Numpy: package for multidimensional array

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

In [2]:
```
simple_list=[6,7,9]
```

In [3]:
```
np.array(simple_list)
```

Out[3]:
```
array([6, 7, 9])
```

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

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

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

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

In [6]:
```
np.arange(5,10) #print numbers from 5 to 10-1
```

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

In [7]:
```
np.arange(5,100)
```

Out[7]:
```
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, 99])
```

In [8]:
```
np.arange(1,31,5) #shift+tab->help box
#print number from 1 to 30 with gap of 5
```

Out[8]:
```
array([ 1,  6, 11, 16, 21, 26])
```

In [9]:
```
np.arange(5)
```

Out[9]:
```
array([0, 1, 2, 3, 4])
```

In [10]:
```
np.zeros(10)#print 10 zeros in float datatype
```

Out[10]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

In [12]:
```
np.zeros(10,int)
```

Out[12]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [13]:
```
np.ones(100)
```

Out[13]:
```
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.])
```

In [14]:
```
np.ones((2,3))
```

Out[14]:
```
array([[1., 1., 1.],
       [1., 1., 1.]])
```

In [15]:
```
np.zeros((2,5),int)
```

Out[15]:
```
array([[0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0]])
```

In [16]:
```
np.ones((2,5))
```

Out[16]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

In [17]:
```
np.linspace(0,2,5) #linearly spaced and number of value 5
```

Out[17]:
```
array([0. , 0.5, 1. , 1.5, 2. ])
```

In [18]:
```
np.linspace(0,20,8) #linearly spaced and number of value 8
```

Out[18]:
```
array([ 0.        ,  2.85714286,  5.71428571,  8.57142857, 11.42857143,
       14.28571429, 17.14285714, 20.        ])
```

In [19]:
```
np.eye((10)) #identity matrix of 10*10
```

Out[19]:
```
array([[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]])
```

In [20]:
```
np.random.rand(3,5) #3row 5col. uniform distribution from zero to one
```

Out[20]:
```
array([[0.0869448 , 0.24865793, 0.4427412 , 0.18852145, 0.01494222],
       [0.53237812, 0.82925839, 0.2238398 , 0.05599063, 0.76677497],
       [0.79065133, 0.24479188, 0.81798377, 0.08974423, 0.50613501]])
```

In [22]:
```
arr=np.random.randn(2,4) #both positive and negative num
arr
```

Out[22]:
```
array([[ 1.0350649 , -0.87435185,  0.04409244, -0.73682866],
       [-0.13332327,  1.04537704,  0.49299101, -0.57605865]])
```

In [30]:
```
arr=np.random.randint(2,100) #each time we will get diffrent number from 2 to 99
arr
```

Out[30]:
```
8
```

In [32]:
```
arr=np.random.randint(20,56,100) #20 to 55 , hundred random numbers
arr
```

Out[32]:
```
array([29, 40, 30, 40, 53, 22, 49, 50, 55, 25, 31, 39, 31, 24, 52, 47, 47,
       52, 49, 41, 21, 48, 37, 45, 38, 40, 28, 52, 29, 28, 33, 31, 20, 52,
       39, 34, 24, 38, 38, 24, 53, 43, 24, 48, 52, 42, 38, 35, 36, 49, 45,
       45, 38, 27, 20, 53, 24, 48, 32, 44, 50, 41, 45, 35, 22, 50, 25, 50,
       21, 36, 40, 22, 23, 31, 32, 52, 39, 52, 39, 50, 48, 53, 30, 49, 31,
       20, 41, 43, 55, 23, 52, 48, 39, 55, 47, 48, 39, 43, 24, 42])
```

In [35]:
```
rand_array=np.random.randint(0,100,20)

rand_array
```

Out[35]:
```
array([16, 57, 17, 84, 13, 13, 55, 82, 62, 42,  4, 74, 72, 23, 93, 24, 12,
       17, 31,  5])
```

In [37]:
```
sample_array=np.arange(30)
sample_array.reshape(5,6)
```

Out[37]:
```
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]])
```

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

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[38], line 1
----> 1 sample_array.reshape(4,3)

ValueError: cannot reshape array of size 30 into shape (4,3)
```

In [39]:
```
sample_array.reshape(3,10)
```

Out[39]:
```
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]])
```

In [45]:
```
rand_array.max()
```

Out[45]:
```
93
```

In [41]:
```
a=np.eye(5)
```

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

Out[42]:
```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

In [43]:
```
a.T
```

Out[43]:
```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

In [44]:
```
rand_array.argmax() #index of maxm
```

Out[44]:
```
14
```

In [46]:
```
sample_matrix=np.array([[50,20,1,23],[24,23,21,33],[56,76,24,7]])
sample_matrix
```

Out[46]:
```
array([[50, 20,  1, 23],
       [24, 23, 21, 33],
       [56, 76, 24,  7]])
```

In [47]:
```
sample_matrix[1,2] #row and col no. starts with zero to n-1
```

Out[47]:
```
21
```

In [48]:
```
sample_matrix[2,:]
```

Out[48]:
```
array([56, 76, 24,  7])
```

In [49]:
```
sample_matrix[2]
```

Out[49]:
```
array([56, 76, 24,  7])
```

In [52]:
```
sample_matrix[:,(3,2)]#all rows and 3rd & 2nd column
```

Out[52]:
```
array([[23,  1],
       [33, 21],
       [ 7, 24]])
```

In [ ]: