# Interpreter: -

Open the interpreter in Windows with command prompt WIN + R and type python or python3.

Second way to open the interpreter in Windows is python -c command [arg]. . .

Closing the interpreter Control + Z or quit ().

We can use python modules as script by using below command

> python -m module [arg] . . .

# Command Line and Environment: -

Generic Syntax:

python [ -bBdEhiIOqsSuvVWx?] [ -c command | -m module – name | script | -] [ arg]

## Interface Options: -

* When called with standard input connected to a tty device, it prompts for commands and executes them until an EOF is read. For Windows, (EOF can be provided by Ctrl + Z and Enter)
* When called with a file name argument or with a file as standard input, it reads and executes script from that file.
* When called with a directory name argument, it reads and executes a script from that directory.
* When called with a **command -c**,it executes python commands from no. of statements separated by NEWLINE character.

e.g. > python -c “print(‘Hello World’)”

* When called with a **command -m module**, the given module located on python default path will be executed as a script.

e.g. > python -m “test”

# Non- Interactive mode:

In Non – Interactive mode, entire input file parsed before the execution.

All the arguments in command end up in the **sys.argv**. Note that the first argument **sys.argv[0]** is program file source location.

* - C<command>:

Execute python code in commands. Command can be one or more python statements separated by the newlines with significant white spaces in each statements.

In this case, sys.argv[0] is having ‘-C’ value and sys.path should have the current directory location.

* - M<module-name>:

Search module name in current working directory and execute it’s \_\_main\_\_ module.

In this case, we don’t need to provide the .py extension. The module name should be valid python module name.

Package names are also permitted. When a package name is supplied instead of a normal module, python will try to execute the <Pack>.\_\_main\_\_ module.

Note: this option will be valid for precompiled modules but not for built in modules or extension modules written in C.

In this case, sys.argv[0] is having ‘-M’ value before finding the module location and sys.path should have the current directory location.

* - I<Isolation Mode>:

- I can be used in Isolation mode where sys.path contains neither the current directory nor the user’s site-packages directory. All PYTHON\* environment variables will be ignored, too.

# Simulating Python In o Interpreter: