

### rSpec Object Block Expectations

`obj.should_satisfy {|arg| ... }`

`obj.should_not_satisfy {|arg| ... }`

### rSpec Object Equality Expectations

`obj.should_equal <value>`

`obj.should_not_equal <value>`

`obj.should == <value>`

### rSpec Floating Point Comparison

`obj.should_be_close <val>, <delta>`

`obj.should_not_be_close <val>, <delta>`

### rSpec Identity Expectations

`obj.should_be <value>`

`obj.should_not_be <value>`

### rSpec Predicate Expectations

`obj.should_predicate [args]`

`obj.should_be_predicate [args]`

`obj.should_not_predicate [args]`

`obj.should_not_be_predicate [args]`

`obj.should_be > <value>`

`obj.should_be < <value>`

### rSpec Pattern Matching Expectations

`obj.should_match <regex>`

`obj.should_not_match <regex>`

`obj.should =~ <regex>`

### rSpec Ancestor Class Expectations

`obj.should_be_an_instance_of <class>`

`obj.should_not_be_an_instance_of  
<class>`

`obj.should_be_a_kind_of <class>`

`obj.should_not_be_a_kind_of <class>`

### rSpec Type Expectations

`obj.should_respond_to <sym>`

`obj.should_not_respond_to <sym>`

### rSpec Raise Expectations

`proc.should_raise <exception>`

`proc.should_not_raise <exception>`

`proc.should_raise`

`proc.should_not_raise`

### rSpec Throwing Expectations

`proc.should_throw <symbol>`

`proc.should_not_throw <symbol>`

`proc.should_not_throw`

### rSpec Collection Existence Expectations

`obj.should_include <obj>`

`obj.should_not_include <obj>`

### rSpec Collection Size Expectations

`obj.should_have(<n>).things`

`obj.should_have_at_least(<n>).things`

`obj.should_have_at_most(<n>).things`

### rSpec Creating a Mock

`mock = mock(<name>)`

`mock = mock(<name>, <options>)`

#### Example:

`mock = mock("blah", :null_object => true)`

### rSpec Mock Expecting Messages

`mock.should_receive(<msg>)`

`mock.should_not_receive(<msg>)`

### rSpec Mock Expecting Arguments

`mock.with(<args>)`

`mock.with(:no_args)`

`mock.with(:any_args)`

### rSpec Mock Receive Counts

`mock.any_number_of_times`

`mock.once`

`mock.twice`

`mock.exactly(<n>).times`

`mock.at_least(:once)`

`mock.at_least(:twice)`

`mock.at_least(<n>).times`

### rSpec Mock Return Values

`mock.and_return(<val>)`

`mock.and_return(<val1>, <val2>, <valN>)`

`mock.and_return(...)`

### rSpec Mock Raising and Throwing

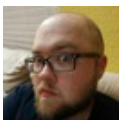
`mock.and_raise(<exception>)`

`mock.and_throw(<symbol>)`

### rSpec Mock Ordering

`mock.should_receive(:flip).once.ordered`

`mock.should_receive(:flop).once.ordered`



By **Ryan Johnson** (CITguy)  
[cheatography.com/citguy/](https://cheatography.com/citguy/)

Published 15th November, 2011.  
Last updated 15th November, 2011.  
Page 1 of 1.

Sponsored by **Readability-Score.com**  
Measure your website readability!  
<https://readability-score.com>