1) Create an array with zeros and ones

In [2]:
```
import numpy as np
print(np.zeros(2))
print(np.ones(3))
```

```
[0. 0.]
[1. 1. 1.]
```

2) Create an array and print output

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

```
[1 2 3 4]
```

3) Create an array whose initial content is random and print the output

In [4]:
```
b=np.empty(2)
print(b)
```

```
[0. 0.]
```

4) create an array with a range of values with even intervals

In [12]:
```
print(np.arange(0,10,2))
```

```
[0 2 4 6 8]
```

5)create an array with values that are spaced linearly in a specified intervals

In [6]:
```
print(np.linspace(1,15,num=2))
```

```
[ 1. 15.]
```

In [7]:
```
print(np.linspace(1,15,num=2,dtype=np.int64))
```

```
[ 1 15]
```

6)Access and manipulate element in an array

In [8]:
```
a=np.array([1,2,3,4])
print(a)
print(a[a<4])
```

```
[1 2 3 4]
[1 2 3]
```

7)Create a 2-dimensional array and check the shape of the array

In [33]:
```
c=np.array(([1,2,3,4],[5,6,7,8]))
print(c)
print(np.shape(c))
```

```
[[1 2 3 4]
 [5 6 7 8]]
```

```
(2, 4)
```

8)using the arrange() and linespace() func to evenly space values in a specified intervals

In [38]:
```
d=np.arange(8)
print(d)
print(np.linspace(1,15,num=2,dtype=np.int64))
```

```
[0 1 2 3 4 5 6 7]
[ 1 15]
```

9)create an array of random values between 0 and 1 in a given shape

In [43]:
```
b=np.empty(2)
print(b)
```

```
[4.9e-324 7.4e-323]
```

10)repeat each element of an array by a specified number of times using repeat() and tile() func

In [44]:
```
print(np.repeat(a,2))
print(np.tile(a,2))
```

```
[1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8]
[[1 2 3 4 1 2 3 4]
 [5 6 7 8 5 6 7 8]]
```

11)how do you know the shape and size of the array?

SHAPE - It will print the number of rows and columns in an array SIZE - It will print the length of an array

12)create an array that indicates the total number of elements in an array

In [46]:
```
a=np.array([1,2,3,4])
print(a)
print(np.size(a))
```

```
[1 2 3 4]
4
```

13)To find the number of dimensions of an array

In [48]:
```
c=np.array(([1,2,3,4],[5,6,7,8]))
print(c)
print(np.ndim(c))
```

```
[[1 2 3 4]
 [5 6 7 8]]
2
```

14)Create an array and reshape into a new array

In [53]:
```
e=np.arange(4)
print(e)
print(e.reshape(2,2))
```

```
[0 1 2 3]
[[0 1]
 [2 3]]
```

15)Create a null array of size 10

In [55]:
```
f=np.zeros(10)
print(np.size(f))
```

```
10
```

16)Create any array with values ranging from 10 to 49 and print the numbers whose remainder are zero when divided by 7

In [62]:
```
g=np.arange(10,50,1)
print(g[g%7==0])
```

```
[14 21 28 35 42 49]
```

17)Create an array and check any two conditions and print output

In [64]:
```
a=np.array([1,2,3,4])
print(a)
a1=a[(a>2)]
a2=a[(a<3)]
print(a1)
print(a2)
```

```
[1 2 3 4]
[3 4]
[1 2]
```

18)Using arithmetic operator print output using array

In [15]:
```
a2=np.array([1,2,3,4])
a3=np.array([5,6,7,8])
print(a2+a3)
print(a2-a3)
print(a2*a3)
print(a3/a2)
print(a2//a3)
print(a2%a3)
```

```
[ 6  8 10 12]
[-4 -4 -4 -4]
[ 5 12 21 32]
[5.         3.         2.33333333 2.        ]
[0 0 0 0]
[1 2 3 4]
```

19)Use relational operators and print the results using array

In [16]:
```
a4=a2[0]>a3[0]
print(a4)
```

```
False
```

20)Difference between python and ipython?

ipython is interactive python notebook compared to python. # It has many functionality compared to python shell.
ipython is known as jupyter notebook.