

RSpec cheatsheet

version .6.3

cheatsheet © Carlos Gabaldon (blog.notesonrails.com)

Context API

```
context "A new account" do

  setup do
    @account = Account.new
  end

  specify "should have a balance of $0" do
    @account.balance.should_equal Money.new(0, :dollars)
  end

end
```

context

The context method holds an object(s) in a known state.

setup

The setup method is used to set up the context.

specify

The specify method holds the code that expresses the event and the expected outcome.

Expectation API

Arbitrary Block

```
target.should_satisfy {largl ...}
target.should_not_satisfy {largl ...}
```

Equality

```
target.should_equal <value>
target.should_not_equal <value>
```

Type

```
target.should_be_an_instance_of <class>
target.should_not_be_an_instance_of <class>
target.should_be_a_kind_of <class>
target.should_not_be_a_kind_of <class>
target.should_respond_to <symbol>
target.should_not_respond_to <symbol>
```

Pattern Matching

```
target.should_match <regex>
target.should_not_match <regex>
```

Floating Point Comparison

```
target.should_be_close <value>, <tolerance>
target.should_not_be_close <value>, <tolerance>
```

Identity

```
target.should_be <value>
target.should_not_be <value>
target.should_predicate [optional args]
target.should_be_predicate [optional args]
target.should_not_predicate [optional args]
target.should_not_be_predicate [optional args]
```

Collections

```
target.should_include <object>
target.should_not_include <object>
target.should_have(<number>).things
target.should_have_at_least(<number>).things
target.should_have_at_most(<number>).things
```

Raising

```
proc.should_raise <exception>
proc.should_not_raise <exception>
proc.should_raise
proc.should_not_raise
```

Throwing

```
proc.should_throw <symbol>
proc.should_not_throw <symbol>
proc.should_not_throw
```

more @ <http://rspec.rubyforge.com>