

Installing Python and Associated Tools

MSSC 6000 – Scientific Computing

Jay Pantone

January 25, 2022

1 Goals

This guide covers how to do the following tasks on Windows, Mac, and Linux machines:

- installing Python 3
- install “Git for Windows” on Windows machines only
- install Sublime Text (and IDE for writing your programs)
- install “pip”, the package manager for Python
- use a few basic terminal commands
- write and run a basic python program
- use interactive mode

2 Installing Python

Windows: Go to [Python.org](https://python.org) and download the latest version of Python 3. At the moment, this is Python 3.10.2. When installing, check the box that says “add Python 3 to PATH”, and check the later box that says “precompile standard libraries”. All other defaults are okay.

Mac: Install the Homebrew package manager (which is very useful on its own) by going to the website brew.sh and following the instructions on the main page (copy the given code snippet into the Mac terminal, and hit enter). Then run the command `brew install python@3.10`.

Linux: Linux may come with an updated Python already installed, but if not you can follow the Mac directions for basically all parts of this guide.

3 Git for Windows

Git for Windows provides Windows users with a terminal window that behaves mostly like the terminal on Mac and Windows. This is different from Windows's built in command console that runs entirely different commands. To install it, go to <https://gitforwindows.org/> and follow the directions. All default options are okay.

4 Sublime Text

Code can be written in any text editor, even notepad, but to make your life easier, you should use an integrated development environment (IDE). This is a fancy text editor designed for code. They range from having just very simple features (like syntax highlighting to make your code different colors and easier to read) to very advanced. A good IDE for beginners is Sublime Text. More advanced ones include VS Code and Atom.

To install Sublime Text on Windows, go to <https://www.sublimetext.com/>. On Mac or Linux, you can do the same thing, or you can install via Homebrew (if installed in Step 1) with the command `brew cask install sublime-text`.

5 Pip

PIP stands for “(P)ip (I)nstalls (P)ackages”. It's a package manager for the many packages that can be installed to add features to your Python code. Pip usually comes pre-installed when you install Python, but if not you can run it with the command `python3.10 -m ensurepip --upgrade`, substituting `python3.10` for whatever your version is, if necessary.

Suppose you want to install the “requests” packages, which adds commands to python to fetch data from the internet. There are two versions to run any pip command. The short but less reliable version is `pip install requests`. Since machines sometimes have many versions of Python installed, particularly Macs, this might run the “pip” for a different older version. To make sure you're running the right version, you can run `python3.10 -m pip install requests`.

On Mac and Linux, the commands mentioned above can just be run in the terminal. On Windows, you will want to run them in the terminal provided when you open Git for Windows (sometimes the program itself is called Git Bash). You may need to substitute “py” instead of “python3.10” in Git for Windows. Let me know if you have any trouble with this.

Python users with more experience, especially with the command line, may be interested in reading about “virtual environments”.

6 Basic Terminal Commands

We will talk more about the terminal in later lectures. For now it suffices to know that when the terminal is open, it is always “in” some folder in your computers file structure (e.g., your user folder, your Desktop folder, your Downloads folder, etc).

To know which folder your terminal is currently in, run the command `pwd` (stands for “present working directory”) and it will print the result.

To see all of the files and folders in that directory, run the command `ls`. To go into a folder, type `cd [name of folder]`. To go up a folder, type `cd ..`, because “..” always means “one folder above where I currently am”.

7 Running a Python program

Most python programs will be written in a file and saved to your hard drive, just like you might write a Word document and save it. A program consists of your list of commands, saved to a file with the extension “.py”. (An IDE will then automatically recognize that this is a Python file and help you out with syntax highlighting and other features accordingly.)

Start by writing a file called “first_program.py” with a single line saying

```
print("hello world")
```

and saving it.

To run a python program you need to go into the terminal (Mac/Linux) or Git for Windows (Windows) and call the correct python command with the file name of the program as input. For example, if my file is located at the location `/Users/Jay/Desktop/first_program.py`, the command on Mac/Linux to run it will be

```
python3.10 /Users/Jay/Desktop/first_program.py.
```

On Windows, you may need to change “python3.10” to “python” or “py” depending on your installation. Please let me know if you get stuck.

Helpful Tip: On all systems, you can drag and drop a file onto the terminal window, and it will paste in the full file location for you!

8 Interactive Mode

When you’re writing Python code in a file, none of the lines of code are actually executed until you run the file using the terminal. There is a different mode, called “interactive mode” where each line you write is executed right away. This is great for testing things out and doing quick computations, but not great for writing programs you plan to reuse. Once you close the window, anything you wrote is gone.

To activate interactive mode on Mac or Linux, simply run the command `python3.10` without giving it a file name, and then you will be dropped into python interactive mode. On

Windows, open the Git for Windows terminal and try the commands `wipty py` or `wipty python`. To leave interactive mode, use the command `exit()` or hit `Control+d`.

You can also mix these modes together and run a file, and then when it's completed, be dropped into interactive mode, by using the `-i` flag, as in: `python3.10 -i file_name.py`.

Lastly, there is a python tool called IPython that you might be interested in that gives you a nicer version of interactive mode, with tools like easier indentation and nicer copy+pasting.