Uncle John's Bathroom Reader

RSpec Basics 1

by Speedy Spec

You've heard so much about TDD/BDD over the years, and now you've finally decided to give it a try at Lookout. Good for you! You're so smart.

For this edition of **Uncle John's Bathroom Reader**, we're going to discuss **RSpec Basics**. This is not comprehensive, just enough to entertain you for a couple of minutes (hopefully). You can find more details and links at the bottom of this flyer.

RSpec works best if you use it to *describe* the code you're preparing to write, i.e."test-first development."

```
describe Square do
  describe '#width' do
  end
end
```

In this example, we are describing the `Square` class and its `width` method. You'll notice that we're not actually testing anything yet, just describing the code we're about to write. Let's add some actual tests:

```
describe '#width' do
    context 'with a default Square' do
      subject { Square.new.width }
      it { should_not be_nil }
      it { should eql(0) }
  end
end
```

The first thing to note here is the use of **subject** which sets the subject for all of the **it** blocks, also known as "examples." Since the examples are all testing the behavior of a default Square object, we group them together in a **context** block to inform the reader.

The examples above are also using "implicit" subjects, instead of explicit subjects to make the spec more readable. Explicit subjects are frowned upon, but look like this:

```
it { Square.new.width.should_not be_nil
```
or
```
subject { Square.new.width }
it { subject.should_not be_nil }
```

RSpec is designed to be as readable as possible, if your examples sound weird when you read them aloud, then you might be missing something.

That's it for this issue of **Uncle John's Bathroom Reader**, now go wash your hands.

RSpec Keywords

`describe`: Method used to specify what object or "thing" we are testing.

`context`: Syntax-sugar for "scoping" tests, either for the reader's benefit or to use separate **before** / **after** blocks.

`it`: Method for specifying an "example" block. All tests are specified with this block.

`subject`: Method used to specify what the subject used within the **it** blocks will be. When using **describe** with a class, the subject will automatically be an instance of that class

`let`: Method for setting variables to be used inside of examples, like **subject** the block passed to **let** will be lazily evaluated.

`mock`, `double`: Methods which will create mock objects which can be for setting expectations or stubbing methods.

`before`: Method which will invoke a block before examples run in the current context or describe block. Can take `:each`, to run before each example, or `:all` to run once before all examples run.

`after`: Method which will invoke a block after examples run in the current context or describe block. Accepts the same arguments as **before**.