Testing In Python

@igorgue

[we need a logo for the group]
Examples:
https://github.com/igorgue/testing-in-python-examples

What's Testing?

"Verify software correctness automatically, controlling input and output" - me

Unit, Integration and Functional Tests

Unit Test Example

```python
def save_transaction(transaction_id=None, value=None):
    if not value:
        raise KVError('You need a value to store')

    if transaction_id:
        save_to_db(key=transaction_id, value=value)
    else:
        save_to_db(key=None, value=value)
```

```python
def save_transaction(transaction_id=None, value=None):
    if not value:
        raise KVError('You need a value to store')

    if transaction_id:
        save_to_db(key=transaction_id, value=value)
    else:
        save_to_db(key=None, value=value)
```

Integration Tests

```python
def save_transaction(transaction_id=None, value=None):
    if not value:
        raise KVError('You need a value to store')

    if transaction_id:
        save_to_db(key=transaction_id, value=value)
    else:
        save_to_db(key=None, value=value)


def save_to_db(key, value):
    database = get_database()

    if not key:
        key = generate_id()

    database[key] = value
    database.close()
```

```python
def save_transaction(transaction_id=None, value=None):
    if not value:
        raise KVError('You need a value to store')

    if transaction_id:
        save_to_db(key=transaction_id, value=value)
    else:
        save_to_db(key=None, value=value)

def save_to_db(key, value):
    database = get_database()

    if not key:
        key = generate_id()

    database[key] = value
    database.close()
```

Functional Tests

```python
import subprocess

def test_get_transaction():
    assert subprocess.check_output(['./example.py', '1']) == '$0.99\n'
    assert subprocess.check_output(['./example.py', '2']) == '$9.99\n'
    assert subprocess.check_output(['./example.py', '3']) == '$2.99\n'
```

Demo

https://github.com/igorgue/testing-in-python-examples

This testing thing...

Care about code not code coverage

Be careful anyways

Thanks.