Appendix B – Running Programs

Support the Author: Buy the book on Amazon or
the book/ebook bundle directly from No Starch Press.

Read the author's other free Python books:

Appendix B. Running Programs

Running Programs and Command Line Arguments

If you have a program open in IDLE's file editor, running it is a simple matter of pressing F5 or selecting the Run ▸ Run Module menu item. This is an easy way to run programs while writing them, but opening IDLE to run your finished programs can be a burden. There are more convenient ways to execute Python scripts.

Shebang Line

The first line of all your Python programs should be a *shebang* line, which tells your computer that you want Python to execute this program. The shebang line begins with `#!`, but the rest depends on your operating system.

- On Windows, the shebang line is `#! python3`.

- On OS X, the shebang line is `#! /usr/bin/env python3`.

- On Linux, the shebang line is `#! /usr/bin/python3`.

You will be able to run Python scripts from IDLE without the shebang line, but the line is needed to run them from the command line.

Running Python Programs on Windows

On Windows, the Python 3.4 interpreter is located at *C:\Python34\python.exe*. Alternatively, the convenient *py.exe* program will read the shebang line at the top of the *.py* file's source code and run the appropriate version of Python for that script. The *py.exe* program will make sure to run the Python program with the correct version of Python if multiple versions are installed on your computer. Use the *-3* command line argument to make *py.exe* run the latest Python 3 version installed. Use the `cd` command to change to the directory (that is, folder) that contains your Python script file. Then use the *py* program to run your script:

```
C:\Users\Al>cd c:\folderwithyourscript
C:\folderwithyourscript>py -3 yourscript.py
```

To make it convenient to run your Python program, create a *.bat batch file* for running the Python program with *py.exe*. To make a batch file, make a new text file containing a single line like the following:

```
@py.exe C:\path\to\your\pythonScript.py %*
```

Replace this path with the absolute path to your own program, and save this file with a *.bat* file extension (for example, *pythonScript.bat*). This batch file will keep you from having to type the full absolute path for the Python program every time you want to run it. I recommend you place all your batch and *.py* files in a single folder, such as *C:\MyPythonScripts* or *C:\Users\YourName\PythonScripts*.

The *C:\MyPythonScripts* folder should be added to the system path on Windows so that you can run the batch files in it from the Run dialog. To do this, modify the PATH environment variable. Click the **Start** button and type **Edit environment variables for your account**. This option should auto-complete after you've begun to type it. The Environment Variables window that appears will look like Figure B-1.

From System variables, select the Path variable and click **Edit**. In the Value text field, append a semicolon, type **C:\MyPythonScripts**, and then click **OK**. Now you can run any Python script in the *C:\MyPythonScripts* folder by simply pressing WIN-R and entering the script's name. Running pythonScript, for instance, will run *pythonScript.bat*, which in turn will save you from having to run the whole command `py.exe C:\ MyPythonScripts\pythonScript.py` from the Run dialog.
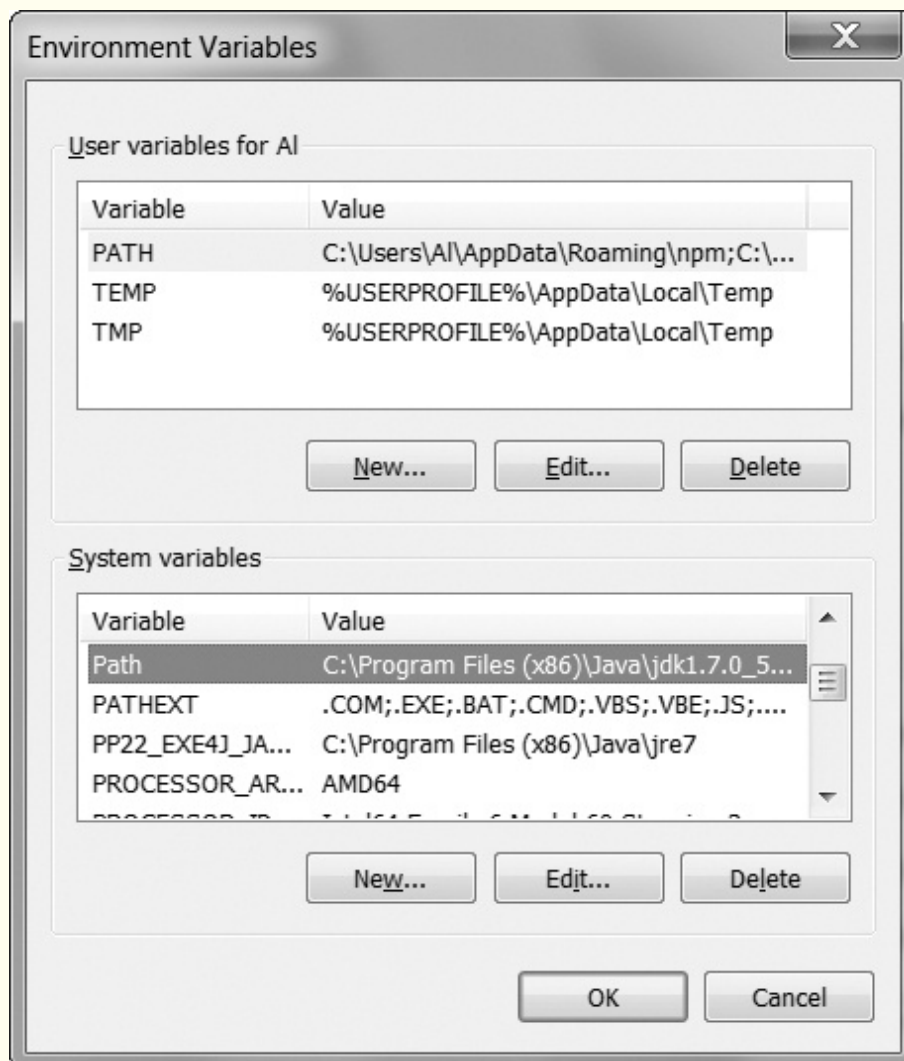

Figure B-1. The Environment Variables window on Windows

Running Python Programs on OS X and Linux

On OS X, selecting Applications ‣ Utilities ‣ Terminal will bring up a *Terminal* window. A Terminal window is a way to enter commands on your computer using only text, rather than clicking through a graphic interface. To bring up the Terminal window on Ubuntu Linux, press the WIN (or SUPER) key to bring up Dash and type in **Terminal**.

The Terminal window will begin in the home folder of your user account. If my username is *asweigart*, the home folder will be */Users/asweigart* on OS X and */home/asweigart* on Linux. The tilde (~) character is a shortcut for your home folder, so you can enter `cd ~` to change to your home folder. You can also use the `cd` command to change the current working directory to any other directory. On both OS X and Linux, the `pwd` command will print the current working directory.

To run your Python programs, save your *.py* file to your home folder. Then, change the *.py* file's permissions to make it executable by running `chmod +x pythonScript.py`. File permissions are beyond the scope of this book, but you will need to run this command on your Python file if you want to run the program from the Terminal window. Once you do so, you will be able to run your script whenever you want by opening a Terminal window and entering `./pythonScript.py`. The shebang line at the top of the script will tell the operating system where to locate the Python interpreter.

Support the author by purchasing the print/ebook bundle from No Starch Press or separately on Amazon.

Read the author's other Creative Commons licensed Python books.

CODING WITH MINECRAFT
BUILD TALLER, FARM FASTER, MINE DEEPER, AND AUTOMATE THE BORING STUFF
AL SWEIGART

CRACKING CODES WITH PYTHON
AN INTRODUCTION TO BUILDING AND BREAKING CIPHERS
AL SWEIGART

4TH EDITION
INVENT YOUR OWN COMPUTER GAMES WITH PYTHON
AL SWEIGART

SCRATCH PROGRAMMING PLAYGROUND
LEARN TO PROGRAM BY MAKING COOL GAMES
AL SWEIGART

Making Games with Python & Pygame
A guide to programming with graphics, animation, and sound
Al Sweigart