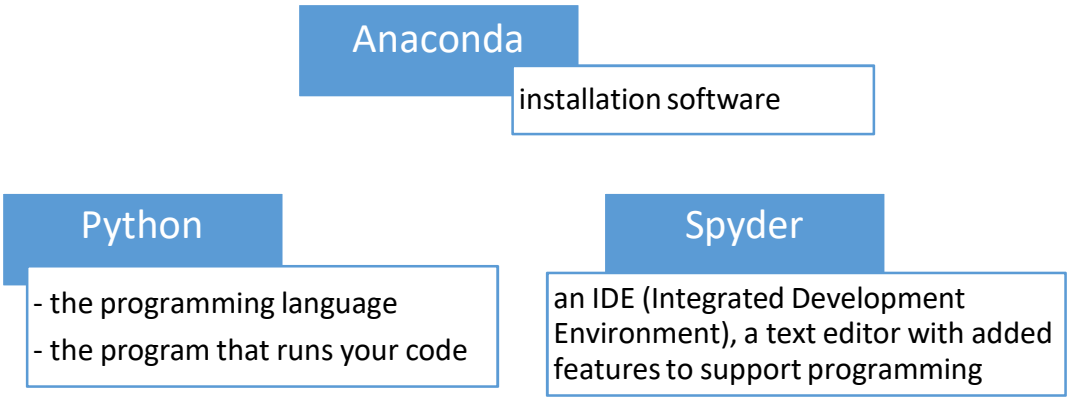
GETTING STARTED WITH PYTHON USING LINUX

August 31, 2018

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.



Installing Anaconda

1. Go to https://www.anaconda.com/download/#linux
2. Choose Download for Linux and click on Python 3.x (3.6 is latest as of this writing) to download the graphical installer. If you have a 64-bit computer use that installation. If not, use the 32-bit install. Do NOT download 2.7. This is a course on Python 3.

   For Ubuntu 16.04 LTS, one of many versions of Linux, you can determine whether you have a 64-bit or 32-bit machine by clicking on the sprocket wheel (cog wheel) and choosing Details under System. It will tell you your OS type. Mine says 64-bit and yours will probably be a 64-bit OS as well.
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. Once installation is complete, you can open a terminal and type in the command spyder and the Spyder integrated development environment will start up.
7. (OPTIONAL) 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. To run this open a terminal window and run the command:

   conda update conda

Then run

conda update anaconda

Alternatively you can do this from within Spyder:

From the Tools menu, choose ‘Open a terminal’ or ‘Open a command prompt’. Run in that terminal the same commands as above.

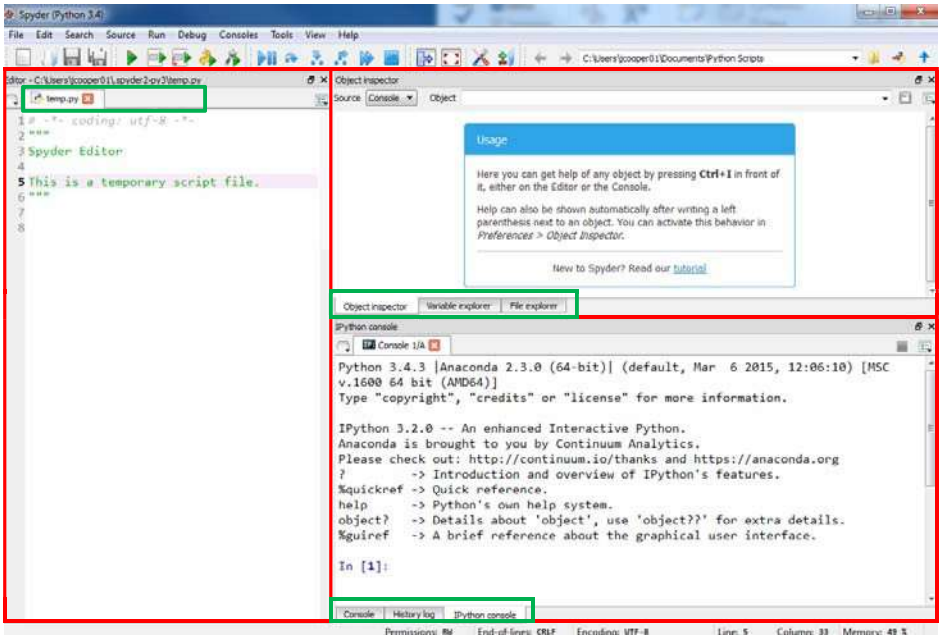
After running these commands, you may close the terminal.

Getting to Know Spyder

You will write your programs and run them inside the Spyder IDE whose icon is 

The simplest way to start spyder is to open a terminal and run the command spyder whereupon it will start up the Spyder Integrated Development Environment (IDE). Spyder is a word made from **S**cientific **Py**thon **D**evelopment **E**nvironment. We are using it here because it has very convenient features.

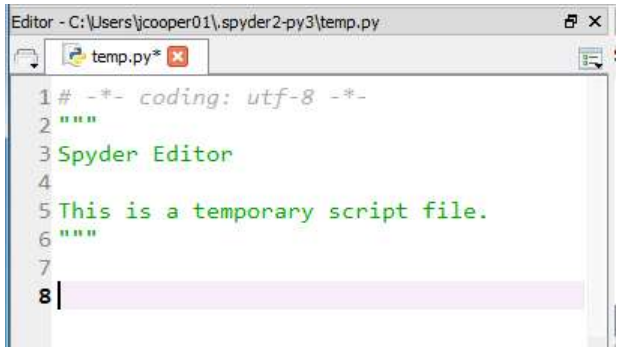
After Spyder has launched, you should see an interface that is organized into multiple window panes (marked with red rectangles) each of which has or could have multiple tabs (marked with green rectangles).



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".

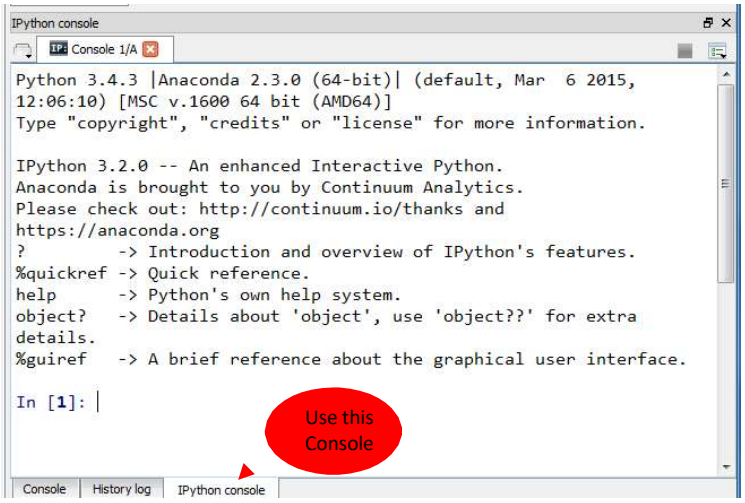
- **Editor** – where you can write and save sequences of commands – essentially, where you will write your full program



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.

- **IPython Console or IP Console** – where Python runs your code.

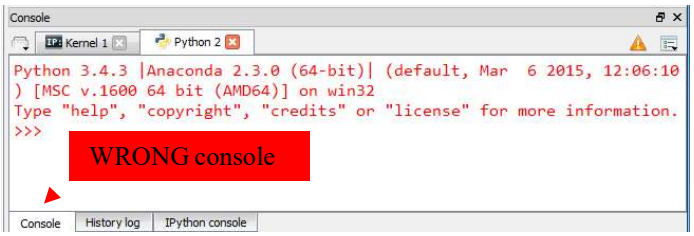


Spyder offers two choices for a console – a basic one and an IPython (Interactive Python) console. It is the IPython console we want to use. We will often refer to it, however, as just the console.

The prompt (where you can type in a command) for an IPython console is

```
In [1]:
```

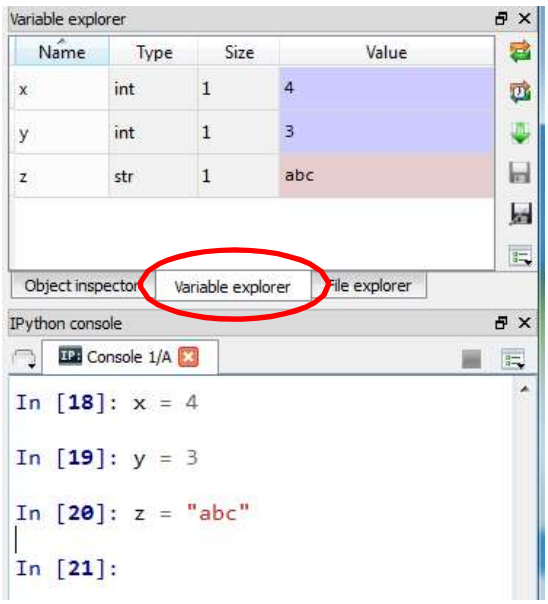
Caution: If your console looks like the one shown to the right, where the prompt is >>> then you are in the *wrong* type of console and need to click on the tab for the IPython console.



- **Object Inspector or Help –** On starting Spyder, the object inspector (in the latest version **Help**) just gives you a link to the tutorial (this can be useful if you want to know more details about using Spyder; it is accessed for the tutorial from the Help menu.). 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?
  - No. Python 2.7 will not work. The programs are auto-graded using a Python 3 system and it will not be able to compile Python 2.7 programs, because Python 2.7 is not compatible with Python 3.x. Yes, this is unusual. But the developers of Python decided to change certain features when going to Python 3 that made it impossible for them to maintain complete compatibility with Python 2. However, should you wish to shift to Python 2, later, you will be able to adapt at the expense of very little time and effort. 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?
  - You could, but it will take more effort than it is worth. One big benefit of the Anaconda installation is that it includes IPython and IPython notebooks. These allow us to keep a whole lesson in one file, separated into cells. We can easily execute the code in a single cell, so that we don't have to write complete programs to run and test our code. After you finish the course, you can change to a different IDE with little effort, however. Most IDEs are similar to Spyder, but don't include the teaching convenience of the cells.