

- » Discovering why Python is hot
- » Finding the tools for success
- » Writing Python in VS Code

Chapter 1

Starting with Python

Because you're reading this chapter, you probably realize that Python is a great language to know if you're looking for a good job in programming, or if you want to expand your existing programming skills into exciting cutting-edge technologies such as artificial intelligence (AI), machine learning (ML), data science, or robotics, or even if you're just building apps in general. So we're not going to try to sell you on Python. It sells itself.

Our approach leans heavily toward the hands-on. A common failure in many programming tutorials is that they already assume you're a professional programmer in some language, and they skip over things they assume you already know.

This book is different in that we *don't* assume that you're already programming in Python or some other language. We *do* assume that you can use a computer and understand basics such as files and folders.

We also assume you're not up for settling down in an easy chair in front of the fireplace to read page after page of theoretical stuff about Python, like some kind of boring novel. You don't have that much free time to kill. So we're going to get right into it and focus on *doing*, hands-on, because that's the only way most of us learn. We've never seen anyone read a book about Python and then sit

at a computer and write Python like a pro. Human brains don't work that way. We learn through practice and repetition, and that requires being hands-on.

Why Python Is Hot

We promised we weren't going to spend a bunch of time trying to sell you on Python, and that's not our intent here. But we would like to talk briefly about *why* it's so hot.

Here are the main reasons cited for Python's current popularity:

- » Python is relatively easy to learn.
- » Everything you need to learn (and do) in Python is free.
- » Python offers more ready-made tools for current hot technologies such as data science, machine learning, artificial intelligence, and robotics than most other languages.

Choosing the Right Python

There are different *versions* of Python out roaming the world, prompting many a beginner to wonder things such as

- » Why are there different versions?
- » How are they different?
- » Which one should I learn?

All good questions, and we'll start with the first. A version is kind of like a car year. You can go out and buy a 1968 Ford Mustang, a 1990 Ford Mustang, a 2019 Ford Mustang, or a 2020 Ford Mustang. They're all Ford Mustangs. The only difference is that the one with the highest year number is the most current Ford Mustang. That Mustang is different from the older models in that it has some improvements based on experience with earlier models, as well as features current with the times.

Programming languages (and most other software products) work the same way. But as a rule, we don't ascribe year numbers

to them because they're not released on a yearly basis. They're released whenever they're released. The principle is the same, though. The version with the highest number is the newest, most recent model, sporting improvements based on experience with earlier versions, as well as features relevant to the current times.

In this book, we focus on versions of Python that are current in late 2023 from Python 3.11 and higher. Don't worry about version differences after the first and second digits. Version 3.11.1 is similar enough to version 3.11.2 that version differences aren't important, especially to a beginner. Most of what's in Python is the same across all recent versions. So you need not worry about investing time in learning a version that is or will soon be obsolete.

Tools for Success

Now we need to start getting your computer set up so that you can learn, and do, Python hands-on. For one, you'll need a good Python interpreter and editor. The *editor* lets you type the code, and the *interpreter* lets you run that code. When you run (or execute) code, you're telling the computer to "do whatever my code tells you to do."

The term *code* refers to anything written in a programming language to provide instructions to a computer. The term *coding* is often used to describe the act of writing code. A code editor is an app that lets you type code, in much the same way an app such as Microsoft Word or Apple Pages helps you type regular, plain-English text.

Just as there are many brands of toothpaste, soap, and shampoo in the world, there are many brands of code editors that work well with Python. There isn't a right one or a wrong one, a good one or a bad one, a best one or a worst one. Just a lot of different products that basically do the same thing but vary slightly in their approach and what that editor's creators think is good.

If you've already started learning Python and are happy with whatever you've been using, you're welcome to continue using that and ignore our suggestions. If you're just getting started with this stuff, we suggest you use VS Code, because it's an excellent, free learning environment.

Installing Python and VS Code

The editor we recommend and will be using in this book is called Visual Studio Code, officially. But most often, it is spoken or written as *VS Code*. The main reasons why it's our favorite follow:

- » It is an excellent editor for learning coding.
- » It is an excellent editor for writing code professionally and is used by millions of professional programmers and developers.
- » It's relatively easy to learn and use.
- » It works pretty much the same on Windows, Mac, and Linux.
- » It's free.
- » It integrates beautifully with GitHub Copilot, so you can use modern, generative AI to speed both learning and actual coding.

To use VS Code as your editor for learning and doing Python, you need to download and install Python, VS Code, and a VS Code extension. With luck, you already have some experience working with apps, so this won't be difficult. You will have to follow onscreen instructions as you go along. If faced with any choices you're not sure about along the way, you can just choose the default (suggested) option. Here are the steps to download and install Python and VS Code:

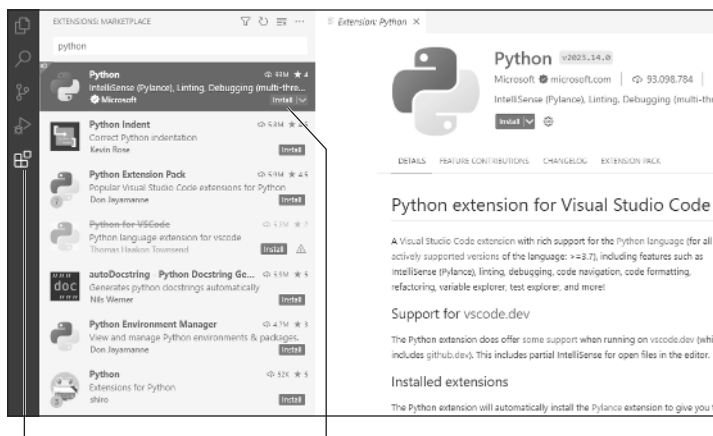
- 1. Use any web browser to browse to www.python.org.**
- 2. Click Download and, if asked to select a version, choose the suggested stable version.**
- 3. Open the folder to which you downloaded Python and double-click the icon for the file you downloaded to install Python.**

You can just follow the onscreen instructions, and accept any suggested defaults, during the installation process.
- 4. Browse to <https://code.visualstudio.com/> and download the current version of VS Code for your operating system.**
- 5. Open the folder to which you downloaded Visual Studio code, double-click the icon for the downloaded file, and follow the onscreen instructions to install VS Code.**

After VS Code is installed, you should be able to start it like any other app in your system. In Windows, click Start and look around on the Start menu for Visual Studio Code icon. On a Mac, you should be able to find it in your Applications folder, or Launchpad.

Installing the Python extension

To use VS Code for Python coding, you need to install the VS Code Python extension for Python. When you open VS Code, you will see some icons listed down the left side of the window. Placing the mouse cursor over any icon reveals its name. Click the Extensions icon, shown in Figure 1-1, and then enter **Python** in the Search box at the top of the Extensions panel. Click the Install button with the Python extension from Microsoft (see Figure 1-1).



Extensions icon Install the Python extension from Microsoft

FIGURE 1-1: Obtaining the Python extension in VS Code.

When you've finished installing the Python extension, you might notice that both Python and Pylance were added as extensions to VS Code. Don't worry; that's normal. Pylance just gives you some additional capabilities that make it easier to learn and write Python code within the VS Code editor. To ensure that the extension is activated, exit VS Code and then restart it.

Letting AI write your Python code

Modern generative AI is perfectly capable of writing Python code for you. It's not as simple as commanding it to "Write a Python

app that will make me a billionaire,” however. It doesn’t work that way — yet. Unfortunately. You need to break things down into smaller chunks, and probably use accurate tech terminology, too. In other words, you still have to learn enough Python to be able to write your AI prompts accurately. Virtually all of these prompts — no matter which AI service you use — will start with “Write python code for . . .” because AI can do a lot of things. If you don’t tell it, specifically, that you want it to write Python code, you might get no code, HTML, JavaScript, or whatever. So just make sure you understand that, first and foremost.

As we write this in late 2023, generative AI is still fairly new and evolving rapidly. We can’t make any promises in terms of pricing or availability. Those things are likely to change often over the coming years, and competing businesses jockey for position and market share. But as of this writing, you can prompt the following AI services to write Python code:

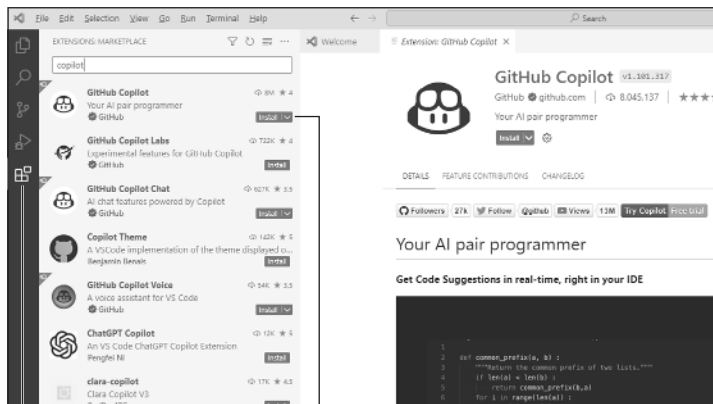
- » ChatGPT (<https://chat.openai.com>)
- » Claude (<https://claude.ai>)
- » Google Bard (<https://bard.google.com>)
- » Microsoft Copilot (<https://copilot.microsoft.com>)

Most of these tools are free (right now), but again, we can’t make any promises about the future.

Using GitHub Copilot

GitHub Copilot is another AI tool that’s capable of writing code for you. It’s based on OpenAI’s GPT-4, like ChatGPT. However, it’s specifically geared toward working with code and integrates directly into VS Code. You’re certainly not *required* to use GitHub Copilot to learn Python or use this book, but you might find that it really helps your learning process. As we write this book, GitHub is offering Copilot for free to students. It offers some paid plans, too, starting at \$10 a month. To use Copilot, you need to sign up for GitHub and purchase (or request) access to Copilot. Again, this tool is so new that any instructions we give here are subject to change. You may need to search Google or YouTube for *use Copilot with VS Code* to find the most up-to-date instructions. Basically, here’s how it works:

1. **If you don't already have a GitHub account, go to <https://github.com> and create an account.**
Make sure you know your GitHub username and password, because you'll need them to set up your account.
2. **Open VS Code if it isn't already open.**
3. **Click Extensions in the left column, and then enter Copilot in the Search box to search for Copilot.**
A list of Copilot extensions appears.
4. **Click the Install button at the right of the plain-old GitHub Copilot extension (not "Copilot Labs" or any of the others that appear), as shown in Figure 1-2.**



Extensions icon

Install GitHub Copilot

FIGURE 1-2: The GitHub Copilot extension in VS Code.

You'll see some instructions and tips on a pane to the right. You need not do anything with those right now, though. Near the lower-left corner of VS Code, you should see an avatar icon for Accounts (see Figure 1-3). Click that icon and choose Sign In with GitHub to Use GitHub Copilot. Follow the on-screen instructions to sign into your GitHub account and set up Copilot. But remember: Setting up a Copilot account isn't a requirement, just an option. So don't feel you have to complete the process of purchasing Copilot right now. But if you do add Copilot as an extension, you should see its name under Installed Extensions whenever you're viewing extensions in VS Code.

Also, near the lower-right corner of the screen, you'll see a tiny Copilot icon (also shown in Figure 1-3). You can click that icon at any time to deactivate Copilot if you feel it's in your way while learning. Click it again to activate Copilot whenever you're ready.

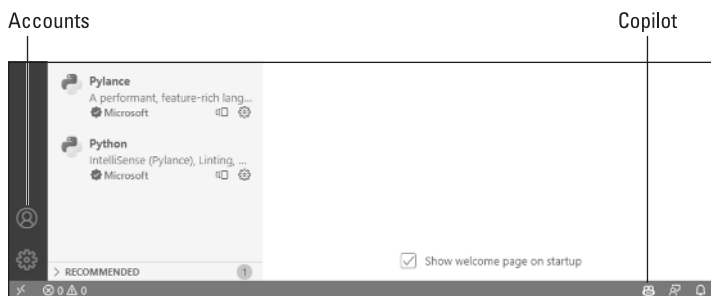


FIGURE 1-3: The Accounts and Copilot icons in VS Code.

The simple tasks you've completed in this chapter will serve you well through your learning process, as well as your professional programming after you've mastered the basics. Head on over to Chapter 2, and we'll delve a bit deeper into Python and using the tools you now have available on your computer.