





I'm Remy.

Who uses jQquery.

I'm Remy.

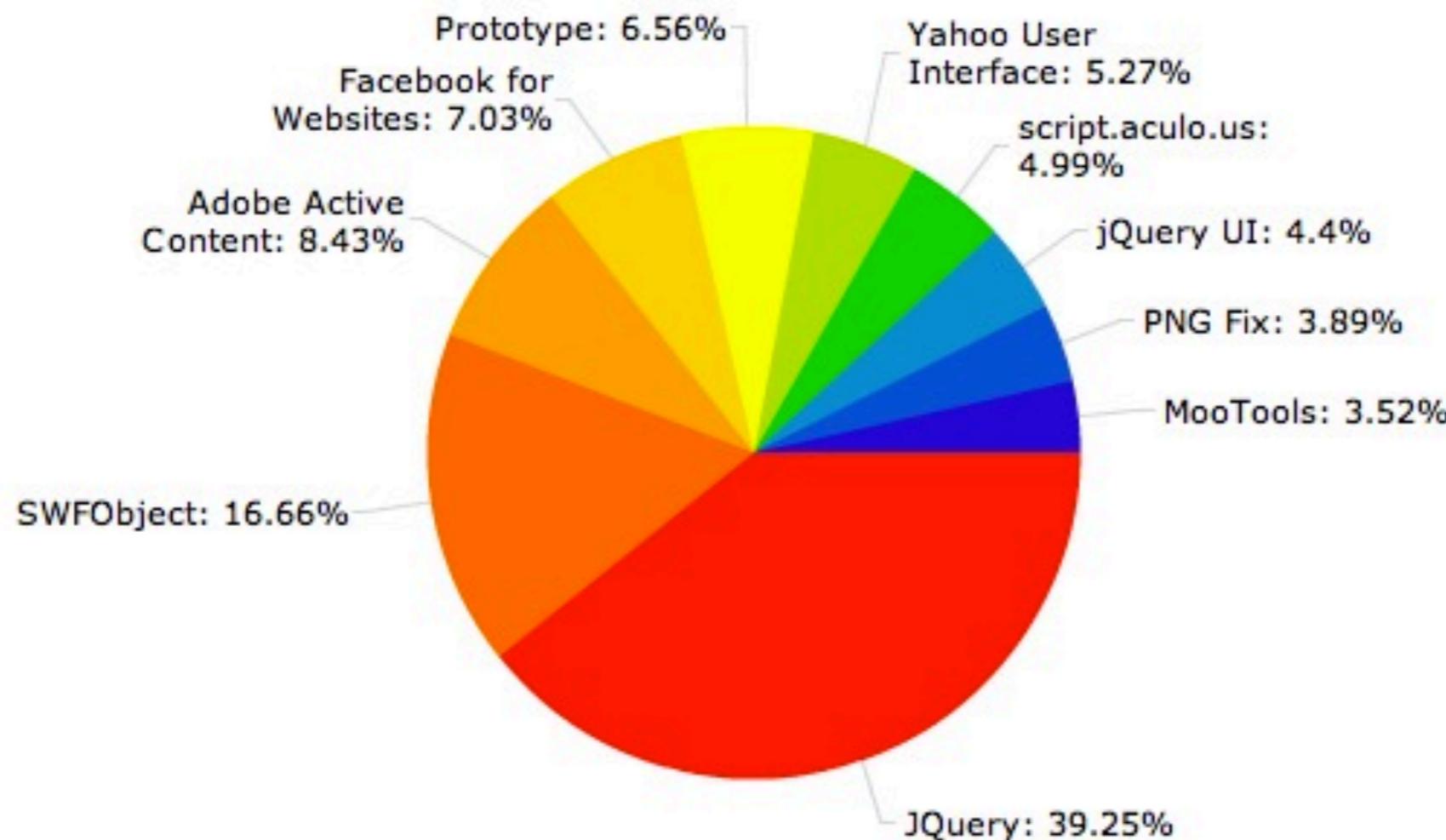
Who uses jQuery?

- Who & Why
- Get going
- What's new
- Oldies but goodies
- Dev patterns

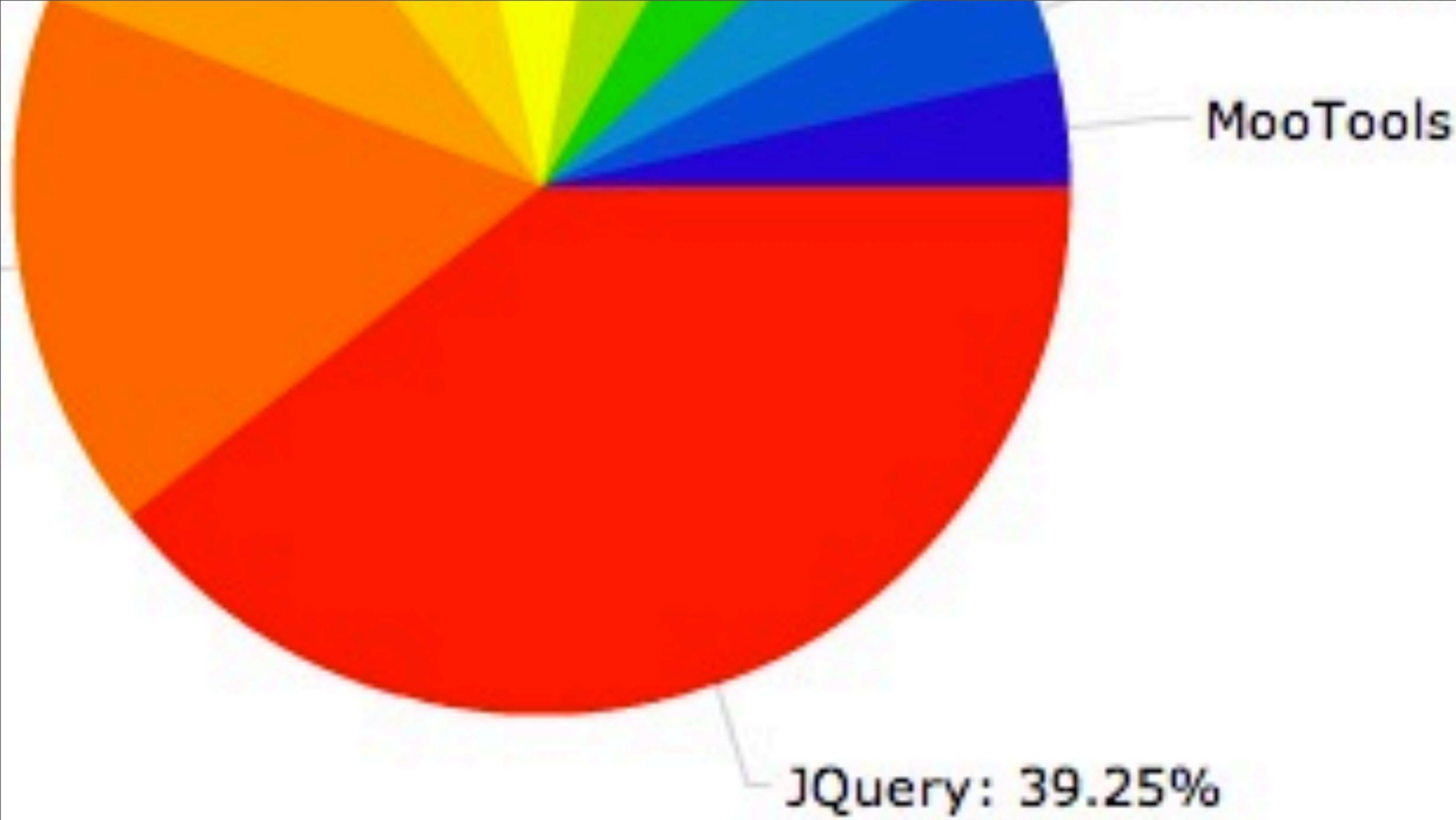
WARNING!

A LOT OF CODE AHEAD.

Who's using jQuery?



Distribution is calculated from a cross section of website domains provided by URLs entered at BuiltWith.com and the Quantcast Top Million. Last calculated on May 18 2010



Who's using jQuery?

reddit.com

espn.com

ibm.com

boxee.tv

bit.ly

tv.com

dell.com

twitter.com

twitpic.com

whitehouse.gov

microsoft.com

amazon.com

netflix.com

bing.com

monster.com

overstock.com

time.com

capitalone.com

stackoverflow.com

wordpress.com

w3.org

Why use jQuery

- **Simplifies and speeds up web development**
- **Avoid cross-browser headaches**
- **Large pool of plugins**
- **Large and active community**
- **Tested on 50 browsers, 11 platforms**
- **Accessible to both coder and designer**

Why use jQuery

- ~~Simplifies and speeds up web development~~
- ~~Avoid cross-browser headaches~~
- Large pool of plugins
- Large and active community
- Tested on 50 browsers, 11 platforms
- Accessible to both coder and designer

Installation

Hosting options

- Host it yourself (good for offline dev)
- Hotlink it:
 - Media Temple
 - Microsoft
 - Google (my pick)

http://docs.jquery.com/Downloading_jQuery

CDN FTW

```
<html>  
  <head>  
    <styles>  
      <!-- make me beautiful -->  
    </styles>  
  </head>  
  <body>  
    <content>  
      <!-- make me learned -->  
    </content>  
    <behaviour>  
      <!-- ooo, matron -->  
    </behaviour>  
  </body>  
</html>
```

**Styles first
let's the page
render
without
scripts
blocking**

```
<html>
  <head>
    <styles>
      <!-- make me beautiful -->
    </styles>
  </head>
  <body>
    <content>
      <!-- make me learned -->
    </content>
    <behaviour>
      <!-- ooo, matron -->
    </behaviour>
  </body>
</html>
```

**Finally, add
your
behaviour,
the jQuery
and sexy
magic jazz.**

```
<html>
  <head>
    <styles>
      <!-- make me beautiful -->
    </styles>
  </head>
  <body>
    <content>
      <!-- make me learned -->
    </content>
    <behaviour>
      <!-- ooo, matron -->
    </behaviour>
  </body>
</html>
```

The quick start guide

DOM Scripting

```
var tables = document.getElementsByTagName('table');

for (var t = 0; t < tables.length; t++) {

    var rows = tables[t].getElementsByTagName('tr');

    for (var i = 1; i < rows.length; i += 2) {

        if (!/(^|s)odd(s|$)/.test(rows[i].className)) {

            rows[i].className += ' odd';

        }

    }

}

}
```

jQuery simplifies

```
$('table tr:nth-child(odd)').addClass('odd');
```

jQuery simplifies

jQuery function

```
$('table tr:nth-child(odd)').addClass('odd');
```

jQuery simplifies

jQuery function

```
$('table tr:nth-child(odd)').addClass('odd');
```

CSS expression

jQuery simplifies

jQuery function

jQuery method

```
$('table tr:nth-child(odd)').addClass('odd');
```

CSS expression

jQuery simplifies

```
$('table tr:nth-child(odd)').addClass('odd');
```

API features

- 1. Events: click,
mouseover, load,
hover, keydown**
- 2. Ajax: JSON, post,
get, events, roll
your own**
- 3. Effects: slide,
fade, show/hide,
roll your own**
- 4. DOM navigation:
children, parent,
find, filter**

API features

- 1. Events: click,
mouseover, load,
hover, keydown**
- 2. Ajax: JSON, post,
get, events, roll
your own**
- 3. Effects: slide,
fade, show/hide,
roll your own**
- 4. DOM navigation:
children, parent,
find, filter**

API features

- 1. Events: click,
mouseover, load,
hover, keydown**
- 2. Ajax: JSON, post,
get, events, roll
your own**
- 3. Effects: slide,
fade, show/hide,
roll your own**
- 4. DOM navigation:
children, parent,
find, filter**

API features

- 1. Events: click,
mouseover, load,
hover, keydown**
- 2. Ajax: JSON, post,
get, events, roll
your own**
- 3. Effects: slide,
fade, show/hide,
roll your own**
- 4. DOM navigation:
children, parent,
find, filter**

API features

1. Events: click,
mouseover, load,
hover, keydown

3. Effects: slide,
fade, show/hide,
roll your own

and a whole lot more!

2. Ajax: JSON, post,
get, events, roll
your own

4. DOM navigation:
children, parent,
find, filter

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $this.fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

```
$('div').click(function () {  
    alert('you clicked me');  
    $(this).fadeOut();  
    return false;  
}).fadeIn();
```

1.4.2

The sexy bits.

Attributes in \$

```
$('<a id="printlink">Print this  
page</a>').click(function () {  
    window.print();  
    return false;  
});
```

```
$('<a/>')

.attr('id', 'printlink')

.text('Print this page')

.click(function () {

    window.print();

    return false;

});
```

```
$('<a/>', {
  id: 'printlink',
  text: 'Print this page',
  click: function () {
    window.print();
    return false;
  }
});
```

Bulk binding

```
$('a')  
.focus(updateDisplay)  
.keydown(updateDisplay)  
.blur(somethingElse);
```

```
$('a').bind( {  
    focus: updateDisplay,  
    keydown: updateDisplay,  
    blur: somethingElse  
} );
```

```
$('a').bind({  
  'keydown focus':  
    updateDisplay,  
  blur: somethingElse  
});
```

callback for setters

- **css**
- **attr**
- **val**
- **text**
- **html**
- **addClass**
- **toggleClass**
- **removeClass**
- **after**
- **before**
- **append**
- **prepend**
- **wrap**
- **wrapAll**
- **wrapInner**
- **replaceWith**

```
$('div').each(function () {
  $(this).html(
    $(this).html().replace(/foo/, 'bar')
  );
});
```

```
$('div').each(function () {  
  $(this).html(  
    $(this).html().replace(/foo/, 'bar')  
  );  
});
```

```
$('div').html(function (i, html) {  
  return html.replace(/foo/, 'bar');  
});
```

```
$( 'input' ).attr( 'value' , function ( i, attr ) {  
    return attr + i;  
});
```

```
<input type="text" value="remy" />  
<input type="text" value="bruce" />  
<input type="text" value="andy" />  
<input type="text" value="ribot" />
```

```
$( 'input' ).attr( 'value' , function ( i, attr ) {  
    return attr + i;  
});
```

```
<input type="text" value="remy0" />  
<input type="text" value="bruce1" />  
<input type="text" value="andy2" />  
<input type="text" value="ribot3" />
```

```
$('li').addClass(function () {
  return 'who-' + $(this).find('a').text();
});
```

```
<li><a href="#remy">Remy</a></li>
<li><a href="#bruce">Bruce</a></li>
<li><a href="#andy">Andy</a></li>
<li><a href="#ribot">Ribot</a></li>
```

```
$('li').addClass(function () {
  return 'who-' + $(this).find('a').text();
});
```

```
<li class="who-Remy"><a href="#remy">Remy</a></li>
<li class="who-Bruce"><a href="#bruce">Bruce</a></li>
<li class="who-Andy"><a href="#andy">Andy</a></li>
<li class="who-Ribot"><a href="#ribot">Ribot</a></li>
```

```
$('*').css('backgroundColor', function (i, css) {
  css = css.replace(/\^\\d,/g, ' ').split(',');
  return 'rgb(' + $.map(css, function (i) {
    return 255 - i;
  }).join(',') + ')';
});
```

.attr() – jQuery API

<http://api.jquery.com/attr/>

RSS Google

jQuery Plugins UI Meetups Forum Blog About Donate

 write less, do more.

Download Documentation Tutorials Bug Tracker Discussion

JQUERY API *Search jQuery*

jQuery API

New or Changed in 1.4
Raw XML API Dump
Dynamic API Browser
jQuery API Book

Browse the jQuery API

All
+ Ajax
Attributes
Core
CSS
Data

.attr()

Contents: Categories: [Attributes](#) | [Manipulation](#) > [General Attributes](#)

- ▼ attr(attributeName)
.attr(attributeName)
- ▼ attr(attributeName, value)
.attr(attributeName, value)
.attr(map)
.attr(attributeName, function(index, attr))

.attr(attributeName) *Returns: String*

Description: Get the value of an attribute for the first element in the set of matched

.attr() – jQuery API

<http://api.jquery.com/attr/>

RSS Google

jQuery Plugins UI Meetups Forum Blog About Donate

 **jQuery**
write less, do more.

Download Documentation Tutorials Bug Tracker Discussion

JQUERY API Search *jQuery*

jQuery API

New or Changed in 1.4

Raw XML API Dump

Dynamic API Browser

jQuery API Book

Browse the jQuery API

All

+ Ajax

Attributes

Core

CSS

Data

.attr()

Contents:

Categories: [Attributes](#) | [Manipulation](#) > [General Attributes](#)

↳ attr(attributeName)
.attr(attributeName)

↳ attr(attributeName, value)
.attr(attributeName, value)
.attr(map)
.attr(attributeName, function(index, attr))

.attr(attributeName) *Returns: String*

Description: Get the value of an attribute for the first element in the set of matched

```
$('*').css('backgroundColor', function (i, css) {
  css = css.replace(/\^\\d,/g, '').split(',');
  return 'rgb(' + $.map(css, function (i) {
    return 255 - i;
  }).join(',') + ')';
});
```

jQuery.parseJSON

Segue: .live

Problem

When the page has finished loading, the jQuery runs

```
<h1>Super Ted</h1>



<script src="jquery.js">
<script>
$( 'img' ).click(function() {
    showDetails(this);
});
</script>
```

**Clicking
these images
"shows
details"**

```
<h1>Super Ted</h1>



<script src="jquery.js">
<script>
$( 'img' ).click(function() {
    showDetails(this);
});
</script>
```

**Now Ajax
(or something
else) add new
images to the
page**

```
<h1>Super Ted</h1>



<script src="jquery.js">
<script>
$( 'img' ).click( function() {
    showDetails(this);
});
</script>
```

**Now Ajax
(or something
else) add new
images to the
page**

```
<h1>Super Ted</h1>




<script src="jquery.js">
<script>
$( 'img' ).click(function() {
  showDetails(this);
});
</script>
```

**Clicking this
image doesn't
do anything.**

```
<h1>Super Ted</h1>




<script src="jquery.js">
<script>
$( 'img' ).click(function() {
    showDetails(this);
});
</script>
```

Solution:
event delegation

```
$('p').live('click', fn);
```

```
<p>Ninja</p>
<p>Dizzy</p>
<script src="jquery.js"></script>
<script>
$('p').live('click', function () {
  $(this).after('<p>' + $(this).text()
    + ' clicked</p>');
});
</script>
```

```
$('p').live('click', fn);
```

```
<p>Ninja</p>
<p>Dizzy</p>
<script src="jquery.js"></script>
<script>
$('p').live('click', function () {
  $(this).after('<p>' + $(this).text()
    + ' clicked</p>');
});
</script>
```

```
$('p').live('click', fn);
```

```
<p>Ninja</p>
<p>Dizzy</p>
<script src="jquery.js"></script>
<script>
  $('p').live('click', function () {
    $(this).after('<p>' + $(this).text()
      + ' clicked</p>');
  });
</script>
```

```
$('p').live('click', fn);
```

```
<p>Ninja</p>
<p>Dizzy</p>
<script src="jquery.js"></script>
<script>
$('p').live('click', function () {
  $(this).after('<p>' + $(this).text()
                + ' clicked</p>');
});
</script>
```

```
$('p').live('click', fn);
```

```
<p>Ninja</p>
<p>Dizzy</p>
<script src="jquery.js"></script>
<script>
$('p').live('click', function () {
  $(this).after('<p>' + $(this).text()
    + ' clicked</p>');
});
</script>
```

Ninja

Dizzy

```
$('p').live('click', fn)
```

```
$('p').live('click', fn)
```

- 1. jQuery registers 'click' for live events**

```
$('p').live('click', fn)
```

- 1. jQuery registers 'click' for live events**
- 2. When click fires on the top level element, run the selector ('p')**

```
$('p').live('click', fn)
```

- 1. jQuery registers 'click' for live events**
- 2. When click fires on the top level element, run the selector ('p')**
- 3. If the \$(event.target).is(selector) then run handler**

live events

Now Full Coverage

- submit
- change
- focus (via "focusin")
- blur (via "focusout")

focusout

"This is distinct from the blur event in that it supports detecting the loss of focus from parent elements (in other words, it supports events bubbling).

delegate

```
$('root').delegate(selector, event, fn);
```

```
$('div').delegate('a', 'click', fn);
```

**div handles the
event delegation**

```
$('div').delegate('a', 'click', fn);
```

```
$('div').delegate('a', 'click', fn);
```

**watching for when
'a' elements are -**

- clicked.

```
$('div').delegate('a', 'click', fn);
```

```
$('div').delegate('a', 'click', fn);
```

**running the event
handler when that
happens**

.live is slow(er)

```
$('div').live('click', doStuff);
```

wasteful selection of 'div's



```
$('div').live('click', doStuff);
```

Better done using delegate

```
$('div').live('click', doStuff);
```

```
$ (document) . delegate (  
    'div' ,  
    'click' ,  
    doStuff  
);
```

```
$ . root = $( document );  
$ . root . delegate(  
    'div',  
    'click',  
    doStuff  
);
```

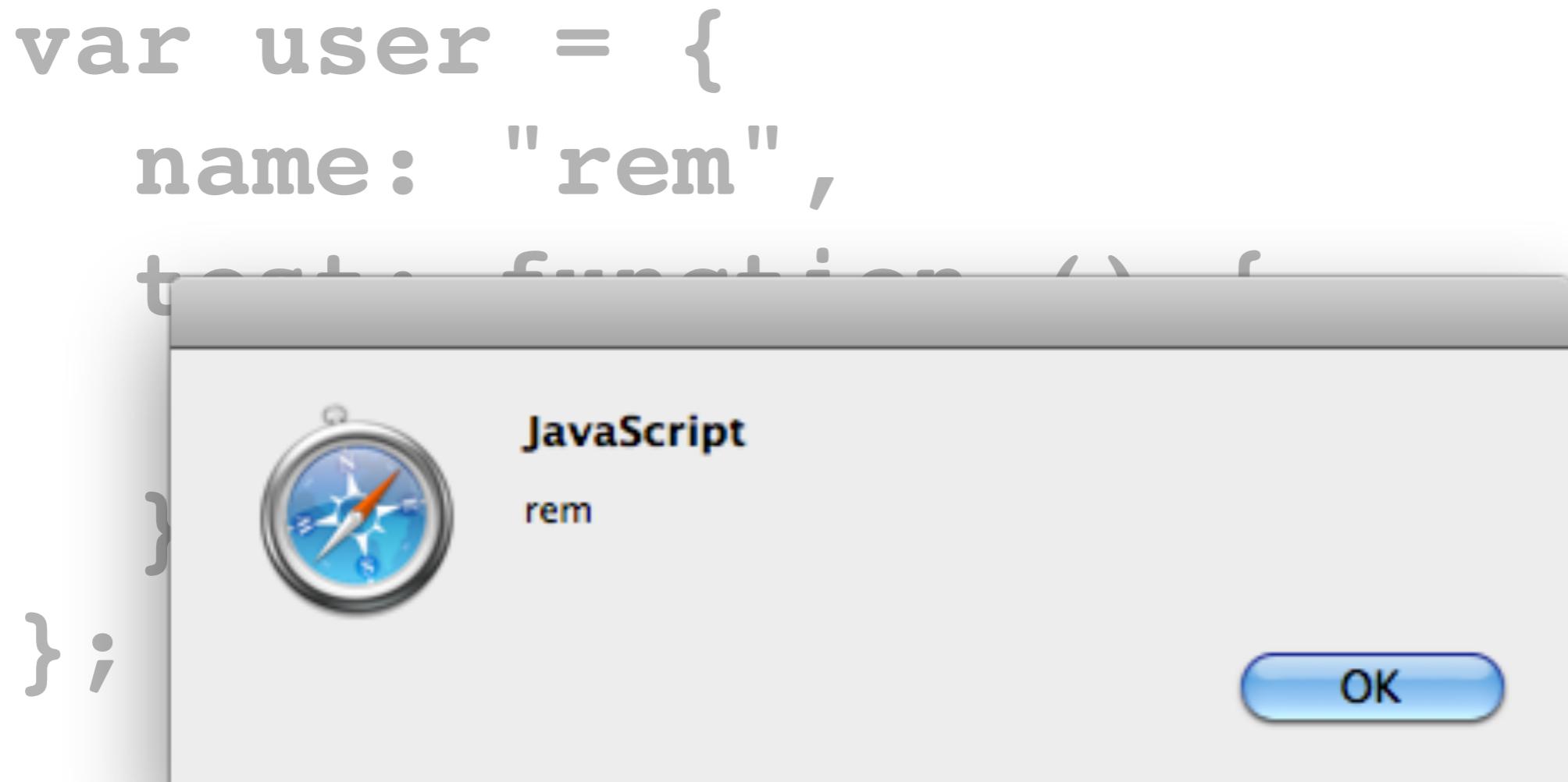
Hat tip: Cory Hart

\$.proxy

```
var user = {
    name: "rem",
    test: function () {
        alert(this.name);
  }
};
```

```
var user = {
  name: "rem",
  test: function () {
    alert(this.name);
  }
};
```

user.test();



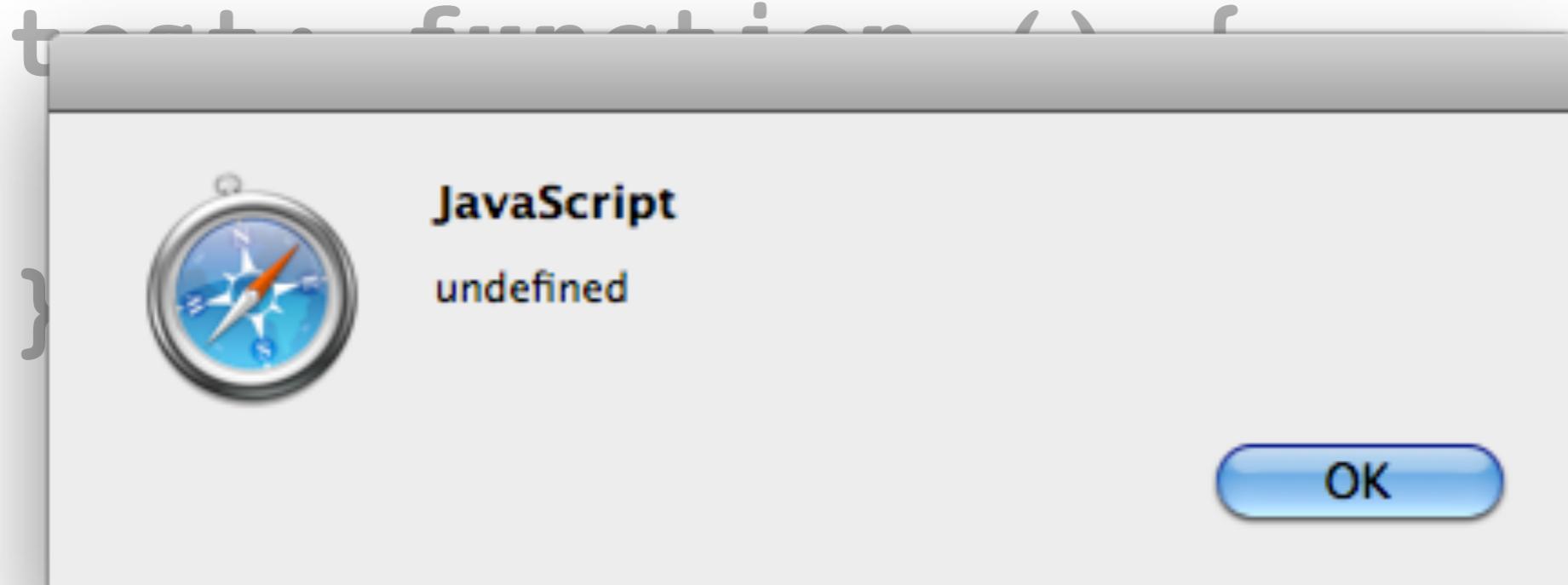
```
user.test();
```

```
var user = {  
    name: "rem",  
    test: function () {  
        alert(this.name);  
    }  
};
```

```
// user.test();  
$('a').click(user.test);
```

```
var user = {  
    name: "rem",  
    test: function () {  
        return undefined;  
    }  
};
```

Fail :(



```
// user.test();  
$('a').click(user.test);
```

```
var user = {  
    name: "rem",  
    test: function () {  
        alert(this.name);  
    }  
};
```

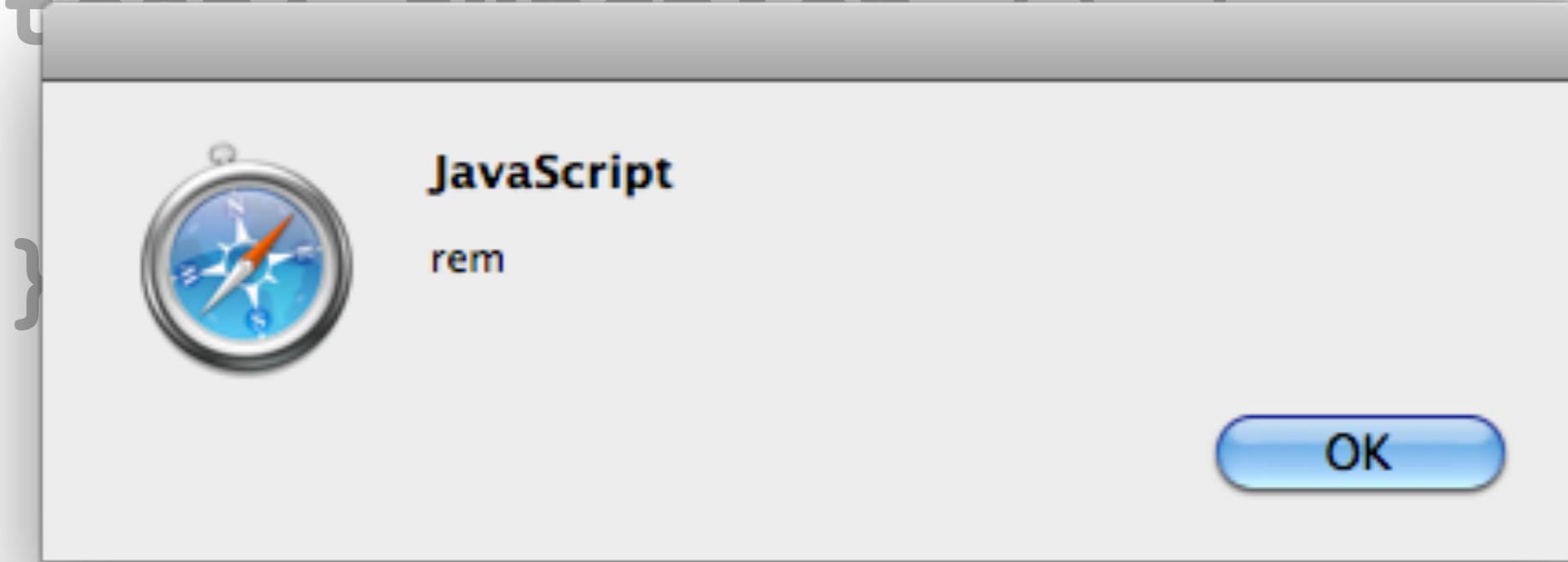
Fail :(

```
// user.test();  
$('a').click(user.test);
```

```
var user = {
    name: "rem",
    test: function () {
        alert(this.name);
    }
};

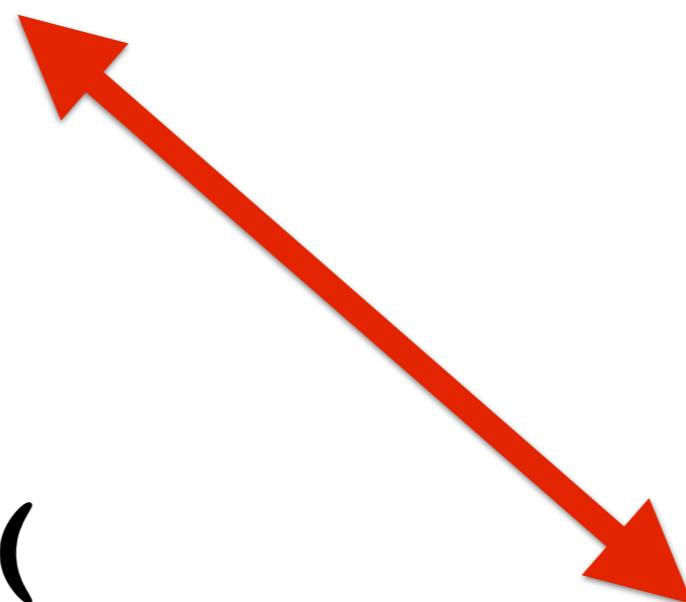
$('a').click(
    $.proxy(user.test, user)
);
```

```
var user = {  
    name: "rem",  
    test: function () {  
        alert("JavaScript  
rem")  
    }  
};  
  
$('a').click(  
    $.proxy(user.test, user)  
);
```



```
var user = {  
    name: "rem",  
    test: function () {  
        alert(this.name);  
    }  
};
```

```
$('a').click(  
    $.proxy(user.test, user)  
)
```



What about *that* this?

```
var user = {
    name: "rem",
    test: function () {
        alert(this.name);
    }
};

$('a').click(
    $.proxy(user.test, user)
);
```

```
var user = {
    name: "rem",
    test: function (event) {
        alert(event.currentTarget);
        alert(this.name);
    }
};

$('a').click(
    $.proxy(user.test, user)
);
```

Per property easing

```
$(' #ball ').animate( {  
    top: 500,  
    left: 200,  
, 2000,  
'easeOutBounce' );
```



```
$( '#ball' ).animate( {
  top: [ 500, 'easingOutBounce' ],
  left: 500
}, 2000);
```

```
$(' #ball ') . animate( {  
    top: [500, 'easingOutBounce'] ,  
    left: 500  
}, 2000);
```



delay

```
$(' #ball')  
  .animate({  
    left: 400,  
  }, 200)  
  .animate({  
    top: 400,  
  }, 200)  
  .animate({  
    left: 50,  
  }, 200)  
  .animate({  
    top: 50,  
  }, 200);
```



```
$(' #ball ')\n  .animate({\n    left: 250,\n  }, 200)\n  .delay(1000)\n  .animate({\n    top: 250,\n  }, 200)\n  .delay(1000)\n  .animate({\n    left: 50,\n  }, 200)\n  .delay(1000)
```



What's not new

```
$('div').length
```

```
$('remy').length
```

```
$ .fn .doStuff = function () {  
    return  
        this.each(function () {  
            // do...stuff  
        }) ;  
};
```

```
$ .fn .doStuff = function () {  
    return this.length ?  
        this.each(function () {  
            // do...stuff  
        }) :  
        this;  
};
```

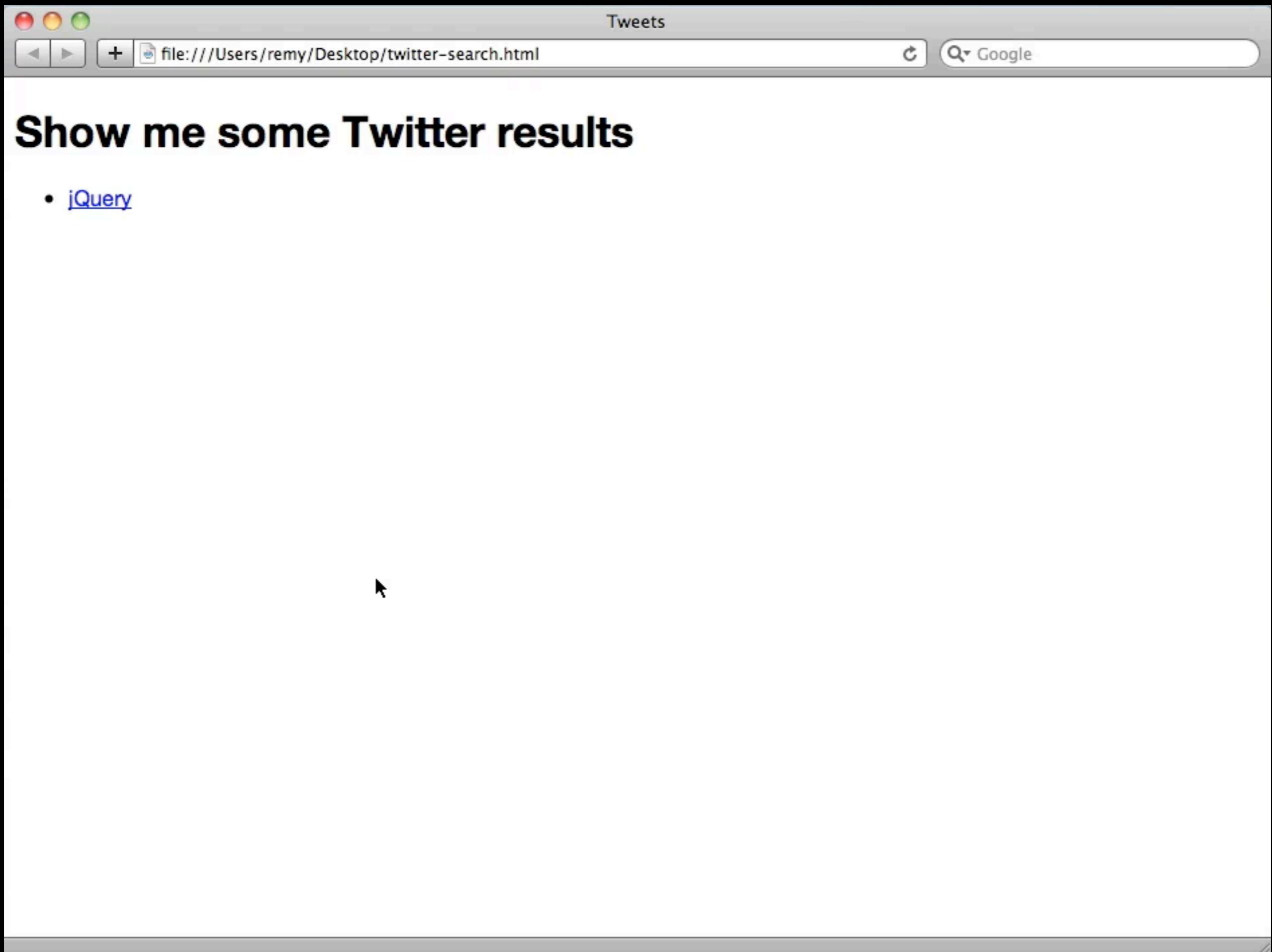
```
$ .fn .doStuff = function () {  
    return this.length ?  
        this.each(function () {  
            // do...stuff  
        }) :  
        this;  
};
```

```
$ .fn .doStuff = function () {  
    return this.length ?  
        this.each(function () {  
            // do...stuff  
        }) :  
        this;  
};
```

\$. fx . off = true

Loading...

Ajax events



Ajax events

- 1. ajaxStart**
- 2. ajaxSuccess (or ajaxError)**
- 3. ajaxComplete**

```
$('#status').ajaxStart(function () {  
    $(this).fadeIn();  
}).ajaxComplete(function () {  
    $(this).fadeOut();  
});
```

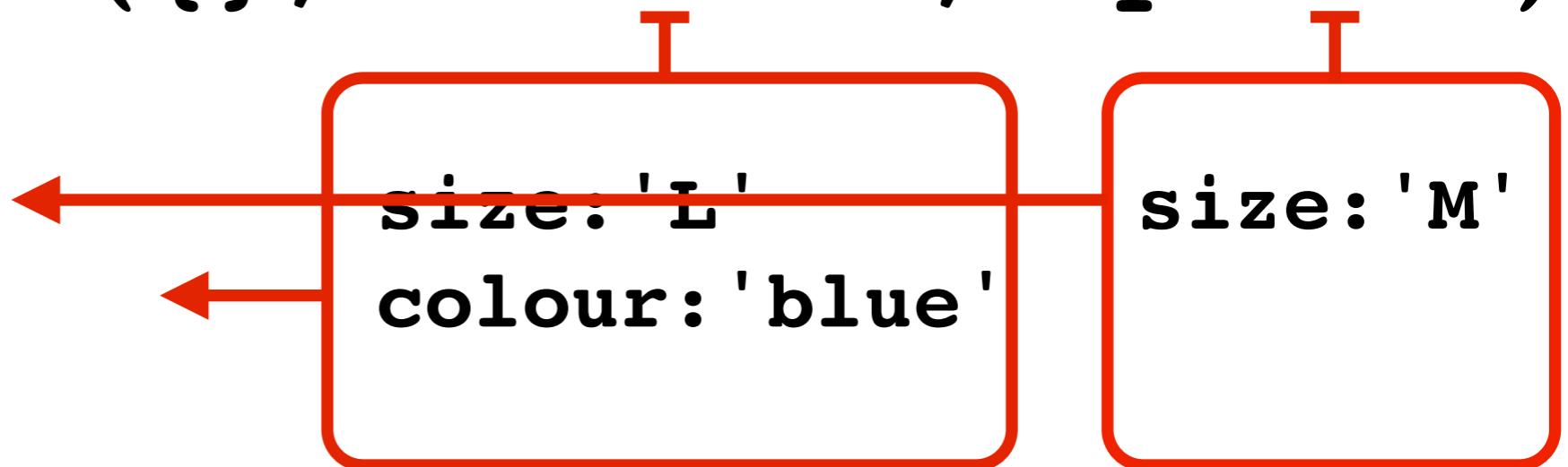
\$. extend

```
s = $.extend({}, defaults, options);
```

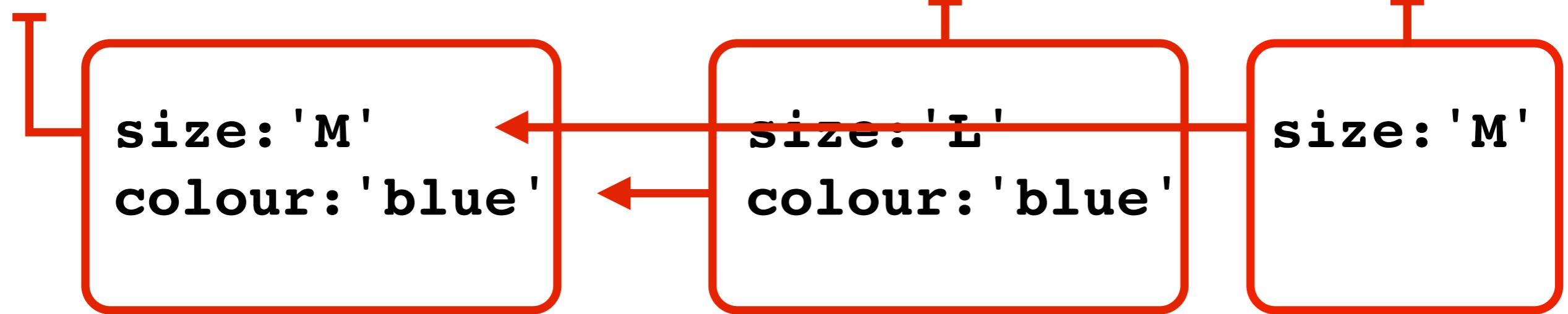
size: 'L'
colour: 'blue'

size: 'M'

```
s = $.extend({}, defaults, options);
```

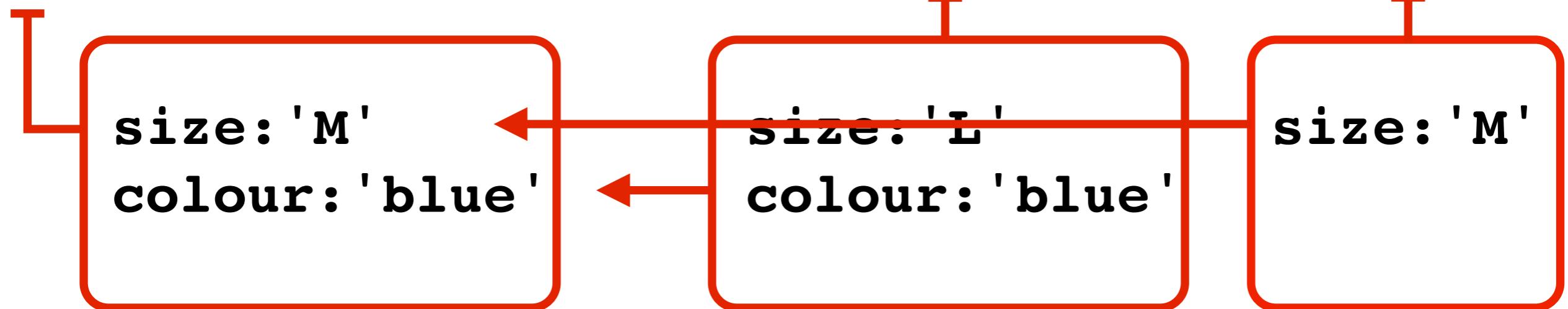


```
s = $.extend( {}, defaults, options );
```

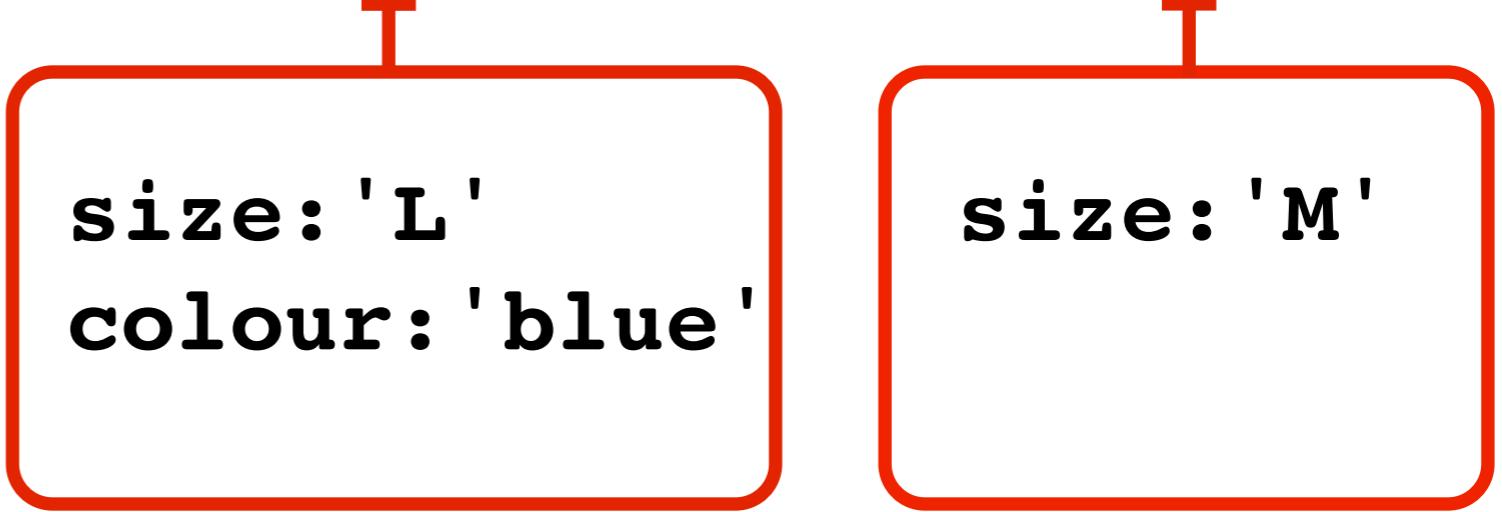


's' is the result of {} extended

```
s = $.extend( {}, defaults, options );
```



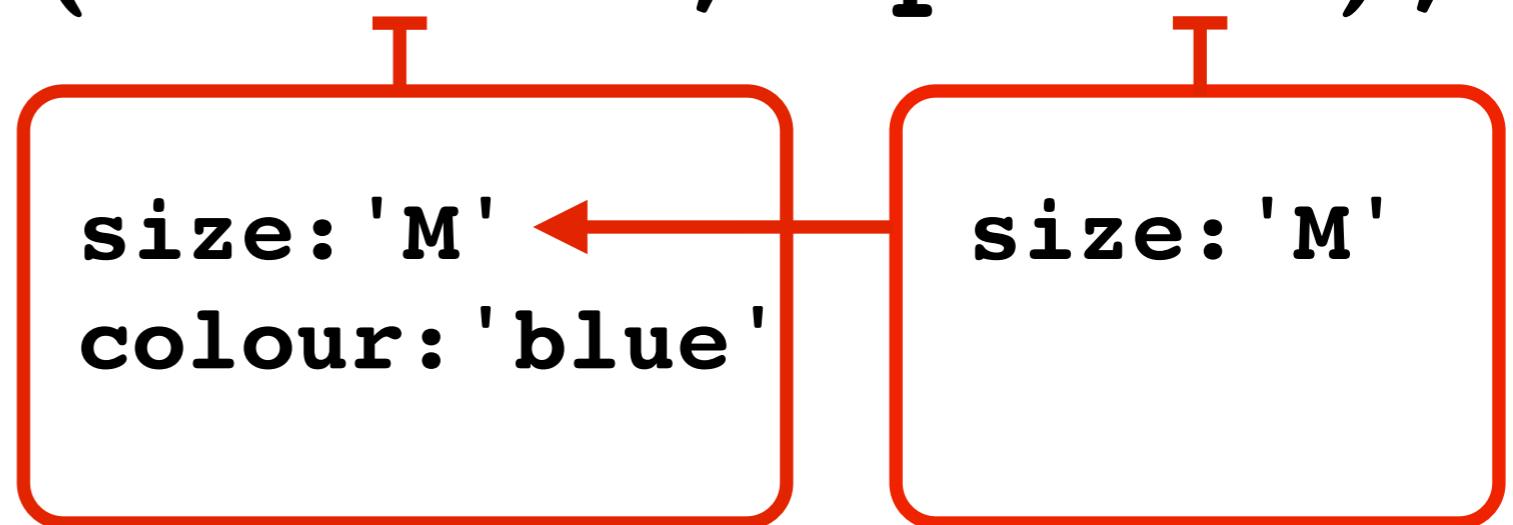
```
s = $.extend(defaults, options);
```



```
size: 'L'  
colour: 'blue'
```

```
size: 'M'
```

```
s = $.extend(defaults, options);
```



```
s = $.extend(defaults, options);
```

s === defaults

size: 'M'
colour: 'blue'

size: 'M'

```
var linkTemplate = {
  mouseover: function () {
    // default mouse over stuff
  },
  focus: function () {
    // default focus
  }
  click: function () {
    // do default click stuff
  }
};
```

```
$('<a/>', $.extend({},linkTemplate,{  
  click: function () {  
    // do custom click  
  }  
}));
```

```
$('<a/>', $.extend({},linkTemplate,{  
  click: function () {  
    // do custom click  
  }  
}));
```

```
$('<a/>', $.extend({},linkTemplate,{  
  click: function () {  
    // do custom click  
  }  
}));
```

```
$('<a/>', $.extend({},linkTemplate,{  
  click: function () {  
    // do custom click  
  }  
}));
```

```
$('<a/>', $.extend({},linkTemplate,{  
  click: function () {  
    // do custom click  
  }  
}));
```

7 deadly sins

1

.live or .delegate?

live vs. delegate

- Wasteful selection
 - Doesn't chain
 - Doesn't work with DOM elements
- No wasteful selection
 - Less distance for event to bubble
 - Chains

Use .delegate

2

```
$('div').get(0)
```

```
// Get the Nth element in the matched element set OR
// Get the whole matched element set as a clean array
get: function( num ) {
    return num == null ?
        // Return a 'clean' array
        this.toArray() :
        // Return just the object
        ( num < 0 ? this.slice(num)[ 0 ] : this[ num ] );
},
```

```
// Get the Nth element in the matched element set OR
// Get the whole matched element set as a clean array
get: function( num ) {
    return num == null ?
        // Return a 'clean' array
        this.toArray() :
        // Return just the object
        ( num < 0 ? this.slice(num)[ 0 ] : this[ num ] );
},
```

```
$('div')[0]
```

3

```
$('div').click(fn)
```

```
jQuery.each( "blur focus focusin focusout load resize scroll unload click dblclick " +
"mousedown mouseup mousemove mouseover mouseout mouseenter mouseleave " +
"change select submit keydown keypress keyup error").split(" "), function( i, name ) {

// Handle event binding
jQuery.fn[ name ] = function( fn ) {
    return fn ? this.bind( name, fn ) : this.trigger( name );
};

if ( jQuery.attrFn ) {
    jQuery.attrFn[ name ] = true;
}
});
```

```
jQuery.each( "blur focus focusin focusout load resize scroll unload click dblclick " +  
"mousedown mouseup mousemove mouseover mouseout mouseenter mouseleave " +  
"change select submit keydown keypress keyup error").split(" "), function( i, name ) {  
  
    // Handle event binding  
    jQuery.fn[ name ] = function( fn ) {  
        return fn ? this.bind( name, fn ) : this.trigger( name );  
    };  
  
    if ( jQuery.attrFn ) {  
        jQuery.attrFn[ name ] = true;  
    }  
});
```

```
$('div').bind('click', fn)
```

4

```
$ (this) . data ( 'key' )
```

```
$(this).data(  
    key1: 'foo',  
    key2: 'bar'  
) ;
```

```
var obj = $(this).data();
obj.key1; // 'foo'
```

```
data: function( key, value ) {
  if ( typeof key === "undefined" && this.length ) {
    return jQuery.data( this[0] );
  } else if ( typeof key === "object" ) {
    return this.each(function() {
      jQuery.data( this, key );
    });
}
} 
```

```
data: function( key, value ) {
  if ( typeof key === "undefined" && this.length ) {
    return jQuery.data( this[0] );
  } else if ( typeof key === "object" ) {
    return this.each(function() {
      jQuery.data( this, key );
    });
  }
}
```

```
$ .data(this, 'key')
```

```
var obj = $.data(this);  
obj.key1; // "foo"
```



5

**CSS belongs in
style sheets.**

```
$('.foo').css({  
    position: 'absolute',  
    top: 0,  
    left: '50%',  
    background: '#c00'  
});
```

```
.foo {  
    position: absolute;  
    top: 0;  
    left: 50%;  
    background: #c00;  
}
```

```
$('div').addClass('foo');
```

6

```
$('div', '#foo')
```

```
// HANDLE: $(expr, context)
// (which is just equivalent to: $(context).find(expr)
} else {
    return jQuery( context ).find( selector );
}
```

```
$(' #foo').find('div')
```



Avoid the DOM

**The DOM is slow,
reduce when you
touch it.**

```
$ .each(tweets, function (tweet) {  
  $('#tweets').append(tweet);  
});
```

```
var html = [];
$.each(tweets, function (tweet) {
    html.push(tweet);
});
$('#tweets').append(html.join(''))
```

```
var html = [];
$.each(tweets, function (tweet) {
    html.push(tweet);
});
$('#tweets').append(html.join(''))
```

slides: <http://j.mp/jquery-rs>

You've now been
jQuerified!

Any questions?

@rem

remy@leftlogic.com