Numpy Practise

In [1]:
```
import numpy as np
```

In [2]:
```
np.__version__
```

Out[2]:
```
'1.26.4'
```

In [3]:
```
import sys
sys.version
```

Out[3]:
```
'3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.192
9 64 bit (AMD64)]'
```

Creating Arrays

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)
```

In [7]:
```
arr
```

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

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

Out[8]:
```
numpy.ndarray
```

In [9]:
```
type(my_list)
```

Out[9]:
```
list
```

In [11]:
```
np.arange(15)
```

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

In [12]:
```
np.arange(0,15,2)
```

Out[12]:
```
array([ 0,  2,  4,  6,  8, 10, 12, 14])
```

In [15]:
```
np.arange(20,10) #here the starting values ig greater than end index thats why e
```

Out[15]:
```
array([], dtype=int32)
```

```
In [16]: np.arange(10,20)

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

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

Out[17]: array([-10,  -9,  -8,  -7,  -6,  -5,  -4,  -3,  -2,  -1,   0,   1,   2,
                  3,   4,   5,   6,   7,   8,   9])

In [18]: np.zeros(3)

Out[18]: array([0., 0., 0.])

In [19]: np.zeros(10,dtype=int)

Out[19]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

In [21]: np.zeros((2,2),dtype=int)

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

In [22]: np.zeros((4,4),dtype=int)

Out[22]: array([[0, 0, 0, 0],
                [0, 0, 0, 0],
                [0, 0, 0, 0],
                [0, 0, 0, 0]])

In [24]: np.ones(5,dtype=int)

Out[24]: array([1, 1, 1, 1, 1])

In [27]: np.ones((2,3),dtype=int)

Out[27]: array([[1, 1, 1],
                [1, 1, 1]])

In [29]: np.ones(3)

Out[29]: array([1., 1., 1.])

In [31]: zero = np.zeros([2,2])
         print(zero)
         print(type(zero))

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

In [33]: np.zeros((2,3)) # (rows,col)

Out[33]: array([[0., 0., 0.],
                [0., 0., 0.]])

In [39]: n = (6,7)
         n1 = (10,20)
         print(np.zeros(n))
```

```
[[0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 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 [42]:
```
print(np.zeros(n,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 [43]:
```
n
```

Out[43]:
```
(6, 7)
```

In [44]:
```
n1
```

Out[44]:
```
(10, 20)
```

In [45]:
```
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. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 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 [46]:
```
np.ones(4, dtype=int)
```

Out[46]:
```
array([1, 1, 1, 1])
```

In [47]:
```
np.twos((2,3))
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[47], line 1
----> 1 np.twos((2,3))

File ~\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 [49]:
```
np.three(2,3) #there is no such function calles three or two in numpy module
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[49], line 1
----> 1 np.three(2,3)

File ~\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 'three'
```

In [50]:
```
range(5)
```

Out[50]:
```
range(0, 5)
```

In [52]:
```
r = range(0,5)
r
```

Out[52]:
```
range(0, 5)
```

In [53]:
```
for i in r:
    print(i)
```

```
0
1
2
3
4
```

In [54]:
```
list(r)
```

Out[54]:
```
[0, 1, 2, 3, 4]
```

In [56]:
```
list(range(0,10))
```

Out[56]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [57]:
```
list(range(0,10,2))
```

Out[57]:
```
[0, 2, 4, 6, 8]
```

In [58]:
```
rand(3,2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[58], line 1
----> 1 rand(3,2)

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

In [71]:
```
np.random.rand(5)
```

Out[71]:
```
array([0.02803278, 0.65880138, 0.09295226, 0.07486787, 0.22677194])
```

In [76]:
```
np.random.rand(2,4) #(row,col)
```

```
Out[76]:  array([[0.24289938, 0.28414846, 0.86695332, 0.66025406],
                 [0.75225251, 0.99272165, 0.45923257, 0.08496971]])
```

```
In [91]:  np.random.randint(4,5) #it gives a random value from 4 - 5...but doesnt give 2nd
```

```
Out[91]:  4
```

```
In [92]:  np.random.randint(5,10)
```

```
Out[92]:  9
```

```
In [97]:  np.random.randint(5,10,5) #(low value, high val, no. of elements to be in array)
```

```
Out[97]:  array([6, 9, 6, 5, 9])
```

```
In [98]:  np.random.randint(69,96,6)
```

```
Out[98]:  array([84, 93, 70, 71, 75, 79])
```

```
In [109…  np.random.randint(5) #it give a random value under 5
```

```
Out[109…  0
```

```
In [111…  np.random.randint(30,20,10) #as 1st arg has greate val than 2nd arg, it gives er
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[111], line 1
----> 1 np.random.randint(30,20,10)

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

File numpy\\random\\_bounded_integers.pyx:1425, in numpy.random._bounded_integer
s._rand_int32()

ValueError: low >= high
```

```
In [112…  np.random.randint(1,12,6)
```

```
Out[112…  array([ 2, 11,  1,  3, 10, 11])
```

```
In [114…  np.random.randint(1,100,(10,10)) #(low val,high val, (no. of row, no. of col))
```

```
Out[114…  array([[82, 94, 38, 60, 73, 41,  5, 89, 47, 38],
                 [92, 89, 55, 18, 34, 60, 89, 37, 46, 31],
                 [62, 80,  6, 20, 23, 25, 80, 71, 15, 11],
                 [98,  8, 87, 11, 24, 15, 36, 91, 75, 15],
                 [ 9, 93,  8, 83,  1, 64, 88, 39, 38, 86],
                 [65, 62, 34,  7, 64, 52, 44, 28, 98, 85],
                 [51, 84, 36, 64, 75, 10, 20, 85, 19, 80],
                 [55,  1, 91, 44, 21, 51, 37, 40, 53, 54],
                 [71, 90, 56, 68, 59, 22, 54, 73, 38, 70],
                 [ 7, 58, 39, 33, 21, 96, 90, 20, 77, 58]])
```

```
In [115…  np.random.randint(1,100,(12,12))
```

```
Out[115…  array([[49, 58, 20, 79, 89, 47, 34, 93, 68, 65, 18, 65],
                 [84, 41, 64,  5, 52, 46, 76, 63, 16, 34, 50, 32],
                 [35, 92,  1, 41, 69, 25,  4, 23, 60, 40, 85, 37],
                 [14, 31, 32, 41, 56, 13, 84, 16, 20, 25,  2, 69],
                 [ 1, 32, 49, 77, 31, 89, 83, 86, 65, 88, 18, 32],
                 [37, 30, 47, 70, 48, 77, 80, 53, 76, 95,  2,  4],
                 [91, 15, 68, 37, 94, 56, 98, 57, 77, 28, 74, 53],
                 [74, 16, 42, 75, 32, 90, 47, 17, 45, 91,  5, 54],
                 [80, 46, 48, 63, 40,  6, 66, 61, 21, 32, 25, 56],
                 [83, 52, 70, 56, 35, 98, 82, 96, 25, 19, 54, 68],
                 [47, 24, 42, 82, 54, 11, 83, 21,  1, 94, 10, 89],
                 [30, 43, 71, 15, 82, 47, 21, 20,  1, 90, 84, 95]])
```

```
In [117…  np.arange(1,13).reshape(3,4) #'arange' tells the which numbers should be include
```

```
Out[117…  array([[ 1,  2,  3,  4],
                 [ 5,  6,  7,  8],
                 [ 9, 10, 11, 12]])
```

```
In [120…  np.arange(1,37).reshape(6,6) #be carefull while defining (rows, cols) acc to the
```

```
Out[120…  array([[ 1,  2,  3,  4,  5,  6],
                 [ 7,  8,  9, 10, 11, 12],
                 [13, 14, 15, 16, 17, 18],
                 [19, 20, 21, 22, 23, 24],
                 [25, 26, 27, 28, 29, 30],
                 [31, 32, 33, 34, 35, 36]])
```

```
In [124…  np.arange(1,13).reshape(12,1)
```

```
Out[124…  array([[ 1],
                 [ 2],
                 [ 3],
                 [ 4],
                 [ 5],
                 [ 6],
                 [ 7],
                 [ 8],
                 [ 9],
                 [10],
                 [11],
                 [12]])
```

```
In [126…  b = np.random.randint(10,20,(5,4)) #random no. b/w 10-20 in a 5*4 matrix
          b
```

```
Out[126…  array([[10, 18, 12, 17],
                 [13, 12, 19, 19],
                 [14, 16, 15, 13],
                 [16, 10, 10, 11],
                 [10, 14, 13, 18]])
```

```
In [127…  type(b)
```

```
Out[127…  numpy.ndarray
```

```
In [128…  b
```

```
Out[128…  array([[10, 18, 12, 17],
                 [13, 12, 19, 19],
                 [14, 16, 15, 13],
                 [16, 10, 10, 11],
                 [10, 14, 13, 18]])
```

```
In [131…  b[1,2] #[row,col] it gives the value on it
```

```
Out[131…  19
```

```
In [136…  b[1,1]
```

```
Out[136…  12
```

```
In [137…  b[:]
```

```
Out[137…  array([[10, 18, 12, 17],
                 [13, 12, 19, 19],
                 [14, 16, 15, 13],
                 [16, 10, 10, 11],
                 [10, 14, 13, 18]])
```

```
In [139…  b[::-1] #reverse/flip the array list
```

```
Out[139…  array([[10, 14, 13, 18],
                 [16, 10, 10, 11],
                 [14, 16, 15, 13],
                 [13, 12, 19, 19],
                 [10, 18, 12, 17]])
```

```
In [141…  b[1:3] #prints the row at 1st and 2nd index
```

```
Out[141…  array([[13, 12, 19, 19],
                 [14, 16, 15, 13]])
```

```
In [142…  b
```

```
Out[142…  array([[10, 18, 12, 17],
                 [13, 12, 19, 19],
                 [14, 16, 15, 13],
                 [16, 10, 10, 11],
                 [10, 14, 13, 18]])
```

```
In [143…  b[1,-1]
```

```
Out[143…  19
```

```
In [144…  b[0:-2]
```

```
Out[144…  array([[10, 18, 12, 17],
                 [13, 12, 19, 19],
                 [14, 16, 15, 13]])
```

```
In [145…  b[0,-2]
```

```
Out[145…  12
```

```
In [147…  b[-5,-3]
```

