## Primer Feature Test

* Crear el archivo feature\_spec.js
* Hablar sobre el framework y como agrupa y organiza los tests.
* Crear el sgte test

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | describe("Features", *function*() { | | 2 | it("should let enter the secret word", *function*() { | | 3 | *var* wordInput = $('<input>'); | | 4 | *var* wordForm = $('<button>'); | | 5 | *var* wordOutput = $('<div>'); | | 6 | *var* game = new Game({ | | 7 | wordInput: wordInput, | | 8 | wordForm: wordForm, | | 9 | wordOutput: wordOutput | | 10 | }); | | 11 |  | | 12 | wordInput.val('cocodrillo'); | | 13 | wordForm.submit(); | | 14 | wordOutput.html().should.equal('\_\_\_\_\_\_\_\_\_\_'); | | 15 | }); | | 16 | }); | |  |

* Copiar y pegar el feature\_runner.html

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | <!DOCTYPE HTML> | | 2 | <html lang="en-US"> | | 3 | <head> | | 4 | <title>Test Runner Template</title> | | 5 | <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> | | 6 | <!-- MOCHA STYLE SHEET --> | | 7 | <link rel="stylesheet" href="../css/mocha.css"> | | 8 | </head> | | 9 | <body> | | 10 | <div id="mocha"></div> | | 11 | <!-- MOCHA LIBRARY--> | | 12 | <script type="text/javascript" src="../lib/mocha-1.4.2.js"></script> | | 13 | <!-- OTRAS LIBRERÍAS--> | | 14 | <script type="text/javascript" src="../lib/chai-1.2.0.js"></script> | | 15 | <script type="text/javascript" src="../lib/jquery-1.8.0.js"></script> | | 16 |  | | 17 | <!-- MOCHA y CHAI CONFIGURATION --> | | 18 | <script type="text/javascript"> | | 19 | /\* | | 20 | Mocha nos permite utilizar diferentes DSL(sintaxis): | | 21 | - BDD (Ejm: RSpec) | | 22 | - TDD (Ejm: JUnit) | | 23 | - QUnit (Similar a otra JS Testing Framework) | | 24 | \*/ | | 25 | mocha.setup('bdd'); | | 26 | chai.should(); | | 27 | </script> | | 28 |  | | 29 | <!-- TEST FILES (SPECS) --> | | 30 | <script type="text/javascript" src="feature\_spec.js"></script> | | 31 |  | | 32 | <!-- SRC FILES (SPECS) --> | | 33 |  | | 34 | <!-- EJECUTAR MOCHA --> | | 35 | <script type="text/javascript"> | | 36 | mocha.run(); | | 37 | </script> | | 38 | </body> | | 39 | </html> | |  |

* Ver que el Test ha fallado.
* Crear el archivo src/game.js y agregarlo al featurerunner.
* Implementar la clase hasta que se vea únicamente error del aserto.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | *function* Game(*attributes*) { | | 2 | this.init(attributes); | | 3 | } | | 4 |  | | 5 | *Game*.prototype = { | | 6 | init: *function*(*attributes*) { | | 7 | this.wordInput= attributes.wordInput; | | 8 | this.wordForm= attributes.wordForm; | | 9 | this.wordOutput=attributes.wordOutput; | | 10 | } | | 11 | }; | |  |

* Crear el archivo spec/game\_spec.js
* Copiar el archivo spec/feauture\_runner.html, pegarlo y renombrarlo a spec/unit\_runner.html.
  + Internamente remplazar el title por Unit Test Runner
  + Cambiar la referencia a feature\_spec.js por game\_spec.js
* Escribir el primer test.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | describe("game", *function*() { | | 2 | it("should show format secret word with dashes", *function*() { | | 3 | *var* game=new Game({ | | 4 | wordinput: $('<input>'), | | 5 | wordForm: $('<input>'), | | 6 | wordOutput: $('<div>') | | 7 | }); | | 8 |  | | 9 | *var* word = game.formatSecretWord('gato'); | | 10 |  | | 11 | word.should.equal('\_\_\_\_'); | | 12 | }); | | 13 | }); | |  |

* Hacer pasar el test.

