Installing & Setting Up Python Tools

ASTR 211, Spring 2021

Mason V. Tea, 1/27/2021

We'll be learning to use a handful of different tools over the course of the semester, including:

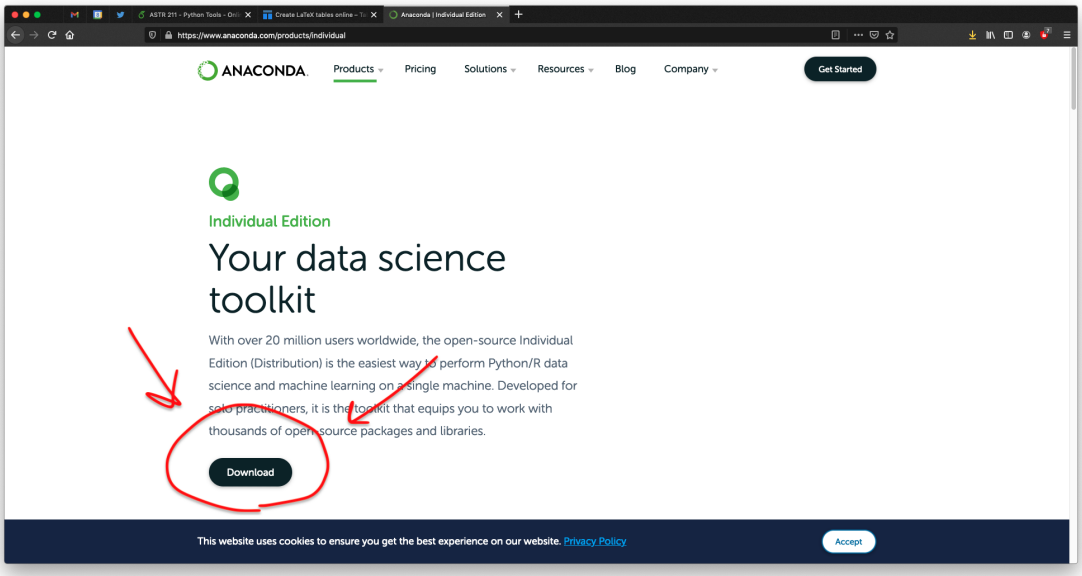
- **Atom**: A text editor with syntax highlighting (color-coding for code) and other plugins that make coding easier.
- **Anaconda**: A package of Python libraries and tools, including `numpy`, `matplotlib`, `astropy`, and others we'll use throughout the course.
- **Jupyter Notebook**: A web-based Python interface, which let's you write and execute code and take notes in discrete "cells".
- **Python**: Python 3, to be exact.

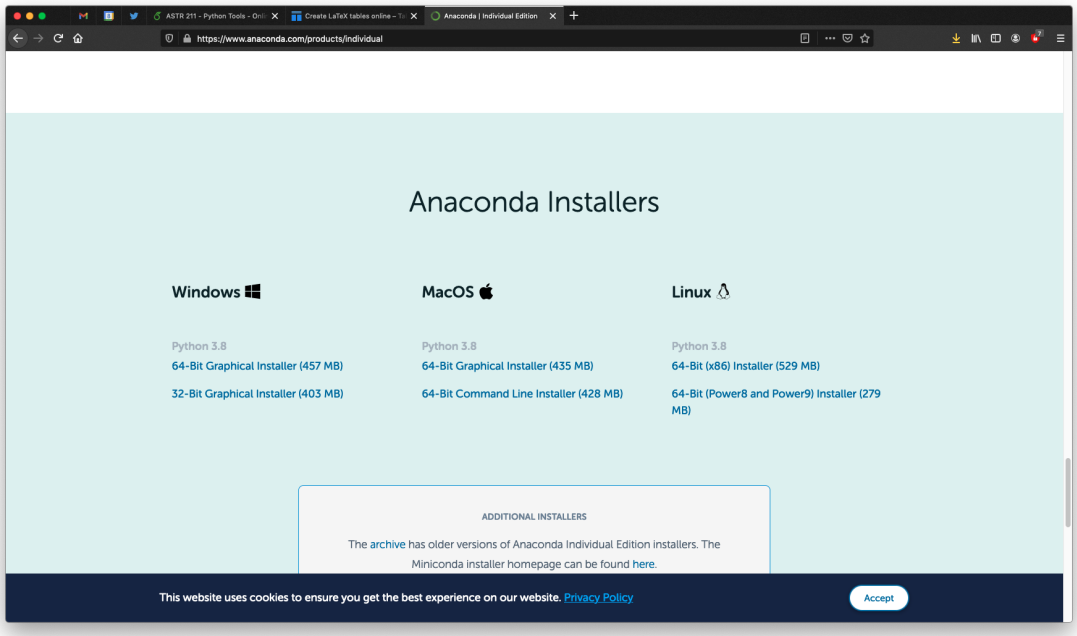
We aren't going to be using these all at once, but everyone should go ahead and get their accounts & installations out of the way in one fell swoop. We'll go through each one in detail below. If you're having trouble getting any/all of these things to work, email me (mtea@wes) or Roy Kilgard (rkilgard@wes) with your issue.

Anaconda: Python, Jupyter and Libraries (*All OSs*)

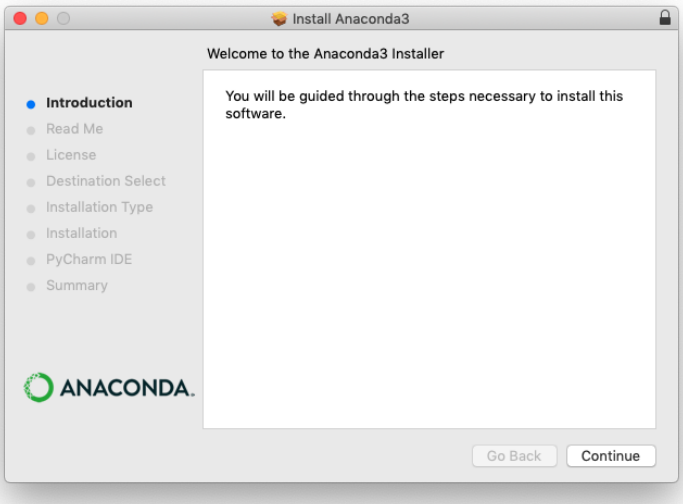
Of course, in order to program with Python, we'll need to install the dependencies for language itself. Luckily, the Anaconda package comes with the latest install of Python (ver. 3.8 at the time of writing), as well as all the libraries we could ever. Plus Jupyter Notebook.

So, Anaconda is what we're going to want to install first. To do so, navigate to the Anaconda Individual Edition page, hit the Download button, and choose the correct installer version for your computer. (I suggest using the graphical installer, rather than command line.)

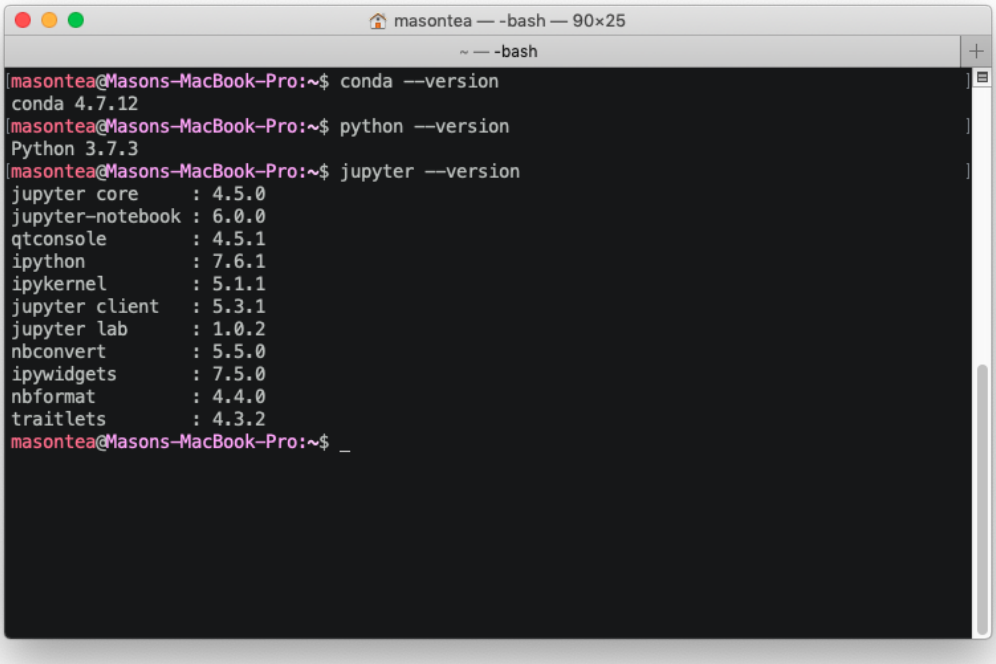




Upon running the file, you'll be greeted with the screen below. Just follow the guided installation using the default settings, and exit the installer when it's finished.

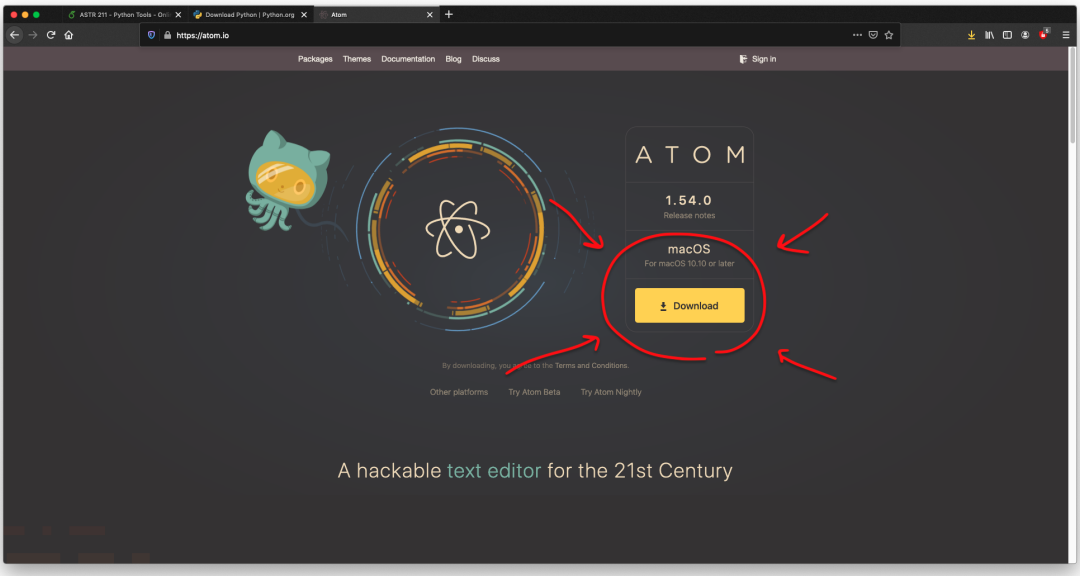


After that, we can use the terminal to check that the Anaconda library package, Python and Jupyter Notebook have been installed properly. To do this, open up the terminal, type `python --version`, and hit enter. You should see `Python 3.x.x` printed on the screen in response. You can do the same for Anaconda with `conda --version` and Jupyter Notebook with `jupyter --version`. As long as you don't get any errors, you should be set.



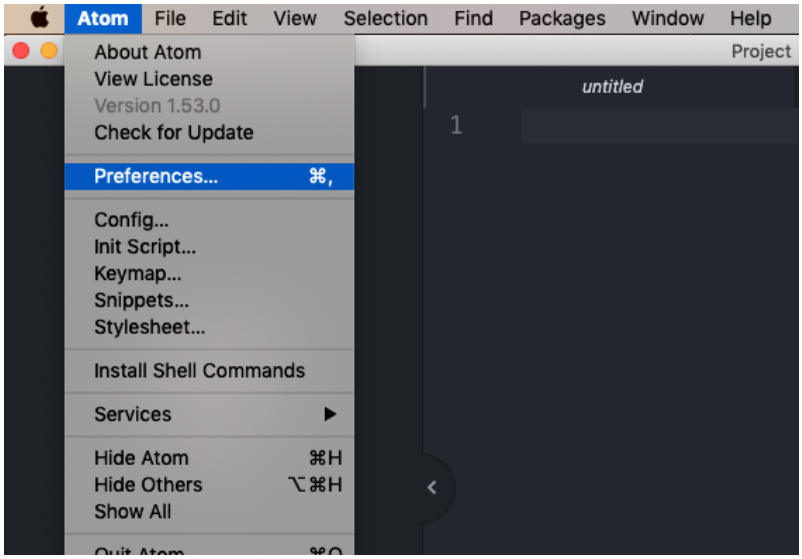
Atom (*Mac*)

Now that you've got Python installed, you're going to want to have somewhere to write code. For Macs, I recommend Atom, a text editor with lots of functionality. To download Atom, just go to `https://atom.io/`, hit the download button, and run the file.

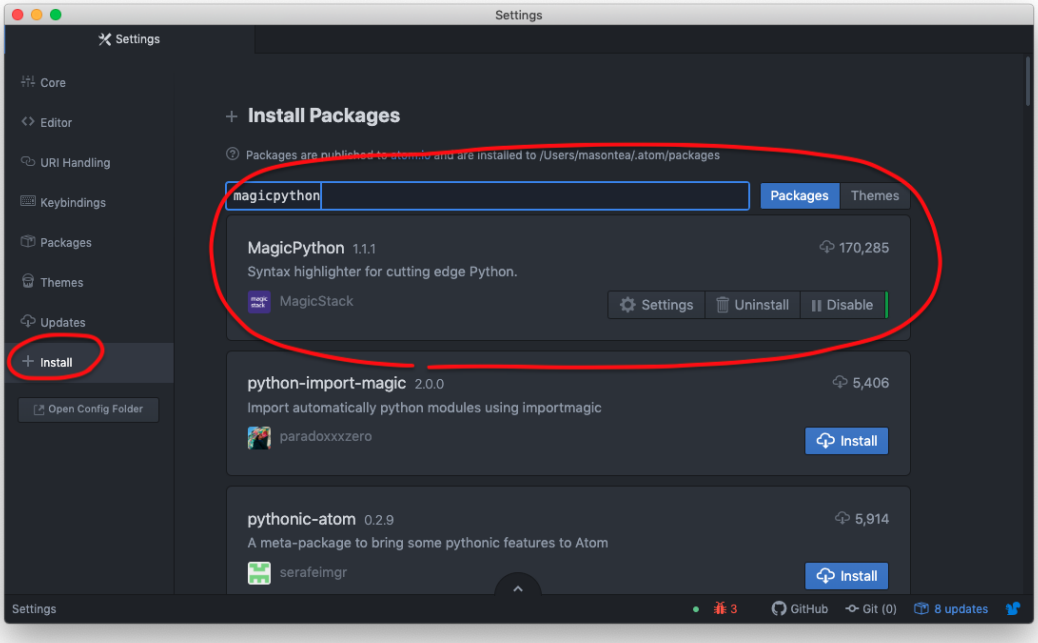


Atom's themes and packages are why I enjoy it so much, but you can use any text editor with syntax highlighting for Python. For those of you that don't know what that means, it's going to be in your best interest to use Atom instead of your favorite text editor.

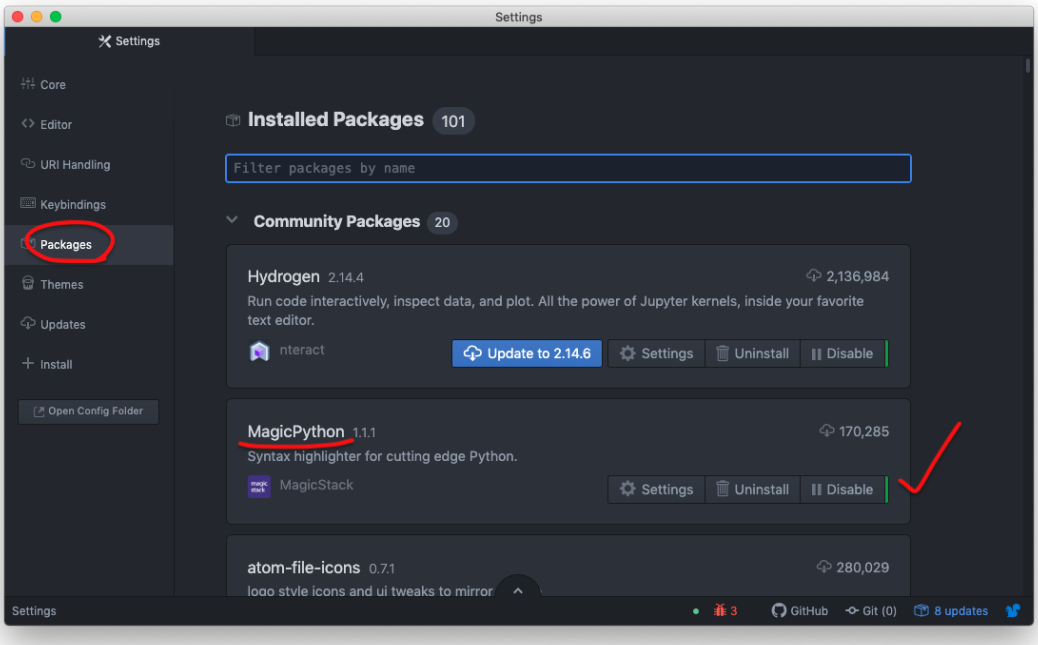
Syntax highlighting color-codes your program for you, which makes it much easier to read and interpret. Atom provides this functionality for different languages through *packages*. The one we want is called magicpython. To install it, open up Atom and hit "Atom" all the way up in the top toolbar, then hit "Preferences".



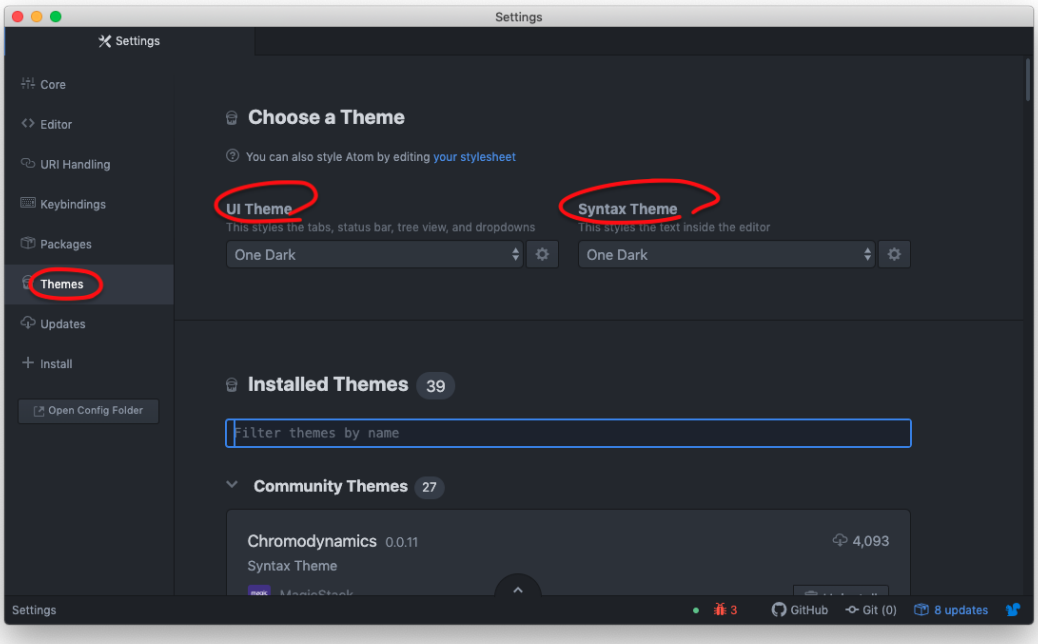
That should take you to this screen. Here, you should hit "Install" in the lefthand sidebar, search "magicpython" (it should be the first option), and hit "Install" once you've found it.



It should be enabled automatically, but to make sure, navigate to the "Packages" tab in the lefthand sidebar and check to see if magicpython has an "Disable" button. If it does, you're good to go. If not, hit "Enable."



Once you've written some code, you can customize the color scheme for your syntax highlighting, as well as the look and feel of the whole Atom app. To do this, go back to "Preferences" and navigate to the "Themes" tab in the lefthand sidebar. Here, you can change the UI (user interface) theme or the Syntax (code highlighting) theme from their respective dropdowns.

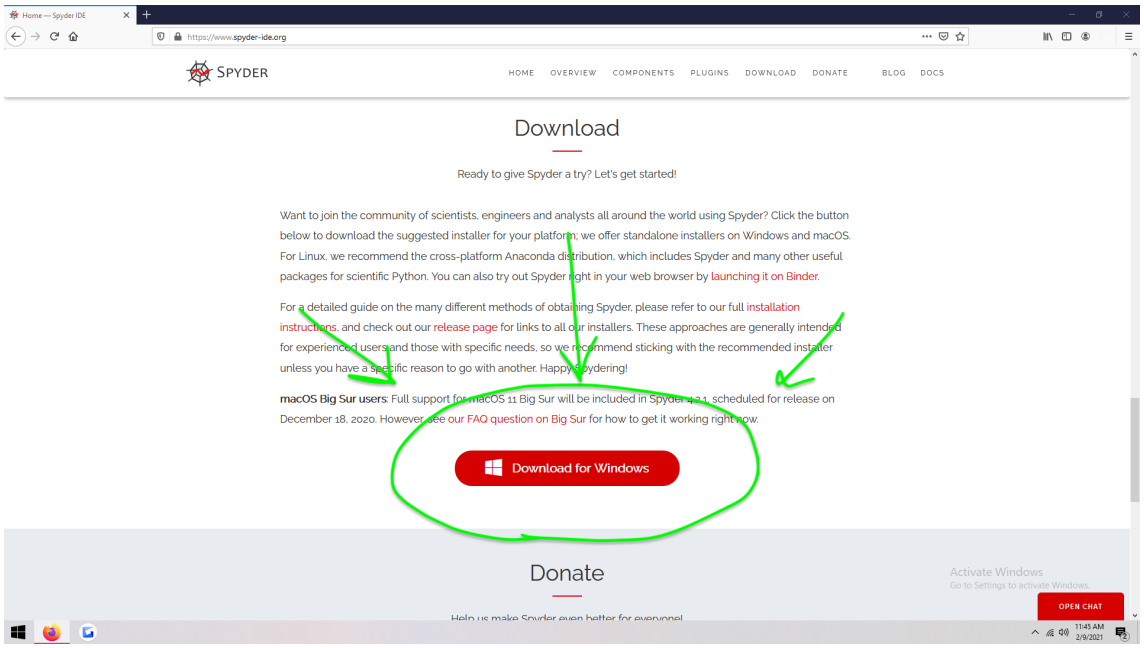


There are lots of neat packages and themes to download that I urge you to explore if you plan on doing a lot of coding in the future, but this is all we need for the purposes of the course. Other than that, it works the same as any other text editor — you can create files, save them, etc. in the toolbar at the top of the screen. We'll go over how to run any Python code you write here later on.

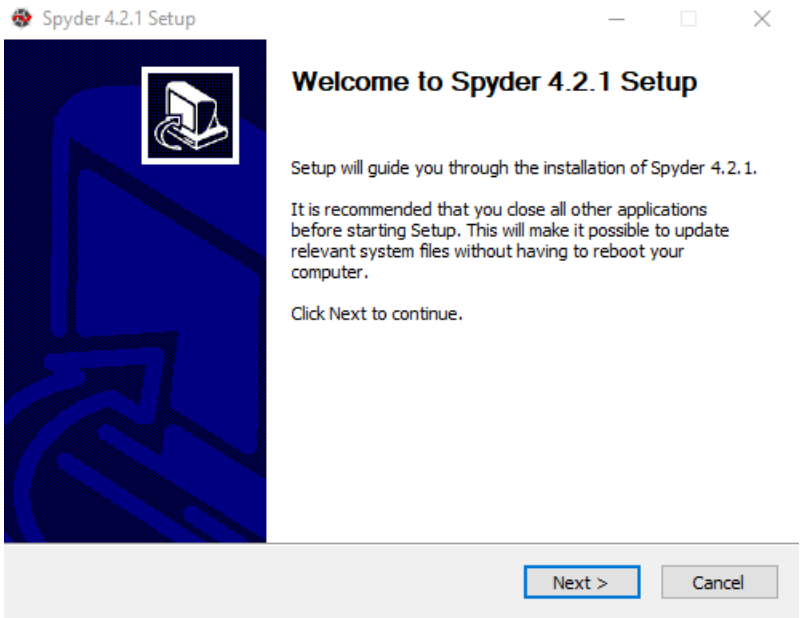
Spyder (*PC*)

While I don't have much experience with running Python on PC, colleagues of mine prefer to use Spyder to both write and run their code. This is opposed to Atom, which requires you to use the command line to run code — the Unix command line that we'll be going over in class and the terminal on PCs are very different beasts.

To install Spyder, navigate to the Download page on their website and download the recommended version of the software.



Once downloaded, double click the file and run the installer. You should see a window similar to this when you do; go ahead and click through the standard settings and let it install.



Once installed, you can go ahead and open up the program, where you should be met with a UI like this, as well as a prompt to take the tour of the program — I suggest that you do so, because I won't be much help at first!