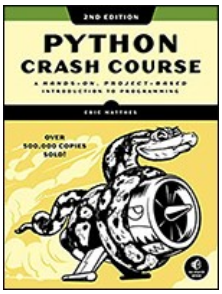

Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 2nd Edition

by Eric Matthes

Chapter 1: Getting Started

Overview

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 3.7, but everything in this book should run on Python 3.6 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 a comprehensive guide to installing the latest version of Python on each major operating system as well.

Some old Python projects still use Python 2, but you should use Python 3. If Python 2 is installed on your system, it's probably there to support some older programs that your system needs. We'll leave this installation as is, and make sure you have a more recent version to work with.

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:

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

The `>>>` prompt indicates that you should be using the terminal window, and the bold text is the code you should type in and then execute by pressing ENTER. Most of the examples in the 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 ❶, 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 a `Hello world!` message 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, any Python program you write should work as well.

About the Sublime Text Editor

Sublime Text is a simple text editor that can be installed on all modern operating systems. Sublime Text lets you run almost all of your programs directly from the editor instead of through a terminal. Your code runs in a terminal session embedded in the Sublime Text window, which makes it easy to see the output.

Sublime Text is a beginner-friendly editor, but many professional programmers use it as well. If you become comfortable using it while learning Python, you can continue using it as you progress to larger and more complicated projects. Sublime Text has a very liberal licensing policy: you can use the editor free of charge as long as you want, but the developers request that you purchase a license if you like it and want to keep using it.

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 Sublime Text to start and consider other editors once you've gained some experience as a programmer. In this chapter, I'll walk you through installing Sublime Text on your operating system.

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 Sublime Text. These are the only two steps that are different for each operating system.

In the sections that follow, you'll run the *Hello World!* program and troubleshoot anything that didn't work. I'll walk you through this process for each operating system, so you'll have a beginner-friendly Python programming environment.

Python on Windows

Windows doesn't always come with Python, so you'll probably need to install it, and then install Sublime Text.

Installing Python

First, check whether Python is installed on your system. Open a command window by entering `command` into the Start menu or by holding down the SHIFT key while right-clicking on your desktop and selecting **Open command window here** from the menu. 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, Python isn't installed.

In that case, or if you see a version of Python earlier than Python 3.6, 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 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 3.7.2 (v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 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, and make sure you see the output `Hello Python interpreter!`

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

Any time 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 Sublime Text

You can download an installer for Sublime Text at *https://sublimetext.com/*. Click the download link and look for a Windows installer. After downloading the installer, run the installer and accept all of its defaults.

Python on macOS

Python is already installed on most macOS systems, but it's most likely an outdated version that you won't want to learn on. In this section, you'll install the latest version of Python, and then you'll install Sublime Text and make sure it's configured correctly.

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 which version of Python is installed, enter `python` with a lowercase *p*—this also starts the Python interpreter within the terminal, allowing you to enter Python commands. You should see output telling you which Python version is installed on your system and a `>>>` prompt where you can start entering Python commands, like this:

```
$ python
Python 2.7.15 (default, Aug 17 2018, 22:39:05)
[GCC 4.2.1 Compatible Apple LLVM 9.1.0 (clang-902.0.39.2)] on darwin
Type "help", "copyright", "credits", or "license" for more information.
>>>
```

This output indicates that Python 2.7.15 is currently the default version installed on this computer. Once you've seen this output, press CTRL-D or enter `exit()` to leave the Python prompt and return to a terminal prompt.

To check whether you have Python 3 installed, enter the command `python3`. You'll probably get an error message, meaning you don't have any versions of Python 3 installed. If the output shows you have Python 3.6 or a later version installed, you can skip ahead to "Running Python in a Terminal Session" on page 8. If Python 3 isn't installed by default, you'll need to install it manually. Note that 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, not Python 2; they differ significantly enough that you'll run into trouble trying to run the code in this book using Python 2.

If you see any version earlier than Python 3.6, follow the instructions in the next section to install the latest version.

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 Python version. Click the button, which should automatically start downloading the correct installer for your system. After the file downloads, run the installer.

When you're finished, enter the following at a terminal prompt:

```
$ python3 --version
Python 3.7.2
```

You should see output similar to this, in which case, you're ready to try out Python. Whenever you see the command **`python`**, make sure you use **`python3`**.

Running Python in a Terminal Session

You can now try running snippets of Python code by opening a terminal and typing **`python3`**. Enter the following line in the terminal session:

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

Your 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 Sublime Text

To install the Sublime Text editor, you need to download the installer at *https://sublimetext.com/*. Click the **Download** link and look for an installer for macOS. After the installer downloads, open it and then drag the Sublime Text icon into your *Applications* folder.

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 and a `>>>` prompt where you can start entering Python commands, like this:

```
$ python3
Python 3.7.2 (default, Dec 27 2018, 04:01:51)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

This output indicates that Python 3.7.2 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 3.6 or later to run the code in this book. If the Python version installed on your system is earlier than Python 3.6, refer to Appendix A to install the latest version.

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 Sublime Text

On Linux, you can install Sublime Text from the Ubuntu Software Center. Click the Ubuntu Software icon in your menu, and search for **Sublime Text**. Click to install it, and then launch it.

Running a Hello World Program

With a recent version of Python and Sublime Text installed, you're almost ready to run your first Python program written in a text editor. But before doing so, you need to make sure Sublime Text is set up to use the correct version of Python on your system. Then you'll write the *Hello World!* program and run it.

Configuring Sublime Text to Use the Correct Python Version

If the `python` command on your system runs Python 3, you won't need to configure anything and can skip to the next section. If you use the `python3` command, you'll need to configure Sublime Text to use the correct Python version when it runs your programs.

Click the Sublime Text icon to launch it, or search for Sublime Text in your system's search bar and then launch it. Go to **Tools ▸ Build System ▸ New Build System**, which will open a new configuration file for you. Delete what you see and enter the following:

Python3.sublime-build

```
{
    "cmd": ["python3", "-u", "$file"],
}
```

This code tells Sublime Text to use your system's `python3` command when running your Python program files. Save the file as *Python3.sublime-build* in the default directory that Sublime Text opens when you choose Save.

Running hello_world.py

Before you write your first program, make a folder called *python_work* somewhere on your system 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.

Open Sublime Text, and save an empty Python file (**File ▸ Save As**) called *hello_world.py* in your *python_work* folder. The extension *.py* tells Sublime Text that the code in your file is written in Python, which 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 text editor:

hello_world.py

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

If the `python` command works on your system, you can run your program by selecting **Tools ▸ Build** in the menu or by pressing CTRL-B (⌘-B on macOS). If you had to configure Sublime Text in the previous section, select **Tools ▸ Build System** and then select **Python 3**. From now on you'll be able to select **Tools ▸ Build** or just press CTRL-B (or ⌘-B) to run your programs.

A terminal screen should appear at the bottom of the Sublime Text window, showing the following output:

```
Hello Python world!
[Finished in 0.1s]
```

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 even a missing colon, a mismatched quotation mark, 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.
- 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 the book's companion website at *https://nostarch.com/pythoncrashcourse2e/*. The online version of these instructions might work better for you because you can simply cut and paste code.
- 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

Most of the programs you write in your text editor you'll run directly from the editor. But 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*:

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

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

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.

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*:

```
❶ ~$ cd Desktop/python_work/
❷ ~/Desktop/python_work$ ls
   hello_world.py
❸ ~/Desktop/python_work$ python hello_world.py
   Hello Python world!
```

At ❶ you use the `cd` command to navigate to the *python_work* folder, which is in the *Desktop* folder. Next, you use the `ls` command to make sure *hello_world.py* is in this folder ❷. Then you run the file using the command `python hello_world.py` ❸.

It's that simple. You just use the `python` (or `python3`) command to run Python programs.

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 draft 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 use variables as well.