

30/10/25

NUMPY

```
In [5]: import numpy as np
```

```
In [9]: np.__version__
```

```
Out[9]: '1.26.4'
```

creating a list

```
In [12]: my_list=[0,1,2,3,4,5]
my_list
```

```
Out[12]: [0, 1, 2, 3, 4, 5]
```

```
In [14]: type(my_list)
```

```
Out[14]: list
```

converting list to array

```
In [17]: arr=np.array(my_list)
arr
```

```
Out[17]: array([0, 1, 2, 3, 4, 5])
```

```
In [19]: type(arr)
```

```
Out[19]: numpy.ndarray
```

arange function

```
In [22]: np.arange(0)
```

```
Out[22]: array([], dtype=int32)
```

```
In [24]: np.arange(1)
```

```
Out[24]: array([0])
```

```
In [26]: np.arange(10)
```

```
Out[26]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [28]: np.arange(5.0)
```

```
Out[28]: array([0., 1., 2., 3., 4.])
```

```
In [30]: np.arange(5.2)
```

```
Out[30]: array([0., 1., 2., 3., 4., 5.])
```

```
In [32]: np.arange(9)
```

```
Out[32]: array([0, 1, 2, 3, 4, 5, 6, 7, 8])
```

```
In [34]: np.arange(0,5)
```

```
Out[34]: array([0, 1, 2, 3, 4])
```

```
In [42]: #arange with 3 arguments  
np.arange(0,10,2)
```

```
Out[42]: array([0, 2, 4, 6, 8])
```

```
In [44]: np.arange(0,10,1)
```

```
Out[44]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [46]: np.arange(0,10,-1)
```

```
Out[46]: array([], dtype=int32)
```

```
In [48]: np.arange(10,20,2)
```

```
Out[48]: array([10, 12, 14, 16, 18])
```

```
In [50]: np.arange(20,10) #start argument is always less than end argument
```

```
Out[50]: array([], dtype=int32)
```

```
In [52]: np.arange(-20,10)
```

```
Out[52]: array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8])
```

```
In [54]: np.arange(-16,10)
```

```
Out[54]: array([-16, -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [56]: ar=np.arange(-30,20)
ar
```

```
Out[56]: array([-30, -29, -28, -27, -26, -25, -24, -23, -22, -21, -20, -19, -18,
-17, -16, -15, -14, -13, -12, -11, -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])
```

```
In [58]: np.arange(10,10)
```

```
Out[58]: array([], dtype=int32)
```

```
In [60]: np.arange()
```

```
-----  

TypeError                                                 Traceback (most recent call last)  

Cell In[60], line 1  

----> 1 np.arange()  

  

TypeError: arange() requires stop to be specified.
```

```
In [62]: np.arange(0,10,5)
```

```
Out[62]: array([0, 5])
```

```
In [64]: np.arange(0,10,3)
```

```
Out[64]: array([0, 3, 6, 9])
```

zeros function

```
In [67]: np.zeros(10)
```

```
Out[67]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
In [69]: np.zeros(3)
```

```
Out[69]: array([0., 0., 0.])
```

```
In [71]: np.zeros(3,dtype=int)
```

```
Out[71]: array([0, 0, 0])
```

```
In [73]: np.zeros()
```

```
-----  

TypeError                                                 Traceback (most recent call last)  

Cell In[73], line 1  

----> 1 np.zeros()  

  

TypeError: zeros() missing required argument 'shape' (pos 0)
```

```
In [75]: np.zeros(0)
```

```
Out[75]: array([], dtype=float64)
```

```
In [77]: np.zeros((2,2),dtype=int)
```

```
Out[77]: array([[0, 0],
 [0, 0]])
```

```
In [79]: zero=np.zeros([2,2])
print(zero)
print('###')
print(type(zero))
```

```
[[0. 0.]
 [0. 0.]]
###
<class 'numpy.ndarray'>
```

```
In [81]: np.zeros((2,10))
```

```
Out[81]: array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
 [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

```
In [83]: np.zeros((2,10),dtype=int)
```

```
Out[83]: array([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]])
```

```
In [85]: np.zeros((10,10),dtype=int)
```

```
Out[85]: 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],
 [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]])
```

ones function

```
In [88]: np.ones(0)
```

```
Out[88]: array([], dtype=float64)
```

```
In [90]: np.ones(3)
```

```
Out[90]: array([1., 1., 1.])
```

```
In [92]: np.ones((3,2))
```

```
Out[92]: array([[1., 1.],
   [1., 1.],
   [1., 1.]])
```

```
In [94]: np.ones((3,3),dtype=int)
```

```
Out[94]: array([[1, 1, 1],
   [1, 1, 1],
   [1, 1, 1]])
```

```
In [96]: np.ones((2,10))
```

```
Out[96]: array([[1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
   [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]])
```

```
In [98]: np.ones((2,10),dtype=int)
```

```
Out[98]: array([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
   [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]])
```

```
In [100... np.ones((10,10),dtype=int)
```

```
Out[100... 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, 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]])
```

```
In [102... np.ones((2,2),dtype=str)
```

```
Out[102... array([['1', '1'],
   ['1', '1']], dtype='<U1')
```

```
In [104... np.ones((5,5),dtype=bool)
```

```
Out[104... array([[ True,  True,  True,  True,  True],
   [ True,  True,  True,  True,  True]])
```

```
In [106... np.zeros((5,5),dtype=bool)
```

```
Out[106... array([[False, False, False, False, False],
   [False, False, False, False, False]])
```

RANDOM MODULE

random function

```
In [109... np.random.random()
```

```
Out[109... 0.054723204410006
```

```
In [111... np.random.random(1)
```

```
Out[111... array([0.50526873])
```

```
In [113... np.random.random(10)
```

```
Out[113... array([0.26450105, 0.05046421, 0.99249518, 0.80032766, 0.88400116, 0.86374163, 0.89193484, 0.87696372, 0.45888466, 0.72869939])
```

```
In [115... np.random.random(1,5)
```

```
-----  
TypeError Traceback (most recent call last)  
Cell In[115], line 1  
----> 1 np.random.random(1,5)  
  
File numpy\random\mtrand.pyx:439, in numpy.random.mtrand.RandomState.random()  
  
TypeError: random() takes at most 1 positional argument (2 given)
```

```
In [117... np.random.random(3,5)
```

```
-----  
TypeError Traceback (most recent call last)  
Cell In[117], line 1  
----> 1 np.random.random(3,5)  
  
File numpy\random\mtrand.pyx:439, in numpy.random.mtrand.RandomState.random()  
  
TypeError: random() takes at most 1 positional argument (2 given)
```

rand function

```
In [120... np.random.rand()
```

```
Out[120... 0.4270819822157721
```

```
In [122... np.random.rand(3)
```

```
Out[122... array([0.55460563, 0.17610771, 0.74334662])
```

```
In [124... np.random.rand(3)
```

```
Out[124... array([0.67794339, 0.36404049, 0.83775559])

In [126... np.random.rand(1)

Out[126... array([0.41490745])

In [128... np.random.rand(10)

Out[128... array([0.83656144, 0.08249211, 0.41566706, 0.81813899, 0.93061759,
0.42551557, 0.13552496, 0.08515244, 0.58729237, 0.10123999])

In [130... np.random.rand(3,5)

Out[130... array([[0.33191193, 0.95447136, 0.17552169, 0.4292918 , 0.71364752],
[0.49621212, 0.6196966 , 0.14569192, 0.18667107, 0.93480491],
[0.72200063, 0.77995602, 0.3718195 , 0.9783005 , 0.32037688]])

In [132... np.random.rand(1,10,2)

Out[132... array([[[0.33708565, 0.10067304],
[0.85538754, 0.34322648],
[0.22922823, 0.39385081],
[0.02684948, 0.07961938],
[0.13496391, 0.81868488],
[0.6822128 , 0.83742817],
[0.2518126 , 0.18130769],
[0.22509786, 0.95997423],
[0.08160632, 0.86769367],
[0.76611716, 0.40683447]]])
```

randint function

```
In [135... np.random.randint()

-----
TypeError                                     Traceback (most recent call last)
Cell In[135], line 1
      1 np.random.randint()
File numpy\random\mtrand.pyx:679, in numpy.random.mtrand.RandomState.randint()
      1
TypeError: randint() takes at least 1 positional argument (0 given)

In [137... np.random.random(12)

Out[137... array([0.70402099, 0.96985765, 0.13006924, 0.02868311, 0.05608449,
0.62089704, 0.90317129, 0.76576664, 0.74665505, 0.42929067,
0.49797696, 0.2137795 ])

In [139... np.random.randint(4,5)

Out[139... 4
```

```
In [141... np.random.randint(6,7)
```

```
Out[141... 6
```

```
In [143... np.random.randint(11,12)
```

```
Out[143... 11
```

```
In [145... np.random.randint(1,10)
```

```
Out[145... 2
```

```
In [147... np.random.randint(0,10,4)
```

```
Out[147... array([8, 7, 4, 7])
```

```
In [149... np.random.randint(1,10)
```

```
Out[149... 9
```

```
In [151... np.random.randint(30,20,10)
```

ValueError

Traceback (most recent call last)

Cell In[151], line 1

----> 1 np.random.randint(30,20,10)

File `numpy\random\mtrand.pyx:780`, in `numpy.random.mtrand.RandomState.randint()`

File `numpy\random_bounded_integers.pyx:1425`, in `numpy.random._bounded_integers._rand_int32()`

ValueError: low >= high

```
In [153... np.random.randint(-30,20,10)
```

```
Out[153... array([-12, -16, -18, -13, -1, 2, 3, -8, -8, 7])
```

```
In [155... np.random.randint(10,40,(10,10))
```

```
Out[155... array([[19, 29, 23, 27, 21, 14, 10, 11, 24, 39],  
[29, 33, 29, 35, 29, 35, 32, 16, 37, 37],  
[37, 21, 25, 21, 17, 34, 22, 10, 36, 26],  
[39, 11, 31, 38, 27, 38, 18, 18, 18, 29],  
[37, 10, 37, 32, 31, 25, 19, 39, 17, 26],  
[23, 13, 20, 17, 19, 14, 12, 25, 31, 29],  
[32, 28, 29, 39, 20, 39, 29, 12, 36, 19],  
[34, 32, 35, 16, 26, 25, 34, 16, 34, 33],  
[27, 20, 36, 18, 13, 38, 16, 20, 20, 15],  
[22, 32, 14, 16, 21, 12, 34, 21, 29, 39]])
```

```
In [157... np.random.randint(1,100,(10,10))
```

```
Out[157... array([[36, 81, 30, 39, 36, 42, 95, 35, 72, 97],
   [83, 18, 2, 24, 89, 61, 74, 27, 7, 10],
   [78, 47, 79, 1, 77, 30, 36, 27, 16, 44],
   [45, 83, 7, 93, 54, 1, 64, 65, 28, 45],
   [1, 72, 46, 9, 96, 55, 35, 81, 19, 65],
   [64, 20, 95, 99, 85, 10, 33, 71, 45, 91],
   [1, 48, 91, 70, 52, 10, 96, 36, 91, 34],
   [28, 20, 30, 21, 54, 47, 9, 42, 71, 8],
   [27, 96, 32, 54, 54, 24, 60, 90, 93, 52],
   [10, 91, 21, 52, 46, 73, 26, 94, 37, 31]])
```

```
In [159... np.arange(1,13)
```

```
Out[159... array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

reshape fuction

```
In [162... np.arange(1,13).reshape(3,4) #3*4=12
```

```
Out[162... array([[ 1,  2,  3,  4],
   [ 5,  6,  7,  8],
   [ 9, 10, 11, 12]])
```

```
In [164... np.arange(1,12).reshape(3,4) #3*4=12
```

```
-----
ValueError                                     Traceback (most recent call last)
Cell In[164], line 1
----> 1 np.arange(1,12).reshape(3,4)

ValueError: cannot reshape array of size 11 into shape (3,4)
```

```
In [166... np.arange(1,25).reshape(4,6) #4*6=24
```

```
Out[166... 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 [168... np.arange(1,25).reshape(6,4) #4*6=24
```

```
Out[168... 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]])
```

indexing and slicing in matrix

```
In [171... a=np.random.randint(10,20,(5,4))
```

a

```
Out[171... array([[17, 16, 18, 14],  
                   [16, 11, 17, 14],  
                   [11, 19, 10, 12],  
                   [12, 19, 11, 12],  
                   [13, 17, 14, 14]])
```

```
In [173... type(a)
```

```
Out[173... numpy.ndarray
```

```
In [175... a[:]
```

```
Out[175... array([[17, 16, 18, 14],  
                   [16, 11, 17, 14],  
                   [11, 19, 10, 12],  
                   [12, 19, 11, 12],  
                   [13, 17, 14, 14]])
```

```
In [177... a[0]
```

```
Out[177... array([17, 16, 18, 14])
```

```
In [179... a[1:3]
```

```
Out[179... array([[16, 11, 17, 14],  
                   [11, 19, 10, 12]])
```

```
In [181... a[1,3]
```

```
Out[181... 14
```

```
In [183... a[3,3]
```

```
Out[183... 12
```

```
In [185... a[0,3]
```

```
Out[185... 14
```

```
In [187... a[0, -1]
```

```
Out[187... 14
```

```
In [189... print(a)
```

```
[[17 16 18 14]  
 [16 11 17 14]  
 [11 19 10 12]  
 [12 19 11 12]  
 [13 17 14 14]]
```

```
In [191... a[0:-2]
```

```
Out[191... array([[17, 16, 18, 14],  
                  [16, 11, 17, 14],  
                  [11, 19, 10, 12]])
```

```
In [193... a[-1]
```

```
Out[193... array([13, 17, 14, 14])
```

```
In [195... a[-5]
```

```
Out[195... array([17, 16, 18, 14])
```

```
In [197... a[-4]
```

```
Out[197... array([16, 11, 17, 14])
```

```
In [199... a[0:3]
```

```
Out[199... array([[17, 16, 18, 14],  
                  [16, 11, 17, 14],  
                  [11, 19, 10, 12]])
```

```
In [201... a[0:5]
```

```
Out[201... array([[17, 16, 18, 14],  
                  [16, 11, 17, 14],  
                  [11, 19, 10, 12],  
                  [12, 19, 11, 12],  
                  [13, 17, 14, 14]])
```

```
In [203... a[0:5:2]
```

```
Out[203... array([[17, 16, 18, 14],  
                  [11, 19, 10, 12],  
                  [13, 17, 14, 14]])
```

```
In [205... a[::]
```

```
Out[205... array([[17, 16, 18, 14],  
                  [16, 11, 17, 14],  
                  [11, 19, 10, 12],  
                  [12, 19, 11, 12],  
                  [13, 17, 14, 14]])
```

```
In [207... a[:::-1]
```

```
Out[207... array([[13, 17, 14, 14],  
                  [12, 19, 11, 12],  
                  [11, 19, 10, 12],  
                  [16, 11, 17, 14],  
                  [17, 16, 18, 14]])
```

```
In [209... a[:::-2]
```

```
Out[209... array([[13, 17, 14, 14],
       [11, 19, 10, 12],
       [17, 16, 18, 14]])
```

```
In [211... a1=np.arange(6)
print(a1)
print(a1.shape)
```

```
[0 1 2 3 4 5]
(6,)
```

converting 1d to 2d using the newaxis fuction

```
In [214... a2=a1[np.newaxis,:]
print(a2)
print(a2.shape)
```

```
[[0 1 2 3 4 5]]
(1, 6)
```

```
In [216... x=np.arange(12).reshape(4,3)
x
print(x.shape)
```

```
(4, 3)
```

```
In [218... y=np.arange(20)
print(y)
print(y.shape)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
(20,)
```

```
In [220... z=y[np.newaxis,:]
print(z)
```

```
[[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]]
```

```
In [222... print(z.shape)
```

```
(1, 20)
```

```
In [224... z.ndim
```

```
Out[224... 2
```

```
In [226... a=np.arange(16)
a
```

```
Out[226... array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
```

```
In [228... a.ndim
```

Out[228... 1

```
In [230... a3=np.arange(4)
a4=a3[np.newaxis,:]
print(a4)
print(a4.shape)
print(a4.ndim)
```

```
[[0 1 2 3]]
(1, 4)
2
```

```
In [232... a3=np.arange(4)
a4=a3[:,np.newaxis]
print(a4)
print(a4.shape)
print(a4.ndim)
```

```
[[0]
[1]
[2]
[3]]
(4, 1)
2
```

converting 1d to 2d using expand_dims

```
In [235... a3=np.arange(4)
a5=np.expand_dims(a3,1)
print(a5)
print(a5.shape)
print(a5.ndim)
```

```
[[0]
[1]
[2]
[3]]
(4, 1)
2
```

```
In [237... a3=np.arange(4)
a5=np.expand_dims(a3,0)
print(a5)
print(a5.shape)
print(a5.ndim)
```

```
[[0 1 2 3]]
(1, 4)
2
```

converting 1d to nd using reshape

```
In [240... #converting 1d to 2d
a=np.arange(4).reshape(2,2)
print(a)
print(a.shape)
print(a.ndim)
```

```
[[0 1]
 [2 3]]
(2, 2)
2
```

```
In [242... #converting 1d to 3d
a1=np.arange(8).reshape(2,2,2)
print(a1)
print(a1.shape)
print(a1.ndim)
```

```
[[[0 1]
 [2 3]]

 [[4 5]
 [6 7]]]
(2, 2, 2)
3
```

```
In [244... #converting 1d to nd using reshape
a2=np.arange(16).reshape(2,2,2,2)
print(a2)
print(a2.shape)
print(a2.ndim)
```

```
[[[[ 0  1]
 [ 2  3]]

 [[ 4  5]
 [ 6  7]]]

 [[[ 8  9]
 [10 11]]

 [[12 13]
 [14 15]]]]
(2, 2, 2, 2)
4
```

converting nd to 1d using flatten and ravel

```
In [247... x1=np.arange(16).reshape(2,2,2,2)
print(x1)
print(x1.shape)
print(x1.ndim)
```

```
[[[[ 0  1]
   [ 2  3]]]
```

```
[[ 4  5]
 [ 6  7]]]
```

```
[[[ 8  9]
   [10 11]]]
```

```
[[[12 13]
   [14 15]]]
(2, 2, 2, 2)
4
```

```
In [249... x2=x1.flatten()
print(x2)
print(x2.shape)
print(x2.ndim)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
(16,)
1
```

```
In [251... x3=x1.ravel()
print(x3)
print(x3.ndim)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
1
```

```
In [253... #using reshape we can convert nd to 1d
x4=x1.reshape(-1)
print(x4)
print(x4.ndim)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
1
```

creating a basic array

```
In [256... import numpy as np
a=np.array([1,2,3,4,5,6,7,8,9])
print(a)
print(type(a))
print(a.shape)
print(a.size)
```

```
[1 2 3 4 5 6 7 8 9]
<class 'numpy.ndarray'>
(9,)
9
```

Empty fuction

```
In [259... np.empty(2)
```

```
Out[259... array([2.12199579e-314, 6.36598737e-314])
```

```
In [261... np.empty(10)
```

```
Out[261... array([0.83656144, 0.08249211, 0.41566706, 0.81813899, 0.93061759,
0.42551557, 0.13552496, 0.08515244, 0.58729237, 0.10123999])
```

```
In [263... np.empty(5)
```

```
Out[263... array([ 2.12199579e-314, 6.95293141e-310, 9.15009576e-321,
6.95293141e-310, -3.55262760e-231])
```

```
In [265... np.empty(3)
```

```
Out[265... array([2.12199579e-314, 6.36598737e-314, 1.06099790e-313])
```

```
In [267... np.empty(50)
```

```
Out[267... array([ 0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
0.00000000e+000, 0.00000000e+000,
1.69593623e-152, -1.11084700e-310, 0.00000000e+000,
0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
-1.20209287e-310, 0.00000000e+000, 0.00000000e+000,
0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
0.00000000e+000, 0.00000000e+000])
```

linspace fuction

```
In [270... np.linspace(0,10,5)
```

```
Out[270... array([ 0. , 2.5, 5. , 7.5, 10. ])
```

```
In [272... np.linspace(1,10,5)
```

```
Out[272... array([ 1. , 3.25, 5.5 , 7.75, 10. ])]
```

```
In [274... np.linspace(0,100,10)
```

```
Out[274... array([ 0.          , 11.11111111, 22.22222222, 33.33333333, 44.44444444, 55.55555556, 66.66666667, 77.77777778, 88.88888889, 100.        ])
```

```
In [276... np.linspace(1,100,10)
```

```
Out[276... array([ 1., 12., 23., 34., 45., 56., 67., 78., 89., 100.])
```

```
In [278... np.linspace(0,100,20)
```

```
Out[278... array([ 0.          , 5.26315789, 10.52631579, 15.78947368, 21.05263158, 26.31578947, 31.57894737, 36.84210526, 42.10526316, 47.36842105, 52.63157895, 57.89473684, 63.15789474, 68.42105263, 73.68421053, 78.94736842, 84.21052632, 89.47368421, 94.73684211, 100.        ])
```

```
In [280... np.linspace(1,100,10,dtype=int)
```

```
Out[280... array([ 1, 12, 23, 34, 45, 56, 67, 78, 89, 100])
```

```
In [282... np.linspace(1,100,20,dtype=int)
```

```
Out[282... array([ 1, 6, 11, 16, 21, 27, 32, 37, 42, 47, 53, 58, 63, 68, 73, 79, 84, 89, 94, 100])
```

```
In [284... arr,step=np.linspace(1,100,10,retstep=True)
print(arr)
print(step)
```

```
[ 1. 12. 23. 34. 45. 56. 67. 78. 89. 100.]
11.0
```

logspace fuction

```
In [287... np.logspace(1,3,num=3) #defaltly take base 10-->10**1=10,10**2=100,10**3=100
```

```
Out[287... array([ 10., 100., 1000.])
```

```
In [289... np.logspace(1,3,num=3,base=2) #base 2 so-->2**1=2,2**2=4,2**3=8
```

```
Out[289... array([2., 4., 8.])
```

```
In [291... np.logspace(0,10,num=5,base=5)
```

```
Out[291... array([1.00000000e+00, 5.59016994e+01, 3.12500000e+03, 1.74692811e+05, 9.76562500e+06])
```

```
In [293... np.logspace(1,100,num=5)
```

```
Out[293... array([1.0000000e+001, 5.62341325e+025, 3.16227766e+050, 1.77827941e+075,
1.0000000e+100])

In [295... np.logspace(1,100,num=5,base=5)

Out[295... array([5.0000000e+00, 9.96500708e+17, 1.98602732e+35, 3.95815527e+52,
7.88860905e+69])

In [297... np.logspace(1,100)

Out[297... array([1.0000000e+001, 1.04811313e+003, 1.09854114e+005, 1.15139540e+007,
1.20679264e+009, 1.26485522e+011, 1.32571137e+013, 1.38949549e+015,
1.45634848e+017, 1.52641797e+019, 1.59985872e+021, 1.67683294e+023,
1.75751062e+025, 1.84206997e+027, 1.93069773e+029, 2.02358965e+031,
2.12095089e+033, 2.22299648e+035, 2.32995181e+037, 2.44205309e+039,
2.55954792e+041, 2.68269580e+043, 2.81176870e+045, 2.94705170e+047,
3.08884360e+049, 3.23745754e+051, 3.39322177e+053, 3.55648031e+055,
3.72759372e+057, 3.90693994e+059, 4.09491506e+061, 4.29193426e+063,
4.49843267e+065, 4.71486636e+067, 4.94171336e+069, 5.17947468e+071,
5.42867544e+073, 5.68986603e+075, 5.96362332e+077, 6.25055193e+079,
6.55128557e+081, 6.86648845e+083, 7.19685673e+085, 7.54312006e+087,
7.90604321e+089, 8.28642773e+091, 8.68511374e+093, 9.10298178e+095,
9.54095476e+097, 1.00000000e+100])

In [299... np.logspace(1,100,10)

Out[299... array([1.e+001, 1.e+012, 1.e+023, 1.e+034, 1.e+045, 1.e+056, 1.e+067,
1.e+078, 1.e+089, 1.e+100])

In [301... np.logspace(1,100,10,base=5)

Out[301... array([5.0000000e+00, 2.44140625e+08, 1.19209290e+16, 5.82076609e+23,
2.84217094e+31, 1.38777878e+39, 6.77626358e+46, 3.30872245e+54,
1.61558713e+62, 7.88860905e+69])
```

sorting and concatting arrays

```
In [304... #sorting
arr=np.array([34,12,76,15,2,7,98,45,6,8,23,9,40,50,100,500,763,432])
arr

Out[304... array([ 34,  12,  76,  15,   2,    7,   98,   45,    6,    8,   23,    9,   40,
 50, 100, 500, 763, 432])

In [306... np.sort(arr)

Out[306... array([  2,    6,    7,    8,    9,   12,   15,   23,   34,   40,   45,   50,   76,
 98, 100, 432, 500, 763])

In [308... np.sort(arr)[::-1]

Out[308... array([763, 500, 432, 100,  98,  76,  50,  45,  40,  34,  23,  15,  12,
 9,   8,   7,   6,   2])
```

```
In [310... arr1 = np.array([2, 1, 5, 3, 7, 4, 6, 8])
np.sort(arr1)
```

```
Out[310... array([1, 2, 3, 4, 5, 6, 7, 8])
```

```
In [312... np.sort(arr1)[::-1]
```

```
Out[312... array([8, 7, 6, 5, 4, 3, 2, 1])
```

```
In [314... #concatinating
a1=np.array([1,2,3,4])
b1=np.array([5,6,7,8])
np.concatenate((a1,b1))
```

```
Out[314... array([1, 2, 3, 4, 5, 6, 7, 8])
```

```
In [316... a2=np.arange(1,13).reshape(3,4)
print(a2)
b2=np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12]])
print(b2)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

```
In [318... np.concatenate((a2,b2))
```

```
Out[318... array([[ 1,  2,  3,  4],
 [ 5,  6,  7,  8],
 [ 9, 10, 11, 12],
 [ 1,  2,  3,  4],
 [ 5,  6,  7,  8],
 [ 9, 10, 11, 12]])
```

```
In [320... a3=np.array([[1,2,3],[4,5,6]])
b3=np.array([[9,4,5],[7,5,3]])
np.concatenate((a3,b3))
```

```
Out[320... array([[1, 2, 3],
 [4, 5, 6],
 [9, 4, 5],
 [7, 5, 3]])
```

```
In [322... a4=np.array([[1,2],[3,4]])
b4=np.array([[5,6]])
np.concatenate((a4,b4) ,axis=0)
```

```
Out[322... array([[1, 2],
 [3, 4],
 [5, 6]])
```

dimentions, size, shape

```
In [325...]: x=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]]])  
print(x)
```

```
[[[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 [327...]: #dimentions  
x.ndim
```

```
Out[327...]: 3
```

```
In [329...]: #size  
x.size
```

```
Out[329...]: 24
```

```
In [331...]: x.shape
```

```
Out[331...]: (3, 2, 4)
```

```
In [ ]:
```