# Getting Started

Python is a general-purpose programming language with a large and growing ecosystem of free, open-source libraries and tools for scientific computing. But before we can start working with Python, we need to get it all set up.

**Here's what we're going to need:**

- **Python 3.5 (or above)**: The core Python interpreter, which actually runs your code.
- **Jupyter Notebook:** An enhanced browser-based interactive Python interface that we'll be using in this course (previously called the IPython Notebook).
- **Pandas:** The Python data analysis library. Useful for working with table-style data.
- **Matplotlib:** Python's main plotting library. We'll use it to make graphs and other visualizations

## Installation

Getting everything up and running can often be the hardest part of getting started with Python.

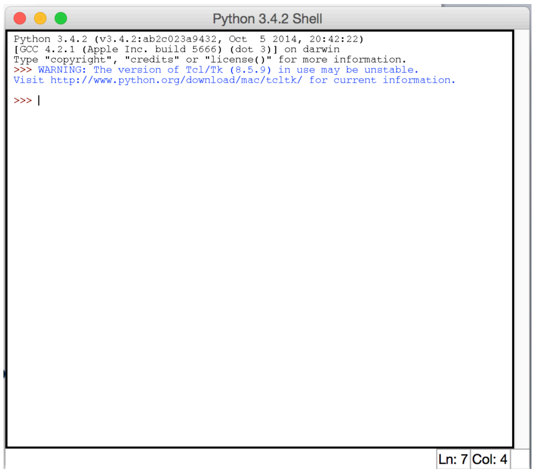
**Easy option:** Download and install one of the prepared scientific Python distribution from the list here: http://www.scipy.org/install.html
I recommend Anaconda (http://continuum.io/downloads), which is rapidly becoming the standard Scientific Python distribution.

**Advanced option:** If you're more comfortable fiddling with your system, I suggest downloading and installing all the packages and dependencies below. (You may need to have a C/C++ compiler set up for this).

- First, download and install the most recent Python **3.x** version from the official Python website: http://www.python.org/download/. Some computers already come with a version of Python installed, but it's still a good idea to ensure that you have the latest official release.
- Next, download and install pip: http://www.pipinstaller.org/en/latest/installing.html
- From the command line, use pip to install Jupyter: *pip install jupyter notebook* or possibly *sudo pip install jupyter notebook*, if you're on a Mac or Linux system and depending on your settings.
- Download and install the following packages:
    - Numpy: http://sourceforge.net/projects/numpy/
    - Matplotlib: http://matplotlib.org/downloads.html
    - Pandas: http://pandas.pydata.org/pandas-docs/stable/install.html

## Launching Python

There are a few ways to start working with Python. The simplest is IDLE, Python's Integrated Development Environment. Once you've installed Python, find the IDLE application and launch it. It should look something like this:

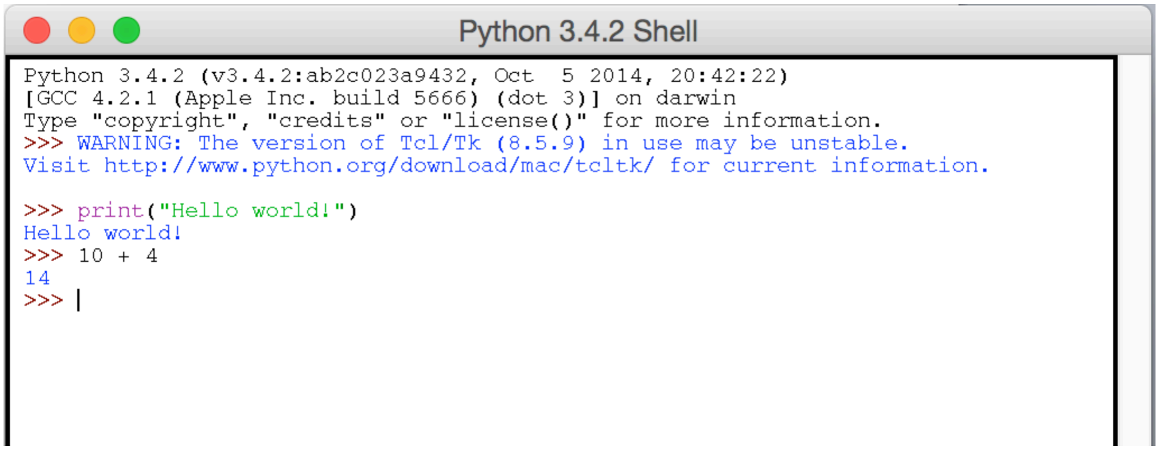


```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct  5 2014, 20:42:22)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

>>> |
```

Now you can enter Python commands, and the interpreter will execute them one at a time. For example, enter the following, and press Enter to execute:

*print("Hello, world!")*

*10 + 4*



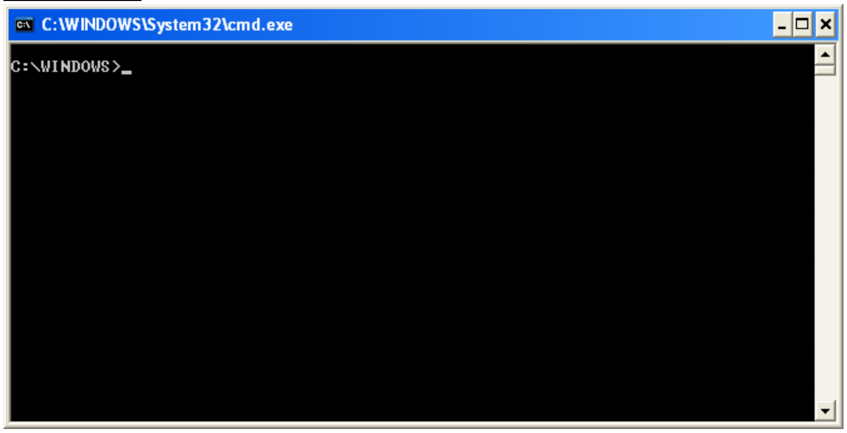
```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct  5 2014, 20:42:22)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

>>> print("Hello world!")
Hello world!
>>> 10 + 4
14
>>> |
```

Go ahead and close the IDLE window; we're going to be using a much more powerful environment – IPython.

There are two ways to run IPython: through a command line or terminal, and in a browser using the IPython notebook. We'll use the Notebook for most of this class, but let's quickly try out the command line version first.

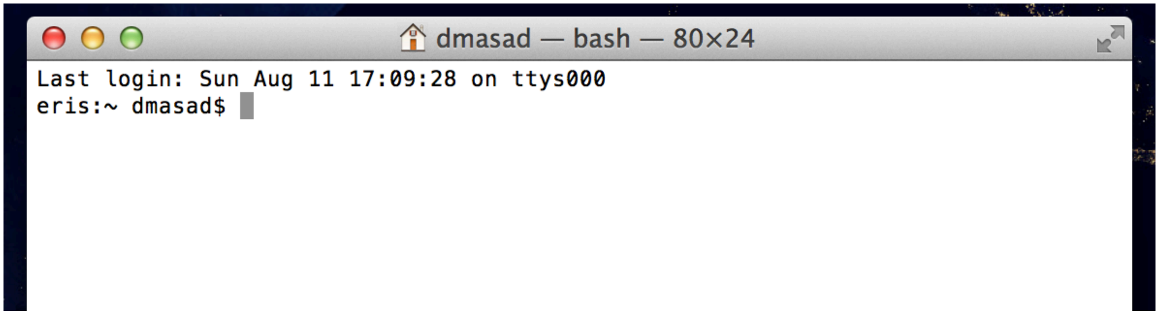
Open a terminal or command line. If you’re on a *Nix system (Mac or Linux) you have a **Terminal** application you can launch. If you’re on a Windows system, you’re looking for a program called **Cmd** or **Command Prompt.** Either way, it should look something like this:

**Windows**



```
C:\WINDOWS>_
```

**Mac**

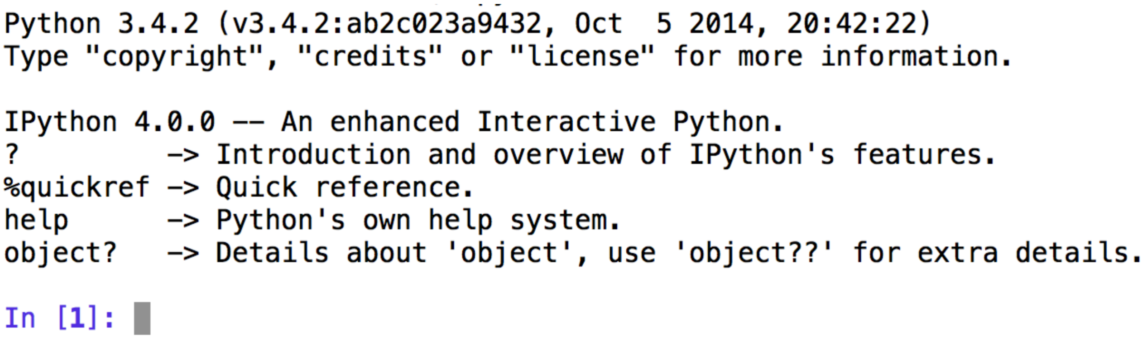


```
Last login: Sun Aug 11 17:09:28 on ttys000
eris:~ dmasad$
```

Type in

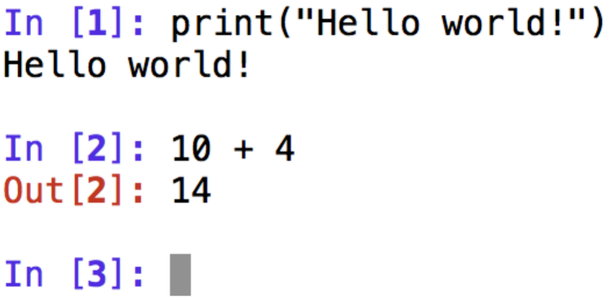
*ipython*

And hit Enter. You should get something like this:



```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct  5 2014, 20:42:22)
Type "copyright", "credits" or "license" for more information.

IPython 4.0.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]:
```

This is a lot like the IDLE interpreter. Enter the same commands from above, and see what happens. You should get pretty similar output, like this:



```
In [1]: print("Hello world!")
Hello world!

In [2]: 10 + 4
Out[2]: 14

In [3]: 
```

Type

*exit*

to quit.

## Launching the Jupyter Notebook

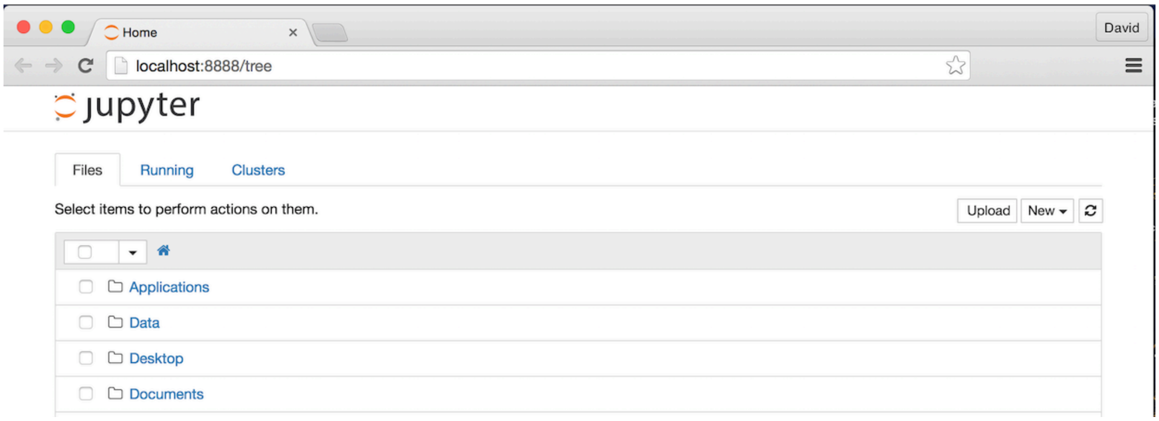
IPython's real strength comes from the Jupyter Notebook, a browser-based interactive environment specifically designed for scientific computing and data analysis.

I've posted a video tutorial on getting started with the Jupyter Notebook on the course website.

To launch the notebook, just type

*jupyter notebook*

into the command line, and press Enter. If everything is set up right, the notebook should launch in a browser window, and look something like this:



Select items to perform actions on them.

```
□ Applications
□ Data
□ Desktop
□ Documents
```

If you get errors, it probably means that one or more of the dependencies didn't get set up and installed properly. See if you can solve the issue yourself, or post the error message on the discussion board.

If you accidentally close the browser window, don't worry! You can get back to the Notebook by reopening your browser and going to: http://127.0.0.1:8888/

If everything launches properly, just hit the New Notebook button, and you're ready to go!

**Working with Notebook Files**

Every notebook you create is stored as a file with a .ipynb extension, in the folder or directory you navigated to in the Notebook interface. For example, I took the screenshot above after launching the Notebook from the *home* directory on my Mac. I can navigate to my *Documents* directory by clicking on it, and then create a new notebook file there with the *New* button, on the right.

**Be sure to save the work you do in the Notebook by pressing Ctrl- (or Command-) S, or choosing Save from the Notebook file menu.**

You can have multiple notebook documents open at once in different browser tabs or windows. When you're done and want to close down the entire Notebook, go back to your Terminal or Command Prompt, and hold down **Control-C**. Press **y** to confirm that you want to shut it down.

That's it; now that you're all set up, you're ready to move on to the next part of the lesson!