Numpy

```
In [1]: import sys
        sys.version
```

```
Out[1]: '3.13.5 | packaged by Anaconda, Inc. | (main, Jun 12 2025, 16:37:03) [MSC v.192
        9 64 bit (AMD64)]'
```

```
In [2]: import numpy as np #np,ab,etc we can give anything
```

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

```
Out[3]: '2.1.3'
```

Create list

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

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

```
In [5]: type(my_list)
```

```
Out[5]: list
```

```
In [6]: arr=np.array(my_list)
        arr
```

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

```
In [7]: type(arr)
```

```
Out[7]: numpy.ndarray
```

```
In [8]: print(type(arr))
        print(type(my_list))
```

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

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[9], line 1
----> 1 np.arange()

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

```
In [10]: np.arange(10) #arange the number from start to end with step number
```

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

```
In [11]: np.arange(10,20)

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

In [12]: np.arange(10,50,5)

Out[12]: array([10, 15, 20, 25, 30, 35, 40, 45])

In [13]: np.arange(10,30,3)

Out[13]: array([10, 13, 16, 19, 22, 25, 28])

In [14]: np.arange(10,50,5,4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[14], line 1
----> 1 np.arange(10,50,5,4)

TypeError: Cannot interpret '4' as a data type
```

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

Out[15]: array([], dtype=int64)

In [16]: np.arange(8,20)

Out[16]: array([ 8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19])

In [17]: np.arange(20,8) #first arg should be less than second arg

Out[17]: array([], dtype=int64)

In [18]: np.arange(-20,8)

Out[18]: 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])

In [19]: n=np.arange(-20,8)
         n

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])

In [20]: np.zeros(3)

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

In [21]: np.zeros(3,dtype=int) #this is called hyper-parameter tuning(user-defined)

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

In [22]: z=np.zeros(5) #parameter tuning

In [23]: z
```

```
Out[23]:  array([0., 0., 0., 0., 0.])
```

```
In [24]:  np.zeros((5,3)) # here 5 is row and 3 is column
```

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

```
In [25]:  np.zeros((2,2)) #2d array
```

```
Out[25]:  array([[0., 0.],
                 [0., 0.]])
```

```
In [26]:  np.zeros((3,3),dtype=int) #3d array
```

```
Out[26]:  array([[0, 0, 0],
                 [0, 0, 0],
                 [0, 0, 0]])
```

```
In [27]:  nd=np.zeros((5,9),dtype=int) #nd array or multi-d array
          nd
```

```
Out[27]:  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]])
```

```
In [28]:  zero=np.zeros([2,2])
          print(zero)
          print(type(zero))
```

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

```
In [29]:  zero=np.zeros([2,2])
          print(zero)
          print('####')
          print(type(zero))
```

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

```
In [30]:  n1=(6,7)
          n2=(6,8)
          print(np.zeros(n1))
```

```
[[0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 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 [31]:  print(np.zeros(n1,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]]
```

```
In [32]: n2
```

```
Out[32]: (6, 8)
```

```
In [33]: print(np.zeros(n1))
```

```
[[0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 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 [34]: np.ones(3)
```

```
Out[34]: array([1., 1., 1.])
```

```
In [35]: n1
```

```
Out[35]: (6, 7)
```

```
In [36]: np.ones(3,dtype=int)
```

```
Out[36]: array([1, 1, 1])
```

```
In [37]: np.ones((10,10),dtype=int)
```

```
Out[37]: 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 [38]: nd1=np.ones((10,10),dtype=int)
         nd1
```

```
Out[38]: 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 [39]:

```
np.twos((2,3))
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[39], line 1
----> 1 np.twos((2,3))

File ~\Downloads\anaconda\Lib\site-packages\numpy\__init__.py:414, in __getattr__
(attr)
    411     import numpy.char as char
    412     return char.chararray
--> 414 raise AttributeError("module {!r} has no attribute "
    415                      "{!r}".format(__name__, attr))

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

In [ ]:

In [40]:

```
range(5)
```

Out[40]:

```
range(0, 5)
```

In [41]:

```
r=range(5)
r
```

Out[41]:

```
range(0, 5)
```

In [42]:

```
for i in r:
    print(i)
```

```
0
1
2
3
4
```

In [43]:

```
list(range(5))
```

Out[43]:

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

In [44]:

```
range(1,10)
```

Out[44]:

```
range(1, 10)
```

In [45]:

```
list(range(1,10,3))
```

Out[45]:

```
[1, 4, 7]
```

In [46]:

```
y=list(range(12))
y
```

Out[46]:

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

In [ ]:

In [47]:

```
rand(3,2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[47], line 1
----> 1 rand(3,2)

NameError: name 'rand' is not defined
```

```
In [48]: random.rand(2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[48], line 1
----> 1 random.rand(2)

NameError: name 'random' is not defined
```

```
In [49]: np.random.rand(2)

Out[49]: array([0.15593244, 0.02338778])

In [50]: np.random.rand(3)

Out[50]: array([0.0187662 , 0.44106039, 0.40051955])

In [51]: np.random.rand(2,3)

Out[51]: array([[0.14723543, 0.15512262, 0.45168403],
                [0.115749  , 0.10375799, 0.65248818]])

In [52]: np.random.randint(3) # input is either 0 or 1 or 2 but not 3 and > 3

Out[52]: 0

In [53]: np.random.randint(2,3) # 1st must be < 2nd number

Out[53]: 2

In [54]: np.random.randint(2, 10, 4)

Out[54]: array([4, 8, 3, 9], dtype=int32)

In [55]: np.random.randint(-30,10,5)

Out[55]: array([  0,   8, -29,   7, -30], dtype=int32)

In [56]: np.random.randint(10,40,(10,10)) #everytime we execute the output numbers will b
         # declaring without variable

Out[56]: array([[16, 25, 28, 24, 12, 19, 17, 19, 15, 21],
                [32, 33, 28, 29, 26, 12, 39, 14, 31, 16],
                [25, 17, 23, 13, 20, 35, 25, 24, 12, 10],
                [14, 28, 34, 29, 35, 12, 15, 19, 27, 26],
                [17, 15, 18, 26, 25, 29, 23, 21, 23, 12],
                [16, 21, 30, 32, 21, 14, 15, 37, 15, 30],
                [24, 20, 32, 15, 24, 19, 24, 39, 14, 31],
                [29, 33, 20, 19, 11, 15, 13, 22, 27, 17],
                [12, 35, 23, 13, 20, 24, 36, 34, 14, 38],
                [37, 28, 12, 20, 21, 33, 25, 13, 28, 18]], dtype=int32)
```

In [57]:
```
m=np.random.randint(10,40,(10,10)) #declaring with variable
m
```

Out[57]:
```
array([[26, 16, 35, 27, 16, 23, 24, 35, 25, 12],
       [29, 24, 17, 33, 37, 30, 21, 35, 24, 12],
       [18, 14, 39, 20, 25, 21, 10, 24, 16, 11],
       [33, 29, 16, 17, 12, 13, 11, 37, 17, 11],
       [37, 10, 36, 32, 26, 36, 25, 21, 23, 33],
       [38, 37, 39, 14, 35, 15, 35, 34, 22, 22],
       [29, 36, 33, 39, 12, 19, 11, 24, 25, 33],
       [14, 39, 34, 27, 14, 11, 12, 19, 25, 25],
       [11, 36, 29, 25, 37, 36, 15, 20, 15, 16],
       [18, 27, 33, 16, 24, 14, 11, 27, 27, 24]], dtype=int32)
```

In [58]:
```
arr
```

Out[58]:
```
array([0, 1, 2, 3, 4, 5])
```

In [59]:
```
arr.reshape(2,3) # arranging or reshaping the above arr into 2 rows and 3 column
```

Out[59]:
```
array([[0, 1, 2],
       [3, 4, 5]])
```

In [60]:
```
arr.reshape(3,3)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[60], line 1
----> 1 arr.reshape(3,3)

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

In [61]:
```
arr.reshape(6,1)
```

Out[61]:
```
array([[0],
       [1],
       [2],
       [3],
       [4],
       [5]])
```

In [62]:
```
arr.reshape(1,6)
```

Out[62]:
```
array([[0, 1, 2, 3, 4, 5]])
```

In [63]:
```
np.arange(1,13).reshape(3,4)
```

Out[63]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```

In [64]:
```
np.arange(1,13).reshape(12,1)
```

```
Out[64]: array([[ 1],
                [ 2],
                [ 3],
                [ 4],
                [ 5],
                [ 6],
                [ 7],
                [ 8],
                [ 9],
                [10],
                [11],
                [12]])
```

slicing in matrix

```
In [65]: m
```

```
Out[65]: array([[26, 16, 35, 27, 16, 23, 24, 35, 25, 12],
                [29, 24, 17, 33, 37, 30, 21, 35, 24, 12],
                [18, 14, 39, 20, 25, 21, 10, 24, 16, 11],
                [33, 29, 16, 17, 12, 13, 11, 37, 17, 11],
                [37, 10, 36, 32, 26, 36, 25, 21, 23, 33],
                [38, 37, 39, 14, 35, 15, 35, 34, 22, 22],
                [29, 36, 33, 39, 12, 19, 11, 24, 25, 33],
                [14, 39, 34, 27, 14, 11, 12, 19, 25, 25],
                [11, 36, 29, 25, 37, 36, 15, 20, 15, 16],
                [18, 27, 33, 16, 24, 14, 11, 27, 27, 24]], dtype=int32)
```

```
In [66]: b=np.random.randint(10,20,(5,4)) # 5x4 matrix
         b
```

```
Out[66]: array([[15, 16, 12, 13],
                [18, 17, 13, 17],
                [14, 19, 16, 14],
                [16, 14, 17, 19],
                [11, 18, 15, 10]], dtype=int32)
```

```
In [67]: type(b)
```

```
Out[67]: numpy.ndarray
```

```
In [68]: b
```

```
Out[68]: array([[15, 16, 12, 13],
                [18, 17, 13, 17],
                [14, 19, 16, 14],
                [16, 14, 17, 19],
                [11, 18, 15, 10]], dtype=int32)
```

```
In [69]: b[:]
```

```
Out[69]: array([[15, 16, 12, 13],
                [18, 17, 13, 17],
                [14, 19, 16, 14],
                [16, 14, 17, 19],
                [11, 18, 15, 10]], dtype=int32)
```

```
In [70]: b[1:4]
```

```
Out[70]: array([[18, 17, 13, 17],
                [14, 19, 16, 14],
                [16, 14, 17, 19]], dtype=int32)
```

```
In [71]: b
```

```
Out[71]: array([[15, 16, 12, 13],
                [18, 17, 13, 17],
                [14, 19, 16, 14],
                [16, 14, 17, 19],
                [11, 18, 15, 10]], dtype=int32)
```

```
In [72]: b[1,2]
```

```
Out[72]: np.int32(13)
```

```
In [73]: b[-1:]
```

```
Out[73]: array([[11, 18, 15, 10]], dtype=int32)
```

```
In [74]: b[:-1]
```

```
Out[74]: array([[15, 16, 12, 13],
                [18, 17, 13, 17],
                [14, 19, 16, 14],
                [16, 14, 17, 19]], dtype=int32)
```

```
In [75]: b[1,2]
```

```
Out[75]: np.int32(13)
```

```
In [ ]:
```

Operations

```
In [76]: a=np.random.randint(10,20,10)
```

```
In [77]: a
```

```
Out[77]: array([19, 18, 19, 13, 19, 13, 10, 19, 19, 13], dtype=int32)
```

```
In [78]: id(a)
```

```
Out[78]: 2693339363536
```

```
In [79]: arr
```

```
Out[79]: array([0, 1, 2, 3, 4, 5])
```

```
In [80]: arr2=np.random.randint(0,100,(10,10))
```

```
In [81]: arr2
```

```
Out[81]:  array([[15, 81, 43, 77, 36, 41, 70,  3,  7, 48],
                 [ 9, 20, 26, 61, 55, 64,  4, 46, 92, 65],
                 [76, 60, 69, 89,  0, 53, 55, 72, 27, 49],
                 [98, 30, 90, 40, 80, 83, 66, 25, 29, 51],
                 [62, 55, 55, 24, 26, 22, 67, 96, 21, 47],
                 [20, 12, 96, 13, 59, 99, 40, 20, 76, 61],
                 [26, 91, 62, 99, 49, 92, 91, 91, 54, 86],
                 [38, 78, 54, 75, 49, 14, 24, 53,  2, 22],
                 [46, 56,  3, 62, 94, 84, 27,  4, 40, 12],
                 [76, 32, 25, 19,  7, 15, 52, 12, 88,  0]], dtype=int32)
```

```
In [82]:  arr[:]
```

```
Out[82]:  array([0, 1, 2, 3, 4, 5])
```

```
In [83]:  arr
```

```
Out[83]:  array([0, 1, 2, 3, 4, 5])
```

```
In [84]:  arr[:4]
```

```
Out[84]:  array([0, 1, 2, 3])
```

```
In [85]:  arr2[:]
```

```
Out[85]:  array([[15, 81, 43, 77, 36, 41, 70,  3,  7, 48],
                 [ 9, 20, 26, 61, 55, 64,  4, 46, 92, 65],
                 [76, 60, 69, 89,  0, 53, 55, 72, 27, 49],
                 [98, 30, 90, 40, 80, 83, 66, 25, 29, 51],
                 [62, 55, 55, 24, 26, 22, 67, 96, 21, 47],
                 [20, 12, 96, 13, 59, 99, 40, 20, 76, 61],
                 [26, 91, 62, 99, 49, 92, 91, 91, 54, 86],
                 [38, 78, 54, 75, 49, 14, 24, 53,  2, 22],
                 [46, 56,  3, 62, 94, 84, 27,  4, 40, 12],
                 [76, 32, 25, 19,  7, 15, 52, 12, 88,  0]], dtype=int32)
```

```
In [86]:  arr2[0:5]
```

```
Out[86]:  array([[15, 81, 43, 77, 36, 41, 70,  3,  7, 48],
                 [ 9, 20, 26, 61, 55, 64,  4, 46, 92, 65],
                 [76, 60, 69, 89,  0, 53, 55, 72, 27, 49],
                 [98, 30, 90, 40, 80, 83, 66, 25, 29, 51],
                 [62, 55, 55, 24, 26, 22, 67, 96, 21, 47]], dtype=int32)
```

```
In [87]:  arr2[1,4]
```

```
Out[87]:  np.int32(55)
```

```
In [88]:  arr2[-5,5]
```

```
Out[88]:  np.int32(99)
```

```
In [89]:  arr2[::-1]
```

```
Out[89]:  array([[76, 32, 25, 19,  7, 15, 52, 12, 88,  0],
                 [46, 56,  3, 62, 94, 84, 27,  4, 40, 12],
                 [38, 78, 54, 75, 49, 14, 24, 53,  2, 22],
                 [26, 91, 62, 99, 49, 92, 91, 91, 54, 86],
                 [20, 12, 96, 13, 59, 99, 40, 20, 76, 61],
                 [62, 55, 55, 24, 26, 22, 67, 96, 21, 47],
                 [98, 30, 90, 40, 80, 83, 66, 25, 29, 51],
                 [76, 60, 69, 89,  0, 53, 55, 72, 27, 49],
                 [ 9, 20, 26, 61, 55, 64,  4, 46, 92, 65],
                 [15, 81, 43, 77, 36, 41, 70,  3,  7, 48]], dtype=int32)
```

```
In [90]:  arr.max()
```

```
Out[90]:  np.int64(5)
```

```
In [91]:  arr.min()
```

```
Out[91]:  np.int64(0)
```

```
In [92]:  arr
```

```
Out[92]:  array([0, 1, 2, 3, 4, 5])
```

```
In [93]:  arr.mean()
```

```
Out[93]:  np.float64(2.5)
```

```
In [94]:  arr.median()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[94], line 1
----> 1 arr.median()

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

```
In [95]:  from numpy import*
          a= array([1,2,3,4,5])
          median(a)
```

```
Out[95]:  np.float64(3.0)
```

without using import*, find median and mode

```
In [96]:  arr
```

```
Out[96]:  array([0, 1, 2, 3, 4, 5])
```

```
In [97]:  arr.reshape(3,2)
```

```
Out[97]:  array([[0, 1],
                 [2, 3],
                 [4, 5]])
```

In [98]:
```
arr.reshape(6,1)
```

Out[98]:
```
array([[0],
       [1],
       [2],
       [3],
       [4],
       [5]])
```

In [99]:
```
arr.reshape(1,6)
```

Out[99]:
```
array([[0, 1, 2, 3, 4, 5]])
```

In [100…
```
arr.reshape(2,4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[100], line 1
----> 1 arr.reshape(2,4)

ValueError: cannot reshape array of size 6 into shape (2,4)
```

In [101…
```
arr.reshape(2,3,order='C')
```

Out[101…
```
array([[0, 1, 2],
       [3, 4, 5]])
```

In [102…
```
arr.reshape(2,3,order='F') # print element with fortran
```

Out[102…
```
array([[0, 2, 4],
       [1, 3, 5]])
```

In [103…
```
arr
```

Out[103…
```
array([0, 1, 2, 3, 4, 5])
```

In [104…
```
arr.reshape(2,3)
```

Out[104…
```
array([[0, 1, 2],
       [3, 4, 5]])
```

Indexing

In [105…
```
mat = np.arange(0,100).reshape(10,10)
```

In [106…
```
mat
```

Out[106…

```
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 [107…

```
row=4
col=5
```

