SPPU New Syllabus

A Book Of

Python Programming

For MCA(Management) : Semester - II

[Course Code IT-21 : Credit - 03]

CBCS Pattern

As Per New Syllabus, Effective from June 2020

Vijay T. Patil

M.E.(Computer Engineering)
Head, Department of Computer Engineering (NBA Accredited)
Vidyalankar Polytechnic, Mumbai-37

Mrs. Manisha Pokharkar

M.E.(Computer Engineering)
Sr. Lecturer, Department of Computer Engineering (NBA Accredited)
Vidyalankar Polytechnic, Mumbai-37

Price ₹ 375.00

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

N5316-A

PYTHON PROGRAMMING **ISBN 978-93-5451-148-6**

Third Edition : April 2023
© : Authors

The text of this publication, or any part thereof, should not be reproduced or transmitted in any form or stored in any computer storage system or device for distribution including photocopy, recording, taping or information retrieval system or reproduced on any disc, tape, perforated media or other information storage device etc., without the written permission of Authors with whom the rights are reserved. Breach of this condition is liable for legal action.

Every effort has been made to avoid errors or omissions in this publication. In spite of this, errors may have crept in. Any mistake, error or discrepancy so noted and shall be brought to our notice shall be taken care of in the next edition. It is notified that neither the publisher nor the authors or seller shall be responsible for any damage or loss of action to any one, of any kind, in any manner, therefrom. The reader must cross check all the facts and contents with original Government notification or publications.

Published By :
NIRALI PRAKASHAN
Abhyudaya Pragati, 1312, Shivaji Nagar,
Off J.M. Road, Pune – 411005
Tel - (020) 25512336/37/39
Email : niralipune@pragationline.com

Polyplate

Printed By :
YOGIRAJ PRINTERS AND BINDERS
Survey No. 10/1A, Ghule Industrial Estate
Nanded Gaon Road
Nanded, Pune - 411041

DISTRIBUTION CENTRES

PUNE

Nirali Prakashan
(For orders outside Pune)
S. No. 28/27, Dhayari Narhe Road, Near Asian College
Pune 411041, Maharashtra
Tel : (020) 24690204; Mobile : 9657703143
Email : bookorder@pragationline.com

Nirali Prakashan
(For orders within Pune)
119, Budhwar Peth, Jogeshwari Mandir Lane
Pune 411002, Maharashtra
Mobile : 9657703145, 9890997937
Email : niralilocal@pragationline.com

MUMBAI

Nirali Prakashan
Rasdhara Co-op. Hsg. Society Ltd., 'D' Wing Ground Floor, 385 S.V.P. Road
Girgaum, Mumbai 400004, Maharashtra
Mobile : 7045821020, Tel : (022) 2385 6339 / 2386 9976
Email : niralimumbai@pragationline.com

DISTRIBUTION BRANCHES

DELHI

Nirali Prakashan
Room No. 2 Ground Floor
4575/15 Omkar Tower, Agarwal Road
Darya Ganj, New Delhi 110002
Mobile : 9555778814/9818561840
Email : delhi@niralibooks.com

BENGALURU

Nirali Prakashan
Maitri Ground Floor, Jaya Apartments,
No. 99, 6th Cross, 6th Main,
Malleswaram, Bengaluru 560003
Karnataka; Mob : 9686821074
Email : bengaluru@niralibooks.com

NAGPUR

Nirali Prakashan
Above Maratha Mandir, Shop No. 3,
First Floor, Rani Jhanshi Square,
Sitabuldi Nagpur 440012 (MAH)
Tel : (0712) 254 7129
Email : nagpur@niralibooks.com

KOLHAPUR

Nirali Prakashan
438/2, Bhosale Plaza, Ground Floor
Khasbag, Opp. Balgopal Talim
Kolhapur 416 012, Maharashtra
Mob : 9850046155
Email : kolhapur@niralibooks.com

JALGAON

Nirali Prakashan
34, V. V. Golani Market, Navi Peth,
Jalgaon 425001, Maharashtra
Tel : (0257) 222 0395
Mob : 94234 91860
Email : jalgaon@niralibooks.com

SOLAPUR

Nirali Prakashan
R-158/2, Avanti Nagar, Near Golden
Gate, Pune Naka Chowk
Solapur 413001, Maharashtra
Mobile 9890918687
Email : solapur@niralibooks.com

marketing@pragationline.com | www.niralibooks.com

Also find us on www.facebook.com/niralibooks

Preface ...

We take this opportunity to present this book entitled as **"Python Programming"** to the students of Second Semester - MCA (Management). The object of this book is to present the subject matter in a most concise and simple manner. The book is written strictly according to the New Syllabus (CBCS Pattern).

The book has its own unique features. It brings out the subject in a very simple and lucid manner for easy and comprehensive understanding of the basic concepts, its intricacies, programs and practices. This book will help the readers to have a broader view on Python Programming. The language used in this book is easy and will help students to improve their knowledge of programming and understand the matter in a better and happier way.

We sincerely thank Shri. Dineshbhai Furia and Shri. Jignesh Furia of Nirali Prakashan, for the confidence reposed in us and giving us this opportunity to reach out to the students of management studies.

We have given our best inputs for this book. Any suggestions towards the improvement of this book and sincere comments are most welcome on niralipune@pragationline.com.

Authors

■■■

Syllabus ...

1. **Introduction & Components of Python [Weightage 15%, Sessions 07]**
   - 1.1 Understanding Python
   - 1.2 Role of Python in AI and Data science
   - 1.3 Installation and Working with Python
   - 1.4 The Default Graphical Development Environment for Python - IDLE
   - 1.5 Types and Operation
   - 1.6 Python Object Types - Number, Strings, Lists, Dictionaries, Tuples, Files, User Defined Classes
   - 1.7 Understanding Python Blocks
   - 1.8 Python Program Flow Control
   - 1.9 Conditional Blocks using if, else and elif
   - 1.10 Simple for Loops in Python
   - 1.11 for loop using ranges, string, list and dictionaries
   - 1.12 Use of while loop in Python
   - 1.13 Loop Manipulation using pass, continue, break and else
   - 1.14 Programming using Python conditional and loops Block

   Extra Reading: Python installation with windows, Linux and MAC OS, creating virtual environment, configuring python on EC2 instance, understanding Python IDE – (VSCode, PyCharm, Spyder), Installing Anaconda and Setting-up environment for Python.
2. **Python Functions, Modules & Packages [Weightage 15%, Sessions 07]**
   - 2.1 Function Basics - Scope, nested function, non-local statements
   - 2.2 Built-in functions
   - 2.3 Arguments Passing, Anonymous Function: lambda
   - 2.4 Decorators and Generators
   - 2.5 Module basic usage, namespaces, reloading modules - math, random, datetime, etc.
   - 2.6 Package: import basics
   - 2.7 Python namespace packages
   - 2.8 User defined modules and packages

   Extra Readings: GUI framework in Python
3. **Python Object Oriented Programming [Weightage 15%, Sessions 06]**
   - 3.1 Concept of Class, Object and Instances, Method call
   - 3.2 Constructor, class attributes and destructors
   - 3.3 Real time use of class in live projects
   - 3.4 Inheritance, super class and overloading operators
   - 3.5 Static and Class Methods
   - 3.6 Adding and retrieving dynamic attributes of classes
   - 3.7 Programming using OOPS
   - 3.8 Delegation and Container

   Extra Readings: Integrating GUI framework with OOP

4. Python Regular Expression [Weightage 10%, Sessions 04]

4.1 Powerful Pattern matching and searching

4.2 Power of Pattern Searching using regex in Python

4.3 Real time parsing of data using regex

4.4 Password, e-mail, URL validation using regular expression

4.5 Pattern finding programs using regular expression

Extra Readings: Web scrapping and pattern matching with regex

5. Python Multithreading and Exception Handling [Weightage 10%, Sessions 05]

5.1 Exception Handling

5.2 Avoiding code break using exception handling

5.3 Safe guarding file operation using exception handling

5.4 Handling and helping developer with error code

5.5 Programming using Exception handling

5.6 Multithreading

5.7 Understanding threads

5.8 Synchronizing the threads

5.9 Programming using Multithreading

Extra Readings: Multiprocessing, deadlock, synchronization, monitors and messaging queue

6. Python File Operation [Weightage 05%, Sessions 02]

6.1 Reading config files in Python

6.2 Writing log files in Python

6.3 Understanding read functions, read(), readline() and readlines()

6.4 Understanding write

6.5 Functions write() and writelines()

6.6 Manipulating file pointer using seek

6.7 Programming using file operations

Extra Readings: Reading and Writing the files on AWS S3 bucket

7. Python Database Interaction [Weightage 10%, Sessions 05]

7.1 Introduction to NoSQL database

7.2 Advantages of NoSQL database

7.3 SQL vs. NoSQL

7.4 Introduction to MongoDB with Python

7.5 Exploring Collections and Documents

7.6 Performing basic CRUD operations with MongoDB and Python

Extra Readings: Graph database like Neo4j with python

8. Python for Data Analysis [Weightage 20%, Sessions 09]

8.1 NumPy
8.2 Introduction to NumPy
8.3 Creating arrays, Using arrays and Scalars
8.4 Indexing Arrays, Array Transposition
8.5 Universal Array Function
8.6 Array Input and Output
8.7 Pandas
8.8 What are Pandas? Where it is used?
8.9 Series in pandas, pandas DataFrames, Index objects, Relndex
8.10 Drop Entry, Selecting Entries
8.11 Data Alignment, Rank and Sort
8.12 Summary Statics, Missing Data, Index Hierarchy
8.13 Matplotlib
8.14 Python for Data Visualization
8.15 Introduction to Matplotlib
8.16 Visualization Tools

Extra Readings: Text analytics with NLP and Python

■■■

Contents ...

1. **Introduction & Components of Python** 1.1 – 1.96

2. **Python Functions, Modules & Packages** 2.1 – 2.42

3. **Python Object Oriented Programming** 3.1 – 3.28

4. **Python Regular Expression** 4.1 – 4.18

5. **Python Multithreading and Exception Handling** 5.1 – 5.24

6. **Python File Operation** 6.1 – 6.18

7. **Python Database Interaction** 7.1 – 7.18

8. **Python for Data Analysis** 8.1 – 8.54

Summer 2022 P.1 – P.4

■■■

1...

Introduction & Components of Python

Objectives...

- To understand basic concepts in Python Programming.
- To learn Features and Environment for Python Programming.
- To learn Data Types in Python Programming.
- To study Looping in Python Programming.
- To understand Loop Manipulation Statements in Python.

1.1 UNDERSTANDING PYTHON

- Python is a high-level, interpreted, interactive and object-oriented programming language. Today, Python is the trendiest programming language.
- Python is a popular programming language because it provides more reliability of code, clean syntax of code, advanced language features, scalability of code, portability of code, support object-oriented programming, broad standard library, easy to learn and read, supports GUI mode, interactive, versatile and interpreted, interfaces to all major commercial databases, and so on.

1.1.1 History of Python Programming Language

- Python laid its foundation in the late 1980s. Python was developed by Guido Van Rossum at National Research Institute for Mathematics and Computer Science in Netherlands in 1990.
- Inspired by Monty Python's Flying Circus, a BBC comedy series, he named the language Python.
- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small-Talk, UNIX shell and other scripting languages.
- ABC programming language is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System. Like Perl, Python source code is now available under the GNU General Public License (GPL).

- In February 1991, Guido Van Rossum published Python 0.9.0 (first release) to **alt.sources**. In addition to exception handling, Python included classes, lists and strings.
- In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce which aligned it heavily in relation to functional programming.
- Python 2.0 added new features like list comprehensions, garbage collection system and it supported Unicode.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify the fundamental errors of the language. In Python 3.0, the print statement has been replaced with a `print( )` function.
- Python widely used in both industry and academic circles because of its simple, concise and extensive support of libraries.
- Python is available for almost all operating systems such as Windows, Mac, Linux/UNIX etc. Python can be downloading from **http://www.python.org/downloads**.

1.1.2 Applications of Python Programming

- Google's App Engine web development framework uses Python as an application language.
- Maya, a powerful integrated 3D modeling and animation system provides a Python scripting API.
- Linux Weekly News published by using a web application written in Python programming.
- Google makes extensive use of Python in its Web Search Systems.
- The popular YouTube Video sharing service is largely written in Python programming.
- The NSA (National Security Agency) uses Python programming for cryptography and intelligence analysis.
- iRobot uses Python programming to develop commercial and military robotic devices.
- The Raspberry Pi single-board computer promotes Python programming as its educational language.
- Nextflix and Yelp have both documented the role of Python in their software infrastructures.
- Industrial Light and Magic, Pixar and others uses Python programming in the production of animated movies.

1.1.3 Features of Python

1. **Easy to Learn and Use:** Python is easy to learn and use. It is developer-friendly and high level programming language. It has few keywords, simple structure, and a clearly defined syntax that makes it easily understandable for beginners. Python language is more expressive means that it is more understandable and

readable for programmers. In Python programming, programs are easy to write and execute as it omits some cumbersome, poorly understandable and confusing features of other programming languages such as C++ and Java.

2. **Interpreted Language:** Python is an interpreted language. The interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners. There are excellent, straightforward tools to work with Python code, especially interactive interpreter. In Python, we need not to learn a build system, IDE, special text editor, or anything else to start using Python. All we need only a command prompt and an interactive editor.

   Python provides a Python Shell (also known as Python Interactive Shell) which is used to execute a single Python command and get the result as shown below. If Python is installed on the PC then to open the Python Shell on Windows, open the command prompt, write Python and press enter. As we can see, a Python Prompt comprising three Greater Than symbols (>>>) appears as shown in Fig. 1.1.

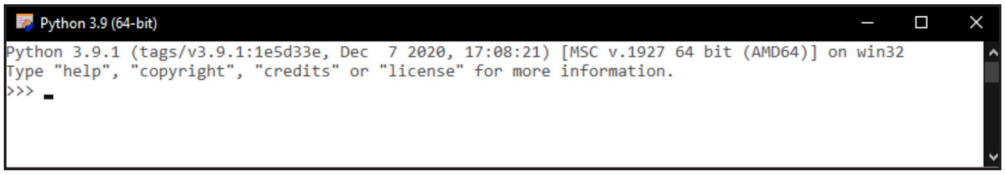


Fig. 1.1: Python Command Prompt

3. **Interactive Mode:** Python programming language has support for interactive mode, which allows interactive testing and debugging of code. Graphical User Interfaces (GUIs) can be developed using Python. Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh and the X Window system of UNIX. There are many free and commercial editors available for Python. Following table lists Python Editors.

Table 1.1: Python Editors

Editor	Description	Logo
IDLE	IDLE is a popular Integrated Development Environment written in Python and it has been integrated with the default language. Mainly used by beginner's level developers who want to practice Python development.	
PyCharm	PyCharm is one of the widely used Python IDE which was created by Jet Brains. With PyCharm, the developers can write a neat and maintainable code. It helps to be more productive and gives smart assistance to the developers. It takes care of the routine tasks by saving time and thereby increasing profit accordingly.	

contd. ...

Spyder	It was mainly developed for scientists and engineers to provide a powerful scientific environment for Python. It offers an advanced level of edit, debug, and data exploration features. It is very extensible and has a good plugin system and API.	
PyDev	PyDev is an outside plugin for Eclipse. It is basically an IDE that is used for Python development. It is linear in size. It mainly focuses on the refactoring of python code, debugging in the graphical pattern, analysis of code etc. It is a strong Python interpreter.	
Jupyter Notebook	The Jupyter Notebook is a browser-based graphical interface to the IPython shell. Allows us to create and share documents that contain live code, equations, visualizations and narrative text.	

In this book, IDLE is used for Python programming. IDLE (Integrated Development and Learning Environment) is an Integrated Development Environment (IDE) for Python. To start IDLE interactive shell, search for the IDLE icon in the start menu and double click on it and we will get the following window.

In Python IDLE shell not only we can execute commands one by one like in Python Command Prompt but also can create **.py** files and see execution of those files.

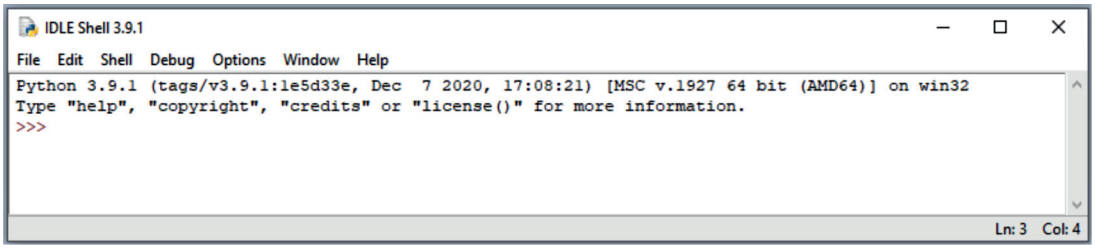


Fig. 1.2: Python Shell

4. **Free and Open Source:** Python programming language is developed under an OSI approved open source license, making it freely available at official web address. The source code is also available for use. The Python software can be freely distributed and anyone can use and read its source code make changes/modify and use the pieces in new free programs.
5. **Platform Independence/Cross Platform Language/Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms. Python can run equally on different platforms such as Windows, Linux, UNIX and Macintosh etc. So, we can say that Python is a portable language. Fig.1.3 shows execution of Python code by interpreter.

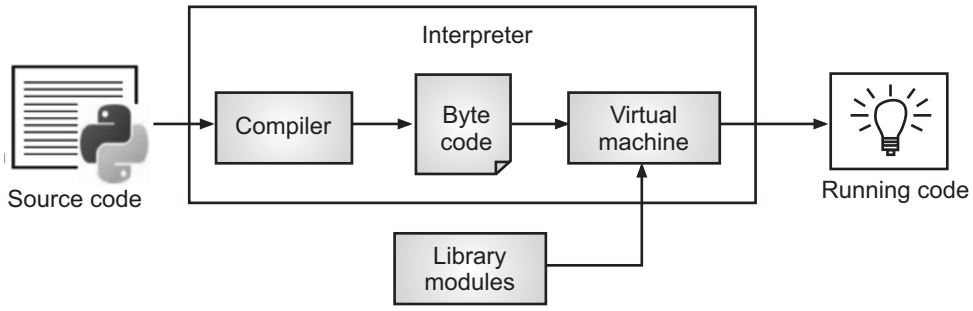


Fig. 1.3: Execution of Python Code

Python source code goes through Compiler which compiles the **source code** into a **byte code**. Byte code is a lower level, platform independent, efficient and intermediate representation of the source code. As soon as source code gets converted to byte code, it is fed into **PVM (Python Virtual Machine).** The PVM is the runtime engine of Python; it's always present as part of the Python system and is the component that truly runs the scripts. Technically, it's just the last step of what is called the Python interpreter.

6. **Object-Oriented Language:** A programming language that can model the real world is said to be object-oriented. It focuses on objects and combines data and functions. Contrarily, a procedure-oriented language revolves around functions, which are codes that can be reused. Python supports both procedure-oriented and object-oriented programming, which is one of the key Python features. It also supports multiple inheritance, unlike Java. Python has a powerful but simplistic way of doing OOP, especially when compared to C++ and Java languages. Python supports object-oriented language and concepts of classes and objects come into existence.
7. **Extensible:** Python programming implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in the python code. Python has a large and broad library and provides a rich set of modules and functions for rapid application development. Python languages bulk library is portable and cross-platform compatible with UNIX, Windows etc.

1.1.4 Limitations of Python

- Python is an interpreter based language. Therefore, it is a bit slower than compiler based languages.
- Python is a high level language like C/C++/Java. It also uses many layers to communicate with the operating system and the computer hardware.
- Graphics intensive applications such as games make the program to run slower.
- Due to the flexibility of the data types, Python's memory consumption is also high.

1.1.5 Structure of a Python Program

- Fig. 1.4 shows a typical program structure of Python programming.

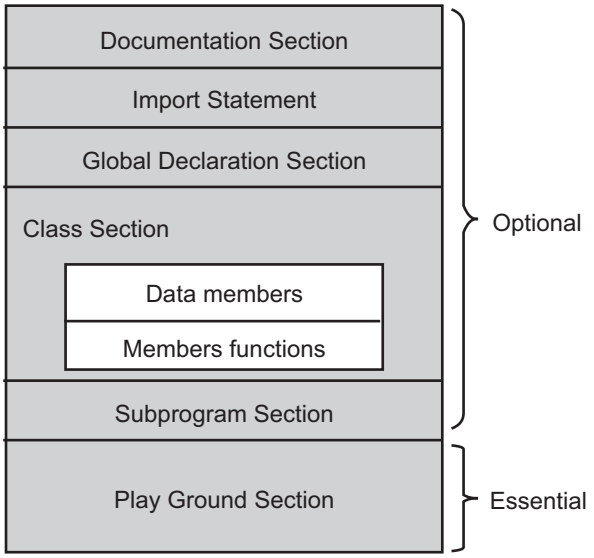


Fig. 1.4: Program structure of Python programming

- Python programs are structured as a sequence of statements. A **Python statement** is smallest program unit. Statements are the instructions that are written in a program to perform a specific task. A Python statement is a complete instruction executed by the Python interpreter. By default, the Python interpreter executes all statements sequentially, but we can change the order of execution using control statements.
- Program structure of Python programming contains following sections:
  - **Documentation Section** includes the comments that specify the purpose of the program. Comments are a non-executable statement that is ignored by the compiler while program execution. Python comments are written anywhere in the program.
  - **Import Statement Section** is used includes different built-in or user defined modules.
  - **Global Declaration Section** is used to define the global variables for the programs.
  - **Class Section** describes the information about the user defined classes in the Python program. A class is a collection of data members and member functions called method that operates on data members.
  - **Subprogram Section** includes user defined functions. The functions include the set of statements that need to be executed when the function is called from anywhere.
  - **Play Ground Section** is the main section of Python program and the main section starts where the function calling.

1.1.6 Running Python Script

- Python has two basic modes, namely normal and interactive.
  - The normal **script mode** is the mode where the scripted and finished .py files are run in the Python interpreter.
  - The **interactive mode** is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory.
  - As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole.

1. **Interactive Mode:** Interactive mode is used for quickly and conveniently running single line or blocks of code. Here's an example using the Python shell that comes with a basic Python installation. The ">>>" indicates that the shell is ready to accept interactive commands. For example, if we want to print the statement "Interactive Mode", simply type the appropriate code and hit enter.

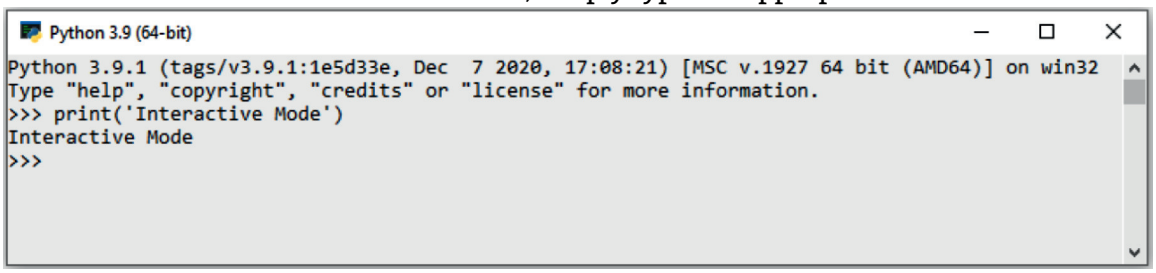


Fig. 1.5 (a): Interactive Mode

2. **Script Mode:** In the standard Python shell, we can go to "File" → "New File" (or just hit Ctrl + N) to pull up a blank script to write the code. Then save the script with a ".py" extension. We can save it anywhere we want for now, though we may want to make a folder somewhere to store the code as we test Python output. To run the script, either select "Run" → "Run Module" or press F5 key.

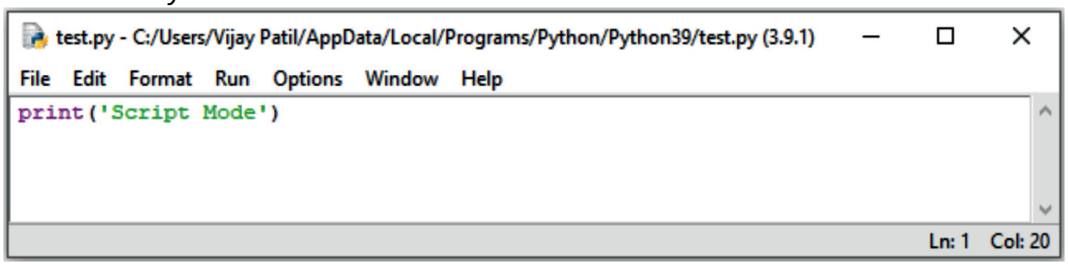


Fig. 1.5 (b): Script Mode

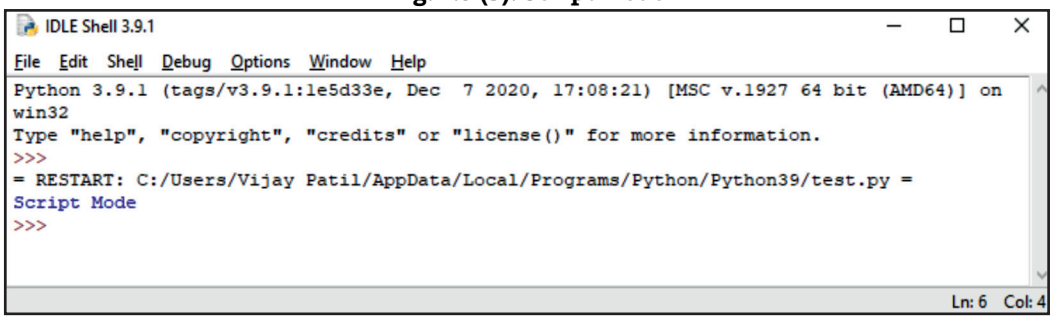


Fig. 1.5 (c) : Execute Script

1.1.7 Internal Working of Python

- Python is an object-oriented programming language like C++ and Java. Python is called an interpreted language means Python programs are executed by the interpreter. Python uses code modules that are interchangeable instead of a single long list of instructions that was standard for functional programming languages.
- The standard implementation of Python is called "CPython". It is the default and widely used implementation of Python. When a programmer tries to run a Python code as instructions in an interactive manner in a Python shell, then Python performs various operations internally. All such internal operations can be broken down into a series of steps as shown in Fig. 1.6.

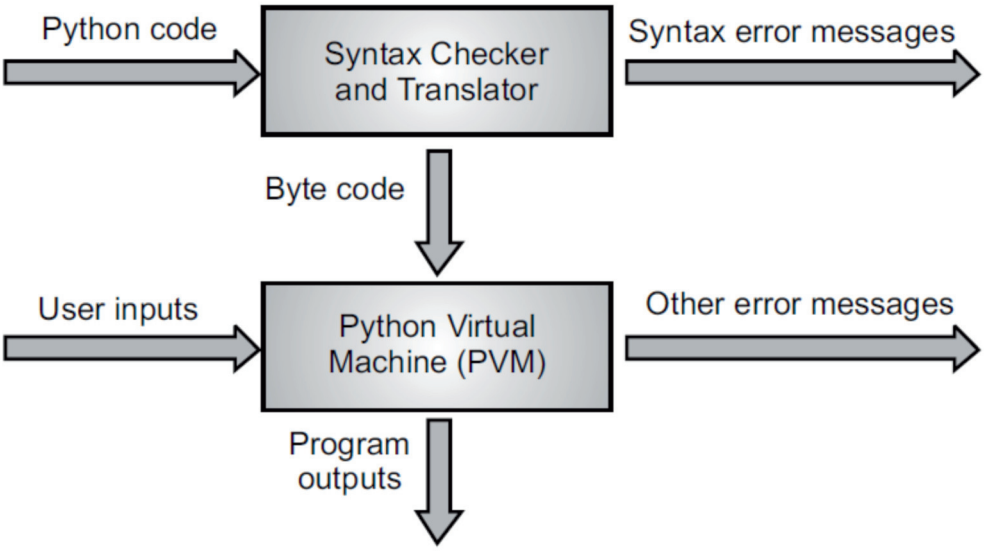


Fig. 1.6: Internal Operations

- The Python interpreter performs the following tasks to execute a Python program:
  1. The interpreter reads a Python expression or statement, also called the source code, and verifies that it is well formed. In this step, as soon as the interpreter encounters such an error, it halts translation with an error message.
  2. If a Python expression is well formed, the interpreter then translates it to an equivalent form in a low-level language called byte code. When the interpreter runs a script, it completely translates it to byte code.
  3. This byte code is next sent to another software component, called the Python Virtual Machine (PVM), where it is executed. If another error occurs during this step, execution also halts with an error message.

1.2 ROLE OF PYTHON IN AI AND DATA SCIENCE

- Artificial Intelligence (AI) is an area of computer science that gives priority to creating intelligent machines that can work like human-beings such as speech recognition, visual perception, translation, and even decision making. *NumPy* for

scientific computation, *PyBrain* for Python Machine Language, *SciPy* for advanced Computing, are the libraries to make Python is the suitable language for AI. IDE (Integrated Development Environment) supports to avoid struggles of developers with the use of many algorithms.

- Python has various library packages to develop and decode the AI project and Data Science. Some of the specialized packages as follows:
  - **NumPy:** NumPy stands for Numerical Python. NumPy is a Python library that provides mathematical functions to handle large dimension arrays. It is used as a container for generic data comprising of an N-Dimensional Array Object, Integrating tools for C/C++ Codes, Random Number Capacity, and other functions. The library provides vectorization of mathematical operations on the NumPy array type, which enhances performance and speeds up the execution.
  - **Pandas:** It gives easy-to-use data structures and analytics tools for Python and it is an open-source library. Pandas is one of the most popular Python library for data manipulation and analysis. Pandas provide useful functions to manipulate large amount of structured data. Pandas provide the easiest method to perform analysis. It provides large data structures and manipulating numerical tables and time series data. Pandas is a perfect tool for data wrangling. Pandas is designed for quick and easy data manipulation, aggregation, and visualization. There are two data structures in Pandas:
    - **Series:** It handles and stores data in one-dimensional data.
    - **DataFrame:** It handles and stores two-dimensional data.
  - **Matplotlib:** It is a 2D plotting library to create publication-quality images. Matplotlib is another useful Python library for Data Visualization. Matplotlib allows making quick line graphs, pie charts, histograms, and other professional grade figures. Matplotlib has interactive features like zooming and planning and saving the Graph in graphics format.
  - **SciPy:** SciPy is another popular Python library for data science and scientific computing. SciPy provides great functionality to scientific mathematics and computing programming. SciPy contains sub-modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers, Statmodel and other tasks common in science and engineering.
  - **AIMA:** "Artificial Intelligence – A Modern Approach" by Russel and Norvig used for implementing algorithms for general AI.
  - **pyDatalog:** It is a logic programming engine in Python.
  - **EasyAI:** It used to create a simple Python engine for two-player games with AI. (Ex: Negamax)

- **PyBrain:** A flexible, simple but effective algorithm for Machine Learning which provides predefined environments to test and compare algorithms.
- **PyML:** It is a bilateral framework for SVMs and other kernel methods. And also it supported on Linux and Mac OS X.
- **Scikit-Learn/ Sklearn:** It is the most popular library for Machine Learning that is a highly effective tool for data analysis. Sklearn is built on NumPy, SciPy, and matplotlib. Sklearn provides easy and simple tools for data mining and data analysis. It provides a set of common machine learning algorithms to users through a consistent interface. Scikit-Learn help to quickly implement popular algorithms on datasets and solve real-world problems.
- **MDP-Toolkit:** It has both supervised and unsupervised learning algorithm which can be expanded to the use of network architecture.
- **NLTK:** Natural Language Toolkit (NLTK) is used to do linguistic data and documentation for research and development in text analysis and Natural Language Processing.

1.3 INSTALLATION AND WORKING WITH PYTHON

- Python distribution is available for a wide variety of platforms such as UNIX, Linux, Macintosh and Windows. We need to download only the binary code applicable for the platform and install Python. The most up-to-date and current source code, binaries, documentation, news, etc. is available on the official website of Python https://www.python.org/.

1.3.1 Installation of Python in Windows

Step 1 : Open any internet browser then type **http:///www.python.org/downloads/** in address bar and Enter. The Home page will appear as shown in Fig. 1.7 (a).

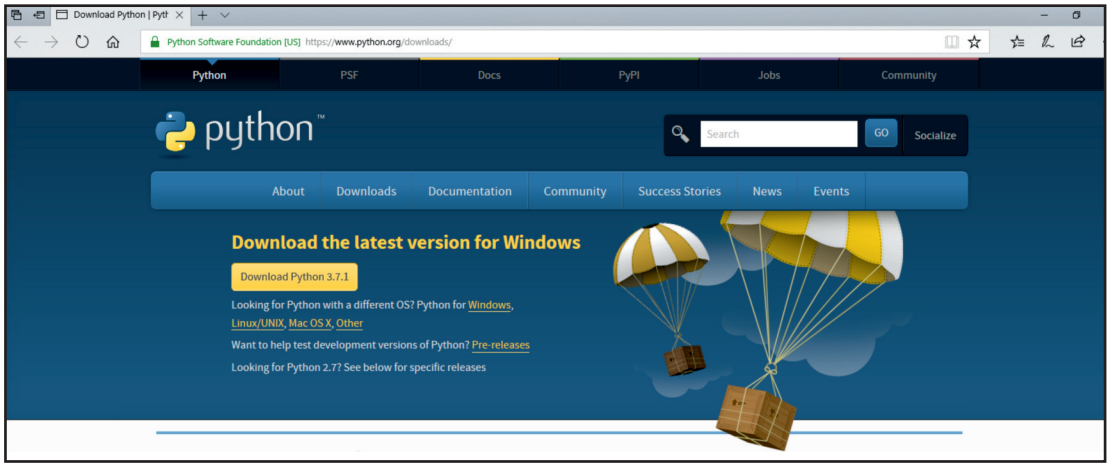


Fig. 1.7 (a): Home Page

Step 2 : Click on download to get the latest version of Python.



Fig. 1.7 (b): Python Release Versions

Step 3 : Once Python version is demanded (e.g. 3.7.1. User can download latest versions). Open the Python 3.7.1 version pack and double click on it to start installation.



Fig. 1.7 (c): Python Installation

Step 4 : Click on "install now" option for installation.

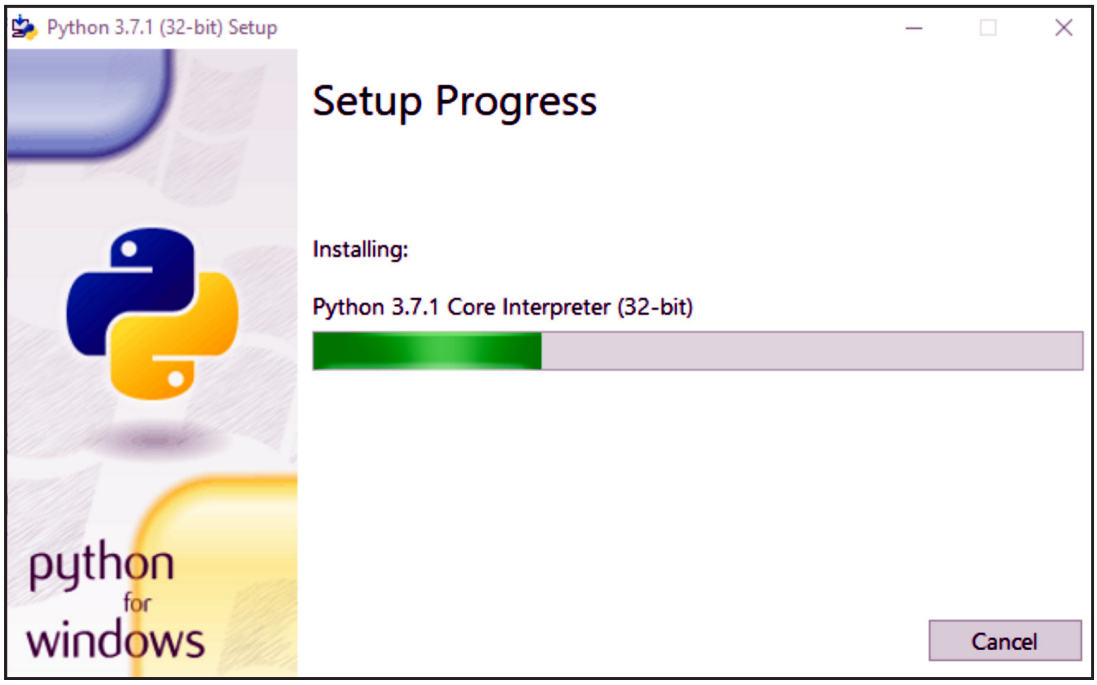


Fig. 1.7 (d): Setup Progress

Step 5 : After complete the installation close the windows.

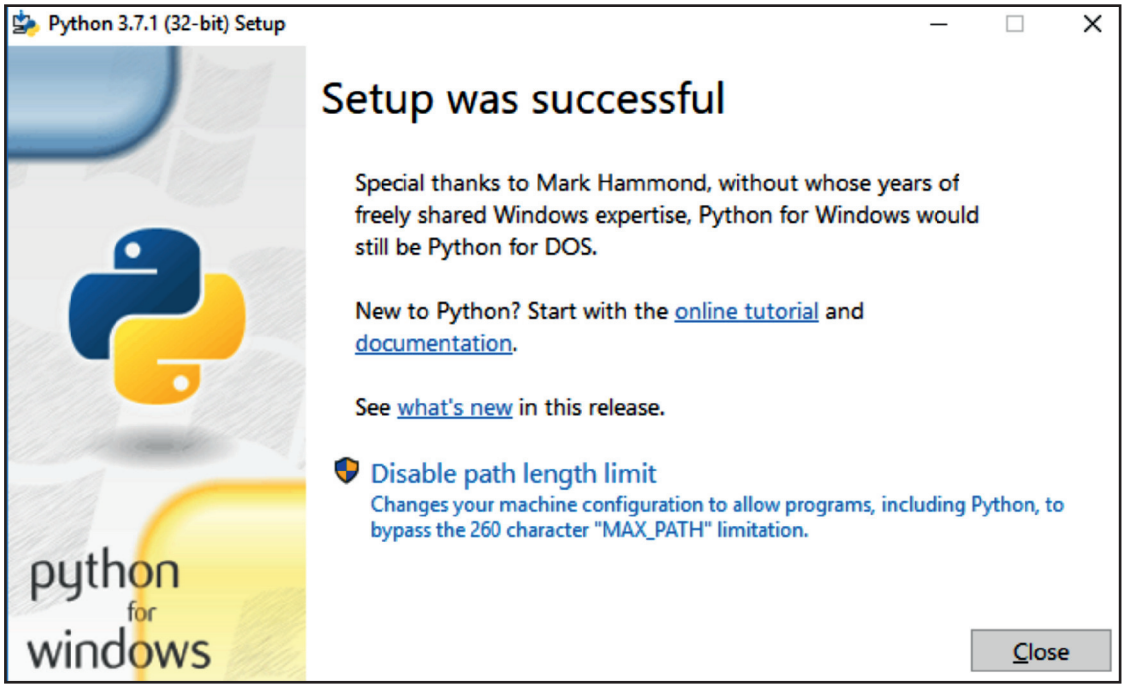


Fig. 1.7 (e): Complete the Installation

1.3.2 Starting Python in Command Line Modes

- Python script can be executed at command line also. This can be done by invoking the interpreter on the application. In command line mode, we type the Python programming program on the Python shell and the interpreter prints the result. These steps are given below:

Step 1 : Press Start button.

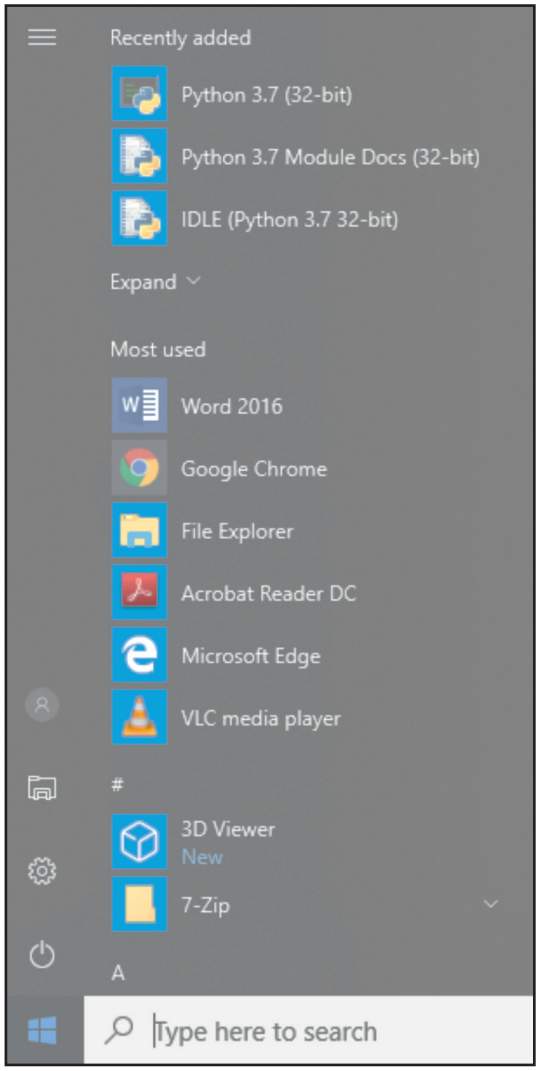


Fig. 1.8 (a): Python 3.7 (32-bit) options

Step 2 : Click on **All programs** section and then click on Python 3.7 (32-bit). You will see the Python interactive prompt in Python command line.

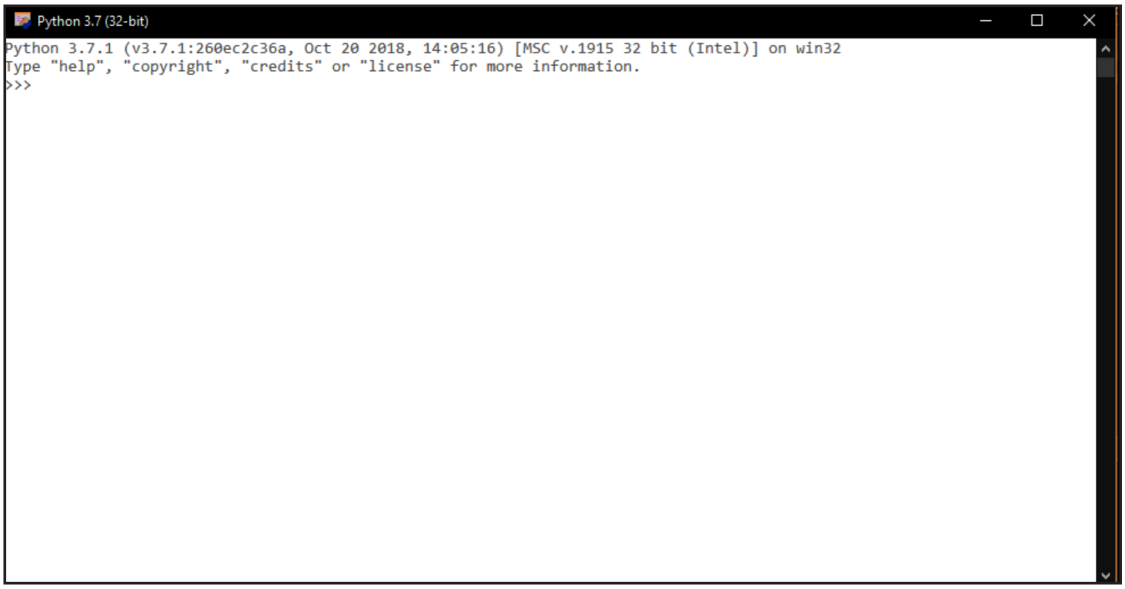


Fig. 1.8 (b): Command Prompt

Python command line contains an opening message >>> called Command prompt. The cursor at command prompt waits for to enter Python command. A complete command is called a statement. For example, check first command to print message.

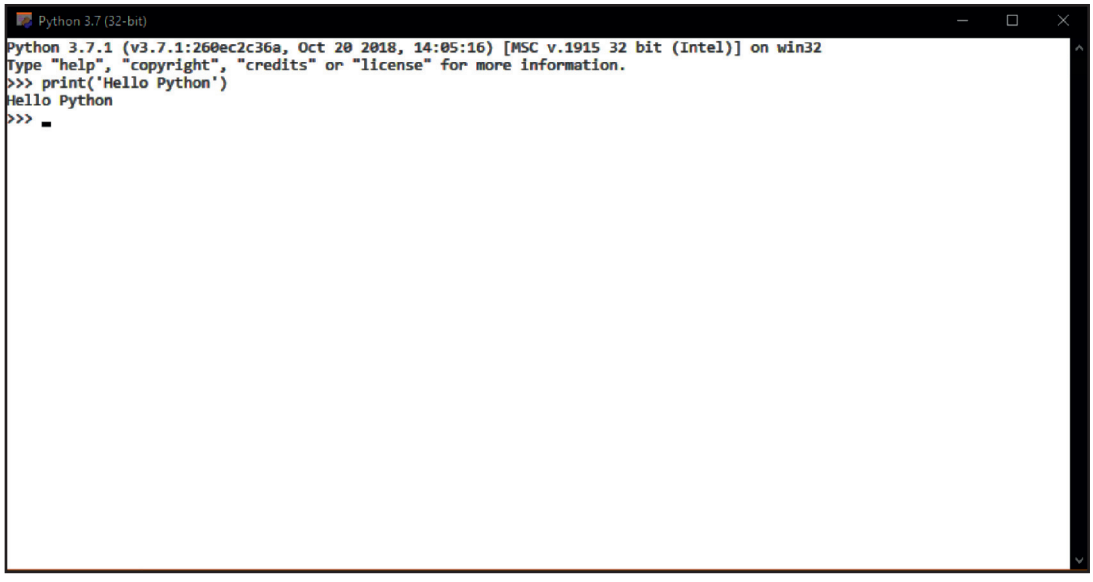


Fig. 1.8 (c): Python command line

Step 3 : To exit from the command line of Python, press Ctrl+Z keys followed by Enter key or type exit() or quit() and press Enter key.

1.4 THE DEFAULT GRAPHICAL DEVELOPMENT ENVIRONMENT FOR PYTHON - IDLE

- **Starting Python IDLE:** When we install Python3, we also get IDLE (Integrated Development Environment). IDLE includes a color syntax-highlighting editor, a debugger, the Python Shell, and a complete copy of Python3's online documentation set. The steps are given below:

  Step 1 : Press start button and click on IDLE (Python 3.7 32-bit) options.

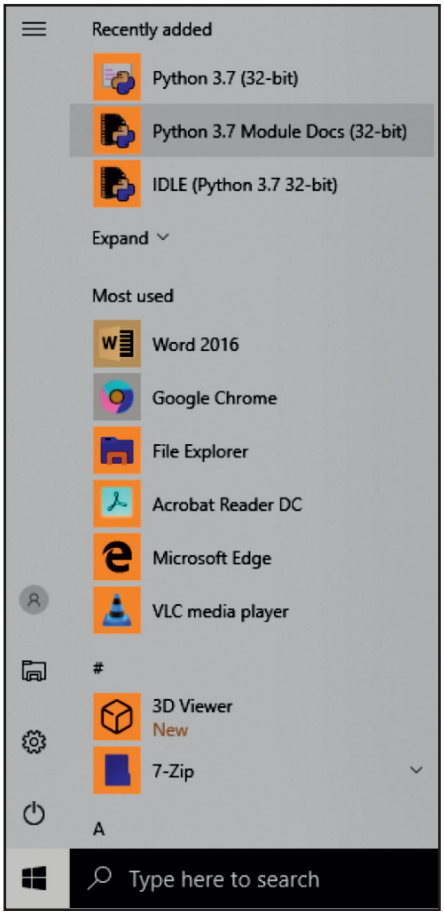


Fig. 1.9 (a): IDLE (Python 3.7 32-bit) options

Step 2 : We will see the Python interactive prompt i.e. interactive shell.

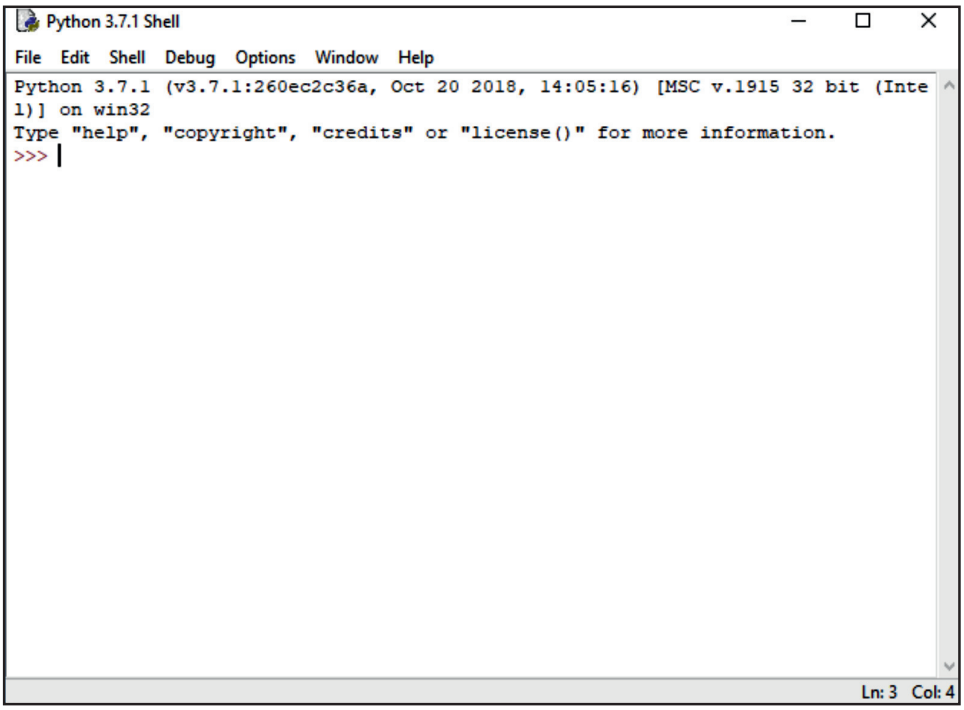


Fig. 1.9 (b): Python Interactive Prompt

- Python interactive shell prompt contains opening message >>>, called Shell Prompt. A cursor is waiting for the command. A complete command is called a statement. When you write a command and press enter, the python interpreter will immediately display the result.

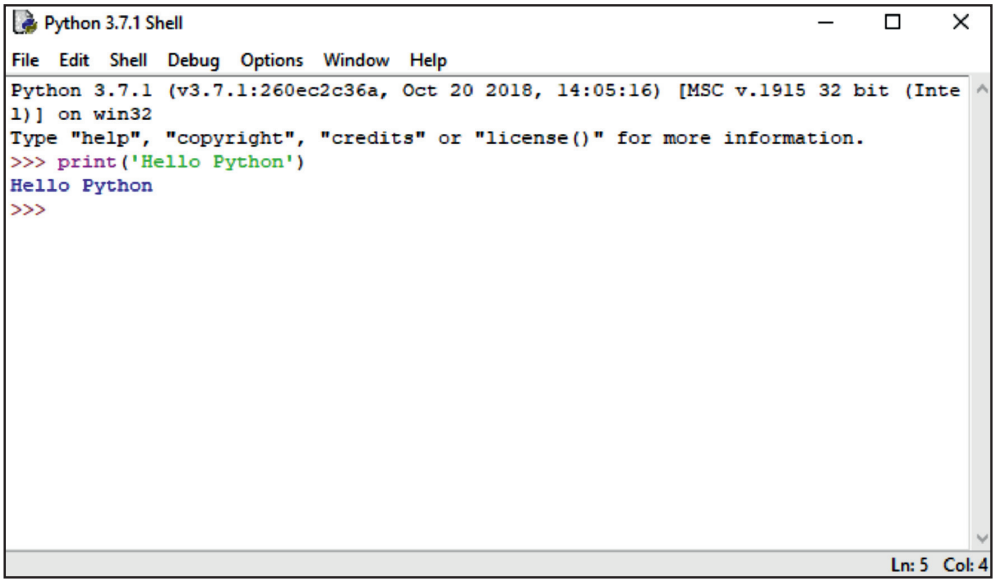


Fig. 1.9 (c): Display result by Interpreter

Executing Python Programs:

- In Python IDLEs-shell window, click on **File menu**, and then on **New File** option or press **Ctrl+N keys.**

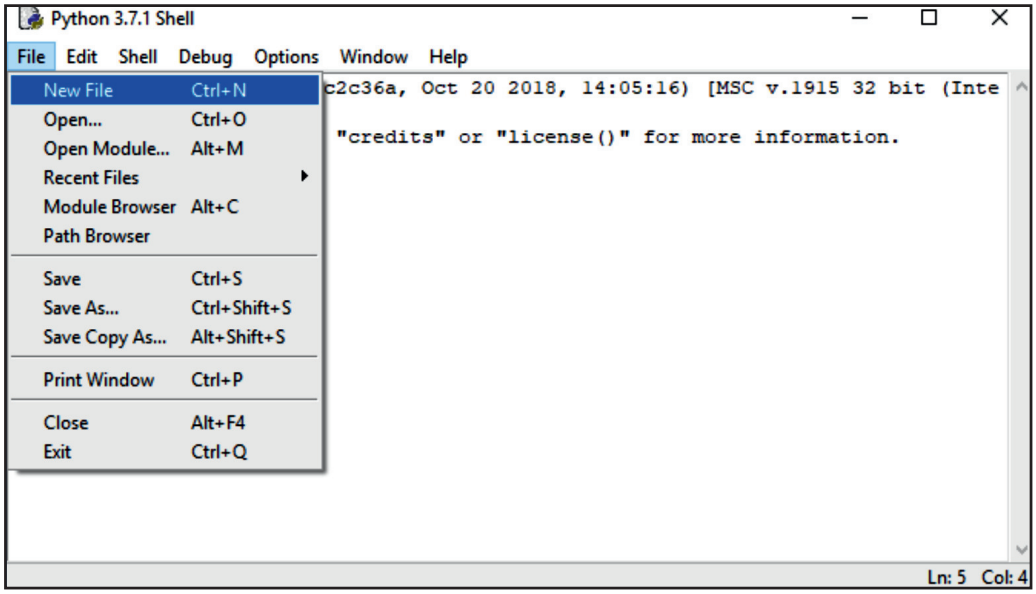


Fig. 1.9 (d): New file

- As soon as you click on **New File**, the window appears as:

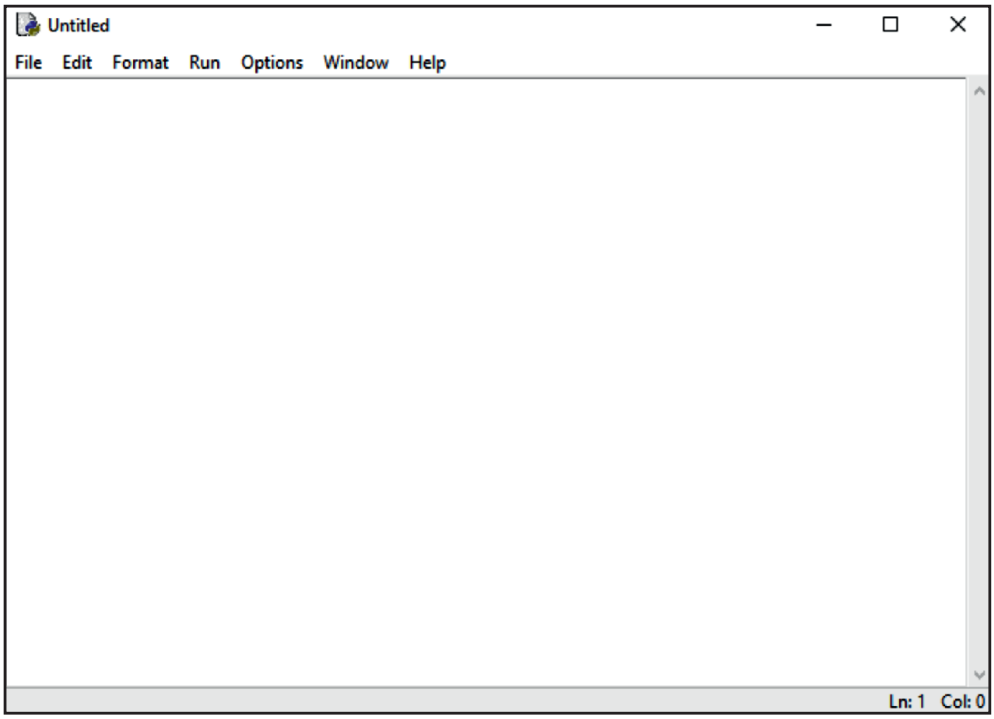


Fig. 1.9 (e): Script Mode

- Write Python program in script mode.

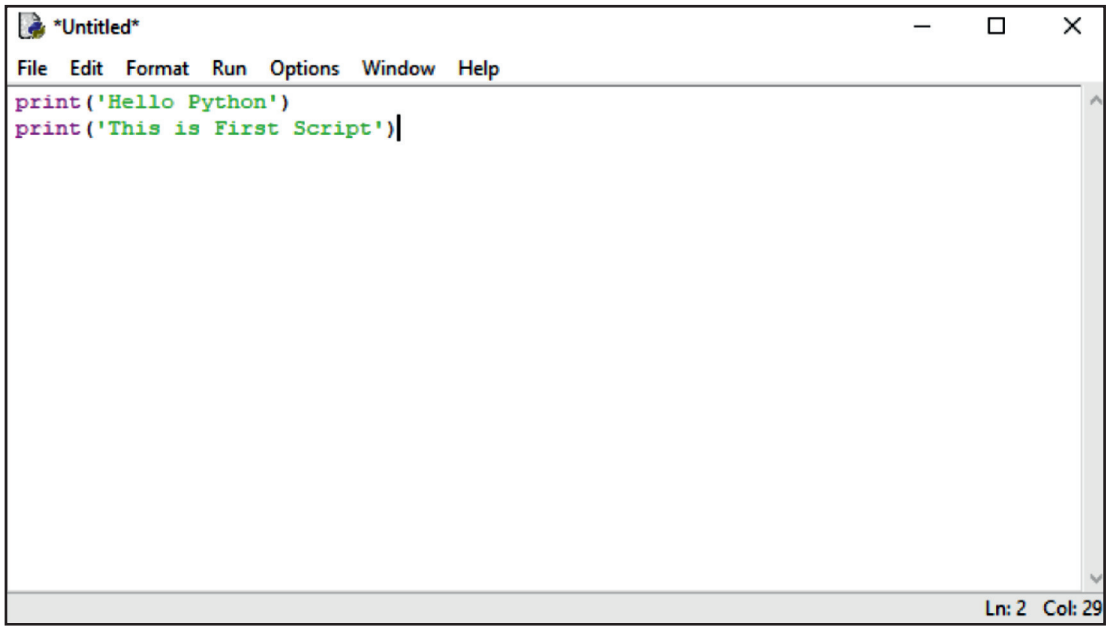


Fig. 1.9 (f): Program in script mode

- Save the above code with filename. By default, Python interpreter will save it using the filename.py. Here we save the script with file name test.py.

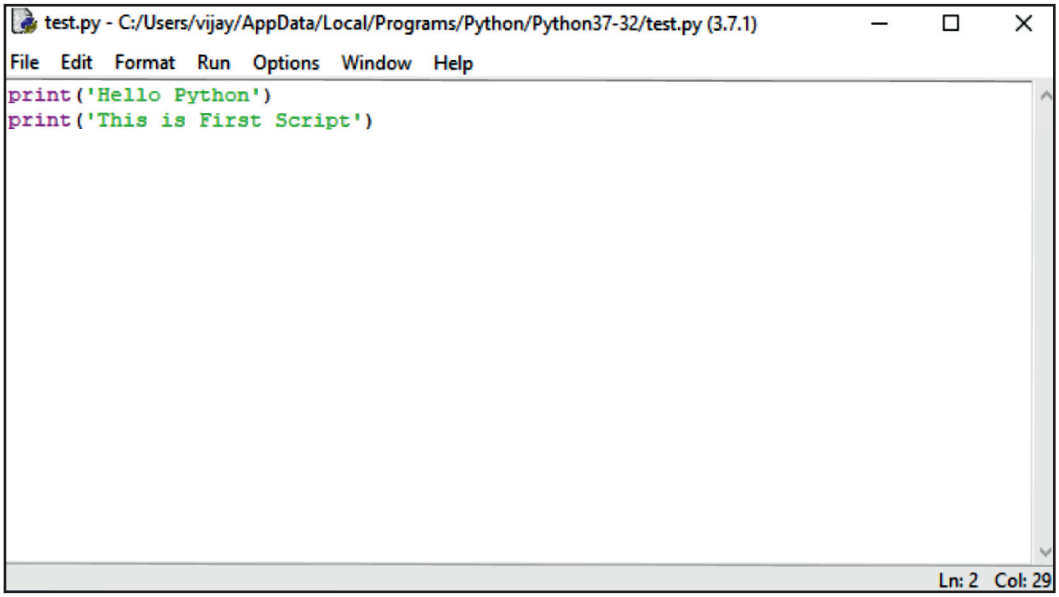


Fig. 1.9 (g): Save the script with file name

- To run the Python program, click on **Run** and then **Run Module** option or we can press **Ctrl+F5** keys.

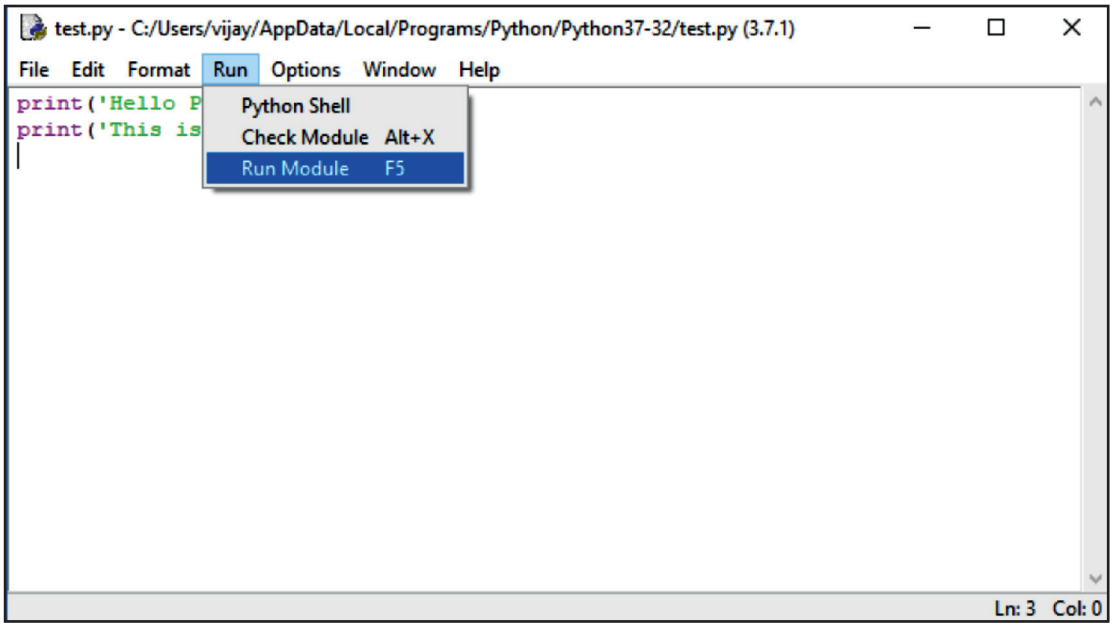


Fig. 1.9 (h): Run Module

- After clicking Run Module, we will get the output of program on Python Shell.

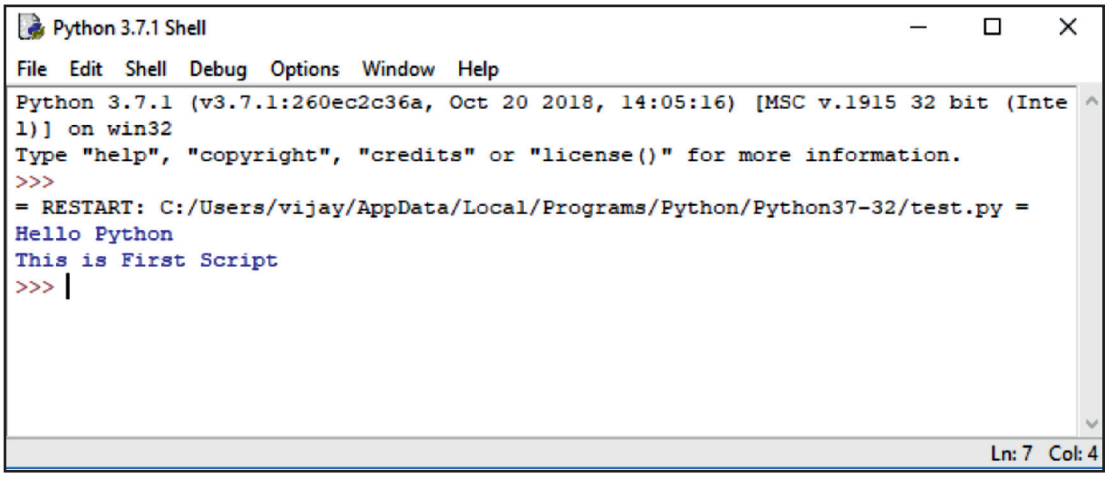


Fig. 1.9 (i): Output on Python shell

1.5 TYPES AND OPERATION

- An operator is a symbol that specifies a specific action. An operator is a special symbol that tells the interpreter to perform a specific operation on the operands. The operands can be literals, variables or expressions.
- An operand is a data item on which operator acts. Operators are the symbol, which can manipulate the value of operands. Some operators require two operands while others require only one.
- Consider the expression 5 + 2 = 7. Here, 5 and 2 are called the operands and + is called the operator.

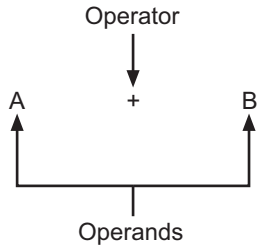


Fig. 1.10: Concept of Operator and Operands

- In Python, the operators can be unary operators or binary operator.

Unary Operators:

- Unary operators are operators with only one operand. These operators are basically used to provide sign to the operand. +, –, ~ are called unary operators.

  Syntax: `operator operand`

  Example:

```
>>> x=10
>>> +x
10
>>> -x
-10
>>> ~x
-11
```

- The invert (~) operator returns the bitwise inversion of long integer arguments. Inversion of x can be computed as ~ (x + 1).

Binary Operators:

- Binary operators are operators with two operands that are manipulated to get the result. They are also used to compare numeric values and string values.
- **Syntax:** `operand1 operator operand2`

  Binary operators are: **, *, /, %, +, -, <<, >>, &, |, ^, <, >, <=, >=, ==, !=, <>.

  Example:

```
>>> x=10
>>> y=20
>>> x+y
30
>>> -x
-10
>>> 2+3
5
```

Expression:

- An expression is nothing but a combination of operators, variables, constants and function calls that result in a value. In other words, an expression is a combination of literals, variables and operators that Python evaluates to produce a value.

  Example:

```
1 + 8
(3 * 9) / 5
a * b + c * 3
```

- Python operators allow programmers to manipulate data or operands. The types of operator supported by Python includes Arithmetic operators, Assignment operators, Relational or Comparison operators, Logical operators, Bitwise operators, Identity operators and Membership operators.

1.5.1 Arithmetic Operators

- The arithmetic operators perform basic arithmetic operations like addition, subtraction, multiplication and division. All arithmetic operators are binary operators because they can perform operations on two operands.
- There are seven arithmetic operators provided in Python programming such as addition, subtraction, multiplication, division, modulus, floor division, and exponential operators.
- Assume variable a holds the value 10 and variable b holds the value 20.

Table 1.2: Arithmetic Operators

Operator Type	Operator	Description	Example
+	Addition	Adds the value of the left and right operands	>>> a + b 30
–	Subtraction	Subtracts the value of the right operand from the value of the left operand	>>> b - a 10
*	Multiplication	Multiplies the value of the left and right operand	>>> a * b 200
/	Division	Divides the value of the left operand by the right operand	>>> b / a 2.0
**	Exponent	Performs exponential calculation	>>> a ** 2 100
%	Modulus	Returns the remainder after dividing the left operand with the right operand	>>> a % b 10
//	Floor Division	Division of operands where the solution is a quotient left after removing decimal numbers	>>> b // a 2

1.5.2 Assignment Operators (Augmented Assignment Operators)

- Assignment operators are used in Python programming to assign values to variables. The assignment operator is used to store the value on the right hand side of the expression on the left hand side variable in the expression.
- For example, a = 5 is a simple assignment operator that assigns the value 5 on the right to the variable a on the left.
- There are various compound operators in Python like a += 5 that adds to the variable and later assigns the same. It is equivalent to a = a + 5.
- Following table shows assignment operators in Python programming:

Table 1.3: Assignment Operators

Operator	Description	Example
=	Assigns values from right side operands to left side operand.	c = a + b assigns value of a + b into c
+=	It adds right operand to the left operand and assigns the result to left operand.	c += a is equivalent to c = c + a
-=	It subtracts right operand from the left operand and assigns the result to left operand.	c -= a is equivalent to c = c - a
*=	It multiplies right operand with the left operand and assigns the result to left operand.	c *= a is equivalent to c = c * a
/=	It divides left operand with the right operand and assigns the result to left operand.	c /= a is equivalent to c = c / a
%=	It takes modulus using two operands and assigns the result to left operand.	c %= a is equivalent to c = c % a
**=	Performs exponential (power) calculation on operators and assign value to the left operand.	c **= a is equivalent to c = c ** a
//=	Performs exponential (power) calculation on operators and assign value to the left operand.	c //= a is equivalent to c = c // a

1.5.3 Relational or Comparison Operators

- Comparison operators in Python programming are binary operators and used to compare values. Relational operators either return True or False according to the condition.
- Assume variable a hold the value 10 and variable b hold the value 20.

Table 1.4: Relational or Comparison Operators

Operator	Description	Example
== (Equality Operator)	If the values of two operands are equal, then the condition becomes true.	>>> (a==b) False
!= (Not Equality Operator)	If values of two operands are not equal, then condition becomes true.	>>> (a!=b) True
> (Greater than Operator)	If the value of left operand is greater than the value of right operand, then condition becomes true.	>>> (a>b) False
< (Less than Operator)	If the value of left operand is less than the value of right operand, then condition becomes true.	>>> (a<b) True
>= (Greater than Equal to operator)	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	>>> (a>=b) False
<= (Less than Equal to Operator)	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	>>> (a<=b) True

1.5.4 Logical Operators

- The logical operators in Python programming are used to combine one or more relational expressions that result in complex relational operations. The result of the logical operator is evaluated in the terms of True or False according to the result of the logical expression.
- Logical operators perform logical AND, logical OR and logical NOT operations. These operations are used to check two or more conditions. The resultant of this operator is always a Boolean value (True or False).
- For following table, assume the variable **a** hold True and variable **b** hold False value.

Table 1.5: Logical Operators

Operator	Description	Example
AND (Logical AND Operator)	If both the operands are true then condition becomes true.	(a and b) is False.
OR (Logical OR Operator)	If any of the two operands are non-zero then condition becomes true.	(a or b) is True.
NOT (Logical NOT Operator)	Used to reverse the logical state of its operand.	Not (a and b) is True.