Testing in Python 3

Unit Test Vocabulary and Design

Emily Bache

Technical Coach

@emilybache | coding-is-like-cooking.info

Testing in Python 3

Version Check

Version Check

This course was created by using:

- Python 3.11.2
- Pytest 7.2

Version Check

This course is 100% applicable to:

- Python 3
- Any version of pytest from 2019 to 2023

Summary

Unit Testing Vocabulary

Example using ‘unittest’ module

Test Case Design

Unit Testing Fundamentals

A Unit is a Small Piece of Code

A method or function

A module or class

A small group of related classes

An Automated Unit Test

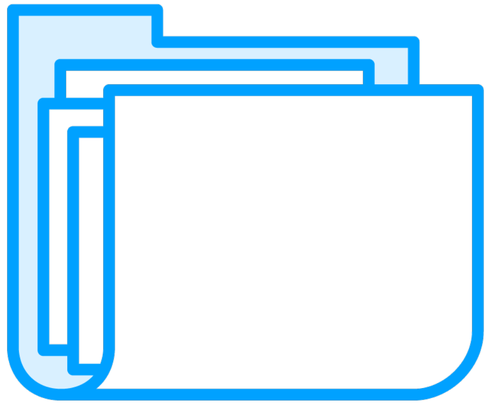
Is designed by a human

Runs without intervention

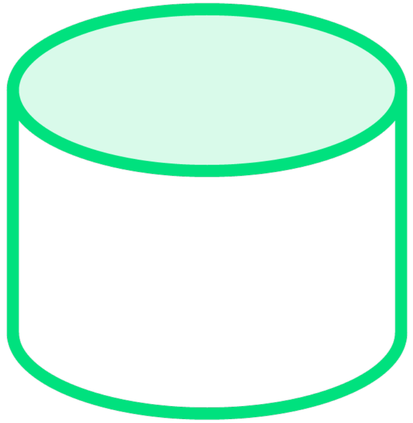
Reports either ‘pass’ or ‘fail’

Strictly Speaking

It’s not a unit test if it uses...



The Filesystem



A Database

The Network

But it might still be a useful test!

Phonebook Demo

Given a list of names and phone numbers

Make a Phonebook

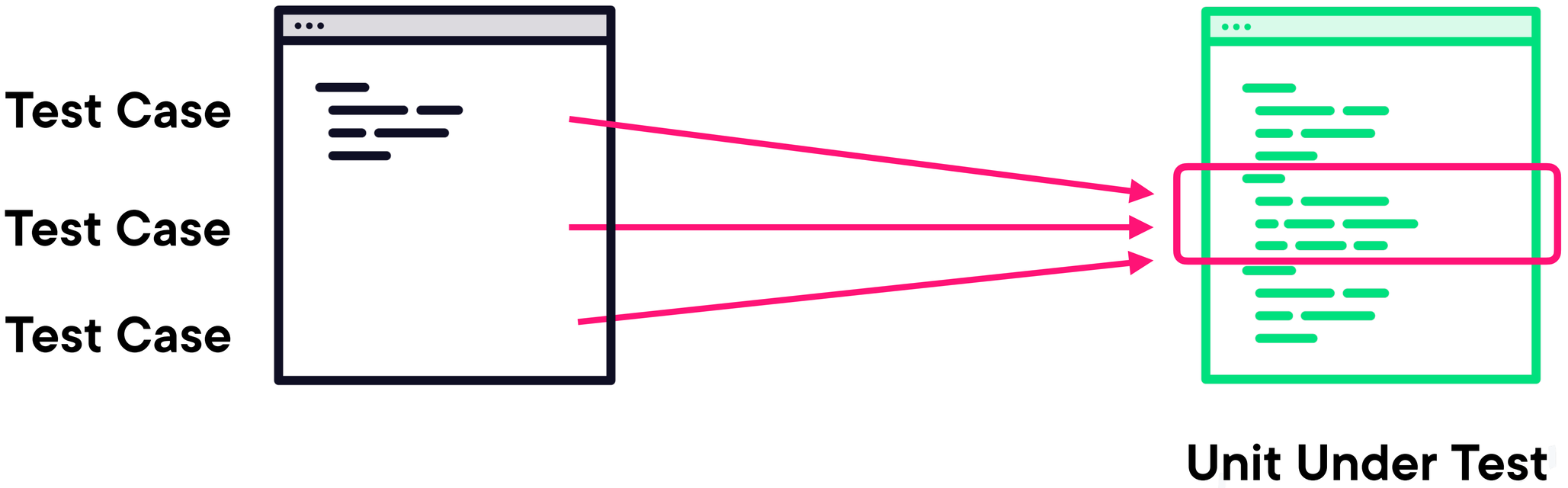
Determine if it is consistent, ie. no number is a prefix of another

- Bob: 91125426, Anna: 97625992
- Emergency: 911
- Bob and Emergency are inconsistent

Phone Numbers Exercise

Create a test case and run it

Unit Test Vocabulary: Test Case



Unit Test Vocabulary: Test Runner

Test Case

Unit Under Test

```
.
----------------------------------------
Ran 1 test in 0.001s

OK
```

Test Runner

Phonebook Exercise

Run same test case in IDE
and terminal

Choosing a Test Runner

Working Interactively
An IDE like PyCharm

Continuous Integration
A Command Line Test Runner

Phone Numbers Exercise

Create another test case

Unit Test Vocabulary: Test Suite

Test Suite

Units Under Test

```
.......
----------------------------------------
Ran 7 tests in 0.000s

OK
```

Test Runner

Phone Numbers Exercise

Skip a test case
Add setUp and tearDown

```
class PhonebookTest(unittest.TestCase):


    def setUp(self) -> None:
        self.phonebook = Phonebook()


    def tearDown(self) -> None:
        pass


    def test_lookup_by_name(self):
        self.phonebook.add("Bob", "12345")
        number = self.phonebook.lookup("Bob")
        self.assertEqual(number, "12345")

    def test_missing_name(self):
        with self.assertRaises(KeyError):
            self.phonebook.lookup("missing")

    def test_empty_phonebook_is_consistent(self):
        is_consistent = \
self.phonebook.is_consistent()
        self.assertTrue(is_consistent)
```

◀ Declare a class containing tests

◀ Set up fixture method

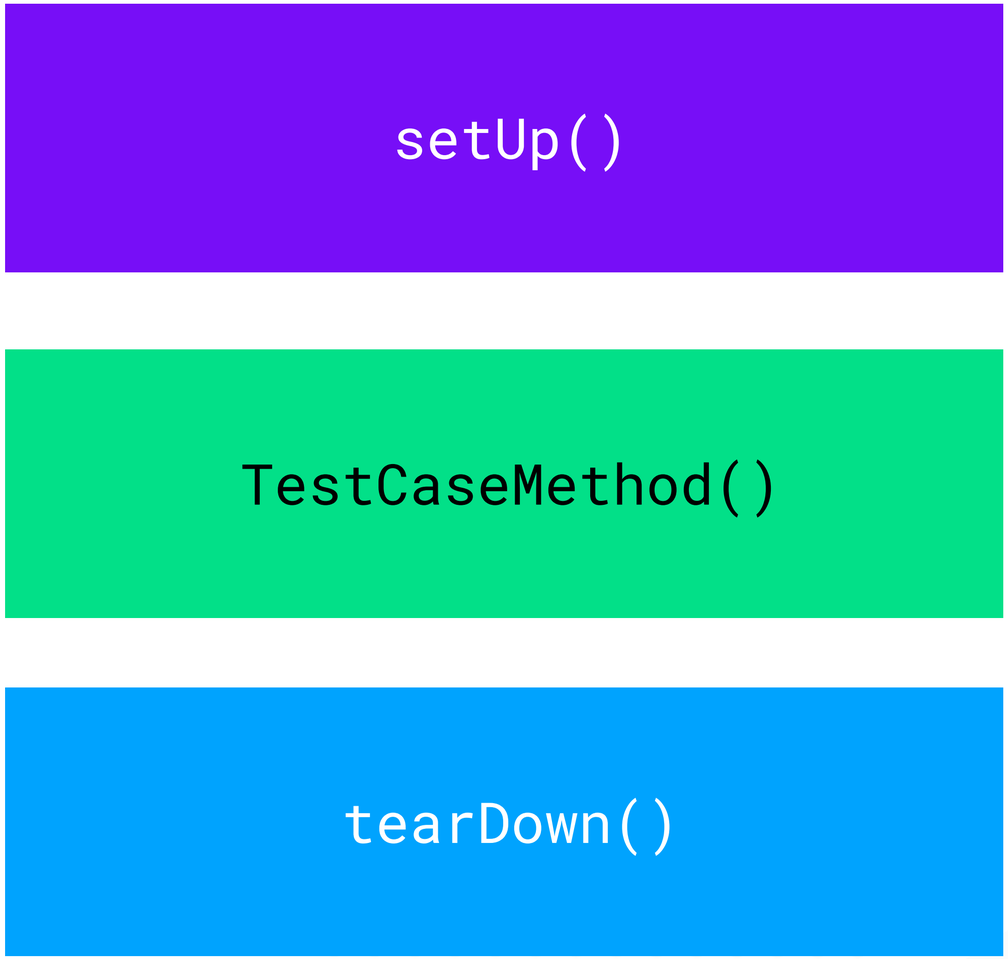
◀ Tear down fixture method

◀ First test case

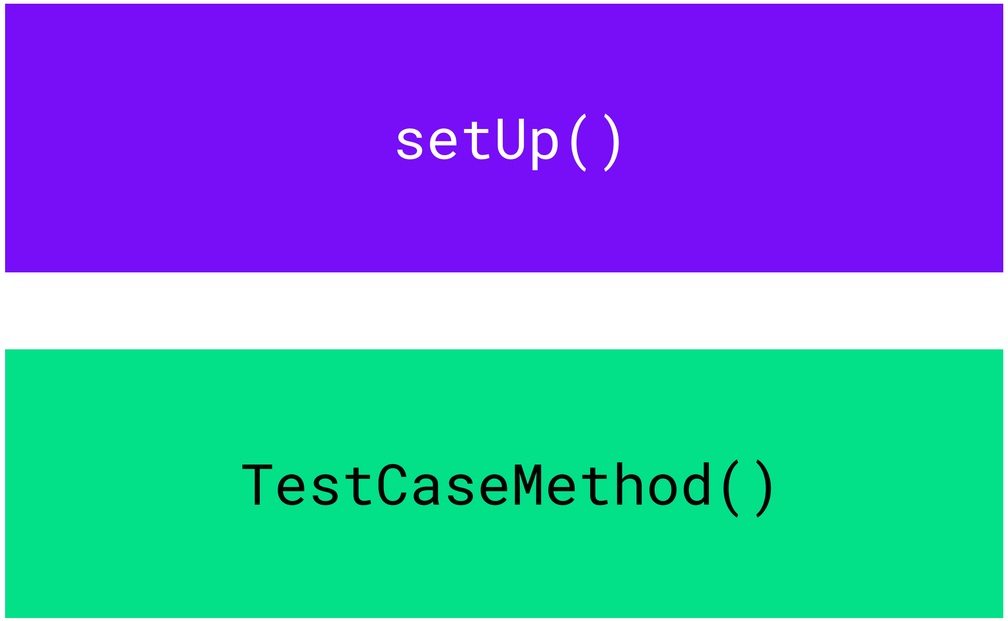
◀ Second test case

◀ Third test case

Test Fixture: Order of Execution



Test Fixture for Strict Unit Tests



Test Fixture: Order of Execution

```
setUp()
```

```
TestCaseMethod()
```

```
tearDown()
```

Test Fixture: Order of Execution

```
setUp()
```

Unit Test Vocabulary

Test Case

```
def setUp (self):
    pass

def tearDown (self):
    pass
```

Test Fixture

Unit Under Test

Test Suite

```
.......
----------------------------------------
Ran 7 tests in 0.000s

OK
```

Test Runner

Phone Numbers Exercise

Example of poor design

Lookup by name

Missing name

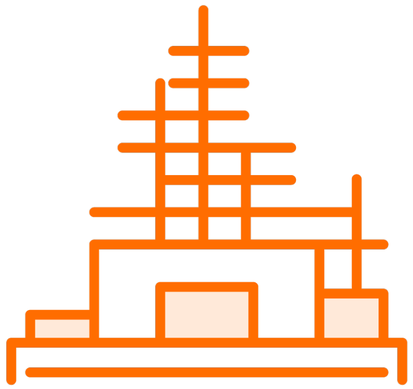
Consistent when empty

Consistent when all different

Inconsistent when duplicates

Inconsistent when duplicates prefix

The Three Parts of a Test



Arrange
Set up the object to be tested, and collaborators

Act
Exercise the unit under test

Assert
Make claims about what happened

```
def test_lookup_by_name(self):
    self.phonebook.add("Bob", "12345")
    number = self.phonebook.lookup("Bob")
    self.assertEqual(number, "12345")
```

- Test Case Name
- Arrange
- Act
- Assert

```
def test_is_consistent(self):
    self.phonebook.add("Bob", "12345")
    self.assertTrue(self.phonebook.is_consistent())
    self.phonebook.add("Anna", "012345")
    self.assertTrue(self.phonebook.is_consistent())
    self.phonebook.add("Sue", "12345")
    self.assertFalse(self.phonebook.is_consistent())
    self.phonebook.add("Sue", "123")
    self.assertFalse(self.phonebook.is_consistent())
```

- Test Case Name
- Arrange
- Act, Assert
- Arrange
- Act, Assert
- Arrange
- Act, Assert
- Arrange
- Act, Assert

Phone Numbers Exercise

Example of better design

Phone Numbers Exercise

show assertEqual, assertTrue, assertFalse

assertRaises

assertIn

show the docs:

https://docs.python.org/3/library/unittest.html

Summary

Vocabulary:

- Test Case
- Test Runner
- Test Suite
- Test Fixture

Test Case Design:

- Test name
- Arrange - Act - Assert