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

```
In [2]:   import sys
          sys.version
```

```
Out[2]:   '3.11.7 | packaged by Anaconda, Inc. | (main, Dec 15 2023, 18:05:47) [MSC v.191
          6 64 bit (AMD64)]'
```

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

```
Out[3]:   '1.26.4'
```

```
In [4]:   #create al list
          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)#multi dimensional array
```

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

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

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

```
In [9]:   np.arange(10)
```

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

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

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

```
In [11]:  np.arange(10,30,3)#10 to 30 at 3 step
```

```
Out[11]:  array([10, 13, 16, 19, 22, 25, 28])
```

```
In [12]:  np.arange(20,10)
```

```
Out[12]:  array([], dtype=int32)
```

```
In [13]:  np.arange(-20,10)#1st argument< 2nd argument.
```

```
Out[13]: 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 [14]:
```python
np.zeros(3)
```

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

In [15]:
```python
np.zeros(3, dtype=int)
```

```
Out[15]: array([0, 0, 0])
```

In [16]:
```python
z=np.zeros(5)
z
```

```
Out[16]: array([0., 0., 0., 0., 0.])
```

In [17]:
```python
np.zeros((2,2))#2d array
```

```
Out[17]: array([[0., 0.],
                [0., 0.]])
```

In [18]:
```python
np.zeros((3,3),dtype=int)#1st no to rowws,2nd is colum
```

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

In [19]:
```python
nd=np.zeros((5,9),dtype=int)
nd
```

```
Out[19]: 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 [20]:
```python
np.ones(3)
```

```
Out[20]: array([1., 1., 1.])
```

In [21]:
```python
np.ones((3),dtype=int)
```

```
Out[21]: array([1, 1, 1])
```

In [22]:
```python
nd1=np.ones((10,10),dtype=int)
nd1
```

```
Out[22]: 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 [23]: `#np.three(3) numpy can use two numbe [0 and 1]`

4th July

In [24]: `np.random.rand(2)`

Out[24]: `array([0.41407274, 0.53269466])`

In [25]: `np.random.rand(3,2)`

Out[25]:
```
array([[0.29900016, 0.31191333],
       [0.03188333, 0.69596862],
       [0.12670352, 0.68585322]])
```

In [26]: `np.random.randint(3)`

Out[26]: `1`

In [27]: `np.random.randint(2,10,3)# between 2 to 10 3 number`

Out[27]: `array([7, 5, 5])`

In [28]: `np.random.randint(-30,20,10)`

Out[28]: `array([  2, -25,  -6, -29,   7,   0, -10,   6, -12, -17])`

In [29]: `np.random.randint(10,40,[10,10])`

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

In [30]:
```
m=np.random.randint(10,45,[10,10])
m
```

Out[30]:
```
array([[16, 38, 12, 28, 12, 11, 29, 10, 24, 35],
       [21, 16, 38, 13, 16, 17, 28, 20, 10, 18],
       [23, 32, 12, 26, 27, 43, 12, 38, 31, 13],
       [21, 12, 17, 12, 37, 14, 35, 39, 42, 43],
       [41, 17, 26, 25, 34, 20, 10, 40, 19, 43],
       [15, 22, 37, 24, 14, 12, 43, 30, 14, 37],
       [20, 24, 25, 19, 13, 20, 44, 11, 35, 13],
       [27, 22, 29, 41, 38, 20, 23, 23, 30, 36],
       [44, 36, 15, 28, 27, 31, 23, 16, 40, 14],
       [35, 14, 22, 33, 24, 40, 32, 33, 13, 32]])
```

In [31]: `a1=np.array([1,2,3,4,5,6])`

```
In [32]: arr.reshape(6,1)
```

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

```
In [33]: b=np.random.randint(10,20,(5,4))
         b
```

```
Out[33]: array([[19, 10, 19, 13],
                [14, 13, 15, 18],
                [18, 10, 18, 13],
                [15, 17, 13, 19],
                [11, 11, 14, 15]])
```

```
In [34]: s=np.arange(1,21).reshape(4,5)# reshape to 1 to 21 arrange in matrix
         s
```

```
Out[34]: array([[ 1,  2,  3,  4,  5],
                [ 6,  7,  8,  9, 10],
                [11, 12, 13, 14, 15],
                [16, 17, 18, 19, 20]])
```

```
In [35]: s1=np.random.randint(10,21,[4,5])
         s1
```

```
Out[35]: array([[13, 14, 10, 16, 12],
                [11, 20, 20, 12, 17],
                [14, 14, 15, 16, 12],
                [10, 15, 17, 11, 19]])
```

```
In [36]: type(s)
```

```
Out[36]: numpy.ndarray
```

```
In [37]: # to create an numpy array
         n=np.array([21,3,5,35,45,65,46])
         n
```

```
Out[37]: array([21,  3,  5, 35, 45, 65, 46])
```

```
In [ ]:
```

```
In [38]: #slicing in matrix
```

```
In [39]: b[:]
```

```
Out[39]: array([[19, 10, 19, 13],
                [14, 13, 15, 18],
                [18, 10, 18, 13],
                [15, 17, 13, 19],
                [11, 11, 14, 15]])
```

```
In [40]: b[1:4]
```

Out[40]: array([[14, 13, 15, 18],
                 [18, 10, 18, 13],
                 [15, 17, 13, 19]])

In [41]: b[-1:]

Out[41]: array([[11, 11, 14, 15]])

In [42]: b[-1:]

Out[42]: array([[11, 11, 14, 15]])

In [43]: b[:-1]

Out[43]: array([[19, 10, 19, 13],
                 [14, 13, 15, 18],
                 [18, 10, 18, 13],
                 [15, 17, 13, 19]])

In [44]: #

In [45]: #: =is row information .[,]=pritn specific no,not row

In [46]: b[1,2]

Out[46]: 15

In [47]: b[1,-1]

Out[47]: 18

In [48]: b

Out[48]: array([[19, 10, 19, 13],
                 [14, 13, 15, 18],
                 [18, 10, 18, 13],
                 [15, 17, 13, 19],
                 [11, 11, 14, 15]])

In [49]: b[-2,-2]

Out[49]: 13

In [50]: b[2:4,(1,2)]

Out[50]: array([[10, 18],
                 [17, 13]])

In [51]: #numpy operation

In [52]: a1=np.array([0, 1, 2, 3, 4, 5])

In [53]: a1.max()

Out[53]: 5

In [54]: m

```
Out[54]: array([[16, 38, 12, 28, 12, 11, 29, 10, 24, 35],
                [21, 16, 38, 13, 16, 17, 28, 20, 10, 18],
                [23, 32, 12, 26, 27, 43, 12, 38, 31, 13],
                [21, 12, 17, 12, 37, 14, 35, 39, 42, 43],
                [41, 17, 26, 25, 34, 20, 10, 40, 19, 43],
                [15, 22, 37, 24, 14, 12, 43, 30, 14, 37],
                [20, 24, 25, 19, 13, 20, 44, 11, 35, 13],
                [27, 22, 29, 41, 38, 20, 23, 23, 30, 36],
                [44, 36, 15, 28, 27, 31, 23, 16, 40, 14],
                [35, 14, 22, 33, 24, 40, 32, 33, 13, 32]])
```

```
In [55]: m[::-1]# reverse ,-2 to2step reverse
```

```
Out[55]: array([[35, 14, 22, 33, 24, 40, 32, 33, 13, 32],
                [44, 36, 15, 28, 27, 31, 23, 16, 40, 14],
                [27, 22, 29, 41, 38, 20, 23, 23, 30, 36],
                [20, 24, 25, 19, 13, 20, 44, 11, 35, 13],
                [15, 22, 37, 24, 14, 12, 43, 30, 14, 37],
                [41, 17, 26, 25, 34, 20, 10, 40, 19, 43],
                [21, 12, 17, 12, 37, 14, 35, 39, 42, 43],
                [23, 32, 12, 26, 27, 43, 12, 38, 31, 13],
                [21, 16, 38, 13, 16, 17, 28, 20, 10, 18],
                [16, 38, 12, 28, 12, 11, 29, 10, 24, 35]])
```

```
In [56]: m[2:6,2:4]
```

```
Out[56]: array([[12, 26],
                [17, 12],
                [26, 25],
                [37, 24]])
```

```
In [57]: a2=np.random.randint(0,100,[10,10])
         a2
```

```
Out[57]: array([[57, 35, 96, 55, 89, 45, 89, 66, 73,  4],
                [ 7, 90, 78, 65, 54, 20, 31,  6, 77, 97],
                [ 3, 77, 59, 19, 95, 69, 81, 65, 87, 66],
                [85, 42, 78, 39, 45, 83, 20, 34,  2, 56],
                [25, 99, 68, 98, 66, 72, 18, 70,  8, 41],
                [24, 45, 40, 84,  6, 48, 79, 74, 31, 11],
                [36, 55,  9,  1, 68, 75, 63, 90, 36, 97],
                [85, 47, 36, 26, 50,  4, 76, 53, 90, 60],
                [76, 98, 96, 48, 95, 69, 88, 57, 99,  0],
                [64, 90, 58, 13, 68, 90, 65,  5, 28, 52]])
```

```
In [58]: a2[1:3,[3,4]]
```

```
Out[58]: array([[65, 54],
                [19, 95]])
```

```
In [59]: a2[5:7,(-4,-3)]
```

```
Out[59]: array([[79, 74],
                [63, 90]])
```

```
In [ ]:
```

```
In [60]: #masking / filter
```

In [61]: `m>30`

Out[61]: array([[False, True, False, False, False, False, False, False, False,
                  True],
                [False, False, True, False, False, False, False, False, False,
                 False],
                [False, True, False, False, False, True, False, True, True,
                 False],
                [False, False, False, False, True, False, True, True, True,
                  True],
                [True, False, False, False, True, False, False, True, False,
                  True],
                [False, False, True, False, False, False, True, False, False,
                  True],
                [False, False, False, False, False, False, True, False, True,
                 False],
                [False, False, False, True, True, False, False, False, False,
                  True],
                [True, True, False, False, False, True, False, False, True,
                 False],
                [True, False, False, True, False, True, True, True, False,
                  True]])

In [62]: `m[m>40]`

Out[62]: array([43, 42, 43, 41, 43, 43, 44, 41, 44])

In [63]: `#cline`

In [64]: `#indexing`

In [65]: ```
k=np.arange(0,100).reshape(10,10)
k
```

Out[65]: 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 [66]: `k[:]`

Out[66]: 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 [67]: `k[1:-3]`

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

In [68]: `k[3]`

Out[68]: `array([30, 31, 32, 33, 34, 35, 36, 37, 38, 39])`

In [69]: `col=8`

In [70]: `k[:,8]`

Out[70]: `array([ 8, 18, 28, 38, 48, 58, 68, 78, 88, 98])`

In [71]: `k[5,:]`

Out[71]: `array([50, 51, 52, 53, 54, 55, 56, 57, 58, 59])`

In [72]: `k.max()`

Out[72]: `99`

In [73]: `k.mean()`

Out[73]: `49.5`

# 6th July

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: