

Codio Activity

PyTest and Test-Driven Development

```
# code source: https://semaphoreci.com/community/tutorials/testing-python-
applications-with-pytest
# wallet.py
class InsufficientAmount(Exception):
    pass

class Wallet(object):
    def __init__(self, initial_amount=0):
        self.balance = initial_amount

    def spend_cash(self, amount):
        if self.balance < amount:
            raise InsufficientAmount('Not enough available to spend
{}'.format(amount))
        self.balance -= amount

    def add_cash(self, amount):
        self.balance += amount
```

To run unit tests in the Python environment (Codio), I installed **pytest** using the following command:

```
python -m pip install pytest
```

The following TDD file was used to exercise the code above:

```
# code source: https://semaphoreci.com/community/tutorials/testing-python-
applications-with-pytest
# test_wallet.py
import pytest
from wallet import Wallet, InsufficientAmount

def test_default_initial_amount():
    wallet = Wallet()
    assert wallet.balance == 0

def test_setting_initial_amount():
    wallet = Wallet(100)
    assert wallet.balance == 100
```

```
def test_wallet_add_cash():
    wallet = Wallet(10)
    wallet.add_cash(90)
    assert wallet.balance == 100

def test_wallet_spend_cash():
    wallet = Wallet(20)
    wallet.spend_cash(10)
    assert wallet.balance == 10

def test_wallet_spend_cash_raises_exception_on_insufficient_amount():
    wallet = Wallet()
    with pytest.raises(InsufficientAmount):
        wallet.spend_cash(100)
```

Then to run the tests, I used the command shown below. The outcome is that all five of the unit tests passed on first run.

```
codio@scarletballet-totembiology:~/workspace$ pytest -q test_wallet.py
.....
5 passed in 0.05 seconds
codio@scarletballet-totembiology:~/workspace$ pytest -q test_wallet.py
.FFF.
```

Next, it was time to modify the test code to ensure the tests fail:

```
# code source: https://semaphoreci.com/community/tutorials/testing-python-
applications-with-pytest
# test_wallet.py
import pytest
from wallet import Wallet, InsufficientAmount

def test_default_initial_amount():
    wallet = Wallet()
    assert wallet.balance == 0

def test_setting_initial_amount():
    wallet = Wallet(1200)
    assert wallet.balance == 100

def test_wallet_add_cash():
    wallet = Wallet(10)
    wallet.add_cash(85)
    assert wallet.balance == 100
```

```
def test_wallet_spend_cash():
    wallet = Wallet(20)
    wallet.spend_cash(0)
    assert wallet.balance == 10

def test_wallet_spend_cash_raises_exception_on_insufficient_amount():
    wallet = Wallet()
    with pytest.raises(InsufficientAmount):
        wallet.spend_cash(100)
```

Now, executing the **pytest** command once more, several of the tests correctly fail, and I can easily view the output to determine the location and reason for the failure:

```
codio@scarletballet-totembiology:~/workspace$ pytest -q test_wallet.py
.FFF.
===== FAILURES =====
----- test_setting_initial_amount -----

    def test_setting_initial_amount():
        wallet = Wallet(1200)
>       assert wallet.balance == 100
E       assert 1200 == 100
E       + where 1200 = <wallet.Wallet object at 0x7f3edac707d0>.balance

test_wallet.py:11: AssertionError
----- test_wallet_add_cash -----

    def test_wallet_add_cash():
        wallet = Wallet(10)
        wallet.add_cash(85)
>       assert wallet.balance == 100
E       assert 95 == 100
E       + where 95 = <wallet.Wallet object at 0x7f3edaecd190>.balance

test_wallet.py:16: AssertionError
----- test_wallet_spend_cash -----

    def test_wallet_spend_cash():
        wallet = Wallet(20)
        wallet.spend_cash(0)
>       assert wallet.balance == 10
E       assert 20 == 10
E       + where 20 = <wallet.Wallet object at 0x7f3edaef1a50>.balance

test_wallet.py:21: AssertionError
3 failed, 2 passed in 0.07 seconds
codio@scarletballet-totembiology:~/workspace$
```