AST 376R: An Introduction to Programming With Python

Jonathan Florez, Shardha Jogee

Contents

1 Introduction

Python is a high-level programming language that is used by scientists in a variety of fields (e.g., astronomy, physics, computer science, data science, engineering), web developers, software engineers, and many other members of the STEM community to process and analyze data. Python was designed to be much more readable than other languages, meaning it has fewer syntactical constructions than other programming languages and uses intuitive keywords where other languages might use punctuation. It has a wide range of applications, but it is largely optimized for handling arrays (vectors and matrices of data), making plots, software development, and designing graphical user interfaces (GUIs). A major advantage of Python is that the scientific and engineering community have built many codes and modules in Python to perform a variety of useful operations. In addition, Python is developed under an open source license, which makes it free and distributable, even for commercial use.

The programming module for this AST 376R course used to focus on the Interactive Data Language (IDL) programming language and if you find it useful, you can still access the IDL tutorial on the class website (www.as.utexas.edu/ sj/a376-sp21/secure/idl/idl_intro.final.pdf). While IDL still remains a widely used programming language in the astronomy community, we are transitioning the programming module to the Python programming language for several reasons. IDL requires an expensive license that makes is less accessible to many users. Python has a similar functionality as IDL and many of the astrophysics data reduction and analysis pipelines are now written in Python. Many machine learning platforms, high level statistics computing packages, and other powerful programming tools are also Python-based. Finally Python is highly versatile, and is widely used in both academia and industry.

The goal of this tutorial is to introduce you to programming practices in Python. In this tutorial, you will use the Terminal application to launch Jupyter Notebooks and run Python. This document will tell you how to launch Python and retrieve the tutorial files online. The rest of the tutorial will be done through Jupyter Notebooks (more information on that below).

2 Covered Topics and Tutorial Files

This tutorial will cover many of the same topics that were covered in the IDL tutorial in this course. We will use many of the Jupyter Notebooks modules written by Jackie Champagne, a UT astronomy graduate student and we would like to express our deep thanks to Jackie for generously sharing her modules for this tutorial. Here is a list of topics we will cover in different lectures:

- Part 1 - Basic Syntax, Variable Types, Arrays
- Part 2 - Functions, Reading and Writing Files
- Part 3 - Control Statements
- Part 4 - Plotting Figures
- Part 5 - Statistical Analyses

- Part 6 - Astropy, FITS files, Aplpy

To get started with this tutorial, create a folder in your home directory named 'python' using the `mkdir` command and move into that directory with the `cd` command. Once in that directory, use `curl` to download the tutorial files in a compressed folder.

```
curl -u cosmos:ast376web www.as.utexas.edu/~sj/a376-sp21/secure/python/
python.tar.gz -o python.tar.gz
```

Unpack the compressed file as you normally would to generate all the files you need for this tutorial.

```
gunzip python.tar.gz
tar xvf python.tar
```

In the next section, we will discuss how to launch Python from your terminal to get started working on the tutorial.

3 Launching and Running Python

There are multiple options for launching and running Python:

- One can use a Python graphical user interface (GUI), which functions as an application with windows, buttons, and widgets that allow you to interact with Python scripts and generated graphics.
- One can run the Python code on your laptop by executing the command '`python` *filename.py*' in your Terminal application, where '*filename.py*' is the name of your Python script. This will execute all the commands that are in your Python script, line by line.
- One can also launch an interactive Python environment in your Terminal by typing `python` or *ipython* and hitting `return` (note that this will work no matter what directory you are in). This is a good option if you want to quickly execute a few Python commands, but not so much if you want to develop in-depth code.
- One can also use Jupyter Notebooks, a web-based programming tool that has become increasingly popular with researchers and data scientists for interactively developing and presenting research projects. This is a great option for developing in-depth code by yourself or with collaborators as it allows you to easily draft, debug, and share code with others. For this course, we will use Jupyter Notebooks to run, develop, and execute some Python code.

To begin the tutorial, make sure you are in your 'python' directory where you uncompressed the Python .tar.gz file. In that directory, type the following to launch Jupyter Notebooks:

```
jupyter notebook
```

This will launch Jupyter Notebooks in your web browser. You will notice that all the files in your 'python' directory will appear in the web browser page that just opened. This is because Jupyter Notebooks is being executed directly on your laptop, even though you are working through a web browser. This also means that running Jupyter Notebooks does not actually not require an internet connection if it is installed on your laptop (as is the case with your UT laptop).

The rest of this tutorial will be done through the Jupyter Notebook (.ipynb) files that are in your working 'python' directory. Each day we will work on a different 'notebook' (the name given to a Jupyter Notebooks file). For the first lecture, we will start with the notebook named 'Part_1.ipynb' and go from there.

4 How to Use Jupyter Notebooks

As you work through the tutorial, it may help to know a little about how Jupyter Notebooks work. Here are a few useful functions one should know when using notebooks:

- To run the code in a cell, click anywhere on the cell and hit `shift+return`. This executes all the lines of code in a given cell. You can also click the 'Run' button near the top of the page to do this while a cell is highlighted.
- If you wish to execute all the code in a notebook, you can click on the 'Cell' tab near the top, scroll down on the pull- down menu and select 'Run all'
- Make sure to always run the first cell in a notebook where the modules are imported before running anything else. If you do not, you will receive errors about modules not being imported
- To completely clear all the stored variables and reset a notebook, you can click on the 'Kernel' tab near the top and scroll down to select 'Restart & Clear Output'. This option may be useful if you think a variable is stored incorrectly or if you want to reset all the loaded modules, variables, etc.
- If you wish to see the value of a variable or the output of a command, simply type `print()` near the bottom of a cell with the variable or command you want to display. Multiple variables/commands can be separated by a comma in `print` (e.g., `print(x,y+z,k,...)`). This is a universal Python function and is not exclusive to Jupyter Notebooks.
- To write text in a cell, instead of code, you can highlight a cell by simply clicking on it and hitting the `m` key, or by going to the 'Cell' tab near the top, scrolling down to 'Cell Type' and selecting 'Markdown'. You will not need be required to do this in the tutorial notebooks, but is useful to know if you ever have to put together your own Notebook from scratch.