Top of Form



Bottom of Form

Top of Form



# Python 3: The Python Environment Lesson 1, Project 1

Handed in: 1 Jun 2015 06:14:22PM Graded: 4 Jun 2015 09:44:23PM

**Here are your instructions:**

Create a **Python3\_Homework01** project and assign it to your **Python3\_Homework** working set. In the **Python3\_Homework01/src** folder, create a program named **adder.py**; in it, create a function that takes two objects and adds them together only if they are both of the integer type. Raise a TypeError otherwise. Then, create a **test\_adder.py** file that tests the correctness of this function.

When they are working to your satisfaction, submit **adder.py** and **test\_adder.py**.

##### Your Comment:

Hello Again Kirby!

I'm back for Round 3. This works except for the commented out code in test\_adder.py (line 36).

When I pass a function as the first argument, the adder function correctly identifies that it is

expecting integers, but my testing suite fails even though in the testing suite I specify that it should

expect to encounter an AssertionError. Thoughts?

-Jason

##### Items Handed In

* [Open Project Handed In](https://students.oreillyschool.com/student/project/?/.handin/147-6538-1/com.ost.jwoloson.147.6538.1.Python3_Homework01.zip)

### Overall Comments:

We haven't really covered with syntax (we will in Python4). There's another way to write

asserts:

self.assertRaises(TypeError, adder, x, y) # <--- better for now

It's a TypeError you should practice raising explicitly, as in: raise TypeError

Do that if either argument is not an int. type(x) == int. isinstance(x, int) both good.

If either x or y not int, raise TypeError.

Then test with multiple assertRaises calls where of special interest are x, y "close to int".

-Kirby

### Grade:

Try Again

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

[Take Me Back](ostreturn:)

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

Bottom of Form