Unit Testing in Python

UNIT TEST FUNDAMENTALS

Emily Bache
TECHNICAL AGILE COACH
@emilybache coding-is-like-cooking.info

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

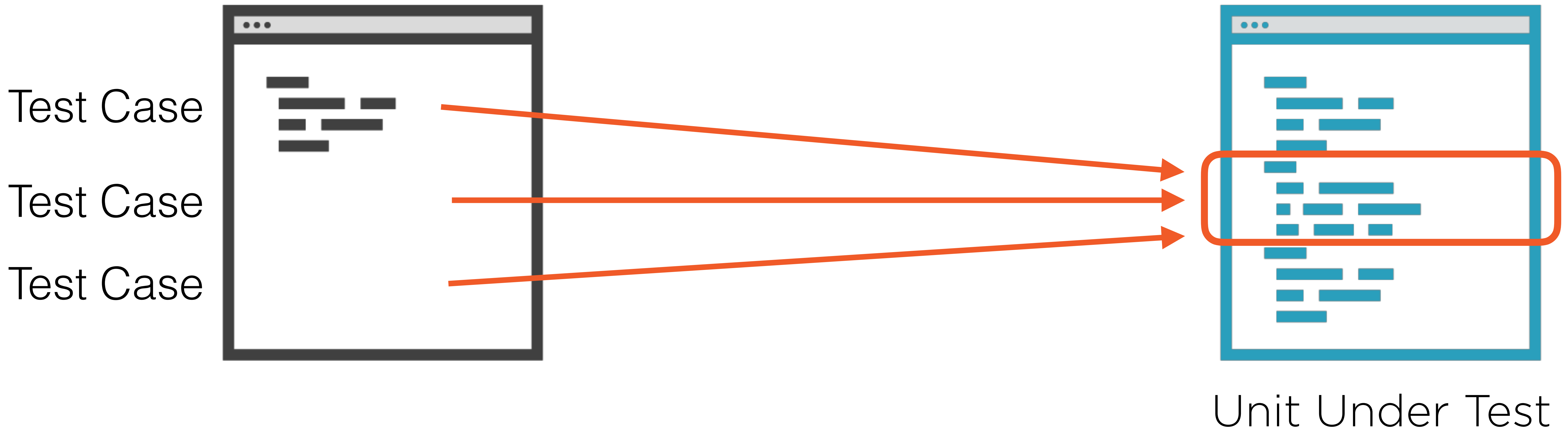
Given a list of names and phone numbers

Make a Phonebook

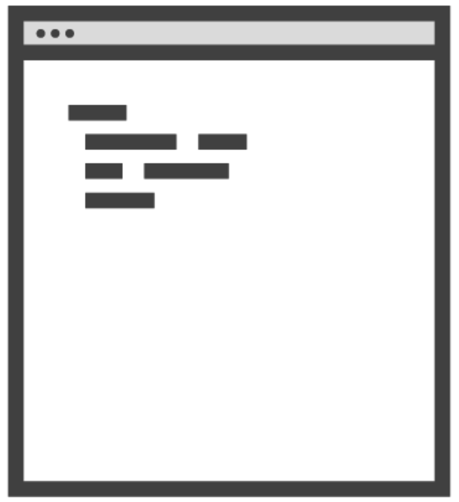
Determine if it is consistent:

- no number is a prefix of another
- e.g. Bob 91125426, Anna 97625992
- Emergency 911
- Bob and Emergency are inconsistent

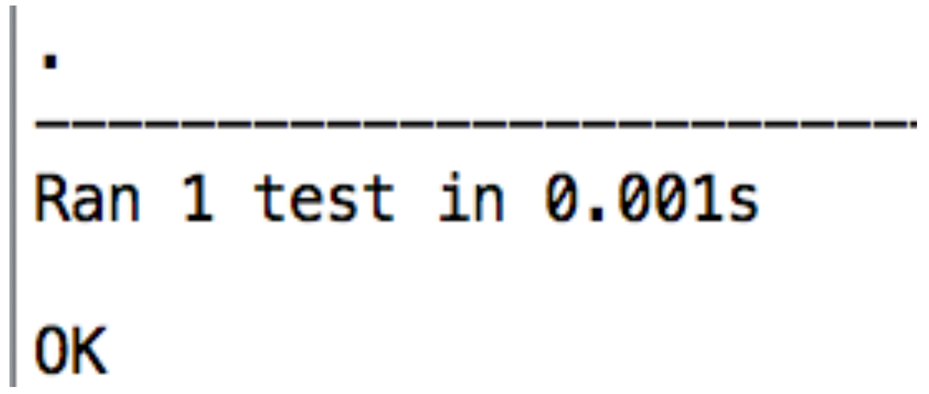
Unit Test Vocabulary: Test Case



Unit Test Vocabulary: Test Runner


Test Case

Unit Under Test

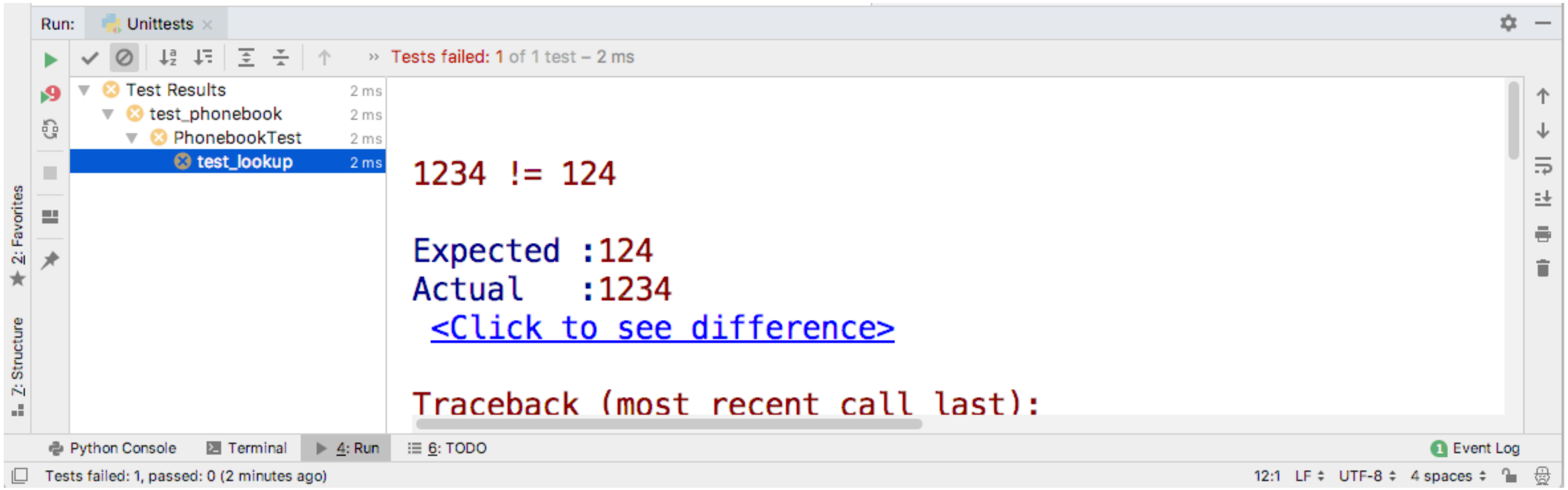


```
.
----------------------------------------------------------------------
Ran 1 test in 0.001s

OK
```

Test Runner

Unit Test Vocabulary: Test Runner



```
Tests failed: 1 of 1 test – 2 ms

Test Results        2 ms
  test_phonebook    2 ms
    PhonebookTest   2 ms
      test_lookup   2 ms
```

```
1234 != 124

Expected :124
Actual   :1234
 <Click to see difference>

Traceback (most recent call last):
```

```
Tests failed: 1, passed: 0 (2 minutes ago)        12:1  LF  UTF-8  4 spaces
```

Test Runner in PyCharm

Choosing a Test Runner

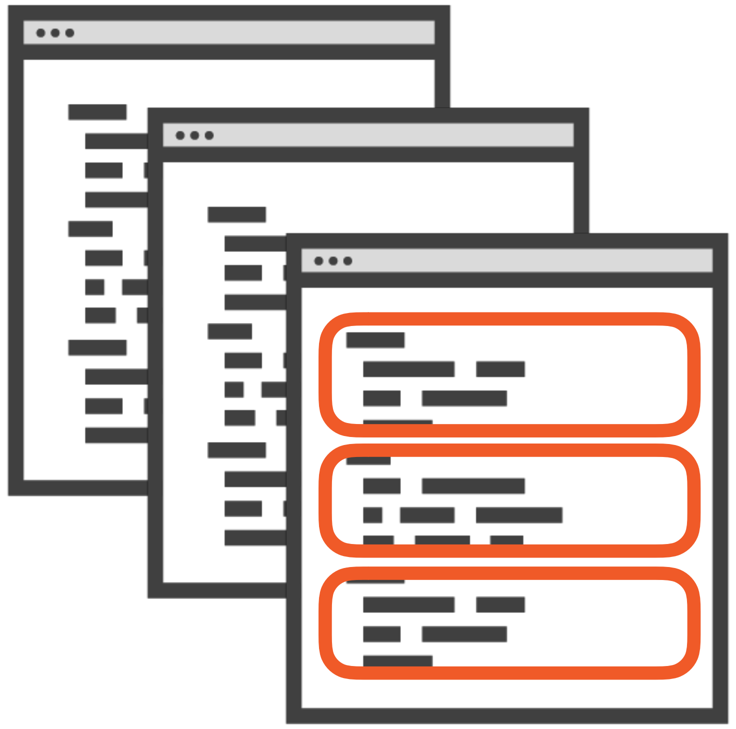
Working Interactively

An IDE like PyCharm

Continuous Integration

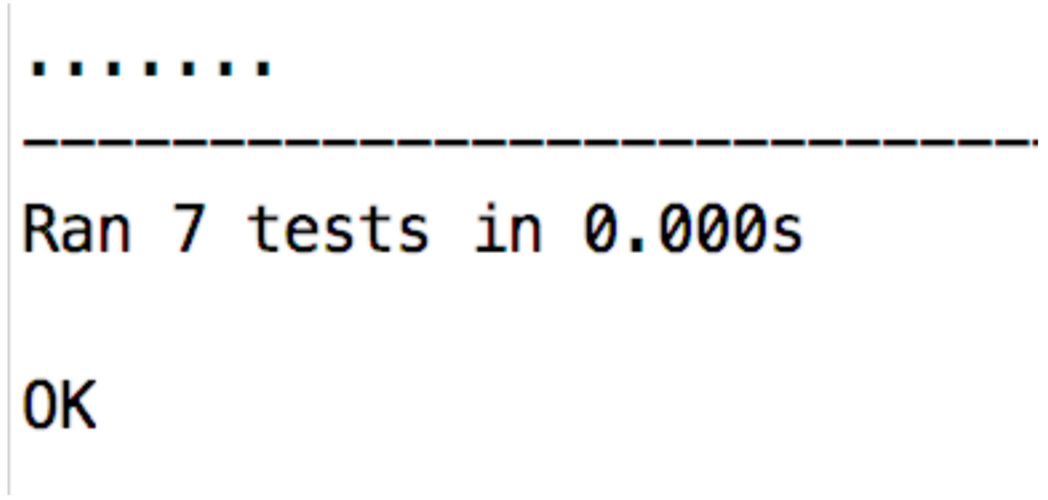
A Command Line Test Runner

Unit Test Vocabulary: Test Suite



Test Suite

Units Under Test



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

OK
```

Test Runner

Test Fixture: Order of Execution

`setUp()`

`TestCaseMethod()`

`tearDown()`

Test Fixture: Order of Execution

`setUp()`

`TestCaseMethod()`

`tearDown()`

Test Fixture: Order of Execution

`setUp()`

Test Fixture for Strict Unit Tests

`setUp()`

`TestCaseMethod()`

```python
class PhoneBookTest(unittest.TestCase):

    def setUp(self):
        self.phonebook = PhoneBook()

    def tearDown(self):
        pass

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

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

◀ Declare a class containing tests

◀ Set up fixture method

◀ Tear down fixture method

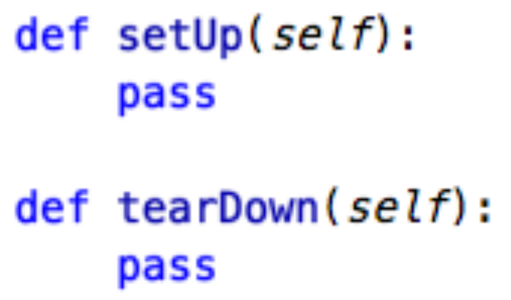
◀ First test case

◀ Second test case

Unit Test Vocabulary

Test Case

Test Suite

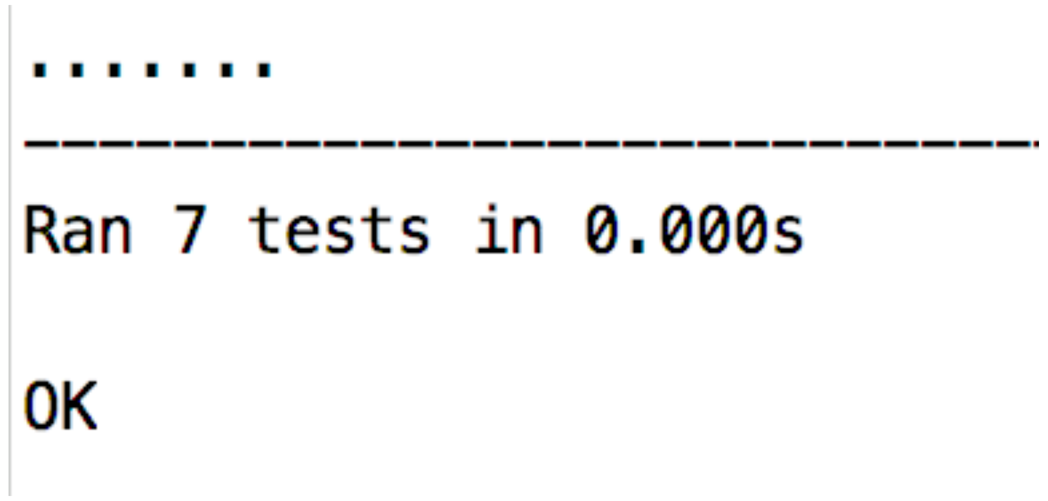


```
def setUp(self):
    pass

def tearDown(self):
    pass
```

Test Fixture

Unit Under Test



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

OK
```

Test Runner

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

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

◀ Test Case Name

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

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

◀ Test Case Name

◀ Arrange

◀ Act

◀ Assert

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

- ◀ Test Case Name
- ◀ Arrange
- ◀ Act
- ◀ Assert

```python
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
- ◀ Act
- ◀ Assert
- ◀ Act
- ◀ Assert
- ◀ Act
- ◀ Assert
- ◀ Act
- ◀ Assert

Summary

Vocabulary:

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

Test Case Design:

- Test name
- Arrange - Act - Assert