PyUnit

PyUnit is the standard unit testing framework for python.

It can be used to easily create test cases to test programs.

In order to use PyUnit, you must first create the classes you wish to test, and then create either additional classes or methods you will use for testing (I prefer to use methods so that is how I will be describing the process)

```
def adder(self,x,y):
    c = x+y
    return c

def TestAdder(self):
    self.failUnless (self.adder(1,8)==9, '1 + 8 != 9!')
    self.failIf(self.adder(16,765)!= 781)
    assert self.adder(3,2) == 5, 'math went wrong'
```

I have created a class called MathTestCase which contains a function called adder. This is a very simple function that adds 2 numbers, and TestAdder is the method I created to test it. This method checks that my adder function will return 9 when 1 and 8 are added together, otherwise it will fail. It also fails if 16 + 765 is not equal to 781 and finally if 3 +2! = 5. Assert and failUnless will fail if the condition is not true while failIf will fail if the condition is true.

The syntax for the failUnless and failIf methods is failUnless(condition, message) where the message will be printed out if the test case fails.

If any of these three statements fails the test itself will fail.

```
.
----------------------------------------------------------------------
Ran 1 test in 0.010s

OK
```

The period indicates none of the tests failed.

Running a test case:

PyUnit has a simple way to execute test cases, that being TextTestRunner.

After defining test cases you can simply create a TextTestRunner instance and define an instance of the class you are testing.

```
>>> runner = unittest.TextTestRunner()
>>> test1 = MathTestCase("TestAdder")
>>> runner.run(test1)
Setting up test cases
Cleaning up test cases
.
----------------------------------------------------------------------
Ran 1 test in 0.060s

OK
<unittest.runner.TextTestResult run=1 errors=0 failures=0>
```

Here, runner is the instance of TextTestRunner, test1 is my test case and the third line is having runner test the predefined case.

Running multiple test cases:

PyUnit includes test suites, which is the method to run multiple test cases at once. In order to do this you define a TextTestRunner in the same way, but instead of a specific test case you create a TestSuite object.

This can be done within a function to make calling easier.

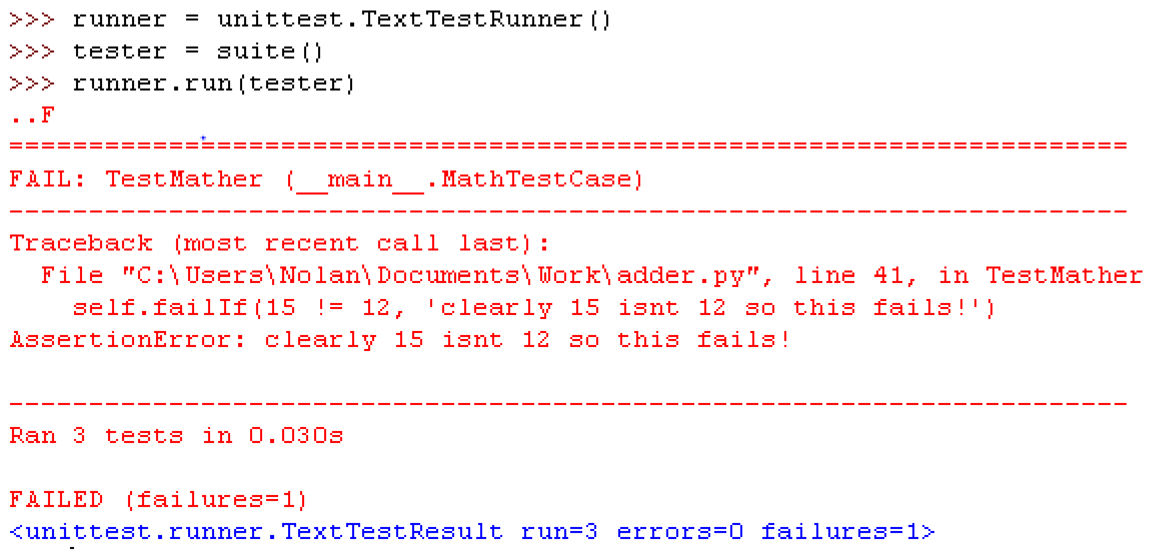
```
def suite():

    suite = unittest.TestSuite()
    suite.addTest(MathTestCase("TestAdder"))
    suite.addTest(MathTestCase("TestSubber"))
    suite.addTest(MathTestCase("TestMather"))

    return suite
```

This has created a TestSuite object named suite that contains 3 test cases, as defined by the methods TestAdder, TestSubber and TestMather.

In my repository I have all the code used for the test cases.



```
>>> runner = unittest.TextTestRunner()
>>> tester = suite()
>>> runner.run(tester)
..F
======================================================================
FAIL: TestMather (__main__.MathTestCase)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\Nolan\Documents\Work\adder.py", line 41, in TestMather
    self.failIf(15 != 12, 'clearly 15 isnt 12 so this fails!')
AssertionError: clearly 15 isnt 12 so this fails!

----------------------------------------------------------------------
Ran 3 tests in 0.030s

FAILED (failures=1)
<unittest.runner.TextTestResult run=3 errors=0 failures=1>
```

In this case runner ran the suite containing 3 test cases, each of which had 3 specific tests to be run. One of these tests failed, so the overall suite failed even though 8 of the 9 tests passed.

Notice the **..F** which shows the first 2 test cases passed while the third failed.

References (places to go if you want more details)

http://pyunit.sourceforge.net/pyunit.html

http://blogs.wrox.com/article/python-test-cases-and-test-suites/