**Python 2: Getting More Out of Python  
Lesson 2, Project 1**

Handed in: 19 Feb 2015 06:32:35PM Graded: 19 Feb 2015 09:55:16PM

Here are your instructions:

Make a **UnitTesting\_Homework** project and assign it to the **Python2\_Homework** working set. In this project, write a unittest test program for the following function. (The test program makes unittest.TestCase assertions about the results of calling the function with known arguments.)

def title(s):

"How close is this function to str.title()?"

return s[0].upper()+s[1:]

Test your results for a given string s by comparing them with s.title(). Because this is purely an exercise, it's OK to put your test code in the same file as the function and just hand in a single file. Your file should be an importable module. You should be able to find an example that shows title(s) and s.title() diverge (have different output). Bonus marks for fixing the error in the function above (making it behave more like the native method).

**Your Comment:**

*no comment given*

**Items Handed In**

* [Open Project Handed In](file:///C:\Code\O'Reilly%20School\Python%202\Lesson%202%20-%20Unit%20Testing\project\%3f\.handin\160-7828-2\com.ost.jwoloson.160.7828.2.UnitTesting_Homework.zip)

**Overall Comments:**

Hi Jason,

This is really good - especially your actionable/informative fail message. You're on the right track. Next steps:

- find some strings that pass and others that fail (technically you really only need one of each, but what the heck? strings are cheap). Best practice: make a couple of different tests - one positive and another negative - and run a few strings through each reporting 'success' for expected convergence or expected divergence.

- unless you can make an affirmative case to the contrary, every variable should be local to a method. This prevents inadvertent changes from far-flung bits of the code and keeps things readable because everything's in one place.

Good work so far.

-Pat

**Grade:**

Try Again

Your instructor would like you to work on this project. Be sure to hand in your corrections!

Re: Email I sent to him for attempt 2

Hi Jason, Thanks for the note. Your fail message is really good. It's informative, actionable, and written in a way that's easy to recycle should you want to use it in a loop containing things to test. Generally, in tests readability is king and should trump other valid virtues like parsimony or efficiency. Tests are generally not part of production code so don't run all the time. What's precious here is human time and energy. Making a message like you have will create an efficient environment for the humans attempting to understand/debug the code in the main app. I provided some additional comments in your recent submission. Have a great night. Best, - Pat