Out[147…

```
18
```

In [148…

```
b[-3,-2]
```

Out[148…

```
15
```

In [149…

```
np.random.randint(10,20,(4,4))
```

Out[149…

```
array([[12, 13, 12, 14],
       [14, 12, 15, 10],
       [18, 17, 15, 15],
       [14, 17, 10, 11]])
```

In [150…

```
b
```

Out[150…

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

In [151…

```
b[-4:2]
```

Out[151…

```
array([[13, 12, 19, 19]])
```

In [152…

```
b[:]
```

Out[152…

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

operations

In [154…

```
a = np.random.randint(10,20,10)
a
```

Out[154…

```
array([19, 10, 17, 14, 19, 11, 13, 16, 13, 10])
```

In [155…

```
type(a)
```

Out[155…

```
numpy.ndarray
```

In [156…

```
id(a)
```

Out[156…

```
1823703106448
```

In [158…

```
arr1 = np.random.randint(0,100,(10,10))
arr1
```

```
Out[158…  array([[98, 68, 52,  0, 52, 21, 70,  1, 18, 72],
                 [92, 66, 60, 27, 22, 15, 11, 54,  9, 26],
                 [86, 67, 61,  4, 61, 60, 79, 67, 72, 54],
                 [57, 15, 53, 42, 89, 76, 18, 65, 98, 30],
                 [13, 96, 29, 16,  0, 65, 57, 19, 69, 94],
                 [77, 58, 88, 15, 17, 93,  7, 64,  5, 78],
                 [ 4, 52,  6, 45, 62, 62, 50,  6, 44, 81],
                 [61, 48, 59,  6, 85, 42, 85, 47, 13,  7],
                 [11, 76, 96, 50, 92, 55, 59, 88, 12, 59],
                 [ 0, 23, 28,  4, 59, 91, 69, 71, 57, 52]])
```

```
In [159…  arr
```

```
Out[159…  array([0, 1, 2, 3, 4, 5])
```

```
In [160…  arr[:4]
```

```
Out[160…  array([0, 1, 2, 3])
```

```
In [161…  arr1[:]
```

```
Out[161…  array([[98, 68, 52,  0, 52, 21, 70,  1, 18, 72],
                 [92, 66, 60, 27, 22, 15, 11, 54,  9, 26],
                 [86, 67, 61,  4, 61, 60, 79, 67, 72, 54],
                 [57, 15, 53, 42, 89, 76, 18, 65, 98, 30],
                 [13, 96, 29, 16,  0, 65, 57, 19, 69, 94],
                 [77, 58, 88, 15, 17, 93,  7, 64,  5, 78],
                 [ 4, 52,  6, 45, 62, 62, 50,  6, 44, 81],
                 [61, 48, 59,  6, 85, 42, 85, 47, 13,  7],
                 [11, 76, 96, 50, 92, 55, 59, 88, 12, 59],
                 [ 0, 23, 28,  4, 59, 91, 69, 71, 57, 52]])
```

```
In [163…  arr1[4,4]
```

```
Out[163…  0
```

```
In [164…  arr1[3,5]
```

```
Out[164…  76
```

```
In [171…  arr1[-7:-3]
```

```
Out[171…  array([[57, 15, 53, 42, 89, 76, 18, 65, 98, 30],
                 [13, 96, 29, 16,  0, 65, 57, 19, 69, 94],
                 [77, 58, 88, 15, 17, 93,  7, 64,  5, 78],
                 [ 4, 52,  6, 45, 62, 62, 50,  6, 44, 81]])
```

```
In [172…  arr1
```

```
Out[172…  array([[98, 68, 52,  0, 52, 21, 70,  1, 18, 72],
                 [92, 66, 60, 27, 22, 15, 11, 54,  9, 26],
                 [86, 67, 61,  4, 61, 60, 79, 67, 72, 54],
                 [57, 15, 53, 42, 89, 76, 18, 65, 98, 30],
                 [13, 96, 29, 16,  0, 65, 57, 19, 69, 94],
                 [77, 58, 88, 15, 17, 93,  7, 64,  5, 78],
                 [ 4, 52,  6, 45, 62, 62, 50,  6, 44, 81],
                 [61, 48, 59,  6, 85, 42, 85, 47, 13,  7],
                 [11, 76, 96, 50, 92, 55, 59, 88, 12, 59],
                 [ 0, 23, 28,  4, 59, 91, 69, 71, 57, 52]])
```

```
In [173…  arr1[::-1]
```

```
Out[173…  array([[ 0, 23, 28,  4, 59, 91, 69, 71, 57, 52],
                 [11, 76, 96, 50, 92, 55, 59, 88, 12, 59],
                 [61, 48, 59,  6, 85, 42, 85, 47, 13,  7],
                 [ 4, 52,  6, 45, 62, 62, 50,  6, 44, 81],
                 [77, 58, 88, 15, 17, 93,  7, 64,  5, 78],
                 [13, 96, 29, 16,  0, 65, 57, 19, 69, 94],
                 [57, 15, 53, 42, 89, 76, 18, 65, 98, 30],
                 [86, 67, 61,  4, 61, 60, 79, 67, 72, 54],
                 [92, 66, 60, 27, 22, 15, 11, 54,  9, 26],
                 [98, 68, 52,  0, 52, 21, 70,  1, 18, 72]])
```

```
In [176…  arr1[::-2]
```

```
Out[176…  array([[ 0, 23, 28,  4, 59, 91, 69, 71, 57, 52],
                 [61, 48, 59,  6, 85, 42, 85, 47, 13,  7],
                 [77, 58, 88, 15, 17, 93,  7, 64,  5, 78],
                 [57, 15, 53, 42, 89, 76, 18, 65, 98, 30],
                 [92, 66, 60, 27, 22, 15, 11, 54,  9, 26]])
```

```
In [178…  arr1[::-3] #skips 3 rows each time from last till first and print on every skip
```

```
Out[178…  array([[ 0, 23, 28,  4, 59, 91, 69, 71, 57, 52],
                 [ 4, 52,  6, 45, 62, 62, 50,  6, 44, 81],
                 [57, 15, 53, 42, 89, 76, 18, 65, 98, 30],
                 [98, 68, 52,  0, 52, 21, 70,  1, 18, 72]])
```

```
In [179…  arr
```

```
Out[179…  array([0, 1, 2, 3, 4, 5])
```

```
In [180…  arr.max()
```

```
Out[180…  5
```

```
In [181…  arr.min()
```

```
Out[181…  0
```

```
In [182…  arr.mean()
```

```
Out[182…  2.5
```

```
In [183…  arr.cumsum()
```

```
Out[183…  array([ 0,  1,  3,  6, 10, 15])
```

In [185…
```
arr.median()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[185], line 1
----> 1 arr.median()

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

In [193…
```
median(arr) #bcoz i imported * in next cell then executed this cell thats why i
```

Out[193…
```
2.5
```

In [190…
```
from numpy import *
a = array([1,2,3,4,9])
median(a)
```

Out[190…
```
3.0
```

In [188…
```
type(a)
```

Out[188…
```
numpy.ndarray
```

In [189…
```
len(a)
```

Out[189…
```
5
```

In [194…
```
#reshape()
```

In [195…
```
arr
```

Out[195…
```
array([0, 1, 2, 3, 4, 5])
```

In [202…
```
arr.reshape(6,1) #be careful while putting no. of rows & col,  it should be acc
```

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

In [203…
```
arr.reshape(1,6)
```

Out[203…
```
array([[0, 1, 2, 3, 4, 5]])
```

In [204…
```
arr.reshape(2,3)
```

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

In [206…
```
arr.reshape(3,2)
```

Out[206…
```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

