

jQuery and Drupal

Steven Wittens

JavaScript

You probably hate it.

It has nothing to do
with Java.

“It’s buggy and broken”

- Browsers do suck
- Implementations of CSS, events, ... broken
Language itself is pretty reliable
- Orders of magnitude better than in the 90’s
- Blame IE6

“Used only for annoying people”

- Usability is a problem with or without JS
- Browsers prevent worst offenses
- Use for good, not for evil

“Only use it because you have to”

- The most successful scripting environment?

Every browser, many OSes (Windows, Dashboard), XUL (Mozilla), Flash (ActionScript), server-side (ASP, Rhino), ...

- Great for UI-coding

Flexible & Powerful

- Imperative, functional and OO
- Lexical scope and closures
+ Anonymous functions
- Everything is an object (including functions)
- Prototype-based inheritance

Learn JavaScript

You know you want to.

Essentials

```
var myInt = 1;
```

```
var myString = "foobar";
```

```
var myArray = [ 1, 'two', "three" ];
```

```
assert(myArray[0] == 1);
```

```
var myHash = { foo: 1, 'bar': 'two' };
```

```
assert(myHash.foo == 1);
```

```
assert(myHash['foo'] == 1);
```

if(), switch(), for(), ... work like you expect them to.

Function = Primitive type

```
function myFunction() {  
    ...  
};
```

```
var myFunction = function () {  
    ...  
};
```

```
var sameFunction = myFunction;  
sameFunction();
```

```
(function () { ... })();
```

Hash = Object

```
var myObject = {  
  foo: 1,  
  bar: 'two',  
  spam: function () { this.foo++; }  
};
```

```
myObject.spam = function () {  
  this.foo += 2;  
};
```

```
myObject.spam();
```

Function = Object Constructor

```
var myClass = function (doodad) {  
  this.doodad = doodad;  
  this.spam = function () { ... };  
  return this;  
};
```

```
var myObject = new myClass("boing");  
myObject.spam();
```

Function = Object Constructor

```
var myClass = function (doodad) {  
    this.doodad = doodad;  
  
    return this;  
};
```

```
myClass.prototype.spam = function () {...};
```

```
var myObject = new myClass("boing");  
myObject.spam();
```

Function = Closure

```
function myFunction() {  
  var counter = 0;  
  
  function closureFunction() {  
    counter++;  
    alert(counter);  
  };  
  
  closureFunction(); // shows '1'  
  closureFunction(); // shows '2'  
};
```

Function = Closure

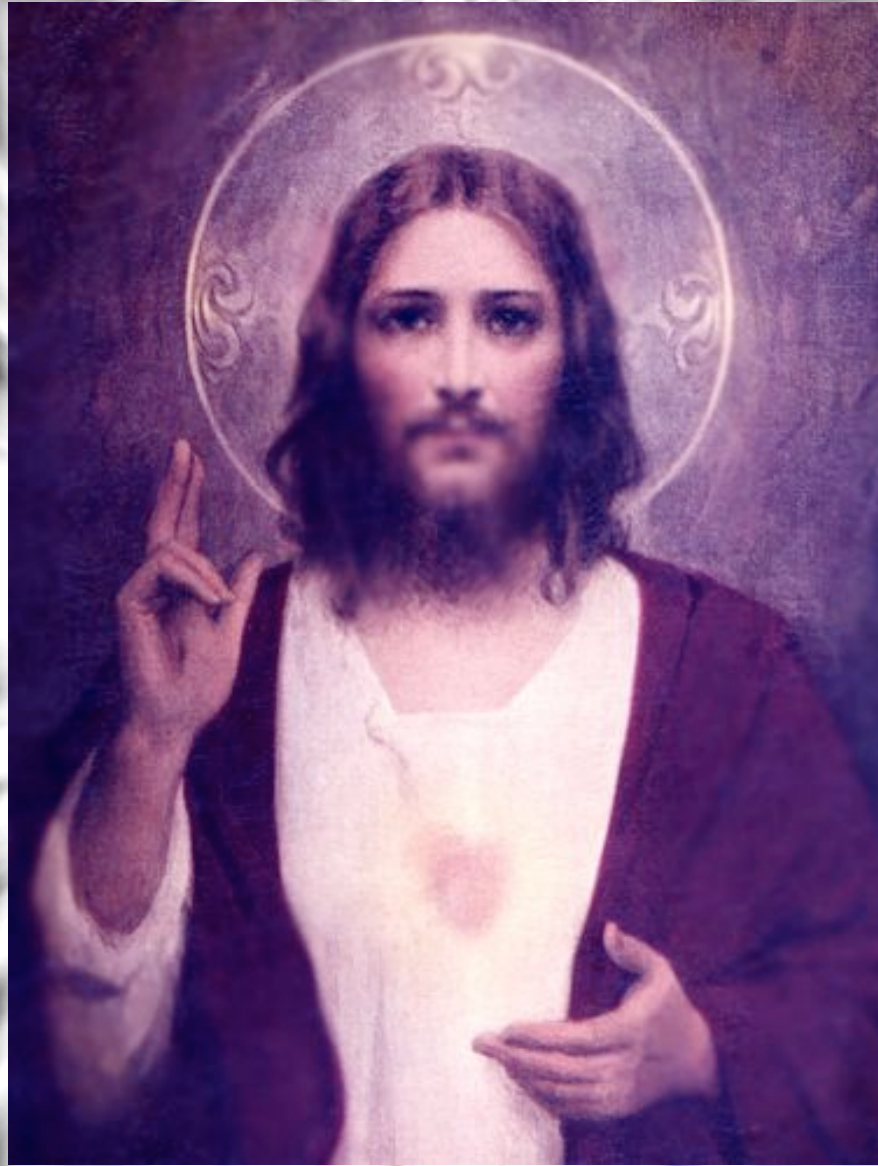
```
function myFunction() {  
    var counter = 0;  
  
    function closureFunction() {  
        counter++;  
        alert(counter);  
    };  
  
    return closureFunction;  
};
```

```
var closure = myFunction();  
closure(); // shows '1'  
closure(); // shows '2'
```

A dramatic landscape photograph featuring a bright, vertical beam of light shining through a cloudy sky onto a lake. The scene is set in a mountainous area with dark, silhouetted peaks and a forested foreground. The text "See the light" is overlaid in white on the right side of the image.

See the light





JavaScript is Cool.

Browser Scripting

Make IE your bitch.

Document Object Model (DOM)

- Representation of the HTML document
- DOM Node = Element or `<Tag></Tag>`
- Root 'document' object

Document Object Model (DOM)

- Navigate or query the document tree:

```
var node = document.getElementById('my-element');  
var child = node.nextSibling.firstChild;
```

- Alter element properties and styles:

```
node.setAttribute('href') = 'http://drupal.org/';  
node.style.backgroundColor = 'red';  
alert(node.getAttribute('title'));
```

DOM Events

- Respond to user interaction: change, click, keydown, mousemove, ...
- Assign an event handler for a particular DOM Node:

```
node.onclick = function () {  
    this.style.color = 'green';  
}
```

'AJAX'

- Make HTTP GET and POST requests from within JS
 - XML
 - HTML ('AHAH')
 - JSON
 - JavaScript
 - Normal POST data

Things fly around
on the page
= AJAX?



Wait, wasn't this talk
about jQuery?

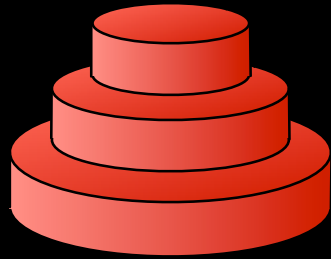
Problems?

- DOM tree is clunky to use
- No multiple handlers per event
- Browser incompatibilities
- No high-level functions

= JavaScript libraries to fill the gap

'New Wave' JavaScript

- jQuery – John Resig
- Released at BarCamp NYC (Jan 2006).
- jQuery 1.0 out (Aug 2006)
- Drupal 5.0



jQuery

- Doesn't mess with the language (~~Prototype~~)
- Doesn't try to be Python (~~Mochikit~~)
- Only essentials: 15KB (~~Scriptaculous~~, ~~Dojo~~)
- More than cosmetic effects (~~Moo.fx~~)
- Makes common tasks easy

Before – Plain DOM

```
var fieldsets = document.getElementsByTagName('fieldset');
var legend, fieldset;
for (var i = 0; fieldset = fieldsets[i]; i++) {
    if (!hasClass(fieldset, 'collapsible')) {
        continue;
    }
    legend = fieldset.getElementsByTagName('legend');
    if (legend.length == 0) {
        continue;
    }
    legend = legend[0];
    ...
}
```

After – With jQuery

```
$('.fieldset.collapsible legend').each(function () {...});
```

jQueryes

- Use CSS3/XPath selectors to find elements

```
$('#my-element')  
$('fieldset.collapsible > legend')  
$('table.prices tr:last-of-type td')
```

- Apply operations to all selected elements

```
$('p:hidden').addClass('emphasis');  
$('p:hidden').slideDown('fast');
```

It Gets Better

- jQuery Methods are chainable. Query object is stateful:

```
$( 'p' ).addClass( 'big' ).wrap( '<span></span>' )  
.filter( '.tagged' ).css( 'background', 'red' ).end()  
.filter( ':hidden' ).slideDown( 'fast' ).end();
```

- Sensible event handling

```
$( 'span.info' )  
.mouseover( function () { ... } )  
.mouseout( function () { ... } );
```

“Do What I Mean”

```
$( '<div id="throbber"></div>' )
  .hide()
  .ajaxStart(function(){
    $(this).show();
  })
  .ajaxStop(function(){
    $(this).hide();
  })
  .appendTo("#someContainer");
```

Demo

jQuery Effects

```
$(document).ready(function() {  
  $('#help_reveal').toggle(  
    function() {  
      $('#help').slideDown('fast');  
      return false;  
    },  
    function() {  
      $('#help').slideUp('fast');  
      return false;  
    },  
  );  
});
```

[Help](#)

Plug-ins

- Already more than 60
- New (chainable) methods
- Additional Effects (Pause, Ease)
- New Abilities (Interface, JSON, Forms)
- Widgets (Thickbox, Farbtastic, ImagePan)

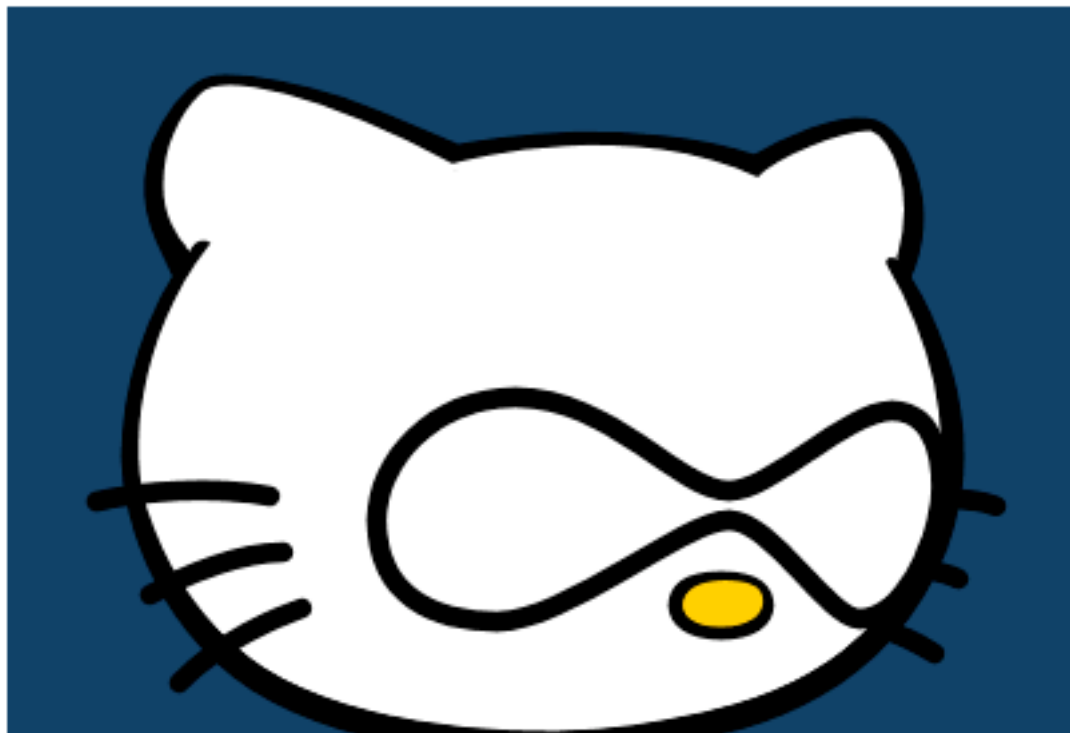
Demo

jQuery Farbtastic

```
$(document).ready(function() {  
  $.farbtastic('#picker').linkTo('#tshirt, #tshirt-logo, #color')  
});
```



Color: #123456



jQuery in Drupal

What changes in 5.0?

JS Usage in Drupal

- Lots of server-side logic
Localization, Formatting, Theming, ...
- Stick to small, self-contained JS features
- No complete client-side applications

Still: demand for good JS library

Server-Side

- Nothing really changes
- Menu callbacks to answer AJAX calls
- Widgets integrated into Form API
- Just a .js file – unobtrusive JavaScript

Client-Side

- Drupal methods are namespaced
- jQuery is included
- Core widgets rewritten to use jQuery
- Less code!

Attractive Platform

- Satisfies demand for ‘bells and whistles’
- Solid base for core
- Dedicated jQuery community
- Huge array of plug-ins
- JS wizardry not needed

Caveats

- Must be accessible
- Must degrade
- Especially for search engines
- Slow if used badly

Questions?

jQuery.com