Mastering Python 3 Programming

Ultimate guide to learn Python coding fundamentals and real-world applications

Subburaj Ramasamy

www.bpbonline.com

First Edition 2024

ISBN: 978-93-55517-128

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

My better half ***Chandra*** *and my sons* ***Prabhu and Kumar***

About the Author

Dr. Subburaj Ramasamy, currently an independent research professor, specializes in machine learning for artificial intelligence, programming languages, secure coding and software reliability engineering. He is an Udemy instructor and has published courses on machine learning, Python and C++. He served as a professor and consultant in the Department of Information Technology, SRM Institute of Science and Technology, Kattankulathur, for 17 semesters. Earlier, he was a Scientist "G"/ Senior Director in the Ministry of Electronics and Information Technology, Government of India. He obtained his PhD degree from the University of Madras. He holds a Masters degree from Indian Institute of Technology, Delhi and Bachelors degree from Alagappa Chettiar College of Engineering and Technology. He guided four research scholars till award of the Ph.D. degree, having published about 90 research papers in international journals and conferences with 500 citations. He was teaching and providing project guidance to B.Tech and M.Tech students at the School of Computing in SRMIST. Dr. Subburaj is an esteemed member of various associations, such as the Computer Society of India and IETE. He received the 17th IETE Lal C Verman Award in 2003. He is the author of several books on topics such as Total Quality Management, Software Reliability Engineering, Programming in Python, Programming in C, and Object Oriented Programming with C++. He was an authorized trainer of Carnegie Mellon University/SEI, USA, for software CMM and certified by Stanford University/ Course era on Machine Learning.

About the Reviewer

Peyush Kumar is an experienced IT professional, with a distinguished career that spans over 25 years. Graduating with a B.Tech in Computer Science from Punjabi University, his journey has been marked by impactful contributions from major tech players, including Zensar, Wipro, and HCLTech. This includes employment in global settings while residing for an extended period in the USA. Beyond showcasing his prowess as a skilled programmer, Peyush has proved himself as a proficient manager, steering software development teams towards success.

Peyush's enthusiasm extends beyond coding; he is also a passionate educator. His teaching legacy is witnessed in the diverse array of students, spanning from high school enthusiasts to seasoned graduates, who have benefited from his insightful guidance. Residing in Noida with his family, Peyush balances his love for technology with an unwavering commitment to family values.

Acknowledgement

I would like to express my sincere gratitude to all those who contributed to the successful completion of this book.

First and foremost, I extend my appreciation to my family and friends for their unwavering support and encouragement throughout this journey.

I am immensely grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. I would like to especially appreciate the alert, sharp and knowledgeable technical reviwer for reviewing the manuscript thoroughly. I also thank the editors and administrators of BPB for their support and assistance, which were invaluable in navigating the complexities of the publishing process.

Last but not least, I want to express my gratitude to the readers who have shown interest in my book.

Thank you to everyone who has played a part in making this book a reality.

Preface

Mastering Python 3 Programming is a must for every literate. Irrespective of whether you are a seasoned programmer looking to expand your skill set or an absolute beginner taking baby steps into the world of programming, this book is designed to be your companion and comprehensive guidebook to mastering Python.

Python has emerged as one of the most sought-after programming languages due to its agility. Its simplicity, readability, and versatility make it a preferred language for a wide range of applications, from web development and data analysis to artificial intelligence and machine learning.

In this book, we have described everything you need to know to become proficient in Python 3. We will start with the fundamentals, including basic syntax, data types, and control structures, before moving on to more advanced topics such as built-in containers like lists and dictionaries, object-oriented programming, functional programming, and advanced file and exception handling.

Throughout the book, you will find bountiful hands-on examples and exercises to help reinforce your learning. Whether you prefer to follow along with the code examples or dive in and start coding on your own, you will have abundant opportunities to practice and experiment with Python.

One of the fascinating things about Python is its thriving ecosystem of batteries included - libraries and frameworks. In the later chapters, we will explore some of the most popular Python libraries for tasks such as data structures and data visualization, giving you the tools you need to start building your own projects right away.

Whether your goal is to grab a job as a Python developer, advance in your current career, or simply explore the exciting world of programming, this book will provide you with the knowledge and skills you need to succeed. So, without further ado, let us dive in and start exploring the wonderful world of Python 3 programming!

The contents of the book are given chapter-wise below:

Chapter 1: Introduction to Python 3 - Discusses the motivation to learn Python and how to install the Python development environment. We will be discussing programming in the interactive mode and script mode and give examples of both using the above. We will be using IDLE, throughout the book owing to its merits. We will learn

that the Python frozen binaries bundle together the byte code of our program files, along with the Python Virtual Machine interpreter and any Python support files our program needs, into a single package, a single binary executable program like .exe file on Windows. By the end of the chapter, students would have experimented with the interactive mode as well as create frozen binaries in the script mode.

Chapter 2: Algorithmic Problem Solving - In this chapter, the student will learn to design computer based solutions to problems using algorithms and flow charts. At the end of the chapter, we describe some of the popular algorithms in Python. This chapter aims to prepare freshers for programming with algorithmic thinking. Furthermore, the algorithms discussed here are used in the rest of the book.

Chapter 3: Numeric Computations and Console Input - An overview of the major tokens of the Python language, such as identifiers, keywords, constants or literals, and operators, is given in this chapter. We will also discuss expressions, operator precedence, and type conversion. In this chapter, we also introduce console input using input and eval functions. We will discuss operations on the bits using bitwise operators.

Chapter 4: Unicode, Strings and Console Output - In this chapter, we will discuss character coding, strings, and slicing. String methods and methods to modify strings will also be discussed. Then, the chapter teaches converting ASCII characters to numbers and vice versa using ord () and chr () functions. We will then move on to discuss console output. Several methods of formatted printing are also discussed in the chapter. Then we discuss the bytes () function, which returns an immutable bytes object. It can convert strings into bytes objects.

Chapter 5: Selection and Loops - In this chapter, we will discuss relational operators and logical operators. Then we will describe the selection constructs of if, elif, and else and give programs to illustrate their use. Next, we will give examples of nested selection constructs and the use of a ternary operator. Then we will move on to discuss iterations using *while* and then *for* and give examples. We will demonstrate the use of break and continue keywords in programs.

Chapter 6: Functions and Recursion - In this chapter, after discussing the features and benefits of structured programming, we will give examples of calling a function multiple times and calling more than one function in a program. We will also discuss void functions. Then we discuss fruitful functions, return values, runtime stack, and Boolean functions. Then, we will discuss the local and global scope of function parameters. This is followed by positional arguments and default arguments.

Chapter 7: Lists - In this chapter, we carry out an in-depth analysis of the Python container list. Then we will discuss list concatenation, slicing, and nesting. Then we will highlight the

differences between functions and methods in Python and give examples of methods and functions using lists. List comprehension, aliasing list, and cloning lists are also discussed. This chapter also includes several case studies.

Chapter 8: Tuples, Sets, and Dictionaries - In this chapter, we will discuss 3 more containers – tuples, sets, and dictionaries. We will discuss the methods and functions used with all three containers. We will carry out mathematical operations using sets and set comprehension. The chapter gives some examples of dictionaries and iterations over dictionaries, dictionary comprehension, and nested dictionaries.

Chapter 9: Introduction to Object-Oriented Programming - In this chapter, we give an overview of the characteristics of **Object-Oriented Programming** (**OOP**). Class as a blueprint for objects, access control, initializer, and destruction of objects are discussed in this chapter. After carrying out operations on objects, the distinction between class variables and instance variables is brought out. The overloading functions supported in Python are illustrated with examples addressing overloading binary operators - minus operator and the equality operator. At the end of the chapter, we discuss documentation strings.

Chapter 10: Inheritance and Polymorphism - In this chapter, we discuss single inheritance, multi-level inheritance, and multiple inheritance. We will give a comparison between containership and inheritance, and implement various types of inheritances. Then we will discuss the *object* class and diamond problem. It is followed by method overriding, polymorphism, and abstract class. This chapter ends with a discussion on iterators. iterable and generators.

Chapter 11: File Handling - This is a comprehensive and important chapter in the book in the context of Machine Learning and big data analytics. We will discuss reading from and writing to text and binary files. Implicit reading is an important addition. It has an interesting case study on counting the occurrence of each word and each alphabet in a text file. **Java Script Object Notation** (**JSON**) and the pickle solution are interesting and useful in many applications. The CSV Files and Command line arguments are also illustrated in this chapter.

Chapter 12: Exception Handling - We begin the chapter with a listing of exception classes provided by the Python standard library. The chapter describes all the five keywords provided by Python for exception handling. The chapter has two case studies, building an exception class and knowingw the cause of the error in the custom exception class

Chapter 13: Gems of Python - This is one of the power-packed chapters in the book, addressing various special features of Python 3 such as lambda functions, modules, date

and time, functional programming, first class functions, decorator function, namespaces and packages.

Chapter 14: Data Structures and Algorithms using Python - This chapter gives several case studies after explaining stack, queue, and exception handling. The chapter briefly explains algorithm analysis - space complexity, time complexity, and Big Oh notation. We discuss four popular sorting algorithms in this chapter, such as bubble sort, selection sort, insertion sort and merge sort.

Chapter 15:Data Visualization - In this chapter, we will use Python libraries - pandas, Matplotlib and Seaborn to plot various graphs, charts and diagrams using two datasets publicly available on the Internet. The reader will be given URLs to download the data and steps to install libraries.

Chapter 16 : Python Applications and Libraries - In this chapter, we give an overview of applications of Python, including libraries used therein as a ready reckoner to help students to select their future areas of learning. This chapter covers areas such as web development, data science and machine learning, **Natural Language Processing** (**NLP**), GUI applications, game development, scripting and automation, web scraping, database applications, **Internet of Things** (**IoT**) and education.

Code Bundle and Coloured Images

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

https://rebrand.ly/syyxw8e

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/Mastering-Python-3-Programming.
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

Chapter 1
Introduction to Python 3

Introduction

Python is one of the popular and widely used high-level programming languages. It is both a procedure-oriented and object-oriented programming language like C++. It is also a web-scripting language like Javascript. The Python Virtual machine, like Java virtual machine, is the key element of Python, which facilitates the portability of Python programs across platforms. Python's **batteries included** philosophy is a boon to the programmers to write compact and error-free code.

In this chapter, we will be learning the history of Python, its features, applications, and advantages. The Python 3 language system matching the operating system in our computer system can be downloaded from **www.python.org**. After installation, if our computer runs under Windows operating system, we will get the Python command line interpreter and **Integrated Development Environment** (**IDLE**) under the *start* menu in Windows in the system. We use the statement prompts in the above, in the interactive mode, to confirm our understanding of the Python language syntax. The IDLE, in addition, can be used for writing and executing the programs in script mode. We get a glimpse of the use of mathematical and other functions received along with the Python language system. We will also write simple Python programs in this chapter.

Structure

The chapter covers the following topics:

- History of Python
- Major users
- Features of the language
- Applications of Python 3
- Advantages of Python 3
- Interpreters and compilers
- Python Virtual Machine
- Interpretation process
- Versions of Python
- Python implementations
- Installing Python
- Interactive mode
- Integrated Development Environment
- Execution of Python programs
- Dynamically typed vs. statically typed language
- Memory administration in Python
- Python calculator
- Special characters
- Batteries included philosophy
- Mathematical functions in Python
- Other built-in functions
- Frozen binaries executables
- Types of errors in programs

Objectives

After studying this chapter, you will understand how to invoke the Python command line interpreter, carry out a few calculations, and display a few messages. You would have invoked IDLE, created a new file, typed a program, saved it in a directory, and executed

the program. You would be able to use mathematical and other functions and execute programs in IDLE.

History of Python

Python is a high-level procedure-oriented programming language like BASIC, COBOL, FORTRAN, and C. It is also an object-oriented programming language like C++, C sharp, and Java. It can thus be used for procedure-oriented programming as well as object-oriented programming. Python was released for use in the year 1991 by *Guido Van Rossum* (b 1956) in Netherlands. Python was named after the popular British comedy troupe, Monty Python's Flying Circus. It is freeware and not a proprietary product. We can say that it is community owned. It is an open-source programming language, and software professionals all over the world contribute to the development of the language. Python is managed officially by Python Software Foundation, a not-for-profit organization with its headquarters in Delaware, United States.

Major users

Many world-class organizations, such as those listed below, use Python for the development of software products in their organizations:

- Google
- Facebook
- Instagram
- YouTube
- Spotify
- Quora
- Netflix
- Dropbox

The **National Aeronautics and Space Administration** (**NASA**), an independent agency of the U.S. Federal Government responsible for the civilian space program, as well as aeronautics and space research, uses Python to develop their professional applications. It is a meritorious programming language, and hence such big names are using it extensively, and the user base is growing continually, besides also being used by tens of thousands of small/mid to large enterprises.

Python 3 is a popular and high-level computer programming language known for its simplicity, readability, learnability, and versatility. We will briefly peruse the features, applications, and advantages of Python 3 language.

Features of Python 3

Python 3 has many special features, some of them are unique. The special features of Python are listed below:

- **Simple and readable syntax:** Python emphasizes code readability and uses a clean and straightforward syntax, which makes it easy for developers to write and understand code.
- **Interpreted language:** Python is an interpreted language, which means code is executed line by line, making it easy to test and debug.
- **High-level language:** Python abstracts many low-level details, allowing developers to focus on solving problems rather than managing memory and other system-level concerns.
- **Dynamically typed:** Python is dynamically typed, which means variable types are determined at runtime, providing flexibility but requiring careful attention to data types.
- **Multi-paradigm:** Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming.
- **Rich standard library:** Python has a rich standard library that provides modules and packages for a wide range of tasks, reducing the need for reinventing the wheel.
- **Cross-platform:** Python is available on multiple platforms, making it a portable choice for software development.
- **Community and ecosystem:** Python has a large and active community of developers, which means a wealth of third- party libraries, frameworks, and tools are available.

Python 3 is a versatile language with a strong community and ecosystem, making it a valuable choice for various programming tasks and industries.

Applications of Python 3

Python is widely used in a variety of applications, both scientific and commercial. Some applications of Python are listed below:

- **Web development:** Python is used for web development with frameworks like Django and Flask, making it easy to build web applications and APIs.
- **Data analysis and visualization:** Python, along with libraries like NumPy, pandas, and Matplotlib, is widely used for data analysis, scientific computing, and data visualization.

- **Machine learning and AI:** Python has become the de facto language for machine learning and artificial intelligence with libraries such as TensorFlow, PyTorch, and scikit-learn.
- **Scientific computing:** Scientists and researchers use Python for numerical and scientific computing tasks due to its rich ecosystem of scientific libraries.
- **Automation and scripting:** Python is often used for automating repetitive tasks and writing system scripts.
- **Game development:** Python has libraries like Pygame for game development.
- **Desktop applications:** Python can be used to build desktop applications using frameworks like PyQt and Tkinter.
- **Networking and cybersecurity:** Python is used for network programming, penetration testing, and cybersecurity tasks.
- **Education:** Python is a popular choice for teaching programming due to its simplicity and readability.

Advantages of Python 3

The advantages of Python as a programming language are plentiful. Some advantages are highlighted here:

- **Ease of learning:** Python's simple and readable syntax makes it an ideal language for beginners and experienced developers alike.
- **Productivity:** Python's high-level abstractions and rich standard library allow developers to write code quickly and efficiently.
- **Large community:** The large and active Python community means extensive documentation, support, and a wealth of third-party libraries.
- **Cross-platform compatibility:** Python code can run on various platforms with minimal modifications, making it highly portable. Availability of interpreters for a host of operating systems such as Windows, Linux, Ubuntu and Apple's Mac OS.
- **Versatility:** Python is suitable for a wide range of applications, from web development to data science and artificial intelligence.
- **Open source:** Python is open source, which means it is freely available and can be used and modified without cost.
- **Interoperability:** Python can easily integrate with other languages like C/C++ and Java, making it suitable for extending existing software.
- **Community-driven updates:** Python's development is community-driven, with regular updates and improvements.