Cheatography

Unit testing in Python Cheat Sheet

by metamad via cheatography.com/1415/cs/468/

Unit testing/TDD basics

Unit testing is an Agile/Continuous Integration principle

Write tests *before* or *during* coding

Writing tests before == detailed requirements

Writing tests during == no over-coding

Unit testing assures code pre- and post-refactoring behaves the same

Passed unit tests == CYA

Shared unit tests during sprints, for example, make it less likely that commits would break others' code

A test case:

* answers a single question

* runs by itself - automation

* determines by itself whether pass or fail (i.e. no human involvement)

* runs in isolation from other test cases

http://www.diveintopython.net/unit_testing/diving_in.html

Unit testing in Python

my_test_thing.py:

```
import unittest

# inherit from TestCase
# (a TestCase is a test fixture)
class MyTest(unittest.TestCase):
.....# method name starts with "test"!
.....def testMethod(self):
..........self.assertEqual(4,3,"4 not equal to 3")

if __name__ == "__main__":
.....# called when exec from CL
.....unittest.main()
```

----

calling from CL:

python my_test_thing.py

prints

```
FAIL: testMethod (__main__.MyTest)
----------------------------------
Traceback (most recent call last):
File "my_test_thing.py", line 6,
in testMethod
self.assertEqual(10,11,"10 != 11")
AssertionError: 10 != 11
--------------------------------
Ran 1 test in 0.000s
FAILED (failures=1)
```

indicates tab: "....."

Cheatographer

metamad
cheatography.com/metamad/

Cheat Sheet

This cheat sheet was published on 19th July, 2012 and was last updated on 24th September, 2012.

Sponsor

FeedbackFair, increase your conversion rate today!
Try it free!
http://www.FeedbackFair.com