

# **jQuery Plugins & Jasmine**

**A lightning quick example**

@mparramon

# What?

With

- some useful js code

Create

- a full-fledged jQuery

plugin

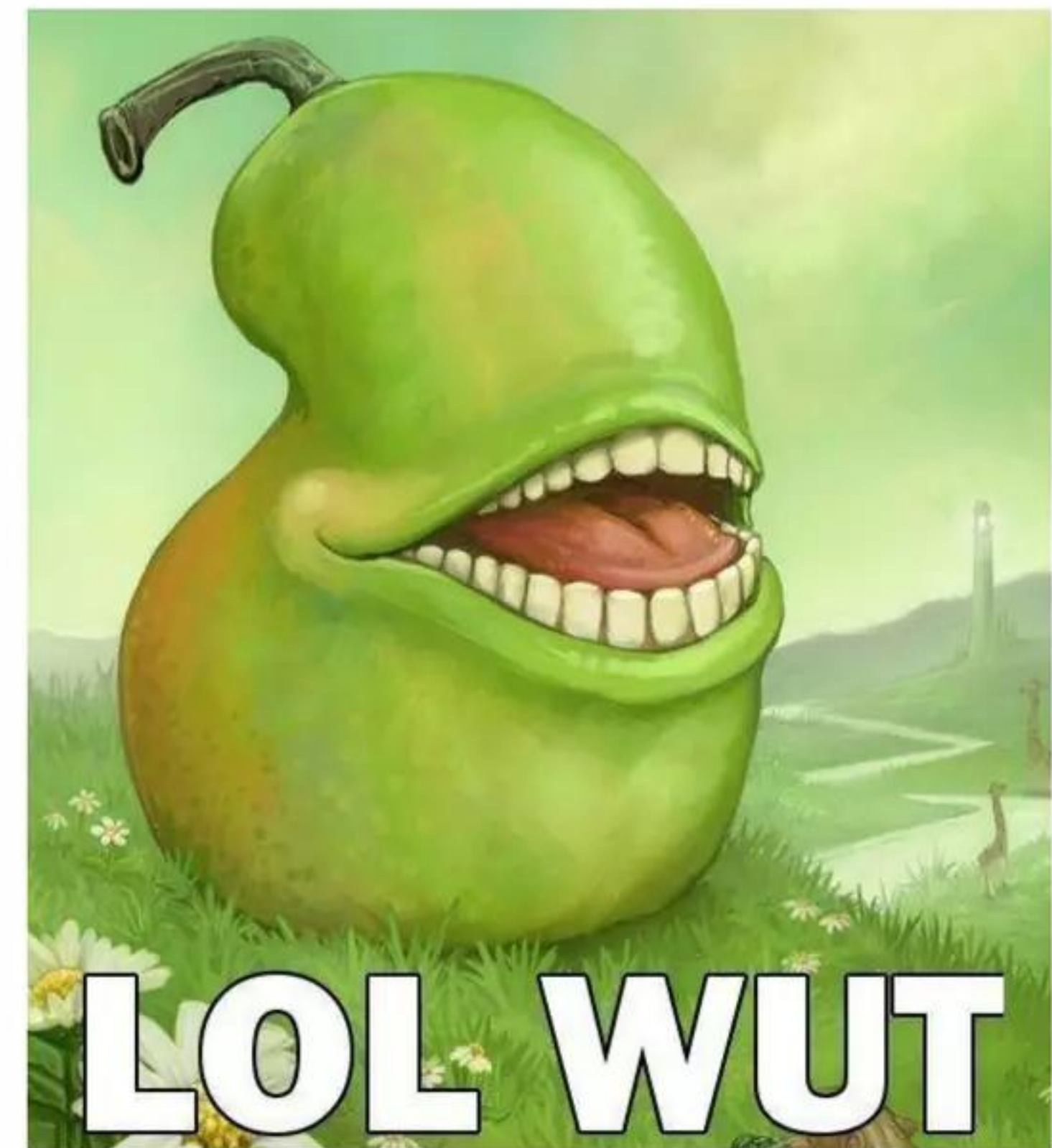
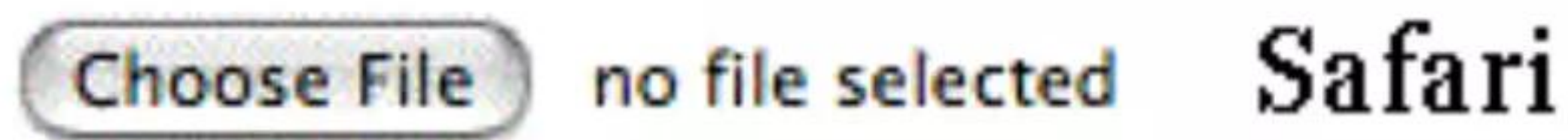
- with unit testing

In 5 minutes!



# The problem

File inputs in some browsers:



# let's fix it!

SELECT 

- add styling
- custom button
- initial value

Based on [quirksmode.org/dom/inputfile.html](http://quirksmode.org/dom/inputfile.html)

# jQuery plugin

```
$('.file_input').enhanceFileInput(options);
```

<http://docs.jquery.com/Plugins/Authoring>

# Basic Framework

```
(function( $ ) {  
    $.fn.myPlugin = function() {  
        // Do your awesome plugin stuff here  
    };  
})( jQuery );
```

# Defaults and options

```
var settings = $.extend( {  
  'initial_value' : '',  
  'condition'     : function(initial_value){return true;},  
  'html'          : "<img src='img/select.png'>"  
}, options);  
  
var initial_value = settings['initial_value'];
```

# Mantaining chainability

```
return this.each(function() {  
    //some awesome plugin stuff...  
  
});
```

# Unit testing with Jasmine

Jasmine 1.3.0 revision 1354052693

finished in 0.016s

• • • • •

Passing 7 specs

No try/catch ■

enhanced\_file\_input

filename

with a windows path, should leave only the file name

with a Unix path, should leave only the file name

enhanceFileInput

should add: [wrapper, input, onchange]

should add html after the enhanced input, if provided

with some initial value

should set it as the new input's value

with a condition for the initial value

should set it as the new input's value if true

should not set it as the new input's value if false

<http://pivotal.github.com/jasmine/>

# Initial setup

Modify SpecRunner.html:

```
<!-- include source files here... -->  
<script type="text/javascript" src="src/jquery-enhancedfileinput.js"></script>  
  
<!-- include spec files here... -->  
<script type="text/javascript" src="spec/jquery-enhancedfileinput_spec.js"></script>
```

# DSL

```
describe("A suite", function() {  
  it("contains spec with an expectation", function() {  
    expect(true).toBe(true);  
  });  
});
```

```
//All sorts of expectations:  
expect(false).not.toBe(true);
```

```
var a = 12;  
expect(a).toEqual(12);
```

```
var message = 'foo bar baz';  
expect(message).toMatch(/bar/);
```

# jasmine-jquery

2 extensions for Jasmine:

- an HTML fixtures API
- a set of jQuery custom matchers

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

# Fixtures

```
//In myfixture.html file:  
<div id="my-fixture">some complex content here</div>  
  
//Inside your test:  
loadFixtures('myfixture.html');  
$('#my-fixture').myTestedPlugin();  
expect($('#my-fixture')).to...;  
  
//In our code, as the basic element to enhance  
beforeEach(function() {  
  setFixtures('<input id="example_input"></div>');  
});
```

# Expectations

```
expect($('<div id="some-id"></div>')).toBe('div#some-id')
expect($('<input type="checkbox" checked="checked"/>')).toBeChecked()

toBeVisible(), toExist(), toHaveAttr(), toHaveProp()...

//In our code, checking for an onchange handler
expect($('#example_input').get(0).onchange).toBe(preserve_value);
```

# Our final code

jquery-enhancedfileinput

↳ lib

↳ jasmine-1.3.0

spec

↳ jquery-enhancedfileinput\_spec.js

↳ jasmine-jquery-1.3.1.js

src

↳ jquery-enhancedfileinput.js

↳ demo.html

↳ README.md

↳ SpecRunner.html

# Questions?

**No time!**



@mparramon

[github.com/mparramont](https://github.com/mparramont)