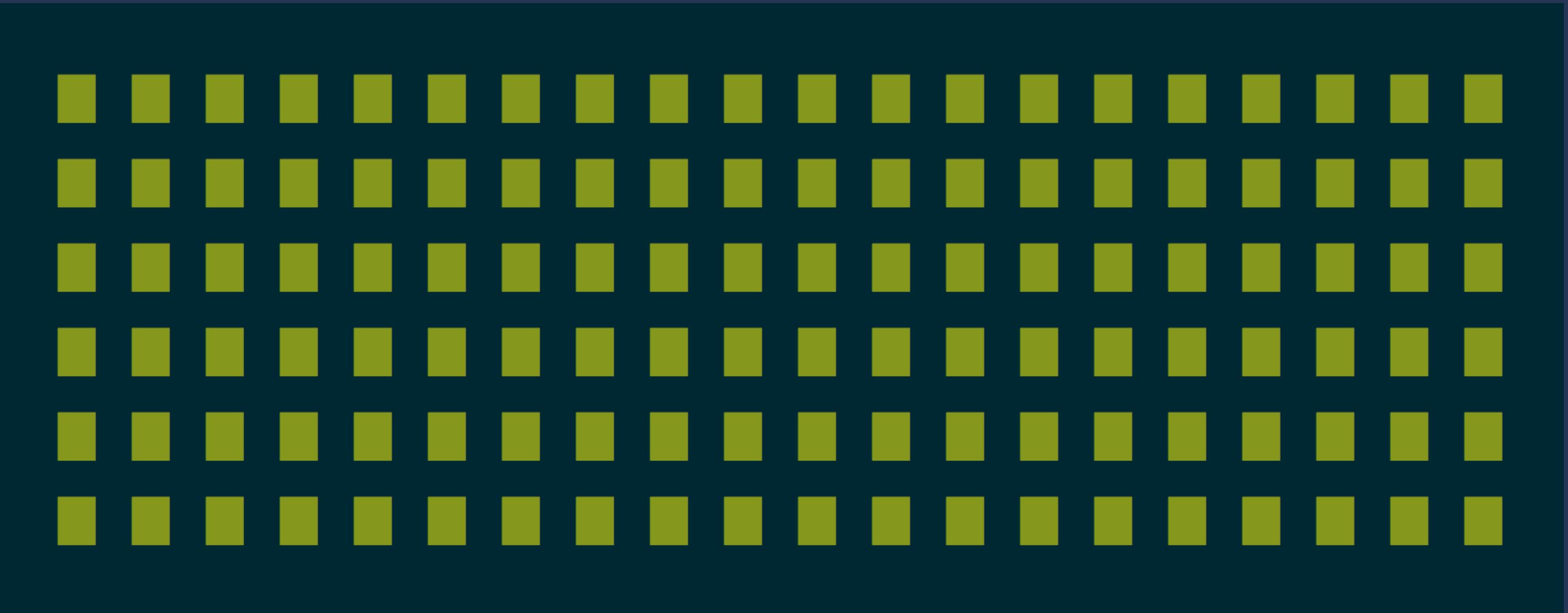
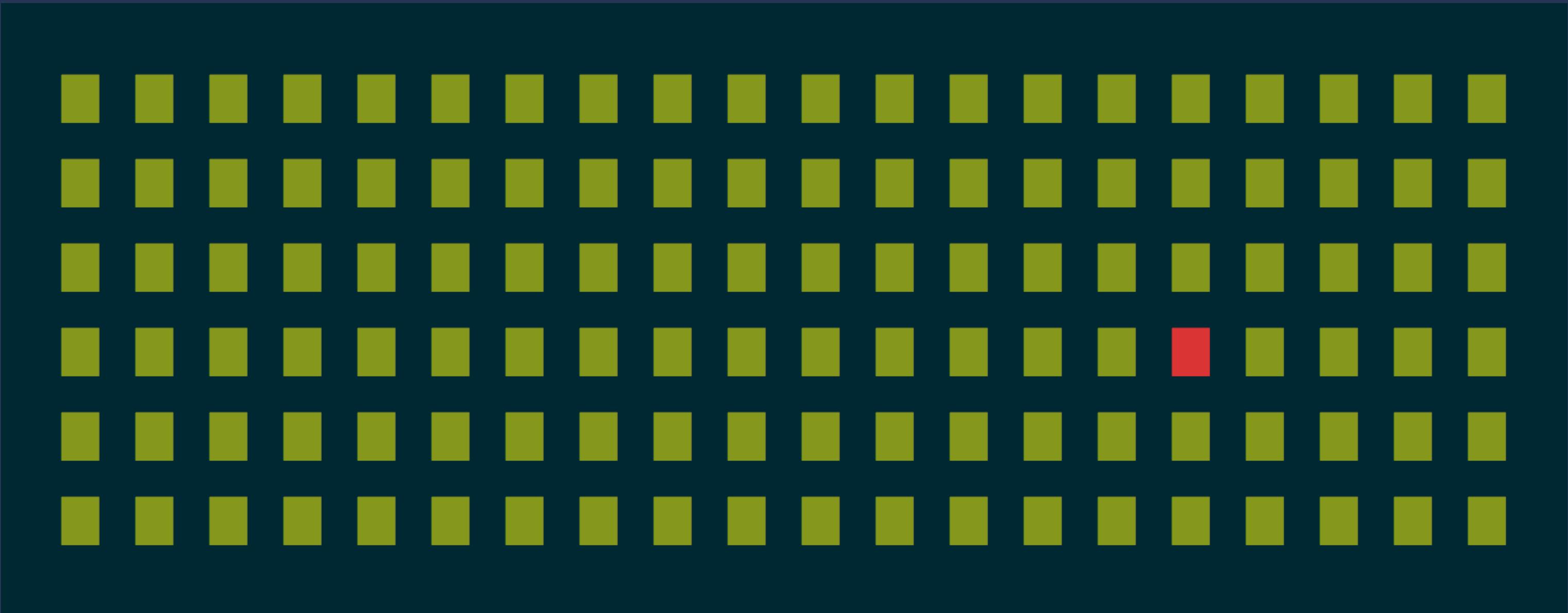


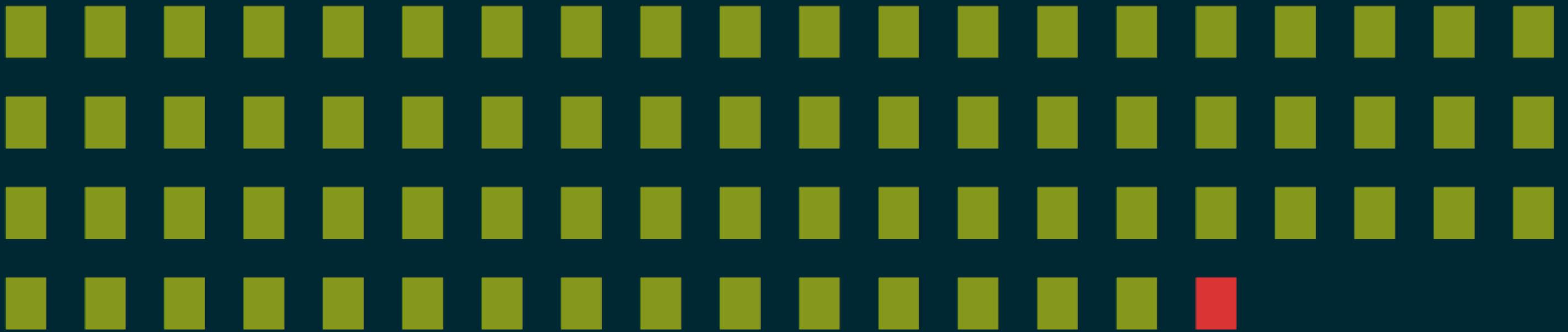


# test Bisection

# **Order-Dependent Test Failures**







**Days since last  
implemented  
binary search:  
17290**

# Bisection Algorithm

1. Fix order
2. Remove half the tests preceding the failure
3. Rerun the remainder
  - **pass**: remove the **other** half, goto **3**
  - **fail**: goto **2** with remaining tests























great!

