Python Programming for Students

Explore Python in multiple dimensions with project-oriented approach

Nidhi Grover Raheja

www.bpbonline.com

First published: 2024

Published by BPB Online
WeWork
119 Marylebone Road
London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55516-084

www.bpbonline.com

Dedicated to

My Family:
Rakesh Kumar Grover *with* ***Saroj Grover*** *&* ***Isha***
Ashok Kumar Raheja *and* ***Renu Raheja***

My husband:
Sameer Raheja
&
Our daughter ***Cherika***

About the Author

Nidhi Grover Raheja is actively working as Technical Trainer in the domains of Python Programming, Data Analytics and Visualization Tools. She possesses over a decade experience and is associated with numerous reputed educational and training institutions in the role of Technical Trainer and Guest Lecturer. She qualified UGC-NET (Lectureship) and GATE in Computer Science. After completing MCA from GGSIPU, Delhi she accomplished M.Tech (CSE) from DCRUST, Sonipat with Distinction. Her interest areas include Python programming with Machine Learning, Deep Learning, Natural Language Processing, Statistical Analysis and Visualization Tools including Tableau and Microsoft Power BI. She endeavors not only training students with experiential learning approach but also continuously tries to shape up their careers with best of skills and knowledge as per standards.

About the Reviewer

Shivkumar Ramanna Chandey is a seasoned technical reviewer with an insatiable passion for cutting-edge technology and a knack for dissecting complex concepts. He currently works as an Assistant Professor in the Department of Computer Science and Information Technology at Nirmala Memorial Foundation College of Commerce and Science, Kandivali East – 400101, affiliated to the University of Mumbai. With more than 05 years of experience in the tech teaching industry, he has honed his skills to ensure that every product, project, or manuscript he reviews receives the meticulous attention it deserves.

He has written various research papers that have been published in reputed International Journals, National Journals, and Conference Proceedings. His area of research includes Cloud Computing, Digital Communication, Blockchain, Cyber Security, Open-Source Software, etc.

By keeping the motto of "Learn Something About Everything and Everything About Something," he has explored various technical fields. His expertise spans a wide spectrum of technologies, including Cloud Computing, Data Science, Blockchain, Web Services, Cyber Security, Web Development, Various Programming Languages, DBMS, Linux, and FOSS, still exploring and learning new technologies to keep himself updated in this digital world. He has flexibility in technical areas and utilizes these skills to solve problems by making use of intellectual thinking.

Acknowledgement

I want to express my deepest gratitude to my family and friends for their unwavering support and encouragement throughout this book's writing, especially my parents, my husband & my daughter for their unconditional love and support.

I am also grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. It was a long journey of revising this book, with valuable participation and collaboration of reviewers, technical experts, and editors.

I would also like to acknowledge the valuable contributions of my colleagues and co-worker during many years working in the education industry, who have taught me so much and provided valuable feedback on my work.

Finally, I would like to thank all the readers who have taken an interest in my book and for their support in making it a reality. Your encouragement has been invaluable.

Preface

Python is a powerful, versatile programming language. Python is challenging other programming languages like Java, C#, etc. with its simple syntax and wide range of applications.

Python is a very promising programming language in today's rapidly evolving technological landscape thanks to its applicability in a wide range of domains, including task-specific python programs, standalone GUI applications, creating sustainable websites, creating interactive games, data analytics and machine learning, artificial intelligence, etc.

This book gives readers the opportunity to learn all facets of Python programming through the use of clear and engaging examples, practical codes, examples of completed projects, and exercises based on unsolved assignments.

Each project presented in the book offers a taste of a real-world approach to problem solving while providing the advantages of experiential learning, which allows readers to learn by doing. Readers will enjoy learning Python thanks to the abundance of examples, programming illustrations, and relevant project assignments.

Chapter 1: Getting Started with Programming in Python – This segment will aid the readers to learn Python programming in a quick and easy way through a series of simple interesting examples. In this chapter the set-up process of the Python development environment is discussed with illustrative screenshots. Furthermore, readers will also learn how to write variables, literals, keywords, and comments in Python. Learners will also get a deep insight upon various data types, Input-Output process, types of operators, type-conversions, and Namespace in Python.

Chapter 2: Flow Control Concepts – This chapter introduces the audience to the core concepts flow control and its types in Python. Readers will walk through various working examples of conditional decision-making flow control using if…else statement and iterative flow control using loops in Python. This chapter will lay a firm base for developing problem solving approaches while writing more complicated applications in Python.

Chapter 3: Data Structures and Algorithms – This chapter gives an overview of data structures and related algorithms. Here readers will get an overview of the fundamental data structures supported in Python such as List, Tuple, Set, Dictionary and Comprehensions. The later sections of this chapter describe various sorting and searching algorithms implemented in Python. By the end of this unit readers will be able to apply different algorithmic approaches to solve real time problems at hand.

Chapter 4: Functions in Python – This chapter is based on creating, calling, and managing functions in Python. The readers will learn about predefined functions in Python and create customizable user-defined functions for a specific functionality. Here, the learners will also get an introduction to recursion approach of writing functions. The later section of this chapter deals with creating and managing modules and packages in Python.

Chapter 5: Object-oriented Programming Concepts – This section focuses on the major concepts of Object-Oriented Programming approach including class & Object, Data Hiding, Data Abstraction, Inheritance and its types, Polymorphism and basics of overloading using Python. This chapter will lead the users towards the path of real time project development.

Chapter 6: Turtle Programming in Python – This section deals with creating graphics using Turtle library in Python. After reading this chapter the learners will be able to draw different shapes, fill colours and create attractive designs using Python and Turtle library. Furthermore, learners will get a glimpse of creating animated Turtle graphics in Python.

Chapter 7: Database Handling Using SQLite – This section deals with the creation and management of data using SQLite database with Python. Here the focus will be to integrate SQLite3 module with Python for developing real time database applications. By the end of this chapter readers will be able to develop CRUD applications in Python using SQLite database.

Chapter 8: GUI Application Development Using Tkinter – This unit deals with developing Graphical User Interface (GUI) applications using Tkinter library. Python collectively with Tkinter provides a quick and easy way to create GUI applications. Throughout this unit readers will walk through numerous examples of developing standard GUI based desktop applications.

Chapter 9: Game Development with PyGame – This section takes us into the fascinating world of game development using the PyGame library. This chapter will help readers to learn the PyGame library from basic to advance with the help of simple and well-explained examples. After reading this unit readers will be able to develop simple games in Python.

Chapter 10: Mobile App Development with Kivy – This section deals with creating simple mobile applications in Python using the Kivy library in Python. Here readers will learn the basics of mobile application by creating simple applications and creating .apk files for the same. These .apk files created will help to users to deploy and use mobile applications in android phones.

Chapter 11: Image and Video Processing with Python – This chapter showcases the techniques of manipulating images and video frames. After reading this unit, readers will be able to modify images and videos with ease. Image and video processing is necessary in several multimedia applications. Therefore, this chapter will enable users to manipulate images and videos for such applications.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/9f68d3

The code bundle for the book is also hosted on GitHub at **https://github.com/bpbpublications/Python-Programming-for-Students**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

1. Getting Started with Programming in Python....1
Introduction....1
Structure....1
Objectives....2
Features of Python....2
Installing Python....3
Keywords....7
Identifier....7
Comments....8
Variable and data types....8
Datatypes in Python....10
Numbers....10
Dictionary....12
Boolean....13
Set....14
Sequence types....15
Type conversion in Python....19
Implicit type conversion....20
Explicit type conversion....20
Input/output using Python....22
Output formatting using format()....24
Operators and expressions....25
Arithmetic operators....25
Assignment operators....26
Comparison operators....27
Logical operators....27
Bitwise operators....28
Identity operators....29
Membership operators....30

Operator precedence *31*
Namespaces in Python 32
Conclusion 33
Points to remember 33
Exercise 33
Sample project with solution *33*
Practice project *36*

2. Flow Control Concepts 37
Introduction 37
Structure 37
Objectives 38
Decision-making in Python 38
Workflow of if…elif…else 38
The if statement *39*
The if…else statement *40*
The if…elif…else statement *41*
Nested if *42*
Loop control in Python 44
Python for Loop *45*
Python while Loop *46*
Infinite while Loop *47*
Nested loop *48*
Flow control statements in Python 49
Conclusion 51
Points to remember 52
Exercise 52
Sample project with solution *52*
Practice project *56*

3. Data Structures and Algorithms 57
Introduction 57
Structure 57

Objectives 58
Introduction to PyCharm IDE 58
Installation steps of PyCharm IDE *58*
Built-in data structures 59
String *60*
List *65*
Tuple *67*
Set *67*
Python set operations *68*
Dictionary *70*
User-defined data structures 71
Linked list *71*
Stack *72*
Queue *74*
Sorting algorithms 75
Time complexity and space complexity *75*
Bubble sort *75*
Selection sort *76*
Insertion sort *77*
Searching algorithms 79
Linear search *79*
Binary search *80*
Conclusion 81
Points to remember 81
Exercise 82
Sample project with solution *82*
Practice project *88*

4. Functions in Python 89
Introduction 89
Structure 89
Objectives 90
Introduction to functions 90

Benefits of using functions *90*
Functions versus methods *90*
Types of Python function *91*
Function declaration and calling *91*
Function arguments 92
Types of function arguments *93*
Default arguments *93*
Keyword arguments *93*
Required arguments *94*
Variable-length arguments *95*
Recursion in Python 96
Anonymous functions 97
Example 1: use of lambda function to find the maximum of two numbers *97*
Example 2: To filter out only even numbers from a list of numbers *97*
Example 3: To find the cube of all elements in a list *98*
Scope and lifetime of variables 98
Modules and packages 99
Conclusion 101
Points to remember 101
Exercise 101
Sample project with a solution *101*
Practice project *104*

5. Object-oriented Programming Concepts 105
Introduction 105
Structure 105
Objectives 106
Introduction to programming paradigms 106
Procedural programming *106*
Object-oriented programming *107*
Object-oriented programming concepts *107*
Class and objects 107
Class attributes and methods *108*

Built-in attributes of class *114*
Constructors in Python 115
Parameterized constructor *115*
Non-parameterized constructor *116*
Default constructor *117*
Encapsulation and data hiding 118
Inheritance in Python 119
Types of Inheritance *120*
Single Inheritance *120*
Multilevel Inheritance *121*
Hierarchical Inheritance *122*
Multiple Inheritance *123*
Hybrid Inheritance *124*
Method Resolution Order *125*
super() in Python *127*
Super function in single inheritance *127*
Super function in multiple inheritance *128*
Polymorphism in Python 129
Compile-time Polymorphism *129*
Method and constructor overloading *129*
Operator overloading *130*
Run-time polymorphism *132*
Conclusion 133
Points to remember 133
Exercise 134
Sample project with solution *134*
Practice project *136*

6. Turtle Programming in Python 137
Introduction 137
Structure 137
Objectives 138
Turtle programming in Python 138

Plotting with Turtle *138*
Creating shapes with Turtle 141
Drawing connecting lines *142*
Draw square, rectangle, and triangle *143*
Draw star pentagon, hexagon, and octagon *145*
Draw circle and oval *147*
Draw spiral *149*
Fill colors in shapes 150
Event programming with Turtle 151
Mouse events *152*
Key events *157*
Conclusion 161
Points to remember 161
Exercise 161
Sample project with solutions *161*
Practice project *167*

7. Database Handling Using SQLite 169
Introduction 169
Structure 169
Objectives 170
Introduction to data and database 170
Relational versus non-relational database *171*
Relational databases *171*
Non-relational databases *172*
SQLite for database handling 172
Downloading SQLite *172*
Installing SQLite in command-line *173*
GUI tools for SQLite *174*
SQLite working in Python *174*
Connecting SQLite database in Python *175*
Datatypes in SQLite 176
Exception handling tasks 177

Database management with SQLite 178
Create new database *178*
Commands in SQLite 181
Data Definition Language (DDL) commands—CREATE, ALTER, and DROP *181*
SQLite table constraints *183*
Data Manipulation Language (DML) commands *187*
Data Query Language (DQL) command—SELECT *188*
Clauses in SQLite commands *188*
Aggregate functions *190*
Joins in SQLite 191
Parameterized Query and sub-queries in SQLite 192
BLOB and DATE TIME in SQLite 194
Conclusion 196
Points to remember 196
Exercise 196
Sample project with solution *196*
Practice project *203*

8. GUI Application Development Using Tkinter 205
Introduction 205
Structure 205
Objectives 206
GUI programming in Python 206
Getting started with Tkinter 206
Introducing Tkinter widgets 209
Button *210*
Label *211*
Entry widget *212*
Text widget *215*
Radiobutton *217*
Checkbutton *218*
Combobox *219*
Listbox *220*

Menu *223*
Spinbox *227*
The Treeview Widget and Treeview Scrollbar *229*
Example 1 ..... *231*
Example 2 ..... *232*
Label frame *234*
Messagebox *235*
Tkinter filedialog *237*
Returning a file path *237*
Saving a file *238*
Select directory *239*
Geometry management in Tkinter 239
Organizing widgets with layout managers *239*
Pack layout *240*
Grid layout *241*
Place layout *243*
Event binding 245
Conclusion 247
Points to remember 247
Exercise 248
Sample project with solution *248*
Practice project *259*

9. Game Development with PyGame 261
Introduction 261
Structure 261
Objectives 262
Introduction to PyGame 262
Installing PyGame *262*
Getting started with PyGame *262*
Color object in pygame *265*
Surface and shapes in pygame *265*
Images in pygame *269*

Events in pygame *271*
Adding text and music in pygame *273*
Sprites and collisions *276*
Conclusion 279
Points to remember 279
Exercise 279
Sample project with solution *280*
Practice project *284*

10. Mobile App Development with Kivy 285
Introduction 285
Structure 285
Objectives 286
Introduction to Kivy 286
Characteristics of Kivy *286*
Installation of Kivy *287*
Kivy app life cycle *288*
Widgets and layouts *290*
UX widgets, events, and binding function *291*
Geometery management using layout managers *297*
Basics of KV language 300
Loading the KV File *301*
Syntax guidelines for KV file *301*
Modules and widgets *301*
Events and properties *304*
Dynamic class *304*
Widget reference *305*
User pages with multiple screens 307
Package Kivy applications with buildozer *310*
Conclusion 314
Points to remember 314
Exercise 314

Sample project with solution *314*
Practice project *320*

11. Image and Video Processing with Python 321
Introduction 321
Structure 321
Objectives 322
Introduction to image processing 322
Manipulating images 323
Image processing libraries in Python *323*
Reading an image *324*
Grayscale conversion and image blurring *325*
Image edge detection 328
Object detection in image 330
Image resize and rotation *332*
Image addition, subtraction, and blending *334*
Video processing tasks in Python 335
Capture Live video from Webcam *338*
Conclusion 339
Points to remember 339
Exercise 340
Sample project with solution *340*
Practice project *342*

Appendix 343
Multiple choice questions 343
Solutions *352*

Index 353-359

Chapter 1
Getting Started with Programming in Python

Introduction

In this chapter, we will discuss the basics of Python programming language. Python is an object-oriented high-level programming language that is easy to write and understand, more interactive, interpreted, and meant for general purposes. Guido van Rossum designed "Python" at **Centrum Wiskunde & Informatica** (**CWI**) in the Netherlands and released its first version in 1991. Guido Van Rossum, being a fan of a famous TV show in The Netherlands, "Monty Python's Flying Circus," named the language after **Monty Python**. Python is a powerful scripting language but can be used as an efficient programming language and also to develop a variety of applications. Python has been an open-sourced language since the beginning, and its source code is also available under the GNU **General Public License** (**GPL**).

Structure

In this chapter, we will discuss the following topics:

- Features of Python
- Installing Python
- Keywords
- Identifier

- Comments
- Variable and data types
- Type conversion in Python
- Input/output using Python
- Operators and expressions

Objectives

By the end of this chapter, the readers will know the important characteristics of Python that make it a popular general-purpose language among users. They will be able to set-up a Python development environment in their systems, making them ready for programming. This chapter focuses on Python basics such as defining variables, literals, keywords, expressions, and comments. By learning about various data types, type conversion concepts, input/output processes, and different operators, readers will be able to adopt problem-solving approaches and write simple beginner-level programs in Python.

Features of Python

Python is a scripting language, being interpreted as a high-level programming language, developed for the purpose of fulfilling general programming requirements. The following features of Python make it very popular among its users:

- **Easy to understand**: When we read or write the Python program, we can feel like reading or writing simple English statements. This makes it a beginner's choice of language.
- **Multipurpose in nature**: Python is a general-purpose language that enables the development of a wide variety of applications such as text processing applications, Web programming, machine learning applications, games, and so on.
- **Python is interpreted**: We are not required to compile Python programs explicitly. Internally, the Python interpreter will take care of the compilation process as well.
- **Open-source language**: We can use Python software without any license, and it is freeware. Its source code is open so that we can customize it based on our requirements.
- **Dynamically typed**: In Python, we are not required to declare the type for variables before assigning values. Rather, the type of value assigned to the variable will determine its datatype automatically. Hence, Python is considered a dynamically typed language.
- **Automatic memory management**: Python removes those objects that are no longer in use. It frees up the memory space occupied by such objects automatically using an internal Garbage Collector.

- **Supports platform independence**: Once we write a Python program, it can run on any platform without rewriting it once again, thus providing the feature of Platform Independence.
- **Platform portability**: Python programs are portable, that is, we can migrate from one platform to another very easily. Python programs will provide the same results on any platform.
- **Support for libraries**: Support for various third-party tools and utilities is available.

Installing Python

For most of Unix systems, Linux, and MAC OS, Python is pre-installed. For Windows Operating Systems, users can easily download the latest Python release from its official download page **https://www.python.org/downloads/**. The code given in this book is implemented on Python 3 release Python 3.11.0. We can click on the desirable operating system; for example, click Windows to view download options for the selected operating system. As per your system configurations, you may choose to download Installer for 32-bit or 64-bit operating systems. This will automatically begin downloading the installer executable file. Once the download is complete, click the executable file to start the installation process as shown in *Figure 1.1*:

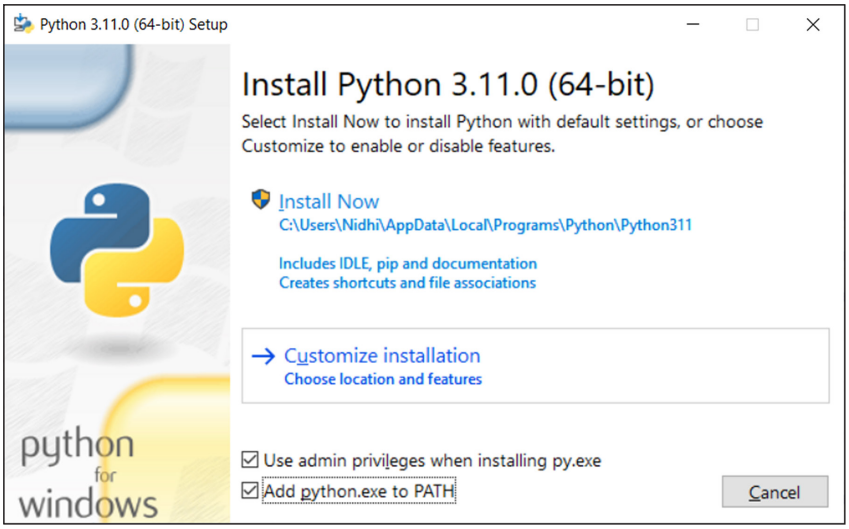

***Figure 1.1**: Installation starting process*

Make sure to check both the checkboxes as highlighted in the preceding figure. This will enable the system to use Admin privileges during the installation process and automatically add the address of the **`python.exe`** file in the **`PATH`** variable of system settings. Next, click on **Install Now** to begin the set-up procedure. Refer to *Figure 1.2*:

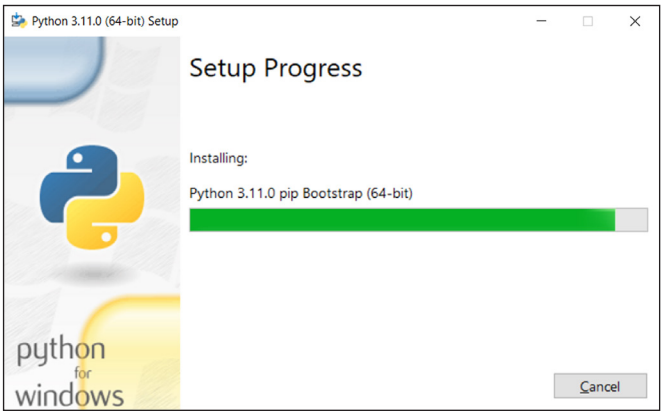

***Figure 1.2**: Installation in progress*

Once the set-up is complete, we can see the successful set-up message, as shown in *Figure 1.3*:

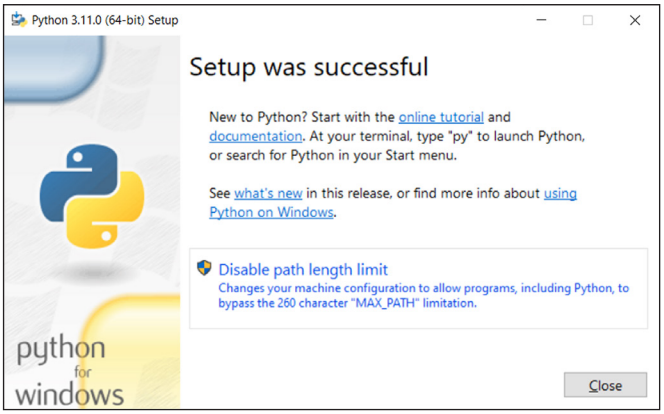

***Figure 1.3**: Set-up success screen*

Click on the **Close** button to exit the set-up window. Now, we are ready to check Python installed on our system. After successful installation, open the command prompt and type the following command on the prompt:

```
python --version
```

The preceding command shows the installed version of Python, as shown in *Figure 1.4*:

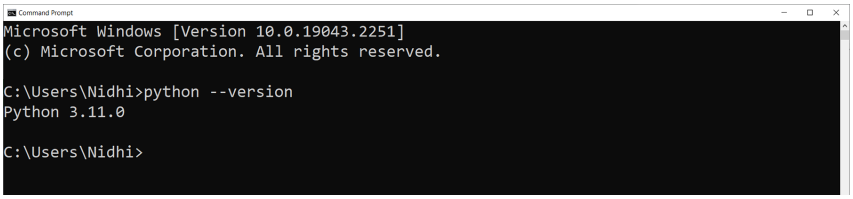

***Figure 1.4**: Check the installed Python version in the command prompt*

Once we install Python 3 successfully on our system, we can use a very simple yet powerful built-in **Integrated Development Environment** (**IDE**) known as **IDLE**. To start IDLE, click on the **Start** menu, then select **Python 3.11 Folder**. Select **IDLE** (Python 3.11 64-bit). Refer to *Figure 1.5*:

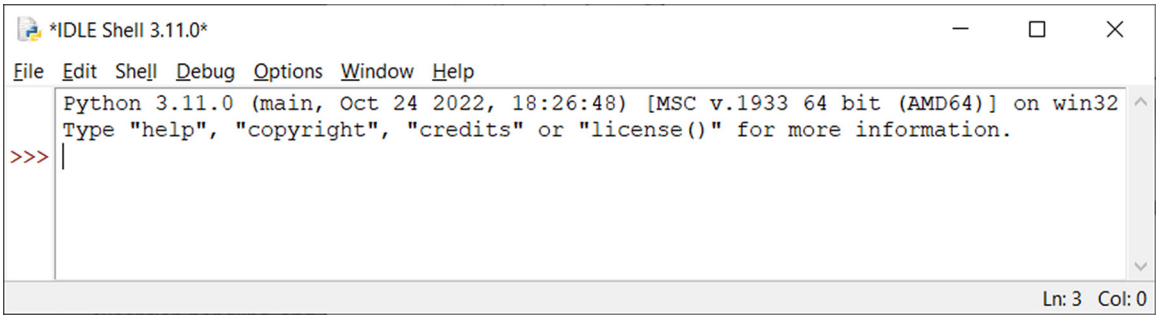

***Figure 1.5**: IDLE shell 3.11.0*

Now, it is time to write our first command in Python IDLE Shell. Here in IDLE Shell, we can write and execute a single command at a time but not the entire Python program in one go. To solve this, we will create a new editing window in the IDLE shell. Go to **File** menu | **New File** or press *Ctrl + N*. This will open a new untitled window where we can write entire Python code and execute it to see the output, as shown in *Figure 1.6*:

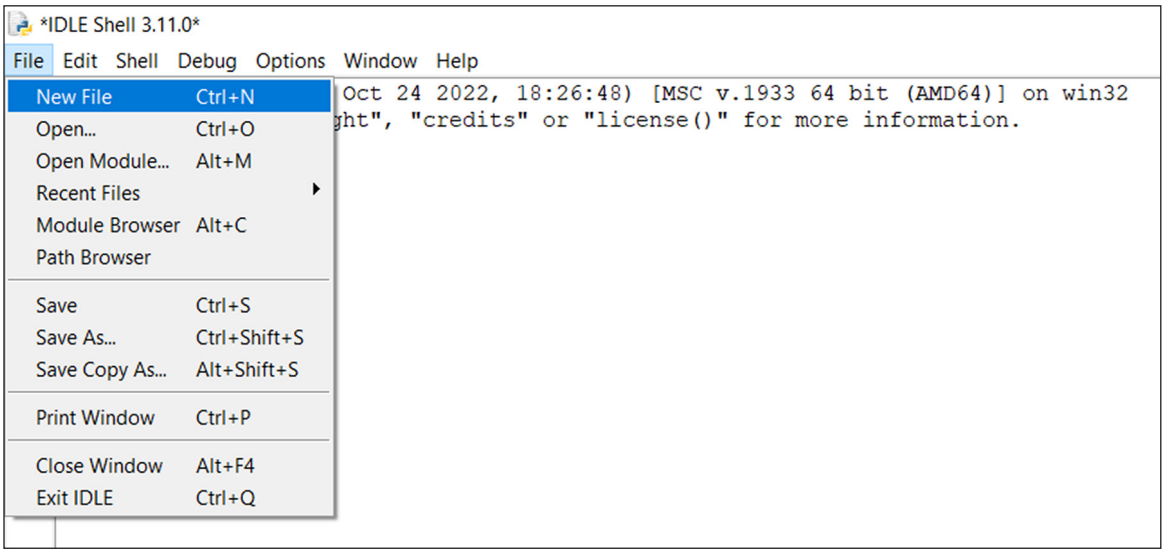

***Figure 1.6**: Opening new module in IDLE*

Let us now write Python code to print the **Hello World** message. For this, we can use the print method in which the **Hello World** message is enclosed in double quotes or single quotes within parentheses brackets. Refer to *Figure 1.7*: