IronRubying your C#

Tim Wingfield

Hockey, hockey, hockey, hockey...hockey all the time...

Central OH
Wendy
Boys
Kona
Hockey
local/regional speaker

Thursday, July 22, 2010
not here to sell you on Ruby
not an intro to Ruby
What can you do with IronRuby?
Right tool, right job

Why ?

expressive
focus on testing
rails

Why ?

expressive
focus on testing
rails

Why 💎 ?

expressive
focus on testing
rails

Why 💎 ?

expressive
focus on testing
rails

Why IronRuby ?

new-ish, 1.0 on April 12, 1.1 July 16
rails
use with existing .net code

Why Iron**Ruby** ?

new-ish, 1.0 on April 12, 1.1 July 16
rails
use with existing .net code

Why 🔴 Iron**Ruby** ?

new-ish, 1.0 on April 12, 1.1 July 16
rails
use with existing .net code

new-ish, 1.0 on April 12, 1.1 July 16
rails
use with existing .net code

What is?

1.8.6

other DLR languages
IronPython
F#

Thursday, July 22, 2010

start: IronRuby.net

Thursday, July 22, 2010
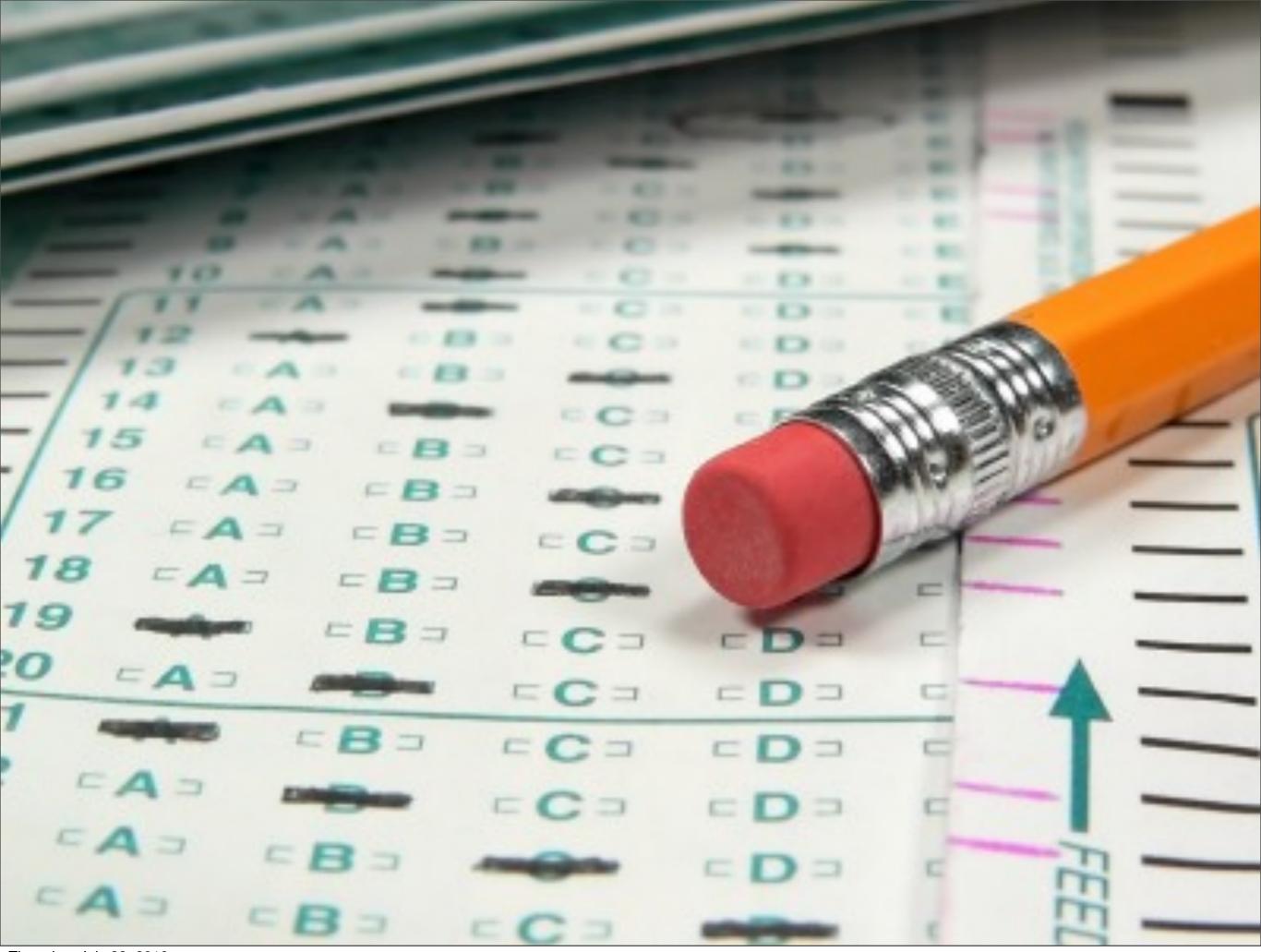
start: IronRuby.net

Thursday, July 22, 2010

start: IronRuby.net

prepend an i => igem, iirb, ir instead of ruby
performance not awesome
rails/sinatra need rack for IIS
C extensions (bson in mongo)
no need to get to the CLR, don't use it

<code>

first bit of code
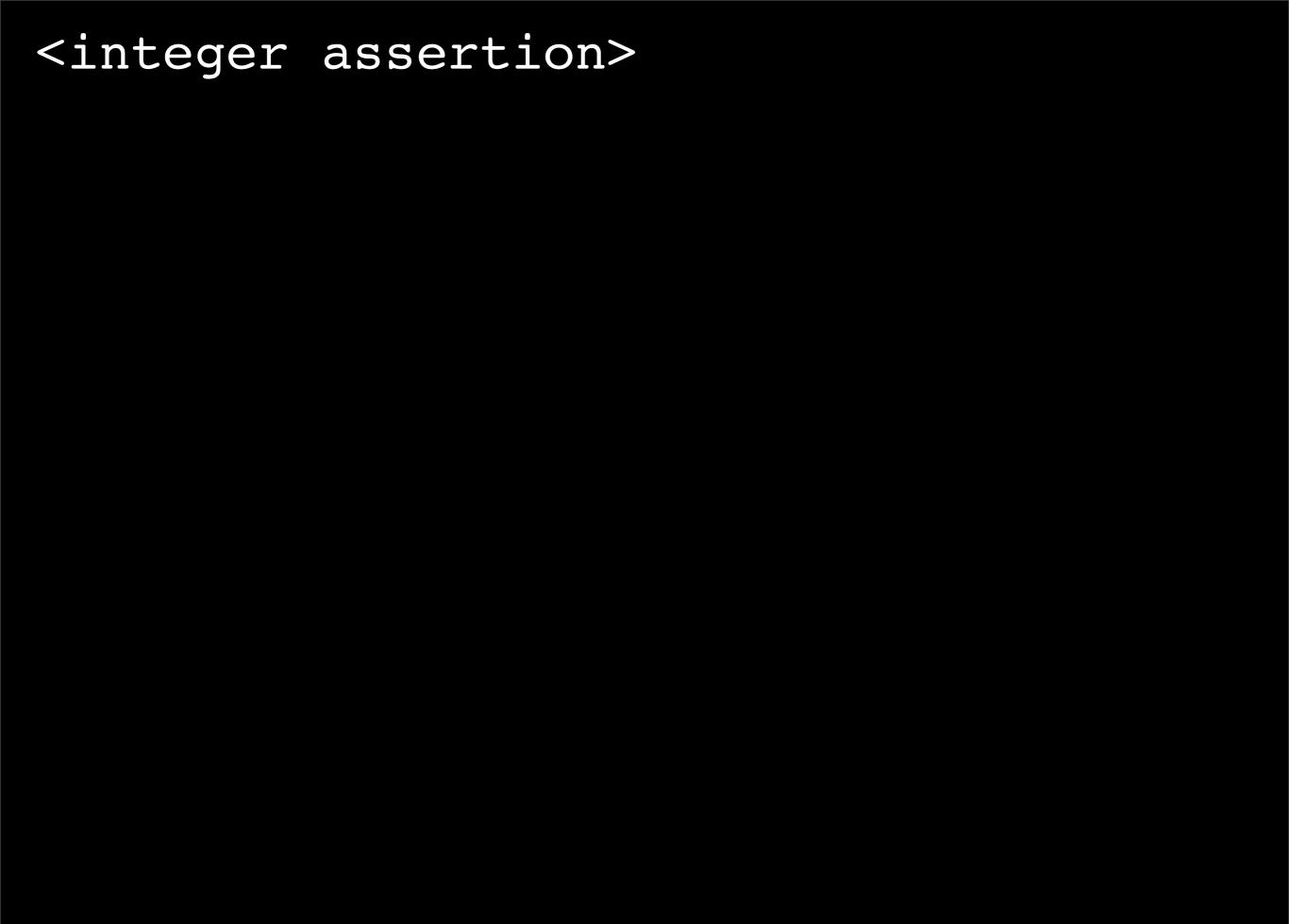loading datetime
loading monopoly object
monkey patch something