In [207…

```
arr.reshape(2,3,order='C')
```

Out[207…

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

In [208…

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

Out[208…

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

In [209…

```
arr.reshape(2,3,order='A') # A almost give you c type output
```

Out[209…

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

Indexing

In [214…

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

Out[214…

```
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 [215…

```
row = 4
col = 5
print(row)
print(col)
```

```
4
5
```

In [216…

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

Out[216…

```
45
```

In [217…

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

Out[217…

```
45
```

In [218…

```
mat
```

```
Out[218…  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 [220…  mat[2,4] #[row,col]
```

```
Out[220…  24
```

```
In [221…  mat[6]
```

```
Out[221…  array([60, 61, 62, 63, 64, 65, 66, 67, 68, 69])
```

```
In [222…  mat[::-1]
```

```
Out[222…  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 [223…  mat
```

```
Out[223…  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 [224…  mat[-5:-3]
```

```
Out[224…  array([[50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
                 [60, 61, 62, 63, 64, 65, 66, 67, 68, 69]])
```

```
In [225…  mat[4:7]
```

```
Out[225…  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]])
```

```
In [228…  # With Slices
          mat[:,col] # col - 5 here, print full 5th col
```

```
Out[228…  array([ 5, 15, 25, 35, 45, 55, 65, 75, 85, 95])
```

```
In [229…  mat[row,:] # row - 4 here, print full 4th row
```

```
Out[229…  array([40, 41, 42, 43, 44, 45, 46, 47, 48, 49])
```

```
In [230…  mat
```

```
Out[230…  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 [231…  mat[:8]
```

```
Out[231…  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]])
```

```
In [232…  mat[::-2]
```

```
Out[232…  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 [233…  mat[1,8]
```

```
Out[233…  18
```

```
In [234…  mat[9,2]
```

```
Out[234…  92
```

```
In [237…  mat[:-1] #except last row all will pe printed
```

```
Out[237…  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]])
```

In [238…

```
mat[0]
```

Out[238…

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

In [240…

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

Out[240…

```
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 [243…

```
mat[0:10:3] #(start index : end index : step/skip index)
```

Out[243…

```
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 [244…

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

Out[244…

```
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 [245…

```
mat[::-5]
```

Out[245…

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

In [246…

```
mat[-4]
```

Out[246…

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

In [248…

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

Out[248…

```
array([[22, 23]])
```

In [249…

```
mat
```

Out[249…

```
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 [251…

```
mat[5:6,4:7]
```

Out[251…

```
array([[54, 55, 56]])
```

In [252…

```
mat[3:4,5:7]
```

```
Out[252…  array([[35, 36]])
```

Masking/filter

```
In [253…  mat
```

```
Out[253…  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 [254…  id(mat)
```

```
Out[254…  1823702098416
```

```
In [255…  type(mat)
```

```
Out[255…  numpy.ndarray
```

```
In [258…  mat<50 # it gives in boolean values which shows where it shows 'true' for values
```

```
Out[258…  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 [259…  mat[mat<50]
```

```
Out[259…  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 [261…  mat[mat>=50]
```

Out[261…

```
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 [263…

```python
mat[mat==50]
```

Out[263…

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

In [264…

```python
mat[mat!=50]
```

Out[264…

```
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 [265…

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

Out[265…

```
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 [266…

```python
a2 = mat[mat==50]
a2
```

Out[266…

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

In [267…

```python
a3 = mat[mat!=50]
a3
```

Out[267…

```
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])
```

python program to generate OTP

In [276…

```python
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: 2332
```

In [277…

```python
def wish():
    print('good even')
```

```
wish()

def wish():
    print('good even')
wish()

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

```
good even
good even
good even
```

In [278…

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

wish()

wish()
```

```
good even
good even
good even
```

In [281…

```
list1 = [1,2,2.77,'nit']
print(list1.pop())
```

```
nit
```

In [282…

```
list1
```

Out[282…

```
[1, 2, 2.77]
```

In [285…

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

Out[285…

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

TOday practise completed

In [ ]: