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

In [2]:
```
l=[1,2,3,4,5,6,7,8,9,0]
l
```

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

In [3]:
```
type(l)
```

Out[3]:
```
list
```

In [4]:
```
arr=np.array(l)
```

In [5]:
```
arr
```

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

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

Out[6]:
```
numpy.ndarray
```

In [7]:
```
##ARANGE FUNCTION IN NUMPY
```

In [8]:
```
np.arange(10)             #arange function with one parameter/one arguement)
                              # the parameter is gives end-1 objects
```

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

In [9]:
```
np.arange(3)
```

Out[9]:
```
array([0, 1, 2])
```

In [10]:
```
np.arange(7)
```

Out[10]:
```
array([0, 1, 2, 3, 4, 5, 6])
```

In [11]:
```
np.arange(8)
```

Out[11]:
```
array([0, 1, 2, 3, 4, 5, 6, 7])
```

In [12]:
```
np.arange(15)
```

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

In [13]:
```
np.arange(2,5)          #arange function with two parameters/two arguement)
                                 #here 1st parameter is start
                                  #2nd parameter is end-1
```

Out[13]:
```
array([2, 3, 4])
```

In [15]:
```
np.arange(7,57)
```

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

In [16]:
```
np.arange(5,99)
```

Out[16]:
```
array([ 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])
```

In [17]:
```
np.arange(20,120)
```

Out[17]:
```
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,  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, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110,
       111, 112, 113, 114, 115, 116, 117, 118, 119])
```

In [18]:
```
np.arange(3,9)
```

Out[18]:
```
array([3, 4, 5, 6, 7, 8])
```

In [19]:
```
np.arange(10,100,20)          #arange function with three parameter/three a
                                 #here 1st parameter is start
                                  #2nd parameter is end-1
                                   #3nd parameter is step
```

Out[19]:
```
array([10, 30, 50, 70, 90])
```

In [20]:
```
np.arange(15,50,10)
```

Out[20]:
```
array([15, 25, 35, 45])
```

```
In [21]: np.arange(10,70,5)

Out[21]: array([10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65])
```

```
In [23]: np.arange(2,10,2,1)          #arange have max three parameter

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         Cell In[23], line 1
         ----> 1 np.arange(2,10,2,1)

         TypeError: Cannot interpret '1' as a data type
```

```
In [25]: np.zeros((2,3))       #it gives 2 rows and 3 columns with value zero
                                  #1st parameter indicates rows
                                   #2nd parameter indicates columns

Out[25]: array([[0., 0., 0.],
                [0., 0., 0.]])
```

```
In [26]: np.zeros((4,5))

Out[26]: array([[0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.]])
```

```
In [28]: np.ones((3,2))       #it gives 3 rows and 2 columns with value one
                                   #1st parameter indicates rows
                                    #2nd parameter indicates columns

Out[28]: array([[1., 1.],
                [1., 1.],
                [1., 1.]])
```

```
In [29]: np.ones((4,5))

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

In [30]:
```
np.twos((2,3))          #there no function names .twos
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[30], line 1
----> 1 np.twos((2,3))

File C:\ProgramData\anaconda3\lib\site-packages\numpy\__init__.py:311, in __
getattr__(attr)
    308     from .testing import Tester
    309     return Tester
--> 311 raise AttributeError("module {!r} has no attribute "
    312                      "{!r}".format(__name__, attr))

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

In [31]:
```
from numpy import*        # importing everythinng from numpy
```

In [32]:
```
arange(4)
```

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

In [34]:
```
np.arange(4)
```

Out[34]:
```
array([0, 1, 2, 3])
```

In [35]:
```
arange(5)
```

Out[35]:
```
array([0, 1, 2, 3, 4])
```

In [36]:
```
np.arange(5)
```

Out[36]:
```
array([0, 1, 2, 3, 4])
```

In [37]:
```
##RANDOM FUNCTION  IN NUMPY
```

In [38]:
```
np.random.randint(10,50)        #it gives the random number from given range
                                  #here its gives  random number from 10 to 5
```

Out[38]:
```
33
```

In [39]:
```
np.random.randint(10,50)
```

Out[39]:
```
41
```

```
In [40]: np.random.randint(10,50)

Out[40]: 38

In [41]: np.random.randint(10,50)

Out[41]: 30

In [42]: np.random.randint(10,50)

Out[42]: 42

In [43]: np.random.randint(1)

Out[43]: 0

In [44]: np.random.randint(3)

Out[44]: 1

In [45]: np.random.randint(3)

Out[45]: 0

In [46]: np.random.randint(3)

Out[46]: 0

In [47]: np.random.randint(3)

Out[47]: 1

In [48]: np.random.randint(10)

Out[48]: 5

In [49]: np.random.randint(10)

Out[49]: 2

In [50]: np.random.randint(10)

Out[50]: 5
```

In [51]:
```
np.random.randint(10)
```
Out[51]:
```
7
```

In [52]:
```
np.random.randint(10)
```
Out[52]:
```
7
```

In [53]:
```
np.random.randint(10)
```
Out[53]:
```
3
```

In [54]:
```
np.random.randint(2,8,5)                    #it gives the random 5 numbers from 2
                                               #each time it gives different numb
```
Out[54]:
```
array([4, 3, 4, 2, 2])
```

In [55]:
```
np.random.randint(2,8,5)
```
Out[55]:
```
array([6, 4, 6, 3, 7])
```

In [56]:
```
np.random.randint(2,8,5)
```
Out[56]:
```
array([7, 2, 6, 7, 7])
```

In [57]:
```
np.random.randint(1,9,3)
```
Out[57]:
```
array([1, 5, 4])
```

In [58]:
```
np.random.randint(1,9,3)
```
Out[58]:
```
array([7, 7, 3])
```

In [60]:
```
np.random.randint(10,20,(4,5))              #it gives 4x5 matrix with number betw
```
Out[60]:
```
array([[12, 14, 19, 19, 19],
       [19, 19, 19, 13, 18],
       [11, 10, 10, 18, 18],
       [10, 13, 10, 11, 17]])
```

In [61]:
```
np.random.randint(1,9,(3,4))
```
Out[61]:
```
array([[7, 7, 7, 1],
       [4, 5, 1, 7],
       [5, 8, 6, 5]])
```

```
In [62]: #OPERATIONS
```

```
In [3]: import numpy as np
```

```
In [4]: a=np.random.randint(1,10,5)
```

```
In [5]: a

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

```
In [6]: np.array(a)

Out[6]: array([4, 4, 2, 4, 6])
```

```
In [9]: arr2=np.random.randint(1,100,10)
        arr2

Out[9]: array([43, 27, 32, 52, 41,  6, 48, 85, 48, 98])
```

```
In [11]: a

Out[11]: array([4, 4, 2, 4, 6])
```

```
In [12]: a.max()

Out[12]: 6
```

```
In [13]: a.min()

Out[13]: 2
```

```
In [14]: a.mean()

Out[14]: 4.0
```

```
In [15]: arr2.mean()

Out[15]: 48.0
```

```
In [16]: arr2.max()

Out[16]: 98
```

In [17]:
```
arr2.min()
```

Out[17]:
```
6
```

In [18]:
```
arr2.median()             #numpy have no attribute like median
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[18], line 1
----> 1 arr2.median()

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

In [19]:
```
a
```

Out[19]:
```
array([4, 4, 2, 4, 6])
```

In [20]:
```
a.reshape(3,2)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[20], line 1
----> 1 a.reshape(3,2)

ValueError: cannot reshape array of size 5 into shape (3,2)
```

In [21]:
```
arr2.reshape(2,5)
```

Out[21]:
```
array([[43, 27, 32, 52, 41],
       [ 6, 48, 85, 48, 98]])
```

In [22]:
```
arr2.reshape(5,2)
```

Out[22]:
```
array([[43, 27],
       [32, 52],
       [41,  6],
       [48, 85],
       [48, 98]])
```

In [23]:
```
a.reshape(1,5)
```

Out[23]:
```
array([[4, 4, 2, 4, 6]])
```

In [24]:
```
a.reshape(5,1)
```

Out[24]:
```
array([[4],
       [4],
       [2],
       [4],
       [6]])
```

In [25]:
```
#INDEXING
```

In [26]:
```
j=np.arange(0,50).reshape(10,5)
```

In [27]:
```
j
```

Out[27]:
```
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 [28]:
```
row=5
col=4
```

In [29]:
```
j[5,4]
```

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

In [30]:
```
j[4,2]
```

Out[30]:
```
22
```

In [31]:
```
j[:]
```

Out[31]:
```
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 [32]: j[0:4]

Out[32]: array([[ 0,  1,  2,  3,  4],
                [ 5,  6,  7,  8,  9],
                [10, 11, 12, 13, 14],
                [15, 16, 17, 18, 19]])
```

```
In [33]: j[4:3]

Out[33]: array([], shape=(0, 5), dtype=int32)
```

```
In [34]: j[4:5]

Out[34]: array([[20, 21, 22, 23, 24]])
```

```
In [35]: j[0:5]

Out[35]: 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]])
```

```
In [37]: j[:,col]

Out[37]: array([ 4,  9, 14, 19, 24, 29, 34, 39, 44, 49])
```

```
In [39]: j[row,:]

Out[39]: array([25, 26, 27, 28, 29])
```

```
In [40]: j

Out[40]: 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 [43]: j[1:2,1:3,]

Out[43]: array([[6, 7]])
```

```
In [48]: ##MASKING
```

```
In [49]: j
```

```
Out[49]: 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 [50]: j<50
```

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

```
In [51]: j<25
```

```
Out[51]: array([[ 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],
                [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 [52]: j[j<25]
```

```
Out[52]: 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])
```

```
In [53]: m=j[j<25]
In [54]: m

Out[54]: 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])
```

```
In [55]: t=j[j>25]
```

```
In [56]: t

Out[56]: array([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 [57]: m

Out[57]: 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])
```

```
In [58]: t

Out[58]: array([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 [59]: median(t)

         ---------------------------------------------------------------------------
         NameError                                 Traceback (most recent call last)
         Cell In[59], line 1
         ----> 1 median(t)

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

```
In [60]: t.reshape(6,4)

Out[60]: array([[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 [ ]:
```