1 SIO 113: Python Programming for Earth Science Students

Authors: Lisa Tauxe, ltauxe@ucsd.edu Hanna Asefaw, hasefaw@ucsd.edu, & Brendan Cych, bcych@ucsd.edu

- Winter 2019
- Lectures: MWF 4-4:50
- Discussion Section: W 3-3:50
- Location: Eckart 225

1.1 Computers in Earth Science

Computers are essential to all modern Earth Science research. We use them for compiling and analyzing data, preparing illustrations like maps or data plots, writing manuscripts, and so on. In this class, you will learn to write computer programs with special applications useful to Earth Scientists. We will learn Python, an object-oriented programming language, and use Jupyter notebooks to write our Python programs.

1.2 Python

So, why learn Python? Because it is:

- Flexible, freely available, cross platform
- Easier to learn than many other languages
- It has many numerical, statistical and visualization packages
- It is well supported and has lots of online documentation
- The name 'Python' refers to 'Monty Python' - not the snake - and many examples in the Python documentation use jokes from the old Monty Python skits. If you have never heard of Monty Python, look it up on youtube; you are in for a treat.

Which Python?
- Python is undergoing a transition from 2.7 to 3. The notebooks in this class, apart from a few exceptions, are compatible with both but they have only been tested on Python 3, so that is what you should be using.
- If you decide to use a personal computer, we recommend that you install the most recent version of Anaconda python for your operating system: https://www.anaconda.com/download/ you will also need a few extra packages (cartopy, version 0.17.0 and PySimpleGUI) which can be installed with little hassle.

1.3 Grading

- 25% Practice Problems for each lecture
- 25% Final Project
- 50% Weekly Homework assignments

1.4 Class Structure

- There will be three lectures a week and one discussion session.
- Students are expected to read the lecture prior to attending class

- Each lecture begins with a quick review (~5 min) and proceeds to the topic of the day. Lecture time will be devoted to practicing the skills that we covered in the lecture.
- At the end of every lecture, students will submit their practice problems - these are Jupyter notebooks that you download with each lecture, complete and turn in on TritonEd. To get credit the practice notebook must run without errors. You will have ample time in the lecture to make sure this happens! We will count up to 25 (out of 27 possible) practice notebooks. The first one (for today's lecture) is due by 5 pm tomorrow. All others are due at the end of each lecture.

- There will be a programming assignment every Friday, due BEFORE CLASS one week after it is assigned. Assignments will count for 50% of the grade. There are nine of them, so you do the math.

- Help with assignments and the solutions will be disussed during weekly discussion section on Wednesdays before class. The TA will help you both before they are due and after they have been graded.

- In lieu of a final exam, there will be a final project. This is a program of your own design, and a notebook cell of approximately one page length that contains a description of the project. There is a great deal of flexibility in what the program will do, but there are a few minimum requirements. We will discuss the final project in more detail later. Final projects will count for 25% of the final grade. In addition to the project, students will prepare a 5 minute presentation of their work, to be given either on the last day of class or Monday March 18th during the "final exam" slot 11:30-2:30. These are fun and attendence at least one of the sessions is required. But you should plan on attending both as the projects can be quite interesting.

1.5 Class Expectations

- Attendence is strongly suggested but not required. Submission of the practice problems IS required as are the weekly homework assigenements and the final project. None of these will be accepted late.

- You may consult any online resources (Stackoverflow is a wonderful resource) or any of your fellow students or your instructors to help you solve your problems. Although this is encouraged, do NOT copy what you find verbatim. You must re-work the solutions through your own brain and in your own words and style, otherwise you will not learn how to program. Copying programs does not help you learn and in fact it is "cheating". Cheating will be reported to the authorities.
- The best way to learn how to program is to do it. This is helped by attending the lectures, completing the practice problems and assignments, and attending the discussion section where your TA can help.

1.6 Schedule

Lecture	Topic	Date	Assignment
1	Intro to notebooks, file systems and paths	7-Jan	
2	Variables and Operations	9-Jan	
3	Data structures	11-Jan	#1
4	Dictionaries, program loops (if, while and for)	14-Jan	
5	functions and modules	16-Jan	
6	NumPy and matplotlib	18-Jan	#2
7	NumPy arrays	23-Jan	
8	Pandas, file I/O	25-Jan	#3
9	data wrangling with Pandas	28-Jan	
10	object oriented programming	30-Jan	
11	lambda, map, filter reduce, list comprehension	1-Feb	#4
12	Pandas filtering and exceptions	4-Feb	
13	subplots, bar charts pie charts	6-Feb	
14	histograms and cumulative distribution functions	8-Feb	#5
15	statistics 101	11-Feb	
16	line and curve fitting	13-Feb	
17	visualization with seaborn	15-Feb	Project Proposal & #6
18	maps	20-Feb	
19	gridding and contouring	22-Feb	#7
20	rose diagrams and equal area projections	25-Feb	
21	matrix math - dot and cross products	27-Feb	
22	plotting great and small circles	1-Mar	#8
23	Graphical User Interfaces (GUIs) and animations	4-Mar	
24	Machine learning	6-Mar	
25	3D plots of points and surfaces	8-Mar	#9
26	Time series - periodograms	11-Mar	
27	Animations & Indian plate motion	13-Mar	
	Final Presentations Part I	15-Mar	
	Final Presentations Part II	18-Mar 11:30-2:30	

1.7 Final project

As you learn new concepts, start thinking about what you'd like to create for your final project.

There is a great deal of flexibility on the exact nature of the final project but it must be related to Earth and Space Science and, at the bare minimum, it must:
- include at least one module with three functions - read in at least one data file - make at least one plot - use at least three markdown blocks:
- a description of what the program does, - instructions on how to use the program,
- a summary of the scientific conclusions

Turn the project in as a zipped directory with all the parts required to run it (the data files, figures, modules, etc.).

Here are a few ideas:

1) Make a movie:

- of lightning strikes across the continental US
- of volcanic eruptions across the western US
- of plate motion over the last 200 Ma
- your choice

2) Make a 2D image of the solar system with orbiting planets
3) Recreate your favorite plot from a number of examples
4) Design your own project

In Week 6, you will be asked to turn in a proposal for the final project. In the proposal, you'll describe the final project and how it relates to Earth and Space Science. At that stage, you still may not have all the skills required to complete your project, but we can let you know if it is possible and substantial enough for the final project.