PYTHON PROGRAMMING

A Comprehensive Guide to Software Development

with

REAL-WORLD APPLICATIONS

AMIN BOULOUMA

Python Programming: A Comprehensive Guide to Software Development

Amin Boulouma

February 17, 2023

• "I'm very impressed with Python Programming, it's huge and very accurate! Thanks for this wonderful book! I'll share it with my friends, who want to start coding!" — *Dimitry Anisimov, Python Backend Developer*

• "Thank you so much. I thought it would be very hard but You make me learn so fast." — *Praveen Chaudhary, Software Developer*

• "Explains basics very clearly" — *Aviral Agrawal, Software Developer*

• "Most Viewed Writer in the topic Python" — *Quora*

• "Most Viewed Writer in the topic Django" — *Quora*

• "Most Viewed Writer in the topic Python Libraries" — *Quora*

• "Most Viewed Writer in the topic Python Web Frameworks" — *Quora*

• "Most Viewed Writer in the topic ElasticSearch" — *Quora*

• "Most Viewed Writer in the topic Python Web Frameworks" — *Quora*

• "Knowledge prize winner - Best answer in the topic Python" — *Quora*

• "Most Viewed Writer in the topic Python" — *Quora*

• "Published writer - Best answer in the topic Python" — *Quora*

• "Top question writer - Best question in the topic Python" — *Quora*

• "Top writer - Write a lot in the topic Python" — *Quora*

Contents

Preface **9**
How to read this book . . . 9

History and Overview of Python **10**
Introduction . . . 10
A brief history of Python . . . 10
An overview of Python's key features and benefits . . . 10
Python's Standard Library and Tools . . . 10
Python in industry and scientific computing . . . 10
Conclusion . . . 10

Chapter 1: Introduction to Python **11**
Setting up a Python development environment . . . 11
The basics of programming concepts . . . 11
Exercises for practice . . . 11
Summary . . . 12

Chapter 2: Variables and Data Types **13**
Understanding Variables in Python . . . 13
The Various Data Types in Python . . . 13
Numbers . . . 13
Strings . . . 13
Lists . . . 13
Dictionaries . . . 14
And more . . . 14
How to Use and Manipulate Variables and Data Types . . . 14
Exercises for Practice . . . 14
Summary . . . 14

Chapter 3: Control Structures **15**
Understanding control structures . . . 15
`if/else` statements . . . 15
`for` loops . . . 15
`while` loops . . . 15
How to use control structures in Python . . . 15
Exercises for Practice . . . 15
Summary . . . 15

Chapter 4: Functions **16**
Understanding functions and their use in Python . . . 16
Defining and calling functions in Python . . . 16
Exercises for Practice . . . 16
Summary . . . 16

Chapter 5: Modules **17**
Understanding modules in Python . . . 17
Importing and using modules in Python . . . 17
Exercises for Practice . . . 17
Summary . . . 17

Chapter 6: Object-Oriented Programming - Part 1 **18**
Understanding the concepts of object-oriented programming . . . 18
Defining classes and objects in Python . . . 18
Exercises for Practice . . . 18
Summary . . . 19

Chapter 7: Object-Oriented Programming - Part 2 **20**
Understanding inheritance and polymorphism . . . 20
Inheritance . . . 20
Polymorphism . . . 20

Using built-in classes in Python . . . 20
Exercises for Practice . . . 21
Summary . . . 21

Chapter 8: Input and Output **22**
Reading from and writing to files in Python . . . 22
Exercises for Practice . . . 22
Summary . . . 22

Chapter 9: Error Handling **23**
`try / except` block . . . 23
`finally` block . . . 23
`raise` statement . . . 23
Exercises for Practice . . . 23
Summary . . . 24

Chapter 10: Regular Expressions **25**
`search()` function . . . 25
`findall()` function . . . 25
`split()` function . . . 25
`sub()` function . . . 25
`compile()` function . . . 25
Exercises for Practice . . . 26
Summary . . . 26

Chapter 11: Debugging **27**
The `pdp` module . . . 27
Exercises for Practice . . . 27
Summary . . . 27

Chapter 12: Decorators **28**
Understanding decorators and their use in Python . . . 28
Using decorators to modify functions and methods . . . 28
Exercises for practice . . . 28
Summary . . . 29

Chapter 13: Generators **30**
Understanding generators and their use in Python . . . 30
Creating and using generators in Python . . . 30
Exercises for practice . . . 31
Summary . . . 31

Chapter 14: Advanced Topics - Part 1 **32**
Understanding advanced topics in Python . . . 32
Lambda function . . . 32
Map . . . 32
Filter . . . 32
Reduce . . . 32
Exercises for practice . . . 33
Summary . . . 33

Chapter 15: Advanced Topics - Part 2 **34**
Understanding more advanced topics . . . 34
Closure . . . 34
Built-in Libraries . . . 34
Using the unittest module for unit testing . . . 34
Exercises for practice . . . 35
Summary . . . 35

Chapter 16: Data Structure - Part 1 **36**
Understanding data structures . . . 36
Stack . . . 36

Queue . . . 36
Linked List . . . 37
Implementing data structures in python . . . 37
Exercises for practice . . . 37
Summary . . . 37

Chapter 17: Data Structure - Part 2 **38**
Understanding more advanced data structures . . . 38
Trees . . . 38
Tries . . . 38
Graphs . . . 39
Implementing data structures in python . . . 40
Exercises for practice . . . 40
Summary . . . 40

Chapter 18: Algorithms - Part 1 **41**
Understanding Basic Algorithms . . . 41
Implementing Algorithms in Python . . . 41
Exercises for Practice . . . 42

Chapter 19: Algorithms - Part 2 **43**
Dynamic Programming . . . 43
Greedy Algorithms . . . 43
Implementing Algorithms in Python . . . 43
Exercises for Practice . . . 43
Summary . . . 43

Chapter 20: Multithreading and Concurrency **45**
Understanding concepts of multithreading and concurrency . . . 45
The `threading` module . . . 45
The `multiprocessing` module . . . 45
Exercises for practice . . . 45
Summary . . . 45

Chapter 21: GUI Programming **46**
Understanding concepts of GUI programming . . . 46
Using Python's tkinter module . . . 46
Exercises for practice . . . 46
Summary . . . 46

Chapter 22: Networking **47**
Understanding concepts of Networking . . . 47
Using Python's socket module . . . 47
Exercises for practice . . . 47
Summary . . . 47

Chapter 23: Web Scraping **48**
Understanding concepts of web scraping . . . 48
Using Python's `BeautifulSoup` and `Scrapy` module . . . 48
Exercises for practice . . . 48
Summary . . . 48

Chapter 24: Web Development **49**
Understanding concepts of web development . . . 49
Using Python's `Flask` and `Django` framework . . . 49
Exercises for practice . . . 49
Summary . . . 49

Chapter 25: Database **50**
Understanding concepts of database . . . 50
Using Python's `SQLite`, `MySQL` and `MongoDB` . . . 50
`sqlite3` module . . . 50

`mysql-connector-python` module . . . 50
`pymongo` module . . . 50
Exercises for practice . . . 51
Summary . . . 51

Chapter 26: Machine Learning **52**
Understanding concepts of machine learning . . . 52
Using Python's scikit-learn and TensorFlow . . . 52
The `sklearn` library . . . 52
Exercises for Practice . . . 53
Summary . . . 53

Chapter 27: Natural Language Processing **54**
Understanding concepts of NLP . . . 54
Using Python's NLTK and spaCy . . . 54
The `nltk` library . . . 54
`spaCy` library . . . 54
Exercises for Practice . . . 54
Summary . . . 54

Chapter 28: Blockchain **55**
Understanding concepts of blockchain . . . 55
Using Python's blockchain libraries . . . 55
The `bitcoin` library . . . 55
The `web3` library . . . 55
Exercises for practice . . . 55
Summary . . . 55

Chapter 29: Quantum Computing **56**
Understanding concepts of Quantum Computing . . . 56
Using Python's Quantum Computing libraries . . . 56
The `qiskit` library . . . 56
The `pyquil` library . . . 57
Exercises for Practice . . . 57
Summary . . . 57

Chapter 30: Robotics **58**
Understanding concepts of Robotics . . . 58
Using Python's Robotics Libraries . . . 58
The `rospy` library . . . 58
The `pyrobot` library . . . 58
Exercises for Practice . . . 59
Summary . . . 59

Chapter 31: Cloud Computing **60**
Understanding concepts of Cloud Computing . . . 60
Using Python's Cloud Computing Libraries . . . 60
The `boto3` library . . . 60
The `openstack` library . . . 60
Exercises for Practice . . . 61
Summary . . . 61

Chapter 32: Big Data **62**
Understanding concepts of Big Data . . . 62
Using Python's Big Data Libraries . . . 62
The `pyspark` library . . . 62
The `dask` library . . . 62
Exercises for Practice . . . 62
Summary . . . 63

Chapter 33: Cyber Security **64**
Understanding concepts of Cyber Security . . . 64

Using Python's Cybersecurity Libraries . . . 64
The `scapy` library . . . 64
The `cryptography` library . . . 64
Exercises for Practice . . . 64
Summary . . . 65

Exercises and Projects Solutions for Chapter 2: Variables and Data Types 66
Variable Input and Data Type . . . 66
Input Dictionary and Print . . . 66
Loop, Numbers, and Squares . . . 66

Exercises and Projects Solutions for Chapter 3: Control Structures 67
Even/Odd Number Check . . . 67
Fibonacci Sequence Printing . . . 67
Continuous Input Until "stop" . . . 67

Exercises and Projects Solutions for Chapter 4: Functions 68
String Reversal Function . . . 68
Number Sum Function . . . 68
Largest List Element . . . 68

Exercises and Projects Solutions for Chapter 5: Modules 69
Random Number Generation . . . 69
Current Date and Time . . . 69

Exercises and Projects Solutions for Chapter 6: Object-Oriented Programming - Part 1 70
Person Name and Age Class . . . 70
Car Make, Model, Year Class . . . 70
Bank Account Balance and Number Class . . . 70

Exercises and Projects Solutions for Chapter 7: Object-Oriented Programming - Part 2 71
`Inherit` Shape class with Circle, Rectangle and `override` area `method` . . . 71
Electric/Gas Vehicle Comparison . . . 71
Student Name, Age, ID Class . . . 71

Exercises and Projects Solutions for Chapter 8: Input and Output 72
Word Count Text File Reader . . . 72
User Input Text File Writer . . . 72
CSV Table Print Reader . . . 72

Exercises and Projects Solutions for Chapter 9: Error Handling 73
Factorial Calculation and Exception Handling . . . 73
Division calculation and exception handling: . . . 73
Second largest element calculation and exception handling: . . . 73

Exercises and Projects Solutions for Chapter 10: Regular Expressions 74
Email address extraction: . . . 74
Phone number validation: . . . 74
Removing whitespace: . . . 74

Exercises and Projects Solutions for Chapter 11: Debugging 75
Syntax Error Debugging . . . 75
Semantic Error Debugging . . . 75
Second Largest Element with pdb . . . 75

Exercises and Projects Solutions for Chapter 12: Decorators 76
Logging decorator: . . . 76
Memoization decorator: . . . 76
Type checking decorator: . . . 76

Exercises and Projects Solutions for Chapter 13: Generators 78
Fibonacci sequence generator: . . . 78
Permutation generator for a list: . . . 78

Power of 2 generator: . . . 78

Exercises and Projects Solutions for Chapter 14: Advanced Topics - Part 1 79
Linear search function: . . . 79
Binary search function: . . . 79
Bubble sort and linear search combination: . . . 79

Exercises and Projects Solutions for Chapter 15: Advanced Topics - Part 2 81
Shortest path in a weighted graph using dynamic programming: . . . 81
Knapsack problem using greedy algorithm: . . . 81
Solution using dynamic programming: . . . 82

Exercises and Projects Solutions for Chapter 16: Data Structure - Part 1 83
Custom Stack with Minimum Element . . . 83
Custom Queue with Maximum Size . . . 83
Custom Linked List Reversal . . . 84

Exercises and Projects Solutions for Chapter 17: Data Structure - Part 2 85
Binary Search Tree Class . . . 85
Trie Word Insertion/Search/Removal . . . 86
Basic Graph Class for shortest path . . . 87

Exercises and Projects Solutions for Chapter 18: Algorithms - Part 1 89
Linear Search for Sorted List . . . 89
Binary search . . . 89
Target not found . . . 89

Exercises and Projects Solutions for Chapter 19: Algorithms - Part 2 90
Shortest Path in Weighted Graph . . . 90
Knapsack Problem with Greedy Algorithm . . . 90

Exercises and Projects Solutions for Chapter 20: Multithreading and Concurrency 92
Multiprocessing for Factorial Calculation . . . 92
Here's an example of a problem that can be solved using multithreading: . . . 92

Exercises and Projects Solutions for Chapter 21: GUI Programming 93
Simple image viewer . . . 93
Simple to-do list application . . . 93

Exercises and Projects Solutions for Chapter 22: Networking 95
Simple chat client using sockets . . . 95
Simple file transfer program using sockets . . . 95
Multiplayer game server . . . 95

Exercises and Projects Solutions for Chapter 23: Web Scraping 97
Scrapes website for image links . . . 97
Scraping website for product prices . . . 97
Real estate agents . . . 97

Exercises and Projects Solutions for Chapter 24: Web Development 99
Simple web app for file upload/download using Flask . . . 99
Simple web app for registration/login using Django . . . 99
Web app for resume storage and keyword search . . . 100

Exercises and Projects Solutions for Chapter 25: Database 102
Simple CRUD app using SQLite . . . 102
Simple data analysis on MySQL database . . . 102
Program for product sales reports using MongoDB . . . 102

Exercises and Projects Solutions for Chapter 26: Machine Learning 104
Simple classification model using `scikit-learn` on Iris dataset . . . 104
Simple convolutional neural network using TensorFlow on CIFAR-10 dataset . . . 104

Exercises and Projects Solutions for Chapter 27: Natural Language Processing **106**
Simple text classifier using `NLTK` on movie reviews . . . 106
Named entity extraction program using `spaCy` . . . 106
Text summarization program using `NLTK` . . . 106

Exercises and Projects Solutions for Chapter 28: Blockchain **108**
Bitcoin price retrieval program using `Bitcoin-Python` . . . 108
Ethereum wallet creation and Ether transfer program using `web3.py` . . . 108
Smart contracts with web3.py . . . 108

Exercises and Projects Solutions for Chapter 29: Quantum Computing **109**
Bell state in Qiskit . . . 109
GHZ state creation and measurement program using PyQuil . . . 109
Deutsch-Jozsa algorithm using `PyQuil` . . . 109

Exercises and Projects Solutions for Chapter 30: Robotics **111**
Robotic car control program using `ROS` . . . 111
Robotic arm control program using `PyRobot` . . . 111
EC2 instance creation program using `boto3` . . . 111

Exercises and Projects Solutions for Chapter 31: Cloud Computing **112**
Image upload program to `OpenStack` using `openstack` library . . . 112
Average calculation program on large dataset using `PySpark` . . . 112
Data filtering program on large dataset using `Dask` . . . 112

Exercises and Projects Solutions for Chapter 32: Big Data **113**
`PySpark` column average . . . 113
`Dask` dataset filtering . . . 113
`Dask` data aggregation . . . 113

Exercises and Projects Solutions for Chapter 33: Cyber Security **114**
`Scapy` ARP spoof detection . . . 114
`Cryptography` file encryption/decryption . . . 114
`Scapy` man-in-the-middle detection . . . 114

The Bibliography **116**

About the Author **117**

Preface

Welcome to "Python Programming: A Comprehensive Guide to Software Development," a book that aims to provide a complete and accessible introduction to programming with Python.

In this book, you will learn how to write computer programs, and you will develop a solid understanding of the basics of Python programming. The book is written in simple and easy-to-understand language, making it ideal for beginners who want to learn programming.

As the author of this book, I have years of experience in software development and teaching. I am passionate about making programming accessible to everyone, and I believe that Python is an excellent language to learn programming because of its simplicity and versatility.

The book is organized into chapters, each of which covers a different aspect of Python programming. Each chapter contains Introduction, Concepts and Examples, Exercises and Practice, and Summary and Project Ideas sections. The material covered in each chapter will be progressively more challenging, preparing you for more advanced topics like machine learning.

Even if you have no prior programming experience, you will be able to follow along and work through the examples in this book. You will also have plenty of opportunities to practice and test the code on your own computer. If you need further resources, suggestions will be provided throughout the book.

Before you dive into the book, it is beneficial to have a basic understanding of computer usage and the command-line interface, but it's not necessary. The book is written to help you learn programming step by step, from the basics to more advanced concepts.

I want to thank everyone who has helped me in the process of writing this book. Your encouragement and support have been invaluable. I also want to encourage you to experiment with the code and to reach out to me with any questions you might have along the way.

I hope that this book will inspire you to learn more about programming and to develop your skills in software development. Happy learning!

How to read this book

How to Read This Book:

This book “Learn programming with Python” is designed to take you through the fundamentals of programming with Python, all the way to advanced and expert levels. The chapters are divided into three sections to match your experience level.

If you are new to programming or have limited experience, we recommend starting with Chapters 1 to 13. These chapters are beginner-friendly and will cover the basics of programming with Python, including variables, loops, functions, and data structures.

If you have some experience with programming, you can skip the beginner chapters and move to Chapters 14 to 25. These chapters are more advanced and cover topics such as object-oriented programming, algorithms, and working with files.

For experienced programmers looking to expand their knowledge of Python, Chapters 26 to 33 are designed for you. These chapters will dive deep into advanced topics, such as web development, machine learning, and data science.

No matter your experience level, it's important to work through the chapters in order, as the content builds on itself. Each chapter will provide exercises and coding challenges to help solidify your understanding of the concepts covered.

We hope you find this book to be a useful resource in your journey to mastering programming with Python. Happy coding!