# Python sys module Python sys module

provides easy functions that allow us to interact with the interpreter directly.

he functions python sys module provides allows us to operate on underlying interpreter, irrespective of it being a Windows Platform, Macintosh or Linux. In this lesson, we will review these functions and what we can do with these.

FEW FUNCTIONS IN SYS

import sys

print("You entered: ",sys.argv[1], sys.argv[2], sys.argv[3])

### **Python sys.exit()**

This method makes the Python interpretor exits the current flow of execution abruptly.

### Python sys.path

This function just displays the PYTHONPATH set in current system.

>>> import sys

>>>sys.path

['', 'C:\\python36\\Lib\\idlelib', 'C:\\python36\\python36.zip',

'C:\\python36\\DLLs', 'C:\\python36\\lib', 'C:\\python36',

'C:\\Users\\acer\\AppData\\Roaming\\Python\\Python36\\site-packages',

'C:\\python36\\lib\\site-packages']

sys.argv returns a list of command line arguments passed to a Python script. The item at index 0 in this list is always the name of the script. The rest of the arguments are stored at the subsequent indices.

## **sys.maxsize**

Returns the largest integer a variable can take.

Example: sys.maxsize

 Copy

>>> import sys

>>>sys.maxsize

9223372036854775807

## **sys.version**

This attribute displays a string containing the version number of the current Python interpreter.

>>> import sys

>>>sys.version

'3.7.0 (v3.7.0:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)]'