

```
In [10]: import numpy as np
```

```
In [11]: import sys  
sys.version
```

```
Out[11]: '3.13.5 | packaged by Anaconda, Inc. | (main, Jun 12 2025, 16:37:03) [MSC v.192  
9 64 bit (AMD64)]'
```

```
In [12]: np.__version__
```

```
Out[12]: '2.3.1'
```

```
In [13]: # create list  
  
my_list = [0,1,2,3,4,5]  
my_list
```

```
Out[13]: [0, 1, 2, 3, 4, 5]
```

```
In [14]: type(my_list)
```

```
Out[14]: list
```

```
In [15]: np. # we learn important function
```

```
Cell In[15], line 1  
    np. # we learn important function  
      ^  
SyntaxError: invalid syntax
```

```
In [16]: cell in[57], line 1
```

```
Cell In[16], line 1  
    cell in[57], line 1  
      ^  
SyntaxError: invalid syntax
```

```
In [17]: np.arange(15)
```

```
Out[17]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

```
In [18]: np.arange(3.0)
```

```
Out[18]: array([0., 1., 2.])
```

```
In [19]: np.arange(10)
```

```
Out[19]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [20]: np.arange(0,5)
```

```
Out[20]: array([0, 1, 2, 3, 4])
```

```
In [21]: np.arange(20,10) # 1st arg < 2nd arg
```

```
Out[21]: array([], dtype=int64)
```

```
In [22]: np.arange(-20,-10)
```

```
Out[22]: array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11])
```

```
In [23]: arr = np.array(my_list)
arr
```

```
Out[23]: array([0, 1, 2, 3, 4, 5])
```

```
In [24]: type(arr)
```

```
Out[24]: numpy.ndarray
```

```
In [25]: print(type(arr))
print(type(my_list))

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

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

```
Out[26]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [27]: np.arange(10,20)
```

```
Out[27]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
In [28]: np.arange(10,50,5)
```

```
Out[28]: array([10, 15, 20, 25, 30, 35, 40, 45])
```

```
In [29]: np.arange(10,30,3)
```

```
Out[29]: array([10, 13, 16, 19, 22, 25, 28])
```

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

```
-----
TypeError                                Traceback (most recent call last)
Cell In[30], line 1
----> 1 np.arange(10,30,30,3)

TypeError: Cannot interpret '3' as a data type
```

```
In [31]: np.arange(20,10)
```

```
Out[31]: array([], dtype=int64)
```

```
In [32]: np.arange(20,8)
```

```
Out[32]: array([], dtype=int64)
```

```
In [33]: np.arange(8,20)
```

```
Out[33]: array([ 8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
In [34]: np.arange(-30,20)
```

```
Out[34]: array([-30, -29, -28, -27, -26, -25, -24, -23, -22, -21, -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, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
In [35]: n=np.arange(-20,8)
n
```

```
Out[35]: array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10, -9, -8,
               -7, -6, -5, -4, -3, -2, -1,  0,  1,  2,  3,  4,  5,
               6,  7])
```

```
In [36]: np.arange(10,10)
```

```
Out[36]: array([], dtype=int64)
```

```
In [37]: np.arange()
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[37], line 1
----> 1 np.arange()

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

```
In [38]: np.arange(10,30,5)
```

```
Out[38]: array([10, 15, 20, 25])
```

```
In [39]: np.arange(0,10,3)
```

```
Out[39]: array([0, 3, 6, 9])
```

```
In [40]: np.arange(10,30,5,8)
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[40], line 1
----> 1 np.arange(10,30,5,8)

TypeError: Cannot interpret '8' as a data type
```

```
In [41]: np.zeros(3)
```

```
Out[41]: array([0., 0., 0.])
```

```
In [42]: np.zeros(5, dtype=int)
```

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

```
In [43]: np.zeros((2,2))
```

```
Out[43]: array([[0., 0.],
               [0., 0.]])
```

```
In [44]: zero = np.zeros([2,2])
print(zero)
print(type(zero))
```

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

```
In [45]: zero = np.zeros((5,9),dtype = int)
         print(zero)
```

```
[[0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0]
 [0 0 0 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.zeros((3,3))
```

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

```
In [47]: np.ones(3,dtype=int)
```

```
Out[47]: array([1, 1, 1])
```

```
In [48]: np.ones((10,30))
```

```
Out[48]: 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., 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., 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.],
               [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 [49]: np.zeros((5,10)) # by default large -- will give row & 2nd arg - columns
```

```
Out[49]: 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.]])
```

```
In [50]: n = (6,7)
         n1 = (6,8)
         print(np.zeros(n1)) # parameter tuning
```

```
[[0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]
 [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 [51]: `print(np.zeros(n,dtype=int)) # hyperparameter tuning`

```
[[0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 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 [52]: `n`

Out[52]: (6, 7)

In [53]: `n1`

Out[53]: (6, 8)

In [54]: `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.]]
```

In [55]: `np.ones(3)`

Out[55]: `array([1., 1., 1.])`

In [56]: `np.ones(4, dtype=int)`

Out[56]: `array([1, 1, 1, 1])`

In [57]: `np.ones(4)`

Out[57]: `array([1., 1., 1., 1.])`

In [58]: `n`

Out[58]: (6, 7)

In [59]: `np.ones(n)`

Out[59]: `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 [60]: `np.ones((5,4),dtype=int) # by default 5- rows & 4 - columns`

```
Out[60]: array([[1, 1, 1, 1],
               [1, 1, 1, 1],
               [1, 1, 1, 1],
               [1, 1, 1, 1],
               [1, 1, 1, 1]])
```

```
In [61]: np.
```

```
Cell In[61], line 1
    np.
    ^
SyntaxError: invalid syntax
```

```
In [62]: np.twos((2,3))
```

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[62], line 1
----> 1 np.twos((2,3))

File ~\AppData\Roaming\Python\Python313\site-packages\numpy\__init__.py:808, in _
_getattr__(attr)
    805     import numpy.char as char
    806     return char.chararray
--> 808 raise AttributeError(f"module {__name__!r} has no attribute {attr!r}")

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

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

```
In [ ]: np.ones(2)
```

```
In [ ]: np.ones((2,4))
```

4th july

```
In [63]: range(5)
```

```
Out[63]: range(0, 5)
```

```
In [64]: r = range(5)
         r
```

```
Out[64]: range(0, 5)
```

```
In [65]: for i in r:
         print(i)
```

```
0
1
2
3
4
```

```
In [66]: list(range(5))
```

```
Out[66]: [0, 1, 2, 3, 4]
```

```
In [67]: list(range(1,10))
```

```
Out[67]: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [68]: list(range(1,10,3))
```

```
Out[68]: [1, 4, 7]
```

```
In [69]: y = list(range(12))
         y
```

```
Out[69]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

```
In [70]: rand(3,2)
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[70], line 1
----> 1 rand(3,2)

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

```
In [71]: rand(3,2)
         random.rand(3,2)
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[71], line 1
----> 1 rand(3,2)
      2 random.rand(3,2)

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

```
In [72]: np.random.rand(5)
```

```
Out[72]: array([0.3386323 , 0.52433801, 0.53216098, 0.96069076, 0.6305547 ])
```

```
In [73]: np.rand(4)
```

```
-----
AttributeError                            Traceback (most recent call last)
Cell In[73], line 1
----> 1 np.rand(4)

File ~\AppData\Roaming\Python\Python313\site-packages\numpy\__init__.py:808, in _
_getattr__(attr)
      805     import numpy.char as char
      806     return char.chararray
--> 808 raise AttributeError(f"module {__name__!r} has no attribute {attr!r}")

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

```
In [74]: np.random.rand(2,4)
```

```
Out[74]: array([[0.09304355, 0.2985629 , 0.32960991, 0.79118428],
                [0.64621235, 0.90325154, 0.66487276, 0.70402237]])
```

```
In [75]: np.random.randint(2,4)
```

Out[75]: 2

In [76]: `np.random.randint(2)`

Out[76]: 0

In [77]: `np.random.rand(2)`

Out[77]: `array([0.45780029, 0.04447832])`

In [78]: `np.random.randint(2,20) # 2nd argument is exclusive`

Out[78]: 12

In [79]: `np.random.randint(0,1)`

Out[79]: 0

In [80]: `np.random.randint(10,20,5)`

Out[80]: `array([18, 11, 15, 18, 12], dtype=int32)`

In [81]: `np.random.randint(1,6,4)`

Out[81]: `array([1, 2, 3, 3], dtype=int32)`

In [82]: `np.random.rand(3)`

Out[82]: `array([0.27973107, 0.8999751 , 0.82976077])`

In [83]: `np.random.randint(1)`

Out[83]: 0

In [84]: `np.random.randint(30,20,10)`

```
-----  
ValueError                                Traceback (most recent call last)  
Cell In[84], line 1  
----> 1 np.random.randint(30,20,10)  
  
File numpy/random/mtrand.pyx:794, in numpy.random.mtrand.RandomState.randint()  
  
File numpy/random/_bounded_integers.pyx:2885, in numpy.random._bounded_integers._rand_int32()  
  
ValueError: low >= high
```

In [85]: `np.random.randint(-30,20,10)`

Out[85]: `array([ -8, -19, -24, -22, -8, -17, -26, 1, 5, 11], dtype=int32)`

In [86]: `np.random.randint(20,30,10)`

Out[86]: `array([28, 24, 21, 21, 28, 27, 23, 23, 28, 22], dtype=int32)`



```
In [87]: np.random.randint(10,21,3)
```

```
Out[87]: array([12, 12, 13], dtype=int32)
```

```
In [88]: np.random.randint(1,12,10)
```

```
Out[88]: array([ 7,  6,  8,  1,  7,  6,  9, 10,  8,  9], dtype=int32)
```

```
In [89]: np.random.randint(10,40,(10,10)) # generate the element 10 -30 with 4*4 mtri
```

```
Out[89]: array([[15, 20, 20, 14, 30, 25, 10, 20, 16, 33],
                [16, 12, 38, 11, 28, 13, 24, 26, 12, 28],
                [25, 11, 19, 15, 35, 17, 14, 30, 26, 28],
                [16, 33, 30, 19, 20, 26, 22, 37, 15, 26],
                [30, 10, 31, 34, 16, 10, 39, 37, 32, 10],
                [39, 17, 22, 14, 20, 39, 19, 35, 18, 18],
                [11, 33, 23, 33, 21, 36, 12, 36, 26, 26],
                [14, 19, 38, 17, 33, 24, 27, 27, 22, 39],
                [32, 23, 17, 29, 26, 32, 29, 34, 19, 10],
                [33, 37, 15, 16, 34, 14, 39, 11, 39, 34]], dtype=int32)
```

```
In [90]: np.random.randint(1,100,(12,12))
```

```
Out[90]: array([[11, 48, 10, 59, 81, 91, 55, 51, 95, 69, 59, 70],
                [39,  8, 48, 62, 96, 24, 32, 44, 69, 30, 67, 69],
                [13, 24, 82, 68, 65, 29, 69, 45, 87, 16, 73, 18],
                [56, 90, 25, 33, 63, 96, 68, 17,  4, 53,  3, 74],
                [98, 84, 58, 49, 20, 56, 23, 66, 29, 13, 32, 13],
                [72, 24, 70, 97, 89, 94,  8, 56, 12,  6, 63,  4],
                [13, 46, 20, 72, 55, 93, 20, 47, 25,  1, 58,  3],
                [29,  1, 70, 35, 37, 13, 30, 38, 88,  8,  7, 75],
                [ 2, 57, 47,  7,  6, 85, 29, 74, 71, 37,  3, 28],
                [45, 56,  3, 70,  4, 71, 21, 58, 54, 43,  7, 40],
                [92, 37, 61, 28, 95, 45, 12, 96, 16, 52, 77, 47],
                [66, 90, 54, 49, 39, 22, 69,  5, 79, 14, 10, 52]], dtype=int32)
```

```
In [91]: np.arange(1,13).reshape(3,4)
```

```
Out[91]: array([[ 1,  2,  3,  4],
                [ 5,  6,  7,  8],
                [ 9, 10, 11, 12]])
```

```
In [92]: np.arange(1,13).reshape(12,1)
```

```
Out[92]: array([[ 1],
                [ 2],
                [ 3],
                [ 4],
                [ 5],
                [ 6],
                [ 7],
                [ 8],
                [ 9],
                [10],
                [11],
                [12]])
```

```
In [93]: b = np.random.randint(10,20,(5,4))
         b
```

```
Out[93]: array([[10, 18, 11, 10],
               [13, 17, 15, 14],
               [18, 15, 17, 14],
               [15, 17, 19, 11],
               [17, 13, 10, 17]], dtype=int32)
```

```
In [94]: type(b)
```

```
Out[94]: numpy.ndarray
```

```
In [95]: b
```

```
Out[95]: array([[10, 18, 11, 10],
               [13, 17, 15, 14],
               [18, 15, 17, 14],
               [15, 17, 19, 11],
               [17, 13, 10, 17]], dtype=int32)
```

```
In [96]: b[:]
```

```
Out[96]: array([[10, 18, 11, 10],
               [13, 17, 15, 14],
               [18, 15, 17, 14],
               [15, 17, 19, 11],
               [17, 13, 10, 17]], dtype=int32)
```

```
In [97]: b[1:3]
```

```
Out[97]: array([[13, 17, 15, 14],
               [18, 15, 17, 14]], dtype=int32)
```

```
In [98]: b
```

```
Out[98]: array([[10, 18, 11, 10],
               [13, 17, 15, 14],
               [18, 15, 17, 14],
               [15, 17, 19, 11],
               [17, 13, 10, 17]], dtype=int32)
```

```
In [99]: b[1,2]
```

```
Out[99]: np.int32(15)
```

```
In [100... b[1,3]
```

```
Out[100... np.int32(14)
```

```
In [101... b[1,-1]
```

```
Out[101... np.int32(14)
```

```
In [102... b
```

```
Out[102... array([[10, 18, 11, 10],
               [13, 17, 15, 14],
               [18, 15, 17, 14],
               [15, 17, 19, 11],
               [17, 13, 10, 17]], dtype=int32)
```

```
In [103... b[2:3]
```

```
Out[103... array([[18, 15, 17, 14]], dtype=int32)
```

```
In [104... b[0:-2]
```

```
Out[104... array([[10, 18, 11, 10],  
        [13, 17, 15, 14],  
        [18, 15, 17, 14]], dtype=int32)
```

```
In [105... b
```

```
Out[105... array([[10, 18, 11, 10],  
        [13, 17, 15, 14],  
        [18, 15, 17, 14],  
        [15, 17, 19, 11],  
        [17, 13, 10, 17]], dtype=int32)
```

```
In [106... b[0,2]
```

```
Out[106... np.int32(11)
```

```
In [107... b
```

```
Out[107... array([[10, 18, 11, 10],  
        [13, 17, 15, 14],  
        [18, 15, 17, 14],  
        [15, 17, 19, 11],  
        [17, 13, 10, 17]], dtype=int32)
```

```
In [108... b[-5,-3]
```

```
Out[108... np.int32(18)
```

```
In [109... b[-4,2]
```

```
Out[109... np.int32(15)
```

```
In [110... np.random.randint(10,20,(4,4))
```

```
Out[110... array([[13, 15, 15, 14],  
        [18, 11, 14, 17],  
        [15, 11, 10, 16],  
        [16, 15, 14, 11]], dtype=int32)
```

```
In [111... b[-4,-2]
```

```
Out[111... np.int32(15)
```

```
In [112... b[-4:2]
```

```
Out[112... array([[13, 17, 15, 14]], dtype=int32)
```

```
In [113... b[:]
```

```
Out[113...] array([[10, 18, 11, 10],  
          [13, 17, 15, 14],  
          [18, 15, 17, 14],  
          [15, 17, 19, 11],  
          [17, 13, 10, 17]], dtype=int32)
```

operations

```
In [114...] a = np.random.randint(10,20,10)  
a
```

```
Out[114...] array([15, 14, 18, 15, 17, 12, 12, 10, 17, 19], dtype=int32)
```

```
In [115...] id(a)
```

```
Out[115...] 2832070774288
```

```
In [116...] arr
```

```
Out[116...] array([0, 1, 2, 3, 4, 5])
```

```
In [117...] arr2 = np.random.randint(0,100,(10,10))  
arr2
```

```
Out[117...] array([[93, 26, 24, 93, 52, 24, 97, 75, 18, 37],  
          [53, 76, 59, 20, 91,  3, 60, 75, 33, 18],  
          [95, 29, 84, 17, 70, 99, 61, 65, 69, 10],  
          [94, 52, 37, 40, 23, 67, 83,  3, 34, 77],  
          [ 2, 17, 61,  6, 24, 71, 87, 64, 39, 52],  
          [25,  8, 70, 77, 73, 47, 90, 72, 19, 44],  
          [31, 72, 99, 13, 67, 30, 68, 27, 27, 40],  
          [56,  3, 98, 70, 28, 54, 90, 78, 75, 45],  
          [29,  6, 63, 75, 27, 50, 71, 68,  4, 91],  
          [ 4, 19, 42, 62, 11, 78, 30, 23, 94, 68]], dtype=int32)
```

```
In [118...] arr[:]
```

```
Out[118...] array([0, 1, 2, 3, 4, 5])
```

```
In [119...] arr
```

```
Out[119...] array([0, 1, 2, 3, 4, 5])
```

```
In [120...] arr[:4]
```

```
Out[120...] array([0, 1, 2, 3])
```

```
In [121...] arr2[:]
```

```
Out[121...] array([[93, 26, 24, 93, 52, 24, 97, 75, 18, 37],
        [53, 76, 59, 20, 91,  3, 60, 75, 33, 18],
        [95, 29, 84, 17, 70, 99, 61, 65, 69, 10],
        [94, 52, 37, 40, 23, 67, 83,  3, 34, 77],
        [ 2, 17, 61,  6, 24, 71, 87, 64, 39, 52],
        [25,  8, 70, 77, 73, 47, 90, 72, 19, 44],
        [31, 72, 99, 13, 67, 30, 68, 27, 27, 40],
        [56,  3, 98, 70, 28, 54, 90, 78, 75, 45],
        [29,  6, 63, 75, 27, 50, 71, 68,  4, 91],
        [ 4, 19, 42, 62, 11, 78, 30, 23, 94, 68]], dtype=int32)
```

```
In [122...] arr2[0:5]
```

```
Out[122...] array([[93, 26, 24, 93, 52, 24, 97, 75, 18, 37],
        [53, 76, 59, 20, 91,  3, 60, 75, 33, 18],
        [95, 29, 84, 17, 70, 99, 61, 65, 69, 10],
        [94, 52, 37, 40, 23, 67, 83,  3, 34, 77],
        [ 2, 17, 61,  6, 24, 71, 87, 64, 39, 52]], dtype=int32)
```

```
In [123...] arr2
```

```
Out[123...] array([[93, 26, 24, 93, 52, 24, 97, 75, 18, 37],
        [53, 76, 59, 20, 91,  3, 60, 75, 33, 18],
        [95, 29, 84, 17, 70, 99, 61, 65, 69, 10],
        [94, 52, 37, 40, 23, 67, 83,  3, 34, 77],
        [ 2, 17, 61,  6, 24, 71, 87, 64, 39, 52],
        [25,  8, 70, 77, 73, 47, 90, 72, 19, 44],
        [31, 72, 99, 13, 67, 30, 68, 27, 27, 40],
        [56,  3, 98, 70, 28, 54, 90, 78, 75, 45],
        [29,  6, 63, 75, 27, 50, 71, 68,  4, 91],
        [ 4, 19, 42, 62, 11, 78, 30, 23, 94, 68]], dtype=int32)
```

```
In [124...] arr2[1,4]
```

```
Out[124...] np.int32(91)
```

```
In [125...] arr2
```

```
Out[125...] array([[93, 26, 24, 93, 52, 24, 97, 75, 18, 37],
        [53, 76, 59, 20, 91,  3, 60, 75, 33, 18],
        [95, 29, 84, 17, 70, 99, 61, 65, 69, 10],
        [94, 52, 37, 40, 23, 67, 83,  3, 34, 77],
        [ 2, 17, 61,  6, 24, 71, 87, 64, 39, 52],
        [25,  8, 70, 77, 73, 47, 90, 72, 19, 44],
        [31, 72, 99, 13, 67, 30, 68, 27, 27, 40],
        [56,  3, 98, 70, 28, 54, 90, 78, 75, 45],
        [29,  6, 63, 75, 27, 50, 71, 68,  4, 91],
        [ 4, 19, 42, 62, 11, 78, 30, 23, 94, 68]], dtype=int32)
```

```
In [126...] arr2[-5,5]
```

```
Out[126...] np.int32(47)
```

```
In [127...] arr2[-5,-5]
```

```
Out[127...] np.int32(47)
```

```
In [128...] arr2[-1,-2]
```

Out[128...] `np.int32(94)`

In [129...] `arr2[::-1]`

Out[129...] `array([[ 4, 19, 42, 62, 11, 78, 30, 23, 94, 68],  
 [29, 6, 63, 75, 27, 50, 71, 68, 4, 91],  
 [56, 3, 98, 70, 28, 54, 90, 78, 75, 45],  
 [31, 72, 99, 13, 67, 30, 68, 27, 27, 40],  
 [25, 8, 70, 77, 73, 47, 90, 72, 19, 44],  
 [ 2, 17, 61, 6, 24, 71, 87, 64, 39, 52],  
 [94, 52, 37, 40, 23, 67, 83, 3, 34, 77],  
 [95, 29, 84, 17, 70, 99, 61, 65, 69, 10],  
 [53, 76, 59, 20, 91, 3, 60, 75, 33, 18],  
 [93, 26, 24, 93, 52, 24, 97, 75, 18, 37]], dtype=int32)`

In [130...] `arr2[::-2]`

Out[130...] `array([[ 4, 19, 42, 62, 11, 78, 30, 23, 94, 68],  
 [56, 3, 98, 70, 28, 54, 90, 78, 75, 45],  
 [25, 8, 70, 77, 73, 47, 90, 72, 19, 44],  
 [94, 52, 37, 40, 23, 67, 83, 3, 34, 77],  
 [53, 76, 59, 20, 91, 3, 60, 75, 33, 18]], dtype=int32)`

In [131...] `arr2[::-3]`

Out[131...] `array([[ 4, 19, 42, 62, 11, 78, 30, 23, 94, 68],  
 [31, 72, 99, 13, 67, 30, 68, 27, 27, 40],  
 [94, 52, 37, 40, 23, 67, 83, 3, 34, 77],  
 [93, 26, 24, 93, 52, 24, 97, 75, 18, 37]], dtype=int32)`

In [132...] `arr2[::-3]`

Out[132...] `array([[93, 26, 24, 93, 52, 24, 97, 75, 18, 37],  
 [53, 76, 59, 20, 91, 3, 60, 75, 33, 18],  
 [95, 29, 84, 17, 70, 99, 61, 65, 69, 10],  
 [94, 52, 37, 40, 23, 67, 83, 3, 34, 77],  
 [ 2, 17, 61, 6, 24, 71, 87, 64, 39, 52],  
 [25, 8, 70, 77, 73, 47, 90, 72, 19, 44],  
 [31, 72, 99, 13, 67, 30, 68, 27, 27, 40]], dtype=int32)`

In [133...] `arr`

Out[133...] `array([0, 1, 2, 3, 4, 5])`

In [134...] `arr.max()`

Out[134...] `np.int64(5)`

In [135...] `arr.min()`

Out[135...] `np.int64(0)`

In [136...] `arr`

Out[136...] `array([0, 1, 2, 3, 4, 5])`

In [137...] `arr.mean()`

```
Out[137...] np.float64(2.5)
```

```
In [138...] arr.median()
```

```
-----  
AttributeError                                Traceback (most recent call last)  
Cell In[138], line 1  
----> 1 arr.median()  
  
AttributeError: 'numpy.ndarray' object has no attribute 'median'
```

```
In [139...] from numpy import *  
a = array([1,2,3,4,9])  
median(a)
```

```
Out[139...] np.float64(3.0)
```

Without work on import \* can you please find the median,mode)

```
In [140...] arr
```

```
Out[140...] array([0, 1, 2, 3, 4, 5])
```

```
In [141...] arr.reshape(3,2)
```

```
Out[141...] array([[0, 1],  
                [2, 3],  
                [4, 5]])
```

```
In [142...] arr.reshape(6,1)
```

```
Out[142...] array([[0],  
                [1],  
                [2],  
                [3],  
                [4],  
                [5]])
```

```
In [143...] np.random.rand(2,3)
```

```
Out[143...] array([[0.90271782, 0.07237785, 0.26342037],  
                  [0.82603786, 0.15892959, 0.49067188]])
```

```
In [144...] np.random.rand(4,6)
```

```
Out[144...] array([[0.21008875, 0.56150657, 0.52748755, 0.40764602, 0.78148559,  
                  0.1152702 ],  
                  [0.26748952, 0.42677472, 0.28635146, 0.53396719, 0.13595893,  
                  0.20176144],  
                  [0.36554062, 0.23613414, 0.39148584, 0.20907491, 0.79038777,  
                  0.93020362],  
                  [0.02472478, 0.08530048, 0.73298726, 0.9454732 , 0.45820114,  
                  0.40562152]])
```

```
In [145...] np.random.randint(3)
```

```
Out[145...] 1
```

```
In [146... np.random.randint(2,10)
```

```
Out[146... 3
```

```
In [147... np.random.randint(2,10,3)
```

```
Out[147... array([7, 9, 8], dtype=int32)
```

```
In [148... np.random.randint(2,10,4)
```

```
Out[148... array([5, 6, 4, 2], dtype=int32)
```

```
In [149... np.random.randint(10,20,30)
```

```
Out[149... array([10, 10, 14, 11, 16, 13, 15, 11, 19, 13, 14, 13, 13, 12, 19, 19, 15,
      15, 13, 17, 14, 17, 11, 10, 11, 17, 17, 13, 18, 15], dtype=int32)
```

```
In [150... np.random.randint(10,40,(10,10))
```

```
Out[150... array([[35, 30, 39, 12, 24, 37, 33, 13, 34, 19],
      [31, 30, 23, 15, 10, 31, 22, 36, 30, 37],
      [28, 18, 39, 33, 35, 12, 13, 32, 31, 17],
      [35, 23, 28, 21, 38, 17, 28, 38, 25, 36],
      [37, 18, 19, 16, 11, 16, 30, 30, 20, 38],
      [30, 12, 11, 33, 32, 38, 30, 35, 12, 25],
      [11, 34, 36, 31, 35, 27, 19, 37, 28, 17],
      [19, 17, 22, 37, 23, 16, 39, 31, 22, 20],
      [25, 28, 25, 22, 23, 23, 30, 10, 37, 19],
      [29, 29, 24, 11, 34, 18, 37, 16, 30, 11]], dtype=int32)
```

```
In [151... b = np.random.randint(10,40,(10,10))
b
```

```
Out[151... array([[31, 39, 37, 22, 11, 39, 34, 23, 23, 32],
      [31, 34, 21, 28, 24, 36, 22, 28, 34, 33],
      [38, 13, 31, 34, 34, 12, 26, 19, 37, 32],
      [37, 21, 14, 25, 14, 17, 10, 13, 23, 26],
      [39, 31, 30, 32, 38, 12, 18, 29, 12, 14],
      [18, 17, 13, 39, 38, 18, 32, 25, 38, 33],
      [22, 39, 12, 34, 39, 22, 39, 16, 15, 36],
      [22, 25, 14, 33, 33, 26, 14, 25, 24, 27],
      [12, 32, 16, 30, 26, 12, 35, 38, 31, 34],
      [17, 16, 11, 14, 13, 28, 38, 31, 28, 20]], dtype=int32)
```

```
arrarray[0,1,2,3,4,5]
```

```
In [152... arr.reshape(2,3)
```

```
Out[152... array([[0, 1, 2],
      [3, 4, 5]])
```

```
In [153... arr.reshape(3,3)
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[153], line 1
----> 1 arr.reshape(3,3)

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



```
In [154... arr.reshape(6,1)
```

```
Out[154... array([[0],  
        [1],  
        [2],  
        [3],  
        [4],  
        [5]])
```

```
In [155... arr.reshape(1,6)
```

```
Out[155... array([[0, 1, 2, 3, 4, 5]])
```

```
In [156... arr
```

```
Out[156... array([0, 1, 2, 3, 4, 5])
```

```
In [157... arr.reshape(2,4)
```

```
-----  
ValueError                                Traceback (most recent call last)  
Cell In[157], line 1  
----> 1 arr.reshape(2,4)  
  
ValueError: cannot reshape array of size 6 into shape (2,4)
```

```
In [158... arr
```

```
Out[158... array([0, 1, 2, 3, 4, 5])
```

```
In [159... arr.reshape(2,3,order='c')
```

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

```
In [160... arr.reshape(2,3,order='f') #print element with fortran
```

```
Out[160... array([[0, 2, 4],  
        [1, 3, 5]])
```

```
In [161... arr.reshape(2,3,order='A')
```

```
Out[161... array([[0, 1, 2],  
        [3, 4, 5]])
```

```
In [162... arr.reshape(2,3)
```

```
Out[162... array([[0, 1, 2],  
        [3, 4, 5]])
```

```
In [163... arr.reshape(1,6)
```

```
Out[163... array([[0, 1, 2, 3, 4, 5]])
```

```
In [164... arr.reshape(1,4)
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[164], line 1
----> 1 arr.reshape(1,4)

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

```
In [165... arr.reshape(2,6)
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[165], line 1
----> 1 arr.reshape(2,6)

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

```
In [166... arr.reshape(3,3)
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[166], line 1
----> 1 arr.reshape(3,3)

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

```
In [ ]: arr
```

```
In [167... arr.reshape(3,2)
```

```
Out[167... array([[0, 1],
                [2, 3],
                [4, 5]])
```

Indexing

```
In [168... mat = np.arange(0,100).reshape(10,10)
```

```
In [169... mat
```

```
Out[169... 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 [170... row = 4
          col = 5
```

```
In [171... col
```

```
Out[171... 5
```

```
In [172... row
```

Out[172...] 4

In [173...] `mat[row,col]`

Out[173...] `np.int64(45)`

In [174...] `mat[4,5]`

Out[174...] `np.int64(45)`

In [175...] `mat[:]`

Out[175...] `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 [176...] `col=6`

In [177...] `mat`

Out[177...] `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 [178...] `mat[6] # befault it represent to rows`

Out[178...] `array([60, 61, 62, 63, 64, 65, 66, 67, 68, 69])`

In [179...] `mat`

Out[179...] `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]])`

## slicing in matrix

```
In [180...] mat[:,col]
```

```
Out[180...] array([ 6, 16, 26, 36, 46, 56, 66, 76, 86, 96])
```

```
In [181...] mat
```

```
Out[181...] 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 [182...] mat[row,:]
```

```
Out[182...] array([40, 41, 42, 43, 44, 45, 46, 47, 48, 49])
```

```
In [183...] mat
```

```
Out[183...] 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 [184...] mat[:,8]
```

```
Out[184...] array([ 8, 18, 28, 38, 48, 58, 68, 78, 88, 98])
```

```
In [185...] mat[:,col]
```

```
Out[185...] 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 [186...] mat[:6]
```

```
Out[186...] 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 [187...] row
```

Out[187... 4

In [188... mat

```
Out[188... 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 [189... mat[:row]

```
Out[189... 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 [190... mat

```
Out[190... 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 [191... mat[row:]

```
Out[191... 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 [192... mat[:]

```
Out[192... 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 [193... mat[:,8]

Out[193...] array([ 8, 18, 28, 38, 48, 58, 68, 78, 88, 98])

In [194...] mat

Out[194...] 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 [195...] mat[:, -1]

Out[195...] array([ 9, 19, 29, 39, 49, 59, 69, 79, 89, 99])

In [196...] mat

Out[196...] 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 [197...] row

Out[197...] 4

In [198...] col

Out[198...] 6

In [199...] mat[:, col]

Out[199...] array([ 6, 16, 26, 36, 46, 56, 66, 76, 86, 96])

In [200...] mat

Out[200...] 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 [201...] mat[1:4]

```
Out[201...] 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]])
```

```
In [202...] mat
```

```
Out[202...] 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 [203...] mat[3:-3]
```

```
Out[203...] array([[30, 31, 32, 33, 34, 35, 36, 37, 38, 39],  
        [40, 41, 42, 43, 44, 45, 46, 47, 48, 49],  
        [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],  
        [60, 61, 62, 63, 64, 65, 66, 67, 68, 69]])
```

```
In [204...] mat
```

```
Out[204...] 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 [205...] mat[0]
```

```
Out[205...] array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [206...] mat[6]
```

```
Out[206...] array([60, 61, 62, 63, 64, 65, 66, 67, 68, 69])
```

```
In [207...] mat[5:7]
```

```
Out[207...] array([[50, 51, 52, 53, 54, 55, 56, 57, 58, 59],  
        [60, 61, 62, 63, 64, 65, 66, 67, 68, 69]])
```

```
In [208...] mat[0:10]
```

```
Out[208...] 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 [209...] mat[0:10:3]
```

```
Out[209...] 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 [210...] mat[0:10]
```

```
Out[210...] 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 [211...] mat[0:10:3]
```

```
Out[211...] 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 [212...] mat
```

```
Out[212...] 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 [213...] mat[4:]
```

```
Out[213...] 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 [214...

```
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...

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

Out[215...

```
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 [216...

```
mat
```

Out[216...

```
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 [217...

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

Out[217...

```
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 [218...

```
mat[::-2]
```

Out[218...

```
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 [219...

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

Out[219...

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

In [220...] `mat[2:6]`

Out[220...] `array([[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 [221...] `mat`

Out[221...] `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 [222...] `mat[2:6,2:4] #1:5 --> only row part ///1:3-- it indicates only column parts`

Out[222...] `array([[22, 23],  
 [32, 33],  
 [42, 43],  
 [52, 53]])`

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[0,1]`

Out[224...] `np.int64(1)`

In [225...] `mat[1,6]`

Out[225...] `np.int64(16)`

In [226...] `mat[1:6]`

Out[226...] `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 [227...] `mat[3:5]`

```
Out[227...] array([[30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
        [40, 41, 42, 43, 44, 45, 46, 47, 48, 49]])
```

```
In [228...] mat[1:2,2:4]
```

```
Out[228...] array([[12, 13]])
```

```
In [229...] mat[2:3,2:3]
```

```
Out[229...] array([[22]])
```

```
In [230...] mat[2:4,3:5]
```

```
Out[230...] array([[23, 24],
        [33, 34]])
```

```
In [231...] mat[3:5,2:4]
```

```
Out[231...] array([[32, 33],
        [42, 43]])
```

```
In [232...] mat[2:3,4:5]
```

```
Out[232...] array([[24]])
```

### Masking

```
In [233...] mat # we also called as filter
```

```
Out[233...] 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 [234...] id(mat)
```

```
Out[234...] 2832073325904
```

```
In [235...] mat[mat<50]
```

```
Out[235...] 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 [236...] mat[mat<=50]
```

```
Out[236...] array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50])
```

```
In [237...] mat > 50
```

```
Out[237...] 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 [238...] mat[mat==50]
```

```
Out[238...] array([50])
```

```
In [239...] mat == 50
```

```
Out[239...] 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],
        [False, False, False, False, False, False, False, False, False,
        False]])
```

```
In [240...] a1 = mat[mat<50]
a1
```

```
Out[240...] 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 [241...] mat
```

```
Out[241...] 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 [242...] a2 = mat[mat>50]
a2
```

```
Out[242...] 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 [243...] a3 = mat[mat<=50]
a3
```

```
Out[243...] array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50])
```

```
In [244...] a4 = mat[mat==50]
a4
```

```
Out[244...] array([50])
```

```
In [245...] b[1:4]
```

```
Out[245...] array([[31, 34, 21, 28, 24, 36, 22, 28, 34, 33],
        [38, 13, 31, 34, 34, 12, 26, 19, 37, 32],
        [37, 21, 14, 25, 14, 17, 10, 13, 23, 26]], dtype=int32)
```

```
In [246...] b[-1:]
```

```
Out[246...] array([[17, 16, 11, 14, 13, 28, 38, 31, 28, 20]], dtype=int32)
```

```
In [247...] b[:-1]
```

```
Out[247...] array([[31, 39, 37, 22, 11, 39, 34, 23, 23, 32],
        [31, 34, 21, 28, 24, 36, 22, 28, 34, 33],
        [38, 13, 31, 34, 34, 12, 26, 19, 37, 32],
        [37, 21, 14, 25, 14, 17, 10, 13, 23, 26],
        [39, 31, 30, 32, 38, 12, 18, 29, 12, 14],
        [18, 17, 13, 39, 38, 18, 32, 25, 38, 33],
        [22, 39, 12, 34, 39, 22, 39, 16, 15, 36],
        [22, 25, 14, 33, 33, 26, 14, 25, 24, 27],
        [12, 32, 16, 30, 26, 12, 35, 38, 31, 34]], dtype=int32)
```

```
In [248...] b[:-2]
```

```
Out[248...] array([[31, 39, 37, 22, 11, 39, 34, 23, 23, 32],
        [31, 34, 21, 28, 24, 36, 22, 28, 34, 33],
        [38, 13, 31, 34, 34, 12, 26, 19, 37, 32],
        [37, 21, 14, 25, 14, 17, 10, 13, 23, 26],
        [39, 31, 30, 32, 38, 12, 18, 29, 12, 14],
        [18, 17, 13, 39, 38, 18, 32, 25, 38, 33],
        [22, 39, 12, 34, 39, 22, 39, 16, 15, 36],
        [22, 25, 14, 33, 33, 26, 14, 25, 24, 27]]], dtype=int32)
```

```
In [249...] b[1:4]
```

```
Out[249...] array([[31, 34, 21, 28, 24, 36, 22, 28, 34, 33],
        [38, 13, 31, 34, 34, 12, 26, 19, 37, 32],
        [37, 21, 14, 25, 14, 17, 10, 13, 23, 26]]], dtype=int32)
```

```
In [250...] b[1,3]
```

```
Out[250...] np.int32(28)
```

```
In [251...] b[1,-1]
```

```
Out[251...] np.int32(33)
```

1.Array creation function

```
In [252...] import numpy as np
```

```
In [253...] # create an array from a list
a = np.array([1,2,3])
print("Array a:",a)
```

```
Array a: [1 2 3]
```

```
In [254...] b = np.arange(0,10,2)
b
```

```
Out[254...] array([0, 2, 4, 6, 8])
```

```
In [255...] d = np.zeros((2,3))
d
```

```
Out[255...] array([[0., 0., 0.],
        [0., 0., 0.]])
```

```
In [256...] e = np.ones((3,2))
e
```

```
Out[256...] array([[1., 1.],
        [1., 1.],
        [1., 1.]])
```

```
In [257...] f = np.eye(4)
f
```

```
Out[257...] array([[1., 0., 0., 0.],
        [0., 1., 0., 0.],
        [0., 0., 1., 0.],
        [0., 0., 0., 1.]])
```

## 2.Array manipulation function

```
In [258... a1 = np.array([1,2,3])
reshaped = np.reshape(a1,(1,3))
print("Reshaped array:",reshaped)
```

Reshaped array: [[1 2 3]]

```
In [259... f1 = np.array([[1,2],[3,4]])
flattened = np.ravel(f1) # flatten to 1d array
print("Flattened array:",flattened)
```

Flattened array: [1 2 3 4]

```
In [260... # Transpose an array
e1 = np.array([[1,2],[3,4]])
transposed = np.transpose(e1) #Transpose the array
print("Transposed array:\n",transposed)
```

Transposed array:

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

```
In [261... # stacked arrays vertically
a2 = np.array([1,2])
b2 = np.array([3,4])
stacked = np.vstack([a2,b2]) # stack a and b vertically
print("stacked arrays :\n",stacked)
```

stacked arrays :

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

## 3.Mathematical Functions

```
In [262... # Add two arrays
g = np.array([1,2,3,4])
added = np.add(g,2) #Add 2 each element
print("Added 2 to get:",added)
```

Added 2 to get: [3 4 5 6]

```
In [263... # square each element
squared = np.power(g,2) # square each element
print("squared g:",squared)
```

squared g: [ 1 4 9 16]

```
In [264... sqrt_val = np.sqrt(g) #square root of each element
print("square root of g:",sqrt_val)
```

square root of g: [1. 1.41421356 1.73205081 2. ]

```
In [265... print(a1)
print(g)
```

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

```
In [266... # Dot product of two arrays
a2 = np.array([1,2,3])
```

```
dot_product = np.dot(a2,g) # Dot product of a and g
print("Dot product of a and g:",dot_product)
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[266], line 3
      1 # Dot product of two arrays
      2 a2 = np.array([1,2,3])
----> 3 dot_product = np.dot(a2,g) # Dot product of a and g
      4 print("Dot product of a and g:",dot_product)

ValueError: shapes (3,) and (4,) not aligned: 3 (dim 0) != 4 (dim 0)
```

```
In [ ]: print(a)
        print(a1)
```

```
In [267... a3 = np.array([1,2,3])
dot_product = np.dot(a1,a)
print("Dot product of a1 and a:", dot_product)
```

Dot product of a1 and a: 14

#### 4.Statistical Functions

```
In [268... s = np.array([1,2,3,4])
mean = np.mean(s)
print("mean of s:",mean)
```

mean of s: 2.5

```
In [269... std_dev = np.std(s)
print("standard deviation of s:",std_dev)
```

standard deviation of s: 1.118033988749895

```
In [270... minimum = np.min(s)
print("min of s:",minimum)
```

min of s: 1

```
In [271... #maximum element of an array
maximum = np.max(s)
print("Max of s:",maximum)
```

Max of s: 4

#### 5.Linear Algebra Functions

```
In [272... # create a matrix
matrix = np.array([[1,2],[3,4]])
```

#### 6.Random Sampling Functions

```
In [273... random_vals = np.random.rand(3) #Array of 3 random values between 0 and 1
print("Random values:",random_vals)
```

Random values: [0.59566357 0.33364165 0.87654007]

```
In [274... # set seed for reproducibility
np.random.seed(0)
```



```
# Generate random integers
rand_ints = np.random.randint(0,10, size=5)
print("Random integers:", rand_ints)
```

Random integers: [5 0 3 3 7]

## 7.Boolean & logical functions

```
In [275... logical_test = np.array([True,False,True])
all_true = np.all(logical_test) #check if all are True
print("All elements True:",all_true)
```

All elements True: False

```
In [276... any_true = np.any(logical_test)
print("Any elements True:",any_true)
```

Any elements True: True

## 8.Set operations

```
In [277... set_a = np.array([1,2,3,4])
set_b = np.array([3,4,5,6])
intersection = np.intersect1d(set_a,set_b)
print("Intersection of a and b:",intersection)
```

Intersection of a and b: [3 4]

```
In [278... # union of two arrays
union = np.union1d(set_a,set_b)
print("union of a and b:",union)
```

union of a and b: [1 2 3 4 5 6]

## 9.Array Attribute Functions

```
In [279... a = np.array([1,2,3])
shape = a.shape
size = a.size
dimensions = a.ndim #number of dimensions
dtype = a.dtype
print("Shape of a:",shape)
print("size of a:",size)
print("Number of dimensions of a:", dimensions)
print("Data type of a:",dtype)
```

Shape of a: (3,)

size of a: 3

Number of dimensions of a: 1

Data type of a: int64

## 10.Other Functions

```
In [280... # create a copy of an array
a = np.array([1,2,3])
copied_array = np.copy(a) # create a copy of array a
print("copied array:",copied_array)
```

copied array: [1 2 3]

```
In [281]: array_size_in_bytes = a.nbytes # size in bytes
          print("size of a in bytes:",array_size_in_bytes)
```

size of a in bytes: 24

```
In [282]: shared = np.shares_memory(a, copied_array) # check if arrays share memory
          print("Do a and copied_array share memory?",shared)
```

Do a and copied\_array share memory? False

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

```
In [2]: np.array([2,4,56,422,32,1]) #1D array
```

```
Out[2]: array([ 2,  4, 56, 422, 32,  1])
```

```
In [3]: a = np.array([2,4,56,422,32,1]) #vector
          print(a)
```

```
[ 2  4 56 422 32  1]
```

```
In [4]: type(a)
```

```
Out[4]: numpy.ndarray
```

```
In [5]: # 2D Array (Matrix)

          new = np.array([[45,34,22,2],[24,55,3,22]])
          print(new)
```

```
[[45 34 22  2]
 [24 55  3 22]]
```

```
In [8]: np.array ( [[2,3,33,4,45],[23,45,56,66,2],[357,523,32,24,2],[32,32,44,33,234]]
```

```
Cell In[8], line 1
      np.array ( [[2,3,33,4,45],[23,45,56,66,2],[357,523,32,24,2],[32,32,44,33,23
4]]
```

^

```
_IncompleteInputError: incomplete input
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: