

Self-testing Code in Ruby

Giovanni Sakti

Starqle

What is Self-testing code?

Self-testing code

- Code that have built-in tests
- The tests serve as a binding contract
- The tests can be run arbitrarily

What is TDD? How it differs from self-testing code?

TDD

- Practices of writing **tests** before the code
- Ensure that the code is self-tested
- It is, however, **optional** to do TDD to write self-testing code

TDD

But some companies enforce TDD because TDD enforces **YAGNI** principle

TDD

We'll see why...

TDD Steps

- Write a test
- Run the test, it should fail
- Write code just enough to pass the test
- Run the test
- Repeat

TDD & YAGNI

Because we only write just enough code to pass the test, there will be no unnecessary codes

Test in Ruby

There are several tools for doing testing
in ruby

Test in Ruby

- RSpec
- Minitest
- test-unit

Test in Ruby

Let's try using RSpec

RSpec Install

```
% gem install rspec
```

RSpec Help

```
% rspec --help
```

Now let's do TDD practice using RSpec

TDD with RSpec (1)

Create a simple test of program that we want to create

```
# game_spec.rb

RSpec.describe Game do
  describe "#score" do
    it "returns 0 for new game" do
      game = Game.new
      expect(game.score).to eq(0)
    end
  end
end
```


TDD with RSpec (2)

Run the example and watch it fail

```
% rspec game_spec.rb  
uninitialized constant Object::Game (NameError)
```

TDD with RSpec (3)

Now write just enough code to make it pass

```
# game.rb

class Game
  attr_reader :score

  def initialize
    @score = 0
  end
end
```

TDD with RSpec (3)

Now write just enough code to make it pass

```
# game_spec.rb  
require './game'  
...
```

TDD with RSpec (4)

Run the example and the test shall pass

```
% rspec game_spec.rb --color --format doc  
  
Game  
  #score  
    returns 0 for all gutter game  
  
Finished in 0.00057 seconds  
1 example, 0 failures
```

TDD with RSpec (5)

Repeat with new features

Thanks