

Midterm: Find, Complete, & Extend a Tutorial

CSC 206.01: Python Programming

Final Due Date: October 31, 2012 at 6:10 PM

1 Overview

You will be finding an on-line tutorial for a Python module or topic that interests you that you will share with the class. After locating the topic of interest you should complete the tutorial, documenting any problems you encountered. While working through it, think of things that you think could be improved: (better organization, more features, added documentation, etc.). Make a copy of your code and try to make these changes. Once you have finished, write up the general functionality of the program, your experience, and what improvements you made (or were unable to figure out how to make) in a 1 page paper. This will all be shared on iLearn and your peers will try out the tutorial you shared.

2 Tutorial Requirements

- The tutorial should be a tutorial of a Python library, framework, or program **NOT** a general Python language tutorial.
- In general the resultant code of the Tutorial should be approximately 75–100 lines. Less (or more) is fine, but check with me first.
- **PLEASE** do not just copy and paste the tutorial. Rewriting will be very beneficial and make explaining it easier. Also, that will be considered cheating.
- If you have something more interesting that you would like to try, that is also fine, just tell me what it is first
- Some examples:
 - A Python Reddit Clone: <http://genshi.edgewall.org/wiki/GenshiTutorial>
 - Anything from chapter 3 onward: <http://inventwithpython.com/pygame/chapters/>
 - Many of the GUI examples from here: <http://zetcode.com/gui/tkinter/>
 - Many of the GUI examples from here: <http://zetcode.com/tutorials/pyqt4/>
 - This kind of thing if you make good improvements: <http://www.cs.usfca.edu/~fedosov/qttut/>
 - If you are into Math, implementing an algorithm in NumPy
 - Just about anything found by the Google query "`INSERT TOPIC HERE`, Python tutorial"
- Remember that you are **NOT** expected to understand all the details of this at first glance. That is the point of the tutorial

3 Writeup Requirements

- Approximately 1 page with 12pt font and 1.5 line spacing
- Must cover:
 - Where you found the tutorial
 - What is the tutorial teaching: both the name of the library/framework and what it is practically used for
 - Any issues you ran into while completing the tutorial
 - One or two paragraphs about how the code works (at a high level)
 - Anything interesting you were introduced to (or anything you didn't like)
 - what changes you made to improve the code
- This is the important part of the assignment; it proves to me that you actually did it.
- I don't grade your grammar, but make an effort. Read your paper before you submit it.

4 Improvement Requirements

- Make a duplicate copy of your tutorial code to make these changes, DON'T make them in the original copy.
- The changes you make will obviously depend on your skill level but this is the fun part, do as much as you can.
- **For beginning students** (much) improved documentation and maybe a few minor code changes would be sufficient
- **For intermediate students** reorganization of code into smaller, more logical blocks (or a class) would be sufficient.
- **For advanced/ambitious students** adding new features and functionality is ideal, will certainly be the most fun.
- Remember to very clearly document what changes decided to make, and what your motivations were.

5 Peer Review Requirements

- Read a few of your peer's assignments (or just the summaries) that are on a different topic (or tutorial) than your mid term.
- Choose one that seems interesting
- Complete the tutorial
- Write a 1-2 paragraph response to the tutorial in the form. Also include your thoughts on the improvements made by the student.