Numpy Crash Course

In [2]:
```
# NumPy (Numerical Python) is a library for working with arrays and mathematical c
```

In [4]:
```
import numpy as np
```

In [6]:
```
np.__version__
```

Out[6]:
```
'1.26.4'
```

Creating Arrays

In [8]:
```
my_list = [0,1,2,3,4,5]
my_list
```

Out[8]:
```
[0, 1, 2, 3, 4, 5]
```

In [10]:
```
type(my_list)
```

Out[10]:
```
list
```

In [12]:
```
arr = np.array(my_list)
```

In [16]:
```
arr
```

Out[16]:
```
array([0, 1, 2, 3, 4, 5])
```

In [18]:
```
type(arr)
```

Out[18]:
```
numpy.ndarray
```

In [20]:
```
np.arange(15) #arange function gives values from 0 to n-1
```

Out[20]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

In [22]:
```
np.arange(3.0)
```

Out[22]:
```
array([0., 1., 2.])
```

In [24]:
```
np.arange(0,5) # here start and stop values given and the end value will be n-1 va
```

Out[24]:
```
array([0, 1, 2, 3, 4])
```

In [26]:
```
np.arange(10,20)
```

Out[26]:
```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

In [28]:
```
np.arange(20,10) #1st arg < 2nd arg
```

Out[28]:
```
array([], dtype=int32)
```

In [30]:
```
np.arange(-20,10)
```

Out[30]:
```
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 [32]:
```
np.arange(-16,10)
```

Out[32]:
```
array([-16, -15, -14, -13, -12, -11, -10,  -9,  -8,  -7,  -6,  -5,  -4,
        -3,  -2,  -1,   0,   1,   2,   3,   4,   5,   6,   7,   8,   9])
```

In [34]:
```
np.arange(-20,-10)
```

Out[34]:
```
array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11])
```

In [36]:
```
np.arange(10,10) #the distance between start and end was zero
```

Out[36]:
```
array([], dtype=int32)
```

In [38]:
```
np.arange(-1000,1000)
```

Out[38]:
```
array([-1000,  -999,  -998, ...,   997,   998,   999])
```

In [40]:
```
np.arange(10,30,5) # 10 is starting point ,30 is end point,5 is step point
#start,end,step format
```

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

In [42]:
```
np.zeros(5) #parameter tunning
```

Out[42]:
```
array([0., 0., 0., 0., 0.])
```

In [44]:
```
np.zeros(5,dtype=int) #hyperparameter tunning
```

Out[44]:
```
array([0, 0, 0, 0, 0])
```

In [46]:
```
np.zeros((10,10)) #parameter because it automatically understood to give float val
```

Out[46]:
```
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

In [48]:
```
np.zeros((10,10),dtype=int) #hyperparamter becasue it gives integer after giving t
```

Out[48]:
```
array([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]])
```

In [50]:
```
zero = np.zeros([2,2])
type(zero)
```

Out[50]:
```
numpy.ndarray
```

In [52]:
```
np.zeros((2,10)) #2 rows and 10 columns of zeros
#by default 1st arg give rows and 2nd arg gives columns
```

Out[52]:
```
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

In [54]:
```
n = (6,7)
n1 = (6,8)
print(np.zeros(n)) #parameter 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 [56]:
```
print(np.zeros(n1,dtype=int)) #hyperparameter 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 0 0 0 0 0 0]]
```

In [58]:
```
np.ones(4,dtype=int)
```

Out[58]:
```
array([1, 1, 1, 1])
```

In [60]:
```
np.ones(n) # n is declared above
```

Out[60]:
```
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 [62]:
```
np.ones((5,4),dtype=int)
```

Out[62]:
```
array([[1, 1, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 1]])
```

In [64]:
```
np.twos((2,3)) #twos are not the functions of numpy
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[64], line 1
----> 1 np.twos((2,3))

File C:\ProgramData\anaconda3\Lib\site-packages\numpy\__init__.py:333, in __getattr
__(attr)
    330     "Removed in NumPy 1.25.0"
    331     raise RuntimeError("Tester was removed in NumPy 1.25.")
--> 333 raise AttributeError("module {!r} has no attribute "
    334                      "{!r}".format(__name__, attr))

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

In [68]:
```
range(5) #range we don't work with numpy where as arange we work with numpy
```

Out[68]:
```
range(0, 5)
```

In [70]:
```
r = range(5)
r
```

Out[70]:
```
range(0, 5)
```

In [72]:
```
for i in r:
    print(i)
```

```
0
1
2
3
4
```

In [74]:
```
list(range(5))
```

Out[74]:
```
[0, 1, 2, 3, 4]
```

In [76]:
```
range(1,10)
```

Out[76]:
```
range(1, 10)
```

In [78]:
```
list(range(1,10))
```

Out[78]:
```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [80]:
```
list(range(1,10,3))
```

Out[80]:
```
[1, 4, 7]
```

In [82]:
```
y = list(range(12))
y
```

Out[82]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

In [84]:
```
rand(3,2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[84], line 1
----> 1 rand(3,2)

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

In [86]:
```
rand(3,2)
random.rand(3,2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[86], line 1
----> 1 rand(3,2)
      2 random.rand(3,2)

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

In [88]:
```
np.random.rand(3,2) # np is a package
                    # random is a module
                    # rand is a function
```

Out[88]:
```
array([[0.26614058, 0.12797775],
       [0.3540834 , 0.36057263],
       [0.63566505, 0.00178527]])
```

In [90]:
```
np.random.rand(4)
```

Out[90]:
```
array([0.56361071, 0.45841093, 0.80249944, 0.77808233])
```

In [92]:
```
np.random.rand(2,4)
```

Out[92]:
```
array([[0.26261448, 0.98773985, 0.77724201, 0.40601075],
       [0.2459439 , 0.851561  , 0.12810256, 0.07183754]])
```

In [94]:
```
np.random.randint(2,4) #generate an integer in betweebn 2,4
                       #everytime you print you get a random number b/w the range
```

Out[94]:
```
3
```

In [96]:
```
np.random.randint(2,20) #2nd arg is exclusive
```

Out[96]:
```
7
```

In [98]:
```
np.random.randint(0,1)
```

Out[98]:
```
0
```

In [100]:
```
np.random.randint(1,2,3)
```

Out[100]:
```
array([1, 1, 1])
```

In [ ]:
```

```

Day11 Numpy2 Aug16

In [216]:
```
np.random.randint(1)
```

Out[216]:
```
0
```

In [224]:
```
np.random.randint(5,9) #Get the random value from 5 to 9
                       # we keep on executing multiple times we get diffrent outpu
```

Out[224]:
```
6
```

In [246]:
```
np.random.randint(10,21,3) # it gives the output must be random number between 10
# if there is 3 values in randit it refers to (startpoint,endpoint,no.ofcolumns)
```

Out[246]:
```
array([19, 14, 18])
```

In [248]:
```
np.random.randint(1,12,10) # get numbers b/w 1 to 12 in 10 columns
```

Out[248]:
```
array([ 8,  3,  1,  9,  5,  2,  2, 11, 11,  3])
```

In [250]:
```
np.random.randint(10,40,(10,10))
# we get the matricx where it contains random numbers from 10 to 40 in 10 rows and
```

Out[250]:
```
array([[35, 26, 24, 13, 13, 35, 30, 36, 26, 34],
       [19, 21, 19, 17, 14, 25, 16, 18, 27, 33],
       [24, 20, 12, 13, 37, 17, 24, 21, 21, 31],
       [24, 37, 17, 14, 37, 21, 23, 27, 16, 12],
       [37, 12, 26, 16, 24, 29, 25, 27, 16, 25],
       [14, 32, 14, 15, 24, 35, 25, 21, 14, 12],
       [30, 24, 16, 33, 35, 20, 25, 19, 31, 28],
       [31, 15, 34, 36, 29, 31, 27, 28, 32, 10],
       [23, 17, 23, 27, 36, 22, 16, 37, 13, 22],
       [14, 10, 14, 30, 23, 16, 29, 39, 21, 21]])
```

In [291]:
```
b = np.random.randint(10,20,(5,4))
b
```

Out[291]:
```
array([[15, 18, 14, 18],
       [19, 18, 16, 11],
       [17, 14, 14, 10],
       [10, 19, 11, 12],
       [10, 12, 14, 14]])
```

":" - is used for slicing , we need to provide rows and columns among the slicing points

In [293]:
```
b[:]
#we use empty slicing to print all the elements
```

Out[293]:
```
array([[15, 18, 14, 18],
       [19, 18, 16, 11],
       [17, 14, 14, 10],
       [10, 19, 11, 12],
       [10, 12, 14, 14]])
```

In [295]:
```
b [0:2] #[row:col] format
#it will give elements from 0 to first column because the last element was n-1
```

Out[295]:
```
array([[15, 18, 14, 18],
       [19, 18, 16, 11]])
```

In [297]:
```
b
```

Out[297]:
```
array([[15, 18, 14, 18],
       [19, 18, 16, 11],
       [17, 14, 14, 10],
       [10, 19, 11, 12],
       [10, 12, 14, 14]])
```

In [299]:
```
b[0:-1]
# we get elements from top 0 to bottom starts from -1,-2....so on so we get n-1 i.
```

Out[299]:
```
array([[15, 18, 14, 18],
       [19, 18, 16, 11],
       [17, 14, 14, 10],
       [10, 19, 11, 12]])
```

"," - is used to get specific indexing number

In [ ]:
```
```

In [303]:
```
b[0,2] # "," is used to get specific indexing number
# here we get the element in the 0 row nad 2nd column
```

Out[303]:
```
14
```

In [309]:
```
np.random.randint(10,20,(4,4))
# we get random numbers between 10 to 20 in 4x4 matrix
```

Out[309]:
```
array([[15, 16, 14, 16],
       [10, 11, 11, 16],
       [10, 18, 12, 13],
       [16, 14, 15, 10]])
```

Operations

In [312]:
```
a = np.random.randint(10,20,5)
a
# we get random numbers b/w 10 and 20 in 5 columns
```

Out[312]:
```
array([14, 12, 17, 10, 17])
```

In [314]:
```
arr
```

Out[314]:
```
array([0, 1, 2, 3, 4, 5])
```

In [340]:
```
arr2 = np.random.randint(0,100,(10,10))
arr2
# we get random numbers b/w 0 to 100 in 10 rows and 10 columns
```

Out[340]:
```
array([[ 8,  3, 84, 10, 48, 84, 86, 41, 58,  2],
       [30, 27, 36,  2, 45, 68, 46, 66, 21, 26],
       [82, 91, 19, 77, 41,  4,  4, 20, 91, 71],
       [83, 35, 15, 78, 22, 14, 29, 77, 50, 29],
       [28, 73, 10, 43, 51, 47, 43,  0, 46, 75],
       [24, 47, 71, 23,  6, 99,  0, 82, 16, 17],
       [36, 73, 41, 63, 85, 90, 29, 48, 60, 16],
       [83, 31, 62, 13, 89, 11, 70, 24, 51, 45],
       [22, 37, 33, 34, 68, 74,  3, 98, 39, 62],
       [15, 16, 78, 16, 38, 24, 26, 27,  6, 49]])
```

In [342]:
```
arr
```

Out[342]:
```
array([0, 1, 2, 3, 4, 5])
```

In [344]:
```
arr[:]
# we get all the elements in the arr
```

Out[344]:
```
array([0, 1, 2, 3, 4, 5])
```

In [346]:
```
arr[:4]
# we get elements from starting to 4th column i.e index 3 (4-1)
```

Out[346]:
```
array([0, 1, 2, 3])
```

In [348]:
```
arr2[:]
#prints all elements in the arr2
```

Out[348]:
```
array([[ 8,  3, 84, 10, 48, 84, 86, 41, 58,  2],
       [30, 27, 36,  2, 45, 68, 46, 66, 21, 26],
       [82, 91, 19, 77, 41,  4,  4, 20, 91, 71],
       [83, 35, 15, 78, 22, 14, 29, 77, 50, 29],
       [28, 73, 10, 43, 51, 47, 43,  0, 46, 75],
       [24, 47, 71, 23,  6, 99,  0, 82, 16, 17],
       [36, 73, 41, 63, 85, 90, 29, 48, 60, 16],
       [83, 31, 62, 13, 89, 11, 70, 24, 51, 45],
       [22, 37, 33, 34, 68, 74,  3, 98, 39, 62],
       [15, 16, 78, 16, 38, 24, 26, 27,  6, 49]])
```

In [350]:
```
arr2[0:5]
# gives elements from 0 to 4th index i.e (5-1)
```

Out[350]:
```
array([[ 8,  3, 84, 10, 48, 84, 86, 41, 58,  2],
       [30, 27, 36,  2, 45, 68, 46, 66, 21, 26],
       [82, 91, 19, 77, 41,  4,  4, 20, 91, 71],
       [83, 35, 15, 78, 22, 14, 29, 77, 50, 29],
       [28, 73, 10, 43, 51, 47, 43,  0, 46, 75]])
```

In [352]:
```
arr2[1,5]
#gives the element where it is placed in 1st index and 5th index
```

Out[352]: 68

In [354]:
```
arr2
```

Out[354]:
```
array([[ 8,  3, 84, 10, 48, 84, 86, 41, 58,  2],
       [30, 27, 36,  2, 45, 68, 46, 66, 21, 26],
       [82, 91, 19, 77, 41,  4,  4, 20, 91, 71],
       [83, 35, 15, 78, 22, 14, 29, 77, 50, 29],
       [28, 73, 10, 43, 51, 47, 43,  0, 46, 75],
       [24, 47, 71, 23,  6, 99,  0, 82, 16, 17],
       [36, 73, 41, 63, 85, 90, 29, 48, 60, 16],
       [83, 31, 62, 13, 89, 11, 70, 24, 51, 45],
       [22, 37, 33, 34, 68, 74,  3, 98, 39, 62],
       [15, 16, 78, 16, 38, 24, 26, 27,  6, 49]])
```

In [356]:
```
arr2[-5,5]
# we get the element which is placed in -5 index and 5th index
# - means backward indexing that shoud start from bottom to top starting at -1
```

Out[356]: 99

In [358]:
```
arr2[-5,-5]
# we get the number from -5 row from bottop to top  and -5 column from left to rig
```

Out[358]: 99

In [360]:
```
arr2
```

Out[360]:
```
array([[ 8,  3, 84, 10, 48, 84, 86, 41, 58,  2],
       [30, 27, 36,  2, 45, 68, 46, 66, 21, 26],
       [82, 91, 19, 77, 41,  4,  4, 20, 91, 71],
       [83, 35, 15, 78, 22, 14, 29, 77, 50, 29],
       [28, 73, 10, 43, 51, 47, 43,  0, 46, 75],
       [24, 47, 71, 23,  6, 99,  0, 82, 16, 17],
       [36, 73, 41, 63, 85, 90, 29, 48, 60, 16],
       [83, 31, 62, 13, 89, 11, 70, 24, 51, 45],
       [22, 37, 33, 34, 68, 74,  3, 98, 39, 62],
       [15, 16, 78, 16, 38, 24, 26, 27,  6, 49]])
```

In [362]:
```
arr2[-1,-2]
# we get the element from -1 row and -2 column
```

Out[362]: 6

In [364]:
```
arr2
```

Out[364]:
```
array([[ 8,  3, 84, 10, 48, 84, 86, 41, 58,  2],
       [30, 27, 36,  2, 45, 68, 46, 66, 21, 26],
       [82, 91, 19, 77, 41,  4,  4, 20, 91, 71],
       [83, 35, 15, 78, 22, 14, 29, 77, 50, 29],
       [28, 73, 10, 43, 51, 47, 43,  0, 46, 75],
       [24, 47, 71, 23,  6, 99,  0, 82, 16, 17],
       [36, 73, 41, 63, 85, 90, 29, 48, 60, 16],
       [83, 31, 62, 13, 89, 11, 70, 24, 51, 45],
       [22, 37, 33, 34, 68, 74,  3, 98, 39, 62],
       [15, 16, 78, 16, 38, 24, 26, 27,  6, 49]])
```

In [366]:
```
arr2[::-1]
# this is in the format[start:stop:step]
```

```
#here there is no start and stop but step is -1
# so we will get all the elements from the bottom to top i.e top to bottom in reve
```

Out[366]:
```
array([[15, 16, 78, 16, 38, 24, 26, 27,  6, 49],
       [22, 37, 33, 34, 68, 74,  3, 98, 39, 62],
       [83, 31, 62, 13, 89, 11, 70, 24, 51, 45],
       [36, 73, 41, 63, 85, 90, 29, 48, 60, 16],
       [24, 47, 71, 23,  6, 99,  0, 82, 16, 17],
       [28, 73, 10, 43, 51, 47, 43,  0, 46, 75],
       [83, 35, 15, 78, 22, 14, 29, 77, 50, 29],
       [82, 91, 19, 77, 41,  4,  4, 20, 91, 71],
       [30, 27, 36,  2, 45, 68, 46, 66, 21, 26],
       [ 8,  3, 84, 10, 48, 84, 86, 41, 58,  2]])
```

In [368]:
```
arr2
```

Out[368]:
```
array([[ 8,  3, 84, 10, 48, 84, 86, 41, 58,  2],
       [30, 27, 36,  2, 45, 68, 46, 66, 21, 26],
       [82, 91, 19, 77, 41,  4,  4, 20, 91, 71],
       [83, 35, 15, 78, 22, 14, 29, 77, 50, 29],
       [28, 73, 10, 43, 51, 47, 43,  0, 46, 75],
       [24, 47, 71, 23,  6, 99,  0, 82, 16, 17],
       [36, 73, 41, 63, 85, 90, 29, 48, 60, 16],
       [83, 31, 62, 13, 89, 11, 70, 24, 51, 45],
       [22, 37, 33, 34, 68, 74,  3, 98, 39, 62],
       [15, 16, 78, 16, 38, 24, 26, 27,  6, 49]])
```

In [370]:
```
arr2[::-2]
#we will get all the elements from bottom to top with a single step i.e -1,-2 skip
```

Out[370]:
```
array([[15, 16, 78, 16, 38, 24, 26, 27,  6, 49],
       [83, 31, 62, 13, 89, 11, 70, 24, 51, 45],
       [24, 47, 71, 23,  6, 99,  0, 82, 16, 17],
       [83, 35, 15, 78, 22, 14, 29, 77, 50, 29],
       [30, 27, 36,  2, 45, 68, 46, 66, 21, 26]])
```

In [372]:
```
arr2
```

Out[372]:
```
array([[ 8,  3, 84, 10, 48, 84, 86, 41, 58,  2],
       [30, 27, 36,  2, 45, 68, 46, 66, 21, 26],
       [82, 91, 19, 77, 41,  4,  4, 20, 91, 71],
       [83, 35, 15, 78, 22, 14, 29, 77, 50, 29],
       [28, 73, 10, 43, 51, 47, 43,  0, 46, 75],
       [24, 47, 71, 23,  6, 99,  0, 82, 16, 17],
       [36, 73, 41, 63, 85, 90, 29, 48, 60, 16],
       [83, 31, 62, 13, 89, 11, 70, 24, 51, 45],
       [22, 37, 33, 34, 68, 74,  3, 98, 39, 62],
       [15, 16, 78, 16, 38, 24, 26, 27,  6, 49]])
```

In [ ]:
```
arr2[::-3]
#we get elements from bottom to top with 2 steps i.e -1,-2 -3 : -1,-2,-3...
```

In [374]:
```
arr
```

Out[374]:
```
array([0, 1, 2, 3, 4, 5])
```

In [376]:
```
arr.max()
# gives the largest number
```

Out[376]: 5

In [378]:
```
arr.min()
# gives the smallest number
```

Out[378]: 0

In [380]:
```
arr.mean()
#gives the mean i.e average of the array
```

Out[380]: 2.5

In [382]:
```
arr.median()
# theres no such attribute which is median we need to use different method
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[382], line 1
----> 1 arr.median()

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

In [386]:
```
from numpy import *
a = array([1,2,3,4,9])
median(a)
```

Out[386]: 3.0

In [390]:
```
arr
```

Out[390]: array([0, 1, 2, 3, 4, 5])

In [388]:
```
arr.reshape(2,3)
# reshape is used to convert our array in the specific form
# here we are using reshape to print the existingarray in 2 rows and 3 columns
```

Out[388]:
```
array([[0, 1, 2],
       [3, 4, 5]])
```

In [392]:
```
arr.reshape(6,1)
# we are reshaping our array in 6 rows and 1 column
```

Out[392]:
```
array([[0],
       [1],
       [2],
       [3],
       [4],
       [5]])
```

In [394]:
```
arr.reshape(1,6)
# we are reshaping our array in 1 rows and 6 column
```

Out[394]: array([[0, 1, 2, 3, 4, 5]])

In [396]:
```
arr
```

Out[396]: array([0, 1, 2, 3, 4, 5])

In [398]:
```
arr.reshape(1,5)
# we cannot reshape the array to reducing the columns or rows because it cant cut
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[398], line 1
----> 1 arr.reshape(1,5)

ValueError: cannot reshape array of size 6 into shape (1,5)
```

Only three types of orders are there in reshape **C - Type** **F - Type** **A - Type**

C type indexing order

In [400]:
```
arr.reshape(3,2,order='C') # C type indexing order
#it will give the array as we give the order
# it gives the array in 3 rows and 2 columns
```

Out[400]:
```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

F - fortran indexing order

In [406]:
```
arr.reshape(3,2,order='F') # Fortran indexing order
#it will give the array as the rows becomes columns and columns becomes rows
# it gives in the order like 0,1,2,3,4,5
```

Out[406]:
```
array([[0, 3],
       [1, 4],
       [2, 5]])
```

In [410]:
```
arr.reshape(2,3,order='F')
# now it was reshaped from rows to columsns and columns to rows
```

Out[410]:
```
array([[0, 2, 4],
       [1, 3, 5]])
```

A -Arbitary indexing order

In [ ]:
```
arr.reshape(3,2,order='A') #
```

In [426]:
```
arr.reshape(2,3)
# we get the values of array in 2 rows and 3 columns
```

Out[426]:
```
array([[0, 1, 2],
       [3, 4, 5]])
```

In [428]:
```
arr.reshape(1,4)
#we cannot give the rows and columns less than the array rows and columns
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[428], line 1
----> 1 arr.reshape(1,4)

ValueError: cannot reshape array of size 6 into shape (1,4)
```

In [430]:
```
arr.reshape(1,6)
```

Out[430]:
```
array([[0, 1, 2, 3, 4, 5]])
```

In [432]:
```
arr.reshape(6,1)
# we get the array values in 6 rows and 1 column
```

Out[432]:
```
array([[0],
       [1],
       [2],
       [3],
       [4],
       [5]])
```

In [436]:
```
arr
```

Out[436]:
```
array([0, 1, 2, 3, 4, 5])
```

In [444]:
```
arr.shape
```

Out[444]:
```
(6,)
```

In [448]:
```
arr.reshape(2,6)
# because theres no
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[448], line 1
----> 1 arr.reshape(2,6)

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

In [438]:
```
arr
```

Out[438]:
```
array([0, 1, 2, 3, 4, 5])
```

In [440]:
```
arr.reshape(3,2)
# we are reshaping array into 3 rows and 2 columns
```

Out[440]:
```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

Indexing

In [453]:
```
mat = np.arange(0,100).reshape(10,10)
mat
#we are printing the Matrix from 0 to 100 in 10 rows and 10 columns
```

Out[453]:
```
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 [455]:
```
row = 4
col = 5
```

In [457]:
```
col
```

Out[457]: 5

In [459]:
```
row
```

Out[459]: 4

In [461]:
```
mat
```

Out[461]:
```
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 [465]:
```
mat[row,col]
# we get the element where row = 4 & col = 5 that we asignes
```

Out[465]: 45

In [467]:
```
mat[4,5]
#this is the same thing that without assigning a value
```

Out[467]: 45

In [469]:
```
mat
```

Out[469]:
```
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 [471]:
```
mat[:]
# prints the matrix
```

Out[471]:
```
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 [473]:
```
col = 6
```

In [475]:
```
mat
```

Out[475]:
```
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 [477]:
```
# With slicing
mat[:,col]
# here it gives all the elements in the column index 6 (where col = 6 assigned abo
```

Out[477]:
```
array([ 6, 16, 26, 36, 46, 56, 66, 76, 86, 96])
```

In [481]:
```
row
```

Out[481]:
```
4
```

In [479]:
```
mat[row,:]
# here it gives all the elements in the row index 4 (where row = 4 assigned above)
```

Out[479]:
```
array([40, 41, 42, 43, 44, 45, 46, 47, 48, 49])
```

In [483]:
```
mat
```

Out[483]:
```
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 [485]:
```
mat[:,col]
# here it prints all the elements in the col = 6
```

Out[485]:
```
array([ 6, 16, 26, 36, 46, 56, 66, 76, 86, 96])
```

In [489]:
```
mat[:col]
# here it prints all the elements from starting to col index = 6
```

Out[489]:
```
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 [491]:
```
mat
```

Out[491]:
```
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 [495]:
```
row
```

Out[495]:
```
4
```

In [493]:
```
mat[:row]
# here we print all the elements from starting to the row index = 4
```

Out[493]:
```
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 [499]:
```
mat[row:]
# here we print all the elements from row index = 4 to the end
```

Out[499]:
```
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 [504]:
```
mat[:]
```

Out[504]:
```
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 [506]:
```
mat[:,8]
# here we get all the elements in the col index = 8
```

Out[506]:
```
array([ 8, 18, 28, 38, 48, 58, 68, 78, 88, 98])
```

In [508]:
```
mat [:,-1]
# here we get all the elements in the col index = -1 where negative index start fr
```

Out[508]:
```
array([ 9, 19, 29, 39, 49, 59, 69, 79, 89, 99])
```

In [510]:
```
mat
```

Out[510]:
```
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 [512]:
```
row
```

Out[512]:
```
4
```

In [514]:
```
col
```

Out[514]:
```
6
```

In [518]:
```
mat
```

Out[518]:
```
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 [516]:
```
mat[:,col]
# we get all the elements from the col index = 6
```

Out[516]:
```
array([ 6, 16, 26, 36, 46, 56, 66, 76, 86, 96])
```

In [522]:
```
mat
```

Out[522]:
```
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 [520]:
```
mat[1,4]
# we get the element in the row index = 1 and column index = 4
```

Out[520]:
```
14
```

In [524]:
```
mat
```

Out[524]:
```
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 [526]:
```
mat[3,-3]
# we get the element in the row index = 3 and column index = -3
# where negative column index start from right to left from -1,-2,-3....
```

Out[526]:
```
37
```

In [528]:
```
mat
```

Out[528]:
```
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 [530]:
```
mat[0]
# gives the elements in the 0th row index
```

Out[530]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [532]:
```
mat[6]
```

```
# gives the elements where row index = 6
```

Out[532]:
```
array([60, 61, 62, 63, 64, 65, 66, 67, 68, 69])
```

In [534]:
```
mat
```

Out[534]:
```
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 [536]:
```
mat [6:]
# we get the elements from the row index =  6 to the end
```

Out[536]:
```
array([[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 [538]:
```
mat[:6]
# we get the elements from starting to the row index = 6
```

Out[538]:
```
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 [540]:
```
mat
```

Out[540]:
```
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 [542]:
```
mat [5:7]
# we get the elements from the row index = 5 to row index = 7
```

Out[542]:
```
array([[50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
       [60, 61, 62, 63, 64, 65, 66, 67, 68, 69]])
```

In [544]:
```
mat
```

Out[544]:
```
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 [546]:
```
mat[0:10]
# we get the elements from the row index = 0 to row index = 10
```

Out[546]:
```
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 [548]:
```
mat[0:10:3]
# we get the elements from the row index = 0 to row index = 10 and the step = 3
# step works like starting element = 0 and starts like 0-1,2-3,4,5-6 prints
```

Out[548]:
```
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 [554]:
```
mat[4:]
# gives the element from the row index = 4 to the end
```

Out[554]:
```
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 [556]:
```
mat[:4]
#gives the elements from the starting point to row index = 4
```

Out[556]:
```
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 [558]:
```
mat[::-1]
#gives all the elements in the reverse order because the step was -1 it works fro
```

Out[558]:
```
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 [560]:
```
mat[::-2]
# it gives the elements from bottom to top with step -2
```

Out[560]:
```
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 [564]:
```
mat[::-3]
# it gives the elements from bottom to top with step -3
```

Out[564]:
```
array([[90, 91, 92, 93, 94, 95, 96, 97, 98, 99],
       [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
       [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
       [ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9]])
```

In [566]:
```
mat[::-5]
# it gives the elements from bottom to top with step -5
```

Out[566]:
```
array([[90, 91, 92, 93, 94, 95, 96, 97, 98, 99],
       [40, 41, 42, 43, 44, 45, 46, 47, 48, 49]])
```

In [570]:
```
mat
```

Out[570]:
```
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 [568]:
```
mat[2:6,2:4]
# it gives elements from only row index from 2 to 6 & col index from 1 to 3
```

Out[568]:
```
array([[22, 23],
       [32, 33],
       [42, 43],
       [52, 53]])
```

In [574]:
```
mat[1:2,2:4]
# it gives elements from only row index from 1 to 2 & col index from 2 to 4
```

Out[574]:
```
array([[12, 13]])
```

In [576]:
```
mat[3:5,2:4]
# it gives elements from only row index from 3 to 5 & col index from 2 to 4
```

Out[576]:
```
array([[32, 33],
       [42, 43]])
```

In [578]:
```
mat
```

Out[578]:
```
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 [580]:
```
mat[2:3,4:5]
# it gives elements from only row index from 2 to 3 & col index from 4 to 5
```

Out[580]:
```
array([[24]])
```

Maksing also called as Filter

In [583]:
```
mat
```

Out[583]:
```
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 [585]:
```
id(mat)
```

Out[585]:
```
2143102888304
```

In [587]:
```
mat < 50
# we are filtering the values less than 50 will be true and remains false
```

Out[587]:
```
array([[ 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,
         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,
        False]])
```

In [589]:
```
mat > 50
# we are filtering the values greater than 50 will be true and remains false
```

Out[589]:
```
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 [591]:
```
mat == 50
# we are filtering the values equal to 50 will be true and remains false
```

Out[591]:
```
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 [593]:
```
mat[mat==50]
#it will shows the values in the matrix = 50 in the array format
```

Out[593]:
```
array([50])
```

In [595]:
```
a1 = mat[mat<50]
a1
# we are assigning a1 as it contains the values which are less than 50 in the matr
```

Out[595]:
```
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 [597]:
```
a2 = mat[mat>50]
a2
# we are assigning a2 as it contains the values which are greater than 50 in the m
```

Out[597]:
```
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 [599]:
```
a3 = mat[mat>=50]
a3
# we are assigning a3 as it contains the values which greater than or  equal to 50
```

Out[599]:
```
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 [601]:
```
a4 = mat[mat==50]
a4
# we are assigning a4 as it contains the values which are equal to 50 in the matri
```

Out[601]:
```
array([50])
```

In [603]:
```
a1
```

Out[603]:
```
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 [605]:
```
a2
```

Out[605]:
```
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 [607]:
```
a3
```

Out[607]:
```
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 [609]:
```
a4
```

Out[609]:
```
array([50])
```

In [ ]: