```
#Numpy is a library which handles nd array(multidimensional aray)

2.numpy holds maths+stattistic+linearalgebra+datastructures

3.when we work with nuber,images,text,speech-->every data should converted to array before we
```

```
import numpy as np
```

```
np.arange(15)
```

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

```
np.random.randint(5,9)
```

```
8
```

Archana Kapu
11:22 PM Today

prints the random value between 5 and 8

```
np.random.randint(20,30,10)
```

```
array([29, 22, 28, 20, 27, 24, 28, 24, 27, 24])
```

Archana Kapu
11:23 PM Today

generates10 randon integers between 20 and 29

```
np.random.randint(10,40,(10,10))
```

```
array([[38, 17, 26, 21, 35, 31, 31, 24, 23, 17],
       [27, 16, 20, 38, 20, 18, 20, 32, 32, 38],
       [21, 10, 20, 27, 38, 33, 18, 24, 34, 28],
       [14, 32, 11, 20, 36, 31, 14, 14, 18, 30],
       [10, 27, 32, 39, 11, 11, 30, 32, 30, 34],
       [38, 12, 20, 16, 39, 16, 36, 14, 32, 23],
       [16, 28, 33, 20, 18, 35, 31, 21, 18, 20],
       [37, 28, 23, 30, 23, 32, 36, 38, 10, 37],
       [29, 11, 29, 16, 16, 20, 35, 34, 37, 10],
       [34, 25, 18, 18, 31, 35, 29, 22, 23, 33]])
```

Archana Kapu
11:26 PM Today

this syntax means generate integers between 10 and 40 in 10X10 matrix format

```
np.random.randint(1,100,(12,12))
```

```
array([[30, 12, 90, 36, 62, 94, 45, 13, 35, 21, 50, 72],
       [99, 27, 17, 57, 83, 30, 49, 16, 66, 60, 60, 90],
       [95, 17, 30, 54, 28, 29,  5, 72,  4, 68, 45,  3],
       [48, 76, 24, 28, 65, 44, 30, 83, 71, 23,  1, 90],
       [22, 97, 30, 11, 78, 34,  7, 84, 46, 94, 88, 83],
       [12,  3, 88, 76, 46, 51, 42, 85, 95, 30, 59, 14],
       [97, 80, 60, 78, 20, 68, 83, 99, 30, 58, 66, 69],
       [53,  1, 25, 76, 47, 67, 80, 71, 44, 47, 51, 67],
       [79, 26, 96, 39, 34, 77, 97, 91, 84, 37, 18,  1],
       [ 1, 76, 84, 93, 27,  2, 69, 23, 72, 15, 87, 79],
       [14, 58, 94, 74, 76, 46, 38, 79, 82,  3, 57, 15],
       [94, 14, 29,  6, 42, 46, 69, 85, 13, 14,  7, 89]])
```

Double-click (or enter) to edit

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

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

Archana Kapu
11:31 PM Today

this syntax means,generate integers beteen 1 and 12 in 3X4 matrix format.

```
np.arange(1,13).reshape(5,4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-15-258db1197ee0> in <cell line: 0>()
----> 1 np.arange(1,13).reshape(5,4)

ValueError: cannot reshape array of size 12 into shape (5,4)
```

Archana Kapu
11:32 PM Today

Expected error

Next steps: Explain error

```
np.arange(1,13).reshape(12,1)
```

```
array([[ 1],
       [ 2],
       [ 3],
       [ 4],
       [ 5],
       [ 6],
       [ 7],
```

Archana Kapu
11:33 PM Today

here genrating the integers between 1 and 12 in 12X1 matrix format

```
       [ 8],
       [ 9],
       [10],
       [11],
       [12]])
```

```
np.arange(1,13).reshape(6,2)
```

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

```
np.arange(1,13).reshape(12,1)
```

Archana Kapu
11:35 PM Today

12x1 matrix

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

```
np.arange(1,13).reshape(1,12)
```

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

slicing in matrix

```
bp=np.random.randint(10,20,(5,4))
```

```
bp
```

```
array([[10, 17, 16, 10],
       [10, 17, 10, 13],
       [13, 15, 10, 10],
       [16, 16, 19, 16],
       [15, 15, 17, 17]])
```

```
type(bp)
```

```
numpy.ndarray
```

```
bp[:]
```

Archana Kapu
11:39 PM Today

empty slice writen all elements in array

```
array([[10, 17, 16, 10],
       [10, 17, 10, 13],
       [13, 15, 10, 10],
       [16, 16, 19, 16],
       [15, 15, 17, 17]])
```

```
bp
```

```
array([[10, 17, 16, 10],
       [10, 17, 10, 13],
       [13, 15, 10, 10],
       [16, 16, 19, 16],
       [15, 15, 17, 17]])
```

```
array([[10, 17, 16, 10],0
       [10, 17, 10, 13],1
       [13, 15, 10, 10],2
       [16, 16, 19, 16],3
       [15, 15, 17, 17]])4
```

<-3-1)

```
bp[1:3]
```

Archana Kapu
11:46 PM Today

array index starts with zero[1:3] means 1 -2

```
array([[10, 17, 10, 13],
       [13, 15, 10, 10]])
```

```
bp[1:2]
```

```
array([[10, 17, 10, 13]])
```

```
bp
```

```
array([[10, 17, 16, 10],
       [10, 17, 10, 13],
       [13, 15, 10, 10],
       [16, 16, 19, 16],
       [15, 15, 17, 17]])
```

```
bp[1,2]
```

```
10
```

Archana Kapu
11:52 PM Today

1st row 2nd col value

```
#bp[1:4]---represents entire row
#bp[1,4]---represents specified value or element of assigned or declared row
```

```
bp[1,3]
```

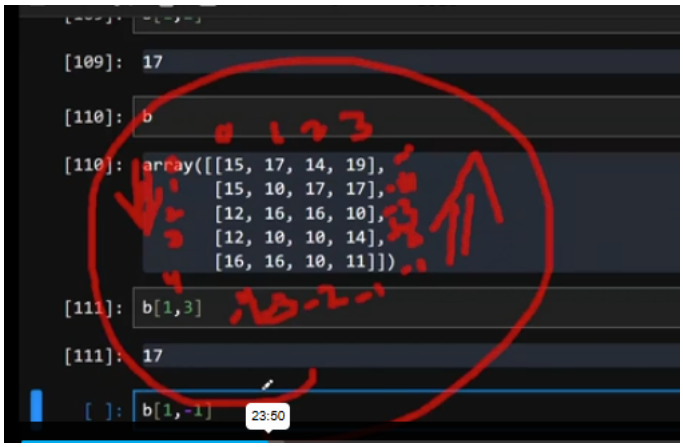
```
13
```

-->accoridng to below image-->here row wise index starst with top to dow 0-4,negative row wise index down to up -1 to -4.

-->here column wise index starst with left to right in positie numbers,and negative index starts from right to left -1 to -4

Archana Kapu
12:03 AM Today

-->here row wise index starst with top to dow 0-4,negative row wise index down to up -1 to -4.
-->here column wise index starst with left to right in positie numbers,and negative index starts from right to left -1 to -4




```
bp
```

```
array([[10, 17, 16, 10],
       [10, 17, 10, 13],
       [13, 15, 10, 10],
       [16, 16, 19, 16],
       [15, 15, 17, 17]])
```

```
bp[1,-1]
```

```
13
```

```
bp[2:3]
```

```
array([[13, 15, 10, 10]])
```

Archana Kapu
12:10 AM Today

0X-2-1==0X-3

```
bp[0:-2]
```

```
array([[10, 17, 16, 10],
       [10, 17, 10, 13],
       [13, 15, 10, 10]])
```

Archana Kapu
12:12 AM Today

0th row and 2nd colun value

```
bp[0,2]
```

```
16
```

```
bp[-5,-3]
```

```
17
```

```
bp[-4,2]
```

```
10
```

operations

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

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

```
id(ab)
```

```
136875089042800
```

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

```
arr=np.array(my_list)
```

```
arr
```

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

```
arr2=np.random.randint(0,100,(10,10))
```

```
arr2
```

```
array([[10,  0, 67, 98,  4, 90, 61, 26, 78, 80],
       [99, 98, 33, 88, 82, 21, 53, 85, 56, 66],
       [73, 86, 23, 69, 93, 34, 69, 23, 80, 25],
       [57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [40, 87, 30, 50, 89, 13, 84, 41, 47, 11],
       [ 8,  6, 85, 78,  6, 21, 10, 59, 89, 93],
       [39,  1, 79, 41, 28, 35, 35, 83, 34, 49],
       [69, 92, 98, 96, 28, 50, 65, 69, 24, 36],
       [90,  0, 88, 31, 62, 86, 48,  7, 24, 13],
       [22, 49, 75, 88, 72, 72, 20, 54, 44, 79]])
```

```
arr
```

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

```
arr[:4]
```

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

```
arr2[:]
```

```
array([[10,  0, 67, 98,  4, 90, 61, 26, 78, 80],
       [99, 98, 33, 88, 82, 21, 53, 85, 56, 66],
       [73, 86, 23, 69, 93, 34, 69, 23, 80, 25],
       [57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [40, 87, 30, 50, 89, 13, 84, 41, 47, 11],
       [ 8,  6, 85, 78,  6, 21, 10, 59, 89, 93],
       [39,  1, 79, 41, 28, 35, 35, 83, 34, 49],
       [69, 92, 98, 96, 28, 50, 65, 69, 24, 36],
       [90,  0, 88, 31, 62, 86, 48,  7, 24, 13],
       [22, 49, 75, 88, 72, 72, 20, 54, 44, 79]])
```

```
arr2[0:5]
```

```
array([[10,  0, 67, 98,  4, 90, 61, 26, 78, 80],
       [99, 98, 33, 88, 82, 21, 53, 85, 56, 66],
       [73, 86, 23, 69, 93, 34, 69, 23, 80, 25],
       [57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [40, 87, 30, 50, 89, 13, 84, 41, 47, 11]])
```

```
arr2
```

```
array([[10,  0, 67, 98,  4, 90, 61, 26, 78, 80],
       [99, 98, 33, 88, 82, 21, 53, 85, 56, 66],
       [73, 86, 23, 69, 93, 34, 69, 23, 80, 25],
       [57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [40, 87, 30, 50, 89, 13, 84, 41, 47, 11],
       [ 8,  6, 85, 78,  6, 21, 10, 59, 89, 93],
       [39,  1, 79, 41, 28, 35, 35, 83, 34, 49],
       [69, 92, 98, 96, 28, 50, 65, 69, 24, 36],
       [90,  0, 88, 31, 62, 86, 48,  7, 24, 13],
       [22, 49, 75, 88, 72, 72, 20, 54, 44, 79]])
```

Archana Kapu
12:23 AM Today

1st row and 4th column value

```
arr2[1,4]
```

```
82
```

Archana Kapu
12:25 AM Today

reverse te matrix

```
arr2[::-1]
```

```
array([[22, 49, 75, 88, 72, 72, 20, 54, 44, 79],
       [90,  0, 88, 31, 62, 86, 48,  7, 24, 13],
       [69, 92, 98, 96, 28, 50, 65, 69, 24, 36],
       [39,  1, 79, 41, 28, 35, 35, 83, 34, 49],
       [ 8,  6, 85, 78,  6, 21, 10, 59, 89, 93],
       [40, 87, 30, 50, 89, 13, 84, 41, 47, 11],
       [57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [73, 86, 23, 69, 93, 34, 69, 23, 80, 25],
       [99, 98, 33, 88, 82, 21, 53, 85, 56, 66],
       [10,  0, 67, 98,  4, 90, 61, 26, 78, 80]])
```

```
arr2
```

```
array([[10,  0, 67, 98,  4, 90, 61, 26, 78, 80],
       [99, 98, 33, 88, 82, 21, 53, 85, 56, 66],
       [73, 86, 23, 69, 93, 34, 69, 23, 80, 25],
       [57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [40, 87, 30, 50, 89, 13, 84, 41, 47, 11],
       [ 8,  6, 85, 78,  6, 21, 10, 59, 89, 93],
       [39,  1, 79, 41, 28, 35, 35, 83, 34, 49],
       [69, 92, 98, 96, 28, 50, 65, 69, 24, 36],
       [90,  0, 88, 31, 62, 86, 48,  7, 24, 13],
       [22, 49, 75, 88, 72, 72, 20, 54, 44, 79]])
```

```
arr2[::-2]
```

```
array([[22, 49, 75, 88, 72, 72, 20, 54, 44, 79],
       [69, 92, 98, 96, 28, 50, 65, 69, 24, 36],
       [ 8,  6, 85, 78,  6, 21, 10, 59, 89, 93],
       [57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [99, 98, 33, 88, 82, 21, 53, 85, 56, 66]])
```

```
arr2[::-3]
```

```
array([[22, 49, 75, 88, 72, 72, 20, 54, 44, 79],
       [39,  1, 79, 41, 28, 35, 35, 83, 34, 49],
       [57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [10,  0, 67, 98,  4, 90, 61, 26, 78, 80]])
```

```
arr2[:-3]
```

```
array([[10,  0, 67, 98,  4, 90, 61, 26, 78, 80],
       [99, 98, 33, 88, 82, 21, 53, 85, 56, 66],
       [73, 86, 23, 69, 93, 34, 69, 23, 80, 25],
       [57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [40, 87, 30, 50, 89, 13, 84, 41, 47, 11],
       [ 8,  6, 85, 78,  6, 21, 10, 59, 89, 93],
       [39,  1, 79, 41, 28, 35, 35, 83, 34, 49]])
```

```
arr2
```

```
array([[10,  0, 67, 98,  4, 90, 61, 26, 78, 80],
       [99, 98, 33, 88, 82, 21, 53, 85, 56, 66],
       [73, 86, 23, 69, 93, 34, 69, 23, 80, 25],
       [57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [40, 87, 30, 50, 89, 13, 84, 41, 47, 11],
       [ 8,  6, 85, 78,  6, 21, 10, 59, 89, 93],
       [39,  1, 79, 41, 28, 35, 35, 83, 34, 49],
       [69, 92, 98, 96, 28, 50, 65, 69, 24, 36],
       [90,  0, 88, 31, 62, 86, 48,  7, 24, 13],
       [22, 49, 75, 88, 72, 72, 20, 54, 44, 79]])
```

```
arr2[3:]
```

```
array([[57, 58, 96, 67, 17, 28, 89, 74, 97, 26],
       [40, 87, 30, 50, 89, 13, 84, 41, 47, 11],
       [ 8,  6, 85, 78,  6, 21, 10, 59, 89, 93],
       [39,  1, 79, 41, 28, 35, 35, 83, 34, 49],
       [69, 92, 98, 96, 28, 50, 65, 69, 24, 36],
       [90,  0, 88, 31, 62, 86, 48,  7, 24, 13],
       [22, 49, 75, 88, 72, 72, 20, 54, 44, 79]])
```

```
arr
```

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

```
arr.max()
```

```
5
```

Archana Kapu
12:35 AM Today

0+1+2+3+4+5/6=2.5

```
arr.min()
```

```
0
```

```
arr.mean()
```

```
2.5
```

```
arr.median()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-78-e8f6ca672427> in <cell line: 0>()
----> 1 arr.median()

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

Next steps: Explain error

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

```
3.0
```

reshaping as 3 formats

1.ctype

2.arbitary

3.fortan

```
arr
```

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

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

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

```
arr.reshape(3,2,order='c')#Print elemnets with ctype
```

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

```
arr.reshape(2,3,order='F')#Print elemnets with fortan
```

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

indexing

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

```
mat
```

```
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]])
```

```
mat.max()
```

```
99
```

```
mat.min()
```

```
0
```

```
mat.mean()
```

```
49.5
```

Archana Kapu
12:48 AM Today

4th row with 5th col value

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

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

```
45
```

Archana Kapu
12:49 AM Today

we can access 6th row

Archana Kapu
12:50 AM Today

default it represents row

```
mat[6]
```

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

```
col=6
```

Archana Kapu
12:52 AM Today

as row value is 4 ,from 4th row to entire

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

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

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

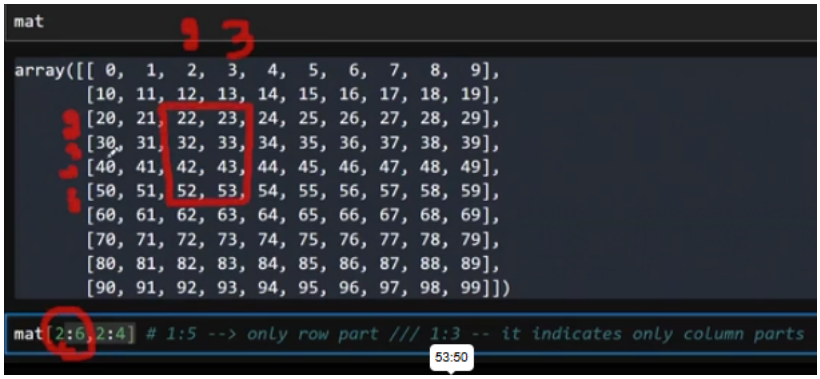
```
array([[40, 41, 42, 43, 44, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
       [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
       [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
       [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
       [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

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

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

```
mat[:,5]
```

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




```
mat[2:6,2:4]#1:5 only rows part--1:3 represents column part
```

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

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

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

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

```
array([[22]])
```

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

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

Masking

```
mat#we also called as filter
```

```
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]])
```

```
id(mat)
```

```
136875086035248
```

```
mat[mat<50]
```

```
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])
```

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

```
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])
```

```
mat[mat>50]
```

```
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])
```

```
mat[mat>50]
```

```
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])
```

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

```
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])
```

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

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

```
a2=mat[mat>60]
a2
```

```
array([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])
```

```
a3=mat[mat>=60]
a3
```

```
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])
```