**Homework 2, Problem 3, HTML Testing**

**Homework 2, Problem 3, HTML Testing**

**Homework 2, CSc 473, Web Site Design, Spring 2014**

|  |  |
| --- | --- |
| **Instructor** | Michael Grossberg |

In this program you are going to work on building your own robot to check html code. The code you will need is in the repository<https://bitbucket.org/mdogy/html_samples_csc473s13hw2> . In this problem you should become comfortable with the unittest library, beautifulsoup4, and the regular expression library (again). I am going to give you some files to start. First of all there are five sample files:

* [hw2p3\_ex1.html](http://bitbucket.org/mdogy/html_samples_csc473s13hw2/raw/tip/hw2p3_ex1.html)
* [hw2p3\_ex2.html](http://bitbucket.org/mdogy/html_samples_csc473s13hw2/raw/tip/hw2p3_ex2.html)
* [hw2p3\_ex3.html](http://bitbucket.org/mdogy/html_samples_csc473s13hw2/raw/tip/hw2p3_ex3.html)
* [hw2p3\_ex4.html](http://bitbucket.org/mdogy/html_samples_csc473s13hw2/raw/tip/hw2p3_ex4.html)
* [hw2p3\_ex5.html](http://bitbucket.org/mdogy/html_samples_csc473s13hw2/raw/tip/hw2p3_ex5.html)

You do *NOT* need to download these directly. I am providing you with some code that will download one of these files and save it as "index.html", then will run tests (which you need to fill in) on that file. The tests should pass on ex5 and fail to varing degrees on the other files. The file which automatically downloads and runs test code is called [switch\_n\_test.py](https://bitbucket.org/mdogy/html_samples_csc473s13hw2/raw/tip/switch_n_test.py) while a tempate for the test code you need to write is [test\_hw2p3.py](https://bitbucket.org/mdogy/html_samples_csc473s13hw2/raw/tip/test_hw2p3.py) . Create a personal repo with the files switch\_n\_test.py and test\_hw2p3.py. Then edit test\_hw2p3.py so that it tests the html to meet the requirements. **DO NOT** change the name of the tests or delete any tests. Assuming you have nosetests installed (which you should), and beautifulsoup4, you should be able to run your tests using the command "python switch\_n\_test.py 5". When you have all the right tests "python switch\_n\_test.py 5" should still pass all the tests while the others, "python switch\_n\_test.py 1" for example, will fail some or all of the tests.

When the robot is up you will be able to see which tests I think should pass and which should fail for a given problem. You will get a 0 if I catch you gaming the tests (regardless of what the robot gives you). You MAY add tests for your own fun but they will be ignored.

Use unittest self.assertTrue, self.assertEqual and the other unittest functions. Tests fail by raise exceptions not by "return True/False" or printing anything. If the test does not raise an exception, the test assumes to pass. Having tests pass all the time is about as good as having a fuel gauge that always shows "full" or a fire alarm that doesn't go off even if there is smoke and intense heat.

Make a repo for this on bit bucket and the link is the answer. When the robot is up past a link to the repo. The only important thing is it contains a modified version of [test\_hw2p3.py](https://bitbucket.org/mdogy/html_samples_csc473s13hw2/raw/tip/test_hw2p3.py) with your code. It does not have to contain switch\_n\_test.py but it can, if you like, along with other files that will be ignored.