In [108…

```
col
```

Out[108…

```
5
```

In [109…

```
row
```

Out[109…

```
4
```

In [110…

```
mat
```

Out[110…

```
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 [111…

```
mat[row,col]
```

Out[111…

```
np.int64(45)
```

In [112…

```
mat[4,5]
```

Out[112…

```
np.int64(45)
```

In [113…

```
mat[:]
```

Out[113…

```
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 [114…

```
col=6
```

In [115…

```
mat
```

Out[115…

```
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 [116…

```
mat[6]
```

Out[116…

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

In [117…

```
mat[:,col]
```

Out[117…

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

In [118…

```
mat[row,:]
```

Out[118…

```
array([40, 41, 42, 43, 44, 45, 46, 47, 48, 49])
```

In [119…

```
mat[:,8]
```

Out[119…

```
array([ 8, 18, 28, 38, 48, 58, 68, 78, 88, 98])
```

In [120…

```
mat[:col]
```

Out[120…

```
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]])
```

In [121…

```
mat[:6]
```

Out[121…

```
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]])
```

In [122…

```
row
```

Out[122…

```
4
```

In [123…

```
mat[:row]
```

Out[123…

```
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]])
```

In [124…

```
mat[row:]
```

Out[124…

```
array([[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 [125…

```
mat[:,-1]
```

Out[125…

```
array([ 9, 19, 29, 39, 49, 59, 69, 79, 89, 99])
```

In [126…

```
mat[0:10:3]
```

Out[126…

```
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 [127…

```
mat[::-4]
```

Out[127…

```
array([[90, 91, 92, 93, 94, 95, 96, 97, 98, 99],
       [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
       [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]])
```

In [128…

```
mat[2:6,2:4] # 1:5 --> only row part /// 1:3 -- it indicates only column parts
```

Out[128…

```
array([[22, 23],
       [32, 33],
       [42, 43],
       [52, 53]])
```

In [129…

```
mat[1,6]
```

Out[129…

```
np.int64(16)
```

In [130…

```
mat[1:6]
```

Out[130…

```
array([[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]])
```

In [131…

```
mat[1:2,2:4]
```

Out[131…

```
array([[12, 13]])
```

In [132…

```
mat[2:4,3:5]
```

Out[132…

```
array([[23, 24],
       [33, 34]])
```

In [133…

```
mat[3:-3]
```

Out[133…
```
array([[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 [134…
```
mat[0]
```

Out[134…
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Masking

In [135…
```
mat # we also called as filter
```

Out[135…
```
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 [136…
```
id(mat)
```

Out[136…
```
2693339193936
```

In [137…
```
mat
```

Out[137…
```
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 [138…
```
mat[mat<50]
```

Out[138…
```
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 [139…
```
mat[mat<=50]
```

Out[139…
```
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 [140…
```
mat>50
```

```
Out[140…  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,  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,  True,  True,  True,  True,
                   True]])
```

```
In [141…  mat[mat==50]
```

```
Out[141…  array([50])
```

```
In [142…  mat==50
```

```
Out[142…  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],
                 [ True, 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]])
```

```
In [143…  mat
```

Out[143…

```
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 [144…

```
a1=mat[mat<50]
a1
```

Out[144…

```
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 [145…

```
a2=mat[mat>50]
a2
```

Out[145…

```
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 [146…

```
a3=mat[mat<=50]
a3
```

Out[146…

```
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 [147…

```
a4=mat[mat==50]
a4
```

Out[147…

```
array([50])
```

python program to generate otp

In [148…

```
import random

def generate_otp(length=4):
    """Generate a numeric OTP of a specified length."""
    digits = '012345'
    otp = ''.join(random.choice(digits) for _ in range(length))
    return otp

# Example usage
otp_length = 4  # You can change this to any length you prefer
otp = generate_otp(otp_length)
print(f"Your OTP is: {otp}")
```

```
Your OTP is: 3150
```

In [149…

```
def wish():
    print('good evening')
wish()
```

```
def wish():
    print('good evening')
wish()

def wish():
    print('good evening')
wish()
```

```
good evening
good evening
good evening
```

In [151…

```
def wish():
    print('good evening')
wish()

wish()

wish()
```

```
good evening
good evening
good evening
```

In [152…

```
list1=['a','b','c',1,5]
print(list1.pop)
```

```
<built-in method pop of list object at 0x0000027317A70C40>
```

In [153…

```
x=[1,2,3]
y=x.copy()
x.append(4)
print(x)
```

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

In [ ]: