In [1]:
```
import numpy as np
```

Create an array with zeros and ones

In [2]:
```
print(np.zeros(10),np.ones(10))
```

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

Create an array and print the output

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

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

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

In [4]:
```
print(np.empty(5))
```

```
[2.12199579e-314 3.28357629e-310 3.20154539e-321 3.79442416e-321
 8.18974939e-308]
```

Create an array with the range of values with even intervals

In [5]:
```
print(np.linspace(0,1000,num=100,dtype=np.int64))
```

```
[   0   10   20   30   40   50   60   70   80   90  101  111  121  131
  141  151  161  171  181  191  202  212  222  232  242  252  262  272
  282  292  303  313  323  333  343  353  363  373  383  393  404  414
  424  434  444  454  464  474  484  494  505  515  525  535  545  555
  565  575  585  595  606  616  626  636  646  656  666  676  686  696
  707  717  727  737  747  757  767  777  787  797  808  818  828  838
  848  858  868  878  888  898  909  919  929  939  949  959  969  979
  989 1000]
```

create an array with values that are spaced linearly in a specified interval

In [6]:
```
print(np.linspace(0,5,num=3,dtype=np.int64))
```

```
[0 2 5]
```

Access and manipulate elements in the array

In [7]:
```
b = np.array([1,2,3])
print(b)
print(b[1])
b[1]=5
print(b[1])
```

```
[1 2 3]
2
5
```

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

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

```
(2, 3)
```

Using the arange() and linspace() function to evenly space values in a specified interval

In [9]:
```
a = np.arange(10,30)
print(a)
print(np.linspace(10,30,num=5))
```

```
[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29]
[10. 15. 20. 25. 30.]
```

Create an array of random values between 0 and 1 in a given shape

In [10]:
```
print(np.random.random())
```

```
0.5112009342456467
```

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

In [11]:
```
a = np.array([1,45,76,45,89,23,65,78])
print(np.repeat(a,2))
```

```
[ 1  1 45 45 76 76 45 45 89 89 23 23 65 65 78 78]
```

In [12]:
```
print(np.tile(a,5))
```

```
[ 1 45 76 45 89 23 65 78  1 45 76 45 89 23 65 78  1 45 76 45 89 23 65 78
  1 45 76 45 89 23 65 78  1 45 76 45 89 23 65 78]
```

How do you know the shape and size of an array?

In [13]:
```
print(np.shape(i))
```

```
(2, 3)
```

In [14]:
```
print(np.size(i))
```

```
6
```

Create an array that indicates the total number of elements in an array

In [15]:
```
a = np.array([[1,23,4,5],[6,5,7,89]])
print(np.size(a))
```

```
8
```

To find the number of dimensions of the array

In [16]:
```
print(np.ndim(a))
```

```
2
```

Create an array and reshape into a new array

In [17]:
```
a = np.array([[1,2,3,4,5,6],[8,9,0,1,2,3]])
print(a.reshape(4,3))
```

```
[[1 2 3]
 [4 5 6]
 [8 9 0]
 [1 2 3]]
```

Create a null array of size 10

In [18]:
```
print(np.zeros(10))
```

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

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

In [19]:
```
a = np.arange(10,49)
b = np.array([a])
print(b[(b%7)==0])
```

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

Create an array and check any two conditions and print the output

In [20]:
```
c=np.array([1,2,3,4,5,6])
if(c[0]<c[1]):
    print(c[0])
```

```
1
```

Use Arithmetic operator and print the output using array

In [21]:
```
a = np.array([10,9,8])
b = np.array([5,3,2])
print(a+b)
```

```
[15 12 10]
```

In [22]:
```
print(a-b)
```

```
[5 6 6]
```

In [23]:
```
print(a*b)
```

```
[50 27 16]
```

In [24]:
```
print(a/b)
```

```
[2. 3. 4.]
```

Use Relational operators and print the results using array

In [25]:
```
a = np.array([1,2,4])
b=np.array([1,6,8])
if(a[0]&b[0]):
    print(a[0])
```

```
1
```

In [26]:
```
a = np.array([1,2,4])
if((a[0]<a[2]) | (a[1]<a[2])):
    print(a[2])
```

```
4
```

In [27]:
```
a = np.array([1,2,4])
if(a[0]==1):
    print(~a[0])
```

```
-2
```

Difference between python and ipython

The differnce between python and i python is mainly that python is a language and ipython is a shell built in python. Ipython can be used for shell environment to execute python code in REPL (Read Eval Print Loop).It is orignally developed from Python programming language