

Python Capstone Project

Physics91SI Spring 2013

Objectives: In this project, you will

- Apply and solidify the Python knowledge, techniques and paradigms you have learned already in the class
- Learn how to find, appraise, and use Python packages
- Experience the challenges and rewards of the process of creating a substantial piece of code

Part 1: Project Proposal

Your project should

- Take six (6) hours outside of class to code
- Use what you have learned in the class
- Feature a central idea which is manageable in about half the time to allow leeway for unexpected setbacks
- Have room for refined functionality so that the rest of the time can be used productively
- Be scientific in nature or application

Due Fri 5/17 at 5 p.m.: Write a project proposal in which you summarize your plans for the project. Please specify the central idea and some of the refined functionality you can implement. Please write it in on corn: set up a repository using `hg init`, write your proposal in a text file called `proposal.txt`, add it to the repo, and then submit it using the command below. Guideline: 200 words.

```
hg push /afs/ir.stanford.edu/class/physics91si/submissions/user/proposal
```

Part 2: Package Search

As part of the project, you will be searching for packages that offer functionality you'll need for your project. By the end, you will be familiar with the features you used, and with what worked well and what didn't. We are asking you to write a short exposition of what you learned so that we can compile a document of helpful packages to share with the class.

Due Wed 5/29 at 5 p.m. (same as project due date): Please select one particularly neat feature from one of the packages you found, such as a function or paradigm it implements, and write a short piece of documentation with the following information. Please call it `packagedoc.txt` and include it with the repository you will set up in Part 3, below. Guideline: 150 words.

- The name of the feature and the name of the package that implements it
- What it says it does and what you did with it

- What's cool about it
- Any setbacks you had when using it and how to avoid them

Part 3: The Project

Write the code and get it working!

A few tips, to make you life easier (not required, but recommended!):

- You may install Python and work on your own computer, with a text editor or IDE of your choice, or you may work on the corn machines. However:
- Save your work on the corn machines regularly. (We will discuss how to upload to and download from corn.) Set up a Mercurial repository with `hg init` so that you can keep your work backed up and revert to past versions, if necessary. You will be submitting your code through corn at the end anyway.
- Document your code using docstrings. See <http://www.python.org/dev/peps/pep-0257/> for guidelines on writing docstrings. This lets you access your comments by typing `help(<function or object>)`, or using `<function>?` in IPython.
- Include error handling! You'll save on bugs if you catch bad input from the beginning.
- Finish the core idea of your project before adding additional functionality.
- Let us know as soon as possible if you have questions, concerns, or bugs that just won't resolve themselves. If you get started early, you'll find these early, and we're more likely to be able to help you.

Due Wed 5/29 at 5 p.m.: Add your document from Part 2 to the repository you created for Part 3. Then submit your code and your package documentation using

```
hg push /afs/ir.stanford.edu/class/physics91si/submissions/user/capstone
```

We will look at your projects over the following week and return them to you at our last class on Tuesday, 6/4.

Part 4: The Presentation

You'll be telling the class about your project.

Due Tue 6/4 in class: Prepare a presentation using the medium of your choice about whatever aspects of your project you want to share. We will stop you after 10 minutes. Guideline: 5 minutes.