SYS module:

* The sys module is used to work with Python runtime environment.
* The sys module provides functions and variables used to manipulate different parts of the python runtime environment

How to use sys module

import sys

How to get help on sys module

dir(sys),help(sys)

“

import sys

# print(sys.version) #print the Python version

# print(sys.version\_info) #print the version info in tuple

# print(sys.modules) #list all the modules that are currently loaded

print(sys.path) #list all the paths where Python looks for modules

# sys.path.append('/path/to/your/module') #add a new path to the list

# ['C:\\SEQ\\PYTHON\_PR\\SYS', 'C:\\Python37\\python37.zip',

#'C:\\Python37\\DLLs', 'C:\\Python37\\lib', 'C:\\Python37', 'C:\\Python37\\lib\\site-packages']

sys.exit(2) #exit the program with a status code of 0 (success)

print("Hello")  # Hello will not be printed because the program exits before this line

“

* Sys.argv returns a list of command line arguments passed to a Python Script

import os

print("Hello")

$ python sys\_argv.py hi hello how are you

Hello

print(sys.argv) #['sys\_argv.py', 'hi', 'hello', 'how', 'are', 'you'] #List

print(sys.argv[0]) # sys\_argv.py

print(type(sys.argv[1])) # <class 'str'>

prog2.py

“

import sys

usr\_str=input("Enter a string: ")

usr\_action=input("Enter an action: lower/upper/title:")

if usr\_action == "lower":

    print(usr\_str.lower())

elif usr\_action == "upper":

    print(usr\_str.upper())

elif usr\_action == "title":

    print(usr\_str.title())

else:

    print("Invalid action. Please enter lower, upper, or title::::")

“

Prog 3:

usr\_str=sys.argv[1]

usr\_action=sys.argv[2]

prog 4:

if len(sys.argv) != 3:

    print("Usage: python sys\_argv.py <string> <action>")

    sys.exit(1)

usr\_str=sys.argv[1]

usr\_action=sys.argv[2]