

Numpy Crash Course

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

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

```
Out[4]: '2.3.5'
```

```
In [5]: import sys  
sys.version
```

```
Out[5]: '3.13.9 | packaged by Anaconda, Inc. | (main, Oct 21 2025, 19:09:58) [MSC v.1929 6  
4 bit (AMD64)]'
```

Creating Arrays

```
In [7]: mylist=[1,2,3,4,5]  
print(mylist)
```

```
[1, 2, 3, 4, 5]
```

```
In [8]: type(mylist)
```

```
Out[8]: list
```

```
In [9]: arr=np.array(mylist)  
arr
```

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

```
In [10]: print(type(arr))  
print(type(mylist))
```

```
<class 'numpy.ndarray'>  
<class 'list'>
```

```
In [11]: arr
```

```
Out[11]: array([1, 2, 3, 4, 5])
```

```
In [12]: np. # we learn important function
```

```
Cell In[12], line 1
```

```
np.  
^
```

```
SyntaxError: invalid syntax
```

arange()

```
In [13]: np.arange(10)
```

```
Out[13]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [14]: np.arange(3.0)
```

```
Out[14]: array([0., 1., 2.])
```

```
In [15]: print(np.arange(10))
print(np.arange(20))
```

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

```
In [16]: np.arange(0,5)
```

```
Out[16]: array([0, 1, 2, 3, 4])
```

```
In [17]: np.arange(10,20)
```

```
Out[17]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
In [18]: np.arange(20,10) # 1st arg < 2nd arg
```

```
Out[18]: array([], dtype=int64)
```

```
In [19]: np.arange(-20,10)
```

```
Out[19]: 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,  9])
```

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

```
Out[20]: 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,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
 19])
```

```
In [21]: np.arange(-15,20)
```

```
Out[21]: array([-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 [22]: np.arange(-30,20)
```

```
Out[22]: 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 [23]: np.arange(10,10)
```

```
Out[23]: array([], dtype=int64)
```

```
In [24]: np.arange(10,20,5)
```

```
Out[24]: array([10, 15])
```

```
In [25]: np.arange(10,50,5) #10 starting 50 end count every 5 steps
```

```
Out[25]: array([10, 15, 20, 25, 30, 35, 40, 45])
```

```
In [26]: np.arange(0,10,2)
```

```
Out[26]: array([0, 2, 4, 6, 8])
```

```
In [27]: np.arange(-10,10,5)
```

```
Out[27]: array([-10, -5,  0,  5])
```

```
In [29]: np.arange(-50,10,5)
```

```
Out[29]: array([-50, -45, -40, -35, -30, -25, -20, -15, -10, -5,  0,  5])
```

```
In [30]: np.arange(10,30,5,8)
```

```
-----  
TypeError                                         Traceback (most recent call last)  
Cell In[30], line 1  
----> 1 np.arange(10,30,5,8)
```

```
TypeError: Cannot interpret '8' as a data type
```

zeros()

```
In [31]: np.zeros(5) # parameter tuning
```

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

```
In [32]: np.zeros(10,dtype=int) #hypermater tuning
```

```
Out[32]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
In [34]: np.zeros((2,2),dtype=int) #two-dimosomal array
```

```
Out[34]: array([[0, 0],  
                 [0, 0]])
```

```
In [36]: np.zeros((2,3))
```

```
Out[36]: array([[0., 0., 0.],  
                 [0., 0., 0.]])
```

```
In [37]: np.zeros((2,3),dtype=int)
```

```
Out[37]: array([[0, 0, 0],  
                 [0, 0, 0]])
```

```
In [38]: np.zeros((-10,5))
```

```
-----  
ValueError  
Cell In[38], line 1  
----> 1 np.zeros((-10,5))
```

Traceback (most recent call last)

```
ValueError: negative dimensions are not allowed
```

```
In [39]: zero = np.zeros([2,2])  
print(zero)  
print(type(zero))
```

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

```
In [43]: zero= np.zeros([3,3])  
print(zero)  
print('#####')  
print(type(zero))
```

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

```
In [45]: zero=np.zeros([4,10])  
print(zero)  
print('#@/?$%&')  
print(type(zero))
```

```
[[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.]]  
#@/?$%&  
<class 'numpy.ndarray'>
```

```
In [46]: np.zeros((10,20))
```

```
In [47]: np.zeros((5,10))
```

```
Out[47]: 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.]])
```

```
In [48]: n=(6,7)
          n1=(6,8)
          print(np.zeros(n1)) # parameter tuning
```

```
In [49]: print(np.zeros(n, dtype=int)) ## hypermeter tunning
```

```

[[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 [51]: t=(5,10)
         t1=(6,10)
         print(np.zeros(t,dtype=int))
```

```
[[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 [52]: print(np.zeros(t1,dtype=int))
```

```
[[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]]
```

```
In [53]: n
```

```
Out[53]: (6, 7)
```

```
In [54]: n1
```

```
Out[54]: (6, 8)
```

```
In [55]: t
```

```
Out[55]: (5, 10)
```

```
In [56]: t1
```

```
Out[56]: (6, 10)
```

```
In [57]: print(np.zeros(t))
```

```
[[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 [58]: print(np.zeros(t1))
```

```
[[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()

```
In [59]: np.ones()
```

```
-----  
TypeError                                 Traceback (most recent call last)  
Cell In[59], line 1  
----> 1 np.ones()  
  
TypeError: ones() missing 1 required positional argument: 'shape'
```

```
In [60]: np.ones(10)
```

```
Out[60]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

```
In [61]: np.ones(10,dtype=int)
```

```
Out[61]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

```
In [62]: np.ones(n)
```

```
Out[62]: 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.]])
```

```
In [63]: print(np.ones(n))  
print(np.ones(n1))  
print(np.ones(t))  
print(np.ones(t1))
```

```
[[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. 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 [65]: np.ones((5,10))
```

```
Out[65]: 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.]])
```

```
In [66]: np.ones((5,10),dtype=int) #bydefault 5-rows 10-columns
```

```
Out[66]: 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]])
```

```
In [67]: np.twos()
```

```
-----
AttributeError                                     Traceback (most recent call last)
Cell In[67], line 1
----> 1 np.twos()

File ~\anaconda3\Lib\site-packages\numpy\__init__.py:795, in __getattr__(attr)
    792     import numpy.char as char
    793     return char.chararray
--> 795 raise AttributeError(f"module {__name__!r} has no attribute {attr!r}")

AttributeError: module 'numpy' has no attribute 'twos'
```

```
In [68]: np.three((2,4))
```

```
-----
AttributeError                                     Traceback (most recent call last)
Cell In[68], line 1
----> 1 np.three((2,4))

File ~\anaconda3\Lib\site-packages\numpy\__init__.py:795, in __getattr__(attr)
    792     import numpy.char as char
    793     return char.chararray
--> 795 raise AttributeError(f"module {__name__!r} has no attribute {attr!r}")

AttributeError: module 'numpy' has no attribute 'three'
```

random()

```
In [70]: np.random.rand((5)) # generate 5-row random values
```

```
Out[70]: array([1.60267255e-01, 9.26634700e-05, 6.29702627e-01, 6.97215100e-01,
   8.18609597e-01])
```

```
In [71]: np.random.rand(10)
```

```
Out[71]: array([0.81222798, 0.75573815, 0.42985781, 0.60307531, 0.65330712,
   0.18019464, 0.27818562, 0.80621058, 0.39735383, 0.32339675])
```

```
In [73]: np.random.rand(5,10)
```

```
Out[73]: array([[0.07058694, 0.37062764, 0.9611033 , 0.92598791, 0.60034967,
   0.48685535, 0.66629631, 0.8848987 , 0.4734606 , 0.26278304],
  [0.2426678 , 0.22487164, 0.21310751, 0.42559243, 0.71355743,
   0.72483267, 0.89697676, 0.67158179, 0.94180266, 0.53936496],
  [0.2212961 , 0.07242226, 0.68401426, 0.43895574, 0.42762672,
   0.94053976, 0.13951976, 0.07729404, 0.13397559, 0.93457421],
  [0.56403825, 0.16954926, 0.22918255, 0.45547222, 0.61768556,
   0.15056598, 0.32690195, 0.70667687, 0.51927073, 0.18850303],
  [0.95355019, 0.95027364, 0.94059152, 0.66441381, 0.27678811,
   0.96875082, 0.81925937, 0.71952768, 0.26812664, 0.58032217]])
```

```
In [74]: np.random.rand(10,30)
```

```
Out[74]: array([[7.38569348e-02, 9.05453667e-01, 1.14830099e-01, 5.30349210e-01,
   2.87985072e-02, 2.05103651e-01, 8.26437963e-01, 1.32962059e-01,
   1.66384780e-02, 6.29850092e-01, 7.35062022e-02, 7.18408247e-01,
   4.95416687e-02, 5.17227790e-01, 1.73225396e-01, 5.53674345e-01,
   4.78652174e-01, 4.44790019e-01, 4.79594480e-01, 1.58980519e-01,
   7.79896140e-01, 7.46245109e-01, 7.73865314e-01, 1.48488059e-01,
   2.32837349e-01, 1.70182040e-01, 1.68119597e-01, 9.36614115e-01,
   8.73569350e-01, 6.50899491e-01],
 [7.24598472e-01, 4.63916837e-01, 8.83954790e-01, 1.87397489e-01,
  5.91184040e-01, 9.16576090e-01, 6.12873356e-01, 2.06624118e-01,
  4.12724047e-01, 9.50115737e-01, 8.86508975e-01, 2.07297692e-01,
  3.73340252e-01, 3.79626685e-01, 4.62243712e-01, 2.70056457e-01,
  3.41512846e-01, 5.07977387e-01, 4.07233574e-01, 1.87902075e-03,
  6.76200045e-01, 7.82010973e-01, 5.02106689e-01, 5.79675490e-01,
  6.08536812e-01, 2.93180725e-01, 7.48126975e-02, 9.76230877e-01,
  3.29620351e-01, 1.67124981e-01],
 [2.81455776e-01, 4.53436241e-02, 5.12126404e-01, 6.76886684e-01,
  3.69981850e-01, 3.01770075e-01, 4.49008304e-01, 3.98127368e-01,
  4.68153794e-01, 1.13791713e-01, 4.75850127e-02, 9.46007327e-01,
  4.35800303e-01, 4.53286623e-01, 8.06626916e-01, 5.61897965e-01,
  7.37647281e-01, 3.31022145e-01, 2.44856351e-01, 7.28718370e-01,
  1.35840744e-01, 5.76916029e-01, 2.13021515e-01, 3.57965935e-01,
  9.52791252e-01, 2.43765267e-01, 6.84801481e-01, 8.83379667e-01,
  6.76672134e-01, 5.10402830e-02],
 [7.90047690e-02, 4.76028650e-03, 7.98832982e-01, 8.03725165e-01,
  5.11128562e-01, 6.50790319e-01, 2.79060110e-01, 9.69287193e-01,
  7.15184246e-01, 4.83651322e-01, 4.82609986e-01, 7.65674480e-01,
  3.43762395e-01, 9.19045513e-01, 3.17756652e-01, 7.80838897e-01,
  9.82729762e-01, 9.90382455e-01, 3.55987441e-01, 9.44960304e-01,
  5.53176179e-02, 1.50626929e-01, 8.79646749e-01, 6.61081141e-01,
  8.22417303e-01, 8.33052539e-01, 9.31502156e-01, 4.92306470e-01,
  1.30244425e-01, 8.43845430e-01],
 [8.62952955e-01, 3.94812615e-01, 1.77979708e-01, 2.43120153e-01,
  4.75230743e-02, 3.28903656e-01, 7.92712062e-01, 4.92496965e-02,
  1.23787034e-01, 3.58908594e-01, 4.44365631e-02, 7.73776663e-01,
  2.01087842e-01, 2.74978710e-01, 4.93053817e-01, 1.23925363e-01,
  2.63426319e-01, 9.06065626e-01, 4.17269051e-01, 4.34440108e-01,
  3.70693719e-01, 1.42642736e-01, 4.94434485e-01, 1.78093393e-02,
  5.04422032e-02, 7.83321956e-01, 8.52151765e-01, 8.86933352e-01,
  8.93576850e-01, 6.74584054e-01],
 [1.88047116e-01, 7.55324612e-01, 1.46440739e-01, 2.62094432e-02,
  7.92739984e-01, 2.28304081e-01, 3.75665351e-01, 2.38443647e-01,
  2.57174652e-01, 6.63454235e-01, 9.77298410e-01, 7.47710174e-01,
  1.90812199e-01, 9.38928171e-02, 2.90980321e-01, 2.46804726e-01,
  6.38456469e-01, 9.01437852e-03, 2.45518336e-01, 2.84978480e-02,
  6.84731502e-01, 9.56850649e-01, 5.59535542e-01, 4.15581894e-01,
  7.93852524e-01, 6.92030688e-01, 4.20526149e-01, 5.39211432e-01,
  4.43500339e-01, 9.94601633e-01],
 [3.78490155e-01, 6.13285964e-01, 7.22217520e-01, 5.64022412e-02,
  6.27762669e-01, 1.73125281e-02, 6.57541950e-01, 3.51393510e-01,
  2.18854175e-01, 9.03444535e-01, 6.06008050e-04, 3.64436800e-03,
  7.22972844e-01, 2.83871891e-01, 5.13560949e-01, 2.31631658e-02,
  9.46928828e-01, 8.30391812e-01, 3.08619256e-01, 8.99397507e-01,
  9.30522333e-01, 1.85844535e-01, 1.81755755e-01, 2.74535845e-01,
  9.69377379e-01, 1.85713057e-01, 6.89362955e-01, 2.57105165e-01,
  6.54043940e-01, 1.90914071e-01],
```

```
[4.52297048e-01, 3.02070378e-01, 3.49285005e-01, 5.99122552e-01,
 3.25337915e-01, 2.02406525e-01, 5.98737278e-01, 1.16977218e-01,
 3.08796088e-01, 6.14891770e-01, 4.56613617e-01, 9.81407759e-02,
 6.03803244e-02, 9.72353523e-02, 2.80263792e-01, 2.59553265e-01,
 8.17583415e-01, 7.37530056e-01, 2.32914402e-01, 6.20506613e-01,
 9.50830157e-01, 3.01065412e-01, 5.47516644e-04, 4.05593694e-01,
 4.22387034e-01, 1.24863825e-01, 8.22241339e-01, 7.64610852e-01,
 6.66548463e-02, 7.83424203e-01],
[7.13053807e-01, 4.80902963e-01, 2.36497242e-02, 5.78710596e-01,
 6.18638121e-01, 2.91711474e-01, 1.65087331e-01, 1.61014619e-01,
 9.06842940e-01, 9.87569487e-01, 2.76843348e-01, 8.76393170e-01,
 8.03079880e-01, 7.21114246e-01, 8.07506247e-01, 1.67802343e-01,
 2.11784891e-01, 3.57633277e-01, 7.21987731e-01, 2.03022081e-01,
 4.20111583e-01, 9.51991254e-01, 4.00754664e-01, 9.94263997e-01,
 8.83863671e-01, 9.28247511e-01, 2.81498175e-01, 6.90645404e-01,
 8.37261515e-01, 8.32294433e-01],
[2.37582616e-02, 4.69284212e-01, 7.38897527e-01, 7.58827236e-01,
 6.47487501e-01, 5.75336751e-01, 6.91512799e-01, 7.76350319e-01,
 6.04457137e-01, 6.42508093e-01, 7.46397581e-01, 4.06047342e-01,
 9.15214577e-01, 9.49543510e-01, 9.71442890e-01, 3.49877509e-01,
 7.24900448e-01, 7.18050292e-02, 8.82132633e-01, 7.18504141e-01,
 4.95226257e-01, 8.37291564e-01, 3.99758957e-01, 3.06213267e-01,
 8.50843946e-01, 7.16567451e-01, 1.02797762e-01, 1.01376238e-02,
 1.19926031e-01, 9.01026252e-01]])
```

In [75]: `np.random.rand(4,5,2) # here generate a 4-array/5-rows/2-columns`

Out[75]: `array([[[0.69760934, 0.4307642],
 [0.17664534, 0.50541794],
 [0.38808523, 0.39696299],
 [0.61248785, 0.86534557],
 [0.35834267, 0.91425627]],

 [[0.08251402, 0.58862704],
 [0.67735345, 0.60214961],
 [0.80407954, 0.50475589],
 [0.4388231 , 0.60988567],
 [0.06724113, 0.13793798]],

 [[0.67876756, 0.74437958],
 [0.5677595 , 0.73708609],
 [0.86270538, 0.40006148],
 [0.54256381, 0.58293264],
 [0.68630877, 0.847059]],

 [[0.37519386, 0.38836291],
 [0.27838953, 0.32474388],
 [0.24463366, 0.4040076],
 [0.74914888, 0.76342369],
 [0.00174244, 0.41216175]]])`

In [77]: `np.random.rand(3,10,4)`

```
Out[77]: array([[[0.51511278, 0.20966931, 0.84757721, 0.73524313],
   [0.5399152 , 0.52470014, 0.39824227, 0.51919653],
   [0.69319042, 0.30778376, 0.09806176, 0.57024609],
   [0.24156546, 0.74447244, 0.00306665, 0.09458193],
   [0.78955525, 0.34964338, 0.03373122, 0.62303213],
   [0.16251398, 0.39326659, 0.72777116, 0.47883744],
   [0.70222071, 0.24857311, 0.52833805, 0.49036018],
   [0.77249687, 0.02011247, 0.3533327 , 0.42109085],
   [0.17659521, 0.85540774, 0.29116566, 0.40887955],
   [0.25848622, 0.58087817, 0.55930334, 0.35419136]],

   [[0.2880815 , 0.56357365, 0.68116658, 0.47614625],
   [0.30059896, 0.23513505, 0.42315134, 0.12585125],
   [0.6840762 , 0.99384749, 0.89075094, 0.74786598],
   [0.34234365, 0.35871056, 0.91886457, 0.78913866],
   [0.3381784 , 0.83199146, 0.85992902, 0.03901346],
   [0.80156303, 0.06636395, 0.22694149, 0.22308273],
   [0.61477546, 0.24976859, 0.84053294, 0.74476235],
   [0.2219502 , 0.1890939 , 0.00278864, 0.00820984],
   [0.41682691, 0.52840305, 0.79359157, 0.44241447],
   [0.45553347, 0.5448616 , 0.14133121, 0.07254657]],

   [[0.24367122, 0.58566778, 0.19014776, 0.53475952],
   [0.67349395, 0.55041648, 0.55653413, 0.30097124],
   [0.14102526, 0.43601358, 0.16079176, 0.68070808],
   [0.25429398, 0.67045278, 0.30972141, 0.9895508 ],
   [0.52669093, 0.74153277, 0.20624942, 0.46173605],
   [0.41435675, 0.56207239, 0.00211903, 0.17753398],
   [0.64677799, 0.45408426, 0.8729059 , 0.69846355],
   [0.7202068 , 0.05393482, 0.41845763, 0.2423294 ],
   [0.86523443, 0.74000607, 0.18129417, 0.01572017],
   [0.44389509, 0.48267605, 0.11887182, 0.87598799]]])
```

random.randint()

```
In [78]: np.random.randint(5)
```

```
Out[78]: 4
```

```
In [82]: np.random.randint(0,5)
```

```
Out[82]: 1
```

```
In [81]: np.random.randint(0,20)
```

```
Out[81]: 15
```

```
In [83]: np.random.randint(10,50)
```

```
Out[83]: 22
```

```
In [84]: np.random.randint(10,20,5) # here we are take 5 integer value between 10-20
```

```
Out[84]: array([13, 10, 11, 15, 14], dtype=int32)
```

```
In [85]: np.random.randint(10,20,10)
```

```
Out[85]: array([14, 15, 14, 17, 11, 10, 11, 11, 10, 12], dtype=int32)
```

```
In [86]: np.random.randint(10,20,20)
```

```
Out[86]: array([10, 18, 16, 18, 13, 12, 12, 15, 14, 19, 18, 14, 17, 10, 16, 12, 10, 17, 12, 14], dtype=int32)
```

```
In [87]: np.random.randint(1,6,4)
```

```
Out[87]: array([5, 1, 3, 3], dtype=int32)
```

```
In [92]: np.random.randint(10,20,(10,10))
```

```
Out[92]: array([[10, 17, 19, 19, 19, 17, 12, 19, 11, 16],  
                [18, 13, 17, 14, 16, 10, 18, 17, 10, 15],  
                [18, 11, 17, 18, 10, 12, 17, 16, 16, 15],  
                [15, 15, 13, 15, 13, 14, 14, 17, 13, 11],  
                [15, 11, 17, 15, 19, 12, 12, 18, 14, 15],  
                [18, 18, 10, 11, 13, 14, 19, 15, 15, 13],  
                [10, 19, 12, 13, 13, 11, 12, 18, 13, 12],  
                [14, 12, 13, 13, 19, 18, 15, 16, 18, 16],  
                [17, 14, 16, 12, 10, 18, 19, 12, 19, 13],  
                [18, 16, 16, 10, 14, 13, 10, 19, 15, 15]], dtype=int32)
```

```
In [93]: print(np.random.randint(10,20,(10,10)))
```

```
[[17 12 14 15 19 13 17 15 10 12]  
 [11 12 14 17 19 14 17 19 16 12]  
 [18 15 16 10 16 19 18 14 12 17]  
 [17 10 19 13 19 10 14 13 17 13]  
 [10 12 15 15 11 10 18 14 15 16]  
 [12 11 18 14 16 14 10 16 15 15]  
 [10 11 14 16 13 17 13 14 12 13]  
 [10 19 12 19 18 13 14 11 12 19]  
 [19 10 12 18 12 10 12 17 18 10]  
 [19 16 17 13 12 15 12 17 12 10]]
```

```
In [95]: np.random.randint(1,5,(10,10))
```

```
Out[95]: array([[4, 1, 2, 3, 1, 3, 1, 1, 2, 1],  
                 [2, 2, 4, 4, 3, 4, 3, 4, 4, 4],  
                 [3, 2, 3, 2, 1, 2, 4, 4, 4, 1],  
                 [3, 3, 4, 4, 3, 2, 4, 1, 4, 1],  
                 [1, 3, 2, 4, 2, 1, 4, 3, 1, 4],  
                 [1, 1, 3, 2, 4, 4, 2, 2, 4, 1],  
                 [1, 4, 1, 2, 3, 1, 4, 1, 2, 2],  
                 [3, 3, 2, 3, 2, 2, 1, 4, 2, 2],  
                 [3, 2, 3, 3, 2, 2, 4, 2, 1, 2],  
                 [4, 1, 2, 1, 2, 3, 1, 4, 2, 4]], dtype=int32)
```

```
In [96]: np.random.randint(1,100,(10,10))
```

```
Out[96]: array([[ 7, 36, 26, 92, 23, 81, 8, 40, 52, 72],
   [52, 32, 8, 30, 92, 40, 29, 81, 13, 31],
   [74, 78, 52, 59, 94, 23, 9, 22, 54, 63],
   [36, 6, 68, 40, 37, 56, 51, 93, 84, 87],
   [96, 97, 9, 12, 77, 55, 34, 98, 56, 4],
   [93, 71, 53, 55, 87, 61, 61, 53, 23, 83],
   [55, 65, 95, 3, 17, 87, 74, 99, 12, 3],
   [26, 68, 43, 79, 18, 95, 7, 90, 41, 77],
   [97, 94, 35, 39, 54, 93, 96, 48, 90, 55],
   [68, 79, 17, 74, 88, 2, 39, 81, 91, 9]], dtype=int32)
```

```
In [97]: np.random.randint(20,50,(5,5))
```

```
Out[97]: array([[23, 38, 47, 37, 37],
   [44, 40, 38, 35, 30],
   [46, 21, 32, 23, 47],
   [46, 48, 41, 31, 45],
   [33, 46, 37, 48, 35]], dtype=int32)
```

```
In [99]: print(np.random.randint(20,50,(5,5)))
```

```
[[38 40 37 38 46]
 [33 31 39 23 38]
 [36 20 36 41 21]
 [41 46 46 41 34]
 [25 43 39 29 47]]
```

```
In [100... np.arange(1,13).reshape(3,4)
```

```
Out[100... array([[ 1,  2,  3,  4],
   [ 5,  6,  7,  8],
   [ 9, 10, 11, 12]])
```

```
In [104... np.arange(1,10).reshape(3,3)
```

```
Out[104... array([[1, 2, 3],
   [4, 5, 6],
   [7, 8, 9]])
```

```
In [106... np.arange(1,15).reshape(3,4)
```

```
-----
ValueError                                                 Traceback (most recent call last)
Cell In[106], line 1
----> 1 np.arange(1,15).reshape(3,4)
```

```
ValueError: cannot reshape array of size 14 into shape (3,4)
```

```
In [107... np.arange(10,50).reshape(3,4)
```

```
-----
ValueError                                                 Traceback (most recent call last)
Cell In[107], line 1
----> 1 np.arange(10,50).reshape(3,4)
```

```
ValueError: cannot reshape array of size 40 into shape (3,4)
```

```
In [109... np.arange(1,13).reshape(12, 1)
```

```
Out[109... array([[ 1],  
[ 2],  
[ 3],  
[ 4],  
[ 5],  
[ 6],  
[ 7],  
[ 8],  
[ 9],  
[10],  
[11],  
[12]])
```

```
In [113... np.arange(1,15).reshape(14, 1)
```

```
Out[113... array([[ 1],  
[ 2],  
[ 3],  
[ 4],  
[ 5],  
[ 6],  
[ 7],  
[ 8],  
[ 9],  
[10],  
[11],  
[12],  
[13],  
[14]])
```

```
In [117... np.arange(1,20).reshape(19,1)
```

```
Out[117... array([[ 1],  
[ 2],  
[ 3],  
[ 4],  
[ 5],  
[ 6],  
[ 7],  
[ 8],  
[ 9],  
[10],  
[11],  
[12],  
[13],  
[14],  
[15],  
[16],  
[17],  
[18],  
[19]])
```

```
In [118... a=np.random.randint(10,20,(5,5))
```

a

```
Out[118... array([[14, 19, 10, 17, 14],  
                   [12, 15, 13, 17, 17],  
                   [17, 19, 10, 15, 11],  
                   [10, 16, 14, 16, 17],  
                   [18, 11, 10, 15, 13]], dtype=int32)
```

In [119... type(a)

Out[119... numpy.ndarray

In [120... a[:]

```
Out[120... array([[14, 19, 10, 17, 14],  
                   [12, 15, 13, 17, 17],  
                   [17, 19, 10, 15, 11],  
                   [10, 16, 14, 16, 17],  
                   [18, 11, 10, 15, 13]], dtype=int32)
```

In [121... a

```
Out[121... array([[14, 19, 10, 17, 14],  
                   [12, 15, 13, 17, 17],  
                   [17, 19, 10, 15, 11],  
                   [10, 16, 14, 16, 17],  
                   [18, 11, 10, 15, 13]], dtype=int32)
```

In [122... a[1:3]

```
Out[122... array([[12, 15, 13, 17, 17],  
                   [17, 19, 10, 15, 11]], dtype=int32)
```

```
In [128... a=np.random.randint(10,20,(5,5))  
a
```

```
Out[128... array([[19, 15, 19, 10, 11],  
                   [14, 16, 12, 12, 18],  
                   [13, 13, 16, 11, 17],  
                   [10, 12, 16, 11, 19],  
                   [16, 12, 18, 19, 19]], dtype=int32)
```

In [129... a[1,3]

Out[129... np.int32(12)

In [131... a[1,4]

Out[131... np.int32(18)

In [132... a[2,4]

Out[132... np.int32(17)

In [133... a[3:4]

```
Out[133]: array([[10, 12, 16, 11, 19]], dtype=int32)
```

```
In [134]: a[0:-2]
```

```
Out[134]: array([[19, 15, 19, 10, 11],  
                 [14, 16, 12, 12, 18],  
                 [13, 13, 16, 11, 17]], dtype=int32)
```

```
In [135]: a[-2,-3]
```

```
Out[135]: np.int32(16)
```

```
In [137]: a[-5,-3]
```

```
Out[137]: np.int32(19)
```

```
In [138]: a[-4,2]
```

```
Out[138]: np.int32(12)
```

```
In [141]: b=a
```

```
b
```

```
Out[141]: array([[19, 15, 19, 10, 11],  
                 [14, 16, 12, 12, 18],  
                 [13, 13, 16, 11, 17],  
                 [10, 12, 16, 11, 19],  
                 [16, 12, 18, 19, 19]], dtype=int32)
```

```
In [142]: print(a)  
print(b)
```

```
[[19 15 19 10 11]  
 [14 16 12 12 18]  
 [13 13 16 11 17]  
 [10 12 16 11 19]  
 [16 12 18 19 19]]  
[[19 15 19 10 11]  
 [14 16 12 12 18]  
 [13 13 16 11 17]  
 [10 12 16 11 19]  
 [16 12 18 19 19]]
```

```
In [143]: id(a),id(b)
```

```
Out[143]: (1962502613936, 1962502613936)
```

```
In [144]: id(a) == id(b)
```

```
Out[144]: True
```

```
In [145]: arr
```

```
Out[145]: array([1, 2, 3, 4, 5])
```

```
In [146... arr1=np.random.randint(1,100,(10,10))
arr1
```

```
Out[146... array([[ 4, 20, 96, 93, 39, 92, 60, 42, 52, 72],
 [18, 46, 89, 65, 86, 23, 30, 35, 49, 30],
 [64, 70, 58, 57, 16, 48, 99, 12, 20, 40],
 [79, 32, 83, 31, 44, 50, 95, 8, 96, 69],
 [ 2, 28, 14, 67, 5, 94, 70, 3, 93, 85],
 [11, 37, 74, 69, 82, 90, 25, 66, 99, 70],
 [ 7, 76, 46, 96, 53, 45, 65, 82, 14, 87],
 [66, 54, 94, 54, 31, 32, 36, 9, 30, 90],
 [65, 89, 15, 25, 19, 16, 59, 63, 85, 29],
 [ 1, 98, 52, 55, 25, 90, 97, 36, 24, 66]], dtype=int32)
```

```
In [147... print(arr1)
```

```
[[ 4 20 96 93 39 92 60 42 52 72]
 [18 46 89 65 86 23 30 35 49 30]
 [64 70 58 57 16 48 99 12 20 40]
 [79 32 83 31 44 50 95 8 96 69]
 [ 2 28 14 67 5 94 70 3 93 85]
 [11 37 74 69 82 90 25 66 99 70]
 [ 7 76 46 96 53 45 65 82 14 87]
 [66 54 94 54 31 32 36 9 30 90]
 [65 89 15 25 19 16 59 63 85 29]
 [ 1 98 52 55 25 90 97 36 24 66]]
```

```
In [148... arr1[:]
```

```
Out[148... array([[ 4, 20, 96, 93, 39, 92, 60, 42, 52, 72],
 [18, 46, 89, 65, 86, 23, 30, 35, 49, 30],
 [64, 70, 58, 57, 16, 48, 99, 12, 20, 40],
 [79, 32, 83, 31, 44, 50, 95, 8, 96, 69],
 [ 2, 28, 14, 67, 5, 94, 70, 3, 93, 85],
 [11, 37, 74, 69, 82, 90, 25, 66, 99, 70],
 [ 7, 76, 46, 96, 53, 45, 65, 82, 14, 87],
 [66, 54, 94, 54, 31, 32, 36, 9, 30, 90],
 [65, 89, 15, 25, 19, 16, 59, 63, 85, 29],
 [ 1, 98, 52, 55, 25, 90, 97, 36, 24, 66]], dtype=int32)
```

```
In [150... arr1[::-1]
```

```
Out[150... array([[ 4, 20, 96, 93, 39, 92, 60, 42, 52, 72],
 [18, 46, 89, 65, 86, 23, 30, 35, 49, 30],
 [64, 70, 58, 57, 16, 48, 99, 12, 20, 40],
 [79, 32, 83, 31, 44, 50, 95, 8, 96, 69],
 [ 2, 28, 14, 67, 5, 94, 70, 3, 93, 85],
 [11, 37, 74, 69, 82, 90, 25, 66, 99, 70],
 [ 7, 76, 46, 96, 53, 45, 65, 82, 14, 87],
 [66, 54, 94, 54, 31, 32, 36, 9, 30, 90],
 [65, 89, 15, 25, 19, 16, 59, 63, 85, 29],
 [ 1, 98, 52, 55, 25, 90, 97, 36, 24, 66]], dtype=int32)
```

```
In [151... arr1[::-1]
```

```
Out[151... array([[ 1, 98, 52, 55, 25, 90, 97, 36, 24, 66],
       [65, 89, 15, 25, 19, 16, 59, 63, 85, 29],
       [66, 54, 94, 54, 31, 32, 36, 9, 30, 90],
       [ 7, 76, 46, 96, 53, 45, 65, 82, 14, 87],
       [11, 37, 74, 69, 82, 90, 25, 66, 99, 70],
       [ 2, 28, 14, 67, 5, 94, 70, 3, 93, 85],
       [79, 32, 83, 31, 44, 50, 95, 8, 96, 69],
       [64, 70, 58, 57, 16, 48, 99, 12, 20, 40],
       [18, 46, 89, 65, 86, 23, 30, 35, 49, 30],
       [ 4, 20, 96, 93, 39, 92, 60, 42, 52, 72]], dtype=int32)
```

In [152... arr1[::-2]

```
Out[152... array([[ 1, 98, 52, 55, 25, 90, 97, 36, 24, 66],
       [66, 54, 94, 54, 31, 32, 36, 9, 30, 90],
       [11, 37, 74, 69, 82, 90, 25, 66, 99, 70],
       [79, 32, 83, 31, 44, 50, 95, 8, 96, 69],
       [18, 46, 89, 65, 86, 23, 30, 35, 49, 30]], dtype=int32)
```

In [153... arr1[::-3]

```
Out[153... array([[ 1, 98, 52, 55, 25, 90, 97, 36, 24, 66],
       [ 7, 76, 46, 96, 53, 45, 65, 82, 14, 87],
       [79, 32, 83, 31, 44, 50, 95, 8, 96, 69],
       [ 4, 20, 96, 93, 39, 92, 60, 42, 52, 72]], dtype=int32)
```

In [154... arr1

```
Out[154... array([[ 4, 20, 96, 93, 39, 92, 60, 42, 52, 72],
       [18, 46, 89, 65, 86, 23, 30, 35, 49, 30],
       [64, 70, 58, 57, 16, 48, 99, 12, 20, 40],
       [79, 32, 83, 31, 44, 50, 95, 8, 96, 69],
       [ 2, 28, 14, 67, 5, 94, 70, 3, 93, 85],
       [11, 37, 74, 69, 82, 90, 25, 66, 99, 70],
       [ 7, 76, 46, 96, 53, 45, 65, 82, 14, 87],
       [66, 54, 94, 54, 31, 32, 36, 9, 30, 90],
       [65, 89, 15, 25, 19, 16, 59, 63, 85, 29],
       [ 1, 98, 52, 55, 25, 90, 97, 36, 24, 66]], dtype=int32)
```

In [155... arr1[0:5] # always remember (n-1) formula

```
Out[155... array([[ 4, 20, 96, 93, 39, 92, 60, 42, 52, 72],
       [18, 46, 89, 65, 86, 23, 30, 35, 49, 30],
       [64, 70, 58, 57, 16, 48, 99, 12, 20, 40],
       [79, 32, 83, 31, 44, 50, 95, 8, 96, 69],
       [ 2, 28, 14, 67, 5, 94, 70, 3, 93, 85]], dtype=int32)
```

In [156... arr1[1,3]

```
Out[156... np.int32(65)
```

In [158... arr2=np.random.randint(50,100,(20,20))
arr2

```
Out[158]: array([[54, 78, 80, 60, 70, 77, 56, 75, 98, 64, 71, 57, 81, 69, 82, 94,
   65, 63, 63, 77],
 [63, 70, 89, 60, 94, 72, 89, 93, 94, 64, 67, 90, 93, 73, 52, 76,
  50, 75, 68, 90],
 [94, 79, 71, 87, 76, 66, 84, 89, 83, 78, 77, 71, 65, 52, 88, 75,
  75, 80, 85, 65],
 [59, 53, 58, 95, 69, 76, 89, 86, 92, 80, 74, 95, 78, 71, 94, 77,
  62, 76, 56, 80],
 [68, 79, 71, 63, 92, 93, 54, 56, 65, 82, 90, 86, 96, 51, 99, 74,
  98, 87, 77, 67],
 [52, 61, 97, 60, 92, 81, 59, 63, 54, 84, 94, 95, 50, 77, 56, 52,
  51, 53, 54, 75],
 [83, 60, 95, 70, 85, 59, 95, 53, 98, 52, 84, 51, 52, 76, 76, 64,
  91, 84, 99, 68],
 [76, 67, 86, 95, 78, 97, 91, 57, 70, 97, 58, 61, 93, 94, 91, 58,
  52, 56, 51, 82],
 [93, 74, 68, 67, 98, 89, 52, 81, 93, 96, 95, 80, 87, 80, 98, 94,
  66, 95, 99, 55],
 [98, 79, 95, 96, 84, 67, 70, 77, 98, 74, 89, 70, 77, 56, 71, 85,
  60, 96, 89, 73],
 [64, 87, 81, 67, 76, 69, 67, 91, 91, 76, 57, 78, 93, 65, 59, 66,
  55, 60, 68, 71],
 [78, 96, 98, 56, 92, 95, 95, 75, 85, 90, 89, 65, 93, 90, 56, 51,
  70, 57, 94, 62],
 [84, 57, 92, 88, 89, 74, 72, 65, 96, 95, 51, 70, 93, 87, 79, 69,
  95, 53, 86, 97],
 [70, 96, 95, 74, 96, 51, 80, 54, 61, 54, 98, 79, 52, 85, 86, 65,
  93, 92, 58, 57],
 [98, 83, 95, 93, 84, 67, 65, 96, 58, 61, 56, 76, 75, 85, 52, 89,
  71, 98, 75, 92],
 [62, 70, 75, 58, 50, 94, 80, 66, 55, 94, 77, 95, 86, 93, 98, 59,
  81, 53, 88, 63],
 [96, 54, 80, 87, 54, 58, 70, 63, 66, 78, 93, 67, 63, 53, 72, 61,
  95, 90, 98, 81],
 [74, 64, 80, 53, 74, 62, 80, 99, 90, 92, 91, 73, 76, 78, 64, 73,
  78, 92, 68, 91],
 [79, 80, 89, 56, 55, 57, 65, 90, 82, 50, 80, 60, 77, 64, 73, 91,
  97, 94, 89, 66],
 [91, 60, 97, 65, 62, 54, 61, 54, 62, 88, 60, 70, 58, 76, 51, 61,
  52, 95, 72, 79]], dtype=int32)
```

```
In [160]: print(type(arr))
print(type(arr1))
print(type(arr2))
```

```
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
```

```
In [161]: arr2
```

```
Out[161... array([[54, 78, 80, 60, 70, 77, 56, 75, 98, 64, 71, 57, 81, 69, 82, 94,
   65, 63, 63, 77],
 [63, 70, 89, 60, 94, 72, 89, 93, 94, 64, 67, 90, 93, 73, 52, 76,
  50, 75, 68, 90],
 [94, 79, 71, 87, 76, 66, 84, 89, 83, 78, 77, 71, 65, 52, 88, 75,
  75, 80, 85, 65],
 [59, 53, 58, 95, 69, 76, 89, 86, 92, 80, 74, 95, 78, 71, 94, 77,
  62, 76, 56, 80],
 [68, 79, 71, 63, 92, 93, 54, 56, 65, 82, 90, 86, 96, 51, 99, 74,
  98, 87, 77, 67],
 [52, 61, 97, 60, 92, 81, 59, 63, 54, 84, 94, 95, 50, 77, 56, 52,
  51, 53, 54, 75],
 [83, 60, 95, 70, 85, 59, 95, 53, 98, 52, 84, 51, 52, 76, 76, 64,
  91, 84, 99, 68],
 [76, 67, 86, 95, 78, 97, 91, 57, 70, 97, 58, 61, 93, 94, 91, 58,
  52, 56, 51, 82],
 [93, 74, 68, 67, 98, 89, 52, 81, 93, 96, 95, 80, 87, 80, 98, 94,
  66, 95, 99, 55],
 [98, 79, 95, 96, 84, 67, 70, 77, 98, 74, 89, 70, 77, 56, 71, 85,
  60, 96, 89, 73],
 [64, 87, 81, 67, 76, 69, 67, 91, 91, 76, 57, 78, 93, 65, 59, 66,
  55, 60, 68, 71],
 [78, 96, 98, 56, 92, 95, 95, 75, 85, 90, 89, 65, 93, 90, 56, 51,
  70, 57, 94, 62],
 [84, 57, 92, 88, 89, 74, 72, 65, 96, 95, 51, 70, 93, 87, 79, 69,
  95, 53, 86, 97],
 [70, 96, 95, 74, 96, 51, 80, 54, 61, 54, 98, 79, 52, 85, 86, 65,
  93, 92, 58, 57],
 [98, 83, 95, 93, 84, 67, 65, 96, 58, 61, 56, 76, 75, 85, 52, 89,
  71, 98, 75, 92],
 [62, 70, 75, 58, 50, 94, 80, 66, 55, 94, 77, 95, 86, 93, 98, 59,
  81, 53, 88, 63],
 [96, 54, 80, 87, 54, 58, 70, 63, 66, 78, 93, 67, 63, 53, 72, 61,
  95, 90, 98, 81],
 [74, 64, 80, 53, 74, 62, 80, 99, 90, 92, 91, 73, 76, 78, 64, 73,
  78, 92, 68, 91],
 [79, 80, 89, 56, 55, 57, 65, 90, 82, 50, 80, 60, 77, 64, 73, 91,
  97, 94, 89, 66],
 [91, 60, 97, 65, 62, 54, 61, 54, 62, 88, 60, 70, 58, 76, 51, 61,
  52, 95, 72, 79]], dtype=int32)
```

```
In [162... arr2[-5,5]
```

```
Out[162... np.int32(94)
```

```
In [163... arr1
```

```
Out[163... array([[ 4, 20, 96, 93, 39, 92, 60, 42, 52, 72],
   [18, 46, 89, 65, 86, 23, 30, 35, 49, 30],
   [64, 70, 58, 57, 16, 48, 99, 12, 20, 40],
   [79, 32, 83, 31, 44, 50, 95, 8, 96, 69],
   [ 2, 28, 14, 67, 5, 94, 70, 3, 93, 85],
   [11, 37, 74, 69, 82, 90, 25, 66, 99, 70],
   [ 7, 76, 46, 96, 53, 45, 65, 82, 14, 87],
   [66, 54, 94, 54, 31, 32, 36, 9, 30, 90],
   [65, 89, 15, 25, 19, 16, 59, 63, 85, 29],
   [ 1, 98, 52, 55, 25, 90, 97, 36, 24, 66]], dtype=int32)
```

```
In [165... arr2
```

```
Out[165... array([[54, 78, 80, 60, 70, 77, 56, 75, 98, 64, 71, 57, 81, 69, 82, 94,
   65, 63, 63, 77],
   [63, 70, 89, 60, 94, 72, 89, 93, 94, 64, 67, 90, 93, 73, 52, 76,
   50, 75, 68, 90],
   [94, 79, 71, 87, 76, 66, 84, 89, 83, 78, 77, 71, 65, 52, 88, 75,
   75, 80, 85, 65],
   [59, 53, 58, 95, 69, 76, 89, 86, 92, 80, 74, 95, 78, 71, 94, 77,
   62, 76, 56, 80],
   [68, 79, 71, 63, 92, 93, 54, 56, 65, 82, 90, 86, 96, 51, 99, 74,
   98, 87, 77, 67],
   [52, 61, 97, 60, 92, 81, 59, 63, 54, 84, 94, 95, 50, 77, 56, 52,
   51, 53, 54, 75],
   [83, 60, 95, 70, 85, 59, 95, 53, 98, 52, 84, 51, 52, 76, 76, 64,
   91, 84, 99, 68],
   [76, 67, 86, 95, 78, 97, 91, 57, 70, 97, 58, 61, 93, 94, 91, 58,
   52, 56, 51, 82],
   [93, 74, 68, 67, 98, 89, 52, 81, 93, 96, 95, 80, 87, 80, 98, 94,
   66, 95, 99, 55],
   [98, 79, 95, 96, 84, 67, 70, 77, 98, 74, 89, 70, 77, 56, 71, 85,
   60, 96, 89, 73],
   [64, 87, 81, 67, 76, 69, 67, 91, 91, 76, 57, 78, 93, 65, 59, 66,
   55, 60, 68, 71],
   [78, 96, 98, 56, 92, 95, 95, 75, 85, 90, 89, 65, 93, 90, 56, 51,
   70, 57, 94, 62],
   [84, 57, 92, 88, 89, 74, 72, 65, 96, 95, 51, 70, 93, 87, 79, 69,
   95, 53, 86, 97],
   [70, 96, 95, 74, 96, 51, 80, 54, 61, 54, 98, 79, 52, 85, 86, 65,
   93, 92, 58, 57],
   [98, 83, 95, 93, 84, 67, 65, 96, 58, 61, 56, 76, 75, 85, 52, 89,
   71, 98, 75, 92],
   [62, 70, 75, 58, 50, 94, 80, 66, 55, 94, 77, 95, 86, 93, 98, 59,
   81, 53, 88, 63],
   [96, 54, 80, 87, 54, 58, 70, 63, 66, 78, 93, 67, 63, 53, 72, 61,
   95, 90, 98, 81],
   [74, 64, 80, 53, 74, 62, 80, 99, 90, 92, 91, 73, 76, 78, 64, 73,
   78, 92, 68, 91],
   [79, 80, 89, 56, 55, 57, 65, 90, 82, 50, 80, 60, 77, 64, 73, 91,
   97, 94, 89, 66],
   [91, 60, 97, 65, 62, 54, 61, 54, 62, 88, 60, 70, 58, 76, 51, 61,
   52, 95, 72, 79]], dtype=int32)
```

```
In [166... arr2[::-1]
```

```
Out[166... array([[54, 78, 80, 60, 70, 77, 56, 75, 98, 64, 71, 57, 81, 69, 82, 94,
   65, 63, 63, 77],
 [63, 70, 89, 60, 94, 72, 89, 93, 94, 64, 67, 90, 93, 73, 52, 76,
  50, 75, 68, 90],
 [94, 79, 71, 87, 76, 66, 84, 89, 83, 78, 77, 71, 65, 52, 88, 75,
  75, 80, 85, 65],
 [59, 53, 58, 95, 69, 76, 89, 86, 92, 80, 74, 95, 78, 71, 94, 77,
  62, 76, 56, 80],
 [68, 79, 71, 63, 92, 93, 54, 56, 65, 82, 90, 86, 96, 51, 99, 74,
  98, 87, 77, 67],
 [52, 61, 97, 60, 92, 81, 59, 63, 54, 84, 94, 95, 50, 77, 56, 52,
  51, 53, 54, 75],
 [83, 60, 95, 70, 85, 59, 95, 53, 98, 52, 84, 51, 52, 76, 76, 64,
  91, 84, 99, 68],
 [76, 67, 86, 95, 78, 97, 91, 57, 70, 97, 58, 61, 93, 94, 91, 58,
  52, 56, 51, 82],
 [93, 74, 68, 67, 98, 89, 52, 81, 93, 96, 95, 80, 87, 80, 98, 94,
  66, 95, 99, 55],
 [98, 79, 95, 96, 84, 67, 70, 77, 98, 74, 89, 70, 77, 56, 71, 85,
  60, 96, 89, 73],
 [64, 87, 81, 67, 76, 69, 67, 91, 91, 76, 57, 78, 93, 65, 59, 66,
  55, 60, 68, 71],
 [78, 96, 98, 56, 92, 95, 95, 75, 85, 90, 89, 65, 93, 90, 56, 51,
  70, 57, 94, 62],
 [84, 57, 92, 88, 89, 74, 72, 65, 96, 95, 51, 70, 93, 87, 79, 69,
  95, 53, 86, 97],
 [70, 96, 95, 74, 96, 51, 80, 54, 61, 54, 98, 79, 52, 85, 86, 65,
  93, 92, 58, 57],
 [98, 83, 95, 93, 84, 67, 65, 96, 58, 61, 56, 76, 75, 85, 52, 89,
  71, 98, 75, 92],
 [62, 70, 75, 58, 50, 94, 80, 66, 55, 94, 77, 95, 86, 93, 98, 59,
  81, 53, 88, 63],
 [96, 54, 80, 87, 54, 58, 70, 63, 66, 78, 93, 67, 63, 53, 72, 61,
  95, 90, 98, 81],
 [74, 64, 80, 53, 74, 62, 80, 99, 90, 92, 91, 73, 76, 78, 64, 73,
  78, 92, 68, 91],
 [79, 80, 89, 56, 55, 57, 65, 90, 82, 50, 80, 60, 77, 64, 73, 91,
  97, 94, 89, 66],
 [91, 60, 97, 65, 62, 54, 61, 54, 62, 88, 60, 70, 58, 76, 51, 61,
  52, 95, 72, 79]], dtype=int32)
```

```
In [168... arr2[::-1]
```

```
Out[168... array([[91, 60, 97, 65, 62, 54, 61, 54, 62, 88, 60, 70, 58, 76, 51, 61,
   52, 95, 72, 79],
 [79, 80, 89, 56, 55, 57, 65, 90, 82, 50, 80, 60, 77, 64, 73, 91,
  97, 94, 89, 66],
 [74, 64, 80, 53, 74, 62, 80, 99, 90, 92, 91, 73, 76, 78, 64, 73,
  78, 92, 68, 91],
 [96, 54, 80, 87, 54, 58, 70, 63, 66, 78, 93, 67, 63, 53, 72, 61,
  95, 90, 98, 81],
 [62, 70, 75, 58, 50, 94, 80, 66, 55, 94, 77, 95, 86, 93, 98, 59,
  81, 53, 88, 63],
 [98, 83, 95, 93, 84, 67, 65, 96, 58, 61, 56, 76, 75, 85, 52, 89,
  71, 98, 75, 92],
 [70, 96, 95, 74, 96, 51, 80, 54, 61, 54, 98, 79, 52, 85, 86, 65,
  93, 92, 58, 57],
 [84, 57, 92, 88, 89, 74, 72, 65, 96, 95, 51, 70, 93, 87, 79, 69,
  95, 53, 86, 97],
 [78, 96, 98, 56, 92, 95, 75, 85, 90, 89, 65, 93, 90, 56, 51,
  70, 57, 94, 62],
 [64, 87, 81, 67, 76, 69, 67, 91, 91, 76, 57, 78, 93, 65, 59, 66,
  55, 60, 68, 71],
 [98, 79, 95, 96, 84, 67, 70, 77, 98, 74, 89, 70, 77, 56, 71, 85,
  60, 96, 89, 73],
 [93, 74, 68, 67, 98, 89, 52, 81, 93, 96, 95, 80, 87, 80, 98, 94,
  66, 95, 99, 55],
 [76, 67, 86, 95, 78, 97, 91, 57, 70, 97, 58, 61, 93, 94, 91, 58,
  52, 56, 51, 82],
 [83, 60, 95, 70, 85, 59, 95, 53, 98, 52, 84, 51, 52, 76, 76, 64,
  91, 84, 99, 68],
 [52, 61, 97, 60, 92, 81, 59, 63, 54, 84, 94, 95, 50, 77, 56, 52,
  51, 53, 54, 75],
 [68, 79, 71, 63, 92, 93, 54, 56, 65, 82, 90, 86, 96, 51, 99, 74,
  98, 87, 77, 67],
 [59, 53, 58, 95, 69, 76, 89, 86, 92, 80, 74, 95, 78, 71, 94, 77,
  62, 76, 56, 80],
 [94, 79, 71, 87, 76, 66, 84, 89, 83, 78, 77, 71, 65, 52, 88, 75,
  75, 80, 85, 65],
 [63, 70, 89, 60, 94, 72, 89, 93, 94, 64, 67, 90, 93, 73, 52, 76,
  50, 75, 68, 90],
 [54, 78, 80, 60, 70, 77, 56, 75, 98, 64, 71, 57, 81, 69, 82, 94,
  65, 63, 63, 77]], dtype=int32)
```

```
In [169... arr2[::-3]
```

```
Out[169...]: array([[91, 60, 97, 65, 62, 54, 61, 54, 62, 88, 60, 70, 58, 76, 51, 61,
   52, 95, 72, 79],
 [96, 54, 80, 87, 54, 58, 70, 63, 66, 78, 93, 67, 63, 53, 72, 61,
  95, 90, 98, 81],
 [70, 96, 95, 74, 96, 51, 80, 54, 61, 54, 98, 79, 52, 85, 86, 65,
  93, 92, 58, 57],
 [64, 87, 81, 67, 76, 69, 67, 91, 91, 76, 57, 78, 93, 65, 59, 66,
  55, 60, 68, 71],
 [76, 67, 86, 95, 78, 97, 91, 57, 70, 97, 58, 61, 93, 94, 91, 58,
  52, 56, 51, 82],
 [68, 79, 71, 63, 92, 93, 54, 56, 65, 82, 90, 86, 96, 51, 99, 74,
  98, 87, 77, 67],
 [63, 70, 89, 60, 94, 72, 89, 93, 94, 64, 67, 90, 93, 73, 52, 76,
  50, 75, 68, 90]], dtype=int32)
```

In [170...]: arr2[::-5]

```
Out[170...]: array([[91, 60, 97, 65, 62, 54, 61, 54, 62, 88, 60, 70, 58, 76, 51, 61,
   52, 95, 72, 79],
 [98, 83, 95, 93, 84, 67, 65, 96, 58, 61, 56, 76, 75, 85, 52, 89,
  71, 98, 75, 92],
 [98, 79, 95, 96, 84, 67, 70, 77, 98, 74, 89, 70, 77, 56, 71, 85,
  60, 96, 89, 73],
 [68, 79, 71, 63, 92, 93, 54, 56, 65, 82, 90, 86, 96, 51, 99, 74,
  98, 87, 77, 67]], dtype=int32)
```

arr.max(),min(),median()

In [171...]: print(arr)

[1 2 3 4 5]

In [172...]: arr.max()

Out[172...]: np.int64(5)

In [173...]: arr.min()

Out[173...]: np.int64(1)

In [174...]: arr

Out[174...]: array([1, 2, 3, 4, 5])

In [175...]: arr3=[0,1,2,3,4,5]
arr3

Out[175...]: [0, 1, 2, 3, 4, 5]

In [178...]: arr3

Out[178...]: [0, 1, 2, 3, 4, 5]

```
In [180... mylist2=[0,1,2,3,4,5]
mylist2
```

```
Out[180... [0, 1, 2, 3, 4, 5]
```

```
In [181... print(mylist2)
```

```
[0, 1, 2, 3, 4, 5]
```

```
In [182... arr3=np.array(mylist2)
arr3
```

```
Out[182... array([0, 1, 2, 3, 4, 5])
```

```
In [183... arr3
```

```
Out[183... array([0, 1, 2, 3, 4, 5])
```

```
In [184... type(arr3)
```

```
Out[184... numpy.ndarray
```

```
In [185... arr3.max()
```

```
Out[185... np.int64(5)
```

```
In [186... arr3.min()
```

```
Out[186... np.int64(0)
```

```
In [187... arr3.mean()
```

```
Out[187... np.float64(2.5)
```

```
In [188... arr3.median()
```

```
-----  
AttributeError  
Cell In[188], line 1  
----> 1 arr3.median()
```

```
Traceback (most recent call last)
```

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

```
In [189... arr3
```

```
Out[189... array([0, 1, 2, 3, 4, 5])
```

reshape()

```
In [190... arr3
```

```
Out[190... array([0, 1, 2, 3, 4, 5])
```

```
In [191... arr3.reshape(3,2)
```

```
Out[191... array([[0, 1],  
[2, 3],  
[4, 5]])
```

```
In [192... arr3.reshape(6,1)
```

```
Out[192... array([[0],  
[1],  
[2],  
[3],  
[4],  
[5]])
```

```
In [193... arr3.reshape(1,6)
```

```
Out[193... array([[0, 1, 2, 3, 4, 5]])
```

```
In [197... arr3.reshape(3,2)
```

```
Out[197... array([[0, 1],  
[2, 3],  
[4, 5]])
```

```
In [198... arr3.reshape(2,3)
```

```
Out[198... array([[0, 1, 2],  
[3, 4, 5]])
```

```
In [199... arr3.reshape(2,3,order='c')
```

```
Out[199... array([[0, 1, 2],  
[3, 4, 5]])
```

```
In [202... arr3.reshape(3,2)
```

```
Out[202... array([[0, 1],  
[2, 3],  
[4, 5]])
```

```
In [203... arr3.reshape(3,3)
```

```
-----  
ValueError  
Cell In[203], line 1  
----> 1 arr3.reshape(3,3)
```

```
Traceback (most recent call last)
```

```
ValueError: cannot reshape array of size 6 into shape (3,3)
```

```
In [204... arr3.reshape(6,1)
```

```
Out[204... array([[0],  
[1],  
[2],  
[3],  
[4],  
[5]])
```

indexing

```
In [206... mat=np.arange(0,100).reshape(10,10)  
mat
```

```
Out[206... 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],  
[50, 51, 52, 53, 54, 55, 56, 57, 58, 59],  
[60, 61, 62, 63, 64, 65, 66, 67, 68, 69],  
[70, 71, 72, 73, 74, 75, 76, 77, 78, 79],  
[80, 81, 82, 83, 84, 85, 86, 87, 88, 89],  
[90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

```
In [207... mat[:]
```

```
Out[207... 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],  
[50, 51, 52, 53, 54, 55, 56, 57, 58, 59],  
[60, 61, 62, 63, 64, 65, 66, 67, 68, 69],  
[70, 71, 72, 73, 74, 75, 76, 77, 78, 79],  
[80, 81, 82, 83, 84, 85, 86, 87, 88, 89],  
[90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

```
In [208... mat[::-1]
```

```
Out[208... 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],  
[50, 51, 52, 53, 54, 55, 56, 57, 58, 59],  
[60, 61, 62, 63, 64, 65, 66, 67, 68, 69],  
[70, 71, 72, 73, 74, 75, 76, 77, 78, 79],  
[80, 81, 82, 83, 84, 85, 86, 87, 88, 89],  
[90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

```
In [209... mat[::-1]
```

```
Out[209]: array([[90, 91, 92, 93, 94, 95, 96, 97, 98, 99],
   [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
   [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
   [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
   [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
   [40, 41, 42, 43, 44, 45, 46, 47, 48, 49],
   [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
   [20, 21, 22, 23, 24, 25, 26, 27, 28, 29],
   [10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
   [ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9]])
```

```
In [210]: mat[::-2]
```

```
Out[210]: array([[90, 91, 92, 93, 94, 95, 96, 97, 98, 99],
   [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
   [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
   [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
   [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]])
```

```
In [211]: mat[::-5]
```

```
Out[211]: array([[90, 91, 92, 93, 94, 95, 96, 97, 98, 99],
   [40, 41, 42, 43, 44, 45, 46, 47, 48, 49]])
```

```
In [212]: mat[0:1]
```

```
Out[212]: array([[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]])
```

```
In [213]: mat
```

```
Out[213]: 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],
   [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
   [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
   [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
   [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
   [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

constructor=_int_will act as a constructor

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

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

```
Out[4]: '2.3.5'
```

```
In [6]: mylist4=[0,1,2,3,4,5]
print(mylist4)
```

```
[0, 1, 2, 3, 4, 5]
```

```
In [7]: arr4=np.array(mylist4)
arr4
```

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

```
In [10]: print( type(mylist4))
print(type(arr4))
```

```
<class 'list'>
<class 'numpy.ndarray'>
```

```
In [15]: mat=np.arange(0,100).reshape(10,10)
mat
```

```
Out[15]: 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],
 [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
 [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
 [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
 [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
 [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

```
In [16]: mat
```

```
Out[16]: 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],
 [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
 [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
 [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
 [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
 [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

```
In [17]: mat[6] # we are print here rows....
```

```
Out[17]: array([60, 61, 62, 63, 64, 65, 66, 67, 68, 69])
```

```
In [18]: mat[:,6] # we wre print here columns... slice means all
```

```
Out[18]: array([ 6, 16, 26, 36, 46, 56, 66, 76, 86, 96])
```

```
In [19]: mat[:, -3] # revrese calculate
```

```
Out[19]: array([ 7, 17, 27, 37, 47, 57, 67, 77, 87, 97])
```

```
In [20]: mat[:, -6]
```

```
Out[20]: array([ 4, 14, 24, 34, 44, 54, 64, 74, 84, 94])
```

```
In [21]: mat[4,7]
```

```
Out[21]: np.int64(47)
```

```
In [22]: mat[:-3]
```

```
Out[22]: 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],
   [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
   [60, 61, 62, 63, 64, 65, 66, 67, 68, 69]])
```

```
In [23]: mat[:3]
```

```
Out[23]: 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]])
```

```
In [24]: mat[:-4:-5]
```

```
Out[24]: array([[90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

```
In [25]: mat
```

```
Out[25]: 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],
   [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
   [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
   [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
   [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
   [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

```
In [26]: mat[:-5:-7]
```

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

```
In [27]: mat[0:11:3] # step slicing start/stop/step here print every 3rd index between 0-11
```

```
Out[27]: array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9],
   [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
   [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
   [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

```
In [28]: mat[1:2,2:4] # it match rows and columns (1:2-rows) (2:4-columns)
```

```
Out[28]: array([[12, 13]])
```

```
In [29]: mat[3:6,3:6]
```

```
Out[30]: array([[33, 34, 35],  
                 [43, 44, 45],  
                 [53, 54, 55]])
```

Masking

we also called filter

```
In [31]: mat
```

```
Out[31]: 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],  
                 [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],  
                 [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],  
                 [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],  
                 [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],  
                 [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

```
In [32]: mat>50
```

```
Out[32]: array([[False, False, False, False, False, False, False, False,  
                  False],  
                 [False, False, False, False, False, False, False, False,  
                  False],  
                 [False, False, False, False, False, False, False, False,  
                  False],  
                 [False, False, False, False, False, False, False, False,  
                  False],  
                 [False, False, False, False, False, False, False, False,  
                  False],  
                 [False, False, False, False, False, False, False, False,  
                  False],  
                 [False, True,  True,  True,  True,  True,  True,  True,  
                  True],  
                 [ True,  True,  True,  True,  True,  True,  True,  True,  
                  True],  
                 [ True,  True,  True,  True,  True,  True,  True,  True,  
                  True],  
                 [ True,  True,  True,  True,  True,  True,  True,  True,  
                  True],  
                 [ True,  True,  True,  True,  True,  True,  True,  True,  
                  True]])
```

```
In [33]: mat>100
```

In [34]: mat[mat>50]

```
Out[34]: array([51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
   68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84,
   85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

```
In [35]: mat[mat>=50] # mat inside mat is called masking
```

```
Out[35]: array([50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66,
   67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83,
   84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

In [36]: mat[mat<50]

```
Out[36]: 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 [37]: mat[mat<=50]
```

```
Out[37]: 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, 50])
```

```
In [38]: mat[mat!=50]
```

```
Out[38]: 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, 51,
    52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68,
    69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
    86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

```
In [39]: mat[mat == 50]
```

```
Out[39]: array([50])
```

```
In [40]: mat[mat!= 100]
```

```
Out[40]: 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, 50,
   51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
   68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84,
   85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

```
In [43]: arr4
```

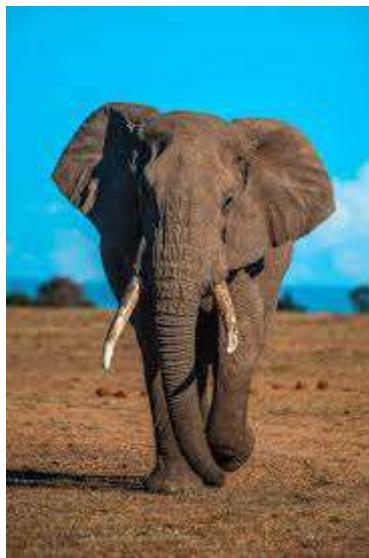
```
Out[43]: array([0, 1, 2, 3, 4, 5])
```

```
In [47]: # first we are called library first
from PIL import Image
```

```
In [52]: image=Image.open(r"C:\Users\abhis\Downloads\Elephant.jpj.jpg")
```

```
In [53]: image
```

```
Out[53]:
```



```
In [54]: type(image)
```

```
Out[54]: PIL.JpegImagePlugin.JpegImageFile
```

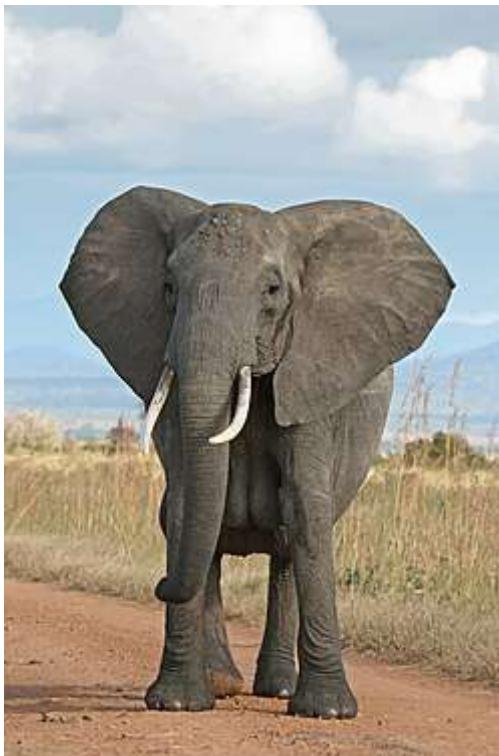
```
In [60]: ele_arr=np.asarray(image)# image convert to array
ele_arr
```

```
Out[60]: array([[[184, 201, 209],  
                 [185, 202, 210],  
                 [186, 203, 211],  
                 ...,  
                 [179, 198, 212],  
                 [178, 197, 211],  
                 [178, 197, 211]],  
  
                [[185, 202, 209],  
                 [186, 203, 211],  
                 [187, 204, 212],  
                 ...,  
                 [179, 198, 212],  
                 [179, 198, 212],  
                 [179, 198, 212]],  
  
                [[193, 208, 215],  
                 [192, 209, 216],  
                 [192, 209, 217],  
                 ...,  
                 [178, 197, 211],  
                 [178, 197, 211],  
                 [178, 197, 211]],  
  
                ...,  
  
                [[206, 172, 144],  
                 [192, 158, 130],  
                 [195, 161, 133],  
                 ...,  
                 [197, 154, 122],  
                 [192, 149, 117],  
                 [195, 152, 120]],  
  
                [[172, 138, 110],  
                 [179, 145, 117],  
                 [187, 153, 125],  
                 ...,  
                 [165, 122, 90],  
                 [197, 154, 122],  
                 [202, 159, 127]],  
  
                [[167, 133, 105],  
                 [167, 133, 105],  
                 [179, 145, 117],  
                 ...,  
                 [190, 147, 115],  
                 [178, 135, 103],  
                 [185, 142, 110]]], shape=(375, 250, 3), dtype=uint8)
```

```
In [61]: image=Image.open(r"C:\Users\abhis\Downloads\African_Bush_Elephant.jpg")
```

```
In [62]: image
```

Out[62]:

In [63]: `type(image)`Out[63]: `PIL.JpegImagePlugin.JpegImageFile`In [65]: `ele_arr=np.asarray(image)`
`ele_arr`

```
Out[65]: array([[[184, 201, 209],  
                 [185, 202, 210],  
                 [186, 203, 211],  
                 ...,  
                 [179, 198, 212],  
                 [178, 197, 211],  
                 [178, 197, 211]],  
  
                [[185, 202, 209],  
                 [186, 203, 211],  
                 [187, 204, 212],  
                 ...,  
                 [179, 198, 212],  
                 [179, 198, 212],  
                 [179, 198, 212]],  
  
                [[193, 208, 215],  
                 [192, 209, 216],  
                 [192, 209, 217],  
                 ...,  
                 [178, 197, 211],  
                 [178, 197, 211],  
                 [178, 197, 211]],  
  
                ...,  
  
                [[206, 172, 144],  
                 [192, 158, 130],  
                 [195, 161, 133],  
                 ...,  
                 [197, 154, 122],  
                 [192, 149, 117],  
                 [195, 152, 120]],  
  
                [[172, 138, 110],  
                 [179, 145, 117],  
                 [187, 153, 125],  
                 ...,  
                 [165, 122, 90],  
                 [197, 154, 122],  
                 [202, 159, 127]],  
  
                [[167, 133, 105],  
                 [167, 133, 105],  
                 [179, 145, 117],  
                 ...,  
                 [190, 147, 115],  
                 [178, 135, 103],  
                 [185, 142, 110]]], shape=(375, 250, 3), dtype=uint8)
```

```
In [66]: ele_arr.shape # Dimension of picture
```

```
Out[66]: (375, 250, 3)
```

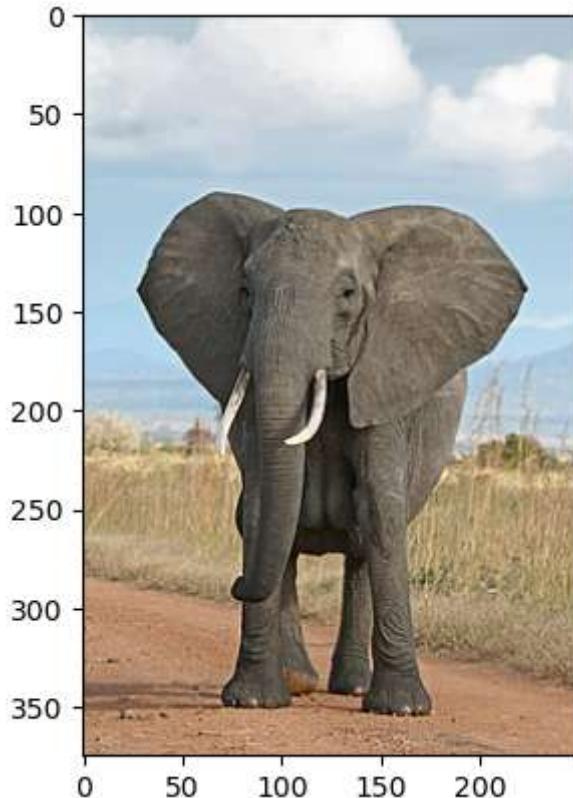
Matplotlib

(matplotlib Library) using for Visualization

```
In [69]: import matplotlib.pyplot as plt # it is matplotlib library
```

Matplotlib is building the font cache; this may take a moment.

```
In [74]: plt.imshow(ele_arr)  
plt.show()# dimision of image
```



```
In [72]: ele_arr.shape
```

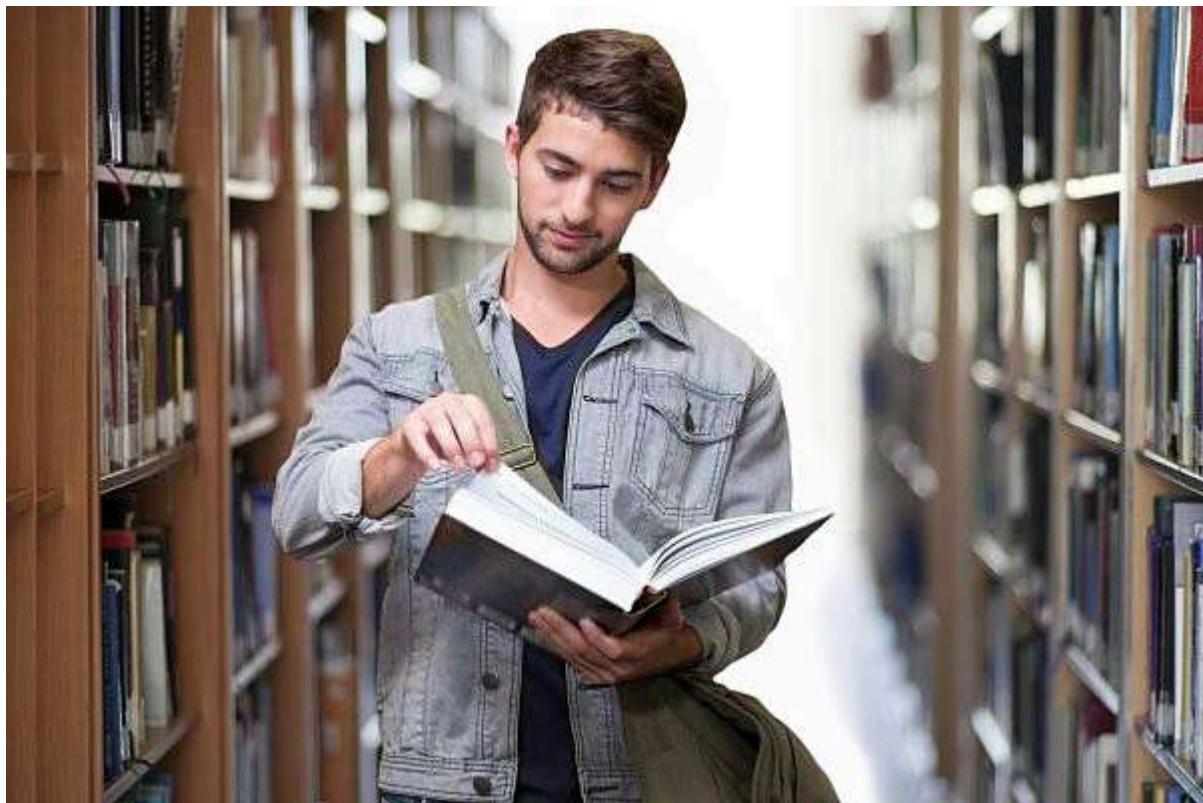
Out[72]: (375, 250, 3)

```
In [77]: from PIL import Image
```

```
In [78]: image=Image.open(r"C:\Users\abhis\Downloads\student-3500990_960_720_opt.jpg")
```

```
In [79]: image
```

Out[79]:

In [80]: `type(image)`Out[80]: `PIL.JpegImagePlugin.JpegImageFile`In [82]: `# image convert to array
stu_arr=np.asarray(image)
stu_arr`

```
Out[82]: array([[[ 91,  55,  31],
   [ 92,  56,  32],
   [ 93,  57,  33],
   ...,
   [136,  35,  53],
   [126,  35,  53],
   [139,  62,  80]],

   [[ 91,  55,  31],
   [ 92,  56,  32],
   [ 93,  57,  33],
   ...,
   [135,  34,  52],
   [123,  32,  50],
   [137,  60,  78]],

   [[ 91,  55,  31],
   [ 92,  56,  32],
   [ 93,  57,  33],
   ...,
   [133,  32,  48],
   [120,  29,  46],
   [139,  61,  77]],

   ...,

   [[ 88,  53,  31],
   [ 91,  56,  34],
   [ 92,  57,  35],
   ...,
   [112, 119, 145],
   [121, 128, 156],
   [120, 129, 158]],

   [[ 87,  52,  30],
   [ 90,  55,  33],
   [ 92,  57,  35],
   ...,
   [134, 144, 169],
   [115, 126, 154],
   [ 97, 108, 138]],

   [[ 86,  51,  29],
   [ 90,  55,  33],
   [ 92,  57,  35],
   ...,
   [137, 149, 171],
   [143, 157, 184],
   [ 86, 101, 132]]], shape=(399, 600, 3), dtype=uint8)
```

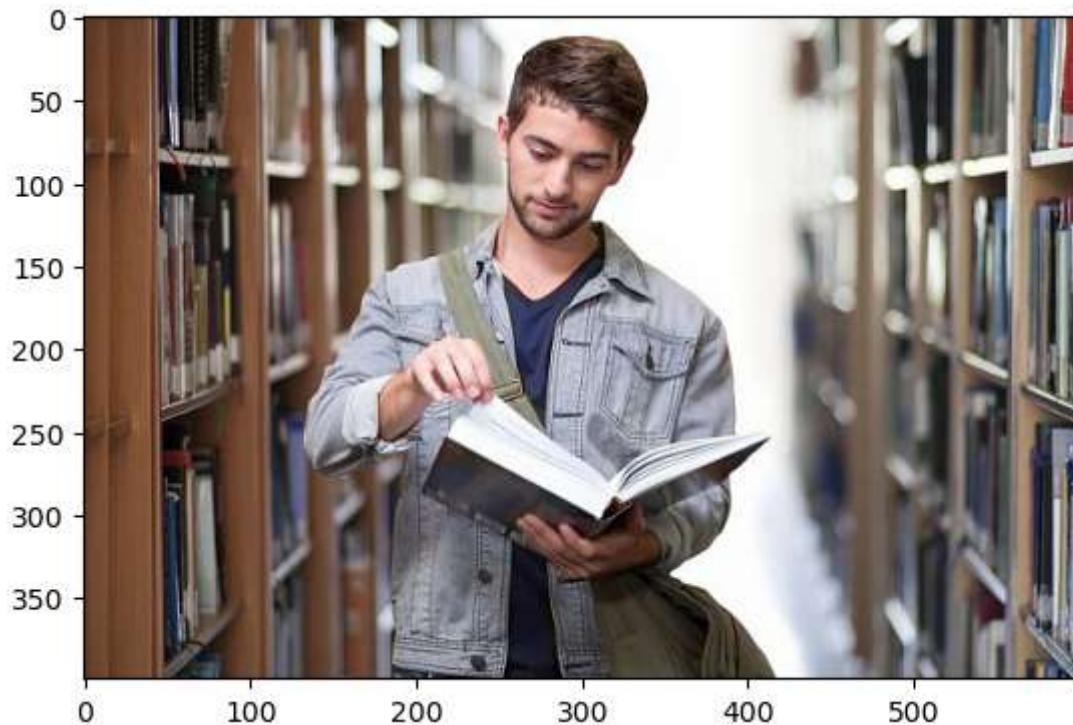
```
In [83]: stu_arr.shape
```

```
Out[83]: (399, 600, 3)
```

```
In [84]: import matplotlib.pyplot as plt
```

```
In [86]: plt.imshow(stu_arr)
```

```
Out[86]: <matplotlib.image.AxesImage at 0x15d730b2d50>
```



```
In [88]: stu_arr.shape
```

```
Out[88]: (399, 600, 3)
```

```
In [89]: image=Image.open(r"C:\Users\abhis\Downloads\0e1b4984c22ff810051677b8c7a29e7d.jpg")
```

```
In [90]: image
```

Out[90]:

In [91]: `type(image)`Out[91]: `PIL.JpegImagePlugin.JpegImageFile`In [92]: `stu_arr=np.asarray(image)`
`stu_arr`

```
Out[92]: array([[[238, 238, 238],  
                 [238, 238, 238],  
                 [238, 238, 238],  
                 ...,  
                 [238, 238, 238],  
                 [238, 238, 238],  
                 [238, 238, 238]],  
  
                [[238, 238, 238],  
                 [238, 238, 238],  
                 [238, 238, 238],  
                 ...,  
                 [238, 238, 238],  
                 [238, 238, 238],  
                 [238, 238, 238]],  
  
                [[238, 238, 238],  
                 [238, 238, 238],  
                 [238, 238, 238],  
                 ...,  
                 [238, 238, 238],  
                 [238, 238, 238],  
                 [238, 238, 238]],  
  
                ...,  
  
                [[239, 239, 239],  
                 [239, 239, 239],  
                 [239, 239, 239],  
                 ...,  
                 [238, 238, 238],  
                 [238, 238, 238],  
                 [238, 238, 238]],  
  
                [[240, 240, 240],  
                 [240, 240, 240],  
                 [240, 240, 240],  
                 ...,  
                 [239, 239, 239],  
                 [239, 239, 239],  
                 [239, 239, 239]],  
  
                [[238, 238, 238],  
                 [238, 238, 238],  
                 [238, 238, 238],  
                 ...,  
                 [238, 238, 238],  
                 [238, 238, 238],  
                 [238, 238, 238]]], shape=(894, 735, 3), dtype=uint8)
```

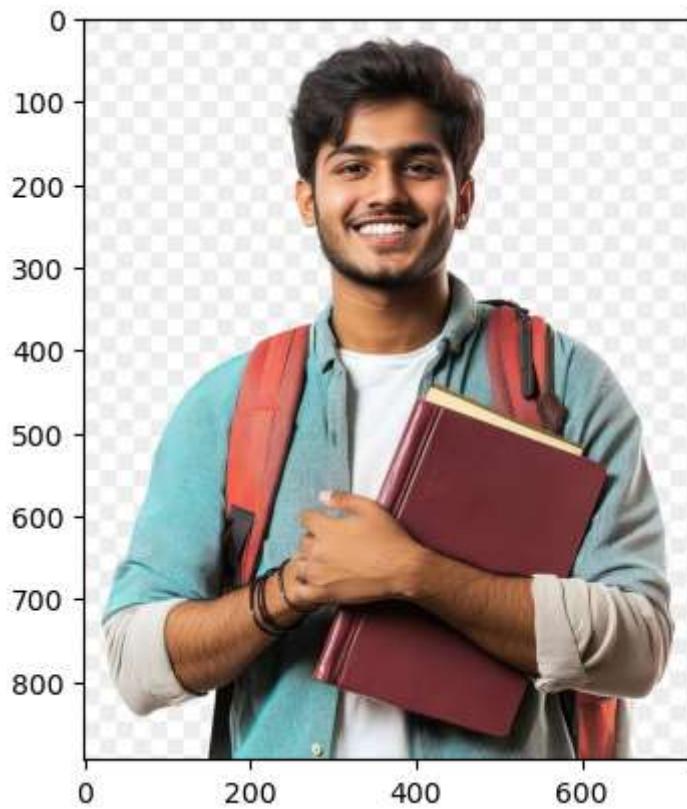
```
In [93]: stu_arr.shape
```

```
Out[93]: (894, 735, 3)
```

```
In [94]: import matplotlib.pyplot as plt
```

In [103...]

```
plt.imshow(image)
plt.show(image)
```



In [104...]

```
stu_arr.shape
```

Out[104...]

```
(894, 735, 3)
```

In [9]:

```
from PIL import Image
```

In [10]:

```
image=Image.open(r"C:\Users\abhis\Downloads\Struggles-behind-the-dream_-Indian-stud
```

In [11]:

```
image
```

Out[11]:

In [12]: `type(image)`Out[12]: `PIL.JpegImagePlugin.JpegImageFile`In [17]: `import numpy as np`In [18]: `np.__version__`Out[18]: `'2.3.5'`In [19]: `ind_stu=np.asarray(image)`
`ind_stu`

```
Out[19]: array([[[251, 253, 255],  
                 [250, 252, 255],  
                 [248, 250, 255],  
                 ...,  
                 [223, 236, 242],  
                 [230, 241, 247],  
                 [233, 244, 250]],  
  
                [[251, 253, 255],  
                 [250, 252, 255],  
                 [248, 250, 255],  
                 ...,  
                 [223, 236, 242],  
                 [229, 240, 246],  
                 [232, 243, 249]],  
  
                [[251, 253, 255],  
                 [250, 252, 255],  
                 [248, 251, 255],  
                 ...,  
                 [222, 235, 241],  
                 [228, 239, 245],  
                 [231, 242, 248]],  
  
                ...,  
  
                [[142, 155, 172],  
                 [140, 153, 169],  
                 [161, 174, 190],  
                 ...,  
                 [124, 150, 173],  
                 [123, 149, 172],  
                 [122, 148, 171]],  
  
                [[139, 152, 169],  
                 [140, 153, 169],  
                 [166, 179, 195],  
                 ...,  
                 [124, 150, 173],  
                 [123, 149, 172],  
                 [122, 148, 171]],  
  
                [[158, 171, 188],  
                 [161, 174, 190],  
                 [187, 200, 216],  
                 ...,  
                 [124, 150, 173],  
                 [123, 149, 172],  
                 [122, 148, 171]]], shape=(900, 1200, 3), dtype=uint8)
```

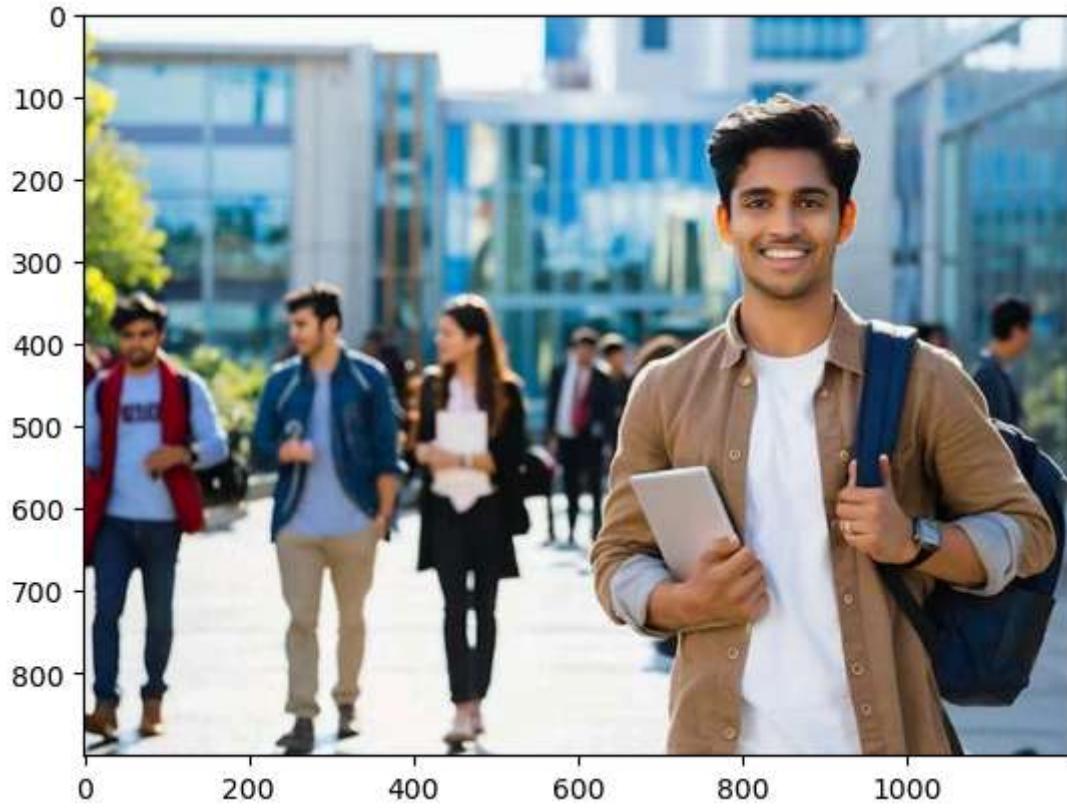
```
In [20]: ind_stu.shape
```

```
Out[20]: (900, 1200, 3)
```

```
In [21]: import matplotlib.pyplot as plt
```

```
In [22]: plt.imshow(image)
```

```
Out[22]: <matplotlib.image.AxesImage at 0x28cf5306120>
```

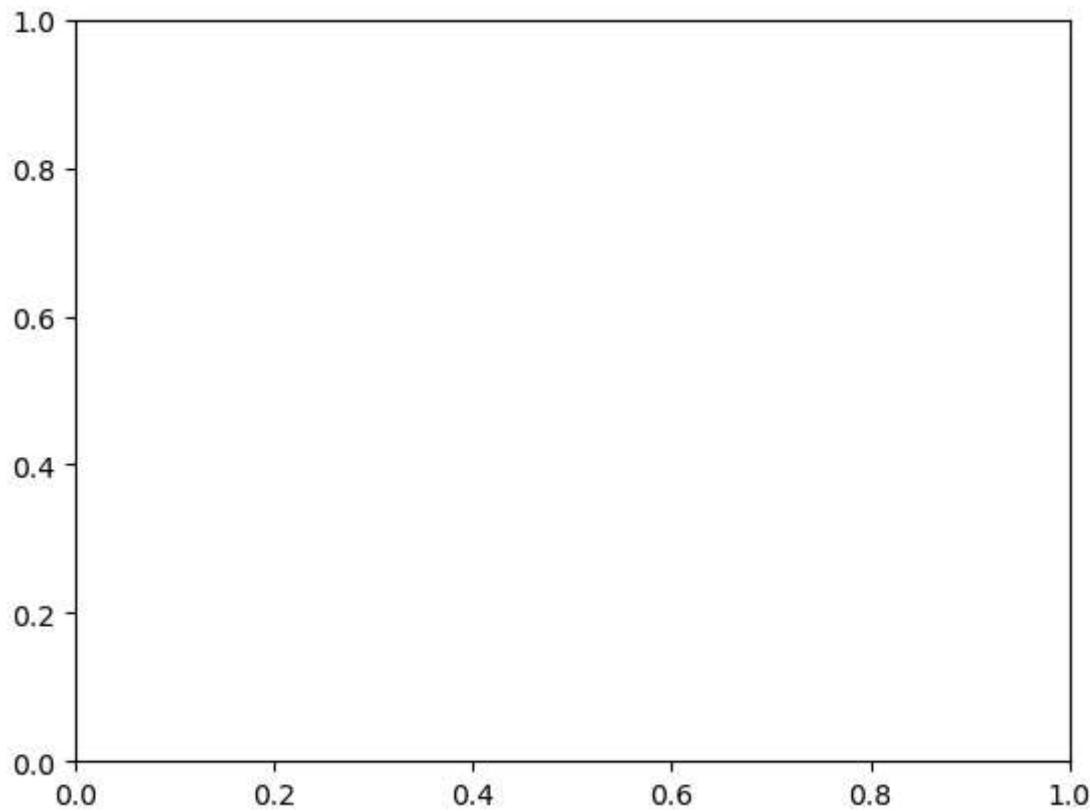


```
In [23]: ind_stu.shape
```

```
Out[23]: (900, 1200, 3)
```

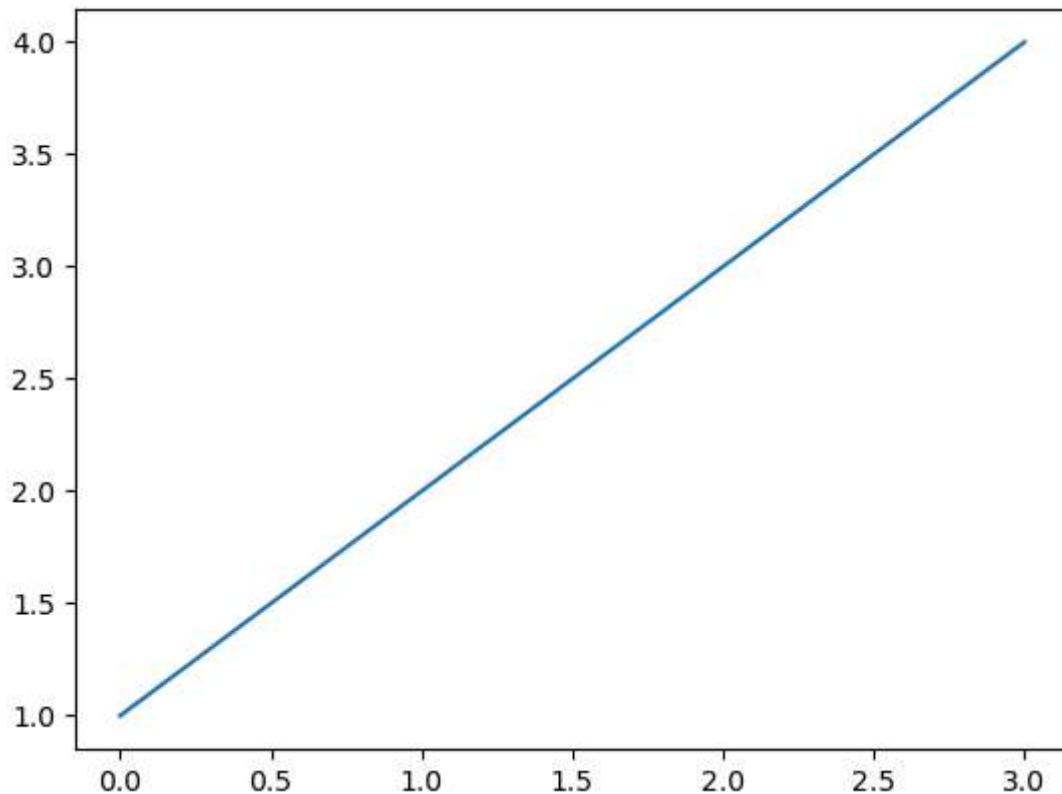
```
In [24]: print(plt.gca()) # get current axix information
```

```
Axes(0.125,0.11;0.775x0.77)
```

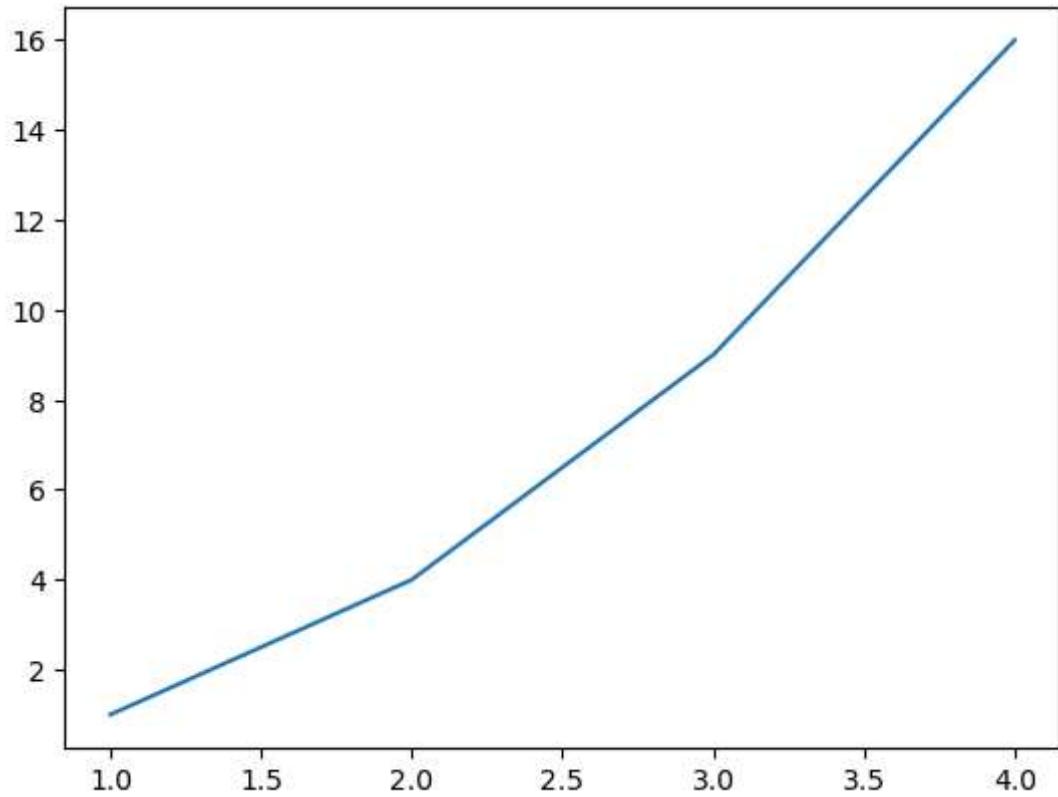


```
In [25]: plt.plot([1,2,3,4])
```

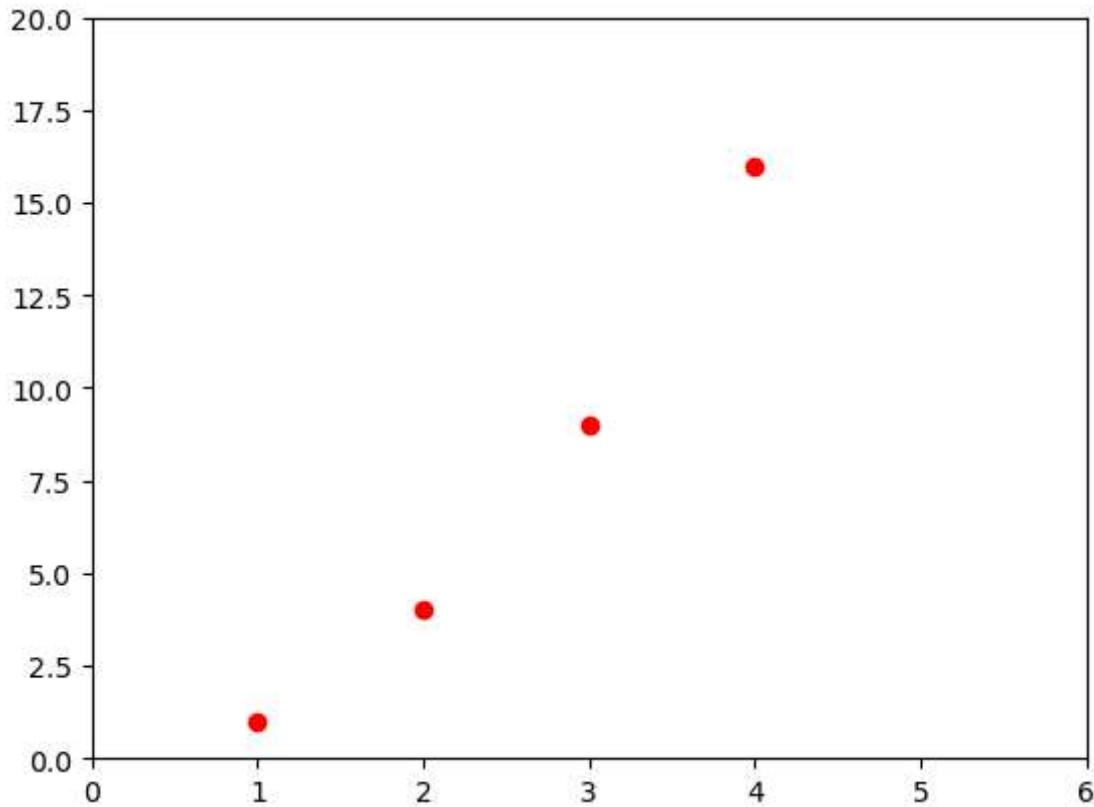
```
Out[25]: [<matplotlib.lines.Line2D at 0x28cf554aad0>]
```



```
In [27]: plt.plot([1,2,3,4],[1,4,9,16])
```

Out[27]: [`<matplotlib.lines.Line2D at 0x28cf540d450>`]

```
In [31]: plt.plot([1,2,3,4],[1,4,9,16],'ro')
plt.axis([0,6,0,20])
plt.show()
```



Numpy + Matplotlib

```
In [ ]: # poc- Proof of concept
```

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

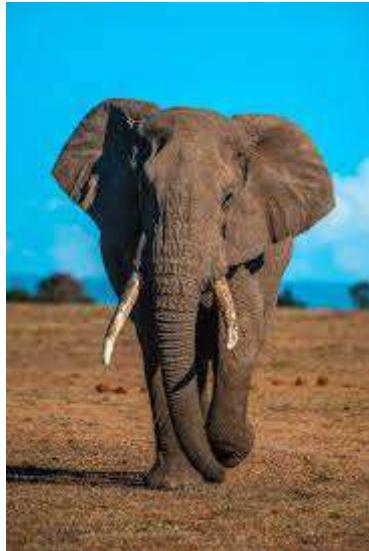
```
In [3]: np.__version__
```

```
Out[3]: '2.3.5'
```

```
In [6]: from PIL import Image
```

```
In [7]: image=Image.open(r'C:\Users\abhis\Downloads\Elephant.jpj.jpg')
image
```

```
Out[7]:
```



```
In [8]: ele_arr=np.asarray(image)
```

```
In [9]: ele_arr
```

```
Out[9]: array([[[ 1, 175, 228],
   [ 0, 175, 228],
   [ 1, 176, 229],
   ...,
   [ 2, 178, 235],
   [ 2, 178, 235],
   [ 2, 178, 235]],

   [[ 0, 175, 228],
   [ 0, 175, 228],
   [ 1, 176, 229],
   ...,
   [ 2, 178, 235],
   [ 2, 178, 235],
   [ 2, 178, 235]],

   [[ 1, 176, 229],
   [ 1, 176, 229],
   [ 1, 176, 229],
   ...,
   [ 3, 179, 236],
   [ 2, 178, 235],
   [ 2, 178, 235]],

   ...,

   [[106, 75, 44],
   [117, 86, 57],
   [106, 75, 46],
   ...,
   [117, 82, 54],
   [106, 71, 43],
   [135, 101, 74]],

   [[104, 73, 42],
   [115, 84, 55],
   [106, 75, 46],
   ...,
   [107, 72, 44],
   [102, 67, 39],
   [120, 86, 59]],

   [[100, 69, 38],
   [111, 80, 51],
   [106, 75, 46],
   ...,
   [104, 69, 41],
   [108, 73, 45],
   [110, 76, 49]]], shape=(275, 183, 3), dtype=uint8)
```

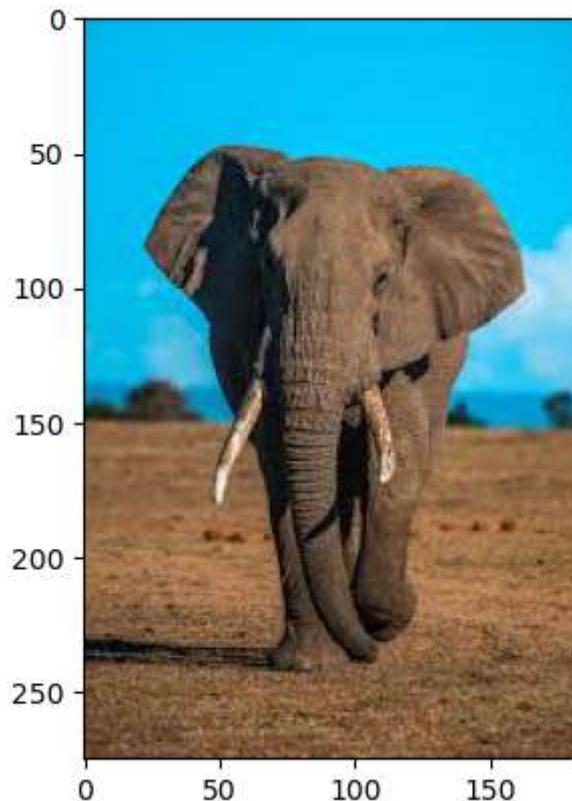
```
In [10]: ele_arr.shape
```

```
Out[10]: (275, 183, 3)
```

```
In [11]: import matplotlib.pyplot as plt
```

```
In [12]: plt.imshow(image)
```

```
Out[12]: <matplotlib.image.AxesImage at 0x1c220f9fc0>
```



```
In [13]: ele_arr.shape
```

```
Out[13]: (275, 183, 3)
```

```
In [ ]:
```