

The DOM Scripting Toolkit: jQuery

Remy Sharp
Left Logic

Why JS Libraries?

- DOM scripting made easy
- Cross browser work done for you
- Puts some fun back in to coding

Why jQuery?

- Lean API, makes your code smaller
- Small (16k gzip'd), encapsulated, friendly library - plays well!
- Plugin API is *really* simple
- Large, active community
- Big boys use it too: Google, BBC, Digg, Wordpress, Amazon, IBM.

What's in jQuery?

- Selectors & Chaining
- DOM manipulation
- Events
- Ajax
- Simple effects

\$ function



- Selectors
- DOM elements or raw HTML
- “Ready” functions

\$

- `$('ul')`
- `$(document.createElement('ul'))`
- `$('#')`
- `$(fn)`

Selectors

```
$('#emails a.subject');
```


Selectors

- “Find something, do something with it”
- CSS 1-3 selectors at the core of jQuery
- Custom selectors

CSS Selectors

`$('div')`

`$('div.foo')`

`$('a[type="application/pdf"]')`

`$('tr td:first-child')`

Tricky Selector

```
<div id="content.block" />
```

```
$('#content.block') // won't work :-)
```

```
$('#content\\.block') // will work :-)
```

Custom Selectors

`$(‘div:visible’)`

`$(‘:animated’)`

`$(‘:input’)`

`$(‘tr:odd’)`

Custom selectors

- Roll your own

```
$.expr[':'], {  
  within: 'jQuery(a).parents(m[3]).length > 0'  
});
```

```
$('#a.subject').is(':within("#email")');
```

Selector Performance

```
$('#email') // same as getElementById  
$('.email') // like getElementsByTagName
```

```
// using context  
$('#div.email')  
$('#emails .subject')  
$('.subject', this)
```

```
// Caching  
var $subjects = $('#emails .subject');
```

Chaining

```
$('#emails .subjects')  
  .click()  
  .addClass('read');
```

Chaining

- jQuery returns itself *
- We can use the selector once, and keep manipulating

* except when requesting string values, such as `.css()` or `.val()`

Chaining Example

```
// simple tabs
$('a.tab').click(function () {
  // tabs = $('div.tab')
  $tabs
  .hide()
  .filter(this.hash) // refers to page.html#hash
  .show();
});
```

More Chaining

```
var image = new Image();  
$(image)  
  .load(function () {  
    // show new image  
  })  
  .error(function () {  
    // handle error  
  })  
  .attr('src', '/path/to/large-image.jpg');
```

Working the DOM

```
$(this).addClass('read').next().show();
```

DOM Walking

- Navigation: children, parent, parents, siblings, next, prev
- Filters: filter, find, not, eq
- Collections: add, end
- Tests: is

```
$('#div')  
  .find('a.subject')  
  .click(fn)  
  .end() // strip filter  
  .eq(0) // like :first  
  .addClass('highlight');
```

Manipulation

- Inserting: after, append, before, prepend, html, text, wrap, clone
- Clearing: empty, remove, removeAttr
- Style: attr, addClass, removeClass, toggleClass, css, hide, show, toggle

```
var $a = $(document.createElement('a')); // or $('<a>')
$a.css('opacity', 0.6).text('Click!').attr('href', '/home/');
$('div').empty().append(a);
```

Data

- `.data()` clean way of linking data to element
- Storing data directly against elements can cause leaks
- All handled by jQuery's garbage collection

```
$(el).data('type', 'forward');  
  
var types =  
$('a.subject').data('type');  
  
$('a.subject').removeData();
```

Events

```
$('#a.subject').click();
```

DOM Ready

- Runs after DOM, but before images

```
$(document).ready(function () {})
```

// or as a shortcut:

```
$(function () {})
```


Binding

- Manages events and garbage collection
- Event functions are bound to the elements matched selector
- Also supports `.one()`

```
$('a.reveal').bind('click', function(event) {  
  // 'this' refers to the current element  
  // this.hash is #moreInfo - mapping to real element  
  $(this.hash).slideDown();  
}).filter(':first').trigger('click');
```

Helpers

- Mouse: click, dblclick, mouseover, toggle, hover, etc.
- Keyboard: keydown, keyup, keypress
- Forms: change, select, submit, focus, blur
- Windows: scroll
- Windows, images, scripts: load, error

Custom Events

- Roll your own
- Bind to element (or elements)
- Trigger them later and pass data

```
$('div.myWidget').trigger('foo', [123/*id*/]);  
// id access via  
// .bind('foo', function (event, id, etc) {})
```

Event Namespaces

- Support for event subsets via namespaces
- If you don't want to unbind all type X events
 - use namespaces

```
$('a').bind('click.foo', fn);  
  
$('a').unbind('foo');
```

Ajax

```
$.ajax({ url : '/foo', success : bar });
```

Ajax Made Easy

- Cross browser, no hassle.
- \$.ajax does it all
- Helpers \$.get, \$.getJSON, \$.getScript, \$.post, \$.load
- All Ajax requests sends:
X-Requested-With = XMLHttpRequest

Simple Ajax

```
/* Loads the #links element with the content  
from link.html matched in the selector */  
$('#links').load('link.html #links li');
```

JSON for Data

- Uses eval for conversion
- Use JSONP for cross server data (flickr, twitter, delicious, etc)
- JSON simple to use, and less space than XML

Twitter Example

```
// Run when the DOM has completed.  
// i.e. don't hang on a script request  
$(function () {  
  var url = 'http://www.twitter.com/statuses/user\_timeline/rem.json?  
callback=?';  
  $.ajax({  
    dataType: 'jsonp',  
    url: url,  
    success: function (data) {  
      $('#status').html(data[0].text); // update with my latest tweet  
    }  
  });  
});
```

\$.ajax

```
$('form.register').submit(function () {  
  var el = this; // cache for use in success function  
  $.ajax({  
    url : $(this).attr('action'),  
    type : 'post',  
    data : { 'username' : $('input.username', this).val() },  
    beforeSend : showValidatingMsg, // function  
    dataType : 'json',  
    success : function (data) {  
      // do something with data - show validation message  
    },  
    error : function (xhr, status, e) {  
      // handle the error - inform the user of problem  
      console.log(xhr, status, e);  
    }  
  });  
  return false; // cancel default browser action  
});
```

Effects

```
$(this).slideDown();
```

Base Effects

```
$('#div:hidden').show(200, fn);
```

```
$('#div:visible').hide(200);
```

```
$('#div').fadeOut(200);
```

```
$('#div').slideDown(100);
```

```
$('#div').animate({
```

```
  'opacity' : 0.5,
```

```
  'left' : '-=10px'
```

```
}, 'slow', fn)
```

Utilities

`$.browser.version`

Utilities

- Iterators: each, map, grep
- Browser versions, model and boxModel support
- isFunction

Core Utilities

- jQuery can plays with others!
- Good guys come last

```
$ === jQuery; // true
```

```
$ === $j; // false
```

```
$j = $.noConflict(); // store jQuery in $j
```

```
$j === $; // false
```

```
$j === jQuery; // true
```

Plugins

```
$('#emails .subjects').doMagic()
```


Plugins

- Simple to write
- Makes your code *more* reusable
- Don't break the chain!

Simple Plugin

```
jQuery.fn.log = function () {  
  // returning continues the chain  
  return this.each(function() {  
    if (this.nodeName === 'A') {  
      console.log(this.href);  
    }  
  });  
};  
  
$('a').log().filter('[hash=#first]').log()
```

Core Utilities

- Extend jQuery, merge objects and create settings from defaults

```
var defaults = { 'limit' : 10, 'dataType' : 'json' };  
var options = { 'limit' : 5, 'username' : 'remy' };  
var settings = $.extend({}, defaults, options);  
  
// settings = { 'limit' : 5, 'dataType' : 'json',  
               'username' : 'remy' }
```

More Info

Remy Sharp:

remy@leftlogic.com

leftlogic.com

remysharp.com

jqueryfordesigners.com

Resources:

jquery.com

docs.jquery.com

[groups.google.com/group/
jquery-en](http://groups.google.com/group/jquery-en)

ui.jquery.com

learningjquery.com