NUMPY

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

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

Out[3]:
```
'1.24.3'
```

```
pip install numpy
```

In [4]:
```
a=([1,2,33,4,5,6])
```

In [5]:
```
a
```

Out[5]:
```
[1, 2, 33, 4, 5, 6]
```

In [6]:
```
type(a)
```

Out[6]:
```
list
```

In [7]:
```
a=np.array([1,23,4,5,6,9,8])
```

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

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

In [10]:
```
a.ndim
```

Out[10]:
```
1
```

In [13]:
```
b=np.array([[6554]])
```

In [14]:
```
b
```

Out[14]:
```
array([[6554]])
```

In [15]:
```
b.ndim
```

Out[15]:
```
2
```

In [16]:
```
a1=np.array(5)
```

In [17]:
```
a1
```

Out[17]:
```
array(5)
```

In [18]:
```
a1.ndim
```

Out[18]:
```
0
```

In [19]:
```
type(a1)
```

Out[19]:
```
numpy.ndarray
```

In [21]:
```
b1=np.array([[[[[[[[[[[[[[[15]]]]]]]]]]]]]]])
```

In [22]:
```
b1.ndim
```

Out[22]:
```
15
```

2d array

In [23]:
```
a=np.array([[10,20],[20,30],[40,50]])
```

In [24]:
```
a
```

Out[24]:
```
array([[10, 20],
       [20, 30],
       [40, 50]])
```

In [25]:
```
a.ndim
```

Out[25]:
```
2
```

3d array

In [26]:
```
a=np.array([[[10,20,30,40],[50,60,70,80]]])
```

In [27]:
```
a
```

Out[27]:
```
array([[[10, 20, 30, 40],
        [50, 60, 70, 80]]])
```

In [28]:
```
a.ndim
```

Out[28]:
```
3
```

zeros

In [31]:
```
a=np.zeros(50)
```

In [32]:
```
a
```

Out[32]:
```
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 [33]:
```
len(a)
```

Out[33]:
```
50
```

In [34]:
```
np.zeros(10,int)
```

Out[34]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [35]:
```
np.zeros(10,complex)
```

Out[35]:
```
array([0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j,
       0.+0.j, 0.+0.j])
```

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

Out[36]:
```
array([False, False, False, False, False, False, False, False, False,
       False])
```

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

Out[38]:
```
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 [41]:
```
np.zeros((10,5),bool)
```

Out[41]:
```
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]])
```

ones

In [44]:
```
np.ones(10,bool)
```

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

In [46]:
```
np.ones((10,9),str)
```

Out[46]:
```
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']], dtype='<U1')
```

full()

In [49]:
```
np.full((5,5),2)
```

Out[49]:
```
array([[2, 2, 2, 2, 2],
       [2, 2, 2, 2, 2],
       [2, 2, 2, 2, 2],
       [2, 2, 2, 2, 2],
       [2, 2, 2, 2, 2]])
```

In [50]:
```
np.full((5),6)
```

Out[50]:
```
array([6, 6, 6, 6, 6])
```

In [54]:
```
np.full((5,3),'nagpur')
```

Out[54]:
```
array([['nagpur', 'nagpur', 'nagpur'],
       ['nagpur', 'nagpur', 'nagpur'],
       ['nagpur', 'nagpur', 'nagpur'],
       ['nagpur', 'nagpur', 'nagpur'],
       ['nagpur', 'nagpur', 'nagpur']], dtype='<U6')
```

In [59]:
```python
np.full((5,4,3),'nagpur')
```

Out[59]:
```
array([[['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur']],

       [['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur']],

       [['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur']],

       [['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur']],

       [['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur'],
        ['nagpur', 'nagpur', 'nagpur']]], dtype='<U6')
```

In [62]:
```python
np.full((5,4,3),int)
```

Out[62]:
```
array([[[<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>]],

       [[<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>]],

       [[<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>]],

       [[<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>]],

       [[<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>],
        [<class 'int'>, <class 'int'>, <class 'int'>]]], dtype=object)
```

In [63]:
```
np.full((5,4,3),5)
```

Out[63]:
```
array([[[5, 5, 5],
        [5, 5, 5],
        [5, 5, 5],
        [5, 5, 5]],

       [[5, 5, 5],
        [5, 5, 5],
        [5, 5, 5],
        [5, 5, 5]],

       [[5, 5, 5],
        [5, 5, 5],
        [5, 5, 5],
        [5, 5, 5]],

       [[5, 5, 5],
        [5, 5, 5],
        [5, 5, 5],
        [5, 5, 5]],

       [[5, 5, 5],
        [5, 5, 5],
        [5, 5, 5],
        [5, 5, 5]]])
```

random module

In [64]:
```
from numpy import random
```

In [101]:
```
random.random()
# range = 0 to 1
```

Out[101]:
```
0.8095175751126674
```

In [111]:
```
random.random(5)
```

Out[111]:
```
array([0.53913226, 0.64404259, 0.23925035, 0.91082576, 0.42662132])
```

In [123]:
```
random.random((3,5))
```

Out[123]:
```
array([[0.84746256, 0.74530652, 0.14625393, 0.01294277, 0.12090887],
       [0.92010195, 0.79568665, 0.30546747, 0.5404947 , 0.36063352],
       [0.72786757, 0.32119202, 0.10551902, 0.90737943, 0.07756242]])
```

randint

In [128]:
```
random.randint(5,20,3)
```

Out[128]:
```
array([ 8, 16,  8])
```

In [138]:
```
random.randint(100,200,100)
```

Out[138]:
```
array([127, 136, 194, 189, 185, 158, 164, 188, 101, 143, 110, 143, 132,
       192, 183, 128, 177, 100, 196, 156, 198, 119, 109, 185, 176, 193,
       132, 146, 136, 197, 122, 180, 130, 173, 130, 164, 173, 131, 122,
       155, 118, 170, 120, 176, 122, 100, 114, 162, 150, 178, 152, 126,
       190, 192, 121, 128, 177, 100, 175, 106, 112, 198, 142, 129, 167,
       166, 164, 166, 179, 137, 106, 127, 179, 191, 154, 158, 185, 108,
       170, 132, 101, 194, 184, 145, 114, 111, 196, 192, 151, 105, 104,
       146, 123, 188, 172, 107, 102, 197, 106, 186])
```

In [141]:
```
random.randint(1,20,(5,3))
```

Out[141]:
```
array([[11, 18, 10],
       [18,  4,  1],
       [ 3, 18, 13],
       [ 1,  8, 19],
       [17,  7, 18]])
```

In [144]:
```
random.randint(1,20,(5,5,3))
```

Out[144]:
```
array([[[13,  4,  8],
        [11, 19, 16],
        [15, 17,  7],
        [ 5,  3,  9],
        [ 7,  1, 17]],

       [[ 9,  3, 12],
        [ 4, 12,  2],
        [ 2, 10,  2],
        [13, 19,  6],
        [ 4,  8, 15]],

       [[ 7,  9,  1],
        [11,  9, 15],
        [11,  3, 18],
        [ 7,  9,  2],
        [10, 10,  8]],

       [[ 9, 12, 14],
        [18, 13, 11],
        [19, 18, 10],
        [11,  5, 17],
        [ 1, 13,  7]],

       [[15,  4, 19],
        [10,  5, 10],
        [ 2,  7, 17],
        [ 4, 11,  9],
        [13, 16,  4]]])
```

rand

In [215]:
```
random.rand(3)
```

Out[215]:
```
array([0.38898426, 0.62289159, 0.80912234])
```

In [216]:
```
random.rand(4,3)
```

Out[216]:
```
array([[0.92360809, 0.34355431, 0.44386853],
       [0.47167345, 0.06358297, 0.7446871 ],
       [0.14999067, 0.2121821 , 0.46350616],
       [0.0453307 , 0.39578985, 0.06041767]])
```

In [220]:
```
random.random((4,3))
```

Out[220]:
```
array([[0.42896015, 0.23759291, 0.40741992],
       [0.77515798, 0.6074869 , 0.59477359],
       [0.12506701, 0.26407525, 0.84643274],
       [0.37019649, 0.79208693, 0.88147613]])
```

randn

In [224]:
```
random.randn(5)
```

Out[224]:
```
array([ 0.41904554,  0.7866593 , -0.1941265 ,  1.00685113,  0.29660942])
```

In [227]:
```
random.randn(3)
```

Out[227]:
```
array([ 0.06408008,  0.21295595, -1.55079814])
```

In [228]:
```
random.randn(3,5)
```

Out[228]:
```
array([[ 0.05418224, -1.18747362, -0.42576796, -0.33482243, -0.29833045],
       [-1.45917668, -0.71063592, -0.95831333,  0.82095072, -0.60762179],
       [-0.86912649, -0.26790388,  0.36533451, -0.50517367,  1.89580572]])
```

linspace

In [230]:
```
np.linspace(-10,10,10)
```

Out[230]:
```
array([-10.        ,  -7.77777778,  -5.55555556,  -3.33333333,
        -1.11111111,   1.11111111,   3.33333333,   5.55555556,
         7.77777778,  10.        ])
```

In [232]:
```
np.linspace(10,50,25)
```

Out[232]:
```
array([10.        , 11.66666667, 13.33333333, 15.        , 16.66666667,
       18.33333333, 20.        , 21.66666667, 23.33333333, 25.        ,
       26.66666667, 28.33333333, 30.        , 31.66666667, 33.33333333,
       35.        , 36.66666667, 38.33333333, 40.        , 41.66666667,
       43.33333333, 45.        , 46.66666667, 48.33333333, 50.        ])
```

choice()

In [234]:
```
c=np.arange(1,10)
```

In [235]:
```
c
```

Out[235]:
```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [243]:
```
np.random.choice(c)
```

Out[243]:
```
9
```

shuffle

In [244]:
```
c
```

Out[244]:
```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [268]:
```
np.random.shuffle(c)
```

In [269]:
```
c
```

Out[269]:
```
array([6, 2, 4, 9, 3, 5, 8, 1, 7])
```

seed

In [364]:
```
random.seed(1)
a=random.randint(50,100,15)
```

In [365]:
```
random.seed(6)
b=random.randint(50,100,15)
```

In [366]:
```
a
```

Out[366]:
```
array([87, 93, 62, 58, 59, 61, 55, 65, 50, 66, 51, 62, 57, 95, 56])
```

In [367]:
```
b
```

Out[367]:
```
array([60, 59, 85, 70, 92, 95, 65, 92, 66, 75, 51, 61, 63, 76, 97])
```

reshape

In [368]:
```
a
```

Out[368]:
```
array([87, 93, 62, 58, 59, 61, 55, 65, 50, 66, 51, 62, 57, 95, 56])
```

In [369]:
```
len(a)
```

Out[369]:
```
15
```

In [370]:
```
a.reshape(3,5)
```

Out[370]:
```
array([[87, 93, 62, 58, 59],
       [61, 55, 65, 50, 66],
       [51, 62, 57, 95, 56]])
```

In [374]:
```
a=np.arange(0,500,10)
```

In [375]:
```
a
```

Out[375]:
```
array([  0,  10,  20,  30,  40,  50,  60,  70,  80,  90, 100, 110, 120,
       130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250,
       260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370, 380,
       390, 400, 410, 420, 430, 440, 450, 460, 470, 480, 490])
```

In [376]:
```
len(a)
```

Out[376]:
```
50
```

In [379]:
```
a.reshape(2,25)
```

Out[379]:
```
array([[  0,  10,  20,  30,  40,  50,  60,  70,  80,  90, 100, 110, 120,
        130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240],
       [250, 260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370,
        380, 390, 400, 410, 420, 430, 440, 450, 460, 470, 480, 490]])
```

In [371]:
```
b=np.array([10,20,30,40,50,60,70,80,90])
```

In [372]:
```
b
```

Out[372]:
```
array([10, 20, 30, 40, 50, 60, 70, 80, 90])
```

In [373]:
```
b.reshape(3,3)
```

Out[373]:
```
array([[10, 20, 30],
       [40, 50, 60],
       [70, 80, 90]])
```

In [380]:
```
b
```

Out[380]:
```
array([10, 20, 30, 40, 50, 60, 70, 80, 90])
```

In [381]:
```
b.reshape(2,5)
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Cell In[381], line 1
----> 1 b.reshape(2,5)

ValueError: cannot reshape array of size 9 into shape (2,5)
```

In [382]:
```
a
```

Out[382]:
```
array([  0,  10,  20,  30,  40,  50,  60,  70,  80,  90, 100, 110, 120,
       130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250,
       260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370, 380,
       390, 400, 410, 420, 430, 440, 450, 460, 470, 480, 490])
```

In [383]:
```
a.reshape(50,1)
```

Out[383]:
```
array([[  0],
       [ 10],
       [ 20],
       [ 30],
       [ 40],
       [ 50],
       [ 60],
       [ 70],
       [ 80],
       [ 90],
       [100],
       [110],
       [120],
       [130],
       [140],
       [150],
       [160],
       [170],
       [180],
       [190],
       [200],
       [210],
       [220],
       [230],
       [240],
       [250],
       [260],
       [270],
       [280],
       [290],
       [300],
       [310],
       [320],
       [330],
       [340],
       [350],
       [360],
       [370],
       [380],
       [390],
       [400],
       [410],
       [420],
       [430],
       [440],
       [450],
       [460],
       [470],
       [480],
       [490]])
```

arange

In [405]:
```
a=np.arange(10,50,2)
```

In [406]:
```
a
```

Out[406]:
```
array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
       44, 46, 48])
```

In [407]:
```
len(a)
```

Out[407]:
```
20
```

In [408]:
```
a.reshape(2,10)
```

Out[408]:
```
array([[10, 12, 14, 16, 18, 20, 22, 24, 26, 28],
       [30, 32, 34, 36, 38, 40, 42, 44, 46, 48]])
```

In [2]:
```
v=np.arange(1,17).reshape(4,4)
```

In [3]:
```
v
```

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

In [4]:
```
v[0]
```

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

In [5]:
```
v[-1]
```

Out[5]:
```
array([13, 14, 15, 16])
```

In [6]:
```
v[2]
```

Out[6]:
```
array([ 9, 10, 11, 12])
```

In [7]:
```
v[:]
```

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

In [8]:
```
v[::]
```

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

In [9]:
```
v[::-1]
```

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

In [10]:
```
v[::2]
```

Out[10]:
```
array([[ 1,  2,  3,  4],
       [ 9, 10, 11, 12]])
```

In [11]:
```
v[:,:]
```

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

In [12]:
```
v[[0,3,2,1]]
# passing rows index no.
```

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

In [13]:
```
v[:,:]
```

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

In [14]:
```
v[:2]
```

Out[14]:
```
array([[1, 2, 3, 4],
       [5, 6, 7, 8]])
```

In [24]:
```
v[2::]
```

Out[24]:
```
array([[ 9, 10, 11, 12],
       [13, 14, 15, 16]])
```

In [16]:
```
v
```

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

In [20]:
```
v[2:,2:]
```

Out[20]:
```
array([[11, 12],
       [15, 16]])
```

In [ ]:
```
# v [ columns : ,: rows ]
```

In [25]:
```
v
```

Out[25]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [13, 14, 15, 16]])
```

In [27]:
```
v[1:3,2:]
```

Out[27]:
```
array([[ 7,  8],
       [11, 12]])
```

In [29]:
```
v[0:1,0:2]
```

Out[29]:
```
array([[1, 2]])
```

In [30]:
```
v[3:4,0:2]
```

Out[30]:
```
array([[13, 14]])
```

In [31]:
```
v
```

Out[31]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [13, 14, 15, 16]])
```

In [32]:
```
v[:-1,-3:4]
```

Out[32]:
```
array([[ 2,  3,  4],
       [ 6,  7,  8],
       [10, 11, 12]])
```

Arithmatic opp

In [33]:
```
a11=np.array([1,2,3,4,5])
a22=np.array([10,20,30,40,50])
```

In [34]:
```
a11
```

Out[34]:
```
array([1, 2, 3, 4, 5])
```

In [35]:
```
a22
```

Out[35]:
```
array([10, 20, 30, 40, 50])
```

In [36]:
```
a11+a22
```

Out[36]:
```
array([11, 22, 33, 44, 55])
```

In [37]:
```
np.add(a11,a22)
```

Out[37]:
```
array([11, 22, 33, 44, 55])
```

```
In [38]: np.multiply(a11,a22)
Out[38]: array([ 10,  40,  90, 160, 250])

In [39]: np.divide(a11,a22)
Out[39]: array([0.1, 0.1, 0.1, 0.1, 0.1])

In [40]: a11
Out[40]: array([1, 2, 3, 4, 5])

In [41]: np.sqrt(a11)
Out[41]: array([1.        , 1.41421356, 1.73205081, 2.        , 2.23606798])

In [42]: np.cbrt(a11)
Out[42]: array([1.        , 1.25992105, 1.44224957, 1.58740105, 1.70997595])

In [43]: np.log(a11)
Out[43]: array([0.        , 0.69314718, 1.09861229, 1.38629436, 1.60943791])

In [45]: np.square(2)
Out[45]: 4

In [48]: np.power(2,3)
Out[48]: 8

In [49]: np.power(12,3)
Out[49]: 1728

In [50]: a11
Out[50]: array([1, 2, 3, 4, 5])

In [51]: np.average(a11)
Out[51]: 3.0

In [52]: a22
Out[52]: array([10, 20, 30, 40, 50])

In [53]: np.max(a22)
Out[53]: 50

In [54]: np.min(a22)
Out[54]: 10
```

In [55]:
```
np.mean(a22)
```

Out[55]:
```
30.0
```

In [57]:
```
np.std(a22)
```

Out[57]:
```
14.142135623730951
```

In [58]:
```
np.abs(-56)
```

Out[58]:
```
56
```

In [59]:
```
abs(-65982)
```

Out[59]:
```
65982
```

concatenate

In [62]:
```
n=np.array([1,2,3,4,5])
n1=np.array([6,7,8,9,10])
```

In [63]:
```
n
```

Out[63]:
```
array([1, 2, 3, 4, 5])
```

In [64]:
```
n1
```

Out[64]:
```
array([ 6,  7,  8,  9, 10])
```

In [65]:
```
np.concatenate((n,n1))
```

Out[65]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [66]:
```
np.concatenate((n+n1))
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Cell In[66], line 1
----> 1 np.concatenate((n+n1))

File <__array_function__ internals>:200, in concatenate(*args, **kwargs)

ValueError: zero-dimensional arrays cannot be concatenated
```

In [67]:
```
ar3=np.array([[10,20,30],[74,85,96],[96,56,41]])
ar4=np.array([[14,52,63],[63,52,41],[45,65,25]])
```

In [68]:
```
ar3
```

Out[68]:
```
array([[10, 20, 30],
       [74, 85, 96],
       [96, 56, 41]])
```

In [69]:
```
ar4
```

Out[69]:
```
array([[14, 52, 63],
       [63, 52, 41],
       [45, 65, 25]])
```

In [70]:
```
np.concatenate((ar3,ar4))
```

Out[70]:
```
array([[10, 20, 30],
       [74, 85, 96],
       [96, 56, 41],
       [14, 52, 63],
       [63, 52, 41],
       [45, 65, 25]])
```

In [71]:
```
np.concatenate((ar3,ar4),axis=0)
```

Out[71]:
```
array([[10, 20, 30],
       [74, 85, 96],
       [96, 56, 41],
       [14, 52, 63],
       [63, 52, 41],
       [45, 65, 25]])
```

