Top of Form 1

Bottom of Form 1

Top of Form 2

**Python 2: Getting More Out of Python  
Lesson 3, Quiz 2**

Handed in: 20 Feb 2015 10:32:56PM Graded: 21 Feb 2015 10:46:20AM

Question 1:

What would be the best methods to use to verify the result of a floating-point computation during a test?

**Your Answer:**

I would use the assertRaises and the assertIsInstance methods:

#Some code left out for efficiency

#create divider function; would normally be in its own

#separate module

def divider(x, y):

return float(x)/y

#import divider if in separate module

#from divider import divider

class Divide\_and\_Conquer(unittest.TestCase):

def test\_1(self):

a = divider(8,2)

self.assertRaises(TypeError,

a)

def test\_2(self):

a = divider(10,5)

self.assertIsInstance(a,

float)

if \_\_name\_\_ == "\_\_main\_\_":

unittest.main()

**Mentor Comments:**

Nope. The last few bits of a floating point number can be garbage. Try this for fun:

from math import pi

original = pi

for i in range (10):

original = original \* 12123.3213423452345

original = original / 12123.3213423452345

if original != pi:

print("blew up on iteration %s" %str(i))

Here, you're looking for a couple of test methods designed especially to look for these near misses which don't necessarily represent problems with code is much as they do inherent inaccuracies in floating-point computations.

Here's a "quick and dirty" way to have a look at all of the test methods. A few of these are deprecated, and others are intended to serve as wrappers (assertEqual is one) but you can find out about in unfamiliar method easily enough by using help ():

>>> import unittest

>>> for name in dir(unittest.TestCase):

... if name[0]!='\_': #don't print internal elements

... print(name)

...

addCleanup

addTypeEqualityFunc

Question 2:

Which function from the tempfile module would you use to create a new temporary directory?

**Your Answer:**

the mkdtemp(path) function within the tempfile module.

**Mentor Comments:**

Nope. The last few bits of a floating point number can be garbage. Try this for fun:

from math import pi

original = pi

for i in range (10):

original = original \* 12123.3213423452345

original = original / 12123.3213423452345

if original != pi:

print("blew up on iteration %s" %str(i))

Here, you're looking for a couple of test methods designed especially to look for these near misses which don't necessarily represent problems with code is much as they do inherent inaccuracies in floating-point computations.

Here's a "quick and dirty" way to have a look at all of the test methods. A few of these are deprecated, and others are intended to serve as wrappers (assertEqual is one) but you can find out about in unfamiliar method easily enough by using help ():

>>> import unittest

>>> for name in dir(unittest.TestCase):

... if name[0]!='\_': #don't print internal elements

... print(name)

...

addCleanup

addTypeEqualityFunc

Question 3:

Which function from the shutil module would you use to delete a directory including all files and subdirectories in it?

**Your Answer:**

the rmtree(path) function within the shutil module.

**Mentor Comments:**

Yes. Be careful with this one. It does \*exactly\* what you ask of it - quickly and silently. One of my students recently wiped out his entire homework directory on accident :-(

Question 4:

What is the first step in test-driven development?

**Your Answer:**

Write tests!

**Mentor Comments:**

*none*

**Overall Comments:**

Hi Jason,

These are all good except for Q1. Could you please have a look at the comments and try again?

-Pat

**Grade:**

Try Again

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

[Take Me Back](ostreturn:) [Next Attempt](file:///C:\Code\O'Reilly%20School\Python%202\Lesson%203%20-%20TDD\viewassignment.php%3fentryid=7843&attempt=2&graded=1)

© 2014, O'Reilly Media, Inc. All rights reserved.

Bottom of Form 2