Text Book Of

PYTHON PROGRAMMING

For

Semester – II

FIRST YEAR DIPLOMA IN ENGINEERING AND TECHNOLOGY
COMPUTER SC. & ENGG/C.S.I.T/COMPUTER ENGG. APPLICATION/
COMPUTER ENGINERRING/CS&IS/AI&ML/CE&IOT

As Per New Syllabus of Jharkhand University of Technology, Ranchi w.e.f. 2023-2024

VIJAY T. PATIL
M.E. Computer Engineering
Head, Department of Computer Engineering
Vidyalankar Polytechnic, Wadala,
Mumbai - 400037

Er. KAUSHAL HANSRAJ SINGH
M.Tech. Mechanical Engineering
Principal, Subhash Institute of Technology Giridih
Near Giridih College, Koridih,
Jharkhand - 815312

Price ₹ 180.00

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

N0571

PYTHON PROGRAMMING **ISBN 978-93-6109-939-7**

First Edition : June 2024

© : 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 ...

Welcome to the world of ***"Python Programming"*** Whether you are a complete beginner or an experienced developer looking to explore the versatile capabilities of Python, this book is designed to be your comprehensive guide on your journey through the Python programming landscape.

Python has rapidly emerged as one of the most popular and influential programming languages, known for its simplicity, readability, and wide range of applications. It is used in web development, data analysis, artificial intelligence, scientific research, automation, and much more. This book aims to provide you with the knowledge and skills you need to harness the power of Python effectively.

If you are an absolute beginner, you will find a gentle introduction to programming concepts and Python syntax. If you are an experienced programmer in another language, you will discover how Python's unique features can simplify your coding tasks and expand your capabilities.

This book is strictly written as per the new revised syllabus of Jharkhand Technical University, Ranchi w.e.f. 2024.

I have structured this book to cover a wide range of topics, starting from the basics and progressing to more advanced concepts. Here is a brief overview of what you can expect:

Getting Started: Introduction of Python, help you set-up your development environment, and guide you through writing your first Python program.

Basic Python's Constructs : You will learn about Python's Constructs such as Python Language and Scripting Language, Technical Strength of Python Python Building blocks, data types, variables, operators.

Control Statements in Python: This will explore Types of Python's Control Statements like Decision-making statement such as if, if…else, else…if ladder, nested if and switch statement, Looping Statement : while loop for loop, nested loop and Manipulating Loop.

Data Structure in Python : This will explore Python's built-in data structures like lists, dictionaries, sets, and tuples, as well as how to work with strings.

List, Dictionary, Arrays and Strings: This will explore you about the features; declaration; initialization, basic operations and indexing of Array, List and Dictionary in Python. This will also explore how to create assign indexing and built-in functions by using strings in Python.

Functions with Basic Object-Oriented Concepts: You will dive into the world of object-oriented programming (OOP) and understand python functions, Creating classes and objects, inheritance, Constructors, destructors and encapsulation.

Modules and Packages in Python: Discover how to leverage Python's extensive standard library and install external packages to extend Python's functionality.

A special thanks to Shri. Dineshbhai Furia, Mr. Jignesh Furia M/s Nirali Prakashan for showing full faith in us to write this text book. I am thankful to Ms. Anagha Medhekar Co-ordinator for their constructive suggestion and excellent Co-operation. I am also thankful to Mr. Ilyas Shaikh for his excellent type setting, Mrs. Yojana G. Deshpande (Figure Drawing) and Mr. Jayant Dedhia and Mr. Vijay Mali (Marketing Executive) for promotion of the book.

We Welcome constructive suggestions from our colleagues and students.

With best wishes,

Author

Syllabus ...

1. **Fundamental Concepts**

   Brief history; features; applications of python; python distributions; versions; python IDES; Python interpreter.

   Execution of python programs, debugging python code; Indentation, Comments; best practices for python programming.

   Character set; tokens; keywords, variables, naming rules for variables, Assignment.

2. **Basics I/O Operations**

   Input - Input (), raw_input(); output - print (), formatting output.

   Datatypes, Scalar type: Numeric (int, long, float, complex). Boolean, bytes, None; Type casting

   Operators, Arithmetic, Comparison/Relational, Logical/Boolean, Bitwise; string operators; Expressions and operator precedence.

3. **Control Flow: Conditional Blocks**

   If statement: general format; Multiway branching; Sufficient examples.

4. **Control Flow: Loops**

   While loop: general format; examples For loop: general format, examples. Range(); nesting loops and conditional statements; Controlling loop execution: Break, continue, pass statements.

5. **Data Collections**

   Concept of mutability, Set-features, declaration, initialization, operations, comprehension.

   Tuple-features; declaration, initialization, basic operations; indexing; slicing; built in functions; Nested tuples.

6. **List**

   Features; declaration, initialization, basic operations; indexing.

   List iterations; Slicing; built in functions; Nested Lists; Comprehensions; Applications.

7. **Dictionary**

   Features; declaration, initialization, basic operations; indexing; adding and removing keys, iterating through dictionaries, built-in functions; Comprehensions; Applications.

8. **Arrays and Strings**

   Arrays: features; create, initialize, indexing, traversal, manipulation.

   Strings: create, assign, indexing, built in functions.

9. **Functions**

   Need of function; types; define function, calling function, function arguments; return and yield; None keyword; Scope of variables.

   Recursion; anonymous functions; sufficient examples.

10. **Modules and Packages**

    Why modules? Module creation; Importing modules; Module Namespace.

    Packages: basics; path setting; Package_init_.py Files; Commonly used modules: Math, random; Emoji.

11. **NumPy**

    Brief about NumPy module; NumPy arithmetic functions; NumPy array manipulation functions; NumPy statistical functions.

    Pandas: Introduction, series, data frame; Create dataframes; formatting data; fundamental data frame operations.

12. **Files**

    Concept; features; file operations; Opening Files; Closing Files; Writing to Files.

    Reading to Files; File methods; Working with files using data frame.

13. **Error and Exception Handling**

    Python errors; exceptions: built in, user defined. How to catch exceptions? Raising exceptions.



Contents ...

1. FUNDAMENTAL CONCEPTS **1.1 – 1.12**

1.1 Brief History 1.1

1.2 Features of Python 1.1

1.3 Applications of Python Programming 1.4

1.4 Installation Python in Windows 1.4

1.5 Python IDE's and Python Interpreter Starting Python in Different Modes 1.6

1.6 Executing Python Programs 1.7

1.7 Debugging Python Code 1.8

1.8 Indentation 1.9

1.9 Comments: Best Practices for Python Programming 1.9

1.10 Character Set 1.10

1.11 Tokens 1.10

1.11.1 Identifiers 1.10

1.11.2 Keywords 1.10

1.11.3 Literals 1.11

1.12 Variables 1.11

- Exercise 1.12
- Multiple Choice Questions 1.12

2. BASICS I/O OPERATIONS **2.1 – 2.14**

2.1 Input and Output Statements in Python 2.1

2.2 Example: Python String Formatting Using Format() Function 2.2

2.3 Data Types 2.2

2.3.1 Numeric Data Type 2.2

2.3.2 Boolean Data Type 2.3

2.3.3 String Data Type 2.3

2.3.4 List Data Type 2.4

2.3.5 Tuple Data Type 2.5

2.3.6 Dictionary Data Type 2.5

2.4 Type Casting 2.6

2.4.1 Implicit Conversion 2.6

2.4.2 Explicit Conversion 2.6

2.5 Operators 2.7

2.5.1 Arithmetic Operators 2.7

2.5.2 Relational or Comparison Operators 2.8

2.5.3 Assignment Operators 2.9

2.5.4 Logical Operator 2.10

2.5.5 Bitwise Operators 2.11

2.5.6 Membership Operators 2.11

2.5.7 Identity Operators 2.12

2.6 Python Expression and Operator Precedence 2.13
- Exercise 2.13
- Multiple Choice Questions 2.14

3. CONTROL FLOW: CONDITIONAL BLOCKS 3.1 – 3.10

3.1 Conditional Statements OR Decision-Making Statements 3.1
3.1.1 The if Statement 3.1
3.1.2 The if-else Statement 3.2
3.1.3 Nested-if Statement 3.3
3.1.4 multi-way if-elif-else Ladder Statements 3.4
3.2 Multiway Branching 3.5
3.2.1 Implementing Switch in Python Using elif Ladder 3.5
3.2.2 Python Switch Case Using Dictionary 3.5
3.2.3 Switch Statements with the Match and Case Keywords in Python 3.6
3.2.4 Using Python Functions and Lambdas 3.6
3.2.5 Switch Case Using a Class 3.7
3.3 Sufficient Examples 3.7
- Exercise 3.9
- Multiple Choice Questions 3.10

4. CONTROL FLOW: LOOPS 4.1 – 4.7

4.1 Looping Statements 4.1
4.1.1 While Loop 4.1
4.1.2 The for Loop 4.2
4.1.3 Nested for and While Loop 4.2
4.2 Range() 4.3
4.3 Controlling Loop Execution 4.3
4.3.1 Break Statement 4.3
4.3.2 Continue Statement 4.3
4.3.3 Pass Statement 4.4
- Exercise 4.7
- Multiple Choice Questions 4.7

5. DATA COLLECTIONS 5.1 – 5.14

5.1 Concept of Mutability 5.1
5.2 Set: Accessing Values in Sets, Deleting Values in Sets, Updating Set, Basic Set Operations 5.1
5.2.1 Accessing Values in Sets 5.2
5.2.2 Deleting Values in Set 5.2
5.2.3 Updating Set 5.2
5.2.4 Basic Set Operations 5.3
5.2.5 Built-in Set Functions and Methods 5.4
5.3 Tuple: Accessing Values in Tuples, Deleting Values in Tuples, Updating Tuples, Basic Tuple Operations 5.5
5.3.1 Creating Tuples 5.5

5.3.2 Accessing Values in Tuple 5.6
5.3.3 Tuples are Immutable 5.6
5.3.4 Tuple Assignment 5.6
5.3.5 Tuples as Return Values 5.7
5.3.6 Variable Length Argument Tuples 5.7
5.3.7 Basic Tuples Operations 5.8
5.3.8 Built-in Tuple Functions 5.10
5.3.9 Nested Tuples 5.11
- Exercise 5.13
- Multiple Choice Questions 5.14

6. LIST 6.1 – 6.12

6.1 Python Lists 6.1
6.1.1 Defining List 6.1
6.1.2 Creating List 6.1
6.1.3 Accessing Values in List 6.2
6.1.4 Lists are Mutable (Updating Lists - Change or Add Elements to a List) 6.2
6.1.5 Traversing a List 6.3
6.1.6 Deleting Elements from List 6.3
6.1.7 Built-in List Operators 6.4
6.1.8 Built-in List Functions and Methods 6.5
6.1.9 Basic List Operations: Indexing, Slicing 6.7
6.1.10 Range() Function 6.9
6.1.11 Aliasing and Cloning Lists 6.9
- Exercise 6.12
- Multiple Choice Questions 6.12

7. DICTIONARY 7.1 – 7.6

7.1 Python Lists 7.1
7.1.1 Creating Dictionary 7.1
7.1.2 Accessing Values in a Dictionary 7.2
7.1.3 Updating Dictionary 7.2
7.1.4 Deleting Elements from Dictionary 7.2
7.1.5 Properties of Dictionary Keys 7.3
7.1.6 Operations in Dictionary 7.3
7.1.7 Dictionary Methods 7.4
- Exercise 7.7
- Multiple Choice Questions 7.7

8. ARRAYS AND STRINGS 8.1 – 8.14

8.1 Arrays 8.1
8.1.1 Adding Element into Array 8.1
8.1.2 Accessing Element from Array 8.2
8.1.3 Removing Elements from Array 8.2

8.1.4 Slicing Array 8.2
8.1.5 Updating Elements in Array 8.3
8.1.6 Searching Element from Array 8.3
8.1.7 Counting Elements in Array 8.3
8.1.8 Reversing Array Element 8.3
8.1.9 Extending Array 8.4
8.2 Strings 8.4
8.2.1 Compound Data Types 8.4
8.2.2 Traversal with a for Loop 8.5
8.2.3 String Slicing 8.5
8.2.4 Strings are Immutable 8.6
8.2.5 String Searching 8.6
8.2.6 Looping and Counting 8.7
8.2.7 String Methods 8.7
8.2.8 The in Operator 8.12
8.2.9 String Comparison 8.12
8.2.10 String Operations 8.13
• Exercise 8.13
• Multiple Choice Questions 8.13
9. FUNCTIONS 9.1 – 9.13
9.1 Introduction to Functions 9.1
9.2 Function Uses 9.1
9.3 Function Definition 9.1
9.4 Defining Function 9.2
9.5 Function Arguments 9.2
9.5.1 Required Arguments 9.2
9.5.2 Keywords Arguments 9.2
9.5.3 Default Arguments 9.3
9.5.4 Variable Length Arguments 9.3
9.6 return Statement 9.3
9.7 Fruitful Function 9.3
9.8 void Functions 9.3
9.9 Anonymous Function (The Lamda Function) 9.4
9.10 Function Parameters 9.4
9.11 Scope of Variable 9.4
9.12 Recursion 9.5
9.13 Maths Functions 9.6
• Exercise 9.12
• Multiple Choice Questions 9.13

10. MODULES AND PACKAGES **10.1 – 10.9**

10.1 Modules 10.1

10.2 Writing Module 10.1

10.3 Importing Modules 10.1

10.4 Importing Objects From Module 10.2

10.5 Aliasing Modules 10.2

10.6 Python Built in Modules 10.3

10.6.1 math Modules 10.3

10.6.2 Random Module 10.4

10.6.3 Statistics Module 10.6

10.7 Python Packages 10.6

10.7.1 Writing Python Packages 10.6

10.7.2 Python namespace Packages 10.7

10.7.3 User Defined Packages 10.8

- Exercise 10.8
- Multiple Choice Questions 10.9

11. NumPy **11.1 – 11.15**

11.1 NumPy 11.1

11.2 Introduction to NumPy 11.2

11.3 Creating Arrays, Using Arrays and Scalars 11.2

11.4 NumPy Array Creation Functions 11.3

11.5 Array Object 11.3

11.6 Basic Array Operations 11.4

11.7 Slicing and Indexing 11.5

11.8 Indexing Arrays 11.5

11.9 Array Transposition 11.6

11.10 Statistical Functions 11.7

11.11 Array Input and Output 11.7

11.12 Pandas 11.9

11.12.1 Installing Pandas 11.10

11.12.2 Pandas Series 11.10

11.12.3 Pandas DataFrame 11.12

11.12.4 Pandas Read CSV 11.14

- Exercise 11.15
- Multiple Choice Questions 11.15

12. FILES **12.1 – 12.8**

12.1 Introduction to File 12.1

12.2 Opening File in Different Modes 12.1

12.3 Different Modes of Opening File 12.2

12.4 Closing Files 12.3
12.5 The File Object Attributes 12.3
12.6 Writing Data to File 12.3
12.7 Reading Data From File 12.4
12.8 File Position 12.5
12.9 Handling Files through OS Module 12.6
12.10 Renaming File 12.6
12.11 Deleting File 12.7
12.12 File Related Methods 12.7
- Exercise 12.8
- Multiple Choice Questions 12.8

13. ERROR AND EXCEPTION HANDLING 13.1 – 13.6

13.1 Exception Handling 13.1
13.2 Exception Handling in Python Programming 13.3
13.3 User Defined Exception 13.5
- Exercise 13.6
- Multiple Choice Questions 13.6



Chapter 1

FUNDAMENTAL CONCEPTS

Syllabus

Brief history; features; applications of python; python distributions; versions; python IDES; Python interpreter

Execution of python programs, debugging python code; Indentation, Comments; best practices for python programming.

Character set; tokens; keywords, variables, naming rules for variables, Assignment

Learning Objectives

At the end of this chapter, the students will be able to:

- Set-up Python.
- Execute Python: Explore different ways to run python program.
- Debug python code.

1.1 BRIEF HISTORY

- Python is an object-oriented, high-level, interpreted programming language. Python is currently the most popular programming language.
- Python's popularity stems from its ability to produce code that is more reliable, has clean syntax, supports object-oriented programming, is scalable and portable, has a large standard library, is easy to learn and read, supports GUI mode, is interactive, versatile, and interpreted, and can be used to interface with all major commercial databases.
- In the late 1980's, Python began to take shape. At the Netherlands' National Research Institute for Mathematics and Computer Science in 1990, Guido Van Rossum created Python.
- He gave the language the name Python in honor of the BBC comedy series Monty Python's Flying Circus.
- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and Unix shell and other scripting languages.
- Python source code is now available under the GNU General Public License (GPL).
- Python provides a rich library support and ease of use make it a popular in academic and industry.
- Almost on every operating system, including Windows, Mac, Linux/Unix, etc., can run Python. You can download Python from http://www.python.org/downloads.

1.2 FEATURES OF PYTHON

1. Easy to Learn/code and Use
2. Interpreted Language
3. Interactive Mode
4. Free and Open Source
5. Platform Independence/Cross-platform Language/Portable
6. Object-Oriented Language
7. Extensible
8. Large Standard Library
9. GUI Programming Support
10. Embedded Language

1. Easy to Learn/Code and Use:

- Python is easy to learn and use. It is developer-friendly and high-level programming language.
- Python 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 language such as C, C++ and Java.

2. Interpreted Language:

- Python is an interpreted language i.e., 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, is 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 the 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 of three Greater Than symbols (>>>) appear, (See Fig. 1.1).

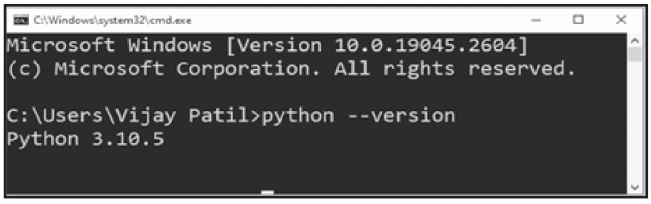


Fig. 1.1

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

Sr. No.	Editor	Description	Icon/Logo
1	IDLE	IDLE is a popular Integrated Development Environment written in Python and it has been integrated with the default language. Mainly used by the beginner level developers who want to practice on Python development	
2	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	
3	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 feature. It is very extensible and has a good plugin system and API	
4	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 inter-preter	

5	Jupyter	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.	
6	Anaconda	Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics etc	

- In this text 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 (See Fig. 1.2).
- 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.

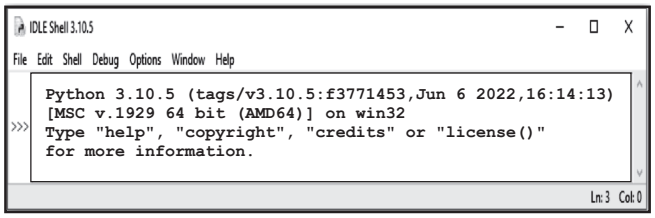


Fig. 1.2

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.
- Python is freely available for everyone. It is freely available on its official website **www.python.org**. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."
- The Python software can be freely distributed and any one can use and read its source code make changes/modifications to it, 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.

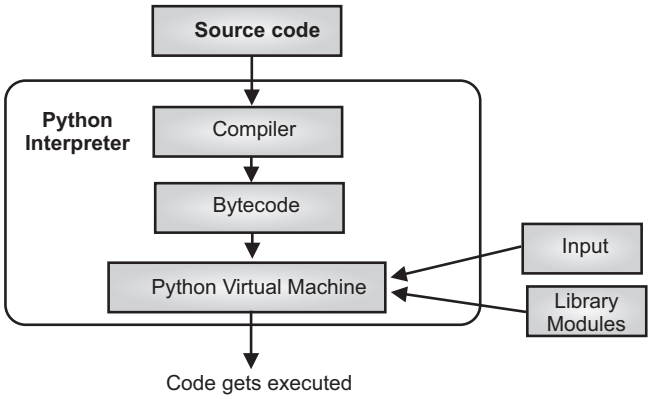


Fig. 1.3

- Python source code goes through Compiler which compiles the source code into a format known as 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 is always present as part of the Python system, and is the component that truly runs the scripts. Technically, it is 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 code that can be reused.