Programming fundamentals with Python

Pepe García

2020-04-20

Programming fundamentals with Python

Plan for today

Learn the principles behind software testing

Install spyder-unittest plugin

Do some testing together!

Testing

Software testing is the process & techniques that try to ensure the quality of software

Manual testing

Manual testing is the kind of testing performed by humans manually,
without any automation.

Since it requires a human doing the work by hand, it's more expensive than automated testing. However, it's a way of testing that it's still very much used for example, in the video game industry.

Automated testing

Automated testing relies on creating code that tests your code automatically

Having an automated test suite helps a lot when developing, since we can notice when new code introduces bugs

Levels of test

unit testing: test how **function** outputs change when their inputs change. This is the simplest form of testing, yet one of the most effective. Used to see if **the function works**.

Functional testing: In functional testing one checks the functional specifications of a system (composed by different **modules and functions**) and check that they hold. Created to see that the **current module works**.

Integration testing: individual components are combined and tested as a group to confirm that the **overall system** works.

Unit testing

We create different **test cases** for testing different **behaviors** of our program

Expectations we make about our code are called **assertions**

Unit testing

```
def is_prime(number):
    for element in range(2, number):
        if number % element == 0:
            return False
    return True
```

Try to think on some tests for this function:

Unit testing

```
from primes import is_prime

def test_zero_is_prime():
    assert is_prime(0) == True

def test_five_is_prime():
    assert is_prime(5) == True

def test_four_is_not_prime():
    assert is_prime(4) == False
```

import the module under test

handle edge cases

handle normal cases

pytest

pytest is the library we're going to use for testing.

https://pytest.org

Installing pytest

We can install pytest with **pip3:**

```
$ pip3 install pytest
```

Disclaimer: Catalina changed a lot of stuff on MacOS, making the previous command to fail.

*try **pip3 install pytest --user** if it fails*

running pytest

For running all python tests in a folder, we should navigate to that folder first, and then run **python3 -m pytest**

```
$ cd the_folder_in_which_i_have_all_the_tests
$ python -m pytest

========================== test session starts ==========================
platform darwin -- Python 3.7.3, pytest-5.2.4, py-1.8.0, pluggy-0.13.0
rootdir: /Users/pepe/projects/pepegar/leetcode
collected 1 item

test_addtwonumbers.py F                                           [100%]

============================== FAILURES ==============================
_____________________________ test_whatever _____________________________
```

Creating tests

for **pytest** to find your tests, they need to be in files named **test_****.py**, and test functions should be named with the **test_** prefix too

```
def test_whatever():
    assert 3 == 4
```

test_something.py

FizzBuzz!

Recap

Software testing ensures quality of our code

With unit testing, we can check that our functions work as we expect

Do some testing together!

Exercises

Througout all our exercises we will model and test an ecommerce business. We will be adding new functionality with each different belt.

Please solve the problems incrementally.

Remember to create a new project in spyder to make easier to setup testing and git. Create a repository in Github with your final solution to the exercise.

Exercise 1

Our e-shop sells the following products:

1. **Guitar**: $1000
2. **Pick box**: $5
3. **Guitar Strings**: $10

Create a function named **checkout** that takes a list that represents a shopping cart and returns the total cost of it. This function should check that **the shopping cart must not be empty**.

Create also some tests for the function. Try to think of the corner cases.

Hint: you can represent a product as a dictionary with a name and a price.

Exercise 2

You want to give more features to the user, so you decide that you will allow them to purchase an insurance package on their purchase and also priority mail. Consider that these two new services can only be purchase once per order.

1. **Insurance**: $5
2. **Priority mail**: $10

Modify your checkout function so it handles these cases correctly, and add more tests that check your functionality.

Exercise 3

You want to add a new feature to your ecommerce, you want to create three different tiers of customers:

- **normal**: No added benefits
- **silver**: 2% discount on **products** from the ecommerce
- **gold**: 5% discount on everything

Modify the checkout function to accept another parameter with the tier of the customer and apply the discounts as needed.

Implement this feature in the checkout function and add tests that prove that your implementation is correct.