RSpec Ruby testing

CONCEPTS

--only-failures and --next-failure need a status file set in RSpec.configure; see the detail on --only-failures.

Specs live in spec/ and end in _spec.rb; bare rspec runs spec/**/*_spec.rb.

Example group describe / context
Nested group context, before, let
Example it { expect ... }

Groups nest; metadata and hooks flow inward

RUN

rspec
Run all specs

rspec *<file> <dir>*
Run files or directories

rspec *<file>*:*<line>*
Run example or group at that line

rspec --dry-run
List examples without running them

FILTER

rspec -e "*<text>***"**
Run examples whose name contains text

rspec --tag *<tag>*
Run examples with a tag

rspec --tag ~*<tag>*
Skip examples with a tag

rspec --only-failures
Run last run's failures

rspec --next-failure
Fix failures one at a time

OUTPUT

rspec --format doc
Nested, readable output

rspec --profile
Show the 10 slowest examples

FAILURES AND ORDER

rspec --fail-fast
Stop after the first failure

rspec --seed *<n>*
Rerun with a random order seed

rspec --order random
Shuffle example order

rspec --bisect
Find the minimal failing set

WRITING SPECS

RSpec.describe *<Class>* **do**
Top-level example group

describe "#*<method>***" do**
Group for one method

context "when *<state>***" do**
Group for one situation

it "*<behaviour>***" do**
One example

let(:*<name>***) { ***<value>* **}**
Lazy, per-example helper

before { *<setup>* **}**
Run before each example

EXPECTATIONS

expect(*<x>***).to eq(***<y>***)**
Value equals

expect(*<x>***).not_to eq(***<y>***)**
Value differs

expect { *<code>* **}.to raise_error(***<Error>***)**
Block raises an error

MOCKS

allow(*<obj>***).to receive(:***<m>***).and_return(***<v>***)**
Stub a method

expect(*<obj>***).to receive(:***<m>***).with(***<args>***)**
Expect a call