Setting Up Python3 on Mac

Beginner's Edition

ABSTRACT

These instructions are designed to help beginners set up a basic Python3 environment on their Mac, including installing Python3, writing basic Python code, and using the terminal for package management.

Andrew Kelton

ENC 3241

Introduction

Introduction

Python3 is a versatile and popular programming language used in fields such as, data science, AI, and web development. This guide walks you through setting up a Python3 development environment on a Mac.

In this document you will learn how to:

- ✓ Install Python3 on your Mac.
- ✓ Configure the terminal to run Python scripts.
- ✓ Install necessary packages using pip3.
- ✓ Write and run your first Python program.

This guide is aimed at beginners, no prior programming experience is required.

Prerequisites

Required items:

- ✓ A Mac computer running macOS
- ✓ Access to the Terminal (pre-installed on Mac)
- ✓ Stable internet connection

Warnings/Precautions

Although download python is relatively safe. Beware there are still some issues that can occur when downloading the language. For errors related to the instructions, refer to *Common Issues and Troubleshooting* (pg. 10).

Malicious Files Downloaded

This can occur if you click a *bad link* and download a malicious version of Python. If this does happen, the executable code downloaded onto your computer can make changes to your device and severely hurt your computer's health. Even though this is extremely rare, I recommend use the verified link I have provided in this guide.

Overwriting of Previous Python3 Download

Depending how your computer is set up to download Python, a newer Python download can overwrite an older version. This will cause all previous libraries and usable code to possibly be deprecated in the newer version.

Errors When Importing Packages

When importing some packages, you may notice an error *Module not Found*. If this occurs, please refer to step 2 (pg. 6).

Introduction

“Quick Start” Guide

How to Write Python Code

How to use Python’s syntax? Python Syntax Tutorial

To fully utilize Python, you need an Integrated Development Environment (IDE). Although you do not have Python installed yet, it is important to also download an IDE, which we will not be covering in this guide.

Recommended IDEs

- Thonny (Best for beginners)
- Visual Studio Code (Best for users with previous experience)
- PyCharm (Easy to use for larger projects)

Vim Guide

We will use the vim editor in step 2. This step will be optional and should only be used for those confident in their abilities to navigate the terminal cautiously. To understand why, please read the *warning* on step 2.

Introduction

Glossary

INDEX OF TERMS AND THEIR DEFINITIONS USED WITHIN THIS DOCUMENT.

Class – Template of code used to create an OBJECT.

cd – TERMINAL command to change directory.

Function – Block of code that runs when called.

IDE – Integrated Development Environment. Application that allows users to write and compile code in one place, without having to navigate the terminal each time they want to run their code.

Library – Collection of code with a pre-determined use. Contains FUNCTIONS, CLASSES, and VARIABLES specific to that library.

Object – Type of VARIABLE that contains data and FUNCTIONS that manipulate its given data.

Path Variables – Type of VARIABLE used in the TERMINAL to allow the computer to easily find a file's location.

Pip – Package manager for Python modules. It allows users to import LIBRARIES of code to implement in their projects.

Pip3 – Same as pip, direct linking between pip and pip3.

Python3 – An interpreted, object-oriented, high-level programming language widely used for complex computation and composite data structures.

Terminal – Text-based user interface, allowing users to interact directly with computer's files and operating system.

Variable – Container for holding data.

Vim – Text-editor used within the terminal to write/create files without leaving the TERMINAL.

Instructions

Step-By-Step Instructions to Set Up Python3

1. Installing Python3

1.1 Open the terminal by pressing: CMD + SHIFT + SPACE and typing *terminal* into the search bar.

1.2 Once the terminal is opened, type the following command to check if Python3 is already installed: *python3 --version*

```
[andrewkelton@Andrews-MacBook-Pro-2 ~ % python3 --version
Python 3.12.4
```

Figure 1.1 1 Example of successful download of python

If you see a version number, like in figure 1.1, Python3 is installed, and you may skip to step 2. If not, continue to *Python3 Official Download*.

WARNING

DO NOT DOWNLOAD ANY VERSION OF PYTHON THAT DOES NOT COME DIRECTLY FROM PYTHON.ORG. ANY OTHER VERSION OF PYTHON IS CONSIDERED MALWARE AND WILL DESTROY YOUR COMPUTER. PROVIDED BELOW IS THE VERIFIED PYTHON.ORG LINK TO DOWNLOAD PYTHON3.

Python3 Official Download

1.1 Go to the official Python website: https://www.python.org/downloads/

1.2 Download the latest Python3 Installer for Mac.

1.3 Run the installer and follow the on-screen instructions.

1.4 After installation, verify python was installed by referring to *Figure 1.1 1*.

1.5 Python3 should now be installed. If there are still issues, refer to Common Issues.

Instructions

2. Setting Up the Terminal for Python3

WARNING *Proceed with this step at your own risk.*

EDITING YOUR PATH MAY CAUSE PIP TO UNLINK FROM PYTHON IF DONE INCORRECTLY. IF YOU DELETE DATA CURRENTLY IN YOUR .ZSHRC FILE, YOU MAY DAMAGE FILE PATHS SET FOR YOU BY YOUR COMPUTER. THE DELETED CONTENT CANNOT BE RESTORED AND MAY CAUSE DAMAGE TO YOUR COMPUTER'S FILES. SKIP THIS STEP IF YOU ARE NOT COMFORTABLE WITH THESE RISKS. PROCEED WITH YOUR OWN CAUTION.

2.1 Opening and Using the Vim Editor to Adjust Your Path

2.1.a Open the Terminal and enter the following to open the vim editor:

vim ~/.zshrc

2.1.b Press *i* on your keyboard to confirm you are writing to the file.

2.1.c Enter the following into the opened file:

export PATH=" /usr/local/bin/python3:$PATH"

```
export PATH="/Library/Frameworks/Python.framework/Versions/3.12/bin:${PATH}"
export SSL_CERT_FILE=/Library/Frameworks/Python.framework/Versions/3.12/etc/openssl/cert.pem
export SSL_CERT_DIR=/Library/Frameworks/Python.framework/Versions/3.12/etc/openssl/certs
```

Figure 2.1 1 Example of .zshrc file opened in vim editor

2.2 Exiting the Vim Editor

2.2.a Press the *esc* key to exit the writing mode.

2.2.b Type *:wq!* and hit *return* on your keyboard.

Your PATH variables are now set, and your environment will not have any trouble locating the Python3 installation.

You may now close the terminal window. If you are troubleshooting, restart your text editor as well, so the changes may be updated.

Instructions

3. Writing your First Python Program

If you have not downloaded a text editor, as described in the *Quick Start Guide*, please review the guide and download a text editor from the *recommended list* (pg. 3). Once you have downloaded an editor, you may continue with step *3.1*.

In this example, I will be using Visual Studio Code. If you are not using Visual Studio Code, follow your text editor's guide to creating and writing files.

3.1 Open Your Text Editor

3.2 Create a New File

3.2.a Click *New File…* as seen in Figure 3.2 1

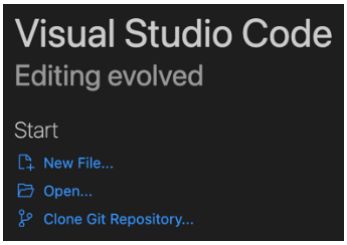

Figure 3.2 1 Visual Studio Code welcome screen

3.2.b A search bar will appear, and you will type:

Python into the search bar. (Figure 3.2 2)

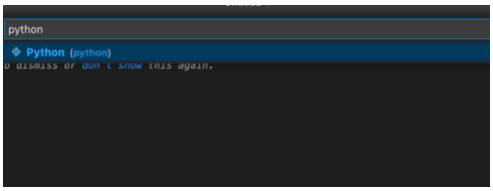

Figure 3.2 2 Visual Studio Code search bar

3.2.c Hit *return* and a new Python file will be created.

3.3 Open your new file if not already opened and type the following into the file `print("Hello, Python!")`

3.4 Save the file as ***hello.py*** into your ***Desktop*** folder.

Instructions

4. Executing Your Python File

4.1 Navigate to and open the terminal.

4.2 Type *cd* and hit *return*, then type *cd Desktop* and hit *return* once more.

4.3 Type *python3 hello.py* and hit *return* to execute your Python file's code.

```
andrewkelton@Andrews-MacBook-Pro-2 ~ % cd
andrewkelton@Andrews-MacBook-Pro-2 ~ % cd Desktop
andrewkelton@Andrews-MacBook-Pro-2 Desktop % python3 hello.py
Hello, Python!
```

Figure 4 1 Example of successful Python3 file execution

If your output differs from the output above (Figure 4 1), review where your file is saved and move it to your ***Desktop.*** If any other error occurs, please refer to *Common Issues and Troubleshooting* (pg. 10).

Instructions

5. Using Python Packages

5.1 Installing Packages with Pip

5.1.a . To install a package, use the following command in your terminal:

pip3 install <package-name>

Note, <package-name> is just a reference point.

For example, to install NumPy, use *pip3 install NumPy*

5.2 Using Packages in Your File

Along with installing packages with pip, Python has preinstalled packages you do not need to install with pip. Take a look at some of the packages listed here https://docs.python.org/3/library/index.html and pick a package to import.

5.2.a In your opened file type: *import <package-name>*
at the top of your file

Note, <package-name> is just a reference point.

For example, to install **math**, use *import math*

5.2.b After choosing and importing a package into your file, choose a variable described in the package to print. I chose to use the math library and printed the value of π (see figures 5.2 1 and 5.2 2 below).

5.2.c Run your file using the method described in step 4.

If you encounter any errors, such as *squiggly lines*, under your imported package name, or package not found errors, please refer to *Common Issues and Troubleshooting* (pg. 10).

```
import math

print(math.pi)
```

Figure 5.2 1

```
andrewkelton@Andrews-MacBook-Pro-2 Assignment2 % python3 test.py
3.141592653589793
andrewkelton@Andrews-MacBook-Pro-2 Assignment2 %
```

Figure 5.2 2 Example of output from imported code file

Troubleshooting Guide

Common Issues and Troubleshooting

Before troubleshooting, ensure the following:

- ***Python3 is downloaded.***
- ***You are in the correct directory/folder.***
- ***You did not misspell any files or variables.***

Problem	Cause	Correction
Python3 is not running when executing Python code	1. Python3 was not installed in the correct location. 2. Pip3 was not installed in the correct location.	Delete the Python3 download from your computer and reinstall.
`Module Not Found` error when importing Python3 package	1. Misspelling when importing package. 2. Package was not downloaded. 3. Pip3 is not linked properly.	1. Ensure the package name is spelled correctly. 2. Ensure you downloaded the package you are importing. 3. Refer to step 3 (pg. 6)
`Permission denied` error in terminal when running Python code	The permission has not been set and is automatically denied.	In the terminal, type: `chmod +x name.py` (`name` being the name of your file)

Conclusion

By following these instructions, you now have a working Python3 development environment on your Mac. You can now start writing Python programs and installing necessary packages using the terminal. To learn more with Python, look up projects to build on your own.

References

Inspired by the following resources:

- Python Official Documentation
- YouTube Video: Python Installation