Science One CS 2015-2016: Getting Started

Note: if you are having trouble with any of the steps here, do not panic! We will resolve them this Friday when we meet from 10am-noon. You can also post questions on Piazza.

Note: if you are using an operating system other than Mac OS or Windows (e.g., Ubuntu), the process will be similar but slightly different. If you get stuck, we will sort it out on Friday.

1. Install Python with Anaconda
    - Go to http://continuum.io/downloads
    - Select your operating system, download **python 2.7** (not python 3.4), follow on-screen instructions using all default/recommended options (but, install for all users if possible)
2. Find your terminal
    - For Mac OS:
        - Click on the search icon in the top right corner
        - Type in “terminal” and press enter
        - If you want, right-click on the Terminal icon in your dock and select Options -> Keep in Dock (you will use this terminal throughout the course)
    - For Windows
        - There are many ways to do this. The Windows terminal is called “Command Prompt”. You can Google how to open the command prompt for your version of Windows. I have Windows 8 and I usually click on the Windows logo at the bottom-left, then click on the magnifying glass for searching, and then just type “command prompt” and press enter.
3. Test out Anaconda
    - In your terminal, type “python” and press enter. You should see something containing the word “Anaconda”. This is what I see:

```
Python 2.7.10 |Anaconda 2.3.0 (x86_64)| (default, May 28 2015, 17:04:42)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
```

4. Make sure packages are installed
    - at the python prompt (which looks like `>>>`),
        - type “`import numpy`” and press enter. If things are installed properly, nothing will happen. If there is a problem, you will get an error message.
        - Type “`import scipy`” and press enter.
        - Type “`import matplotlib`” and press enter.
        - Type “`1+1`” and press enter (you should see the result as “`2`”)
5. Exit python
    - To get out of Python, type “`exit()`” and press enter (or just press control-d)

6. Run spyder, your recommended text editor for the course (spyder is bundled with Anaconda so it should be installed already)
    - Type "`spyder &`" in the terminal and press enter
7. In a desired location on your computer, make a folder where you will keep your code for this course
    - example: /home/yourname/Documents/ScienceOneCS
8. Inside the folder you just made, make another folder named HelloWorld
    - I recommend doing this for each homework assignment (with the appropriate name of the assignment)
9. In spyder, make a new file (File -> New File...) and save it in the HelloWorld folder you just made, with the name HelloWorld.py. This will contain the first program!
    - you can close all the boxes in spyder except the one containing HelloWorld.py to have a clean screen. We will only use spyder as a text editor in this course.

STOP HERE! We will do the rest together in class on Friday.

The First Program - Hello World

1. Comments

Some lines in a .py file are actual code (that will run) and other lines are comments -- they do not run and they do not change what the program does. These comments are for programmers to write notes to themselves and to others who read the code. It is great programming practice to use comments to explain what your code does and help other people that may want to use and modify your code. In addition to this way of using comments, in this course you will use a comment at the beginning of each of your .py files to identify them as your own.

- comments are enclosed in three quotation marks (") on each side

```
""" This is a comment """
```

- comments that do not extend past one line can use the syntax

```
# This is a one line comment
```

- your HelloWorld.py already has a multi-line comment (the """ type) at the beginning. Modify this comment following the example below:

```
"""
Author: Michael Gelbart
UBC CWL: 12345678
Course: Science One Computer Science
Assignment: Hello, World!
Date: 10 September 2015
Description: Prints "Hello, World!"
Input: none
Output: "Hello, World!"
Example running command: python HelloWorld.py
```

"""

2. The actual code
    - type the following line in your HelloWorld.py, after the comment:

```
print "Hello, World!"
```

    - save the file (Command + S). It is great programming practice to save your current file frequently (every few minutes) while working on your program
3. Run your code (there are several ways to do it but this is the recommended way for this course):
    - open another terminal window (the first one you opened is now busy running spyder, don't worry about it)
    - navigate to your HelloWorld folder in the terminal using the three terminal instructions below
    - type `python HelloWorld.py` (and press Enter). You should see `Hello, World!` appear on the following line. You are done!

Three Useful Terminal Commands

& How To Run Your Programs

You can use the terminal to navigate through your computer's file system and run various applications. Here are some useful commands to navigate to a desired folder:

- `pwd` (on Mac) - shows the path of your current directory (i.e. tells you where you are)
    - on Windows, the terminal prompt contains the path of your current directory
- `ls` (on Mac) / `dir` (on Windows) - shows what is in the current directory
- `cd <directory or path>` (on both Mac and Windows) - change directory to the specified directory (NOTE: the use of the greater than and less than signs are meant to indicate that you need to type in whatever directory or path you want to go to. you should not literally type in these symbols into the terminal)
    - `cd ..` goes up one level, to the parent folder
    - extra tip: TAB does autocomplete for file names (start typing the name of a file/folder and press TAB before you finish typing)
    - For Windows only: to switch to another drive, for example to get to D: you need to type just `D:` and press enter rather than typing `cd D:`
    - Note: on Mac OS in the terminal, you can use the symbol `~` as shorthand for your home directory. So for example `cd ~/Desktop` will probably get you to your desktop. This is an absolute path rather than a relative path -- it points directly to your home directory regardless of what directory you are currently in, rather than some place relative to the current directory you are in. The rest of the time when you use `cd` to navigate around you are usually specifying relative paths (path relative to where you are right now)
- Note: if you directory names have spaces in them, you need to precede each space character with a backslash ("\") character. if you are using TAB to autocomplete it will do

this for you. In general, it will be easier if you files and directory names don't have spaces in them.

Once you are in the desired directory (folder), you can run things in that folder. You must navigate to the folder of your assignment (where your .py files are) to run .py files. Once you are there, you type:

```
python <program name>.py <command line arguments>
```

(again, see note above about the "<" and ">" symbols).

You can do a lot more with a terminal, but this is enough for now. For more commands, here are some references:

Mac OS X: http://ss64.com/osx/ (almost the same as Linux terminal commands)
Windows: http://ss64.com/nt/

Congratulations!