Tools

irb

ri

Usage: ri [options] [names...]

Class | Class::method | Class#method | Class.method | method

A '.' matches either class or instance methods, while #method matches only instance and ::method matches only class methods.

rvm docs generate

https://rubygems.org/gems/rdoc-data

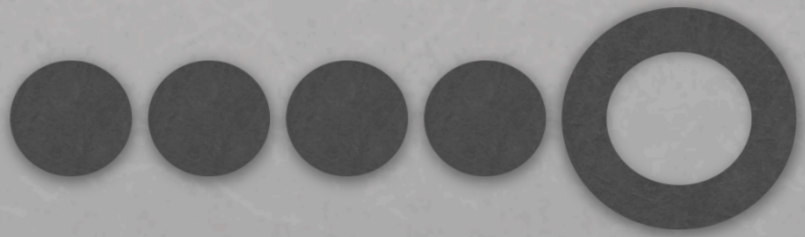
GIT

A quick tour of: clone; add; commit; and push.

GIT CLONE

Github

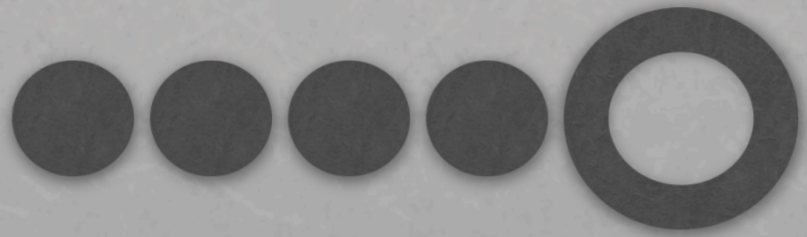
Origin Master



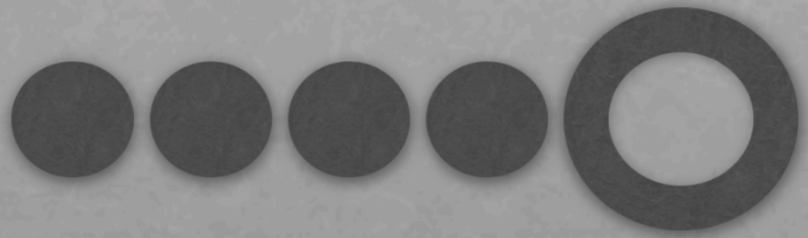
GIT CLONE

Github

Origin Master



Local Master



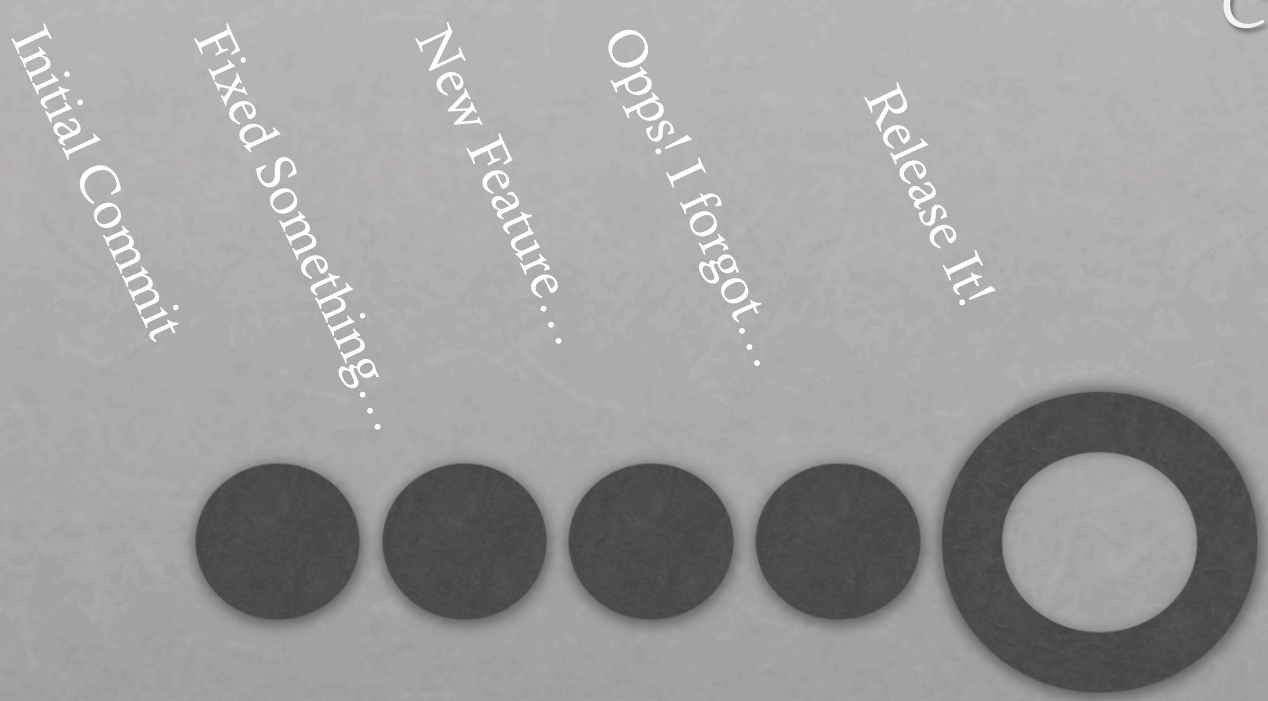
Your Machine

GIT CLONE …

Give me a complete copy of this repository on my machine.

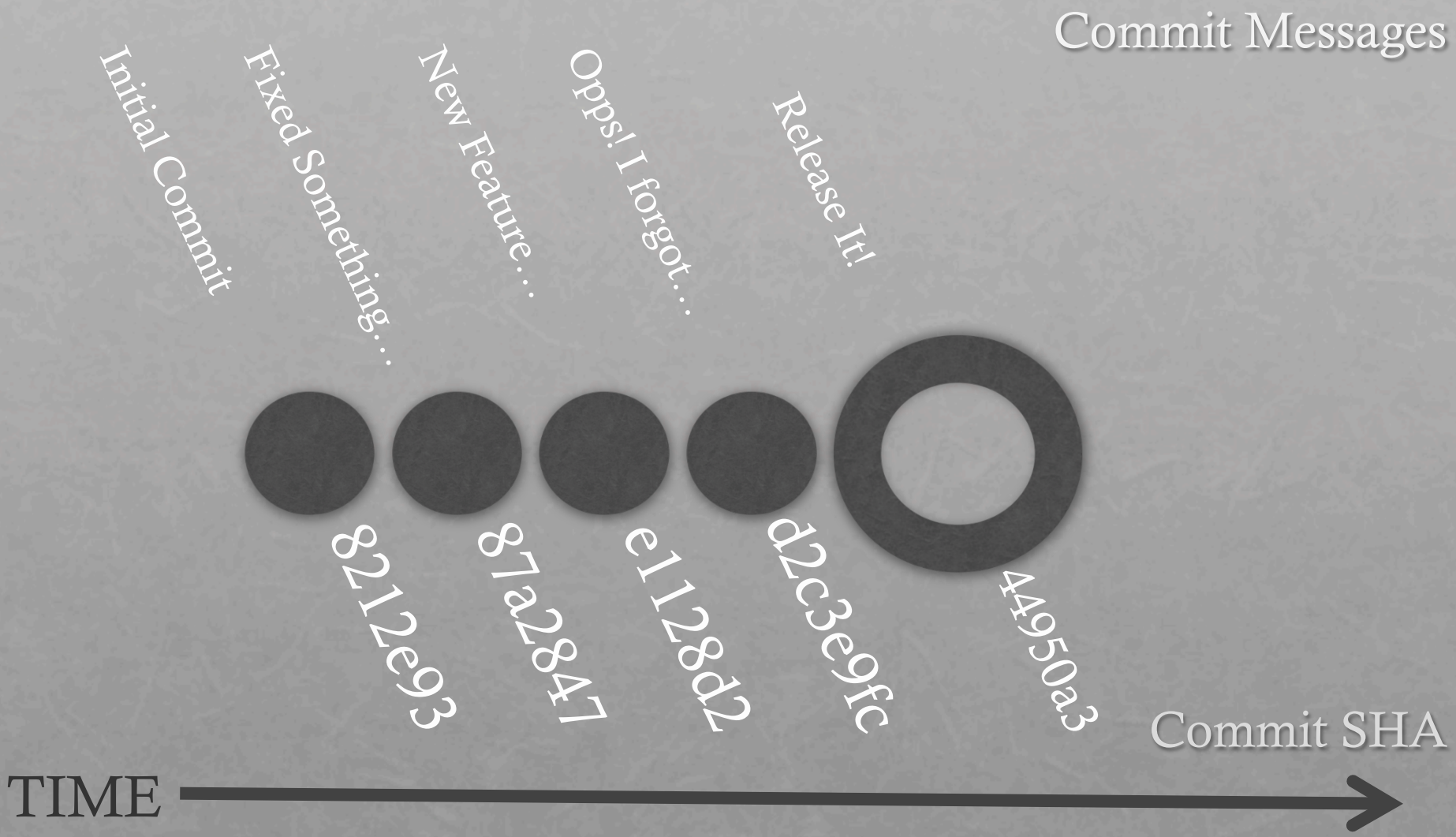
A TIMELINE

Commit Messages

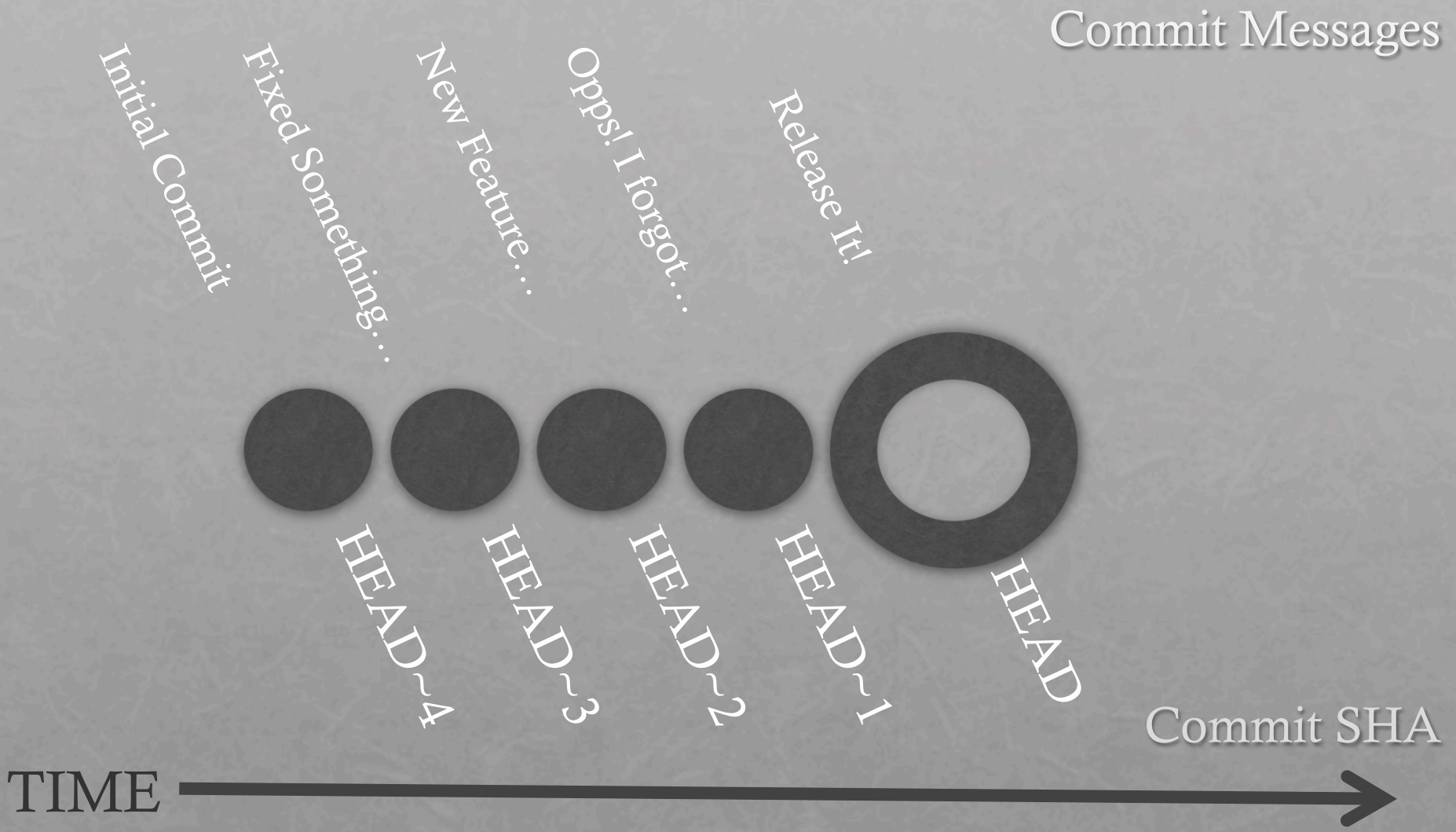


TIME

A TIMELINE

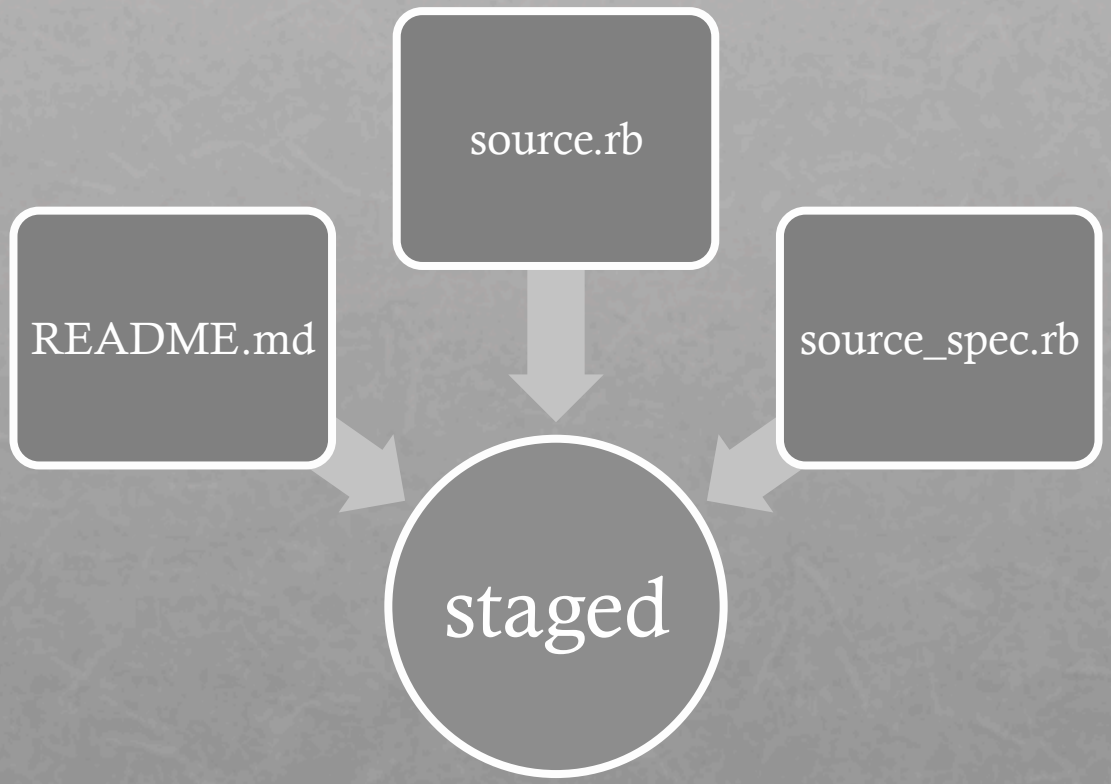

Commit Messages
Initial Commit
Fixed Something…
New Feature…
Opps! I forgot…
Release It!
8212e93
87a2847
e1128d2
d2c3e9fc
44950a3
Commit SHA
TIME

A TIMELINE

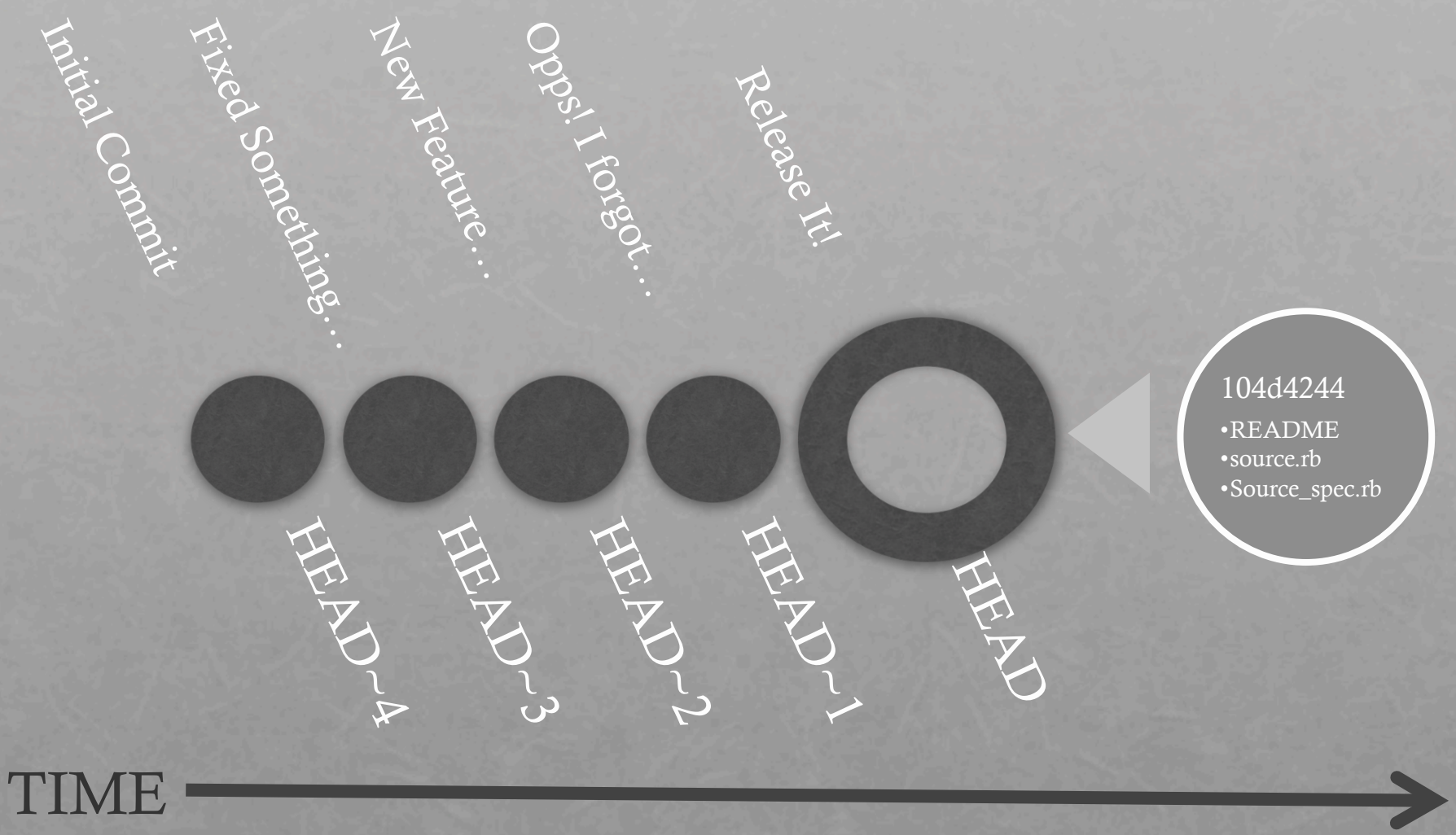

Commit Messages
Initial Commit
Fixed Something…
New Feature…
Opps! I forgot…
Release It!
HEAD~4
HEAD~3
HEAD~2
HEAD~1
HEAD
Commit SHA
TIME

GIT ADD *FILEPATH*

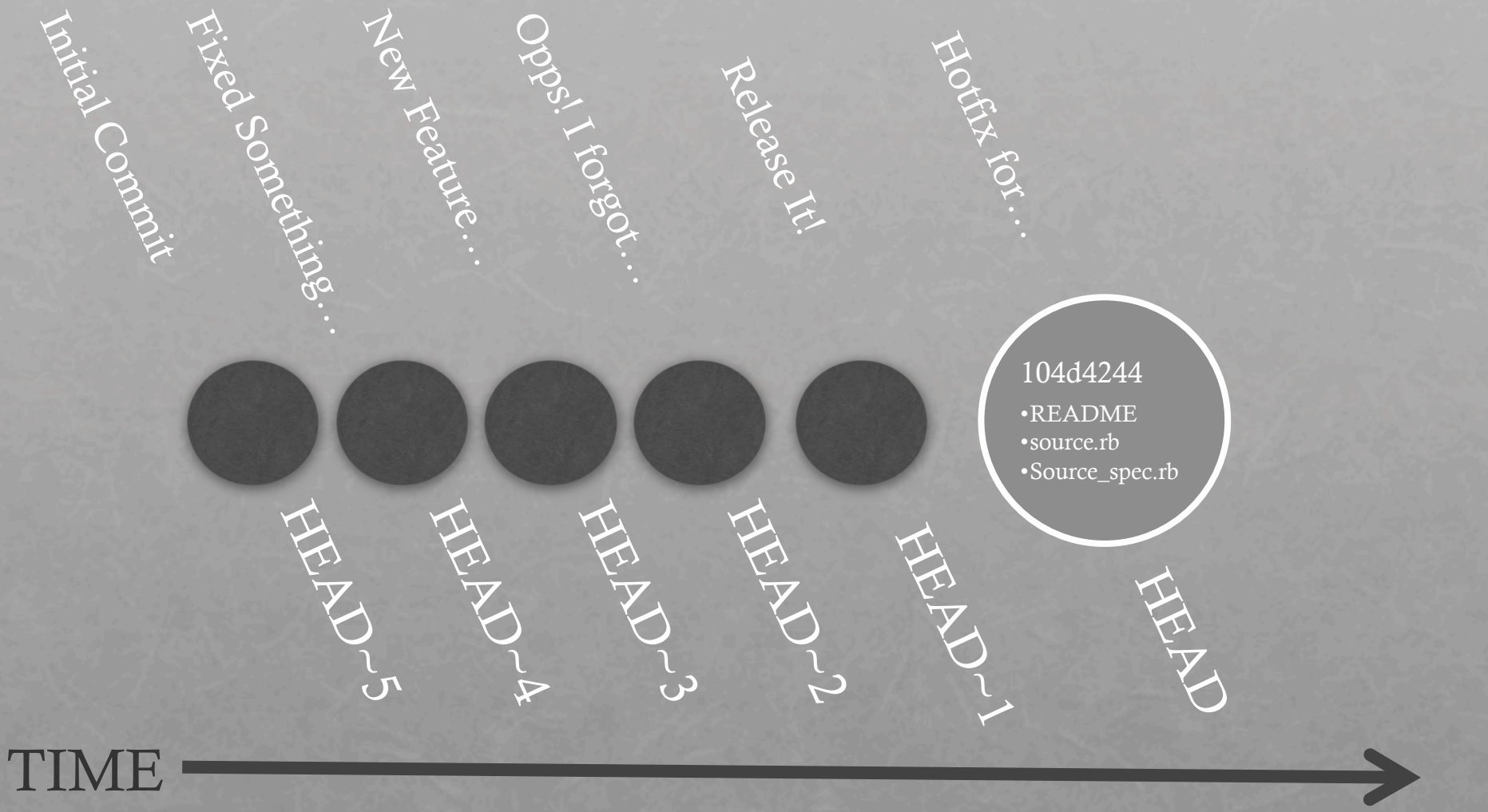
When I say ‘commit` later, I want you to include these files.



GIT COMMIT


Initial Commit
Fixed Something…
New Feature…
Opps! I forgot…
Release It!
HEAD~4
HEAD~3
HEAD~2
HEAD~1
HEAD
104d4244
•README
•source.rb
•Source_spec.rb
TIME

GIT COMMIT


Initial Commit
Fixed Something…
New Feature…
Opps! I forgot…
Release It!
Hotfix for…
104d4244
•README
•source.rb
•Source_spec.rb
HEAD~5
HEAD~4
HEAD~3
HEAD~2
HEAD~1
HEAD
TIME

THE STATE OF THE REPO

Local Master

- **Hotfix for…**
- Release
- Opps! I forgot…
- New Feature…
- Fixed Something…
- Initial Commit

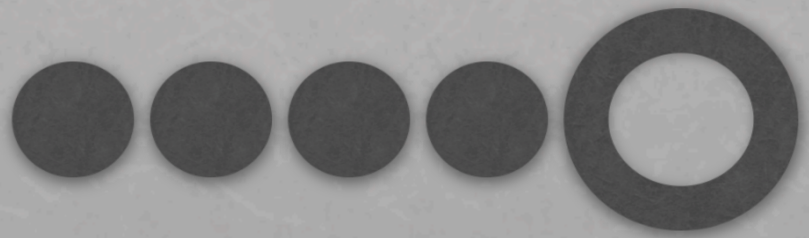
Origin Master

- **Release**
- Opps! I forgot…
- New Feature…
- Fixed Something…
- Initial Commit

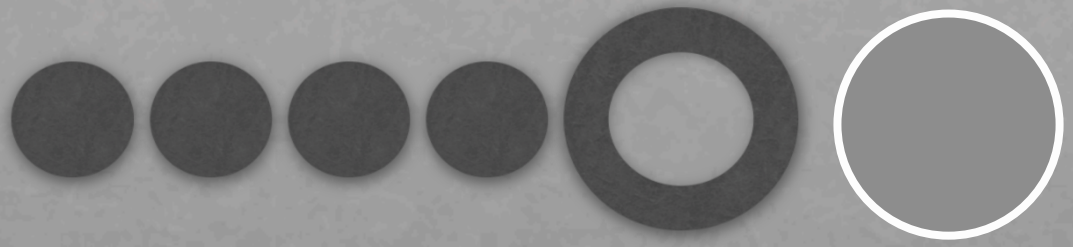
GIT PUSH ORIGIN MASTER

Github

Origin Master



Local Master



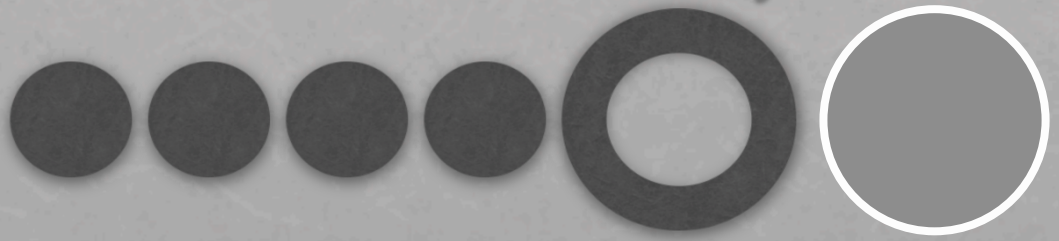
Your Machine

GIT PUSH ORIGIN MASTER

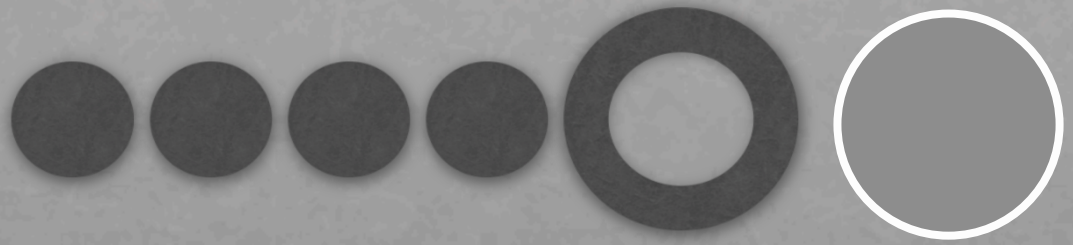
I want 'origin' to know about the changes that I committed locally.

Github

Origin Master



Local Master



Your Machine

PROGIT.ORG/BOOK

More great examples and an excellent resource

GEM

How Rubyists make installing software easy

GEM

$ gem install bundler

GEM

command

Name of the Gem

$ gem install bundler

GEM LIST

```
$ gem list
```

GEM LIST

$ gem list

*** LOCAL GEMS ***

bundler (1.0.18)

guard (0.7.0, 0.5.1)

GEM LIST

$ gem list

*** LOCAL GEMS ***

Name of the Gem

bundler (

Version(s) of the Gem

guard (0.7.0, 0.5.1)

GEM HELP

$ gem help

$ gem help commands

rubygems.org

How Rubyists make dependency management easy

BUNDLE

$ bundle

$ bundle install

GEMFILE

$ bundle

```
Installing rake (0.9.2)
Installing diff-lcs (1.1.2)
Installing thor (0.14.6)
Installing guard (0.5.1)
Installing guard-rspec (0.4.0)
Installing rspec-core (2.6.4)
Installing rspec-expectations (2.6.0)
Installing rspec-mocks (2.6.0)
Installing rspec (2.6.0)
Installing bundler (1.0.18)
Your bundle is complete! Use `bundle show [gemname]` to see where a
bundled gem is installed.
```

GEMFILE.LOCK

$ bundle

```
Using rake (0.9.2)
Using diff-lcs (1.1.2)
Using thor (0.14.6)
Using guard (0.5.1)
Using guard-rspec (0.4.0)
Using rspec-core (2.6.4)
Using rspec-expectations (2.6.0)
Using rspec-mocks (2.6.0)
Using rspec (2.6.0)
Using bundler (1.0.18)
Your bundle is complete! Use `bundle show [gemname]` to see where a
bundled gem is installed.
```

GEMFILE

```
source :rubygems

gem ‘rake’

gem ‘rspec’

gem ‘guard’

gem ‘guard-rspec’
```

GEMFILE

The site to ask for gems

```
source :rubygems

gem 'rake'

gem 'rspec'

gem 'guard'

gem 'guard-rspec'
```

Name of the gem

GEMFILE

The site to ask for gems

```
source :rubygems

gem 'rake'

gem 'rspec'

gem 'guard'

gem 'guard-rspec', '1.0.1'
```

Name of the gem

Optional: Version of the gem

gembundler.com

RSPEC

Behavior Driven Development

RSPEC *FILEPATH*

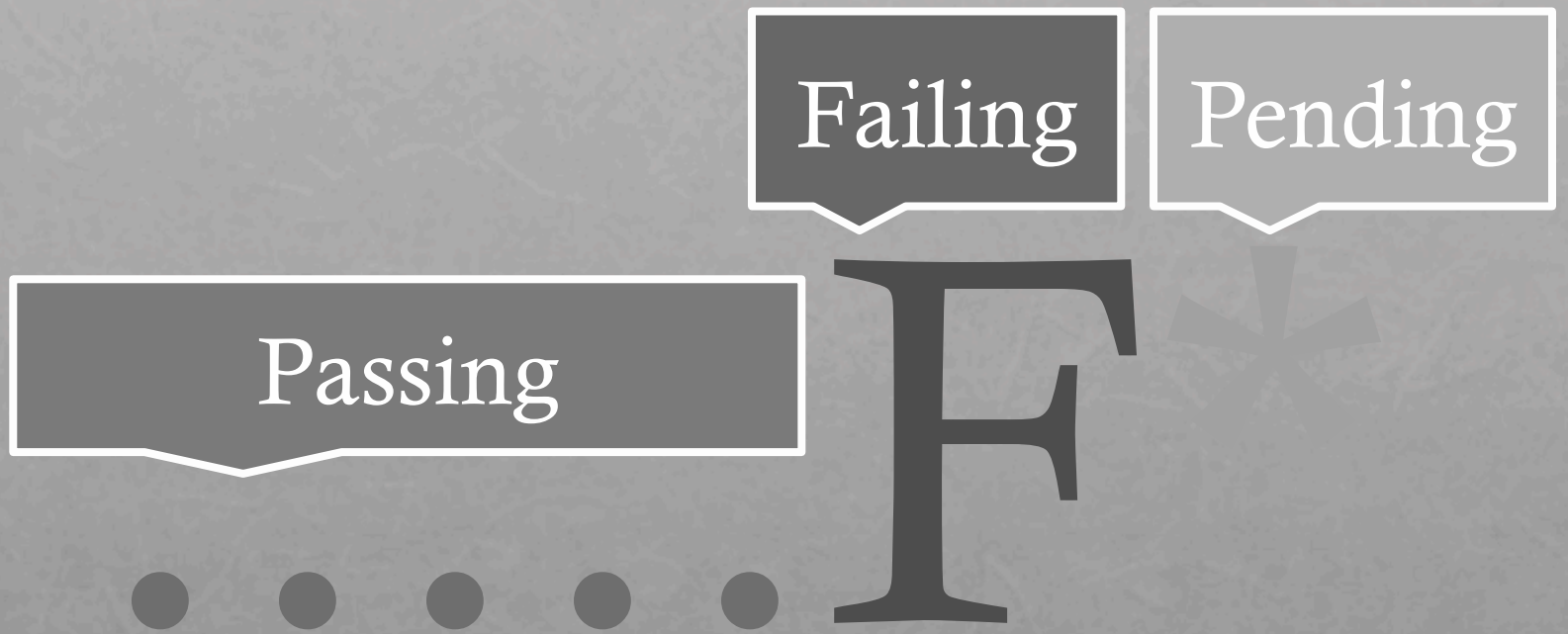
```
$ rspec spec/00_rspec_spec.rb
```

RSPEC *FILEPATH*

```
$ rspec spec/00_rspec_spec.rb

.....F*.

Pending:
  The Rspec ruby gem Domain Specific Language …

Failures:

1)  The Rspec ruby gem Domain Specific Language ….

Finished in 0.0022 seconds
8 examples, 1 failure, 1 pending
```

RSPEC OUTPUT

Passing, Pending, and Failing



PENDING EXAMPLES

Pending:

The Rspec ruby gem Domain Specific Language …

```
# Not Yet Implemented
# ./spec/00_rspec_spec.rb:49
```

PENDING EXAMPLES

Pending:

The Rspec ruby gem Domain Specific Language …

File

Line Number

Not Implemented

./spec/00_rspec_spec.rb:49

FAILING EXAMPLES

Failures:

1) The Rspec ruby gem Domain Specific Language alerts...
Failure/Error: 1.should eq 2

expected 2
got 1

(compared using ==)
./spec/00_rspec_spec.rb:45:in `block (3…`

FAILING EXAMPLES

Failures:

1) The Rspec ruby [illegible] alerts...
Failure/Error: 1.should eq 2

Got

Expected

expected 2
got 1

eq is the same as writing ==

(compared using ==)
./spec/00_rspec_spec.rb:45:in `block (3…`

FAILING EXAMPLES

Failures:

1) The Rspec ruby gem Domain Specific Language alerts...
Failure/Error: 1.should eq 2

expected 2
got 1

(c[illegible])

File

Line Number

./spec/00_rspec_spec.rb:45:in `block (3…`

RSpec --format documentation

```
$ rspec -f d spec/00_rspec_spec.rb
```

```
The Rspec ruby gem
  Domain Specific Language
    creates examples with the #it keyword
    has keywords like #context, and #describe to help organize the…
    has easily readable methods like #should to test any object
    has #should_not to test for negative cases
    creates readable predicate methods
    alerts you when examples fail (FAILED - 1)
    supports placeholder… (PENDING: Not Yet Implemented)
    requires that examples use expectations (like #should) to work…
```

RSpec Language

Behavior Driven Development

relishapp.com/rspec/docs/gettingstarted

DESCRIBE AND IT

Terminal Output

The Rspec ruby gem
  Domain Specific Language
    creates examples with the #it keyword

```
describe "The Rspec ruby gem" do

  context "Domain Specific Language" do

    it "creates examples with the #it keyword" do

      1.should eq 1
```

00_rspec_spec.rb

DESCRIBE AND IT

Terminal Output

```
The Rspec ruby gem
  Domain Specific Language
    creates examples with the #it keyword
```

```
describe "The Rspec ruby gem" do

  context "Domain Specific Language" do

    it "creates examples with the #it keyword" do

      1.should eq 1
```

00_rspec_spec.rb

DESCRIBE AND IT

Terminal Output

```
The Rspec ruby gem
  Domain Specific Language
    creates examples with the #it keyword
```

```
describe "The Rspec ruby gem" do

  context "Domain Specific Language" do

    it "creates examples with the #it keyword" do

      1.should eq 1
```

00_rspec_spec.rb

DESCRIBE AND IT

Terminal Output

```
The Rspec ruby gem
  Domain Specific Language
    creates examples with the #it keyword
```

```
describe "The Rspec ruby gem" do

  context "Domain Specific Language" do

    it "creates examples with the #it keyword" do

      1.should eq 1
```

00_rspec_spec.rb

DESCRIBE AND IT

Terminal Output

```
The Rspec ruby gem
  Domain Specific Language
    creates examples with the #it keyword
```

```
describe "The Rspec ruby gem" do

  context "Domain Specific Language" do

    it "creates examples with the #it keyword" do

      1.should eq 1
```

00_rspec_spec.rb

DESCRIBE AND IT

```
describe "The Rspec ruby gem" do
  context "Domain Specific Language" do
    it "creates examples with the #it keyword" do
      1.should eq 1
    end
  end
end
```

DESCRIBE AND IT

```
describe "The Rspec ruby gem" do
  context "Domain Specific Language" do
    it "creates examples with the #it keyword" do
      1.should eq 1
    end
  end
end
```

This could have been 'describe'

SHOULD

```
1.should eq 1
```

EXPECTATIONS

```
1.should eql 1

1.should == 1

1.should eq 1

1.should equal 1

1.should be 1
```

ALSO ACCEPTING

```
1.should eql(1)

1.should == 1

1.should eq(1)

1.should equal(1)

1.should be(1)
```

…AND

```
1.should(eql(1))

1.should == 1

1.should(eq(1))

1.should(equal(1))

1.should(be(1))
```

SHOULD_NOT

```
1.should_not eql 2
1.should_not == 2
1.should_not eq 2
1.should_not equal 2
1.should_not be 2
```

.should be or .should_not be

be (not *nil* or *false*)

be_true (not *nil* or *false*)

be_false (*nil* or *false*)

be_nil

https://www.relishapp.com/rspec/rspec-expectations/docs/built-in-matchers