In [1]:

```
import numpy as np
a = np.array([10,11,12,13,14])
a
```

Out[1]:

```
array([10, 11, 12, 13, 14])
```

In [2]:

```
b = np.zeros(10)
b
```

Out[2]:

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

In [3]:

```
c = np.ones(12)
c
```

Out[3]:

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

In [6]:

```
#create an empty array with 2 elements
d = np.empty(3)
d
```

Out[6]:

```
array([ 1.34497462e-284,  7.97870417e-312, -0.00000000e+000])
```

In [7]:

```
# with range of elements
e = np.arange(20)
e
```

Out[7]:

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19])
```

In [8]:

```
# with specific range of elements
f = np.arange(11, 30)
f
```

Out[8]:

```
array([11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
       28, 29])
```

In [9]:

```
# continue...
g = np.arange(10,50,10)
g
```

Out[9]:

```
array([10, 20, 30, 40])
```

In [10]:

```
# linerly spaced arrays
h = np.linspace(0,20,num=3)
h
```

Out[10]:

```
array([ 0., 10., 20.])
```

In [11]:

```
# specific data types in array
i = np.ones(6, dtype=np.int8)
i
```

Out[11]:
```
array([1, 1, 1, 1, 1, 1], dtype=int8)
```

In [12]:
```
j = np.ones(6, dtype=np.float64)
j
```

Out[12]:
```
array([1., 1., 1., 1., 1., 1.])
```

2-D Arrays

In [13]:
```
np.zeros((3,4))
```

Out[13]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

In [14]:
```
np.ones((4,6))
```

Out[14]:
```
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.]])
```

In [15]:
```
np.empty((3,4))
```

Out[15]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

3-D Arrays

In [16]:
```
# making and reshaping a 3D array
c = np.arange(24).reshape(2,3,4)
c
```

Out[16]:
```
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]]])
```