In [72]:
```
np.concatenate((ar3,ar4),axis=1)
```

Out[72]:
```
array([[10, 20, 30, 14, 52, 63],
       [74, 85, 96, 63, 52, 41],
       [96, 56, 41, 45, 65, 25]])
```

stack

In [75]:
```
a11
```

Out[75]:
```
array([1, 2, 3, 4, 5])
```

In [76]:
```
a22
```

Out[76]:
```
array([10, 20, 30, 40, 50])
```

In [77]:
```
np.stack((a11,a22))
```

Out[77]:
```
array([[ 1,  2,  3,  4,  5],
       [10, 20, 30, 40, 50]])
```

In [78]:
```
ar3
```

Out[78]:
```
array([[10, 20, 30],
       [74, 85, 96],
       [96, 56, 41]])
```

In [79]:
```
ar4
```

Out[79]:
```
array([[14, 52, 63],
       [63, 52, 41],
       [45, 65, 25]])
```

In [80]:
```
np.stack((ar3,ar4))
```

Out[80]:
```
array([[[10, 20, 30],
        [74, 85, 96],
        [96, 56, 41]],

       [[14, 52, 63],
        [63, 52, 41],
        [45, 65, 25]]])
```

hstack

In [81]:
```
np.hstack((ar3,ar4))
```

Out[81]:
```
array([[10, 20, 30, 14, 52, 63],
       [74, 85, 96, 63, 52, 41],
       [96, 56, 41, 45, 65, 25]])
```

vstack

In [82]:
```
np.vstack((ar3,ar4))
```

Out[82]:
```
array([[10, 20, 30],
       [74, 85, 96],
       [96, 56, 41],
       [14, 52, 63],
       [63, 52, 41],
       [45, 65, 25]])
```

dstack

In [83]:
```
a11
```

Out[83]:
```
array([1, 2, 3, 4, 5])
```

In [84]:
```
a22
```

Out[84]:
```
array([10, 20, 30, 40, 50])
```

In [85]:
```
np.dstack((a11,a22))
```

Out[85]:
```
array([[[ 1, 10],
        [ 2, 20],
        [ 3, 30],
        [ 4, 40],
        [ 5, 50]]])
```

In [86]:
```
np.dstack((ar3,ar4))
```

Out[86]:
```
array([[[10, 14],
        [20, 52],
        [30, 63]],

       [[74, 63],
        [85, 52],
        [96, 41]],

       [[96, 45],
        [56, 65],
        [41, 25]]])
```

In [87]:
```
ar3
```

Out[87]:
```
array([[10, 20, 30],
       [74, 85, 96],
       [96, 56, 41]])
```

In [88]:
```
ar4
```

Out[88]:
```
array([[14, 52, 63],
       [63, 52, 41],
       [45, 65, 25]])
```

split

In [89]:
```
ar=np.arange(4,50)
```

In [90]:
```
ar
```

Out[90]:
```
array([ 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 [91]:
```
len(ar)
```

Out[91]:
```
46
```

In [96]:
```
np.split(ar,23)
```

Out[96]:
```
[array([4, 5]),
 array([6, 7]),
 array([8, 9]),
 array([10, 11]),
 array([12, 13]),
 array([14, 15]),
 array([16, 17]),
 array([18, 19]),
 array([20, 21]),
 array([22, 23]),
 array([24, 25]),
 array([26, 27]),
 array([28, 29]),
 array([30, 31]),
 array([32, 33]),
 array([34, 35]),
 array([36, 37]),
 array([38, 39]),
 array([40, 41]),
 array([42, 43]),
 array([44, 45]),
 array([46, 47]),
 array([48, 49])]
```

In [98]:
```
np.split(ar,2)
```

Out[98]:
```
[array([ 4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2
0,
        21, 22, 23, 24, 25, 26]),
 array([27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 4
3,
        44, 45, 46, 47, 48, 49])]
```

In [99]:
```
np.split(ar,3)
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Cell In[99], line 1
----> 1 np.split(ar,3)

File <__array_function__ internals>:200, in split(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\numpy\lib\shape_base.py:872, in split(a
ry, indices_or_sections, axis)
    870     N = ary.shape[axis]
    871     if N % sections:
--> 872         raise ValueError(
    873             'array split does not result in an equal division') fr
om None
    874 return array_split(ary, indices_or_sections, axis)

ValueError: array split does not result in an equal division
```

array_split

In [100]:
```
np.array_split(ar,3)
```

Out[100]:
```
[array([ 4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]),
 array([20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]),
 array([35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49])]
```

In [ ]: