GIS 5103
GIS Programming

Fall 2019

Exercise 1b

Assigned September 5

Python code can be run in three different ways: 1) at the command line, 2) within a Jupyter Notebook and 3) using an integrated development environment (IDE). Each way has its advantages and disadvantages, so it is important that you know all of your options. You will be required to use the first and second ways for homework, quizzes and exercises; the third is optional.

1 Using Python at the Command Line

1. "Command line": A tricky term with multiple uses

    (a) The application: Sometimes we use the term "command line" to refer to the software application, i.e., "Terminal" on Mac and "Anaconda Prompt" on Windows.

    (b) A place to interact with the computer and start software: Sometimes we use the term to refer to the baseline position of the Terminal or Anaconda Prompt applications.

    - On Mac it looks like:
      `computer_name:~ user_name$`
      The key thing here is the dollar sign (`$`), which indicates you are at the command line prompt.

    - On Windows it looks something like:
      `(base) C:\Users\user_name>`
      The key thing here is the greater than sign ($>$), which indicates you are at the command line prompt.

    (c) A place to enter python commands: Sometimes we use the term to refer to the python interpreter prompt. You will see how to start this in the next section; for now here is what it looks like:

    - The classic python interpreter: `>>>`
    - The modern IPython interpreter: `In [1]:`

2. Launch the command line application.

   - Mac: Launch the Terminal application (Applications > Utilities > Terminal)
   - Windows: Launch the Anaconda Prompt application (Start > Anaconda3 (64 bit) > Anaconda Prompt)

3. Manually enter code using the classic Python interpreter.

   (a) Type
   ```
   python
   ```
   at the command line, then hit enter. You'll know you're in Python when you see the Python prompt: `>>>`.

   (b) Type in the line
   ```
   print("Hello World")
   ```
   then hit enter. Assuming there are no typos, the line should simply print the text in quotes.

   (c) Type
   ```
   exit()
   ```
   to end your Python session.

4. Manually enter code using the more modern (and more useful) IPython interpreter. You should almost always use the IPython interpreter instead of the classic Python interpreter.

   (a) Type
   ```
   ipython
   ```
   at the command line, then hit enter. You'll know you're in the IPython interpreter when you see the IPython prompt: `In [1]:`. Note: the first time you launch IPython it may be a little slow to start.

   (b) Type in the line
   ```
   print("Hello World")
   ```
   then hit enter. Assuming there are no typos, the line should simply print the text in quotes.

   (c) Hit control-D to end your IPython session (you can also type `exit()`).

5. Instead of typing out the code each time, you can put all the code into a script and run that. A python script to pull the last year of weather data for Tallahassee is in a python script called `tlh_weather_get_data.py`, which is available through the course repository. Python scripts use the `.py` extension.

   (a) At the command line (`$` on Mac; `>` on Windows), navigate to the directory containing `tlh_weather_get_data.py` on your hard drive using the `cd` ("change directory") command.

      - Mac how to: `http://www.computerhope.com/unix/ucd.htm`
      - Windows how to: `http://www.computerhope.com/cdhlp.htm`

(b) Run the script directly using IPython. Type
ipython tlh_weather_get_data.py
at the command line. This will run the script immediately. Note: you could also use the classic interpreter (python tlh_weather_get_data.py).

(c) Run the script from within the IPython interpreter. When you run a script this way, you will have access to all the results from your script so you can continue working with them.

    i. Type
ipython
at the command line and hit enter to start IPython.

    ii. Type
run tlh_weather_geta_data.py
at the IPython command prompt to run the script.

    iii. Hit control-D to end your IPython session.

2   Using the Jupyter Notebook

Homework and quizzes will be conducted mostly in Jupyter Notebooks. We will be working with `exercise1b.ipynb` for this exercise, which is in the same directory as `tlh_weather_get_data.py`. Navigate to that directory using `cd` if you are not already there.

1. To start, type `jupyter notebook` at the command line (`$` on Mac; `>` on Windows). This will launch the Jupyter Notebook interface in your web browser. It may take a few seconds for the web page to launch.

2. In the web browser interface click `exercise1b.ipynb`. This will launch a new tab.

3. Read through the exercise Notebook. Play around with it to get a feel for how it works. Be sure to explore: `Rename`, `Save and Checkpoint` and `Close and Halt`.

4. Go back to the tab with the main Jupyter Notebook interface, and launch a `New` (upper-right of web page) `Python 2` Notebook. Enter some stuff in there (e.g., copy and paste lines from the Python script for getting the data for Exercise 1a, and run them).

5. To close the Notebook tabs in your web browser, click `File > Close and Halt` in each Notebook. Just close the Launcher tab the way you would close any tab.

6. Go back to the command line and hit control-C. Follow the directions to end the Jupyter Notebook session.

7. Find the Jupyter Notebook you just created on your hard drive (it will have a `.ipynb` extension). Email this file to yourself (**and** me) announcing that you created your first Jupyter Notebook. **DO NOT SKIP THIS STEP! This is how you will submit your first quiz. If you cannot submit your quiz, I cannot grade it, and you will get zero points.**

8. You might want to watch this video for more information on using the Notebook. Note that this tool was rebranded a couple years ago from the "IPython Notebook" to the "Jupyter Notebook." `https://youtu.be/lmoNmY-cmSI`

3 Spyder IDE (optional)

When you downloaded Anaconda, you also got the Spyder integrated development environment (IDE). Spyder allows you to edit your code (i.e., the `.py` file) and run your code in the same environment. This is pretty self-explanatory (and also not required for the course). An alternative to the IDE is to edit your code in a text editor (Notepad++ or BBEdit) and run it at the command line.