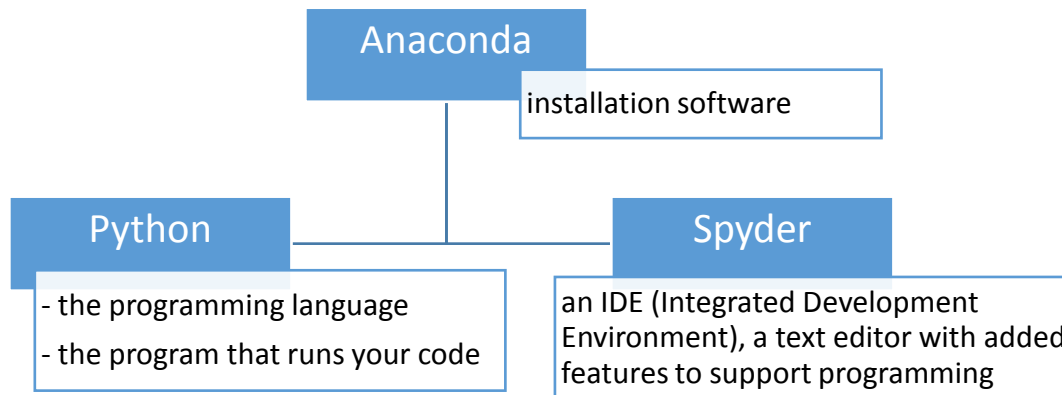# GETTING STARTED WITH PYTHON

Python is a programming language, and you'll be writing your Python code in the programming environment called Spyder. The Anaconda distribution simplifies the installation process by including Python, Spyder, and other packages and tools in one installation file.
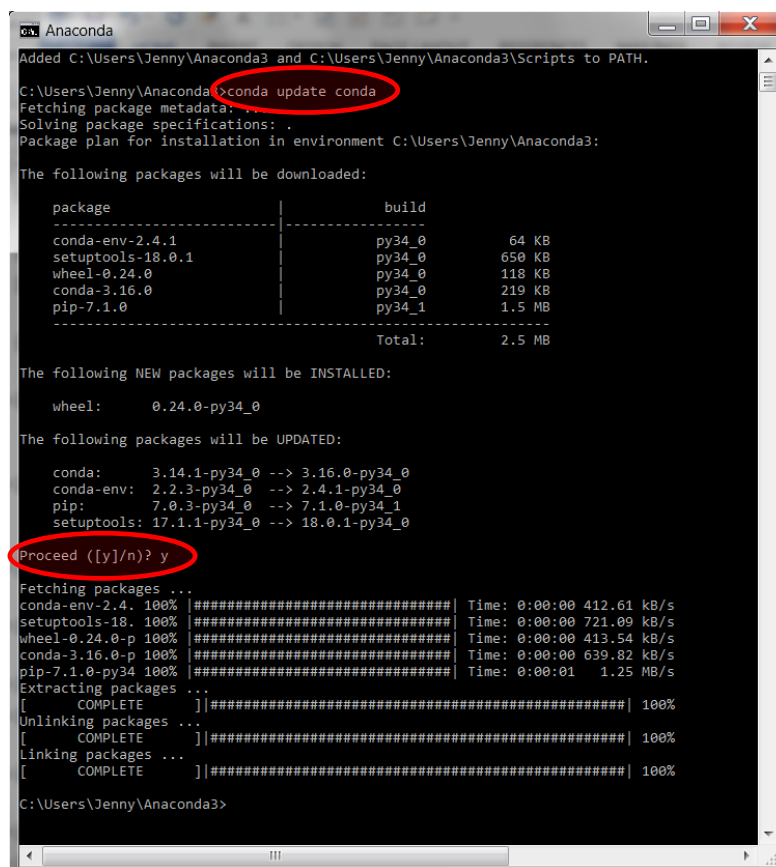
Anaconda

installation software

Python

- the programming language
- the program that runs your code

Spyder

an IDE (Integrated Development Environment), a text editor with added features to support programming

Installing Anaconda

1. Go to http://continuum.io/downloads

2. Scroll down to find your operating system and click on Python 3.5 to download the graphical installer.

    a. For windows, see http://windows.microsoft.com/en-us/windows/32-bit-and-64-bit-windows to find out whether your computer has a 32-bit or 64-bit version of windows. Click on the FAQ "*How can I tell if my computer is running a 32-bit or 64-bit version of windows?*", and follow the instructions to find out. If your computer is running a 32-bit version of Windows, click on "Windows 32-bit Python 3.4 Graphical Installer" under OTHER INSTALLERS, instead of the Windows 64-bit installer, and follow the same directions to install it.
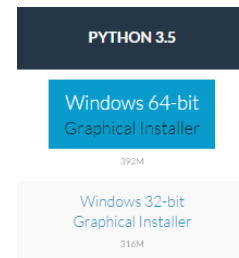
    b. If your mac OS is prior to 10.7, see the FAQ. _____

3. Save the file to your computer.

4. Double click on the downloaded file to open it.

5. Follow the on-screen installation instructions, leaving options as they are currently set. This finishes the installation process.

6. Next, check for any updates using Conda. Conda is one of the extras that is installed through the distribution Anaconda. It handles things like updates, set-up, and package installation through a command line interface. If there are many updates this can take 10 minutes or more. It is rare that you will have to use it.

WINDOWS:
   a. Open Anaconda Command Prompt. Start typing Anaconda Command Prompt into the search box in the start menu, and it will show up.
   b. Type `conda update conda` at the command prompt, typing `y` for Yes and then pressing enter when it asks if you want to proceed. Your installation may identify different packages that need updated.
   c. After that completes, type `conda update anaconda` at the command prompt. If it prompts you to proceed with installation or updating, type `y` for Yes and press enter.
   d. After that, type `conda install seaborn` at the command prompt, then type `y` after the Proceed ([y]/n)? line.
   e. After that completes, you can then close the command prompt window.

MAC:
   a. Open Terminal, which you can find by using Spotlight.
   b. Type `conda update conda` at the command prompt in the terminal, typing `y` for Yes and then pressing enter when it asks if you want to proceed. Your installation may identify different packages that need updated (see Windows image above for the text shown in the terminal window).
   c. After that completes, type `conda update anaconda` at the command prompt. If it prompts you to proceed with installation or updating, type `y` for Yes and press enter.
   d. After that, type `conda install seaborn` at the command prompt, then type `y` after the Proceed ([y]/n)? line.
   e. After that completes, you can then close the command prompt window.
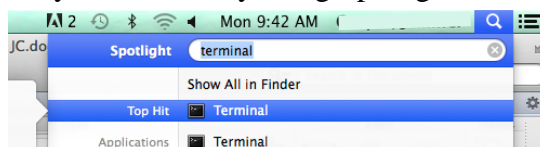
Getting to Know Spyder

You will write your programs and run them inside the Spyder IDE.

Spyder

There are various ways to find and open this program – we only describe one way below.
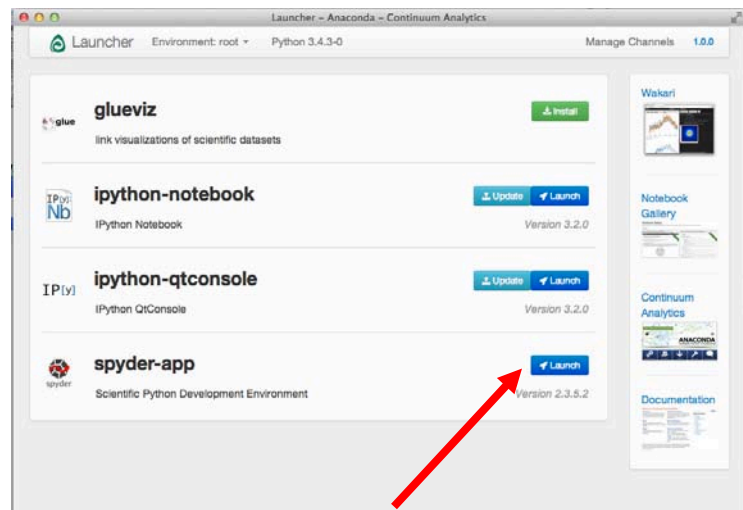
On Windows, in the search box associated with the start menu, type Spyder. Click on the program with its logo. Right clicking on it will give you the option to pin it to your start menu or taskbar for easier access in the future.

On a Mac, use Spotlight to find the program **Launcher**. It has a green tear-drop shaped icon associated with it.
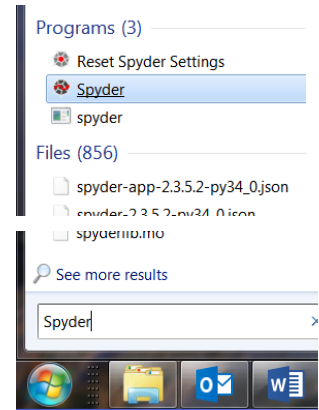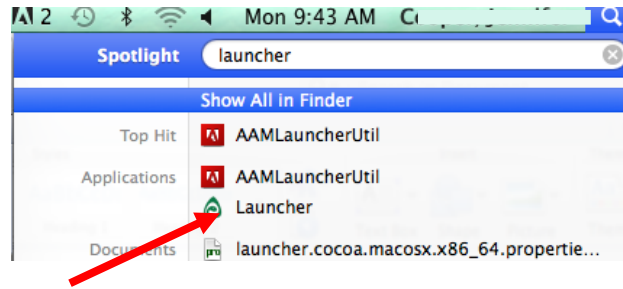
That opens the following window, in which you will click Launch, on the last line for **spyder-app**.

FYI: The icon for Python that goes in the OS X dock, once Spyder has started, is generic, and keeping it on the dock will not work.

After Spyder has launched, you should see an interface that is organized into multiple windows (marked with red rectangles below) each of which has or could have multiple tabs (marked with green rectangles below).

3
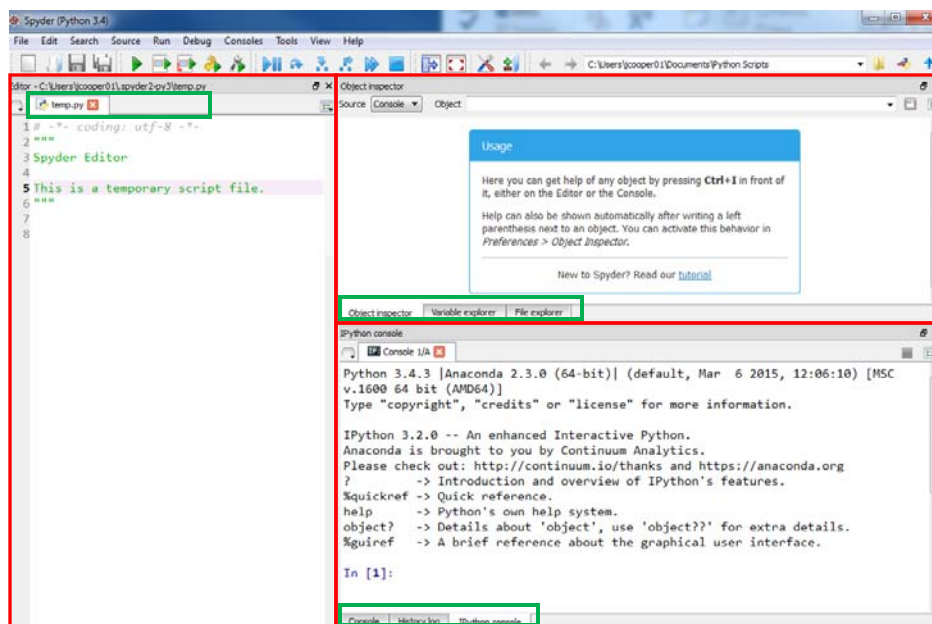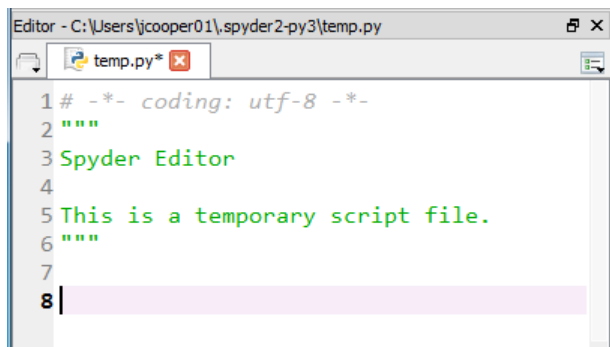
The default starting screen has three windows visible: Editor, Object Inspector, and Console.

If you accidentally close a window or change the layout, you can revert to the default by clicking "View" on the menu bar and then clicking "Reset Window Layout".

- **Editor** – where you can write and save sequences of commands – essentially, where you will write your full program
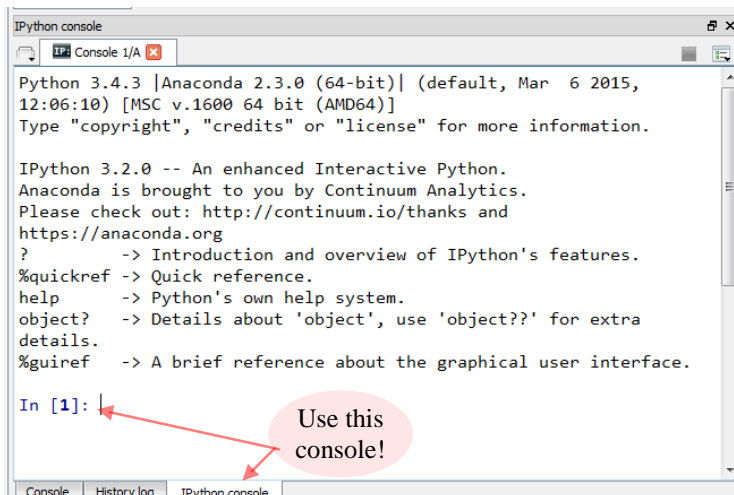

In the sample script that shows up in the editor, anything after # is a comment, meaning that Python will ignore that text. Three double quotation marks are a special type of comment that spans multiple lines.

Here, you would actually begin writing your syntax on line 8 after the comments that say what your file does.

- **Console** – where Python runs your code.

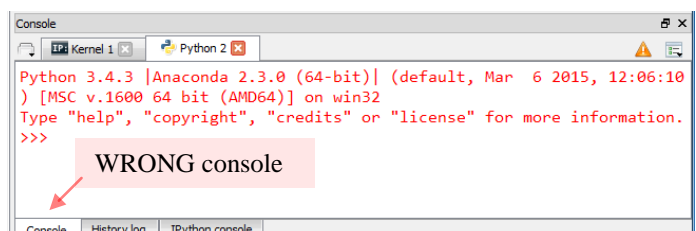Spyder offers two choices for a console – a basic one and an IPython (Interactive Python) console. It is the IPython console we want to use. We will refer to it, however, as just the console.

The prompt (where you can type in a command) for an IPython console is `In [1]:`

*Caution:* If your console looks like the one shown to the right, where the prompt is `>>>` then you are in the *wrong* type of

console and need to click on the tab for the IPython console.

- **Object Inspector –** On starting Spyder, the object inspector (as shown above) just gives you a link to the tutorial (this can be useful if you want to know more details about using Spyder). The rest of the time, this window will continue to give you information.

  The **Object Inspector** gives you more information about a function – it gives you the documentation from the help files and tells you what parameters (or input) the function takes.

  The **Variable Explorer** shows you the value and type of any variables you have created. In the image to the right, we created three variables in the console, and those three variables showed up in the Variable Explorer. This is useful when you are trying to debug your code.
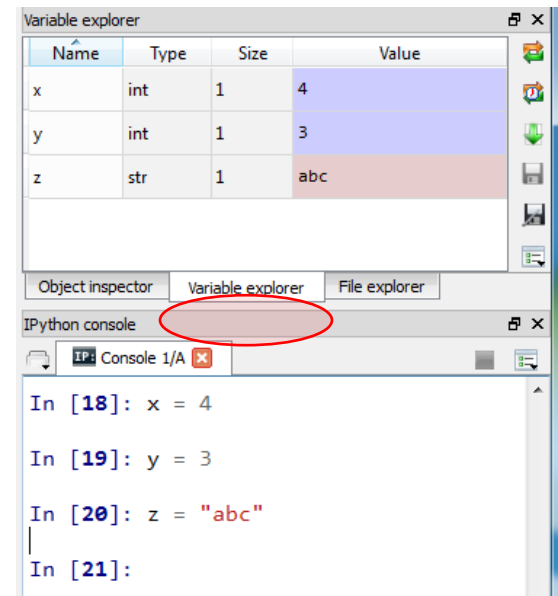
Frequently Asked Questions about Installation

- I already use Python 2.7 or I would rather use Python 2.7. Will that work for this course or do I need to install Python 3.x?
  - o This specialization will use Python 3.x, and we will cover only the syntax that is relevant to Python 3.x. You may use Python 2.7 if you prefer, but you will encounter some differences in the syntax between Python 2.7 and Python 3.x. For example, in Python 3.x you must put parentheses around the objects you would like to display, but you do not have to in 2.7. So, in python 3.x you would have to use the code *print (object)*, but in 2.7, you have the option of whether or not to put parentheses around the object. Often times the error messages that you get are informative, but sometimes they are not. If you choose to use Python 2.7, you are likely to find support for differences you encounter on the discussion board from other learners using Python 2.7. Various websites, including https://wiki.python.org/moin/Python2orPython3, outline some of the syntactic differences.

- I have a different IDE that I like to use. Will that work for this course?
  - o One of the benefits of Anaconda is that it pre-installs the packages we will primarily use in this course. We will offer support for the Spyder IDE in particular. If you are comfortable writing code, running it interactively and as a program, and debugging along with installing packages, you may find that you can continue to use your current IDE.

Where Do I Find the Data Sets?

To download the .csv data files for this course, go to Course Code Books and Data document (in Week 1 materials) and click on the appropriate link. This will give you the option to download the data to your own computer