essence v. ceremony (neal ford)
Raiders of the Lost Ark
zero based arrays?
c#/kpl example w/Brendan

<nunit v. rspec>

start with mocking the long way

```csharp
public class When_verifying_the_nunit_environment
{
    Player p;
    [SetUp]
    public void Because()
    {
        p = new Player();
    }


    [Test]
    public void then_the_player_should_not_be_null()
    {
        Assert.That(p, Is.Not.Null);
    }
}
```

nunit
note ceremony
14 lines

```ruby
describe 'when verifying the environment' do
  before :each do
    @p = Player.new
  end

  it 'should create a CLR object from the Monopoly project' do
    @p.should_not be_nil
  end
end
```
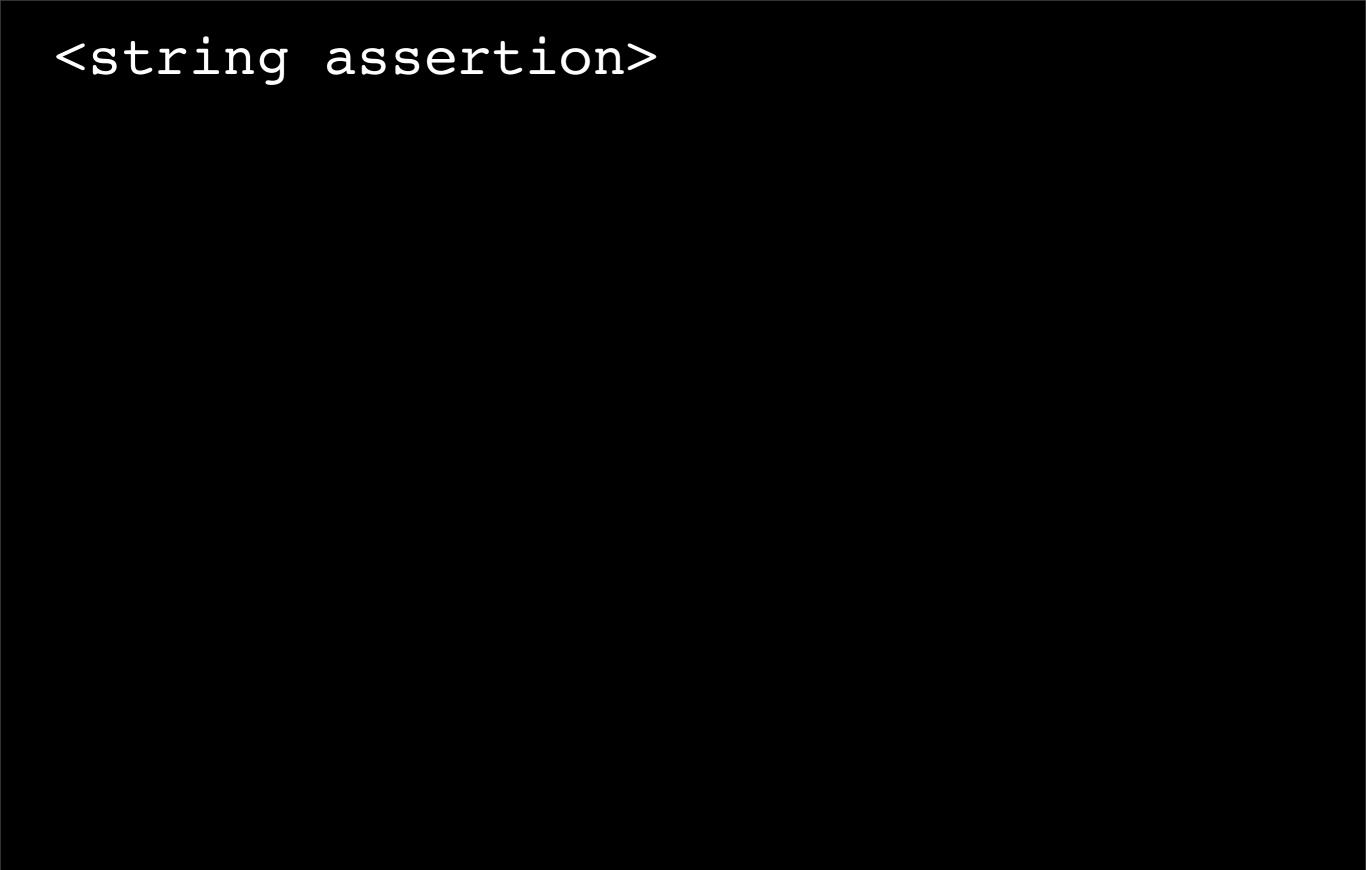
rspec
note essence
8 lines

`<integer assertion>`

`<integer assertion>`

Assert.That(_player.AccountBalance, Is.EqualTo(125));

<integer assertion>

Assert.That(_player.AccountBalance, Is.EqualTo(125));

@player.account_balance.should be(125)

<string assertion>

`<string assertion>`

Assert.That(_message, Is.EqualTo("Payment successful"));

<string assertion>

Assert.That(_message, Is.EqualTo("Payment successful"));

@message.should == "Payment successful"

<mocks>

start with mocking the long way

```csharp
public class MockRepository : IBankRepository
{
    public bool PutMoneyInTheBankWasCalled { get; set; }

    public void PutMoneyInTheBank(int amount)
    {
        this.PutMoneyInTheBankWasCalled = true;
    }

    public bool TakeMoneyFromTheBank(int amount)
    {
        return amount < 500;
    }

    public int CheckFundsAvailable()
    {
        throw new NotImplementedException();
    }
}
```

Typical hand rolled mock
Note "Not Implemented"

```ruby
class MockRepository
  include Monopoly::IBankRepository

  attr_accessor :put_money_in_the_bank_was_called

  def take_money_from_the_bank(amount)
    amount < 500
  end


  def put_money_in_the_bank(amount)
    @put_money_in_the_bank_was_called = true
  end
end
```

```csharp
private Player _player;
private BankerService _service;
private MockRepository _repo;

[SetUp]
public void before_each()
{
    _player = new Player { AccountBalance = 875 };
    _repo = new MockRepository();
    _service = new BankerService(_repo);
}
```

```ruby
before :each do
  @player = Player.new
  @player.account_balance = 875

  @repo = MockRepository.new
  @service = BankerService.new @repo
end
```

<mocks v2>

caricature v. moq

```csharp
private Player _player;
private BankerService _service;
private Mock<IBankRepository> _mock;

[SetUp]
public void before_each()
{
    _player = new Player { AccountBalance = 875 };
    _mock = new Mock<IBankRepository>();
    _mock.Setup(m => m.TakeMoneyFromTheBank(550)).Returns(false);
    _mock.Setup(m => m.TakeMoneyFromTheBank(It.IsAny<int>()))
            .Returns(true);

    _service = new BankerService(_mock.Object);
}
```

```ruby
before :each do
  @player = Player.new
  @player.account_balance = 875

  @isolation = Isolation.for IBankRepository
  @isolation.when_receiving(:take_money_from_the_bank).return(true)
  @isolation.when_receiving(:take_money_from_the_bank)
    .with(550).return(false)

  @service = BankerService.new @isolation
end
```

`<mock verification>`

<mock verification>

```
_mock.Verify(m => m.PutMoneyInTheBank(It.IsAny<int>()),
Times.AtLeastOnce());
```
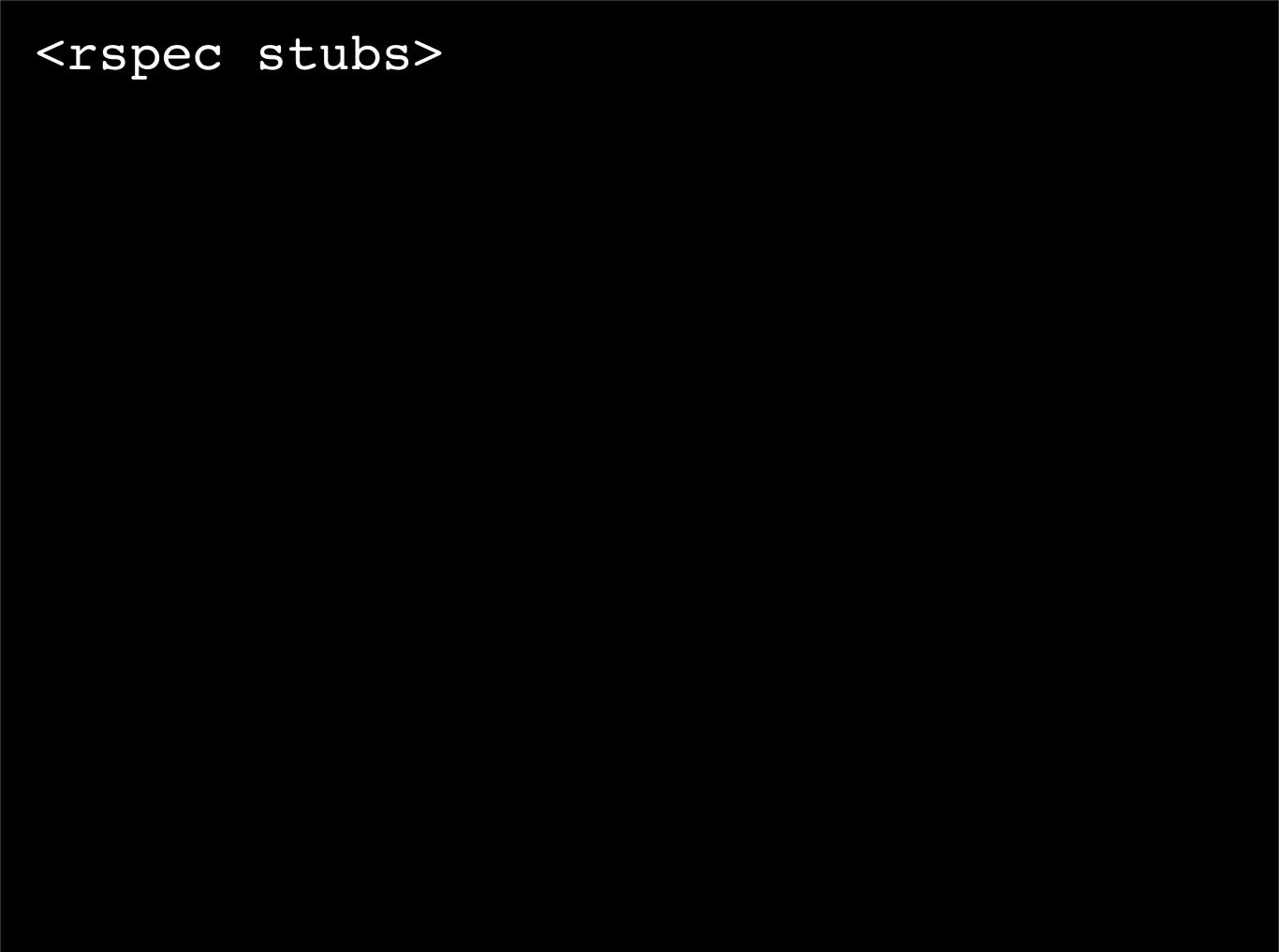
<mock verification>

```
_mock.Verify(m => m.PutMoneyInTheBank(It.IsAny<int>()),
Times.AtLeastOnce());
```

```
@isolation.did_receive?(:put_money_in_the_bank).with(50)
```
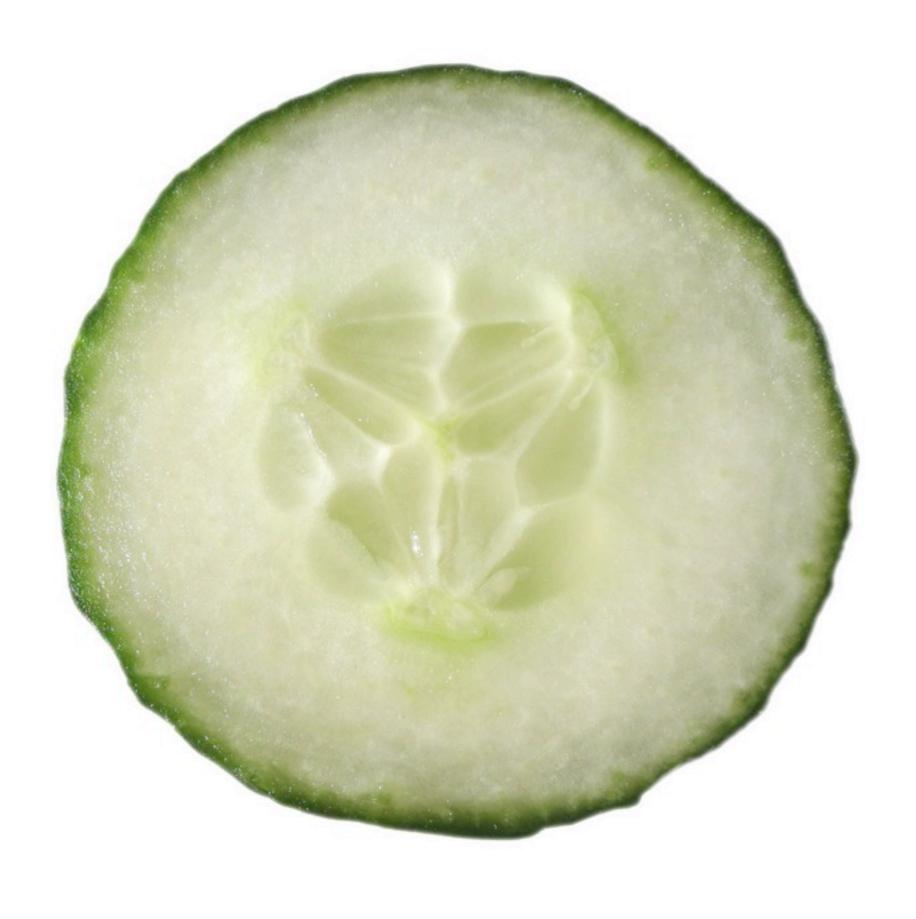
<rspec stubs>

rspec stub

<rspec stubs>

```ruby
class StubbedRepo
  include Monopoly::IPropertyRepository
end
```

rspec stub

`<rspec stubs>`

```ruby
class StubbedRepo
  include Monopoly::IPropertyRepository
end


@repo = StubbedRepo.new
@repo.stub!(:get_all_properties).and_return(@all_properties)
```

rspec stub

use older version of Cucumber (0.6.4)
not cuke for nuke

Why?

<code>

ride the rails

Tim Wingfield

tim@timwingfield.com

blog.timwingfield.com

@timwingfield

github.com/timwingfield/TestingSeaPound