

Data Analytics Python Module:

Week 1 Guide

1 Introduction

This document outlines the basic materials to explore for week 1 of this course, focusing on several different ways to execute Python commands with your computer. In future weeks, this expository material will be integrated into a Jupyter notebook with code examples, but since we are just starting out it makes sense to begin with a more familiar setup. For introductory classes like this, it has become common to have the initial code examples write the phrase “Hello, world!” to the terminal, so we will follow that social convention here. Your assignment for the first week is to follow all of the directions in this .pdf and to reach out for help if you run into problems. In order to be successful later in the course it is very important that you get python properly installed this week.

Before proceeding on to the rest of the sections in this guide, make sure that you have followed the directions in the Python_Installation_Guide.pdf to install Python on your computer and read the following textbook sections:

- [About Python](#)
- [Why Python?](#)
- [Jupyter Notebooks](#)
- [Think Python: The way of the program](#)

2 Scripts

Collections of python commands can be saved in text files that end with the .py extension. You can edit these with your favorite text editor and run them directly from the command line. If you followed the instructions for installing anaconda and creating a fresh environment for this course you should have access to an “Anaconda Prompt” terminal. Launch the terminal now and use the ‘cd’ command to navigate to the directory where the Week1 files for this course are saved. On my computer that is:

```
>>>cd Dropbox\Python_Module\Week1
```

Now that your terminal is in the correct directory, you can run one of the provided scripts by typing the word python and then the name of the file:

```
> python my_first_script.py
```

which should write the phrase “Hello, world!” to the screen. Congratulations on running your first python program ☺ Next, try to run the other .py script in the file:

```
> python input.py
```

which will ask you to enter your name and print a welcoming message. You can open the .py files with a text editor to see the code that python ran. In later weeks we will discuss the syntax that makes these examples work, but for now you can try modifying the entries between then “ ” in the scripts and re-running them through the console. What does this change?

3 Interactive Terminal

In addition to running prepared scripts as in the previous section, we can also type python commands directly into an interactive python terminal. Launch an anaconda prompt and type:

```
> python
```

after you hit enter you will see some lines describing the version of python you have installed and some help functions, followed by `>>>`. At the new python prompt, type:

```
>>> print("Hello, world!")
```

and after you hit enter the message will again be printed to the screen. To leave the python prompt, type:

```
>>> exit()
```

Notice that if you try to type the print command in the regular anaconda prompt (with one `>`) it will generate an error message. Python commands only work within the python session.

4 Spyder

Most programming doesn't take place directly in the console or with a standard text editor. Instead, code is usually developed inside an Integrated Development Environment (IDE). These are graphical applications that allow you to edit, analyze, and run code all within a single window. Anaconda comes with an IDE called Spyder that can be launched from the prompt by typing:

```
> spyder
```

After hitting enter, a new window will open with several panels. For now we will focus on the bottom right panel, which is an interactive terminal like the one we encountered in Section 3 and the left panel, which is a text editor for python files. You can type

```
>>> print("Hello, world!")
```

in the lower right to see the same output as before. After testing that, use the File→Open menu to open one of the .py scripts in the week1 folder. It will display in the left panel and you can modify the script there directly. You can run the script by clicking the green arrow at the top of the panel - the output will display in the interactive terminal on the bottom right.

5 Jupyter Notebooks

Most of our work in this class will take place in Jupyter notebooks, which have become one of the most common tools for doing and sharing python-based data analysis projects. From your anaconda prompt, you can type:

```
>>> jupyter notebook
```

to launch the jupyter interface, which should appear as a new tab in your default web browser. Navigate through the file directory to where you have saved the week1 materials and open the my_first_notebook.ipynb file to continue.

In terms of using notebooks as development environments for full-scale projects the research community is a little divided but the following two keynote speeches from JupyterCon cover a lot of the key points:

- [I Don't Like Notebooks \(Joel Grus\)](#)
- [I like Notebooks \(Jeremy Howard\)](#)

6 Assignment

Your homework for week #1 is to follow the directions in the previous sections of this document, including installing Python. Make sure to run all the scripts and interact with the notebook, as the remainder of the course assumes that you have been able to complete these tasks. If you get stuck or encounter error messages, please reach out to your instructor immediately. While they may not be able to respond right away, it is important to get off to a strong start in a programming-focused course like this one.