

```
In [1]: # Array  
import numpy as np  
a= ([5,5,5,5,5])  
a
```

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

```
In [3]: type (a)
```

```
Out[3]: list
```

```
In [5]: # List of lists  
import numpy as np  
b=np.array([[5,5,5],[5,5,5],[5,5,5]])  
b
```

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

```
In [6]: type(b)
```

```
Out[6]: numpy.ndarray
```

```
In [4]: import numpy as np  
x= ([1,1,1,1])  
x
```

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

```
In [8]: import numpy as np  
x=np.array ([ [1,1,1,1],[2,2,2,1],[3,3,3,3]])  
x
```

```
Out[8]: array([[1, 1, 1, 1],  
              [2, 2, 2, 1],  
              [3, 3, 3, 3]])
```

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

```
Out[9]: array([5, 5, 5])
```

```
In [10]: type(a)
```

```
Out[10]: numpy.ndarray
```

```
In [11]:
```

```
len(a)
```

Out[11]: 3

```
In [13]: a[0]
```

Out[13]: 5

```
In [14]: a[0:]
```

Out[14]: array([5, 5, 5])

```
In [15]: # List of lists  
b=np.array([[1,1,1],[1,1,1],[1,1,1]])  
b
```

Out[15]: array([[1, 1, 1],
[1, 1, 1],
[1, 1, 1]])

```
In [16]: type(b)
```

Out[16]: numpy.ndarray

```
In [18]: len(b)
```

Out[18]: 3

```
In [20]: b[0]
```

Out[20]: array([1, 1, 1])

```
In [21]: b[0:]
```

Out[21]: array([[1, 1, 1],
[1, 1, 1],
[1, 1, 1]])

How to create an array?

```
In [22]: import numpy as np  
a=np.array([1,2,3,4,5])  
a
```

Out[22]: array([1, 2, 3, 4, 5])

```
In [24]: b=np.zeros(2)
```

```
b
```

```
Out[24]: array([0., 0.])
```

```
In [26]: c=np.ones(3)
c
```

```
Out[26]: array([1., 1., 1.])
```

```
In [27]: # How to create an empty array with 2 elements?
d=np.empty(3)
d
```

```
Out[27]: array([1., 1., 1.])
```

```
In [2]: # with range of elements
import numpy as np
e=np.arange(6)
e
```

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

```
In [3]: # with specific range of elements
f=np.arange(2,20)
f
```

```
Out[3]: array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
              19])
```

```
In [4]: # continue.....
g=np.arange(2,20,2)
g
```

```
Out[4]: array([ 2,  4,  6,  8, 10, 12, 14, 16, 18])
```

```
In [6]: # Linerly spaced arrays
h=np.linspace(0,10,num=5) # gives us 5 nums
h
```

```
Out[6]: array([ 0. ,  2.5,  5. ,  7.5, 10. ])
```

```
In [7]: # specific data types in array
i=np.ones(5,dtype=np.int8)
i
```

```
Out[7]: array([1, 1, 1, 1, 1], dtype=int8)
```

```
In [8]: # specific data types in array
j=np.ones(3,dtype=np.float64)
```

```
j
```

```
Out[8]: array([1., 1., 1.])
```

```
In [10]: # 2-D (2dimensional array)  
np.zeros((3,4))
```

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

```
In [11]: np.ones((5,6))
```

```
Out[11]: 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.]])
```

```
In [12]: np.empty((3,4))
```

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

```
In [14]: # 3_D (3 - dimensional array)  
c=np.arange(24).reshape(2,3,4)  
c
```

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