# SIO 113 Introduction to Computational Earth Sciences

*Spring 2021*

*Instructor:* Dave May (dmay@ucsd.edu)

*Assistant:* Brendan Cych (bcych@ucsd.edu)

## 1. Introduction

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.

This course is entirely structured around a special programming environment called Jupyter notebooks. A Jupyter notebook is a development environment where you can write, debug, and execute your python programs.

## 2. Python

### 2.1 What is Python?

From wikipedia

> Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

> Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

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.

### 2.2 Why Python?

- Easier to learn than many other languages.
- Extremely flexible and versatile.
    - Many numerical, statistical and visualization packages.
- Freely available and cross platform (works any system).
- It is well supported, actively developed and has lots of online documentation.
- It can help you improve how you conduct your science (currently), and enable new scientific enterprises to begin.
- Python programmers are in demand.

### 2.3 Which Python?

The notebooks in this repository are compatible with Python 3.6+. While most of the notebooks are compatible with Python 2.7, we do not test or maintain backwards compatibility.

### 2.4 How will we use Python?

We will learn Python via Jupyter notebooks. A Jupyter notebook is a development environment where you can write, debug, and execute your python programs. It is a web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Jupyter notebooks do not have to *exist, or be run on your computer*, they can be created and *run on a remote machine* - and you only *access* them via a web browser (e.g. Firefox, Safari, IE, Chrome). Hence you do not even need to install or run Python on you machine.

We will primarly use the service provided by MyBinder (https://mybinder.org) to run Jupyter notebooks. See Section 9. Resources (point 3) for a one-click option to laucnh you personal Python Jupyter environment for this course.

## 3. Realities of Programming

1. There is no such thing as theoretical computer programming - it is a practical skill. The only way to learn the art of programming is thus to practice it.
2. You must practice, practice, practice. Practice makes perfect.
3. Do not be deterred or discouraged by immediate failures (i.e. “code does not work”).
4. Programming mistakes (bugs) are inevitable.
5. Writing code = introducing bugs, diagnosing bugs and fixing bugs (debugging). Be prepared that 1% of your time may be spent on programming and 99% of your time may be spent debugging.
6. You should only believe code you can run.
7. There is no such thing as a perfect code. Write code for a specific design objective.
8. The learning cycle is different for everybody, as is the time-scale over which you will learn to program. Everyone can learn to program.
9. The only way to learn and improve is to practice. See 1 and 2.

# 4. More Realities of Programming

**"Bugs are inevitable"**

1. If your code works first time you run it - be suspicious. Test it again under different conditions (i.e. inputs).
2. Every line of code you write can introduce a bug.
3. The most difficult bugs to identify are often trivial to fix. Don't give up. Persevere.
4. If you see something strange in your results there is probably a bug. Be suspicious - test everything.
5. A single bug can ruin your beautiful code. Be motivated to carefully debug and test your code. A single small error in the code cannot be ignored. Laziness never pays off.
6. Creating a correct and working code is possible. Perseverance pays off.

# 5. Course Format

*Three lectures per week (28 lectures in total):*

- Wednesday 10:30 - 11:20
- Wednesday 11:30 - 12:20
- Friday 09:00 - 09:50

*One practical / discussion session per week:*

- Friday 10:00 - 10:50

## 5.1 Lectures

For every lecture, there is an accompanying Jupyter notebook. You should read and familiarize yourself with the notebook before the lecture.

Every lecture will be structured as follows:

- A recap from the previous lecture, followed by a short introduction to the new material (~5-10 mins).
- In the next 30-35 mins, we adopt a flipped classroom format. You will be re-reading the preprepared Jupyter notebook. This format is adopted to ensure you learn how to program - remember you will only learn the art of python by doing it yourself. Brendan and myself will assist you, and fill in any knowledge gaps.
- A final wrap-up in the last 5 minutes.

## 5.2 Practical / Discussion Sessions

The format of the discussion sessions is left intentionally open. It is your time. The time is intended for to consildate your understanding of programming and master your new Python skills.

Brendan and myself will be present to assist with any and all questions you may have regarding lecture material, practice problems or assignment questions.

We expect you will use this time to:

- work on your practice problems, homework and or the final project;
- ask questions you may have regarding any lecture material, practice problems, assignments or the final project.

No Python topic is off-limits - if you have a quesions - please ask. We are here to help.

There will be an assignment distributed at the beginning of each practical class which will be presented (~5 mins).

In addition, in some discussions we will have short "pop-up" presentations from SIO faculty to present how and what they are using Python for in their day-to-day scientific research. This should provide useful insight for final project ideas.

## 6. Schedule

Below is a tentative schedule.

"L" = lecture; "P" = practical; "HW" = homework

| Class type | Date | Topic | Action |
|---|---|---|---|
| L | 31-Mar | L1: Introduction, Python, Jupyter notebooks | |
| L | | L2: Variables and operations | |
| L | 02-Apr | L3: Data structures | |
| P | | Discussion 1 | HW#1 distributed |
| L | 07-Apr | L4: Dictionaries, loops | |
| L | | L5: Functions & modules | |
| L | 09-Apr | L6: Numpy and matplotlib | |
| P | | Discussion 2 | HW#2 distributed; HW#1 due |
| L | 14 | L7: Numpy arrays | |

| | | | |
|---|---|---|---|
| L | 14-Apr | L7: Numpy arrays | |
| L | | L8: Numpy, matplotlib, pandas | |
| L | 16-Apr | L9: Pandas filtering | |
| P | | Discussion 3 | HW#3 distributed; HW#2 due |
| L | 21-Apr | L10: Object oriented programming | |
| L | | L11: Lambda, list & dict comprehension, exceptions | |
| L | 23-Apr | L12: Data wrangling with Pandas | |
| P | | Discussion 4 | HW#4 distributed; HW#3 due |
| L | 28-Apr | L13: More Pandas, subplots, bar charts and pie charts | |
| L | | L14: Histograms, distributions | |
| L | 30-Apr | L15: Statistics | |
| P | | Discussion 5 | HW#5 distributed; HW#4 due |
| L | 05-May | L16: Line and curve fitting, scikit-learn | |
| L | | L17: cartopy | |
| L | 07-May | L18: geoplot & geopandas | |
| P | | Discussion 6 | HW#6 distributed; HW#5 due; Project proposal due |
| L | 12-May | L19: Rose diagrams, projections | |
| L | | L20: Matrix math / linear algebra | |
| L | 14-May | L21: Plotting great and small circles | |
| P | | Discussion 7 | HW#7 distributed; HW#6 due |
| L | 19-May | L22: scikit-learn: Cluster analysis | |

| | | | |
|---|---|---|---|
| L | | L23: scikit-learn: Classification | |
| L | 21-May | L24: Gridding & contouring | |
| P | | Discussion 8 | HW#8 distributed; HW#7 due |
| L | 26-May | L25: 3D scatter plots and surfaces | |
| L | | L26: Time series analysis | |
| L | 28-May | L27: Animations | |
| P | | Discussion 9 | HW#9 distributed; HW#8 due |
| L | 02-Jun | Beyond this course | |
| P | | Practical, project work | |
| P | 04-Jun | Practical, project work | |
| P | | Discussion / project work | HW#9 due |
| P | 09-Jun | Student presentations (part 1) | |
| P | | Student presentations (part 2) | |
| - | 11-Jun | | Final project due |

## 7. Assessment

There is no final exam. The assement is progressive and distributed over the quarter. The breakdown of the assessment is as follows.

- *25% Practice problems (25 in total)*
    - We will take the best 23 of 25.
- *50% Homework assignments (9 in total)*
    - We will the best 8 of 9.
- *25% Final project*

All assessments are to be submitted via Canvas.

Precise instructions are provided in each practice problem and homework regarding how the submission should be organized. Please pay close attention to the naming convention requested in each assessment. Failure to adhere to the guidelines provided will result in penalties.

Remember, the rule "You should only believe code you can run": we will be applying this rule when grading your work. If we cannot run the Python code you have submitted you will not receive full marks. You must submitted all data files, inputs required to reproduce your results. We will not chase if your submission is incomplete - this is your responsibility. Please test your submissions before uploading them to Canvas.

## 7.1 Practice Problems

For lectures 2 through 26, there is an accompanying set of exercise to be completed. The intention is that you complete these during each lecture. Due to the course scheduling, we will use the following submission schedule:

- Practice problem from Wed. 10:30 lecture is due by midnight on Wed.
- Practice problem from Wed. 11:30 lecture is due by midnight on Thur.
- Practice problem from Fri. 09:00 lecture is due by midnight on Fri.

## 7.2 Homework

A homework assignment will be distributed at the beginning of each discussion session.

- Each homework is due by midnight on the following Fri.

## 7.3 Final Project

The final project has three important deadlines.

- Project proposal due 07-May (week 6).
- Project presentation 09-Jun.
- Final submission 11-Jun.

The final project is your opportunity to be creative and showcase the Python skills you have learned during this course by applying them to an Earth science application. 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.

For example:

- Make a movie of lightning strikes across the continental US, or of volcanic eruptions across the western US, or of plate motion over the last 200 Ma;
- Make a 2D image of the solar system with orbiting planets;
- Design your own project.

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

Whatever the specifics of your project, at the bare minimum your project must:

- include at least one module with three functions;

- read in at least one datafile;
- make at least one plot;
- use at least three markdown blocks covering (i) a description of what the program does; (b) instructions on how to use the program, (c) a summary of the scientific conclusions.

You will submit the final project to Canvas as a zipped directory with all the parts required to run it (the data files, figures, modules, etc.).

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.

On 09-Jun (exam week), you will give a 5 minute presentation to the class about your project. Unless prior approval is made, attendance is required.

## 8. Expectations

- Read the notebook before coming to the lecture.

  Please note that the course material is dynamic - we improve the material over the duration of the quarter (and fix bugs). Whilst all notebooks are already available, please download them at the beginning of each week. Unless otherwise stated, the provided notebooks for the week will not be altered after Sunday morning.
- Weekly homework assignments are mandatory as is the final project.
- Homework will not be accepted late. For anyone attending asynchronously (temporarily, or for the entirety of the course), please contact me.
- Code submissions must be complete - all required data files must be provided.
- You may consult any online resources, your fellow students & your instructor/TA to help you solve your problems. This is encouraged, however do NOT copy what you find verbatim. You must re-work the solutions in your own words and style, otherwise you will not learn how to program. Copying programs does not help you learn and is considered "cheating". Cheating will be reported to the authorities.
- The best way to learn how to program is to attend the lectures, complete the practice problems and assignments, and attend the discussion section where your instructor/TA can help you.

## 9. Resources

1. All course material is available at the following website

   https://github.com/hpc4geo/Python-for-Earth-Science-Students

   - Lectures are found in the root directory and are named

     ```
     Lecture_XX.ipynb
     ```

- All practice problems are found in the sub-directory

  Practice_Problems/

  and are named

  ```
  Lecture_XX_Practice_Problems.ipynb
  ```

- All homework assignments can be found in the sub-directory

  Assignments/

  and are named

  ```
  Assignment_XX.ipynb
  ```

2. You may download the entire contents of the course material from the following URL

   https://github.com/hpc4geo/Python-for-Earth-Science-Students/archive/refs/heads/main.zip

   Remember content may be altered / updated / improved over the quarter, so if you are downloading the material, do so regularly (weekly).

3. Interative Jupyter notebook environments can be accessed via the following URL

   https://mybinder.org/v2/gh/hpc4geo/Python-for-Earth-Science-Students/main

   or by simply clicking the "launch binder" badge within the README file displayed when accessing the URL in item 1.