Python sys Module Explained with Examples

The sys module provides functions and variables that are used to manipulate the Python runtime environment.

# Key Features of sys Module

## 1. sys.argv

This list contains the command-line arguments passed to a Python script.

Example:  
import sys  
print("Script name:", sys.argv[0])  
print("Arguments:", sys.argv[1:])  
  
Running the script as:  
$ python script.py arg1 arg2  
  
Output:  
Script name: script.py  
Arguments: ['arg1', 'arg2']

## 2. sys.exit([arg])

This function is used to exit the program. The optional argument arg can be an integer (exit code) or another object.  
Example:  
import sys  
if len(sys.argv) < 2:  
 print("Not enough arguments.")  
 sys.exit(1)  
else:  
 print("Arguments provided!")

## 3. sys.path

This is a list of strings that specifies the search path for modules. You can manipulate sys.path to dynamically alter where Python looks for modules.  
Example:  
import sys  
print("Current module search path:")  
for path in sys.path:  
 print(path)

## 4. sys.platform

This string provides a platform identifier, helping you check the operating system.  
Example:  
import sys  
print("Running on platform:", sys.platform)

## 5. sys.version

This provides detailed information about the version of Python being used.  
Example:  
import sys  
print("Python version:", sys.version)

## 6. sys.stdin, sys.stdout, sys.stderr

These are file objects corresponding to standard input, output, and error. You can use them to read/write from/to the terminal.  
Example:  
import sys  
sys.stdout.write("This is printed to standard output.\n")  
sys.stderr.write("This is an error message.\n")

## 7. sys.getsizeof()

Returns the size of an object in bytes.  
Example:  
import sys  
data = [1, 2, 3, 4, 5]  
print("Size of the list:", sys.getsizeof(data), "bytes")