Great!

(also boring)

# Great!

(also boring)

(also hard)



**\$ rspec --bise**

---

<sup>1</sup> <https://github.com/rspec/rspec-core/pull/1917>

```
+ s/kitten_spec.rb
```

```
1 RSpec.describe "A kitten" do
2
3   it "is totally innocent" do
4     expect(ENV["guilty"]).to be_falsey
5   end
6
7   it("is fluffy")      {}
8   it("likes string")  {}
9   it("fears hoovers") {}
10  it("likes laser pointers") {}
11  it("is cute")        {}
12  it("meows")          {}
13  it("purrs")          {}
14  it("likes being skritched") {}
15  it("loves catnip")   {}
16  it("sits in cardboard boxes") {}
17
18  it "WILL DESTROY EVERYTHING YOU HOLD DEAR, MUHAHAHAHAHAAAAAA" do
19    ENV["guilty"] = "true"
20  end
21
22 end
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb
```

```
Randomized with seed 18004
```

```
.....
```

```
Finished in 0.00272 seconds (files took 0.09121 seconds to load)  
12 examples, 0 failures
```

```
Randomized with seed 18004
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb
```

```
Randomized with seed 19582
```

```
.....F.
```

Failures:

1) A kitten is totally innocent

```
Failure/Error: expect(ENV["guilty"]).to be_falsey
expected: falsey value
      got: "true"
```

```
# ./spec/kitten_spec.rb:6:in `block (2 levels) in <top (required)>'
```

```
Finished in 0.01147 seconds (files took 0.08223 seconds to load)
```

```
12 examples, 1 failure
```

Failed examples:

```
rspec ./spec/kitten_spec.rb:5 # A kitten is totally innocent
```

```
Randomized with seed 19582
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb
```

```
Randomized with seed 19582
```

```
.....F.
```

Failures:

1) A kitten is totally innocent

```
Failure/Error: expect(ENV["guilty"]).to be_falsey
```

```
expected: falsey value
```

```
got: "true"
```

```
# ./spec/kitten_spec.rb:6:in `block (2 levels) in <top (required)>'
```

```
Finished in 0.01147 seconds (files took 0.08223 seconds to load)
```

```
12 examples, 1 failure
```

Failed examples:

```
rspec ./spec/kitten_spec.rb:5 # A kitten is totally innocent
```

```
Randomized with seed 19582
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb --seed 19582
```

```
Randomized with seed 19582
```

```
.....F.
```

Failures:

1) A kitten is totally innocent

```
Failure/Error: expect(ENV["guilty"]).to be_falsey
expected: falsey value
      got: "true"
```

```
# ./spec/kitten_spec.rb:6:in `block (2 levels) in <top (required)>'
```

```
Finished in 0.01595 seconds (files took 0.08517 seconds to load)
```

```
12 examples, 1 failure
```

Failed examples:

```
rspec ./spec/kitten_spec.rb:5 # A kitten is totally innocent
```

```
Randomized with seed 19582
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb --seed 19582
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb --seed 19582 --bisect
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb --seed 19582 --bisect
Bisect started using options: "spec/kitten_spec.rb --seed 19582"
Running suite to find failures... (0.35583 seconds)
Starting bisect with 1 failing example and 11 non-failing examples.
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb --seed 19582 --bisect
Bisect started using options: "spec/kitten_spec.rb --seed 19582"
Running suite to find failures... (0.35583 seconds)
Starting bisect with 1 failing example and 11 non-failing examples.
```

```
Round 1: searching for 6 non-failing examples (of 11) to ignore: . (0.35719 seconds)
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb --seed 19582 --bisect
Bisect started using options: "spec/kitten_spec.rb --seed 19582"
Running suite to find failures... (0.35583 seconds)
Starting bisect with 1 failing example and 11 non-failing examples.
```

```
Round 1: searching for 6 non-failing examples (of 11) to ignore: . (0.35719 seconds)
Round 2: searching for 3 non-failing examples (of 5) to ignore: . (0.34423 seconds)
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb --seed 19582 --bisect
Bisect started using options: "spec/kitten_spec.rb --seed 19582"
Running suite to find failures... (0.35583 seconds)
Starting bisect with 1 failing example and 11 non-failing examples.
```

```
Round 1: searching for 6 non-failing examples (of 11) to ignore: . (0.35719 seconds)
Round 2: searching for 3 non-failing examples (of 5) to ignore: . (0.34423 seconds)
Round 3: searching for 1 non-failing example (of 2) to ignore: .. (0.67456 seconds)
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb --seed 19582 --bisect
Bisect started using options: "spec/kitten_spec.rb --seed 19582"
Running suite to find failures... (0.35583 seconds)
Starting bisect with 1 failing example and 11 non-failing examples.
```

```
Round 1: searching for 6 non-failing examples (of 11) to ignore: . (0.35719 seconds)
Round 2: searching for 3 non-failing examples (of 5) to ignore: . (0.34423 seconds)
Round 3: searching for 1 non-failing example (of 2) to ignore: .. (0.67456 seconds)
Round 4: searching for 1 non-failing example (of 1) to ignore: . (0.3402 seconds)
Bisect complete! Reduced necessary non-failing examples from 11 to 1 in 1.72 seconds.
```

```
simon@chow:~ $ rspec spec/kitten_spec.rb --seed 19582 --bisect
Bisect started using options: "spec/kitten_spec.rb --seed 19582"
Running suite to find failures... (0.35583 seconds)
Starting bisect with 1 failing example and 11 non-failing examples.
```

```
Round 1: searching for 6 non-failing examples (of 11) to ignore: . (0.35719 seconds)
Round 2: searching for 3 non-failing examples (of 5) to ignore: . (0.34423 seconds)
Round 3: searching for 1 non-failing example (of 2) to ignore: .. (0.67456 seconds)
Round 4: searching for 1 non-failing example (of 1) to ignore: . (0.3402 seconds)
Bisect complete! Reduced necessary non-failing examples from 11 to 1 in 1.72 seconds.
```

The minimal reproduction command is:

```
rspec ./spec/kitten_spec.rb[1:1,1:12] --seed 19582
```

```
simon@chow:~ $ rspec ./spec/kitten_spec.rb[1:1,1:12] --seed 19582
```

```
simon@chow:~ $ rspec ./spec/kitten_spec.rb[1:1,1:12] --seed 19582 -f d
```

```
simon@chow:~ $ rspec ./spec/kitten_spec.rb[1:1,1:12] --seed 19582 -f d
Run options: include {:ids=>{"./spec/kitten_spec.rb"=>["1:1", "1:12"]}}
```

Randomized with seed 19582

A kitten

WILL DESTROY EVERYTHING YOU HOLD DEAR, MUHAHAHAHAHAHAAAAAA  
is totally innocent (FAILED - 1)

Failures:

1) A kitten is totally innocent

Failure/Error: expect(ENV["guilty"]).to be\_falsey  
expected: falsey value  
got: "true"

```
# ./spec/kitten_spec.rb:6:in `block (2 levels) in <top (required)>'
```

Finished in 0.01139 seconds (files took 0.09309 seconds to load)  
2 examples, 1 failure

Failed examples:

```
rspec ./spec/kitten_spec.rb:5 # A kitten is totally innocent
```

# How do I get it?

## Gemfile

```
1 source 'https://rubygems.org'  
2  
3 gem 'rspec',                  git: 'https://github.com/rspec/rspec'  
4 gem 'rspec-core',               git: 'https://github.com/rspec/rspec-core'  
5 gem 'rspec-mocks',              git: 'https://github.com/rspec/rspec-mocks'  
6 gem 'rspec-expectations',      git: 'https://github.com/rspec/rspec-expectations'  
7 gem 'rspec-support',            git: 'https://github.com/rspec/rspec-support'
```



**Thanks are due to...**

**Myron  
Marston**

**And the rest of the RSpec core team**

# what about minitest?

There's a gem for that:

<https://github.com/seattlerb/minitest-bisect>