In [1]:
```
print("Welcome to Numpy-1")
```

```
Welcome to Numpy-1
```

In [2]:
```
a=[1,2,3,4,5]
b= [i**2 for i in a]
print(b)
```

```
[1, 4, 9, 16, 25]
```

In [4]:
```
import numpy as np
a=np.array([1,2,3,4,5])
a**2
```

Out[4]:
```
array([ 1,  4,  9, 16, 25], dtype=int32)
```

In [5]:
```
type(a)
```

Out[5]:
```
numpy.ndarray
```

In [6]:
```
a.dtype
```

Out[6]:
```
dtype('int32')
```

In [7]:
```
a=range(10000)
```

In [8]:
```
%timeit [i**2 for i in a]
```

```
10.1 ms ± 1.29 ms per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

In [9]:
```
b=np.arange(10000)
```

In [10]:
```
%timeit b**2
```

```
22.5 µs ± 4.64 µs per loop (mean ± std. dev. of 7 runs, 10000 loops each)
```

In [11]:
```
a=range(1,10,5)
```

In [12]:
```
for i in a :
    print(i)
```

```
1
6
```

In [13]:
```
b=np.arange(1,10,0.5)
print(b)
```

```
[1.  1.5 2.  2.5 3.  3.5 4.  4.5 5.  5.5 6.  6.5 7.  7.5 8.  8.5 9.  9.5]
```

In [14]:
```
a=np.array([1,2,3,4,5])
```

```
print(type(a))
print(a.dtype)
```

```
<class 'numpy.ndarray'>
int32
```

In [17]:
```
a=np.array([1,2,3,4,5,'a'])
print(a)
print(type(a))
print(a.dtype)
```

```
['1' '2' '3' '4' '5' 'a']
<class 'numpy.ndarray'>
<U11
```

In [18]:
```
a=np.array([True,False])
print(a)
print(type(a))
print(a.dtype)
```

```
[ True False]
<class 'numpy.ndarray'>
bool
```

In [19]:
```
a=np.array([1,True,False])
print(a)
print(type(a))
print(a.dtype)
```

```
[1 1 0]
<class 'numpy.ndarray'>
int32
```

In [20]:
```
a=np.array([1,2,3,4,5])
a.ndim
```

Out[20]:
```
1
```

In [21]:
```
a.shape
```

Out[21]:
```
(5,)
```

In [22]:
```
# a=np.arange(start,stop,step)
```

In [24]:
```
#np.linspace(start,end,num)
```

In [27]:
```
a=np.linspace(1,10,15)
print(a)
```

```
[ 1.          1.64285714  2.28571429  2.92857143  3.57142857  4.21428571
  4.85714286  5.5         6.14285714  6.78571429  7.42857143  8.07142857
  8.71428571  9.35714286 10.        ]
```

In [28]:
```
a=np.array([[2,3,4],[4,5,6],[7,8,9]])
print(a)
```

```
[[2 3 4]
 [4 5 6]
 [7 8 9]]
```

In [29]:
```
a.ndim
```

Out[29]:
```
2
```

In [30]:
```
a.shape
```

Out[30]:
```
(3, 3)
```

In [33]:
```
type(a)
```

Out[33]:
```
numpy.ndarray
```

In [34]:
```
a=np.arange(1,13)
print(a)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12]
```

In [35]:
```
a.reshape((1,12))
```

Out[35]:
```
array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]])
```

In [36]:
```
a.reshape((12,1))
```

Out[36]:
```
array([[ 1],
       [ 2],
       [ 3],
       [ 4],
       [ 5],
       [ 6],
       [ 7],
       [ 8],
       [ 9],
       [10],
       [11],
       [12]])
```

In [37]:
```
a.reshape(2,6)
```

Out[37]:
```
array([[ 1,  2,  3,  4,  5,  6],
       [ 7,  8,  9, 10, 11, 12]])
```

In [38]:
```
a.reshape((3,4))
```

Out[38]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```

In [39]:
```
a.reshape((-1,6))
```

Out[39]:
```
array([[ 1,  2,  3,  4,  5,  6],
```

```
       [ 7,  8,  9, 10, 11, 12]])
```

In [40]:
```
a.reshape((6,-1))
```

Out[40]:
```
array([[ 1,  2],
       [ 3,  4],
       [ 5,  6],
       [ 7,  8],
       [ 9, 10],
       [11, 12]])
```

In [43]:
```
a.reshape(-1,-1) #here -1 gives the row or colom needed automtically
```

In [42]:
```
a
```

Out[42]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

In [44]:
```
b=np.arange(1,17)
print(b)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16]
```

In [52]:
```
c=b.reshape((1,16))
print(c)
```

```
[[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16]]
```

In [53]:
```
c.shape
```

Out[53]:
```
(1, 16)
```

In [54]:
```
c=b.reshape((16,1))
print(c)
```

```
[[ 1]
 [ 2]
 [ 3]
 [ 4]
 [ 5]
 [ 6]
 [ 7]
 [ 8]
 [ 9]
 [10]
 [11]
 [12]
 [13]
 [14]
 [15]
 [16]]
```

In [55]:
```
c.shape
```

Out[55]:
```
(16, 1)
```

In [57]:

```
b=np.arange(1,13)
c=b.reshape((3,4))
print(c)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

In [58]:
```
c.T
```

Out[58]:
```
array([[ 1,  5,  9],
       [ 2,  6, 10],
       [ 3,  7, 11],
       [ 4,  8, 12]])
```

In [59]:
```
c
```

Out[59]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```

In [60]:
```
c.flatten()
```

Out[60]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

In [62]:
```
b=np.arange(1,13).reshape(2,6)
print(b)
```

```
[[ 1  2  3  4  5  6]
 [ 7  8  9 10 11 12]]
```

In [63]:
```
c
```

Out[63]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```

In [68]:
```
a=c.flatten()
print(a)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12]
```

In [69]:
```
a[0:5]
```

Out[69]:
```
array([1, 2, 3, 4, 5])
```

In [70]:
```
a[5:11]
```

Out[70]:
```
array([ 6,  7,  8,  9, 10, 11])
```

In [71]:
```
a[-1]
```

Out[71]:
```
12
```

In [72]:
```

```

In [ ]:
```
a[4]
```

Out[72]:
```
5
```

In [73]:
```
a[-4:]
```

Out[73]:
```
array([ 9, 10, 11, 12])
```

In [74]:
```
a[::3]
```

Out[74]:
```
array([ 1,  4,  7, 10])
```

In [75]:
```
a[::-1]
```

Out[75]:
```
array([12, 11, 10,  9,  8,  7,  6,  5,  4,  3,  2,  1])
```

In [76]:
```
a[::-2]
```

Out[76]:
```
array([12, 10,  8,  6,  4,  2])
```

In [77]:
```
a[-10:-2:-1]
```

Out[77]:
```
array([], dtype=int32)
```

In [78]:
```
a[12::1]
```

Out[78]:
```
array([], dtype=int32)
```

In [79]:
```
a=np.arange(1,13).reshape((4,3))
```

In [80]:
```
print(a)
```

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

In [81]:
```
#a[row][column] method1
#a[row,column] method2

a[0]
```

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

In [82]:
```
a[0]
```

Out[82]:
```
array([1, 2, 3])
```

In [83]:
```
a[0,:]
```

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

In [84]:
```
a[:2]
```

Out[84]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [85]:
```
a[:2,:]
```

Out[85]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [88]:
```
a[1:3,:]
```

Out[88]:
```
array([[4, 5, 6],
       [7, 8, 9]])
```

In [89]:
```
a[1:3,:2]
```

Out[89]:
```
array([[4, 5],
       [7, 8]])
```

In [90]:
```
a[1:3,1:2]
```

Out[90]:
```
array([[5],
       [8]])
```

In [91]:
```
a
```

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

In [92]:
```
a[::2,::2]
```

Out[92]:
```
array([[1, 3],
       [7, 9]])
```

In [93]:
```
a[1::2,::2]
```

Out[93]:
```
array([[ 4,  6],
       [10, 12]])
```

In [94]:
```
a[[0,1,3],:]
```

Out[94]:
```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [10, 11, 12]])
```

In [95]:
```
a[(0,1,3),:]
```

Out[95]:
```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [10, 11, 12]])
```

In [96]:
```
a[(0,1,3),1:]
```

Out[96]:
```
array([[ 2,  3],
       [ 5,  6],
       [11, 12]])
```

In [98]:
```
a=np.arange(1,13).reshape((2,2))
print(a)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_9988/321618754.py in <module>
----> 1 a=np.arange(1,13).reshape((2,2))
      2 print(a)

ValueError: cannot reshape array of size 12 into shape (2,2)
```

In [99]:
```
a=np.arange(1,13).reshape((6,2))
print(a)
```

```
[[ 1  2]
 [ 3  4]
 [ 5  6]
 [ 7  8]
 [ 9 10]
 [11 12]]
```

In [100…
```
a=np.arange(1,13).reshape((6,10))
print(a)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_9988/3787312821.py in <module>
----> 1 a=np.arange(1,13).reshape((6,10))
      2 print(a)

ValueError: cannot reshape array of size 12 into shape (6,10)
```

In [101…
```
a=np.arange(1,13)
```

In [102…
```
a
```

Out[102…
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

In [103…
```
a
```

Out[103…
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

In [104…
```
a.reshape((-1,1)).shape
```

Out[104…
```
(12, 1)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: