Intro Experimental Physics Python Guidelines

Style Guidelines

- The formal style manual is called PEP-08. It may be difficult to follow if you don't already know Python or how to code, but please try to follow it as best you can.
- Each function and module you write should have a docstring. Please use the numpy documentation format which extends the PEP-257 standard.
- **Your code must follow style conventions and be readable to receive full credit.** Code that is unreadable or too difficult to parse may be assigned a grade of 0 at your TA's discretion.
- You will be evaluated on how well you decompose your code into reusable functions, how well you handle exceptions or invalid inputs, and the quality of the user interface (where applicable), as well as whether your code produces the "correct" answer.
- Remember that the goal of python is to create ***readable*** code. Ideally, someone else should be able to understand easily what your code does just by reading it.
- One of the major goals of functional decomposition is to create ***reusable*** code.
    - Good documentation is essential.
    - Try to make your functions as general as possible.
        - Do not hard-code in special values.
        - Use default arguments rather than putting default values directly into the code.
    - Give constants descriptive names in ALLCAPS and define them at the top of your module, or in a separate constants module.
    - Consider and test for cases that might cause your function to fail, even if they won't happen in the current program you're writing.

Course-specific requirements

- Please begin each module (.py file) with a docstring that includes as its final lines
    - Your name
    - Your TA's name
    - The assignment number (e.g. PS0)
    - If necessary, the parts of the assignment the .py completes.

Resources:

- MIT 6.001 OCW course website
- GitHub guides
- GitHub help
- Python.org
- numpy reference manual
- David Pine's python manual
- Stack Overflow
- Your TAs
- Google
- A different kind of Python sketch