

**Stop Making
Excuses and
Start Testing
Your
JavaScript!**



About Me

- Senior UI Engineer at Netflix
- Lover of Open Source
- ryan.anklam@gmail.com
- bittersweeryan

Today's Agenda

- Adding Testing to Your Project
- Writing Testable JavaScript
- Solving Common Testing Issues
- Automating Testing

Adding Testing To Your Project

Step 1: Pick Your Environment



Jasmine



Browser



Jasmine



QUnit
js unit testing

Step 2: Pick Your Dialect

expect

```
describe( 'adder', function() {  
  it( 'should return zero when no args...',  
    function() {  
      expect( adder() ).to.equal( 0 );  
    } );  
} );
```



should

```
describe( 'adder', function() {  
  it( 'should return zero when no args...',  
    function() {  
      adder().should.equal( 0 );  
    } );  
} );
```



assert

```
//qunit
```

```
test( 'adder', function() {  
  ok( adder() === 0, 'should return 0' );  
} );
```

```
//chai
```

```
describe( 'adder', function() {  
  assert.equal( adder() , 0, 'should return 0' );  
} ) ;
```



Step 3: Setup



```
$ npm install -g mocha &&  
npm install chai --save-dev
```



adderSpec.js

```
var expect = require( 'chai' ).expect;
var adder = require( '../src/adder' );

describe( 'adder', function() {
  it( 'should add an array of numbers', function() {
    {
      expect( adder( 1, 2, 3 ) ).to.equal( 6 );
    }
  });
})
```




adderSpec.js

```
...  
describe( 'adder', function() {  
    it( 'should return zero if no args are passed  
in', function() {  
        expect( adder(0) ).toEqual(0);  
    });  
})  
...
```



adderSpec.js

```
...  
    it( 'should skip non numbers', function() {  
        expect( adder( 1,2,'a' ) ).to.equal( 3 );  
    });  
...
```



```
$ mocha tests/spec
```

↑
Name of your spec directory



...

2 passing (6ms)

1 failing

1) adder should skip non numbers:

AssertionError: expected 'a3' to equal 3

at Context.<anonymous> (/Users/ranklam/source/talks/html5devconf/2014/samples/tests/spec/adderSpec.js:14:39)

at callFn (/usr/local/lib/node_modules/mocha/lib/runnable.js:223:21)

at Test.Runnable.run (/usr/local/lib/node_modules/mocha/lib/runnable.js:216:7)

at Runner.runTest (/usr/local/lib/node_modules/mocha/lib/runner.js:374:10)

at /usr/local/lib/node_modules/mocha/lib/runner.js:452:12

at next (/usr/local/lib/node_modules/mocha/lib/runner.js:299:14)

at /usr/local/lib/node_modules/mocha/lib/runner.js:309:7

at next (/usr/local/lib/node_modules/mocha/lib/runner.js:247:23)

at Immediate._onImmediate (/usr/local/lib/node_modules/mocha/lib/runner.js:276:5)

at processImmediate [as _immediateCallback] (timers.js:374:17)

Browser

SpecRunner.html

```
<html>
<head>
  <title>Jasmine Spec Runner v2.0.0</title>
  <link rel="stylesheet" type="text/css" href="lib/jasmine-2.0.0/
jasmine.css">

  <script type="text/javascript" src="lib/jasmine-2.0.0/jasmine.js"></
script>
  <script type="text/javascript" src="lib/jasmine-2.0.0/jasmine-
html.js"></script>
  <script type="text/javascript" src="lib/jasmine-2.0.0/boot.js"></
script>

  <!-- include source files here... -->
  <script type="text/javascript" src="../src/adder.js"></script>
  <!-- include spec files here... -->
  <script type="text/javascript" src="spec/adderSpec-jasmine.js"></
script>
</head>
<body>
</body>
</html>
```

The diagram consists of a central box with two sections: 'Source Files' and 'Spec Files'. Two red arrows originate from this box. One arrow points to the comment line '<!-- include source files here... -->' in the code. The other arrow points to the script tag '<script type="text/javascript" src="../src/adder.js"></script>' which immediately follows the source files comment. This illustrates how the 'Source Files' section of the diagram corresponds to the source code files included in the runner.

Browser

Jasmine 2.0.0

finished in 0.002s

● ● X

3 specs, 1 failure

raise exceptions ☐

Spec List | Failures

adder should skip non numbers

Expected 'a3' to equal 3.

```
stack@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmine.js:1293:11
buildExpectationResult@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmine.js:1270:1
Env/expectationResultFactory@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmine.js:4
Spec.prototype.addExpectationResult@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmi
addExpectationResult@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmine.js:442:9
Expectation.prototype.wrapCompare/<@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmi
@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/spec/adderSpec-jasmine.js:11:9
attemptSync@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmine.js:1510:9
QueueRunner.prototype.run@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmine.js:1498
QueueRunner.prototype.execute@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmine.js:
Env/queueRunnerFactory@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmine.js:518:7
Spec.prototype.execute@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmine.js:309:1
wrapChildAtScope/@file:///Users/ranklam/source/talks/html5devconf/2014/samples/tests/lib/jasmine-2.0.0/jasmine.js:1708:21
```


Solving Common Testing Issues

I Have Methods that Call Other
Methods.

Mocks & Stubs

Use **mocks** when testing a function that needs data from another function

Use **stubs** when testing a function that invokes another function

Spies



Stubbing

```
User.prototype.addUser = function() {  
  if ( validateUser() ) {  
    saveUser();  
  }  
  else {  
    addError( 'user not valid' );  
  }  
};
```

Stubbing

```
it('should save a valid user', function() {  
    var user = new User('Ryan', 'Anklam');  
    spyOn(User, 'validate');  
    spyOn(User, 'save');  
  
    user.add();  
  
    expect(user.validate).toHaveBeenCalled();  
    expect(user.save).toHaveBeenCalled();  
});
```


Mocking

```
User.prototype.listUsers = function() {  
    var users = this.getUsers();  
  
    users.forEach( function( user ) {  
        this.addUserToList( user );  
    } );  
};
```

Mocking

```
it('should get a list of users and add them',
function() {
    spyOn(User, 'getUsers').andReturn(
        [
            {
                firstName : 'Ryan',
                lastName  : 'Anklam'
            }
        ]
    );
    spyOn(User, 'addUserToList');
    user.listUsers();

    expect(user.getUsers).toHaveBeenCalled();
    expect(user.addUserToList).toHaveBeenCalled();
});
```

***STOP: 0x000000D1 (0x00000000, 0xF73120AE, 0xC0000008, 0xC0000000)

A problem has been detected and Windows has been shut down to prevent damage to your computer

DRIVER_IRQL_NOT_LESS_OR_EQUAL

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

**** ABCD.SYS - Address F73120AE base at C0000000, DateStamp 36B072A3

Kernel1 Debugger Using: COM2 (Port 0x2F8, Baud Rate 19200)

Beginning dump of physical memory

Physical memory dump complete. Contact your system administrator or technical support group.

I Promise It's Not That Hard

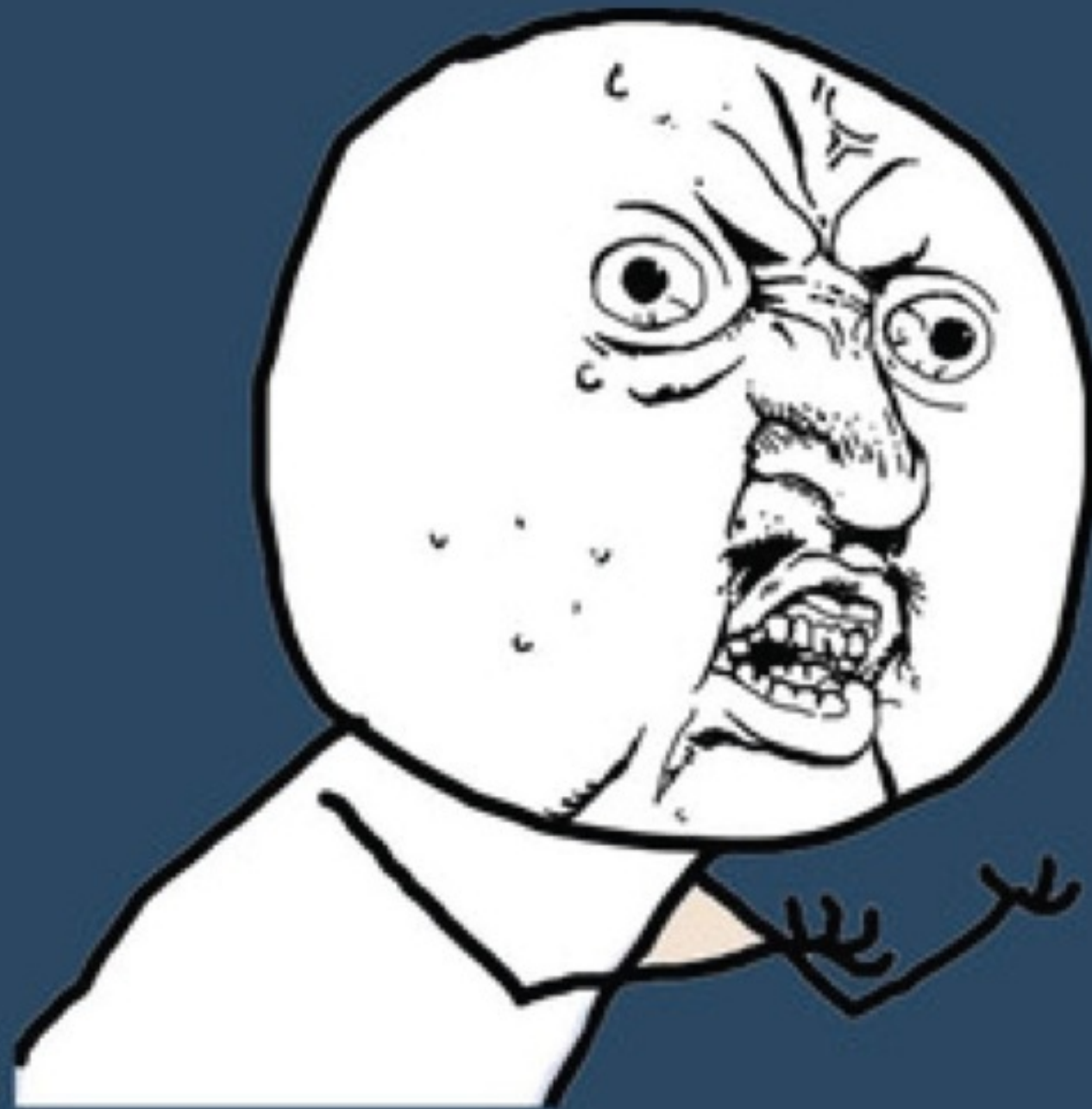
chai-as-promised FTW!

```
describe( 'getStringValue function', function() {  
  
  it('should return a string when it finds a key',  
    function ( ) {  
      return expect( utils.getStringValue( 'foo.bar.baz' ) )  
        .to.eventually.equal( 'Foo bar baz' ] );  
    });  
  
  it('should return the key when no key is found',  
    function ( ) {  
      return expect( utils.getStringValue( 'does.not.exist' ) )  
        .to.eventually.equal( 'does.not.exist' );  
    });  
} );
```

I Need To Test the DOM

Do you REALLY need to?

Y U NO



SPY ON THE DOM

Spy On The DOM

```
function addItem( $el, items ){  
    items.forEach( function( item ){  
        $el.append( item );  
    } );  
}
```

Spy On The DOM

```
it( 'should add items', function() {  
    var $el = $( '<ul/>' ),  
        items = [  
            { name : 'test' }  
        ];  
    spyOn( jQuery.fn, 'append' );  
  
    expect( jQuery.fn.append )  
        .toHaveBeenCalled();  
} );
```

Jasmine jQuery

<https://github.com/velesin/jasmine-jquery>

Create Fixtures as .html files

Load fixtures into a test

Make assertions

Jasmine jQuery

```
beforeEach( function() {  
    //reset the fixture before each test  
    loadFixtures ( 'myfixture.html' );  
});  
  
it( 'should convert a select to a ul', function() {  
    $( '.select' ).dropdown();  
  
    expect( '.select-list' ).toExist();  
});
```

qUnit

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>qUnit Example</title>
    <link rel="stylesheet" href="qunit.css">
  </head>
  <body>
    <div id="qunit"></div>
    <div id="qunit-fixture"></div>
    <script src="qunit.js"></script>
    <script src="tests.js"></script>
  </body>
</html>
```

Gets reset after every test

I Have to Test Async Code

Async Test

```
function readConfig( callback ){  
  fs.readFile( config.fileLocation );  
  try{  
    ...  
    callback( null, 'config ready' );  
  }  
  catch(e) {  
    callback( e );  
  }  
}
```

Async Test

```
it( 'should read the config file' ,  
    function( done ) {  
        var callback = function( readystate ) {  
            expect( readystate )  
                .to.equal( 'config ready' );  
            done();  
        }  
        readConfig( callback );  
    } );
```



Async Test

```
asyncTest( 'should read the config file' ,  
  function( done ) {  
    var callback = function( readystate ) {  
      ok( readystate === 'config ready' );  
      start();  
    }  
    readConfig( callback );  
  } );
```



Writing Testable JavaScript

Remember This?

```
$(function() {
    var $list = $( '.todo-list' ),
        $input = $( '.new-todo' ),
        todos = $.ajax(...);

    todos.forEach( function( item, index) {
        $list.append( '<li class="todo-item">' + item.name +
'</li>' );
    } );

    $input.on( 'keyup', function( e ) {
        if( e.which === 13 ) {
            $list.append( '<li class="todo-item">' +
                $(this).val() + '</li>' );
            $(this).val( '' );
        }
    } );
});
```

Not Testable JavaScript

Locked in callback



```
$(function() {  
    var $list = $( '.todo-list' ),  
        $input = $( '.new-todo' ),  
        todos = $.ajax(...);  
  
    todos.forEach( function( item, index) {  
        $list.append( '<li class="todo-item">' + item.name +  
'</li>' );  
    } );  
  
    $input.on( 'keyup', function( e ) {  
        if( e.which === 13 ) {  
            $list.append( '<li class="todo-item">' +  
                $(this).val() + '</li>' );  
            $(this).val( '' );  
        }  
    } );  
});
```


Make It A Constructor

```
var Todos = function( $list, $input ) {  
    this.$list = $list;  
    this.$input = $input;  
};
```

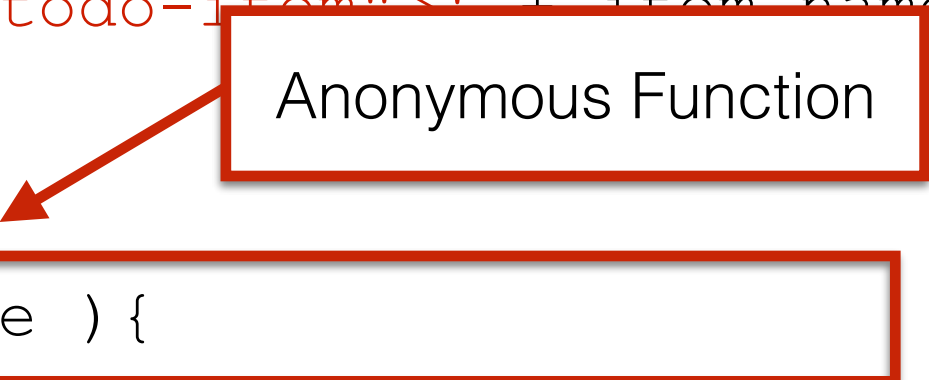
A Little Setup & First Test

```
describe( 'todos', function() {  
  var todos;  
  
  beforeEach( function() {  
    todos = new Todos( $( '<ul/>' ),  
      $( '<input type="text" value="foobar" />' )  
    );  
  });  
} );
```

```
it( 'should create an instance', function() {  
  expect( typeof todos ).toEqual( 'object' );  
} );
```

Not Testable JavaScript

```
$(function() {  
    var $list = $( '.todo-list' ),  
        $input = $( '.new-todo' ),  
        todos = $.ajax(...);  
  
    todos.forEach( function( item, index) {  
        $list.append( '<li class="todo-item">' + item.name +  
'</li>' );  
    } );  
  
    $input.on( 'keyup', function( e ) {  
        if( e.which === 13 ) {  
            $list.append( '<li class="todo-item">' +  
                $(this).val() + '</li>' );  
            $(this).val( '' );  
        }  
    } );  
});
```



Anonymous Function

Not Testable JavaScript

```
$(function() {  
    var $list = $( '.todo-list' );  
    $input = $( '.new-todo' );  
    todos = $.ajax(...);  
  
    todos.forEach( function( item, index) {  
        $list.append( '<li class="todo-item">' + item.name +  
'</li>' );  
    } );  
  
    $input.on( 'keyup', function( e ) {  
        if( e.which === 13 ) {  
            $list.append( '<li class="todo-item">' +  
                $(this).val() + '</li>' );  
            $(this).val( '' );  
        }  
    } );  
});
```

Anonymous Function

Not Testable JavaScript

```
$(function() {  
    var $list = $( '.todo-list' );  
    $input = $( '.new-todo' );  
    todos = $.ajax( ... );  
  
    todos.forEach( function( item, index ) {  
        $list.append( '<li class="todo-item">' + item.name +  
'</li>' );  
    } );  
  
    $input.on( 'keyup', function( e ) {  
        if( e.which === 13 ) {  
            $list.append( '<li class="todo-item">' +  
                $(this).val() + '</li>' );  
            $(this).val( '' );  
        }  
    } );  
});
```

Magic Data

Not Testable JavaScript

Can't configure

```
$(function() {  
    var $list = $( '.todo-list' ),  
        $input = $( '.new-todo' ),  
        todos = $.ajax(...);  
  
    todos.forEach( function( item, index) {  
        $list.append( '<li class="todo-item">' + item.name +  
'</li>' );  
    } );  
  
    $input.on( 'keyup', function( e ) {  
        if( e.which === 13 ) {  
            $list.append( '<li class="todo-item">' +  
                $(this).val() + '</li>' );  
            $(this).val( '' );  
        }  
    } );  
});
```

Refactoring

```
$(function() {  
    var $list = $( '.todo-list' );  
    $input = $( '.new-todo' );  
    todos = $.ajax( ... );  
  
    todos.forEach( function( item, index ) {  
        $list.append( '<li class="todo-item">' + item.name +  
</li>' );  
    } );  
  
    $input.on( 'keyup', function( e ) {  
        if( e.which === 13 ) {  
            $list.append( '<li class="todo-item">' +  
                $(this).val() + '</li>' );  
            $(this).val( '' );  
        }  
    } );  
});
```

One and done

A Few More Tips

- Avoid Singletons
- Write MAINTAINABLE tests
- Use named functions instead of anonymous callbacks
- Keep your functions small

Automating Testing

Using Your Build Tool

Testem

Karma

Thank You

@bittersweetryan

ryan.anklam@gmail.com