# TESTING

99 little bugs in the code.
99 little bugs in the code.
Take one down, patch it around.
127 little bugs in the code...

## TYPES OF TESTING

- Unit testing
- Integration testing
- System testing
- Functional testing
- Stress testing
- Performance testing
- Usability testing
- Acceptance testing

# UNIT TEST FRAMEWORKS

- JUnit (Java)
- unittest (Python)
- Test::Unit, RSpec (Ruby)

# MOCKING FRAMEWORKS

- Java
  - Mockito
    http://mockito.org/
  - EasyMock
    http://easymock.org/
  - JMockit
    http://jmockit.org/

- Python
  - unittest.mock
    http://www.toptal.com/python/an-introduction-to-mocking-in-python
  - mockito-python
    https://code.google.com/p/mockito-python/
- Ruby
  - rspec-mocks
    https://github.com/rspec/rspec-mocks
  - mocha
    http://gofreerange.com/mocha/docs/

```
class Pony:
    string owner;
    string previous_owner;
    int value;

def Pony buy_pony(Pony pony, string buyer, int price):
    if price < pony.value:
        return null;
    pony.previous_owner = pony.owner;
    pony.owner = buyer;
    pony.price = price;
    return db.save(pony);
```

```
def test_buy_pony_higher_price:
    Pony pony = new Pony();
    pony.owner = “you”;
    pony.price = 1000;
    Pony my_pony = market.buy(pony, “me”, 1100);
    assert_not_null(my_pony);
    assert_equal(“me”, my_pony.owner);
    assert_equal(“you”, my_pony.previous_owner);
    assert_equal(1100, my_pony.value);
```

```
def test_buy_pony_higher_price:
    db = Mock.mock(DB.class)
    Pony pony = new Pony();
    Mock.when(db.save(pony)).return(pony);
    pony.owner = “you”;
    pony.price = 1000;
    Pony my_pony = market.buy(pony, “me”, 1100);
    assert_not_null(my_pony);
    assert_equal(“me”, my_pony.owner);
    assert_equal(“you”, my_pony.previous_owner);
    assert_equal(1100, my_pony.value);
```