PyUnit – Python Testing Framework

Kurt Schmidt
Dept. of Computer Science, Drexel University

May 30, 2018

Examples from these slides can be found in

tux: kschmidt/public_html/CS265/Labs/Python/PyUnit
Intro
These slides assume you’ve read a bit about testing

Testing harness for Python

Tests (python functions) are gathered in `unittest.TestCase`, a class (suite)

Test can be run from inside the script, or, from the command line
Some concepts:

- **test fixture**: Preparation needed to perform one or more tests
- **test case**: Individual unit of testing
- **test suite**: Collection, containing test cases and/or other suites
- **test runner**: Orchestrates the execution of tests. Displays results to the user
Quick Tour
Example

class SimpleTest( unittest.TestCase ) :

    def test_refs( self ) :
        l = [ 'a', 'b', 'c' ]
        m = [ 1, l, 3 ]
        self.assertIs( m[1], l )

    def test_isupper( self ) :
        self.assertTrue( 'FOO'.isupper() )
        self.assertFalse( 'Foo'.isupper() )

    def test_split( self ) :
        s = 'hello world'
        self.assertEqual( s.split(), ['hello', 'world'] )
        # check that s.split fails when the separator is not a string
        with self.assertRaises( TypeError, "Call Underdog!" ) :
            s.split( 2 )

if __name__ == '__main__' :
    sys.argv.append( '-v' )
    unittest.main()
Quick Description

- Note, subclass of `unittest.TestCase`
- Test methods begin with the string `test`
- `TestCase` provides various `assert...` methods
- Invoke `unittest.main()` to automatically run tests
- Invoke your script with `-v` to get a bit more output.
test_isupper (__main__.SimpleTest) ... ok
test_ref (__main__.SimpleTest) ... ok
test_split (__main__.SimpleTest) ... ok

Ran 3 tests in 0.000s

OK
Assertions
assert... Methods

These are some common ones, and have negated forms also

assertTrue( x )
assertFalse( x )
assertEqual( a, b ) a == b
assertIs( a, b ) a is b
assertNone( x ) x is None
assertIn( a, b ) a in b

There are more, but these will get you started.
We can also check that exceptions are thrown

```
assertRaises( exc, f, *args, **kwargs ) Checks that
f( *args, **kwargs ) throws an exception
```

```
self.assertRaises( Money.BadUnitsException, Money.__add__, u2, p1 )
# Can’t add dollars to Polish zloty
```

Also now comes as a context manage, taking an optional message (shown above).
The Test Fixture
Set-up Methods

**SetUp**  Called before each test

**TearDown**  Called after each test, assuming *SetUp* succeeded
Invocation
Running your tests

- Call `unittest.main()`, as shown in the example
- Or you can pass your test class(es) directly to the module:

```
$ python3 -m unittest test_money
```