rr

(Double Ruby)

Why?

- Terse syntax
  - Don't need to explicitly verify expectations
- Closer to Test Doubles
  - http://xunitpatterns.com/Test%20Double.html

Terse Syntax

```
#flexmock
flexmock(User).should_receive(:find).with('42').and_return(jane)

#rspec
User.should_receive(:find).with('42').and_return(jane)

# Mocha
User.expects(:find).with('42').returns {jane }
# Rspec using return value blocks
User.should_receive(:find).with('42') {jane} # RR
mock(User).find('42') {jane}
```

Methods

mock versus stub

stub(User).find('42') {jane} versus
mock(User).find('42') {jane}

Spies

Adding a DoubleInjection to an Object + Method (done by stub,mock, or dont_allow) causes RR to record any methodinvocations to the Object + method. Assertions can then be made on the recorded method calls.

```
subject = Object.new stub(subject).foo subject.foo(1) subject.should have_received.foo(1)
subject.should have_received.bar # this fails
```

Flexible Parameters

```
mock(object).foobar(/on/)mock(object).foobar(1..10) mock(object).fo
obar(hash_including( :red => "#FF0000", :blue =>
"#0000FF")) mock(object).foobar.times(any_times) # watch out
```

My examples

```
describe :show do
  it "should return the points for the user" do
    @user = new_iphone_user
    mock(@user).points { 99 }
    log_in_as @user
    get :show
    response.body.should == '{"points":99}'
    response.status.should == 200
  end
end
```

cron

```
def set_hour(h)
  n = Object.new
  stub(n).hour { h }
  stub(Time).now { n }
end

describe 'groupon fetching' do
  it "should fetch divisions every day at 15:00ish"
do
    set_hour 15
    mock(Groupon::DivisionFetcher).fetch
    Cron.run
  end
```

```
it "should mark deleted divisions as inactive" do
  mock(UrlFetcher).fetch_groupon('divisions') { @divisions_json }
  Groupon::DivisionFetcher.fetch
  mock(UrlFetcher).fetch_groupon('divisions') { {} }
  Groupon::DivisionFetcher.fetch

  GrouponDivision.count.should == 6
  GrouponDivision.all.each { |div| div.active.should == false }
end
```

```
it 'should ask Groupon for deal details for the division' do
  stub(UrlFetcher).fetch_groupon('deals', anything)
  mock(Groupon::DealFetcher).get_deal_guids(anything) { [] }
  mock(Groupon::DealFetcher).fetch_each([], @div)
  Groupon::DealFetcher.fetch_all
end
```

.....

```
  def self.fetch url
    raise 'should not request remote resource directly in test environment' if ::Rails.env
== 'test'
```

```
it 'should handle attachments correctly' do
  mock(AttachmentHelper).new(is_a Deal) do |ah|
    mock(ah).add_only_new_attachments(
        'large_image_url', ['deal_url', 'website_url'])
  end
  Groupon::DealFetcher.create_deal @wrapper, @div
end
```

Bonus: the meta spec

```
describe 'The app' do
  it 'should have valid spec file names' do
    `find spec -name "*.rb" | grep -v _spec.rb | grep -v
spec/factories | grep -v spec_helper.rb | grep -v
spec/support`.size.should == 0
  end
end
```

Resources:

github.com/btakita/rr