

Python Basics: A Practical Introduction

to Python 3

Real Python

## Contents

1	Introduction	20
2	Setting Up Python	29
3	Your First Python Program	42
4	Strings and String Methods	62
5	Numbers and Math	102
6	Functions and Loops	132
7	Finding and Fixing Code Bugs	164
8	Conditional Logic and Control Flow	181
9	Tuples, Lists, and Dictionaries	226
10	Object-Oriented Programming (OOP)	276
11	Modules and Packages	298
12	File Input and Output	320
13	Installing Packages With pip	379
14	Creating and Modifying PDF Files	394
15	Working With Databases	442
16	Interacting With the Web	458
17	Scientific Computing and Graphing	487
18	Graphical User Interfaces	523
19	Final Thoughts and Next Steps	620
19.4	Acknowledgements	

## Chapter 1

### Introduction

Hello, and welcome to Python Basics: A Practical Introduction to Python 3. I hope you're ready to learn why so many professional and hobbyist developers are drawn to Python and how you can begin using it on your own projects, small and large, right away. This book is targeted at beginners who either know a little programming but not the Python language and ecosystem or are starting fresh with no programming experience whatsoever. If you don't have a computer science degree, don't worry. David, Dan, Joanna, and Fletcher will guide you through the important computing concepts while teaching you the Python basics and, just as importantly, skipping the unnecessary details at first.

**Python Is a Full-Spectrum Language** When learning a new programming language, you don't yet have the experience to judge how well it will serve you in the long run. If you're considering learning Python, let me assure you that this is a good choice. One key reason is that Python is a full-spectrum language. What do I mean by this? Some languages are very good for beginners. They hold your hand and make programming super easy. We can go to the extreme and look at visual languages such as Scratch. In Scratch, you get blocks that represent programming concepts like variables, loops, method calls, and so on, and you drag and drop them on a visual surface. Scratch may be easy to get started with for sim13 Contents ple programs, but you cannot build professional applications with it. Name one Fortune 500 company that powers its core business logic with Scratch. Come up empty? Me too, because that would be insanity. Other languages are incredibly powerful for expert developers. The most popular one in this category is likely C++ and its close relative, C. Whichever web browser you used today was likely written in C or C++. Your operating system running that browser was very likely also built with C/C++. Your favorite first-person shooter or strategy video game? You nailed it: C/C++. You can do amazing things with these languages, but they are wholly unwelcoming to newcomers looking for a gentle introduction.

## Chapter 2

### Setting Up Python

This book is about programming computers with Python. You could read this book from cover to cover without ever touching a keyboard, but you'd miss out on the fun part—coding! To get the most out of this book, you need a computer with Python installed on it and a way to create, edit, and save Python code files. In this chapter, you'll learn how to:

Chapter 2 Setting Up Python This book is about programming computers with Python. You could read this book from cover to cover without ever touching a keyboard, but you'd miss out on the fun part—coding! To get the most out of this book, you need a computer with Python installed on it and a way to create, edit, and save Python code files. In this chapter, you'll learn how to:

- Install the latest version of Python 3 on your computer
- Open IDLE, Python's built-in Integrated Development and Learning Environment

Let's get started! 29

#### 2.1 A Note on Python Versions

2.1 A Note on Python Versions Many operating systems, including macOS and Linux, come with Python preinstalled. The version of Python that comes with your operating system is called the system Python. The system Python is used by your operating system and is usually out of date. It's essential that you have the most recent version of Python so that you can successfully follow along with the examples in this book. Important Do not attempt to uninstall the system Python! You can have multiple versions of Python installed on your computer. In this chapter, you'll install the latest version of Python 3 alongside any system Python that may already exist on your machine. Note Even if you already have Python 3.9 installed, it's still a good idea to skim this chapter to double-check that your environment is set up for following along with this book. This chapter is split into three sections: Windows, macOS, and Ubuntu Linux. Find the section for your operating system and follow the steps to get set up, then skip ahead to the next chapter. If you have a different operating system, then check out Real Python's "Python 3 Installation & Setup Guide" to see if your OS is covered. Readers on tablets and mobile devices can refer to the "Online Python Interpreters" section for some browser-based options. 30

#### 2.2 Windows

2.2 Windows Follow these steps to install Python 3 and open IDLE on Windows.

## Chapter 3

**Your First Python Program** Now that you have the latest version of Python installed on your computer, it's time to start coding! In this chapter, you will:

- Write your first Python program
- Learn what happens when you run a program with an error
- Learn how to declare a variable and inspect its value
- Learn how to write comments

Ready to begin your Python journey? Let's go!

### 3.1. Write a Python Program

If you don't already have IDLE open, then go ahead and open it. There are two main windows that you'll work with in IDLE: the interactive window, which is the one that opens when you start IDLE, and the editor window. You can type code into both the interactive window and the editor window. The difference between the two windows is in how they execute code. In this section, you'll learn how to execute Python code in both windows.

**The Interactive Window** IDLE's interactive window contains a Python shell, which is a textual user interface used to interact with the Python language. You can type a bit of Python code into the interactive window and press Enter to immediately see the results. Hence the name interactive window. The interactive window opens automatically when you start IDLE. You'll see the following text, with some minor differences depending on your setup, displayed at the top of the window: Python 3.9.0 (tags/v3.9.0:1b293b6) [MSC v.1916 32 bit (Intel)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> This text shows the version of Python that IDLE is running. You can also see information about your operating system and some commands you can use to get help and view information about Python. The >>> symbol in the last line is called the prompt. This is where you'll type in your code.

### 3.1. Write a Python Program

Go ahead and type `1 + 1` at the prompt and press Enter: >>> `1 + 1` 2 >>> Python evaluates the expression, displays the result (2), then displays another prompt. Every time you run some code in the interactive window, a new prompt appears directly below the result. Executing Python in the interactive window can be described as a loop with three steps: 1. Python reads the code entered at the prompt. 2. Python evaluates the code. 3. Python prints the result and waits for more input. This loop is commonly referred to as a read-evaluate-print loop and is abbreviated as REPL. Python programmers sometimes refer to the Python shell as the Python REPL, or just "the REPL" for short. Let's try something a little more interesting than adding numbers. A rite of passage for every programmer is writing a program that prints the phrase "Hello, World" on the screen. At the prompt in the interactive window, type the word `print` followed by a set of parentheses with the text "Hello, World" inside: