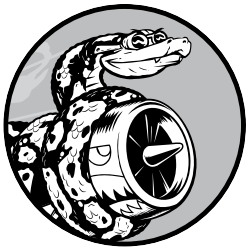
1

GETTING STARTED



In this chapter, you'll run your first Python program, *hello_world.py*. First, you'll need to check whether a recent version of Python is installed on your computer; if it isn't, you'll install it. You'll also install a text editor to work with your Python programs. Text editors recognize Python code and highlight sections as you write, making it easy to understand your code's structure.

Setting Up Your Programming Environment

Python differs slightly on different operating systems, so you'll need to keep a few considerations in mind. In the following sections, we'll make sure Python is set up correctly on your system.

Python Versions

Every programming language evolves as new ideas and technologies emerge, and the developers of Python have continually made the language more versatile and powerful. As of this writing, the latest version is Python 3.11, but everything in this book should run on Python 3.9 or later. In this section, we'll find out if Python is already installed on your system and whether you need to install a newer version. Appendix A contains additional details about installing the latest version of Python on each major operating system as well.

Running Snippets of Python Code

You can run Python's interpreter in a terminal window, allowing you to try bits of Python code without having to save and run an entire program.

Throughout this book, you'll see code snippets that look like this:

```
>>> print("Hello Python interpreter!")
Hello Python interpreter!
```

The three angle brackets (>>>) prompt, which we'll refer to as a *Python prompt*, indicates that you should be using the terminal window. The bold text is the code you should type in and then execute by pressing ENTER. Most of the examples in this book are small, self-contained programs that you'll run from your text editor rather than the terminal, because you'll write most of your code in the text editor. But sometimes, basic concepts will be shown in a series of snippets run through a Python terminal session to demonstrate particular concepts more efficiently. When you see three angle brackets in a code listing, you're looking at code and output from a terminal session. We'll try coding in the interpreter on your system in a moment.

We'll also use a text editor to create a simple program called *Hello World!* that has become a staple of learning to program. There's a long-held tradition in the programming world that printing the message `Hello world!` to the screen as your first program in a new language will bring you good luck. Such a simple program serves a very real purpose. If it runs correctly on your system, then any Python program you write should work as well.

About the VS Code Editor

VS Code is a powerful, professional-quality text editor that's free and beginner-friendly. VS Code is great for both simple and complex projects, so if you become comfortable using it while learning Python, you can continue using it as you progress to larger and more complicated projects. VS Code can be installed on all modern operating systems, and it supports most programming languages, including Python.

Appendix B provides information on other text editors. If you're curious about the other options, you might want to skim that appendix at this

point. If you want to begin programming quickly, you can use VS Code to start. Then you can consider other editors, once you've gained some experience as a programmer. In this chapter, I'll walk you through installing VS Code on your operating system.

NOTE *If you already have a text editor installed and you know how to configure it to run Python programs, you are welcome to use that editor instead.*

Python on Different Operating Systems

Python is a cross-platform programming language, which means it runs on all the major operating systems. Any Python program you write should run on any modern computer that has Python installed. However, the methods for setting up Python on different operating systems vary slightly.

In this section, you'll learn how to set up Python on your system. You'll first check whether a recent version of Python is installed on your system, and install it if it's not. Then you'll install VS Code. These are the only two steps that are different for each operating system.

In the sections that follow, you'll run *hello_world.py* and troubleshoot anything that doesn't work. I'll walk you through this process for each operating system, so you'll have a Python programming environment that you can rely on.

Python on Windows

Windows doesn't usually come with Python, so you'll probably need to install it and then install VS Code.

Installing Python

First, check whether Python is installed on your system. Open a command window by entering `command` into the Start menu and clicking the **Command Prompt** app. In the terminal window, enter `python` in lowercase. If you get a Python prompt (`>>>`) in response, Python is installed on your system. If you see an error message telling you that `python` is not a recognized command, or if the Microsoft store opens, Python isn't installed. Close the Microsoft store if it opens; it's better to download an official installer than to use Microsoft's version.

If Python is not installed on your system, or if you see a version earlier than Python 3.9, you need to download a Python installer for Windows. Go to *https://python.org* and hover over the **Downloads** link. You should see a button for downloading the latest version of Python. Click the button, which should automatically start downloading the correct installer for your system. After you've downloaded the file, run the installer. Make sure you select the option **Add Python to PATH**, which will make it easier to configure your system correctly. Figure 1-1 shows this option selected.

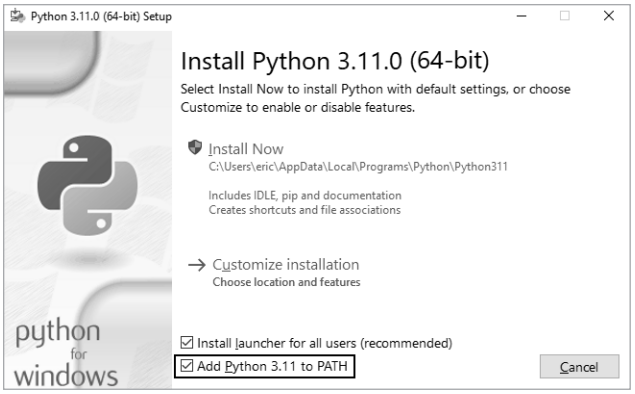


Figure 1-1: Make sure you select the checkbox labeled Add Python to PATH.

Running Python in a Terminal Session

Open a new command window and enter **`python`** in lowercase. You should see a Python prompt (>>>), which means Windows has found the version of Python you just installed.

```
C:\> python
Python 3.x.x (main, Jun . . . , 13:29:14) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

NOTE *If you don't see this output or something similar, see the more detailed setup instructions in Appendix A.*

Enter the following line in your Python session:

```
>>> print("Hello Python interpreter!")
Hello Python interpreter!
>>>
```

You should see the output `Hello Python interpreter!` Anytime you want to run a snippet of Python code, open a command window and start a Python terminal session. To close the terminal session, press CTRL-Z and then press ENTER, or enter the command **`exit()`**.

Installing VS Code

You can download an installer for VS Code at *https://code.visualstudio.com*. Click the **Download for Windows** button and run the installer. Skip the following sections about macOS and Linux, and follow the steps in "Running a Hello World Program" on page 9.

Python on macOS

Python is not installed by default on the latest versions of macOS, so you'll need to install it if you haven't already done so. In this section, you'll install the latest version of Python, and then install VS Code and make sure it's configured correctly.

NOTE *Python 2 was included on older versions of macOS, but it's an outdated version that you shouldn't use.*

Checking Whether Python 3 Is Installed

Open a terminal window by going to **Applications ▸ Utilities ▸ Terminal**. You can also press ⌘-spacebar, type **`terminal`**, and then press ENTER. To see if you have a recent enough version of Python installed, enter **`python3`**. You'll most likely see a message about installing the *command line developer tools*. It's better to install these tools after installing Python, so if this message appears, cancel the pop-up window.

If the output shows you have Python 3.9 or a later version installed, you can skip the next section and go to "Running Python in a Terminal Session." If you see any version earlier than Python 3.9, follow the instructions in the next section to install the latest version.

Note that on macOS, whenever you see the `python` command in this book, you need to use the `python3` command instead to make sure you're using Python 3. On most macOS systems, the `python` command either points to an outdated version of Python that should only be used by internal system tools, or it points to nothing and generates an error message.

Installing the Latest Version of Python

You can find a Python installer for your system at *https://python.org*. Hover over the **Download** link, and you should see a button for downloading the latest version of Python. Click the button, which should automatically start downloading the correct installer for your system. After the file downloads, run the installer.

After the installer runs, a Finder window should appear. Double-click the *Install Certificates.command* file. Running this file will allow you to more easily install additional libraries that you'll need for real-world projects, including the projects in the second half of this book.

Running Python in a Terminal Session

You can now try running snippets of Python code by opening a new terminal window and typing **`python3`**:

```
$ python3
Python 3.x.x (v3.11.0:eb0004c271, Jun . . . , 10:03:01)
[Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

This command starts a Python terminal session. You should see a Python prompt (>>>), which means macOS has found the version of Python you just installed.

Enter the following line in the terminal session:

```
>>> print("Hello Python interpreter!")
Hello Python interpreter!
>>>
```

You should see the message `Hello Python interpreter!`, which should print directly in the current terminal window. You can close the Python interpreter by pressing CTRL-D or by entering the command **`exit()`**.

NOTE *On newer macOS systems, you'll see a percent sign (%) as a terminal prompt instead of a dollar sign ($).*

Installing VS Code

To install the VS Code editor, you need to download the installer at *https://code.visualstudio.com*. Click the **Download** button, and then open a **Finder** window and go to the **Downloads** folder. Drag the **Visual Studio Code** installer to your Applications folder, then double-click the installer to run it.

Skip over the following section about Python on Linux, and follow the steps in "Running a Hello World Program" on page 9.

Python on Linux

Linux systems are designed for programming, so Python is already installed on most Linux computers. The people who write and maintain Linux expect you to do your own programming at some point, and encourage you to do so. For this reason, there's very little to install and only a few settings to change to start programming.

Checking Your Version of Python

Open a terminal window by running the Terminal application on your system (in Ubuntu, you can press CTRL-ALT-T). To find out which version of Python is installed, enter **`python3`** with a lowercase *p*. When Python is installed, this command starts the Python interpreter. You should see output indicating which version of Python is installed. You should also see a Python prompt (>>>) where you can start entering Python commands:

```
$ python3
Python 3.10.4 (main, Apr  . . . , 09:04:19) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

This output indicates that Python 3.10.4 is currently the default version of Python installed on this computer. When you've seen this output, press CTRL-D or enter **`exit()`** to leave the Python prompt and return to a

terminal prompt. Whenever you see the `python` command in this book, enter `python3` instead.

You'll need Python 3.9 or later to run the code in this book. If the Python version installed on your system is earlier than Python 3.9, or if you want to update to the latest version currently available, refer to the instructions in Appendix A.

Running Python in a Terminal Session

You can try running snippets of Python code by opening a terminal and entering `python3`, as you did when checking your version. Do this again, and when you have Python running, enter the following line in the terminal session:

```
>>> print("Hello Python interpreter!")
Hello Python interpreter!
>>>
```

The message should print directly in the current terminal window. Remember that you can close the Python interpreter by pressing CTRL-D or by entering the command `exit()`.

Installing VS Code

On Ubuntu Linux, you can install VS Code from the Ubuntu Software Center. Click the Ubuntu Software icon in your menu and search for *vscode*. Click the app called **Visual Studio Code** (sometimes called *code*), and then click **Install**. Once it's installed, search your system for *VS Code* and launch the app.

Running a Hello World Program

With a recent version of Python and VS Code installed, you're almost ready to run your first Python program written in a text editor. But before doing so, you need to install the Python extension for VS Code.

Installing the Python Extension for VS Code

VS Code works with many different programming languages; to get the most out of it as a Python programmer, you'll need to install the Python extension. This extension adds support for writing, editing, and running Python programs.

To install the Python extension, click the Manage icon, which looks like a gear in the lower-left corner of the VS Code app. In the menu that appears, click **Extensions**. Enter `python` in the search box and click the **Python** extension. (If you see more than one extension named *Python*, choose the one supplied by Microsoft.) Click **Install** and install any additional tools that your system needs to complete the installation. If you see a message that you need to install Python, and you've already done so, you can ignore this message.

NOTE *If you're using macOS and a pop-up asks you to install the* command line developer tools*, click* ***Install****. You may see a message that it will take an excessively long time to install, but it should only take about 10 or 20 minutes on a reasonable internet connection.*

Running hello_world.py

Before you write your first program, make a folder called *python_work* on your desktop for your projects. It's best to use lowercase letters and underscores for spaces in file and folder names, because Python uses these naming conventions. You can make this folder somewhere other than the desktop, but it will be easier to follow some later steps if you save the *python_work* folder directly on your desktop.

Open VS Code, and close the **Get Started** tab if it's still open. Make a new file by clicking **File ▸ New File** or pressing CTRL-N (⌘-N on macOS). Save the file as *hello_world.py* in your *python_work* folder. The extension *.py* tells VS Code that your file is written in Python, and tells it how to run the program and highlight the text in a helpful way.

After you've saved your file, enter the following line in the editor:

hello_world.py

```
print("Hello Python world!")
```

To run your program, select **Run ▸ Run Without Debugging** or press CTRL-F5. A terminal screen should appear at the bottom of the VS Code window, showing your program's output:

```
Hello Python world!
```

You'll likely see some additional output showing the Python interpreter that was used to run your program. If you want to simplify the information that's displayed so you only see your program's output, see Appendix B. You can also find helpful suggestions about how to use VS Code more efficiently in Appendix B.

If you don't see this output, something might have gone wrong in the program. Check every character on the line you entered. Did you accidentally capitalize `print`? Did you forget one or both of the quotation marks or parentheses? Programming languages expect very specific syntax, and if you don't provide that, you'll get errors. If you can't get the program to run, see the suggestions in the next section.

Troubleshooting

If you can't get *hello_world.py* to run, here are a few remedies you can try that are also good general solutions for any programming problem:

- When a program contains a significant error, Python displays a *traceback*, which is an error report. Python looks through the file and tries to identify the problem. Check the traceback; it might give you a clue as to what issue is preventing the program from running.

- Step away from your computer, take a short break, and then try again. Remember that syntax is very important in programming, so something as simple as mismatched quotation marks or mismatched parentheses can prevent a program from running properly. Reread the relevant parts of this chapter, look over your code, and try to find the mistake.
- Start over again. You probably don't need to uninstall any software, but it might make sense to delete your *hello_world.py* file and re-create it from scratch.
- Ask someone else to follow the steps in this chapter, on your computer or a different one, and watch what they do carefully. You might have missed one small step that someone else happens to catch.
- See the additional installation instructions in Appendix A; some of the details included in the Appendix may help you solve your issue.
- Find someone who knows Python and ask them to help you get set up. If you ask around, you might find that you unexpectedly know someone who uses Python.
- The setup instructions in this chapter are also available through this book's companion website at *https://ehmatthes.github.io/pcc_3e*. The online version of these instructions might work better because you can simply cut and paste code and click links to the resources you need.
- Ask for help online. Appendix C provides a number of resources, such as forums and live chat sites, where you can ask for solutions from people who've already worked through the issue you're currently facing.

Never worry that you're bothering experienced programmers. Every programmer has been stuck at some point, and most programmers are happy to help you set up your system correctly. As long as you can state clearly what you're trying to do, what you've already tried, and the results you're getting, there's a good chance someone will be able to help you. As mentioned in the introduction, the Python community is very friendly and welcoming to beginners.

Python should run well on any modern computer. Early setup issues can be frustrating, but they're well worth sorting out. Once you get *hello_world.py* running, you can start to learn Python, and your programming work will become more interesting and satisfying.

Running Python Programs from a Terminal

You'll run most of your programs directly in your text editor. However, sometimes it's useful to run programs from a terminal instead. For example, you might want to run an existing program without opening it for editing.

You can do this on any system with Python installed if you know how to access the directory where the program file is stored. To try this, make sure you've saved the *hello_world.py* file in the *python_work* folder on your desktop.

On Windows

You can use the terminal command `cd`, for *change directory*, to navigate through your filesystem in a command window. The command `dir`, for *directory*, shows you all the files that exist in the current directory.

Open a new terminal window and enter the following commands to run *hello_world.py*:

```
C:\> cd Desktop\python_work
C:\Desktop\python_work> dir
hello_world.py
C:\Desktop\python_work> python hello_world.py
Hello Python world!
```

First, use the **`cd`** command to navigate to the *python_work* folder, which is in the *Desktop* folder. Next, use the **`dir`** command to make sure *hello_world.py* is in this folder. Then run the file using the command **`python hello_world.py`**.

Most of your programs will run fine directly from your editor. However, as your work becomes more complex, you'll want to run some of your programs from a terminal.

On macOS and Linux

Running a Python program from a terminal session is the same on Linux and macOS. You can use the terminal command `cd`, for *change directory*, to navigate through your filesystem in a terminal session. The command `ls`, for *list*, shows you all the nonhidden files that exist in the current directory.

Open a new terminal window and enter the following commands to run *hello_world.py*:

```
~$ cd Desktop/python_work/
~/Desktop/python_work$ ls
hello_world.py
~/Desktop/python_work$ python3 hello_world.py
Hello Python world!
```

First, use the **`cd`** command to navigate to the *python_work* folder, which is in the *Desktop* folder. Next, use the **`ls`** command to make sure *hello_world.py* is in this folder. Then run the file using the command **`python3 hello_world.py`**.

Most of your programs will run fine directly from your editor. But as your work becomes more complex, you'll want to run some of your programs from a terminal.

TRY IT YOURSELF

The exercises in this chapter are exploratory in nature. Starting in Chapter 2, the challenges you'll solve will be based on what you've learned.

1-1. python.org: Explore the Python home page (*https://python.org*) to find topics that interest you. As you become familiar with Python, different parts of the site will be more useful to you.

1-2. Hello World Typos: Open the *hello_world.py* file you just created. Make a typo somewhere in the line and run the program again. Can you make a typo that generates an error? Can you make sense of the error message? Can you make a typo that doesn't generate an error? Why do you think it didn't make an error?

1-3. Infinite Skills: If you had infinite programming skills, what would you build? You're about to learn how to program. If you have an end goal in mind, you'll have an immediate use for your new skills; now is a great time to write brief descriptions of what you want to create. It's a good habit to keep an "ideas" notebook that you can refer to whenever you want to start a new project. Take a few minutes now to describe three programs you want to create.

Summary

In this chapter, you learned a bit about Python in general, and you installed Python on your system if it wasn't already there. You also installed a text editor to make it easier to write Python code. You ran snippets of Python code in a terminal session, and you ran your first program, *hello_world.py*. You probably learned a bit about troubleshooting as well.

In the next chapter, you'll learn about the different kinds of data you can work with in your Python programs, and you'll start to use variables as well.