

Testing Clientside JavaScript

Jasmine

Joe Eames

@josepheames

www.testdrivenjs.com

www.pluralsight.com



pluralsight 
hardcore developer training

Overview

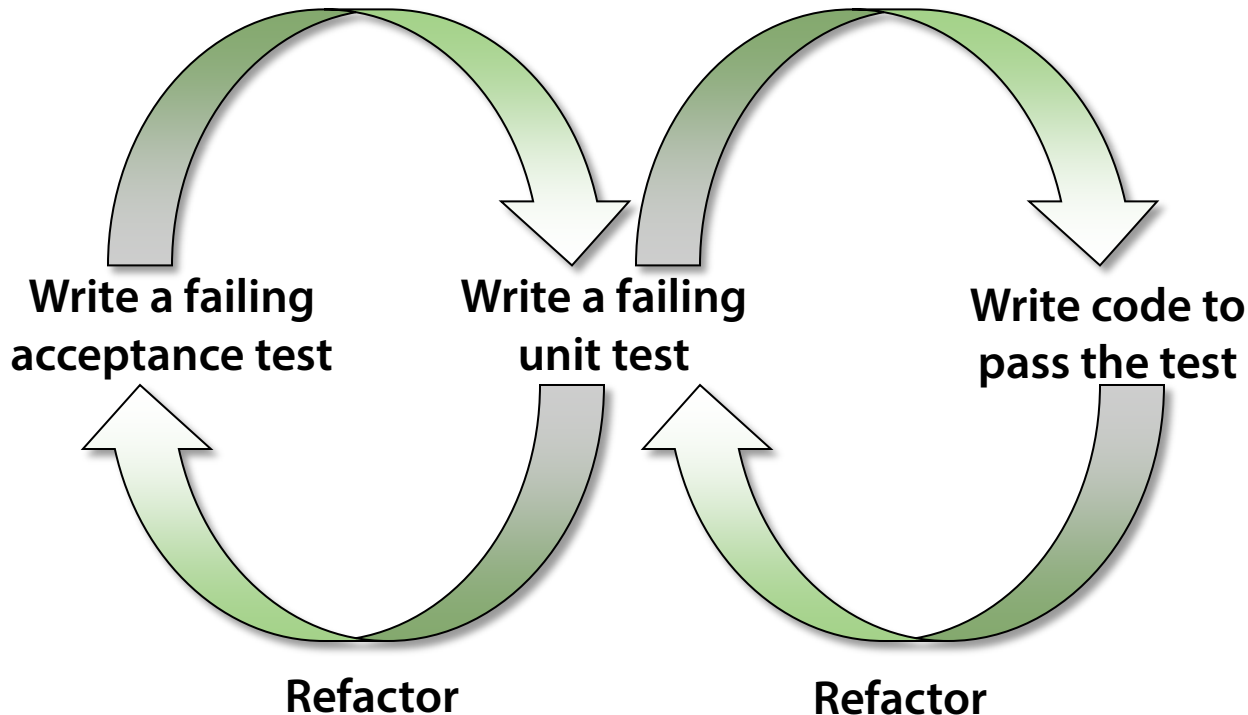
- Introduction to BDD & Jasmine
- Setting up Jasmine
- Organizing Tests
- Running Tests
- Spies
- Integrating with the DOM
- Integrating with CI
- Asynchronous Tests

Introduction to Jasmine

- **Open Source Framework**
- **Behavior Driven Development framework**
- **Supports both client-side and server-side testing**

Behavior Driven Development

- Created in 2006 by Dan North
- Is a superset of TDD
- Focuses on Language



Setting Up Jasmine

- Get Source Code
 - <https://github.com/pivotal/jasmine>
- Create Spec Runner File
- Create Test Files

Organizing Tests – Folders & Files

Project Root

Scripts

Module1

Source1.js

Module2

Source2.js

Project Root

Scripts

Module1

Source1.js

Module2

Source2.js

Tests

Libs

Module1

Source1Spec.js

Module2

Source2Spec.js

Organizing Jasmine

- **Describe Blocks**
 - Can Be Nested

```
describe('User', function() {  
    /* tests */  
});
```

```
describe('User', function() {  
    describe('when new', function() {  
        /* tests */  
    });  
});
```

Grouping With Testrunner files

TestFile1.Html
 TestFile1.js
 SourceFile1.js
TestFile2.Html
 TestFile2.js
 SourceFile2.js

TestRunner1.Html
 TestFile1.js
 SourceFile2.js
 SourceFile1.js
TestRunner2.Html
 TestFile2.js
 TestFile3.js
 SourceFile3.js
TestRunner3.Html
 TestFile4.js
 TestFile5.js
 SourceFile4.js
 SourceFile5.js

Writing Jasmine Tests

- `it()`
- `beforeEach` & `afterEach`
- `Matchers`
- `Custom Matchers`

Writing Jasmine Tests

```
describe('Player', function() {  
  it('should be able to play a song', function() {  
    /* code and assertions */  
  });  
});
```

beforeEach() and afterEach()

```
describe('User', function() {  
  var sut;  
  beforeEach(function() {  
    sut = new SUT();  
  });  
  afterEach(function() {  
    /* cleanup code */  
  });  
  
  it('should be able to play a song', function() {  
    /* code and assertions */  
  });  
});
```

Built-in Matchers

- `expect(x).toEqual(y);`
- `expect(x).toBe(y);`
- `expect(x).toMatch(pattern);`
- `expect(x).toBeDefined();`
- `Expect(x).toBeUndefined();`
- `expect(x).toBeNull();`
- `expect(x).toBeTruthy();`
- `expect(x).toBeFalsy();`
- `expect(x).toContain(y);`
- `expect(x).toBeLessThan(y);`
- `expect(x).toBeGreaterThan(y);`
- `expect(function(){fn();}).toThrow(ex);`

Built-in Matchers

- `expect(x).not.toEqual(y);`
- `expect(x).not.toBe(y);`
- `expect(x).not.toMatch(pattern);`
- `expect(x).not.toBeDefined();`
- `Expect(x).not.toBeUndefined();`
- `expect(x).not.beNull();`
- `expect(x).not.toBeTruthy();`
- `expect(x).not.toBeFalsy();`
- `expect(x).not.toContain(y);`
- `expect(x).not.toBeLessThan(y);`
- `expect(x).not.toBeGreaterThan(y);`
- `expect(function(){fn();}).not.toThrow(ex);`

Creating Custom Matchers

- Typically created in a beforeEach
- `this.addMatchers()`

```
beforeEach(function() {  
  this.addMatchers({  
    toBeFive: function() {  
      return this.actual === 5;  
    }  
  });  
});
```

Creating Custom Matchers

- The error message can be customized.

```
beforeEach(function() {  
  this.addMatchers({  
    toBeFive: function() {  
      this.message = function() {  
        return this.actual + " is not 5, dummy";  
      }  
      return this.actual < 5;  
    }  
  });  
});
```

Writing Tests - Demo

Running Jasmine Tests

- Run in a browser
- Hide & Display passed tests
- Hide & Display skipped tests
- Filter tests

Skipping Tests

- Add an “x” in front of the describe or the it function

```
xit('should be skipped', function() {  
    /* this test is skipped */  
});
```

```
xdescribe('MyClass', function() {  
    /* all these tests are skipped */  
});
```

Integrating Jasmine with the DOM

- Test the DOM only when appropriate
- Requires additional setup and teardown

Integrating Jasmine with the DOM

- Test the DOM only when appropriate
- Requires additional setup and teardown

Integrating With CI

- **Basics: Use PhantomJS and capture output**
 - <http://code.google.com/p/phantomjs/downloads/list>
- **jasmine.teamcity_reporter.js**
 - <https://github.com/larrymyers/jasmine-reporters>
- **run-jasmine.js**

Asynchronous Tests

- **Useful for testing:**
 - setTimeout and setInterval
 - UI effects
 - Ajax calls
- **runs() & waitsFor() method**
- **Clock.useMock()**
- **Use jasmine.async**
 - <https://github.com/derickbailey/jasmine.async>

Summary

- BDD style
- Versatile test runner
- Custom Matchers
- Spies
- Integrating with the DOM
- Asynchronous Tests