

jquery

In with the new,
out with the old.

Agenda

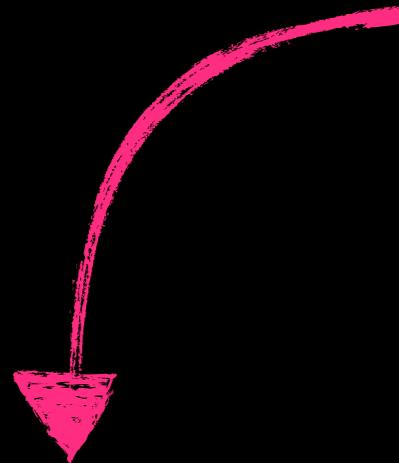
- Understanding jQuery
- What's new & interesting
- Where you don't need it

**Grokking
jQuery**

DOM library

```
$('div').append(html).click(doStuff);
```

Just a CSS selector



```
$('div').append(html).click(doStuff);
```

```
$('div').append(html).click(doStuff);
```

```
$('div').append(html).click(doStuff);
```

```
if ( $( 'foo' ) ) {  
    doAmazing();  
}
```

1) `$('foo')`

2) `$()`

3) `[]`

4) `[] .join(', ') // works`

```
if ( $( 'foo' ) ) {  
    doAmazing();  
}
```

```
if ( $('foo').length ) {  
    doAmazing();  
}
```

**If your jquery code fails,
and there's no error:**

it's the selector

DOM Navigation

Find

changes the collection

Filter

creates a subset

debugging

Firefox File Edit View History Bookmarks Tools Window Help JS Bin* http://offline.jsbin.com/ Google

Code Render Save Download View: JavaScript HTML Real-time preview Help & tutorials

JavaScript

```
if (document.getElementById('hello')) {  
  document.getElementById('hello').innerHTML = 'Hello World - this wa  
}  
  
!
```

HTML

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset=utf-8 />  
<title>JS Bin</title>  
<!--[if IE]>  
  <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">  
<![endif]-->  
<style>  
  article, aside, figure, footer, header, hgroup,  
  menu, nav, section { display: block; }  
</style>  
</head>  
<body>  
  <p id="hello">Hello World</p>  
</body>
```

Console HTML CSS Script DOM Net

Edit html

```
<!DOCTYPE html>  
▼ <html lang="en">  
  ▶ <head>  
  ▶ <body class="source beta" style="background-color: #f0f0f0;"> jQuery15205336018542333171=Object { events=[...]}  
    ▶ <div id="streaming">  
    ▶ <div id="control">  
      ▶ <div class="control">  
        ▶ <div class="buttons">  
          <a class="tab button source group left" href="#source" accesskey="1" style="background-color: #e0e0e0;"> Code </a> jQuery15205336018542333171=Object { events=[...]}  
          <a class="tab button preview group right gap" href="#preview" accesskey="2" style="background-color: #e0e0e0;"> Render </a> jQuery15205336018542333171=Object { events=[...]}  
          ▶ <a id="revert" class="button light group left enable" href="#" title="Revert" style="display: inline;"> Revert </a> jQuery15205336018542333171=Object { odddisplay="inline", events=[...]}  
          <a id="save" class="button save group" href="/save" style="background-color: #e0e0e0;"> Save </a> jQuery15205336018542333171=Object { events=[...]}  
          <a id="download" class="button download group right light gap" href="#" style="background-color: #e0e0e0;"> Download </a> jQuery15205336018542333171=Object { events=[...]}  
          <a id="stream" class="button left" href="#" style="background-color: #e0e0e0;"> Stream </a> jQuery15205336018542333171=Object { events=[...]}  
        
```

**NEW
stuff**

Ajax changes

& deferreds

```
$ .get('foo.html', function (html) {  
var jqXHR = $ .get('foo.html');  
$('#latest') .append(html);  
});  
jqXHR.done(function (html) {  
  $('#latest') .append(html);  
});
```

```
$.ajax({  
  url: 'foo.json',  
  dataType: 'json',  
  success: function () {  
    // this === xhr  
  },  
  error: function () {  
  }  
});
```

```
$.ajax({  
  url: 'foo.json',  
  dataType: 'json',  
  context: document.body,  
  success: function () {  
    // this === body element  
  },  
  error: function () {  
  }  
});
```

```
var jqXHR = $.ajax({  
  url: 'foo.json',  
  dataType: 'json',  
  context: document.body  
});
```

```
jqXHR.then(success, fail);
```

```
var jqXHR = $.ajax({  
  url: 'foo.json',  
  dataType: 'json',  
  context: document.body  
});
```

```
jqXHR.then(success, fail);
```

```
// much later in the code
```

```
jqXHR.done(success2);
```

**jqXHR is a
promise**



rem Remy Sharp

Anyone fancy the challenge of defining a promise (computing) in layman terms
(and fits in a tweet!)?

11 minutes ago

— in reply to @rem ↑



@tobie

Tobie Langel

@rem A promise lets you delegate work to someone else knowing you're guaranteed to receive a call once the work's finished or if it failed.

3 minutes ago via Twitter for Mac  Favorite  Retweet  Reply



tobie Tobie Langel

@rem A promise lets you delegate work to someone else knowing you're guaranteed to receive a call once the work's finished or if it failed.

9 minutes ago



— in reply to @tobie ↑



@justinbmeyer

justinbmeyer

@tobie @rem Although useful without values, for me, a promise is less about work and callbacks, and more about getting a value.

5 minutes ago via web Favorite Retweet Reply

```
var jqXHR = $.ajax({  
  url: 'foo.json',  
  dataType: 'json',  
  context: document.body  
});
```

// much later in the code

```
jqXHR.done(success);
```

```
.done(ok)      // success  
.fail(fail)   // error  
.always(fn)   // complete  
.then(ok, fail)
```

```
var dfd = $.Deferred();  
  
.resolve  
.reject  
.promise
```

```
var jqXHR = $.ajax({  
  url: 'foo.json',  
  dataType: 'json',  
  context: document.body  
});
```

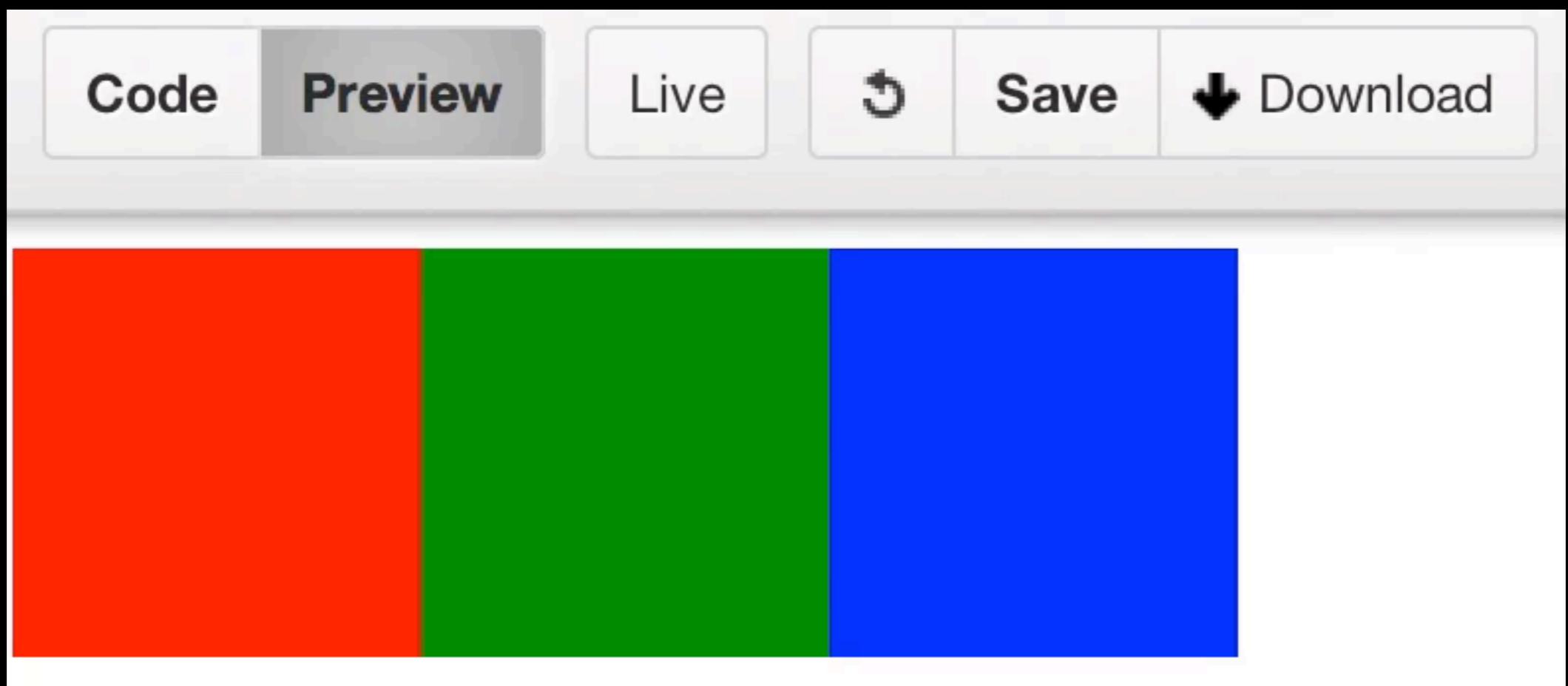
// much later in the code

```
jqXHR.done(success);
```

```
function save(data) {  
    var dfd = new $.Deferred();  
  
    // some validation done...  
  
    if (!valid) {  
        dfd.reject(reason);  
    } else {  
        $.ajax({  
            url: '/app/save',  
            data: data,  
            success: dfd.resolve,  
            error: dfd.reject  
        });  
    }  
  
    return dfd.promise();  
}
```

```
unction save(data) {  
var dfd = new $.Deferred();  
  
// some validation done...  
  
if (!valid) {  
dfd.reject(reason);  
} else {  
$.ajax({  
url: '/app/save',  
data: data,  
success: dfd.resolve,  
error: dfd.reject  
});  
}  
  
return dfd.promise();  
  
save({ /* some data */ })  
.done(function () {  
// show it's all good  
})  
.fail(function (reason) {  
// shows reason  
});
```

\$.when



\$.when

```
var $boxes = $('div'),  
    rand = Math.random,  
    dfds = [];  
  
for (var i = 0; i < 3; i++) {  
    dfds.push($.Deferred());  
    $boxes.eq(i).fadeTo(rand() * 5000, 1, dfds[i].resolve);  
}  
  
$.when.apply($, dfds).done(function () {  
    alert('all done!');  
});
```

\$. sub()

```
(function ($) {
  $ = $.sub(); // sandbox $.fn

  $.fn.logger = function () {
    return this.each(function(){
      console.log(this)
    });
  }

  // my code that uses .logger()
})(jQuery);

// now: $.fn.logger === undefined
```

```
(function ($) {
  $ = $.sub(); // sandbox $.fn

  $.fn.logger = function () {
    return this.each(function(){
      console.log(this)
    });
  }
}

// my code that uses .logger()
})(jQuery);
```

```
// now: $.fn.logger === undefined
```

doing it

wrong

.ready?

```
...
<!-- all my funky markup -->
...
<script>
$(document).ready(function () {
    // add the extra funk
});
</script>
</body>
</html>
```

—.ready—

```
...
<!-- all my funky markup -->
...
<script>
    // add the extra funk

</script>
</body>
</html>
```

Add class

```
...
<!-- all my funky markup -->
...
<script>
$( '#foo' ).addClass( 'amazing' );
</script>
</body>
</html>
```

Add class

```
...
<!-- all my funky markup -->
...
<script>

var f = document.getElementById('foo');
foo.className += ' amazing';

</script>
</body>
</html>
```

Effects vs. CSS

- If the browser can do it natively, let it do it natively.
- No one lost business because IE didn't do a cross fade.

return false

```
$('a').click(function () {  
    // do the magic  
    amazingMagic();  
  
    return false;  
});
```

return false

- Do you know what's happening?
- `event.preventDefault()`
- `event.stopPropagation()`

Get to know this

```
$('a').click(function (e) {  
  e.preventDefault();  
  
  var href = $(this).attr('href');  
  // do the magic  
  amazingMagic(href);  
});
```

Get to know this

```
$('a').click(function (e) {  
  e.preventDefault();  
  
  var href = this.href;  
  // do the magic  
  amazingMagic(href);  
});
```

Poorly designed plugins

```
$.fn.myplugin = function () {
  var me = $(this).each(function() {
    return $(this).bind('someEvent', function () {
      // does something
    });
  });

  return me;
};
```

```
$.fn.myplugin = function () {  
    var me = $(this).each(fn);  
  
    return me;  
};
```

```
$.fn.myplugin = function () {
  return $(this).each(fn);
};
```

```
$.fn.myplugin = function () {
  return $(this).each(function() {
    return $(this).bind('someEvent', function () {
      // does something
    });
  });
};
```

```
$.fn.myplugin = function () {  
    return $(this).each(function() {  
        return $(this).bind('someEvent', function () {  
            // does something  
        });  
    });  
};
```

```
$.fn.myplugin = function () {
  return this.each(function() {
    return $(this).bind('someEvent', function () {
      // does something
    });
  });
};
```

```
$.fn.myplugin = function () {  
    return this.each(function() {  
        return $(this).bind('someEvent', function () {  
            // does something  
        });  
    });  
};
```

```
$.fn.myplugin = function () {
  return this.each(function() {
    $(this).bind('someEvent', function () {
      // does something
    });
  });
};
```

```
$.fn.myplugin = function () {  
    return this.each(function() {  
        $(this).bind('someEvent', function () {  
            // does something  
        });  
    });  
};
```

```
$.fn.myplugin = function () {
  return this.bind('someEvent', function () {
    // does something
  });
};
```

```
(function ($) {
  $.fn.myplugin = function () {
    return this.bind('someEvent', function () {
      // does something
    });
  };
})(jQuery);
```

Questions?

@rem

remy@leftlogic.com