Milestone 7

Due Tuesday, June 6 in class

Learning Objectives:

III. **What broke and how do I fix it?** *How do I debug a whole project?* This milestone asks you to make your code, from start to finish, work and produce correct results. This means you'll have to deal with all the little bugs that may have come up while working on previous milestones. You'll also have to test the results and improve the code until it works as expected.

IV. **How do I communicate science and Python with others?** *How do I document an entire project?* This milestone gives you the opportunity to practice quickly and efficiently documenting your project so that other programmers can understand how it works. *How do I motivate and explain a project?* This milestone asks you to prepare you for your final project presentation. This gives you an opportunity to practice motivating, demonstrating, and explaining your module.

A. **Python is a physical system. Experiment!**
B. **Let me Google that for you.**
C. **Computing time is cheap—use it.**
D. **Read the error output. Read it.**
E. **Don't reinvent the wheel.**
F. **Write and test, write and test…**

While You Work: Habit Summary #6

We've talked about six useful habits that scientific programmers have. (See above) You've started using these habits, possibly without knowing it! This part of the milestone will help you notice and solidify those habits.

While you're working, you will doubtless make use of one of these habits. When you notice yourself using **the remaining habit about which you haven't already written a summary**, write down the habit and what you used it for. See Milestone 3 for an example.

Part 0: Finish Your Code

You know what you have to get done before your project is done. **Make sure to test your code** so that you know it's returning the correct results.

Part 1: Document Your Code, Sparsely

An instructor looking through your code should be able to tell roughly what is happening in each module, section, function, and class, without asking you. Document your code with docstrings and comments as necessary to make this the case. Don't go overboard; you presentation is more important.

Part 2: Prepare a 5–10-Minute Presentation

You'll give this in class next Tuesday. Please discuss:

1. What does your module do? What motivation would the class have for learning about your module? (1–2 minutes, as introduction)
2. How do you use your module? (2–4 minutes of demonstration)
3. What is one programming obstacle you overcame during the project? What did you learn that helped you overcome it? This can be a large-scale obstacle, such as how to organize the project, or a small-scale obstacle, such as a single resilient bug, or anything in between. (2–4 minutes of explaining a short block of code, 10–20 lines)