In [1]:
```
import numpy as np
a=np.array([5,6,9])
a[0]
```

Out[1]:
```
5
```

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

Out[2]:
```
2
```

In [4]:
```
a=np.array([5,6,9])
a.ndim
```

Out[4]:
```
1
```

In [5]:
```
a.itemsize
```

Out[5]:
```
4
```

In [6]:
```
a.dtype
```

Out[6]:
```
dtype('int32')
```

In [11]:
```
a=np.array([[1,2],[3,4],[5,6]], dtype=np.float64)
a.itemsize
```

Out[11]:
```
8
```

In [12]:
```
a
```

Out[12]:
```
array([[1., 2.],
       [3., 4.],
       [5., 6.]])
```

In [13]:
```
a.size
```

Out[13]:
```
6
```

In [14]:
```
a.shape
```

Out[14]:
```
(3, 2)
```

In [15]:
```
a = np.array([[1,2],[3,4],[5,6]], dtype=complex)
a
```

Out[15]:
```
array([[1.+0.j, 2.+0.j],
       [3.+0.j, 4.+0.j],
       [5.+0.j, 6.+0.j]])
```

In [23]:
```
np.zeros((3,4)) #np.ones((3,4))
```

Out[23]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

In [ ]:
```
#1=range(5)
```

In [29]:
```
np.arange(1,5)
```

Out[29]:
```
array([1, 2, 3, 4])
```

In [30]:
```
np.arange(1,5,2) #2 is steps
```

Out[30]:
```
array([1, 3])
```

In [31]:
```
np.linspace(1,5,10)
```

Out[31]:
```
array([1.        , 1.44444444, 1.88888889, 2.33333333, 2.77777778,
       3.22222222, 3.66666667, 4.11111111, 4.55555556, 5.        ])
```

In [32]:
```
np.linspace(1,5,5)
```

Out[32]:
```
array([1., 2., 3., 4., 5.])
```

In [34]:
```
np.linspace(1,5,20)
```

Out[34]:
```
array([1.        , 1.21052632, 1.42105263, 1.63157895, 1.84210526,
       2.05263158, 2.26315789, 2.47368421, 2.68421053, 2.89473684,
       3.10526316, 3.31578947, 3.52631579, 3.73684211, 3.94736842,
       4.15789474, 4.36842105, 4.57894737, 4.78947368, 5.        ])
```

In [35]:
```
a=np.array([[1,2],[3,4],[5,6]])
a
```

Out[35]:
```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

In [36]:
```
a.shape
```

Out[36]:
```
(3, 2)
```

In [38]:
```
a.reshape(2,3)
```

Out[38]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [39]:
```
a.reshape(6,1)
```

Out[39]:
```
array([[1],
       [2],
       [3],
       [4],
       [5],
       [6]])
```

In [40]:
```
a.ravel()
```

Out[40]:
```
array([1, 2, 3, 4, 5, 6])
```

In [41]:
```
a #will not touch original array
```

Out[41]:
```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

Mathmatical funtions

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

Out[42]:
```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

In [43]:
```
a.min()
```

Out[43]:
```
1
```

In [44]:
```
a.max()
```

Out[44]:
```
6
```

In [45]:
```
a.sum()
```

Out[45]:
```
21
```

In [46]:
```
a.sum(axis=0)
```

Out[46]:
```
array([ 9, 12])
```

In [48]:
```
a.sum(axis=1)
```

Out[48]:
```
array([ 3,  7, 11])
```

In [50]:
```
np.sqrt(a)
```

Out[50]:
```
array([[1.        , 1.41421356],
```

```
               [1.73205081, 2.        ],
               [2.23606798, 2.44948974]])
```

In [52]:
```
np.std(a)
```

Out[52]:
```
1.707825127659933
```

In [55]:
```
a=np.array([[1,2],[3,4]])
b=np.array([[5,6],[7,8]])
```

In [53]:
```
a
```

Out[53]:
```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

In [56]:
```
b
```

Out[56]:
```
array([[5, 6],
       [7, 8]])
```

In [57]:
```
a+b
```

Out[57]:
```
array([[ 6,  8],
       [10, 12]])
```

In [58]:
```
a*b
```

Out[58]:
```
array([[ 5, 12],
       [21, 32]])
```

In [59]:
```
a/b
```

Out[59]:
```
array([[0.2       , 0.33333333],
       [0.42857143, 0.5       ]])
```

In [60]:
```
a.dot(b) #Matrix product
```

Out[60]:
```
array([[19, 22],
       [43, 50]])
```

Indexing, Silencing, Iterating, Stacking