In [ ]:
```
#LINEAR ALGEBRA
#x+y=2
#2x+3y=6
# Find x and y value
```

In [4]:
```
import numpy as np
a=np.array([[1,1],[2,3]])
b=np.array([[2,6],[3,8]])
print(a,type(a))
print(b,type(b))
```

```
[[1 1]
 [2 3]] <class 'numpy.ndarray'>
[[2 6]
 [3 8]] <class 'numpy.ndarray'>
```

In [5]:
```
#solving linear algebra equations
np.linalg.solve(a,b)
```

Out[5]:
```
array([[ 3., 10.],
       [-1., -4.]])
```

In [6]:
```
#solving linear Algebra quotations
sol=np.linalg.solve(a,b)
print(sol,type(sol))
```

```
[[ 3. 10.]
 [-1. -4.]] <class 'numpy.ndarray'>
```

In [7]:
```
print("Value of x =",sol[0])
print("Value of y =",sol[1])
```

```
Value of x = [ 3. 10.]
Value of y = [-1. -4.]
```

In [8]:
```
print("Type of Linear Object =",type(np.linalg))
```

```
Type of Linear Object = <class 'module'>
```

In [9]:
```
a
```

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

In [11]:
```
a.T     # T is used for transpose of the matrix
```

Out[11]:
```
array([[1, 2],
       [1, 3]])
```

In [16]:
```
a= np.matrix([1,2,3,4,5,6,7,8,9])
print (a,type(a),id(a))
```

```
[[1 2 3 4 5 6 7 8 9]] <class 'numpy.matrix'> 2495294208928
```

In [18]:
```
a.shape=[3,3]
a
```

Out[18]:
```
matrix([[1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]])
```

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

Out[19]:
```
matrix([[1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]])
```

In [20]:
```
a.T
```

Out[20]:
```
matrix([[1, 4, 7],
        [2, 5, 8],
        [3, 6, 9]])
```

In [25]:
```
a=np.sort(a)
a
```

Out[25]:
```
matrix([[1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]])
```

In [26]:
```
a= np.transpose(a)
```

In [27]:
```
a
```

Out[27]:
```
matrix([[1, 4, 7],
        [2, 5, 8],
        [3, 6, 9]])
```