In [2]:

```
import numpy as np
```

1. Create an array with zeros and ones

In [4]:

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

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

In [8]:

```
a=(np.zeros(4,dtype=np.int64))
print(a)
```

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

In [16]:

```
b=np.ones(3,dtype=np.int64)
print(b)
```

```
[1 1 1]
```

2.Create an array and print the output

In [17]:

```
c=np.array([2,4,6,8])
print(c)
```

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

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

In [32]:

```
d=(np.empty(5,dtype=np.int64))
print(d)
```

```
[         4294967297 4786574715898560512                 984
       3070918394624  437781257694460440]
```

4. Create an array with the range of values with even intervals

In [37]:

```
e=np.arange(0,20,+2)
print(e)
```

```
[ 0  2  4  6  8 10 12 14 16 18]
```

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

In [40]:

```
f=np.linspace(1,50,num=25,dtype=np.int64)
print(f)
```

```
[ 1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47
 50]
```

6. Access and manipulate elements in the array

In [42]:

```
print(e[4])
```

```
8
```

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

In [44]:

```
g=np.array([[5,6,7],[8,9,10]])
print(g)
```

```
[[ 5  6  7]
 [ 8  9 10]]
```

In [45]:

```
print(np.shape(g))
```

```
(2, 3)
```

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

In [58]:

```
print(np.arange(0,51,+2))
print(np.linspace(0,50,num=26,dtype=np.int64))
```

```
[ 0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46
 48 50]
[ 0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46
 48 50]
```

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

In [63]:

```
print(np.empty(10,dtype=np.int8))
```

```
[ 48 -56  63  69 -53   2   0   0   0   0]
```

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

In [65]:

```
print(np.repeat(g,3))
```

```
[ 5  5  5  6  6  6  7  7  7  8  8  8  9  9  9 10 10 10]
```

In [66]:

```
print(np.tile(g,3))
```

```
[[ 5  6  7  5  6  7  5  6  7]
 [ 8  9 10  8  9 10  8  9 10]]
```

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

In [67]:

```
print(np.shape(g))
print(np.size(g))
```

```
(2, 3)
6
```

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

In [68]:

```
print(np.size(f))
```

```
25
```

13. To find the number of dimensions of the array

In [69]:

```
print(np.ndim(g))
```

```
2
```

14. Create an array and reshape into a new array

In [75]:

```
x=np.arange(10)
print(x)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

In [79]:

```
y=x.reshape(5,2)
print(y)
```

```
[[0 1]
 [2 3]
 [4 5]
 [6 7]
 [8 9]]
```

15.create a null array of size 10

In [82]:

```
print(np.size(g))
```

```
6
```

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

In [86]:

```
i=np.arange(10,49)
print(i[i%7==0])
```

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

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

In [94]:

```
print(i[(i>20)&(i<40)])
```

```
[21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39]
```

18. Use Arithmetic operator and print the output using array

In [95]:

```
print(i[30]+i[10])
```

```
60
```

19. Use Relational operators and print the results using array

In [97]:

```
print(i[(i%5!=0)&(i%2!=0)])
```

```
[11 13 17 19 21 23 27 29 31 33 37 39 41 43 47]
```

20. Difference between python and ipython

```
python:
   It is routinely used by system administrators and web developers. Also, many
scientists are using Python thanks to libraries such as NumPy, SciPy, pandas, and
matplotlib.
ipython:
    IPython is an interactive command-line terminal for Python. It was created by
Fernando Perez in 2001. IPython offers an enhanced read-eval-print loop (REPL)
environment particularly well adapted to scientific computing.
```

In [ ]: