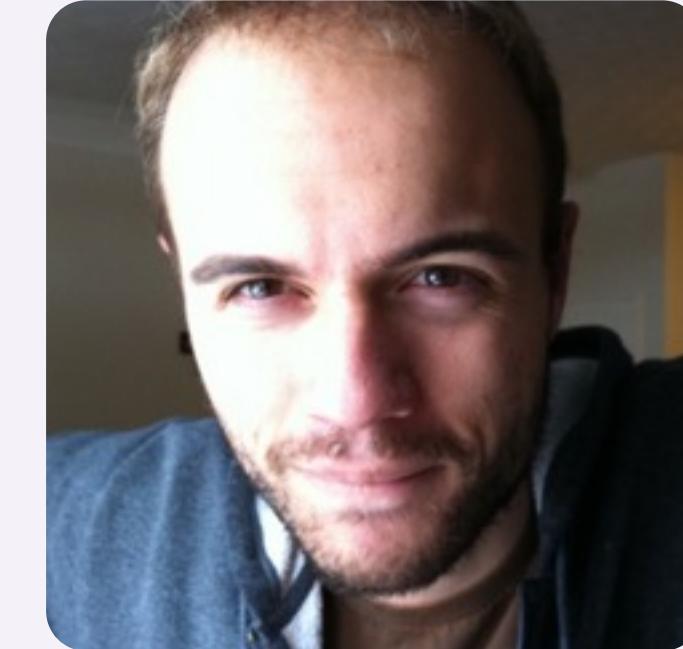


# test double

[hello@testdouble.com](mailto:hello@testdouble.com)

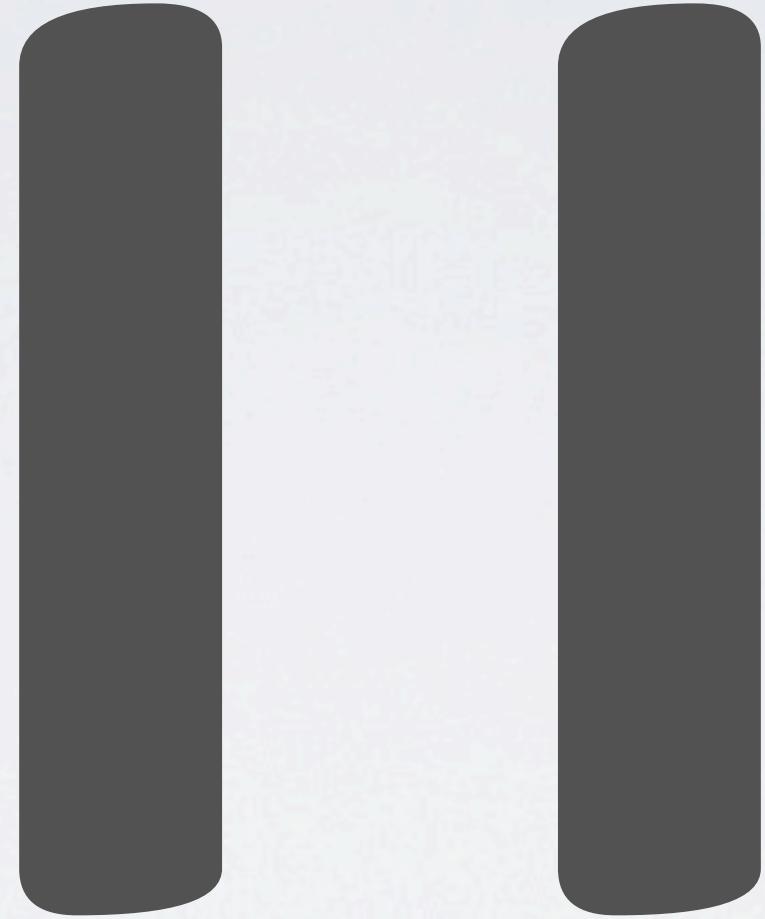


@andrewvida



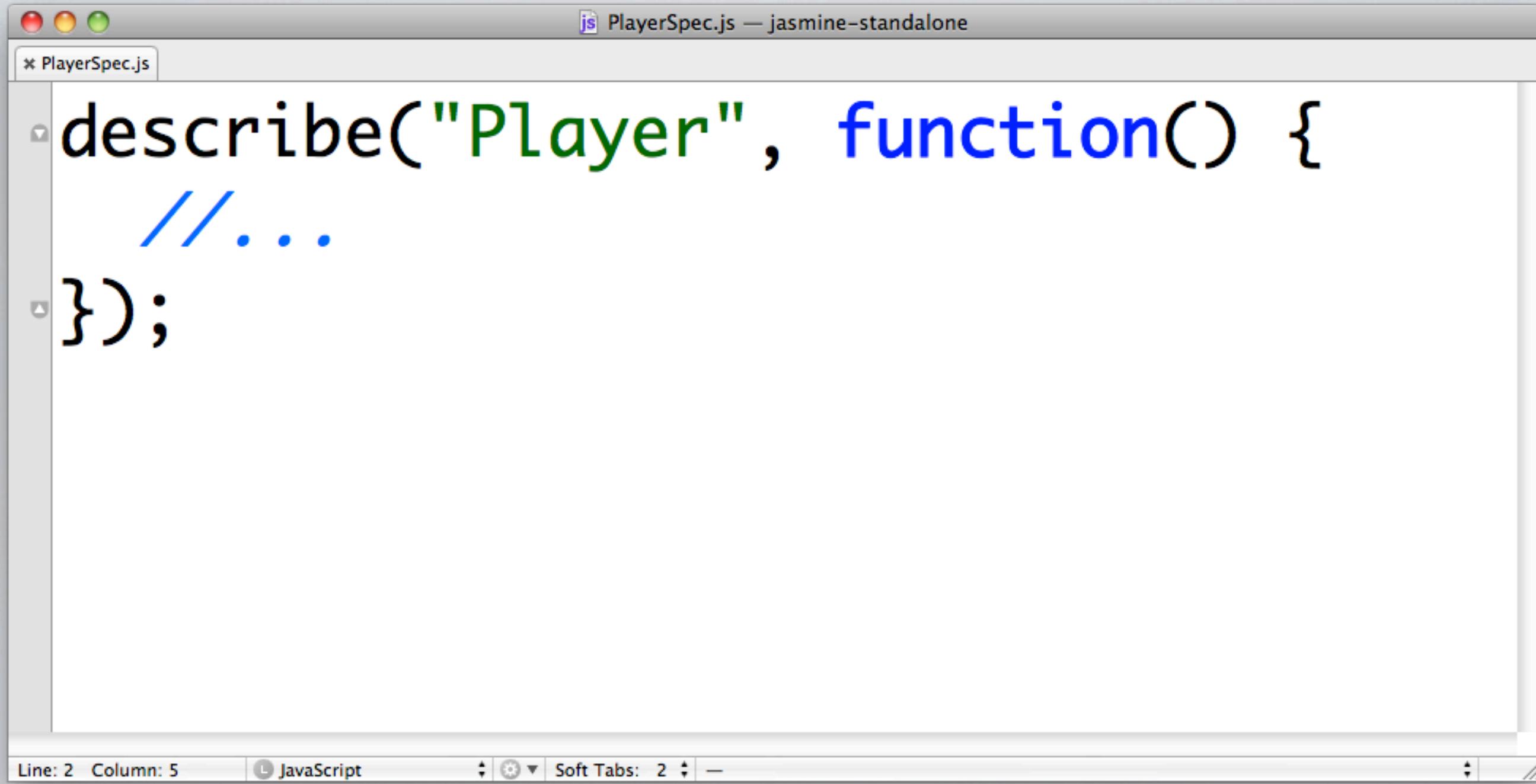
@searls

demo



pause

"vanilla" Jasmine

A screenshot of a Mac OS X-style code editor window titled "PlayerSpec.js — jasmine-standalone". The file tab shows "PlayerSpec.js". The code editor contains the following JavaScript code:

```
describe("Player", function() {  
    //...  
});
```

The code uses color-coded syntax highlighting where "describe" is black, "Player" is green, "function" is blue, and the block comment is blue. The editor's status bar at the bottom shows "Line: 2 Column: 5" and "JavaScript".

Line: 2 Column: 5    JavaScript    Soft Tabs: 2

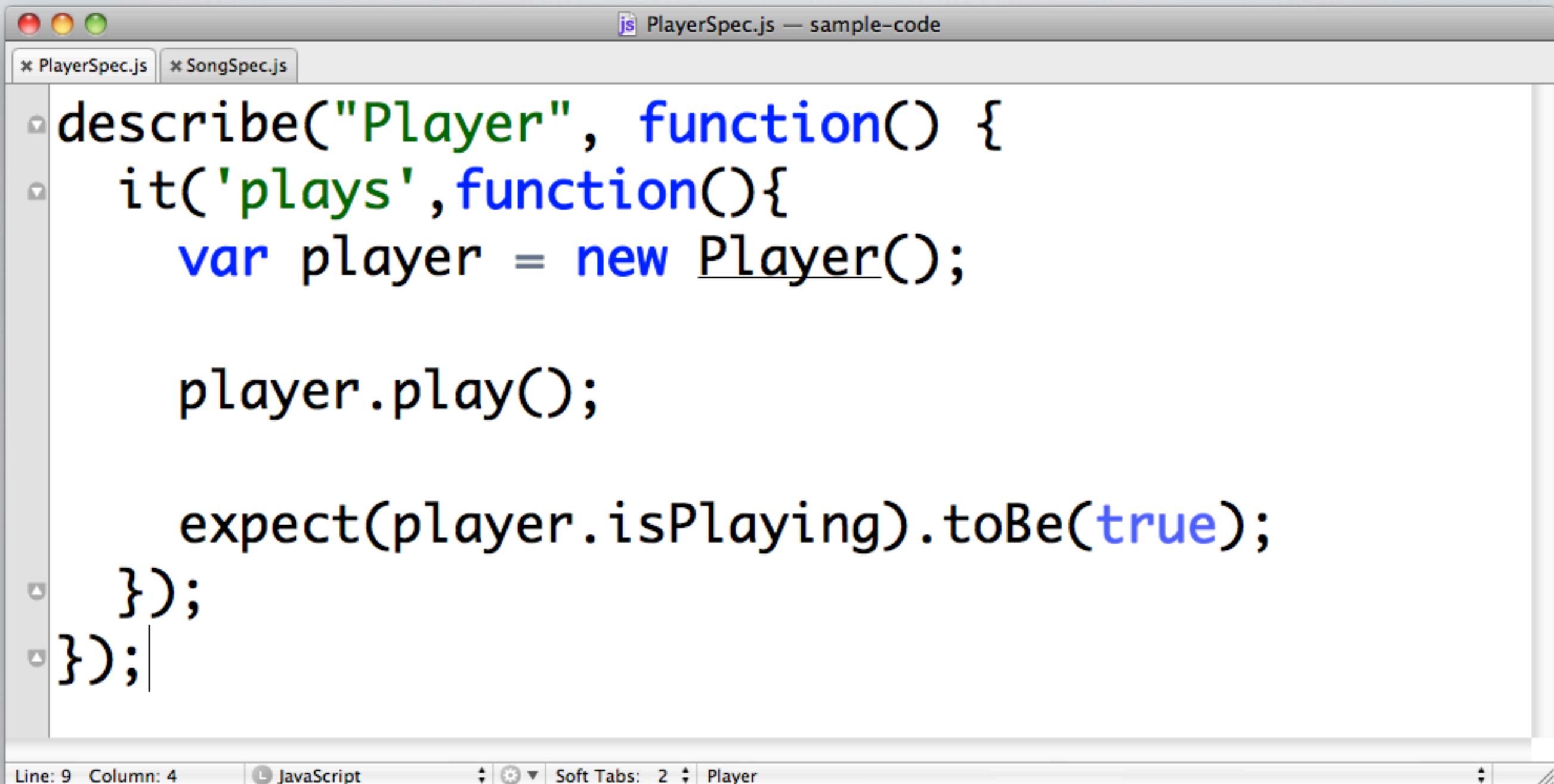
Let's start fresh!

A screenshot of a code editor window titled "PlayerSpec.js — sample-code". The file tab shows "\* PlayerSpec.js". The code contains a single pending example:

```
describe("Player", function() {
  it('plays');
});
```

The code editor interface includes a status bar at the bottom with "Line: 2 Column: 13", a "JavaScript" language selector, and "Soft Tabs: 2" settings.

Pending examples



```
js PlayerSpec.js — sample-code
PlayerSpec.js SongSpec.js
describe("Player", function() {
  it('plays', function(){
    var player = new Player();
    player.play();
    expect(player.isPlaying).toBe(true);
  });
});
```

Line: 9 Column: 4    JavaScript    Soft Tabs: 2    Player

Great expectations

Jasmine Test Runner

file:///Volumes/HD/Users/justin/code/javascript/jasmine-intro/sample-code/SpecRunner.html Google

Jasmine Test Runner

**Jasmine** 1.0.1 revision 1286311016 Show  passed  skipped

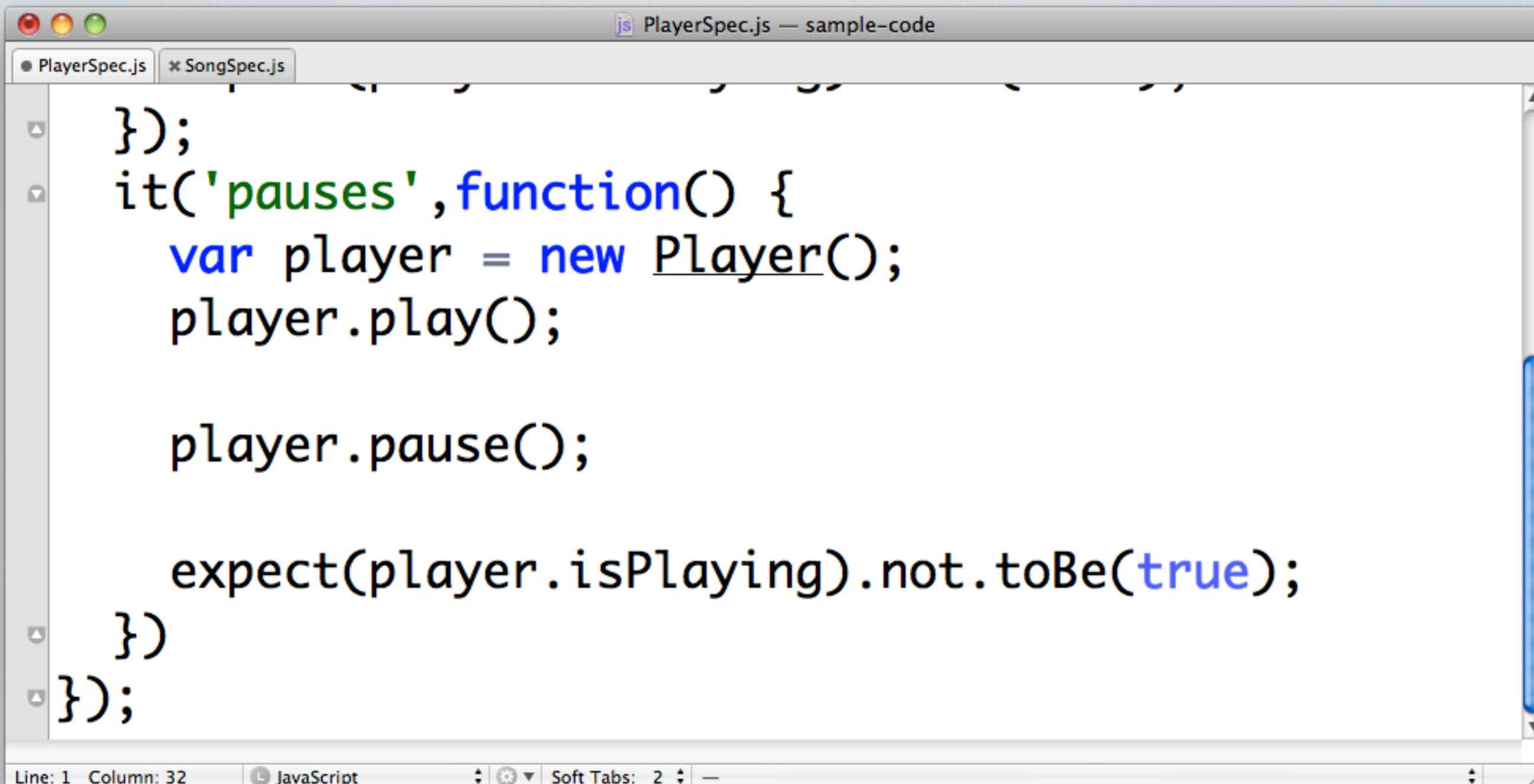
**1 spec, 0 failures in 0.011s** Finished at Wed Jan 12 2011 23:31:21 GMT-0500 (EST) run all

Player run

plays run

A screenshot of a web browser window titled "Jasmine Test Runner". The address bar shows the local file path "file:///Volumes/HD/Users/justin/code/javascript/jasmine-intro/sample-code/SpecRunner.html". The main content area displays the test results. At the top left, it says "Jasmine 1.0.1 revision 1286311016". To the right, there are filter options: "Show" with a checked "passed" checkbox and an unchecked "skipped" checkbox. Below this, a summary line reads "1 spec, 0 failures in 0.011s" followed by the text "Finished at Wed Jan 12 2011 23:31:21 GMT-0500 (EST)". To the right of the summary are two buttons: "run all" and "run". The test results are listed in a table-like structure. The first row has a pink background and contains the word "Player" with a "run" button to its right. The second row has a green background and contains the word "plays" with a "run" button to its right. All elements are rendered in a clean, modern style typical of web-based test runners.

Green!



```
});  
it('pauses', function() {  
  var player = new Player();  
  player.play();  
  
  player.pause();  
  
  expect(player.isPlaying).not.toBe(true);  
})  
});
```

Line: 1 Column: 32    JavaScript    Soft Tabs: 2

# Our second example

Jasmine Test Runner

file:///Volumes/HD/Users/justin/code/javascript/jasmine-intro/sample-code/SpecRunner.htm

Jasmine Test Runner

Jasmine 1.0.1 revision 1286311016

2 specs, 0 failures in 0.01s Finished at Wed Jan 12 2011 23:18:40 GMT-0500 (EST)

Show  passed  skipped

run all

Player

plays

pauses

run

run

run

The screenshot shows the Jasmine Test Runner interface running in a web browser. The title bar says 'Jasmine Test Runner'. The address bar shows the local file path. A tab bar below the title bar has 'Jasmine Test Runner' selected. The main content area displays the test results. At the top left, it says 'Jasmine 1.0.1 revision 1286311016'. Below that, it shows '2 specs, 0 failures in 0.01s' and the date/time 'Finished at Wed Jan 12 2011 23:18:40 GMT-0500 (EST)'. To the right of the date/time are three buttons: 'run all', 'run', and 'run'. The 'run all' button is highlighted with a pink background. The 'Player' section contains two items: 'plays' and 'pauses', each with its own 'run' button to the right. All buttons have a pink background. The overall interface is clean and modern, typical of a developer's tool.

Green!

PlayerSpec.js — sample-code

\* PlayerSpec.js \* SongSpec.js

```
describe("Player", function() {
  it('plays', function(){
    var player = new Player();
    player.play();
    expect(player.isPlaying).toBe(true);
  });
  it('pauses', function() {
    var player = new Player();
    player.play();
    player.pause();
    expect(player.isPlaying).not.toBe(true);
  })
});
```

Line: 17 Column: 4    JavaScript    Soft Tabs: 2    Player

duplication!

PlayerSpec.js — sample-code

\* PlayerSpec.js \* SongSpec.js

```
describe("Player", function() {
  var player;
  beforeEach(function(){
    player = new Player();
    player.play();
  });
  it('plays',function(){
    expect(player.isPlaying).toBe(true);
  });
  describe('pausing',function(){
    beforeEach(function(){
      player.pause();
    });
    it('pauses',function() {
      expect(player.isPlaying).not.toBe(true);
    });
  });
});
```

Line: 12 Column: 22    JavaScript    Soft Tabs: 2    Player

nesting

Jasmine Test Runner

file:///Volumes/HD/Users/justin/code/javascript/jasmine-intro/sample-code/SpecRunner.htm

Jasmine Test Runner

Jasmine 1.0.1 revision 1286311016

Show  passed  skipped

2 specs, 0 failures in 0.014s Finished at Wed Jan 12 2011 23:18:49 GMT-0500 (EST)

run all

Player

pausing

run

pauses

run

plays

run

The screenshot shows a Mac OS X window titled "Jasmine Test Runner". The window has standard OS X controls (red, yellow, green buttons) and a toolbar with icons for back, forward, search, and file operations. The URL in the address bar is "file:///Volumes/HD/Users/justin/code/javascript/jasmine-intro/sample-code/SpecRunner.htm". The main content area displays the test results. At the top left, it says "Jasmine 1.0.1 revision 1286311016". On the right, there are checkboxes for "Show passed" (checked) and "skipped". Below that, it shows "2 specs, 0 failures in 0.014s" and the date "Finished at Wed Jan 12 2011 23:18:49 GMT-0500 (EST)". A "run all" button is on the far right. The test tree starts with "Player", which has three children: "pausing", "pauses", and "plays", each with a "run" button to its right. All elements in the tree are highlighted in green, indicating they have passed.

Still green

# Custom Matchers

```
expect(player.isPlaying).not.toBe(true);
```

```
expect(player).not.toBePlaying();
```

The screenshot shows a Mac OS X application window titled "SpecHelper.js — sample-code". The window contains a tab bar with three tabs: "PlayerSpec.js", "SpecRunner.html", and "SpecHelper.js". The "SpecHelper.js" tab is active. The code editor displays the following JavaScript code:

```
beforeEach(function() {
  this.addMatchers({
    toBePlaying: function() {
      var player = this.actual;
      return player.isPlaying;
    }
  })
});
```

The code defines an "toBePlaying" matcher function within a "beforeEach" block. The "toBePlaying" function returns the value of the "player.isPlaying" property. The code editor interface includes a status bar at the bottom with "Line: 8 Column: 4", "JavaScript", "Soft Tabs: 2", and "toBePlaying".

# Adding a custom matcher

```
PlayerSpec.js — sample-code
PlayerSpec.js

});  
describe('pausing', function(){  
  beforeEach(function(){  
    player.pause();  
  });  
  it('pauses', function() {  
    expect(player).not.toBePlaying();  
  });  
});  
});|
```

Line: 18 Column: 4    JavaScript    Soft Tabs: 2    Player

Using our custom matcher

Jasmine Test Runner

file:///Volumes/HD/Users/justin/code/javascript/jasmine-intro/sample-code/SpecRunner.htm

Jasmine Test Runner

Jasmine 1.0.1 revision 1286311016

Show  passed  skipped

2 specs, 0 failures in 0.014s Finished at Wed Jan 12 2011 23:18:49 GMT-0500 (EST)

run all

Player

pausing

run

pauses

run

plays

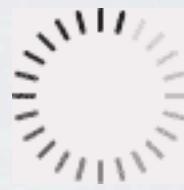
run

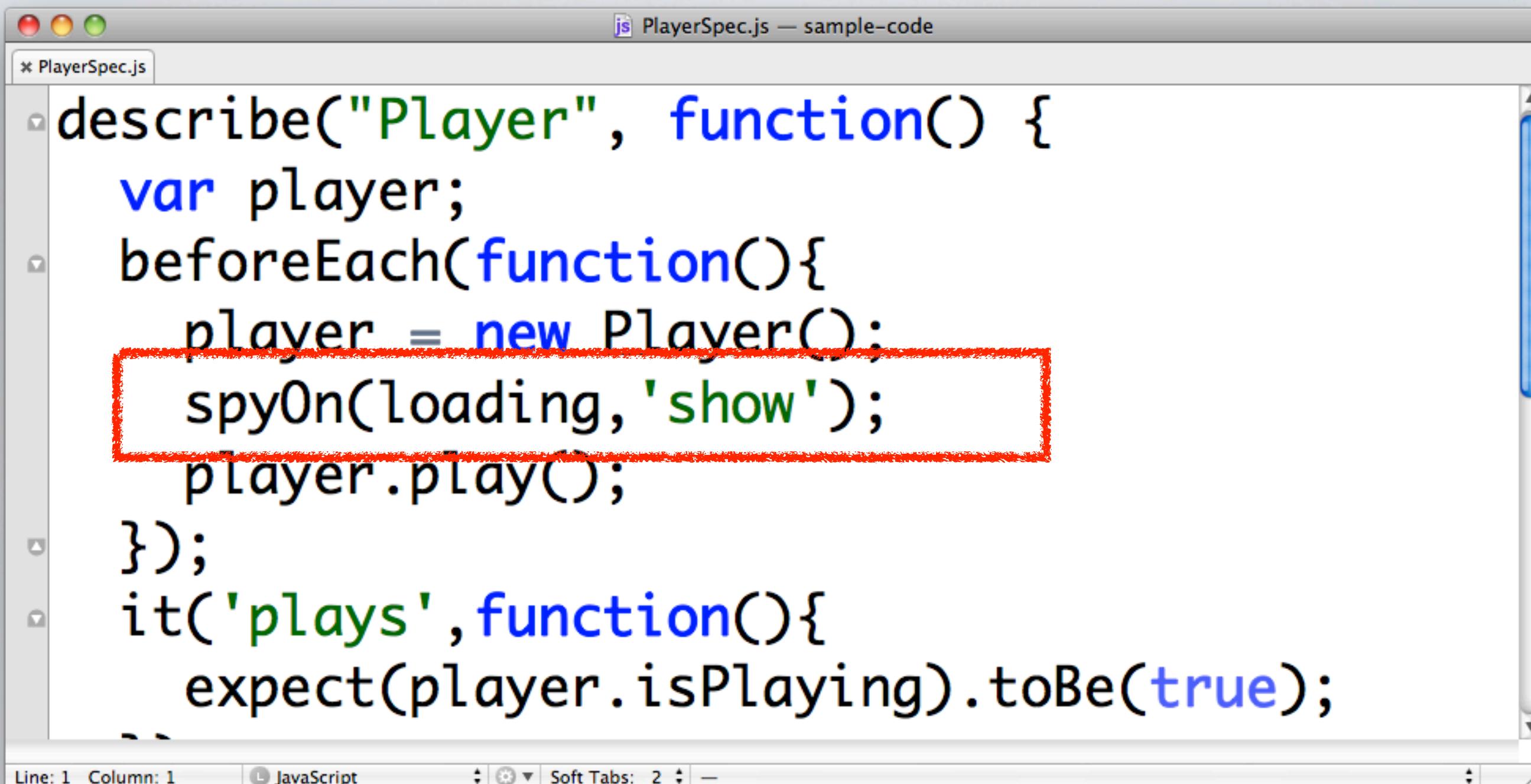
The screenshot shows a Mac OS X window titled "Jasmine Test Runner". The window has standard OS X controls (red, yellow, green buttons) and a toolbar with icons for back, forward, search, and file operations. The URL in the address bar is "file:///Volumes/HD/Users/justin/code/javascript/jasmine-intro/sample-code/SpecRunner.htm". The main content area displays the test results. At the top left, it says "Jasmine 1.0.1 revision 1286311016". On the right, there are checkboxes for "Show passed" (checked) and "skipped". Below that, it shows "2 specs, 0 failures in 0.014s" and the date "Finished at Wed Jan 12 2011 23:18:49 GMT-0500 (EST)". A "run all" button is on the far right. The test tree starts with "Player", which has three children: "pausing", "pauses", and "plays", each with a "run" button to its right. All elements in the tree are highlighted in green, indicating they have passed.

Still green

Spies

Specify a loading animation





```
PlayerSpec.js — sample-code
PlayerSpec.js

describe("Player", function() {
  var player;
  beforeEach(function(){
    player = new Player();
    spyOn(loading, 'show');
    player.play();
  });
  it('plays', function(){
    expect(player.isPlaying).toBe(true);
  });
})
```

Tell Jasmine to spy on loading.show()

```
PlayerSpec.js — sample-code
x PlayerSpec.js

describe("Player", function() {
  var player;
  beforeEach(function(){
    player = new Player();
    spyOn(loading, 'show');
    player.play();
  });
  it('shows a loading indicator', function(){
    expect(loading.show).toHaveBeenCalled();
  });
})
```

Expect the invocation

Jasmine Test Runner

file:///Volumes/Documents/Life/Speaking/jasmine-intro/jasmine-standalone/SpecRunner.html

velesin/ja... gist: 7072... https://gi... velesin/ja... velesin/ja... about:blank Jasmine T... searls/jas... v.gd - UR... +

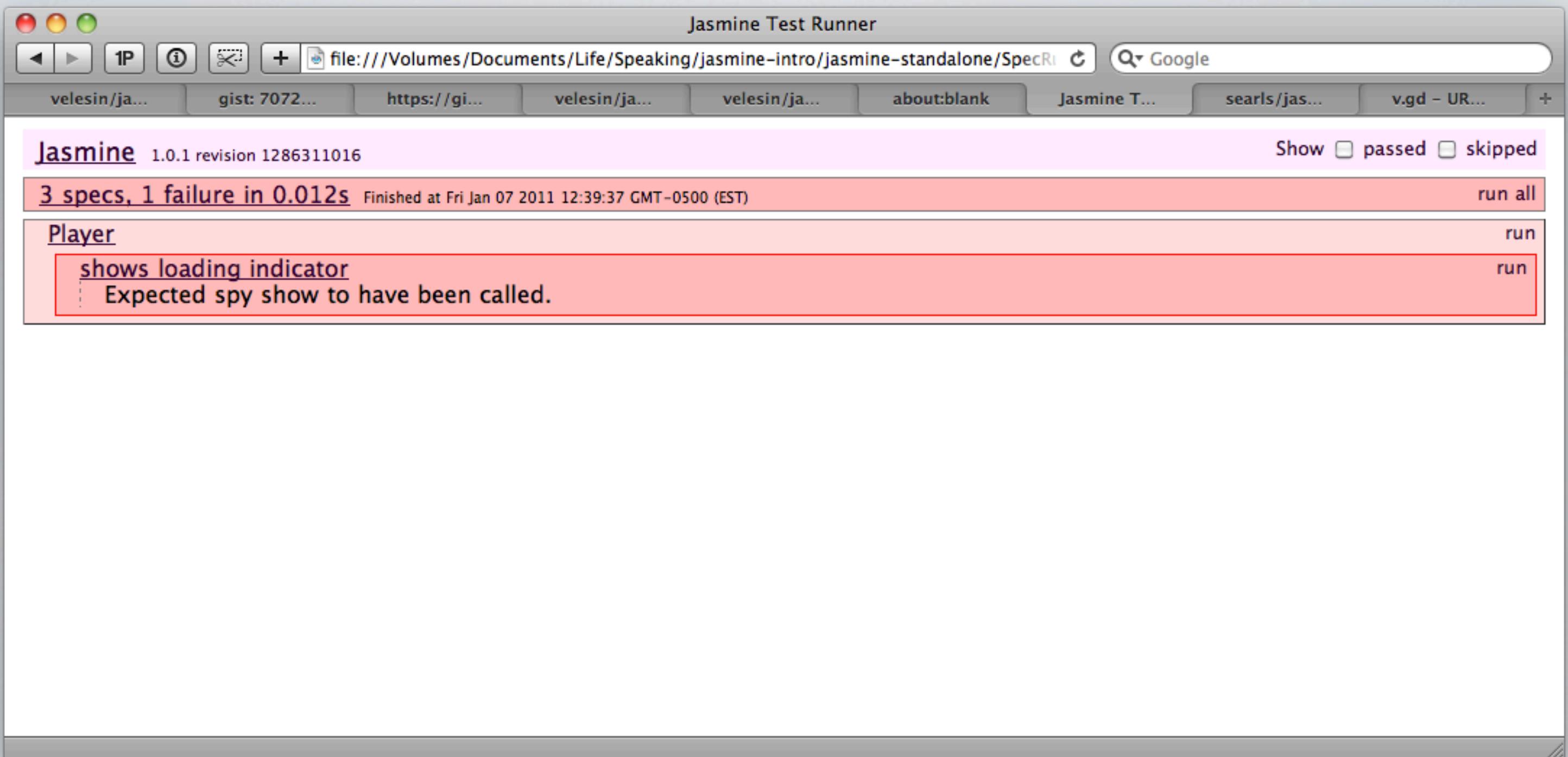
**Jasmine** 1.0.1 revision 1286311016 Show  passed  skipped

**3 specs, 1 failure in 0.012s** Finished at Fri Jan 07 2011 12:39:37 GMT-0500 (EST) run all

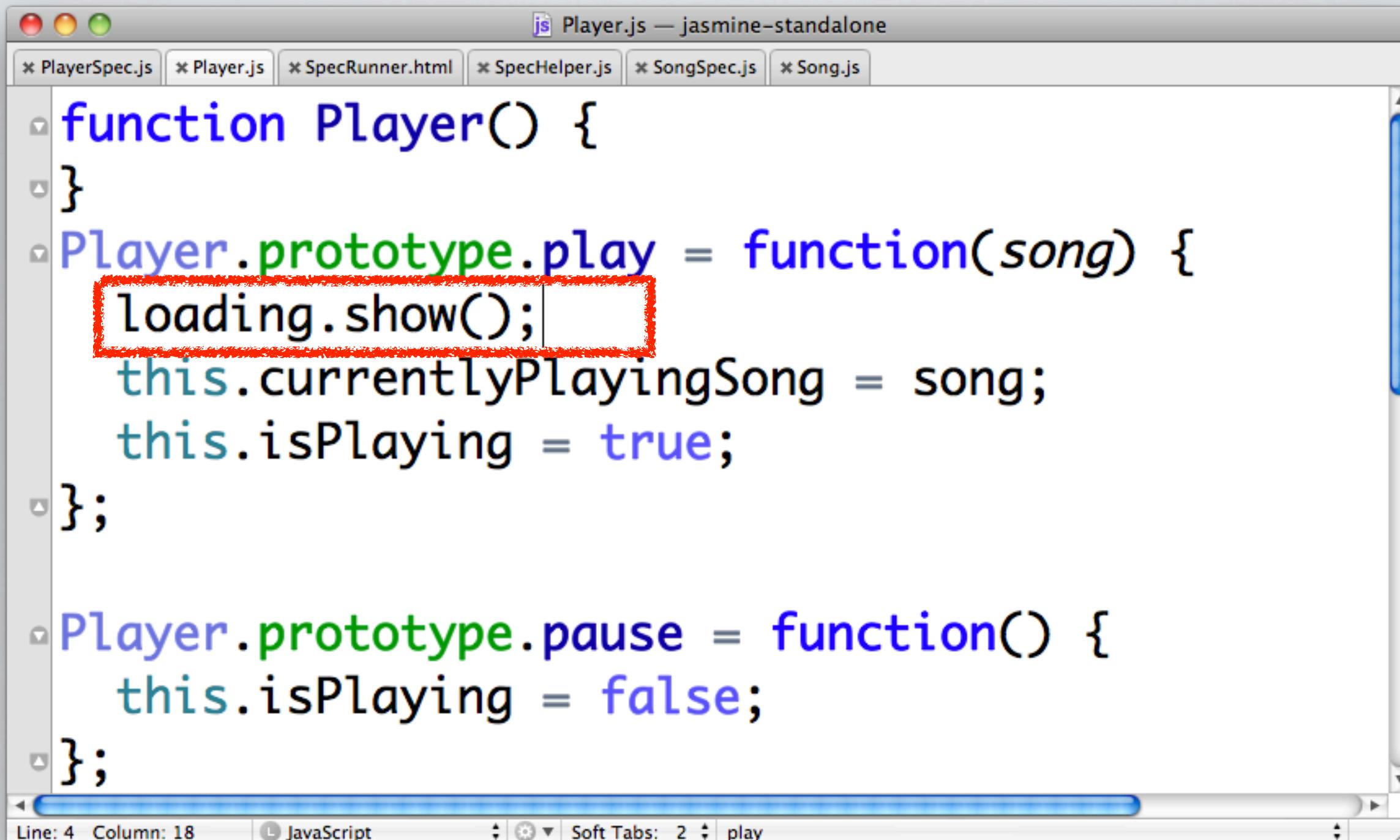
**Player** run

**shows loading indicator** run

Expected spy show to have been called.



Go red



```
function Player() {
}
Player.prototype.play = function(song) {
    loading.show();
    this.currentlyPlayingSong = song;
    this.isPlaying = true;
};

Player.prototype.pause = function() {
    this.isPlaying = false;
};
```

Implement

Jasmine Test Runner

file:///Volumes/Documents/Life/Speaking/jasmine-intro/jasmine-standalone/SpecRi Google

velesin/ja... gist: 7072... https://gi... velesin/ja... velesin/ja... about:blank Jasmine T... searls/jas... v.gd – UR... +

**Jasmine** 1.0.1 revision 1286311016 Show  passed  skipped

**3 specs, 0 failures in 0.011s** Finished at Fri Jan 07 2011 12:41:45 GMT-0500 (EST) run all

Player run

pausing run

is not playing run

shows loading indicator run

is playing run

Go green!

# HTML Fixtures

“How do my specs see my HTML?”

```
1 describe('Song', function(){
2   var song, $currentSongSpan;
3   beforeEach(function() {
4     song = new Song();
5     song.title = "Some Song Title";
6     $currentSongSpan = affix('#currentSong');
7   });
8 });
```

Injecting HTML w/ jasmine-fixture

SongSpec.js

```
1 describe('Song', function(){
2   var song, $currentSongSpan;
3   beforeEach(function() {
4     song = new Song();
5     song.title = "Some Song Title";
6     $currentSongSpan = affix('#currentSong');
7   });
8   it('prints title', function() {
9     song.printTitle();
10    expect($currentSongSpan).toHaveText(song.title);
11  });
12});
```

UNREGISTERED

Jasmine Test Runner

file:///Volumes/Documents/Life/Speaking/jasmine-intro/jasmine-stand Google

velesin... gist: 70... https:// ... velesin... velesin... about:b... Jasmine... searls/j... v.gd - ... +

**Jasmine** 1.0.1 revision 1286311016 Show  passed  skipped

**1 spec, 1 failure in 0.018s** Finished at Fri Jan 07 2011 12:10:35 GMT-0500 (EST) run all

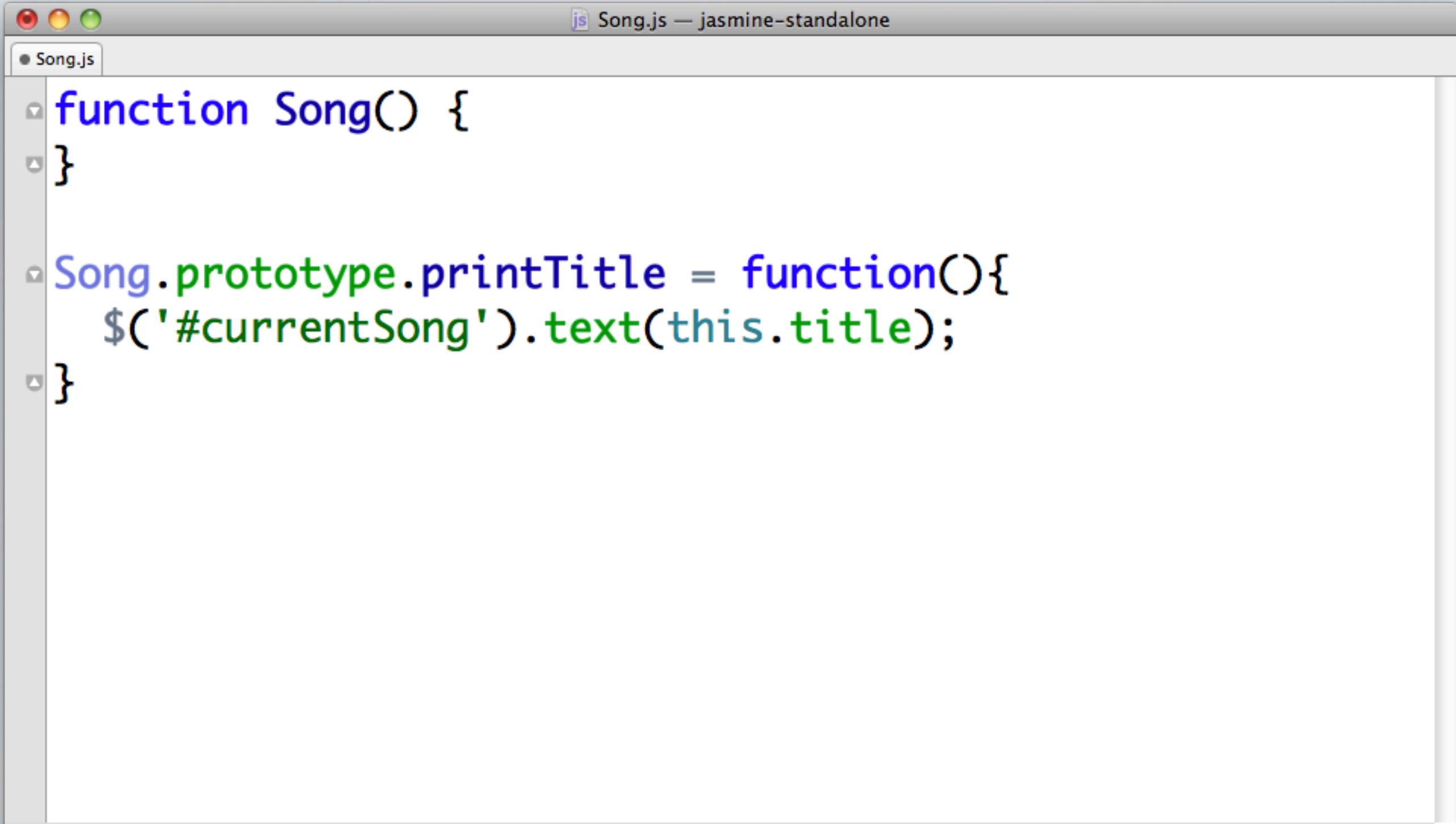
**Song** run

**prints title** run

Expected '<span id="currentSong"></span>' to have text 'Some Song Title'.

The screenshot shows a web browser window with the title "Jasmine Test Runner". The address bar indicates the file is located at "/Volumes/Documents/Life/Speaking/jasmine-intro/jasmine-stand". The main content area shows a test run summary: "1 spec, 1 failure in 0.018s" completed at "Fri Jan 07 2011 12:10:35 GMT-0500 (EST)". Below this, a test named "Song" is listed, which contains a single spec named "prints title". A red box highlights the failure message: "Expected '<span id='currentSong'></span>' to have text 'Some Song Title'." There are "run" buttons next to each section.

Go red



Song.js — jasmine-standalone

Song.js

```
function Song() {  
}  
  
Song.prototype.printTitle = function(){  
    $('#currentSong').text(this.title);  
}
```

Line: 7 Column: 1    L JavaScript    Soft Tabs: 2 printTitle

# Injecting HTML

Jasmine Test Runner

file:///Volumes/Documents/Life/Speaking/jasmine-intro/jasmine-stand Google

velesin... gist: 70... https:// velesin... velesin... about:b... Jasmine... searls/j... v.gd - ... +

**Jasmine** 1.0.1 revision 1286311016 Show  passed  skipped

**1 spec, 0 failures in 0.021s** Finished at Fri Jan 07 2011 12:08:30 GMT-0500 (EST) run all

Player run

pausing run

is not playing run

is playing run

Song run

prints title run

Context	Spec	Status
Player	pausing	pending
	is not playing	pending
	is playing	pending
Song	prints title	pending

Go green!

ON SYNTAX

suppose we want to test:

```
window.Panda = function(){};  
  
Panda.prototype.disposition = function() {  
    return "sad";  
};
```

# ...or in CoffeeScript

```
class window.Panda  
disposition: -> "sad"
```

# xUnit style

```
describe("Panda", function() {  
  it("is sad", function() {  
    var subject = new Panda();  
  
    var result = panda.disposition()  
    expect(result).toBe("sad");  
  });  
});
```

# or in Coffee

```
describe "Panda", ->
  it "is sad", ->
    subject = new Panda()

    result = subject.disposition()

    expect(result).toBe("sad")
```

or, in a nested examples style

```
describe("Panda", function() {
  var subject;
  beforeEach(function() {
    subject = new Panda();
  });
  describe("#disposition", function() {
    var result;
    beforeEach(function() {
      result = subject.disposition()
    });
    it("is sad", function() {
      expect(result).toBe("sad");
    });
  });
});
```

# and expanded

```
describe "Panda", ->  
  beforeEach ->  
    @subject = new Panda
```

```
describe "#disposition", ->  
  beforeEach ->  
    @result = @subject.disposition()  
  it "is sad", ->  
    expect(@result).toBe("sad")
```

# with jasmine-given

```
describe "Panda", ->
```

```
  Given -> @subject = new Panda
```

```
describe "#disposition", ->
```

```
  When -> @result = @subject.disposition()
```

```
  Then -> expect(@result).toBe("sad")
```

Mob programming: fibonacci

# Pair program activity

with a partner, use jasmine to test-drive a "findPrimeFactors" function behaving like this:

```
findPrimeFactors(2) //should return [2]
findPrimeFactors(4) //should return [2]
findPrimeFactors(12) //should return [2,3]
findPrimeFactors(45) //should return [3,5]
```

# In JavaScript/Coffee,

```
[1,2,3] == [1,2,3]
```

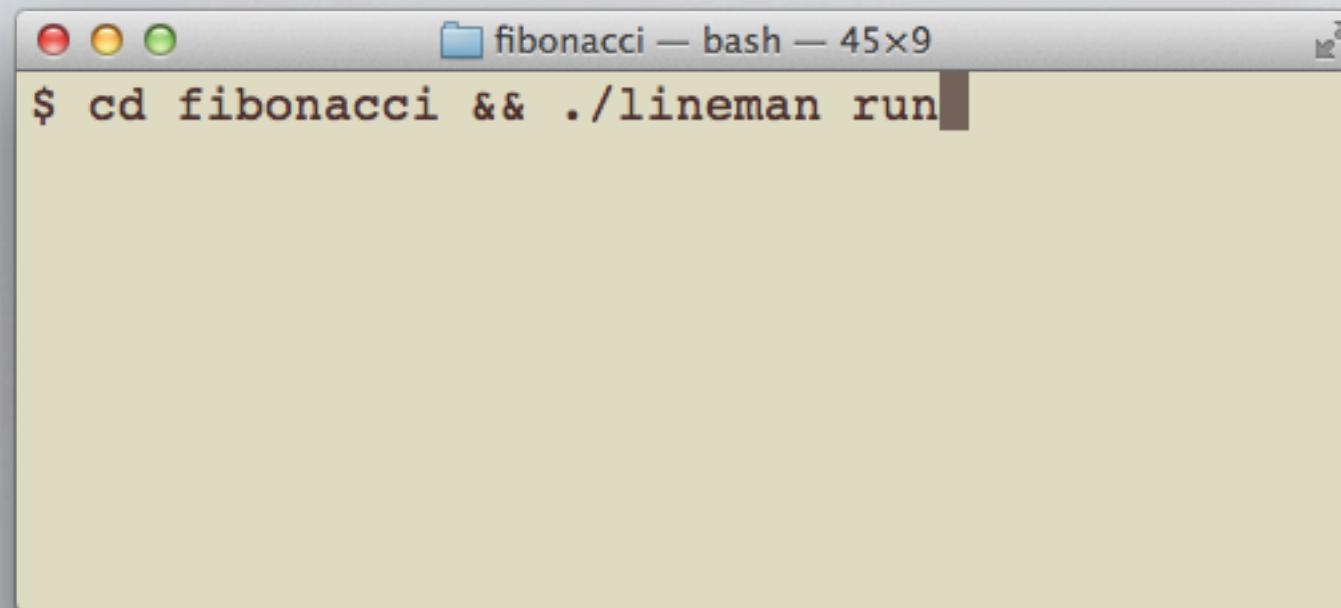
is false! Because identity differs  
instead, try:

```
expect([1,2,3]).toEqual([1,2,3])
```

break is almost over!

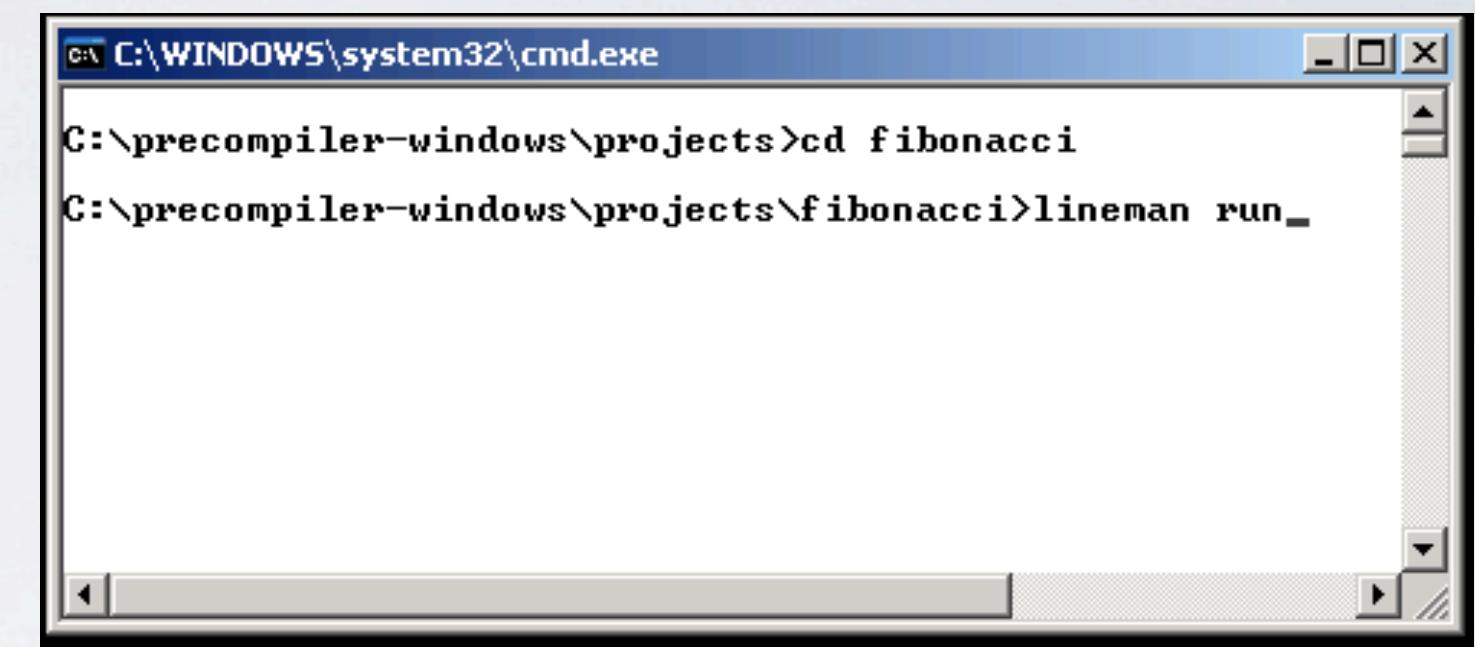
# fibonacci on the web!

**OS X**

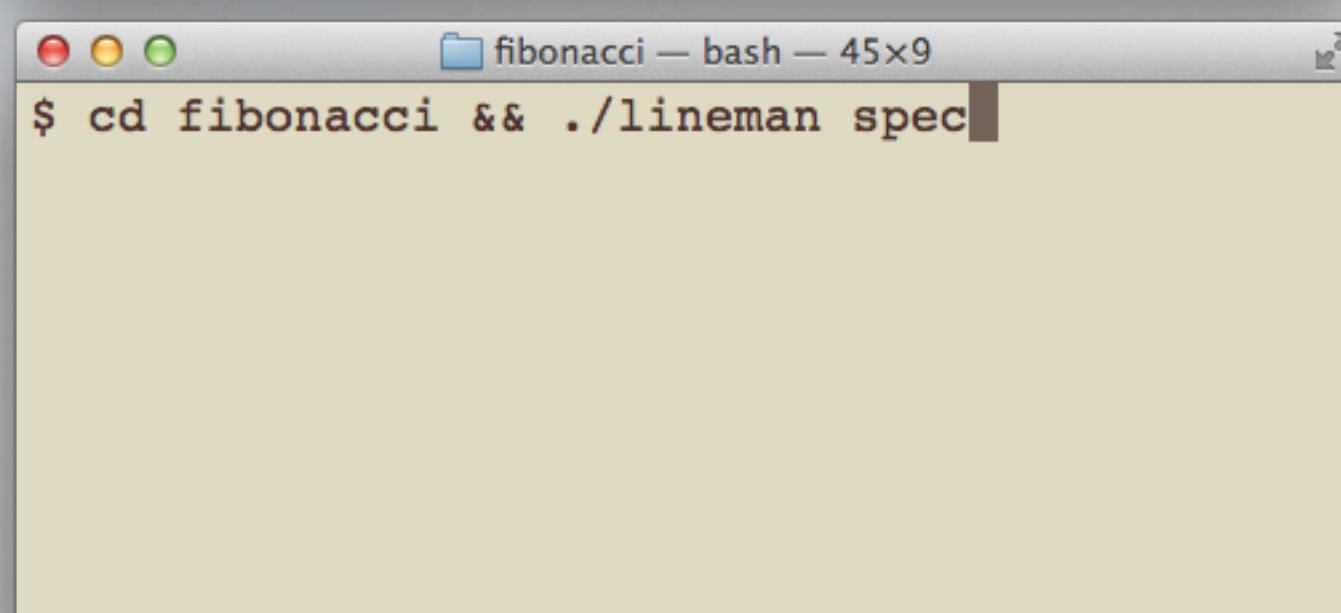


A screenshot of a Mac OS X terminal window titled "fibonacci — bash — 45x9". The window shows the command \$ cd fibonacci && ./lineman run being typed into the terminal.

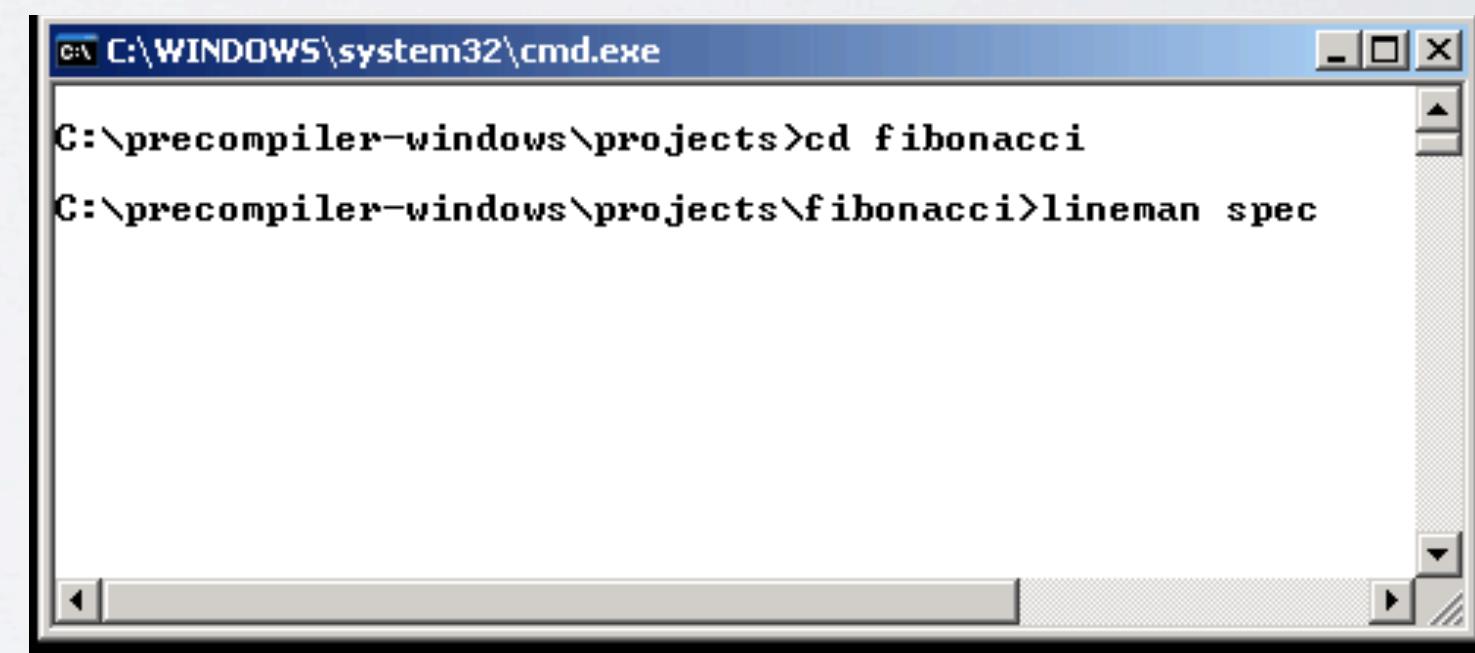
**Windows**



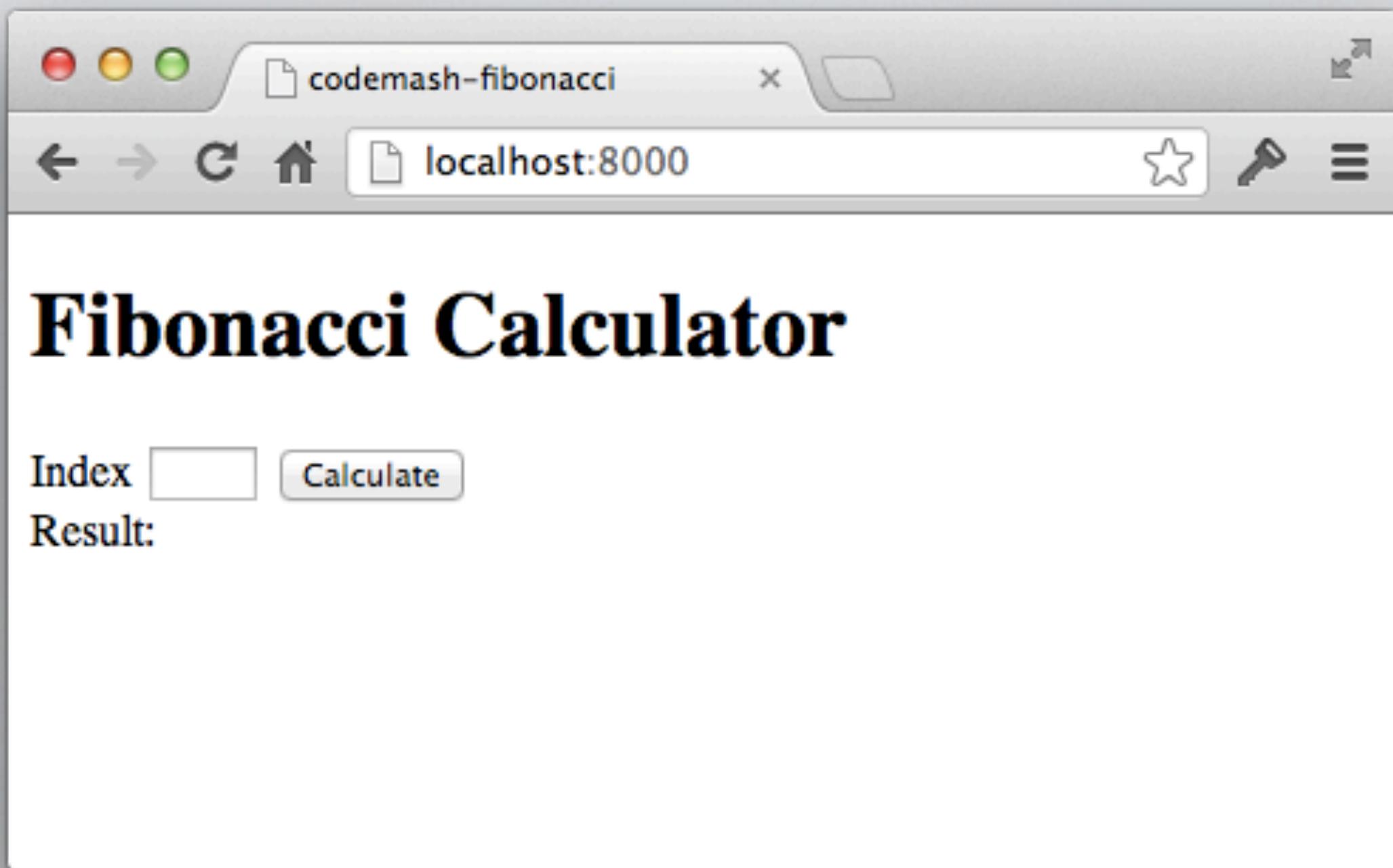
A screenshot of a Windows cmd.exe window titled "C:\WINDOWS\system32\cmd.exe". The window shows the commands C:\precompiler-windows\projects>cd fibonacci and C:\precompiler-windows\projects\fibonacci>lineman run\_ being typed into the command line.



A screenshot of a Mac OS X terminal window titled "fibonacci — bash — 45x9". The window shows the command \$ cd fibonacci && ./lineman spec being typed into the terminal.



A screenshot of a Windows cmd.exe window titled "C:\WINDOWS\system32\cmd.exe". The window shows the commands C:\precompiler-windows\projects>cd fibonacci and C:\precompiler-windows\projects\fibonacci>lineman spec being typed into the command line.



Index

Calculate

Result:

# test-drive code that:

1. User types an index into the form field
2. User clicks the "calculate" button
3. JavaScript sends a POST to "/fibonacci" with the `index` parameter
4. JavaScript handles server response (an integer) by printing it into the <span> with the selector ".result"

```
$ .post '/fibonacci', {index: 4}, (response) ->  
  console.log(response)
```

# EXAMPLE AJAX

# Resources

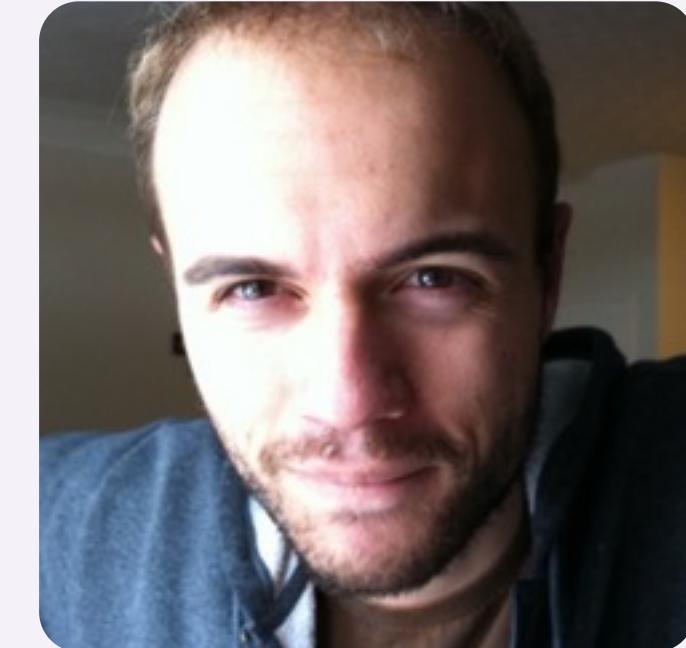
- <http://tryjasmine.com>
- <http://pivotal.github.com/jasmine/>
- <http://github.com/searls/jasmine-fixture>
- <http://github.com/searls/jasmine-given>
- <http://github.com/searls/jasmine-stealth>
- <http://github.com/testdouble/lineman>

# test double

[hello@testdouble.com](mailto:hello@testdouble.com)



@andrewvida



@searls