 4,  5],
   [ 6,  7],
   [ 8,  9],
  [10, 11]])
```

Indexing and slicing in Numpy

```
In [30]: a = np.array([1,2,3,4,5,6,7,8])
```

```
In [31]: a[0]
```

```
Out[31]: 1
```

```
In [32]: a[2]
```

```
Out[32]: 3
```

```
In [33]: a[3:6]
```

```
Out[33]: array([4, 5, 6])
```

```
In [34]: a[-4:-1]
```

```
Out[34]: array([5, 6, 7])
```

```
In [35]: a[-6:7]
```

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

np.random.rand() --> will generate random values in between given range with default float values

```
In [36]: np.random.rand(4)
```

```
Out[36]: array([0.72415912, 0.80971898, 0.35651646, 0.50070835])
```

```
In [37]: np.random.rand(3,5)
```

```
Out[37]: array([[0.67636678, 0.91036541, 0.66084729, 0.0594143 , 0.7635594 ],
 [0.66070451, 0.51372863, 0.8675753 , 0.13607942, 0.45040578],
 [0.1568512 , 0.04651007, 0.60674275, 0.64795158, 0.97175474]])
```

np.random.randint() --> will generate random values in between given range with default interger values ----->[start , stop , size]

```
In [38]: np.random.randint(2,4)
```

```
Out[38]: 3
```

```
In [39]: np.random.randint(2,20) # 2nd argument is exclusive with (n-1)
```

```
Out[39]: 16
```

```
In [40]: np.random.randint(10,20,4)
```

```
Out[40]: array([13, 19, 11, 14])
```

```
In [41]: np.random.randint(-30,20,5)
```

```
Out[41]: array([-21, -19,  11, -29,  -6])
```

```
In [42]: np.random.randint(10,40,(10,10)) # [start , stop (10 x 10 matrix) ]
```

```
Out[42]: array([[17, 31, 13, 16, 13, 17, 33, 14, 26, 28],  
   [10, 30, 16, 14, 25, 35, 20, 24, 38, 24],  
   [22, 20, 10, 34, 37, 17, 15, 11, 28, 14],  
   [16, 30, 36, 26, 14, 10, 36, 13, 10, 31],  
   [26, 31, 38, 36, 26, 35, 20, 33, 14, 25],  
   [23, 10, 17, 31, 10, 10, 13, 30, 25, 29],  
   [37, 13, 36, 16, 29, 20, 14, 12, 24, 23],  
   [23, 22, 16, 23, 12, 28, 11, 21, 20, 38],  
   [14, 25, 25, 20, 10, 18, 31, 25, 13, 31],  
   [16, 25, 29, 16, 38, 35, 36, 12, 10, 37]])
```

```
In [43]: np.random.randint(1,100,(12,12))
```

```
Out[43]: array([[39, 55,  8, 57, 19, 96, 50, 22, 46,  6, 43, 22],  
   [ 9, 34, 27, 54,  7, 54, 38, 70, 40, 88, 58, 46],  
   [ 8, 97, 12, 33, 58, 59,  5, 69, 70, 76, 93, 27],  
   [39, 61,  8, 90, 70, 33, 28, 74, 92,  7,  8, 35],  
   [15, 22, 70, 34, 12, 53,  7, 76, 89, 93, 14, 78],  
   [16, 35,  5, 10, 49, 13, 68, 77, 44, 49, 25, 36],  
   [ 2,  1, 84, 77, 62, 36, 92, 81, 13, 82, 30, 37],  
   [72, 13, 55, 21, 77, 22, 73, 42, 45, 79, 28, 99],  
   [65, 53, 23, 33, 21, 91, 95,  8, 48, 51, 21, 60],  
   [49, 55, 17, 50, 81, 50, 61, 98, 81, 91,  2,  6],  
   [81, 30, 67, 77,  1, 96, 19, 95, 26, 59, 70, 45],  
   [58, 66, 33, 19,  6, 59, 98, 37, 56, 26, 72, 50]])
```

```
In [44]: b = np.random.randint(10,20,(5,4))  
b
```

```
Out[44]: array([[14, 12, 11, 16],  
   [15, 18, 14, 12],  
   [13, 16, 16, 11],  
   [19, 12, 18, 13],  
   [13, 12, 16, 19]])
```

```
In [45]: type(b)
```

```
Out[45]: numpy.ndarray
```

```
In [46]: b
```

```
Out[46]: array([[14, 12, 11, 16],  
                 [15, 18, 14, 12],  
                 [13, 16, 16, 11],  
                 [19, 12, 18, 13],  
                 [13, 12, 16, 19]])
```

```
In [47]: b[:]
```

```
Out[47]: array([[14, 12, 11, 16],  
                 [15, 18, 14, 12],  
                 [13, 16, 16, 11],  
                 [19, 12, 18, 13],  
                 [13, 12, 16, 19]])
```

```
In [48]: b[1:3]
```

```
Out[48]: array([[15, 18, 14, 12],  
                 [13, 16, 16, 11]])
```

```
In [49]: b
```

```
Out[49]: array([[14, 12, 11, 16],  
                 [15, 18, 14, 12],  
                 [13, 16, 16, 11],  
                 [19, 12, 18, 13],  
                 [13, 12, 16, 19]])
```

```
In [50]: b[3:-1]
```

```
Out[50]: array([[19, 12, 18, 13]])
```

```
In [51]: b[-3:3]
```

```
Out[51]: array([[13, 16, 16, 11]])
```

```
In [52]: b[1,2]
```

```
Out[52]: 14
```

```
In [53]: b[1,3]
```

```
Out[53]: 12
```

```
In [54]: b[1,-1]
```

```
Out[54]: 12
```

```
In [55]: b[-1,2]
```

```
Out[55]: 16
```

```
In [56]: b
```

```
Out[56]: array([[14, 12, 11, 16],  
                 [15, 18, 14, 12],  
                 [13, 16, 16, 11],  
                 [19, 12, 18, 13],  
                 [13, 12, 16, 19]])
```

```
In [57]: b[3,-2]
```

```
Out[57]: 18
```

operations

```
In [58]: a = np.random.randint(10,20,10)  
a
```

```
Out[58]: array([16, 11, 19, 18, 14, 12, 15, 13, 15, 14])
```

```
In [59]: id(a)
```

```
Out[59]: 2379709555856
```

```
In [60]: arr = np.random.randint(0,100,(10,10))
```

```
In [61]: arr
```

```
Out[61]: array([[65, 28, 47, 77, 7, 55, 66, 9, 48, 46],  
   [59, 54, 49, 28, 30, 78, 71, 64, 9, 80],  
   [76, 47, 72, 63, 68, 38, 97, 90, 57, 11],  
   [17, 98, 52, 44, 68, 48, 44, 29, 42, 91],  
   [16, 42, 56, 97, 45, 60, 23, 73, 31, 41],  
   [93, 9, 98, 39, 67, 71, 19, 76, 25, 20],  
   [43, 52, 50, 72, 62, 46, 32, 12, 20, 4],  
   [59, 13, 34, 67, 49, 47, 46, 12, 5, 32],  
   [54, 17, 83, 39, 52, 36, 6, 71, 7, 65],  
   [81, 16, 90, 19, 83, 30, 98, 77, 14, 5]])
```

```
In [62]: arr[0]
```

```
Out[62]: array([65, 28, 47, 77, 7, 55, 66, 9, 48, 46])
```

```
In [63]: arr[0:4]
```

```
Out[63]: array([[65, 28, 47, 77, 7, 55, 66, 9, 48, 46],  
   [59, 54, 49, 28, 30, 78, 71, 64, 9, 80],  
   [76, 47, 72, 63, 68, 38, 97, 90, 57, 11],  
   [17, 98, 52, 44, 68, 48, 44, 29, 42, 91]])
```

```
In [64]: arr[3,5]
```

```
Out[64]: 48
```

```
In [65]: arr[3,-1]
```

```
Out[65]: 91
```

```
In [66]: arr[::-1]
```

```
Out[66]: array([[81, 16, 90, 19, 83, 30, 98, 77, 14, 5],  
                 [54, 17, 83, 39, 52, 36, 6, 71, 7, 65],  
                 [59, 13, 34, 67, 49, 47, 46, 12, 5, 32],  
                 [43, 52, 50, 72, 62, 46, 32, 12, 20, 4],  
                 [93, 9, 98, 39, 67, 71, 19, 76, 25, 20],  
                 [16, 42, 56, 97, 45, 60, 23, 73, 31, 41],  
                 [17, 98, 52, 44, 68, 48, 44, 29, 42, 91],  
                 [76, 47, 72, 63, 68, 38, 97, 90, 57, 11],  
                 [59, 54, 49, 28, 30, 78, 71, 64, 9, 80],  
                 [65, 28, 47, 77, 7, 55, 66, 9, 48, 46]])
```

```
In [67]: arr[::-3]
```

```
Out[67]: array([[81, 16, 90, 19, 83, 30, 98, 77, 14, 5],  
                 [43, 52, 50, 72, 62, 46, 32, 12, 20, 4],  
                 [17, 98, 52, 44, 68, 48, 44, 29, 42, 91],  
                 [65, 28, 47, 77, 7, 55, 66, 9, 48, 46]])
```

```
In [68]: arr[::-2]
```

```
Out[68]: array([[65, 28, 47, 77, 7, 55, 66, 9, 48, 46],  
                 [76, 47, 72, 63, 68, 38, 97, 90, 57, 11],  
                 [16, 42, 56, 97, 45, 60, 23, 73, 31, 41],  
                 [43, 52, 50, 72, 62, 46, 32, 12, 20, 4],  
                 [54, 17, 83, 39, 52, 36, 6, 71, 7, 65]])
```

```
In [69]: arr.max()
```

```
Out[69]: 98
```

```
In [70]: arr.min()
```

```
Out[70]: 4
```

```
In [71]: arr.mean()
```

```
Out[71]: 48.23
```

```
In [72]: arr.median()
```

```
-----  
AttributeError  
Cell In[72], line 1  
----> 1 arr.median()
```

Traceback (most recent call last)

```
AttributeError: 'numpy.ndarray' object has no attribute 'median'
```

```
In [73]: from numpy import *  
a= array([1,2,3,4,9])  
median(a)
```

```
Out[73]: 3.0
```

```
In [74]: arr1 = np.array([1,2,3,4,5,6,7,9])
```

```
In [75]: arr1.reshape(2,4,order='C')
```

```
Out[75]: array([[1, 2, 3, 4],  
                [5, 6, 7, 9]])
```

```
In [76]: arr1.reshape(2,4,order='F')
```

```
Out[76]: array([[1, 3, 5, 7],  
                [2, 4, 6, 9]])
```

```
In [77]: arr1.reshape(2,4,order='A')
```

```
Out[77]: array([[1, 2, 3, 4],  
                [5, 6, 7, 9]])
```

Masking

```
In [82]: mat = np.random.randint(0,101,(10,10))
```

```
In [83]: mat
```

```
Out[83]: array([[ 58,  65,  61,  65,  80,  90,  23,   0,   1,   6],
   [ 19,  44,   0,  28,  95,  91,  81,  27,  25,  72],
   [ 17,  92,  31,  48,  66,  31,  89,  88,   3,  58],
   [ 37,  99,  83,   1,  67,  99,  22,  21,  39,  61],
   [ 65,  39,  20,  92,  71,  38,  56,   4,  68,  67],
   [ 98,  37,  90,  76, 100,  39,  75,  39,  86,   4],
   [ 54,  49,  90,  21,  26,   4,  74,  69,  21,  35],
   [  7,  79,  50,  20,  10,  50,  61,  67,  48,  50],
   [ 75,  56,   3,  54,  62,  66,  35,  45,  83,  72],
   [ 56,  91,  87,  73,  82,  82,  71,  57,  86,  79]])
```

```
In [84]: id(mat)
```

```
Out[84]: 2379716638832
```

```
In [85]: mat[mat<50]
```

```
Out[85]: array([23,  0,   1,   6,  19,  44,   0,  28,  27,  25,  17,  31,  48,  31,   3,  37,   1,
   22,  21,  39,  39,  20,  38,   4,  37,  39,  39,   4,  49,  21,  26,   4,  21,  35,
   7,  20,  10,  48,   3,  35,  45])
```

```
In [86]: mat[mat>50]
```

```
Out[86]: array([ 58,  65,  61,  65,  80,  90,  95,  91,  81,  72,  92,  66,  89,
   88,  58,  99,  83,  67,  99,  61,  65,  92,  71,  56,  68,  67,
   98,  90,  76, 100,  75,  86,  54,  90,  74,  69,  79,  61,  67,
   75,  56,  54,  62,  66,  83,  72,  56,  91,  87,  73,  82,  82,
   71,  57,  86,  79])
```

```
In [87]: mat[mat>=50]
```

```
Out[87]: array([ 58,  65,  61,  65,  80,  90,  95,  91,  81,  72,  92,  66,  89,
   88,  58,  99,  83,  67,  99,  61,  65,  92,  71,  56,  68,  67,
   98,  90,  76, 100,  75,  86,  54,  90,  74,  69,  79,  50,  50,
   61,  67,  50,  75,  56,  54,  62,  66,  83,  72,  56,  91,  87,
   73,  82,  82,  71,  57,  86,  79])
```

```
In [89]: mat[mat==50]
```

```
Out[89]: array([50, 50, 50])
```