|  |  |
| --- | --- |
| 5 | *Game*.prototype = { |
| 6 | init: *function*(*attributes*) { |
| 7 | this.wordInput = attributes.wordInput; |
| 8 | this.wordForm = attributes.wordForm; |
| 9 | this.wordOutput = attributes.wordOutput; |
| 10 | }, |
| 11 | formatSecretWord: *function*(*secretWord*) { |
| 12 | *var* word = ''; |
| 13 | for(*var* i = 0; i < secretWord.length; i++) |
| 14 | word += '\_'; |
| 15 |  |
| 16 | return word; |
| 17 | } |
| 18 | }; |

* Crear el segundo test. (podemos crear un test muy similar al feature test pero más simple o solo completar el código para pasar directamente el feature test.

|  |  |
| --- | --- |
| 14 | it("should show the secret word", *function*() { |
| 15 | *var* wordInput = $('<input>'); |
| 16 | *var* wordForm = $('<button>'); |
| 17 | *var* wordOutput = $('<div>'); |
| 18 | *var* game = new Game({ |
| 19 | wordInput: wordInput, |
| 20 | wordForm: wordForm, |
| 21 | wordOutput: wordOutput |
| 22 | }); |
| 23 |  |
| 24 | wordInput.val('a'); |
| 25 | wordForm.submit(); |
| 26 |  |
| 27 | wordOutput.html().should.equal('\_'); |
| 28 | }); |

* Hacemos pasar el test.
  + Primero creamos el onAccept.
  + Asignamos el evento sin proxy
  + Cambiamos por proxy.

|  |  |
| --- | --- |
| 5 | *Game*.prototype = { |
| 6 | init: *function*(*attributes*) { |
| 7 | this.wordInput = attributes.wordInput; |
| 8 | this.wordForm = attributes.wordForm; |
| 9 | this.wordOutput = attributes.wordOutput; |
| 10 | this.wordForm.click($.proxy(this.onFormSubmit, this)); |
| 11 | },  ……… |
| 19 | onFormSubmit: *function*() { |
| 20 | *var* secretWord = this.wordInput.val(); |
| 21 | *var* formatedSecretWord = this.formatSecretWord(secretWord); |
| 22 | this.wordOutput.html(formatedSecretWord); |
| 23 | } |
| 24 | }; |

* Al ejecutar los test, aparecerá un parpadeo.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 20 | onFormSubmit: *function*(*e*) { | | 21 | e.preventDefault(); | | 22 | this.secretWord = this.wordInput.val(); | | 23 | this.formatedSecretWord = this.formatSecretWord(this.secretWord); | | 24 | this.wordOutput.html(this.formatedSecretWord); | | 25 | this.createButtons(); | | 26 | }, | |

* Refactorizamos los unit tests.

|  |  |
| --- | --- |
| 1 | describe("game", *function*() { |
| 2 | *var* wordInput; |
| 3 | *var* wordForm; |
| 4 | *var* wordOutput; |
| 5 | *var* game; |
| 6 |  |
| 7 | beforeEach(*function*() { |
| 8 | wordInput = $('<input>'); |
| 9 | wordForm = $('<button>'); |
| 10 | wordOutput = $('<div>'); |
| 11 | game = new Game({ |
| 12 | wordInput: wordInput, |
| 13 | wordForm: wordForm, |
| 14 | wordOutput: wordOutput |
| 15 | }); |
| 16 | }); |
| 17 |  |
| 18 | it("should format the secret word with dashes", *function*() { |
| 19 | *var* word = game.formatSecretWord('gato'); |
| 20 |  |
| 21 | word.should.equal('\_\_\_\_'); |
| 22 | }); |
| 23 |  |
| 24 | it("should show the secret word", *function*() { |
| 25 | wordInput.val('a'); |
| 26 | wordForm.submit(); |
| 27 |  |
| 28 | wordOutput.html().should.equal('\_'); |
| 29 | }); |
| 30 | }); |

## Code Inspection JSHINT

* Instalación

Npm install –g grunt

* Creación de grunt.js

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | /\*global module:false\*/ | | 2 | *module*.exports = *function*(*grunt*) { | | 3 |  | | 4 | // Build configuration. | | 5 | grunt.initConfig({ | | 6 | lint: { | | 7 | files: ['grunt.js', 'src/\*.js', 'spec/\*.js'] | | 8 | }, | | 9 | jshint: { | | 10 | options: { | | 11 | curly: true, | | 12 | eqeqeq: true, | | 13 | immed: true, | | 14 | latedef: true, | | 15 | newcap: true, | | 16 | noarg: true, | | 17 | sub: true, | | 18 | undef: true, | | 19 | boss: true, | | 20 | eqnull: true, | | 21 | browser: true, | | 22 | jquery: true, | | 23 | trailing:true, | | 24 | noempty:true | | 25 | // strict:true, | | 26 | }, | | 27 | globals: { | | 28 | afterEach: false, | | 29 | beforeEach: false, | | 30 | describe: false, | | 31 | expect: false, | | 32 | it: false | | 33 | } | | 34 | } | | 35 | }); | | 36 |  | | 37 | // Default task. | | 38 | grunt.registerTask('default', 'lint'); | | 39 | }; | |

* Ejecutar grunt.
* Corregir los errores
  + Al inicio de las specs agregar: */\*global Game:false\*/*
  + Método formatSecretWord (agregarle paréntesis):  
     *for(var i = 0; i < secretWord.length; i++) { word += '\_'; }*
* Pasar grunt.

## Segundo Feature Test

* Refactorizamos los feature tests

|  |  |
| --- | --- |
| 2 | *var* wordInput; |
| 3 | *var* wordForm; |
| 4 | *var* wordOutput; |
| 5 | *var* game; |
| 6 |  |
| 7 | beforeEach(*function*() { |
| 8 | wordInput = $('<input>'); |
| 9 | wordForm = $('<button>'); |
| 10 | wordOutput = $('<div>'); |
| 11 | game = new Game({ |
| 12 | wordInput: wordInput, |
| 13 | wordForm: wordForm, |
| 14 | wordOutput: wordOutput |
| 15 | }); |
| 16 | }); |

* Creamos el segundo Test

|  |  |
| --- | --- |
| 27 | it("should be able to play", *function*() { |
| 28 | wordInput.val('cocodrillo'); |
| 29 | wordForm.submit(); |
| 30 |  |
| 31 | $('#o',buttonList).click(); |
| 32 | $('#d',buttonList).click(); |
| 33 |  |
| 34 | wordOutput.html().should.equal('\_o\_od\_\_\_\_o'); |
| 35 | }); |

* Agregamos el elemento buttonList en el beforeeach.

|  |  |
| --- | --- |
| 2 | *var* wordInput; |
| 3 | *var* wordForm; |
| 4 | *var* wordOutput; |
| 5 | *var* buttonList; |
| 6 | *var* game; |
| 7 |  |
| 8 | beforeEach(*function*() { |
| 9 | wordInput = $('<input>'); |
| 10 | wordForm = $('<button>'); |
| 11 | wordOutput = $('<div>'); |
| 12 | game = new Game({ |
| 13 | wordInput: wordInput, |
| 14 | wordForm: wordForm, |
| 15 | wordOutput: wordOutput, |
| 16 | buttonList: buttonList |
| 17 | }); |
| 18 | }); |

* Vemos el test fallar.
* **Creamos tests de CREACIÓN DE BOTONES dentro de un nuevo describe.**

|  |  |
| --- | --- |
| 34 | describe("create letter buttons", *function*() { |
| 35 | it("should show the letter a", *function*() { |
| 36 | wordInput.val('gato'); |
| 37 | wordForm.submit(); |
| 38 |  |
| 39 | $('button:contains("a")', buttonList).should.have.length(1); |
| 40 | }); |
| 41 | }); |

* Hacemos pasar el test.

|  |  |
| --- | --- |
| 5 | *Game*.prototype = { |
| 6 | init: *function*(*attributes*) { |
| 7 | this.wordInput = attributes.wordInput; |
| 8 | this.wordForm = attributes.wordForm; |
| 9 | this.wordOutput = attributes.wordOutput; |
| 10 | this.buttonList = attributes.buttonList; |
| 11 | this.wordForm.click($.proxy(this.onFormSubmit, this)); |
| 12 | },  ……… |
| 20 | onFormSubmit: *function*() { |
| 21 | *var* secretWord = this.wordInput.val(); |
| 22 | *var* formatedSecretWord = this.formatSecretWord(secretWord); |
| 23 | this.wordOutput.html(formatedSecretWord); |
| 24 | this.createButtons(); |
| 25 | }, |
| 26 | createButtons: *function*() { |
| 27 | *var* buttonA='<button>a</button>'; |
| 28 | this.buttonList.append(buttonA); |
| 29 | } |
| 30 | }; |

* Escribimos un nuevo test.

|  |  |
| --- | --- |
| 42 | it("should show the letter z", *function*() { |
| 43 | wordInput.val('gato'); |
| 44 | wordForm.submit(); |
| 45 |  |
| 46 | $('button:contains("z")', buttonList).should.have.length(1); |
| 47 | }); |

* Hacemos pasar el test.

|  |  |
| --- | --- |
| 26 | createButtons: *function*() { |
| 27 | *var* buttonA='<button>'+*String*.fromCharCode(97)+'</button>'; |
| 28 | this.buttonList.append(buttonA); |
| 29 |  |
| 30 | *var* buttonZ='<button>'+*String*.fromCharCode(122)+'</button>'; |
| 31 | this.buttonList.append(buttonZ); |
| 32 | } |

* Escribimos un nuevo test.

|  |  |
| --- | --- |
| 49 | it("should show the letter b", *function*() { |
| 50 | wordInput.val('gato'); |
| 51 | wordForm.submit(); |
| 52 |  |
| 53 | $('button:contains("b")', buttonList).should.have.length(1); |
| 54 | }); |

* Hacemos pasar el test.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 26 | createButtons: *function*() { | | 27 | for(*var* i = 97; i <= 122; i++) { | | 28 | *var* button = '<button>' + *String*.fromCharCode(i) + '</button>'; | | 29 | this.buttonList.append(button); | | 30 | } | | 31 | } | |  |

* **Creamos los test de JUGAR CON LETRAS.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 57 | describe("try letters", *function*() | | 58 | it("should show the letter when the letter match", *function*() { | | 59 | wordInput.val("g"); | | 60 | wordForm.submit(); | | 61 |  | | 62 | game.tryLetter("g"); | | 63 | wordOutput.html().should.equal('g'); | | 64 | }); | | 65 | }); | |

* Hacemos pasar el test.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 32 | tryLetter: *function*(*letter*) { | | 33 | this.wordOutput.html(letter); | | 34 | } | |

* Creamos el siguiente test.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 66 | it("should show underscore when the letter does not match", *function*() { | | 67 | wordInput.val("g"); | | 68 | wordForm.submit(); | | 69 |  | | 70 | game.tryLetter("e"); | | 71 | wordOutput.html().should.equal('\_'); | | 72 | }); | |

* Hacemos pasar el test.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 20 | onFormSubmit: *function*() { | | 21 | this.secretWord = this.wordInput.val(); | | 22 | *var* formatedSecretWord = this.formatSecretWord(this.secretWord); | | 23 | this.wordOutput.html(formatedSecretWord); | | 24 | this.createButtons(); | | 25 | },  …… | | 32 | tryLetter: *function*(*letter*) { | | 33 | if (this.secretWord.indexOf(letter)!=-1) { | | 34 | this.wordOutput.html(letter); | | 35 | } | | 36 | } | |

* Refactorizamos los tests.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 52 | *function* playWith(*secretWord*) { | | 53 | wordInput.val(secretWord); | | 54 | wordForm.submit(); | | 55 | } | | 56 |  | | 57 | describe("try letters", *function*() { | | 58 | it("should show the letter 'g' when the letter match", *function*() { | | 59 | playWith("g"); | | 60 |  | | 61 | game.tryLetter("g"); | | 62 | wordOutput.html().should.equal('g'); | | 63 | }); | | 64 | it("should show underscore when the letter does not match", *function*() { | | 65 | playWith("g"); | | 66 |  | | 67 | game.tryLetter("e"); | | 68 | wordOutput.html().should.equal('\_'); | | 69 | }); | | 70 | }); | |

* Creamos el siguiente test.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 70 | it("should show only the letters that match", *function*() { | | 71 | playWith("ga"); | | 72 |  | | 73 | game.tryLetter("g"); | | 74 | wordOutput.html().should.equal('g\_'); | | 75 | }); | |

* Hacemos pasar el test( Hacerlo por pasos: Primero this.formated(0,0))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 20 | onFormSubmit: *function*() { | | 21 | this.secretWord = this.wordInput.val(); | | 22 | this.formatedSecretWord = this.formatSecretWord(this.secretWord); | | 23 | this.wordOutput.html(this.formatedSecretWord); | | 24 | this.createButtons(); | | 25 | },  …… | | 32 | tryLetter: *function*(*letter*) { | | 33 | if (this.secretWord.indexOf(letter)!=-1) { | | 34 | *var* word=letter + this.formatedSecretWord.substr(1); | | 35 | this.wordOutput.html(word); | | 36 | } | | 37 | } | |

* Creamos el siguiente test.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 76 | it("should show the letter that match in different positions", *function*() { | | 77 | playWith("ga"); | | 78 |  | | 79 | game.tryLetter("a"); | | 80 | wordOutput.html().should.equal('\_a'); | | 81 | }); | |

* Hacemos pasar el test (Hacerlo de a poquito: primero this.formated.substr(0,0) y recién el cambio.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 32 | tryLetter: *function*(*letter*) { | | 33 | *var* index = this.secretWord.indexOf(letter); | | 34 | if(index != -1) { | | 35 | *var* word = this.formatedSecretWord.substr(0, index) + | | 36 | letter + | | 37 | this.formatedSecretWord.substr(index + 1); | | 38 | this.wordOutput.html(word); | | 39 | } | | 40 | } | |

* Creamos el siguiente test.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 82 | it("should show many letters that match", *function*() { | | 83 | playWith("gato"); | | 84 |  | | 85 | game.tryLetter("g"); | | 86 | game.tryLetter("a"); | | 87 | wordOutput.html().should.equal('ga\_\_'); | | 88 | }); | |

* Hacemos pasar el test.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 32 | tryLetter: *function*(*letter*) { | | 33 | *var* index = this.secretWord.indexOf(letter); | | 34 | if(index != -1) { | | 35 | this.formatedSecretWord = this.formatedSecretWord.substr(0, index) + | | 36 | letter + | | 37 | this.formatedSecretWord.substr(index + 1); | | 38 | this.wordOutput.html(this.formatedSecretWord); | | 39 | } | | 40 | } | |

* Creamos el siguiente test.
* Hacemos pasar el test (crear el for encima de todo, reemplazar i por index)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 32 | tryLetter: *function*(*letter*) { | | 33 | for(*var* index = 0; index < this.secretWord.length; index++) { | | 34 | if(letter == this.secretWord[index]) { | | 35 | this.formatedSecretWord = this.formatedSecretWord.substr(0, index) + | | 36 | letter + | | 37 | this.formatedSecretWord.substr(index + 1); | | 38 | this.wordOutput.html(this.formatedSecretWord); | | 39 | } | | 40 | } | | 41 | } | |

* Creamos un test para integrar todo.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 97 | it("should show the result after try a letter", *function*(){ | | 98 | playWith("a"); | | 99 |  | | 100 | $('button:contains("a")', buttonList).click(); | | 101 |  | | 102 | wordOutput.html().should.equal('a'); | | 103 | }); | |

* Creamos el código que pase el test.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 26 | createButtons: *function*() { | | 27 | for(*var* i = 97; i <= 122; i++) { | | 28 | *var* button = $('<button>' + *String*.fromCharCode(i) + '</button>'); | | 29 | button.click($.proxy(this.onButtonClick,this)); | | 30 | this.buttonList.append(button); | | 31 | } | | 32 | }, | | 33 | onButtonClick:*function*(*e*) { | | 34 | *var* letter=$(*e*.target).html(); | | 35 | this.tryLetter(letter); | | 36 | }, | |

* Probamos que el feature test pase.
* Ejecutar grunt nuevamente
  + Corregir el método tryletter(agregar 3 iguales): *if(letter === this.secretWord[index]) {*