

In [2]:

```
x=[4,5,6]
y=[1,2,3]
x+y
```

Out[2]:

```
[4, 5, 6, 1, 2, 3]
```

numpy data types

1. array
2. matrix

In [7]:

```
import numpy as np
x= np.array([4,2,3])
y= np.array([4,1,2])
z=x+y
print(z)
```

```
[8 3 5]
```

In [8]:

```
type(z)
```

Out[8]:

```
numpy.ndarray
```

In [9]:

```
k=z.tolist()
type(k)
```

Out[9]:

```
list
```

In [29]:

```
x=np.array([[4,5,6],[2,7,8],[5,9,0]])
x
```

Out[29]:

```
array([[4, 5, 6],
       [2, 7, 8],
       [5, 9, 0]])
```

In [30]:

```
x.shape
```

Out[30]:

```
(3, 3)
```

In [31]:

```
x.size
```

Out[31]:

```
9
```

In [32]:

```
x.ndim
```

Out[32]:

```
2
```

In [33]:

```
x.min()
```

Out[33]:

```
0
```

In [34]:

```
x.max()
```

Out[34]:

```
9
```

In [35]:

```
x.min(axis=0)#cloumn wise
```

Out[35]:

```
array([2, 5, 0])
```

In [28]:

```
x.min(axis=1)#row wise
```

Out[28]:

```
array([4, 2, 0])
```

In [36]:

```
x=np.array([[4,5],[1,2]],  
           [[3,2],[6,9]])  
x.size
```

Out[36]:

8

In [37]:

```
x.shape
```

Out[37]:

(2, 2, 2)

In [38]:

```
np.log10(100000)
```

Out[38]:

5.0

In [39]:

```
np.sin(np.deg2rad(90))
```

Out[39]:

1.0

In [42]:

```
#populating array with numpy  
x = np.arange(3,20,4)#start stop step  
x
```

Out[42]:

```
array([ 3,  7, 11, 15, 19])
```

In [45]:

```
x = np.linspace(5,20,9)# start, stop, num  
x
```

Out[45]:

```
array([ 5.    ,  6.875,  8.75  , 10.625, 12.5   , 14.375, 16.25  , 18.125,  
       20.    ])
```

In [46]:

```
k = np.ones(5)  
k
```

Out[46]:

```
array([1., 1., 1., 1., 1.])
```

In [55]:

```
k = np.ones((3,2), dtype='int32') #to convert the data type  
k
```

Out[55]:

```
array([[1, 1],  
       [1, 1],  
       [1, 1]])
```

In [56]:

```
m = np.zeros((5,2))  
m
```

Out[56]:

```
array([[0., 0.],  
       [0., 0.],  
       [0., 0.],  
       [0., 0.],  
       [0., 0.]])
```

In [57]:

```
#Linear algebra  
m = np.matrix([[7,4,5],[4,2,3],[4,6,8]])  
np.linalg.inv(m)
```

Out[57]:

```
matrix([[ 0.14285714,  0.14285714, -0.14285714],  
        [ 1.42857143, -2.57142857,  0.07142857],  
        [-1.14285714,  1.85714286,  0.14285714]])
```

In [58]:

```
np.linalg.det(m)
```

Out[58]:

```
-13.999999999999991
```

In [59]:

```
# 3x - 4y = 1  
# 4x + y = 14  
a = [[3,-4],[4,1]]  
b = [1,14]  
np.linalg.solve(a,b)
```

Out[59]:

```
array([3., 2.])
```

In []: