Course   Progress   Dates   Discussion   Instructional Team   Office Hours

🏠 Course / Lesson 1: Advanced Testing / Lesson 1 Content

Previous   Next

Part 2: Why Test?

🔖 Bookmark this page

Part 2: Why Test?

It's likely you already have some familiarity with software testing in general, and writing tests in Python specifically. In this section, we're going to revisit the basic question: why write software tests? Our discussion here will form the foundation of the rest of the lesson.

It's highly recommended for you to type along with these examples, creating the squarer.py file and the corresponding test files!

A Simple Example

Suppose that you have written a simple class for squaring numbers:

```
# squarer.py
class Squarer(object):

    @staticmethod
    def calc(operand):
        return operand*operand
```

Squarer is a simple class, with a single static method: calc. The calc method calculates the square of its argument in the usual way, and returns it to the caller.

Let's use the Squarer class from the REPL (the Python shell):

```
$ python -i
>>> from squarer import Squarer
>>> Squarer.calc(3)
    9
>>> Squarer.calc(100)
    10000
```

Squarer works well: squaring 3 yields 9. Squaring 100 yields 10,000. The Squarer class is working perfectly.

Now imagine that another engineer working on this project has a "bright idea" about how to improve the Squarer class: squaring involves powers, therefore the Squarer class should be using the power operator. They make the following modification to your code:

```
# squarer.py
class Squarer(object):

    @staticmethod
    def calc(operand):
        # return operand*operand
        # OLD return operand**operand
```

Then at some point later you're using the squarer class to square numbers:

```
$ python -i
>>> from squarer import Squarer
>>> Squarer.calc(3)
    27
>>> Squarer.calc(100)

100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000
```

The square of 3 is definitely not 27. And squaring 100 should give 10,000, not
1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

W

Help Center   Contact Us   Privacy   Terms

Built on OPENedX by RACCOONGANG