Numpy Arrays

In [72]:
```
import numpy as np
```

In [73]:
```
arr1=np.array([1,2,3,4,5])
arr1
```

Out[73]:
```
array([1, 2, 3, 4, 5])
```

In [74]:
```
type(arr1)
```

Out[74]:
```
numpy.ndarray
```

In [75]:
```
arr2=np.array([[1,2,3],[4,5,6]])
arr2
```

Out[75]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [76]:
```
arr3=np.zeros((2,3))
arr3
```

Out[76]:
```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

In [77]:
```
arr4=np.ones((3,3))
arr4
```

Out[77]:
```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

In [78]:
```
arr5=np.identity(5)
arr5
```

Out[78]:
```
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 [79]:
```
arr6=np.arange(5,16,2)
arr6
```

Out[79]:
```
array([ 5,  7,  9, 11, 13, 15])
```

In [80]:
```
arr7=np.linspace(10,20,10)
arr7
```

Out[80]:
```
array([10.        , 11.11111111, 12.22222222, 13.33333333, 14.44444444,
       15.55555556, 16.66666667, 17.77777778, 18.88888889, 20.        ])
```

In [81]:
```
arr8=arr7.copy()
arr8
```

Out[81]:
```
array([10.        , 11.11111111, 12.22222222, 13.33333333, 14.44444444,
       15.55555556, 16.66666667, 17.77777778, 18.88888889, 20.        ])
```

In [82]:
```
arr1
```

```
Out[82]: array([1, 2, 3, 4, 5])
```

Properties & Attributes

1. Shape
2. nDim
3. Size
4. ItemSize
5. Dtype
6. astype()

Shape

```
In [83]: arr1.shape

Out[83]: (5,)
```

```
In [84]: arr2.shape

Out[84]: (2, 3)
```

```
In [85]: arr9=np.array([[[1,2],[3,4]],[[5,6],[7,8]]])
```

```
In [86]: arr9

Out[86]: array([[[1, 2],
                 [3, 4]],

                [[5, 6],
                 [7, 8]]])
```

```
In [87]: arr9.shape

Out[87]: (2, 2, 2)
```

ndim

```
In [88]: arr9.ndim

Out[88]: 3
```

```
In [89]: arr2

Out[89]: array([[1, 2, 3],
                [4, 5, 6]])
```

```
In [90]: arr2.ndim

Out[90]: 2
```

```
In [91]: arr1.ndim
```

Out[91]:

```
1
```

Size

In [92]:

```
arr1.size
```

Out[92]:

```
5
```

In [93]:

```
arr9
```

Out[93]:

```
array([[[1, 2],
        [3, 4]],

       [[5, 6],
        [7, 8]]])
```

In [94]:

```
arr9.size
```

Out[94]:

```
8
```

itemSize

In [95]:

```
arr9.itemsize
```

Out[95]:

```
4
```

In [96]:

```
arr8.itemsize
```

Out[96]:

```
8
```

dtype

In [97]:

```
arr8.dtype
```

Out[97]:

```
dtype('float64')
```

In [98]:

```
arr9.dtype
```

Out[98]:

```
dtype('int32')
```

astype

In [99]:

```
arr9.astype('float')
```

Out[99]:

```
array([[[1., 2.],
        [3., 4.]],

       [[5., 6.],
        [7., 8.]]])
```

In [100…

```
lista=range(100)
arr11=np.arange(100)
```

In [101…

```
import sys
```

In [102…

```
print(sys.getsizeof(87)*len(lista))
```

```
2800
```

In [103…

```
print(arr11.itemsize*arr11.size)
```

```
400
```

Execution time calculate

In [104…

```
import time
```

In [105…

```
x=range(100000)
y=range(100000, 200000)

start_time=time.time()

c=[(x,y) for x,y in zip(x,y)]

print(time.time()-start_time)
```

```
0.041504621505737305
```

In [106…

```
a=np.arange(100000)
b=np.arange(100000,200000)

start_time=time.time()
c=a+b

print(time.time()-start_time)
```

```
0.013541460037231445
```

In [107…

```
arr12=np.arange(24).reshape(6,4)
arr12
```

Out[107]:

```
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]])
```

In [108…

```
arr12[2]
```

Out[108]:

```
array([ 8,  9, 10, 11])
```

Slicing: Syntax: Object[start:stop:step]

Print 3rd Column

In [109…

```
arr12[:,2]
```

Out[109]:

```
array([ 2,  6, 10, 14, 18, 22])
```

Print 3rd and 4th column

In [110…

```
arr12[:, 1:3]
```

Out[110]:

```
array([[ 1,  2],
       [ 5,  6],
       [ 9, 10],
       [13, 14],
       [17, 18],
       [21, 22]])
```

In [111…

```
arr12[:2]
```

Out[111]:

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

In [112…

```
arr12[2:4, 1:3]
```

Out[112]:

```
array([[ 9, 10],
       [13, 14]])
```

In [113…

```
arr12
```

Out[113]:

```
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]])
```

In [114…

```
arr12[4:6, 2:4]
```

Out[114]:

```
array([[18, 19],
       [22, 23]])
```

In [115…

```
arr12[4:,2:]
```

Out[115]:

```
array([[18, 19],
       [22, 23]])
```

In [116…

```
for i in arr12:
    print(i)
```

```
[0 1 2 3]
[4 5 6 7]
[ 8  9 10 11]
[12 13 14 15]
[16 17 18 19]
[20 21 22 23]
```

In [117…

```
for i in np.nditer(arr12):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```

In [118…

```
arr1=np.array([1,2,3,4,5,6])
arr2=np.array([4,5,6,7,8,9])
```

In [119…

```
arr1-arr2
```

Out[119]:

```
array([-3, -3, -3, -3, -3, -3])
```

In [120…

```
arr1*arr2
```

Out[120]:

```
array([ 4, 10, 18, 28, 40, 54])
```

In [121…

```
arr1*2
```

Out[121]:

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

In [122…

```
arr2>3
```

Out[122]:

```
array([ True,  True,  True,  True,  True,  True])
```

In [123…

```
arr3=np.arange(6).reshape(2,3)
arr4=np.arange(6,12).reshape(3,2)
```

In [124…

```
arr3.dot(arr4)
```

Out[124]:

```
array([[ 28,  31],
       [100, 112]])
```

In [129…

```
arr4
```

Out[129]:

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

In [130…

```
arr4.max()
```

Out[130]:

```
11
```

In [131…

```
arr4.min(axis=0)
```

Out[131]:

```
array([6, 7])
```

In [132…

```
arr4.min(axis=1)
```

Out[132]:

```
array([ 6,  8, 10])
```

In [133…

```
arr4.sum()
```

Out[133]:

```
51
```

In [134…

```
arr4.sum(axis=0)
```

Out[134]:

```
array([24, 27])
```

In [135…

```
arr4.mean()
```

Out[135]:

```
8.5
```

In [138…

```
arr4.std()
```

Out[138]:

```
1.707825127659933
```

In [139…

```
np.sin(arr4)
```

Out[139]:

```
array([[-0.2794155 ,  0.6569866 ],
       [ 0.98935825,  0.41211849],
       [-0.54402111, -0.99999021]])
```

In [140…

```
np.median(arr4)
```

Out[140]:

```
8.5
```

In [141…

```
np.mean(arr4)
```

Out[141]:

```
8.5
```

In [143…

```
np.exp(arr4)
```

Out[143]:

```
array([[  403.42879349,  1096.63315843],
       [ 2980.95798704,  8103.08392758],
       [22026.46579481, 59874.1417152 ]])
```

Reshaping Numpy Array

In [144…

```
arr4
```

Out[144]:

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

In [145…

```
arr4.ndim
```

Out[145]:

```
2
```

Change the shape of Numpy Array

In [146…

```
arr4.ravel()
```

Out[146]:

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

In [147…

```
arr4.transpose()
```

Out[147]:

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

In [148…

```
arr3
```

Out[148]:

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

In [154…

```
arr5=np.arange(12,18).reshape(2,3)
arr5
```

Out[154]:

```
array([[12, 13, 14],
       [15, 16, 17]])
```

In [156…

```
np.hstack((arr3,arr5))
```

Out[156]:

```
array([[ 0,  1,  2, 12, 13, 14],
       [ 3,  4,  5, 15, 16, 17]])
```

In [157…

```
np.vstack((arr3,arr5))
```

Out[157]:

```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [12, 13, 14],
       [15, 16, 17]])
```

In [158…

```
np.hsplit(arr3, 3)
```

Out[158]:

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

In [159…

```
np.vsplit(arr3,2)
```

Out[159]:

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

In [ ]:

```

```