

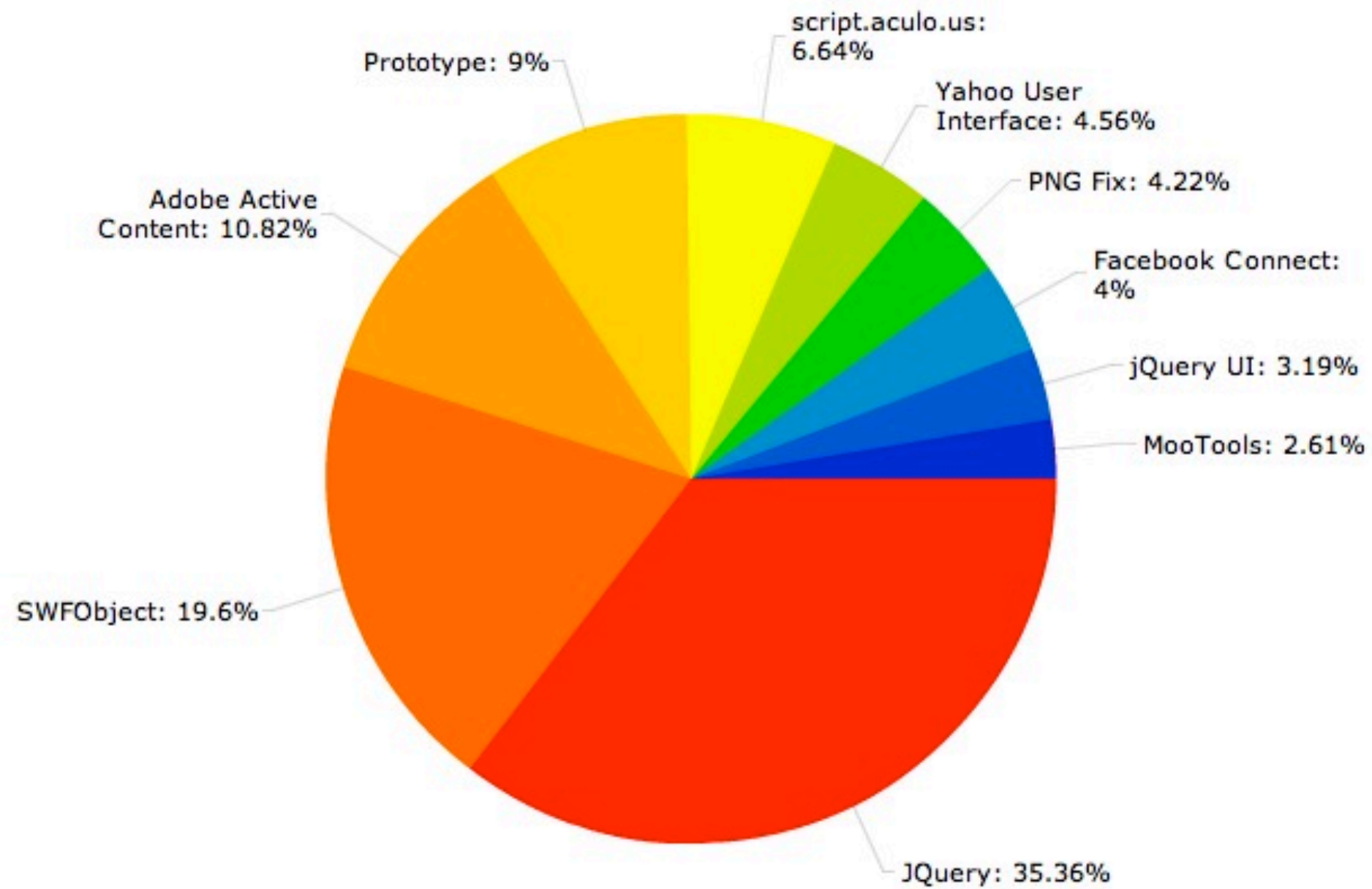
jQuery <3 Developers

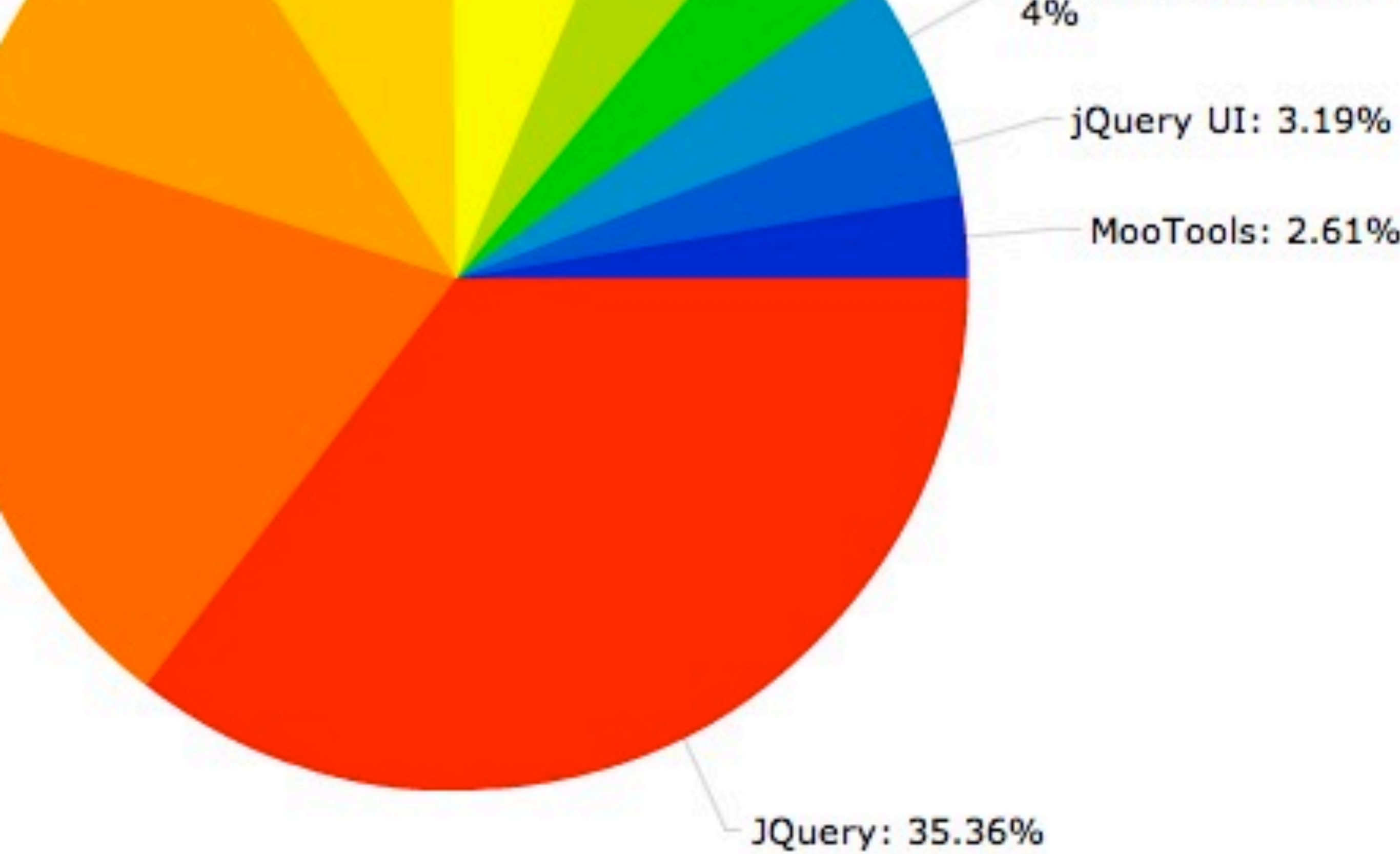
Overview

- Who, what, why of jQuery
- Core concepts of jQuery
- API overview & tips
- Plugin design pattern
- News
- Live demo

Quick survey first

Who?





Who?

reddit.com

espn.com

ibm.com

stackoverflow.com

boxee.tv

bit.ly

twitpic.com

whitehouse.gov

microsoft.com

amazon.com

netflix.com

bing.com

monster.com

tv.com

overstock.com

time.com

capitalone.com

wordpress.com

dell.com

twitter.com

w3.org

<http://trends.builtwith.com/javascript/jquery>

What?

jQuery is a JavaScript library.

- Selector engine: Sizzle
- DOM manipulation
- Events
- Effects
- Ajax

What does that mean?

It means no more of this

```
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

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

jQuery simplifies

jQuery function

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

jQuery simplifies

jQuery function

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

CSS expression

jQuery simplifies

jQuery function

jQuery method

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

CSS expression

jQuery simplifies

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

Why use jQuery

- Helps us to simplify and speed up web development
- Allows us to avoid common headaches associated with browser development
- Provides a large pool of plugins
- Large and active community
- Tested on 50 browsers, 11 platforms
- For both coder and designer (we don't discriminate)

Why use jQuery

- ~~Helps us to simplify and speed up web development~~
- ~~Allows us to avoid common headaches associated with browser development~~
- Provides a large pool of plugins
- Large and active community
- Tested on 50 browsers, 11 platforms
- For both coder and designer (we don't discriminate)

Help!

APIs

docs.jquery.com

api.jquery.com

visualjquery.com

Blogs, tutorials, screencasts,
plugins, development sprints

Google Groups

[jquery-en](#)

[jquery-dev](#)

[jquery-ui-en](#)

[jquery-ui-dev](#)

[jquery-all](#)

Help!

Twitter

[@jquery](#)

[@jquerysites](#)

[@jqueryui](#)

IRC channel

irc.freenode.net/#jquery

APIs

docs.jquery.com

api.jquery.com

visualjquery.com

Blogs, tutorials, screencasts,
plugins, development sprints

Google Groups

[jquery-en](#)

[jquery-dev](#)

[jquery-ui-en](#)

[jquery-ui-dev](#)

[jquery-all](#)

Help!

Twitter

[@jquery](#)

[@jquerysites](#)

[@jqueryui](#)

IRC channel

irc.freenode.net/#jquery

Concept 1:

***knowing the jQuery
parameter types***

- CSS selectors & custom CSS expressions
- HTML
- DOM elements
- A function (shortcut for DOM ready)

jQuery('div') & jQuery(' :first')

- CSS selectors & custom CSS expressions
- HTML
- DOM elements
- A function (shortcut for DOM ready)

```
jQuery( '<li><a href="#">link</a></li>' )
```

- CSS selectors & custom CSS expressions

```
jQuery( 'div' ) & jQuery( ':first' )
```

- HTML

- DOM elements

- A function (shortcut for DOM ready)

`jQuery(document)` or `jQuery(document.createElement('a'))`

- CSS selectors & custom CSS expressions

`jQuery('div') & jQuery(':first')`

- HTML

`jQuery('link')`

- DOM elements

- A function (shortcut for DOM ready)

`jQuery(function(){}) =
jQuery(document).ready(function(){})`

- CSS selectors & custom CSS expressions

`jQuery('div') & jQuery(':first')`

- HTML

`jQuery('link')`

- DOM elements

`jQuery(document) or jQuery(document.createElement('a'))`

- A function (shortcut for DOM ready)

- CSS selectors & custom CSS expressions

`jQuery('div') & jQuery(':first')`

- HTML

`jQuery('link')`

- DOM elements

`jQuery(document) or jQuery(document.createElement('a'))`

- A function (shortcut for DOM ready)

`jQuery(function(){}) =
jQuery(document).ready(function(){})`

Concept 2:

find something,
do something

```
<!DOCTYPE html>
<html>
<body>
  <ul id="nav">
    <li><a>home</a></li>
    <li><a>about</a></li>
  </ul>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id="nav">
    <li><a>home</a></li>
    <li><a>about</a></li>
  </ul>
  <script src="jquery.js"></script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id="nav">
    <li><a>home</a></li>
    <li><a>about</a></li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    jQuery( '#nav li' );
  </script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id="nav">
    <li class="item"><a>home</a></li>
    <li class="item"><a>about</a></li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    jQuery( '#nav li' ).addClass( 'item' );
  </script>
</body>
</html>
```

Concept 3:

create something,
do something


```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    </ul>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    </ul>
<script src="jquery.js"></script>
<script>
  jQuery( '<li>home</li>' );
  jQuery( '<li>about</li>' );
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li>home</li>
    <li>about</li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    jQuery( '<li>home</li>' )
    ➡ .appendTo( '#nav' );
    jQuery( '<li>about</li>' )
    ➡ .appendTo( '#nav' );
  </script>
</body>
</html>
```



createDocumentFragment



createDocumentFragment

```
(IE6 || IE7) &&  
innerHTML && HTML5  
== fail
```

Concept 4:

chaining & operating

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item"><a href="/home">home</a></li>
    <li class="item"><a href="/about">about</a></li>
  </ul>
<script src="jquery.js"></script>
<script>
  jQuery('ul').attr('id', 'nav');
  jQuery('#nav li').addClass('item');
  jQuery('#nav a').each(function () {
    jQuery(this).attr('href', '/' + jQuery(this).text());
  });
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item"><a href="/home">home</a></li>
    <li class="item"><a href="/about">about</a></li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    1 jQuery('ul').attr('id', 'nav');
    jQuery('#nav li').addClass('item');
    jQuery('#nav a').each(function () {
      jQuery(this).attr('href', '/' + jQuery(this).text());
    });
  </script>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item"><a href="/home">home</a></li>
    <li class="item"><a href="/about">about</a></li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    1 jQuery('ul').attr('id', 'nav');
    2 jQuery('#nav li').addClass('item');
    jQuery('#nav a').each(function () {
      jQuery(this).attr('href', '/' + jQuery(this).text());
    });
  </script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item"><a href="/home">home</a></li>
    <li class="item"><a href="/about">about</a></li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    jQuery('ul').attr('id', 'nav');
    jQuery('#nav li').addClass('item');
    jQuery('#nav a').each(function () {
      jQuery(this).attr('href', '/' + jQuery(this).text());
    });
  </script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item"><a href="/home">home</a></li>
    <li class="item"><a href="/about">about</a></li>
  </ul>
<script src="jquery.js"></script>
<script>
  jQuery('ul').attr('id', 'nav');
  jQuery('#nav li').addClass('item');
  jQuery('#nav a').each(function () {
    jQuery(this).attr('href', '/' + jQuery(this).text());
  });
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item"><a href="/home">home</a></li>
    <li class="item"><a href="/about">about</a></li>
  </ul>
<script src="jquery.js"></script>
<script>
  jQuery('ul')
    .attr('id', 'nav')
    .find('li')
    .addClass('item')
    .find('a')
    .each(function () {
      jQuery(this).attr('href', '/' + jQuery(this).text());
    });
</script>
</body>
</html>
```

**jQuery returns itself
containing the current
DOM collection**

Concept 5:

***understanding
implicit iteration***

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item"><a href="/home">home</a></li>
    <li class="item"><a href="/about">about</a></li>
  </ul>
<script src="jquery.js"></script>
<script>
  jQuery( 'ul' )
    .attr( 'id', 'nav' )
    .find( 'li' )
    .addClass( 'item' )
    .find( 'a' )
    .each( function () {
      jQuery( this ).attr( 'href', '/' + jQuery( this ).text() );
    } );
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item"><a href="/home">home</a></li>
    <li class="item"><a href="/about">about</a></li>
  </ul>
<script src="jquery.js"></script>
<script>
  jQuery('ul')
    .attr('id', 'nav')
    .find('li')
    .addClass('item')
    .find('a')
    .each(function () {
      jQuery(this).attr('href', '/' + jQuery(this).text());
    });
</script>
</body>
</html>
```


Review

- Knowing the jQuery parameter types
- Find something, do something
- Create something, do something
- Chaining & Operating
- Understanding Implicit Iteration



***“What about the
bling function?”***

jQuery == \$

\$ == jQuery

\$ is an alias to jQuery

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item">home</li>
    <li class="item">about</li>
  </ul>
<script src="jquery.js"></script>
<script>
  jQuery( 'li' ).addClass( 'item' );
</script>
</body>
</html>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
  <ul id='nav'>
```

```
    <li class="item">home</li>
```

```
    <li class="item">about</li>
```

```
  </ul>
```

```
<script src="jquery.js"></script>
```

```
<script>
```

```
jQuery( 'li' ).addClass( 'item' );
```

```
</script>
```

```
</body>
```

```
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item">home</li>
    <li class="item">about</li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    $( 'li' ).addClass( 'item' );
  </script>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item">home</li>
    <li class="item">about</li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    $( 'li' ).addClass( 'item' );
  </script>
</body>
</html>
```




More than ready

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item">home</li>
    <li class="item">about</li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    $( 'li' ).addClass( 'item' );
  </script>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item">home</li>
    <li class="item">about</li>
  </ul>
  <script src="jquery.js"></script>
  <script>
$(document).ready(function () {
  $('li').addClass('item');
});
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <ul id='nav'>
    <li class="item">home</li>
    <li class="item">about</li>
  </ul>
  <script src="jquery.js"></script>
  <script>
$(document).ready(function () {
  $('li').addClass('item');
});
</script>
</body>
</html>
```



app.js

```
$ (document) .ready (function () {  
    $ ( 'li' ) .addClass ( 'item' );  
} ) ;
```

app.js

```
$ ( document ) . ready ( function ( ) {  
    $ ( ' li ' ) . addClass ( ' item ' );  
} ) ;
```

app.js

```
$(function () {  
    $('li').addClass('item');  
});
```


app.js

```
jQuery(function ($) {  
    $('li').addClass('item');  
});
```

An orange starburst graphic with a yellow outline, containing the word "Tip" in white.

Tip

Include jQuery after styles

jQuery API overview

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

jQuery()

each()

size()

eq()

get()

index()

length

selector

context

data()

removeData()

queue()

dequeue()

jQuery.fn.extend()

jQuery.extend()

jQuery.noConflict()

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

jQuery()

each()

size()

eq()

get()

index()

length

selector

context

data()

removeData()

queue()

dequeue()

jQuery.fn.extend()

jQuery.extend()

jQuery.noConflict()

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html>
<html>
<body>

<p>Element Node</p>

<script src="jquery.js">
</script>
<script>
alert($('p').get(0).nodeName);
</script>

</body>
</html>
```

<http://jsbin.com/ibito/edit>

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html>
<html>
<body>

<p>Element Node</p>

<script src="jquery.js">
</script>
<script>
alert($('p').get(0).nodeName);
alert($('p')[0].nodeName);
</script>

</body>
</html>
```

<http://jsbin.com/idiyi/edit>

- Core
- **Selectors**
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities



- Core
- **Selectors**
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities



```
$ ( ' #nav li.contact ' )
```

- Core
- **Selectors**
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
$ ( ' #nav li.contact ' )
```

```
$ ( ' :visible ' )
```

- Core
- **Selectors**
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
$ ( ' #nav li.contact ' )
```

```
$ ( ' :visible ' )
```

```
$ ( ' :radio:enabled:checked ' )
```

- Core
- **Selectors**
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
$ ( ' #nav li.contact ' )
```

```
$ ( ' :visible ' )
```

```
$ ( ' :radio:enabled:checked ' )
```

```
$ ( ' a[title] ' )
```

- Core
- **Selectors**
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
$ ( ' #nav li.contact ' )
```

```
$ ( ' :visible ' )
```

```
$ ( ' :radio:enabled:checked ' )
```

```
$ ( ' a[title] ' )
```

```
$ ( ' a[title][hash*="foo"] ' )
```

- Core
- **Selectors**
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

`$ (' #nav li.contact ')`

`$ (' :visible ')`

`$ (' :radio:enabled:checked ')`

`$ (' a[title] ')`

`$ (' a[title][hash*="foo"] ')`

`$ (' a:first[hash*="foo"] ')`

- Core
- **Selectors**
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

`$ (' #nav li.contact ')`

`$ (' :visible ')`

`$ (' :radio:enabled:checked ')`

`$ (' a[title] ')`

`$ (' a[title][hash*="foo"] ')`

`$ (' a:first[hash*="foo"] ')`

`$ (' .header, .footer ')`

- Core
- **Selectors**
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

`$ (' #nav li.contact ')`

`$ (' :visible ')`

`$ (' :radio:enabled:checked ')`

`$ (' a[title] ')`

`$ (' a[title][hash*="foo"] ')`

`$ (' a:first[hash*="foo"] ')`

`$ (' .header, .footer ')`

`$ (' .header, .footer, '#main')`



Performance

getElementById

getElementsByTagName


getElementsByClassName

querySelectorAll

**Inside
tip**

**jQuery selectors
fail silently,
just like CSS does!**

- Core
- Selectors
- Attributes
- **Traversing**
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities



add()	eq()
children()	filter()
closest()	is()
contents()	map()
find()	not()
next()	slice()
nextAll()	
offsetParent()	
parent()	
parents()	
prev()	
prevAll()	
siblings()	
andSelf()	
end()	



- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

add()	eq()
children()	filter()
closest()	is()
contents()	map()
find()	not()
next()	slice()
nextAll()	
offsetParent()	
parent()	
parents()	
prev()	
prevAll()	
siblings()	
andSelf()	
end()	

- Core
- Selectors

Searches down

- **Traversing**
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

add()	eq()
children()	filter()
closest()	is()
contents()	map()
find()	not()
next()	slice()
nextAll()	
offsetParent()	
parent()	
parents()	
prev()	
prevAll()	
siblings()	
andSelf()	
end()	

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

Filters across

<code>add()</code>	<code>eq()</code>
<code>ch</code>	<code>filter()</code>
<code>closest()</code>	<code>is()</code>
<code>contents()</code>	<code>map()</code>
<code>find()</code>	<code>not()</code>
<code>next()</code>	<code>slice()</code>
<code>nextAll()</code>	
<code>offsetParent()</code>	
<code>parent()</code>	
<code>parents()</code>	
<code>prev()</code>	
<code>prevAll()</code>	
<code>siblings()</code>	
<code>andSelf()</code>	
<code>end()</code>	

- Core
- Selectors
- Attributes
- **Traversing**
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

add()	eq()
children()	filter()
closest()	is()
contents()	map()
find()	not()
next()	slice()
nextAll()	
offsetParent()	
parent()	
parents()	
prev()	
prevAll()	
siblings()	
andSelf()	
end()	

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html><html><body>
<ul>
  <li><a>one</a></li>
  <li><a>two</a></li>
  <li><a>three</a></li>
</ul>
<script src="jquery.js">
</script>
<script>
$( 'a' ).click(function () {
  alert( $(this)
    .parent()
    .prevAll()
    .length
  )
});
</script>
</body></html>http://jsbin.com/ubuhe/edit
```

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- **Events**
- Effects
- Ajax
- Utilities

ready()

bind()

one()

trigger()

triggerHandler()

unbind()

live()

die()

hover()

toggle()

load()

unload()

error()

blur()

change()

click()

dblclick()

focus()

keydown()

keypress()

keyup()

mousedown()

mouseenter()

mouseleave()

mouseout()

mouseup()

resize()

scroll()

select()

submit()

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- **Events**
- Effects
- Ajax
- Utilities

ready()

IE fires on blur

bind()

one()

trigger()

triggerHandler()

unbind()

live()

die()

hover()

toggle()

load()

unload()

error()

blur()

change()

click()

dblclick()

focus()

keydown()

keypress()

keyup()

mousedown()

mouseenter()

mouseleave()

mouseout()

mouseup()

resize()

scroll()

select()

submit()

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- **Events**
- Effects
- Ajax
- Utilities

ready()

bind()

one()

trigger()

triggerHandler()

unbind()

live()

die()

hover()

toggle()

load()

unload()

error()

blur()

change()

click()

dblclick()

focus()

keydown()

keypress()

keyup()

mousedown()

mouseenter()

mouseleave()

mouseout()

mouseup()

resize()

scroll()

select()

submit()

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- **Events**
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html>
<html>
<body>

<p>1. click</p>
<p>2. click</p>

<script src="jquery.js"></script>
<script>

$( 'p' ).bind( 'click', function() {
    $(this).after( '<p>' +
        ➡ $(this).text() + ' clicked</p>' );
    });

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

<http://jsbin.com/ededi/edit>

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- **Events**
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html>
<html>
<body>

<p>1. click</p>
<p>2. click</p>

<script src="jquery.js"></script>
<script>

$( 'p' ).live( 'click', function() {
    $(this).after( '<p>' +
        ➡ $(this).text() + ' clicked</p>' );
    });

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

<http://jsbin.com/ihalu/edit>

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- **Effects**
- Ajax
- Utilities

`$.fx.off`

`show()`

`hide()`

`toggle()`

`slideDown()`

`slideUp()`

`slideToggle()`

`fadeIn()`

`fadeOut()`

`fadeTo()`

`animate()`

`stop()`

Blanket FX control

- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- **Effects**
- Ajax
- Utilities

`$.fx.off`

`show()`

`hide()`

`toggle()`

`slideDown()`

`slideUp()`

`slideToggle()`

`fadeIn()`

`fadeOut()`

`fadeTo()`

`animate()`

`stop()`

**New in
1.3**

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- **Effects**
- Ajax
- Utilities

`$.fx.off`

`show()`

`hide()`

`toggle()`

`slideDown()`

`slideUp()`

`slideToggle()`

`fadeIn()`

`fadeOut()`

`fadeTo()`

`animate()`

`stop()`

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- **Effects**
- Ajax
- Utilities

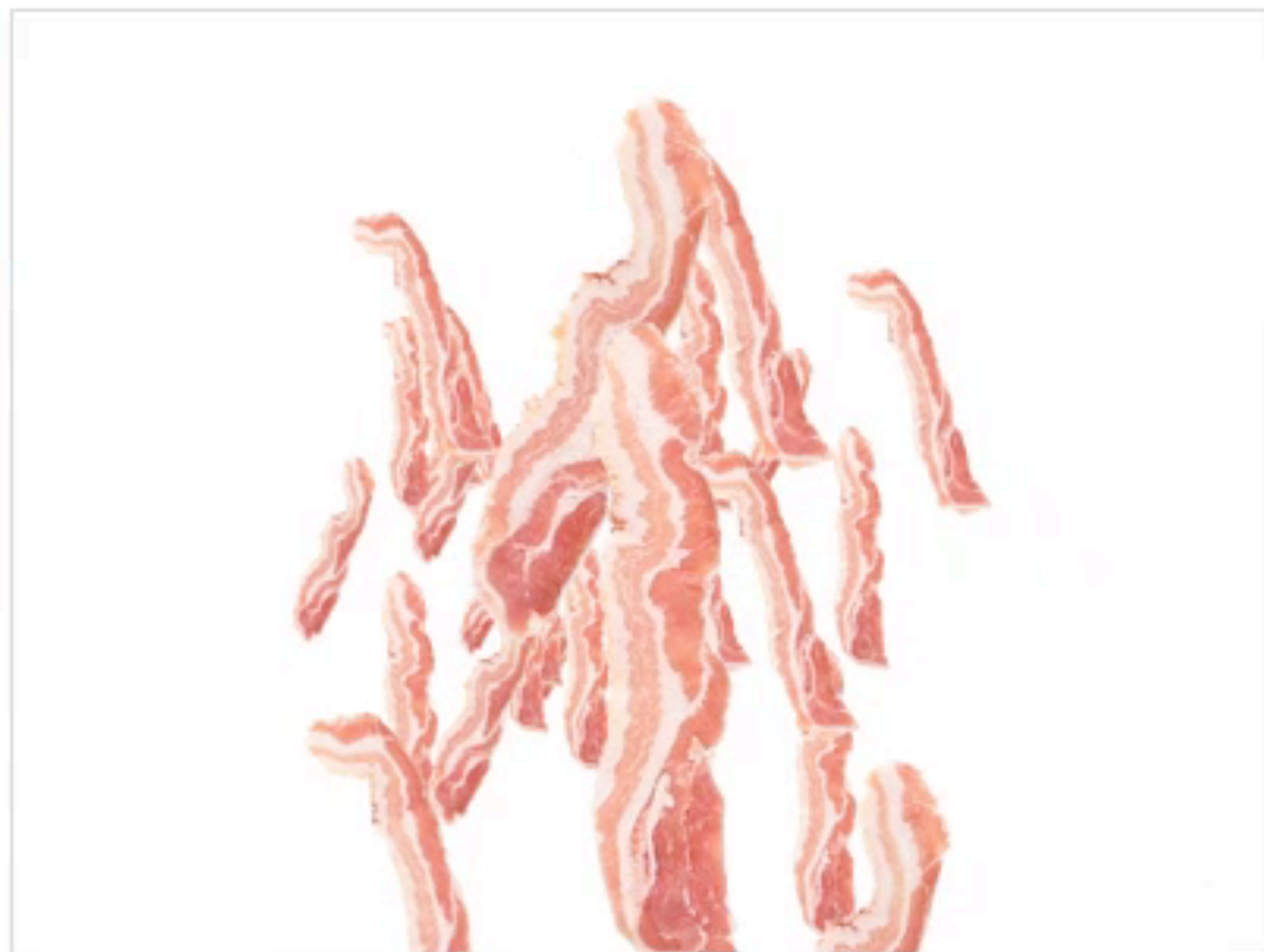
```
<!DOCTYPE html><html><head>
<style>
div {background:#bca; width:100px;
border:1px solid green;}
</style>
</head>
<body>
<div id="block">Hello!</div>
<script src="jquery.js"></script>
<script>

$( '#block' ).animate({
    width: '70%',
    opacity: 0.4,
    marginLeft: '0.6in',
    fontSize: '3em',
    borderWidth: '10px'
}, 1500);

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

<http://jsbin.com/oroto/edit>

`$.animate()` hacking demo – Cloud of Bacon

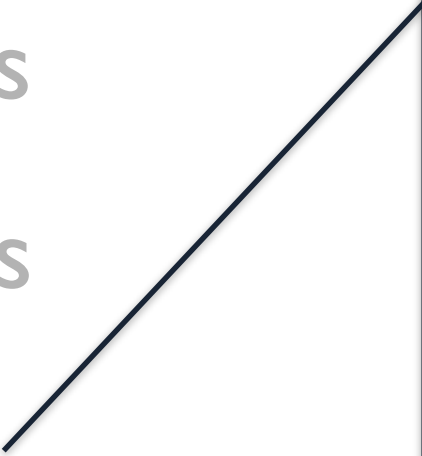


Move camera

Move bacon

For background, see [this blog post](#).

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- **Ajax**
- Utilities



<code>\$.ajax()</code>	<code>ajaxComplete()</code>
<code>\$.get()</code>	<code>ajaxError()</code>
<code>\$.post()</code>	<code>ajaxSend()</code>
<code>\$.getJSON()</code>	<code>ajaxStart()</code>
<code>\$.getScript()</code>	<code>ajaxStop()</code>
<code>\$.ajaxSetup()</code>	<code>ajaxSuccess()</code>
 <code>load()</code>	
 <code>serialize()</code>	
<code>serializeArray()</code>	

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- **Ajax**
- Utilities

`$.ajax()`

`$.get()`

`$.post()`

`$.getJSON()`

`$.getScript()`

`$.ajaxSetup()`

`ajaxCompete()`

`ajaxError()`

`ajaxSend()`

`ajaxStart()`

`ajaxStop()`

`ajaxSuccess()`

`load()`

`serialize()`

`serializeArray()`

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html><html><body>
<script src="jquery.js">
</script>
<script>

$.getJSON( "http://twitter.com/
statuses/user_timeline.json?
screen_name=rem&callback=?",
function(data) {
    $.each(data, function(i, tweet) {
        $( '<p/>' ).html
(tweet.text).appendTo( 'body' );
        if ( i == 30 ) return false;
    });
});
</script></body></html>
```

<http://jsbin.com/acisi/edit>

Plugins

What's a plugin?

A plugin is nothing more than a custom jQuery method created to extend the functionality of the jQuery object

```
$ ( ' u1 ' ) . myPlugin ( )
```


Plugin design in 6 steps

Step 1:

***create a private
scope for \$ alias***

```
<!DOCTYPE html><html><body>  
<script src="jquery.js"></script>  
<script>  
  (function ($) {  
  
  })(jQuery);  
</script></body></html>
```

Step 2:

***attach plugin to fn
alias (aka prototype)***

```
<!DOCTYPE html><html><body>
<script src="jquery.js"></script>
<script>
(function ($) {
$.fn.notHate = function () {
    $(this).text(
        $(this).text().replace(/hate/g, 'love')
    );
}; // end of plugin
})(jQuery);
</script></body></html>
```

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<script src="jquery.js"></script>
<script>
(function ($) {
$.fn.notHate = function () {
    $(this).text(
        $(this).text().replace(/hate/g, 'love')
    );
}; // end of plugin
})(jQuery);

$('p').notHate();
</script></body></html>
```

Step 3:

add implicit iteration

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script src="jquery.js"></script>
<script>
(function ($) {
$.fn.notHate = function () {
    this.each(function () {
        $(this).text(
            $(this).text().replace(/hate/g, 'love')
        );
    });
}; // end of plugin
})(jQuery);

$('p').notHate();
</script></body></html>
```


Step 4:

enable chaining

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script src="jquery.js"></script>
<script>
(function ($) {
$.fn.notHate = function () {
    return this.each(function () {
        $(this).text(
            $(this).text().replace(/hate/g, 'love')
        );
    });
}; // end of plugin
})(jQuery);

$('p').notHate().hide();
</script></body></html>
```

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script src="jquery.js"></script>
```

this == jQuery

```
$.fn.notHate = function () {
    return this.each(function () {
        $(this).text(
            $(this).text().replace(/hate/g, 'love')
        );
    });
}; // end of plugin
})(jQuery);

$('p').notHate().hide();
</script></body></html>
```

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script src="jquery.js"></script>
```

this == jQuery

```
$.fn.notHate = function () {
    return this.each(function () {
        $(this).text(
            $(this).text().replace(/hate/g, 'love')
        );
    });
}; // e
})(jQuery);
```

this == DOM element

```
$('p').notHate().hide();
</script></body></html>
```

Step 5:

add default options

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script src="jquery.js"></script>
<script>
(function ($) {
$.fn.notHate = function () {
    return this.each(function () {
        $(this).text(
            $(this).text().replace(/hate/g,
                ➡ $.fn.notHate.defaults.text)
        );
    });
}; // end of plugin
$.fn.notHate.defaults = {text: 'love'};
})(jQuery);

$('p').notHate();
</script></body></html>
```

Step 6:

add custom options

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script src="jquery.js"></script>
<script>
(function ($) {
$.fn.notHate = function () {
    return this.each(function () {
        $(this).text(
            $(this).text().replace(/hate/g,
                ➡ $.fn.notHate.defaults.text)
        );
    });
}; // end of plugin
$.fn.notHate.defaults = {text: 'love'};
})(jQuery);

$('p').notHate({text: 'love-love-love'});
</script></body></html>
```



```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script src="jquery.js"></script>
<script>
(function ($) {
$.fn.notHate = function (options) {
    return this.each(function () {
        $(this).text(
            $(this).text().replace(/hate/g,
                ➡ $.fn.notHate.defaults.text)
        );
    });
}; // end of plugin
$.fn.notHate.defaults = {text: 'love'};
})(jQuery);

$('p').notHate({text: 'love-love-love'});
</script></body></html>
```

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script src="jquery.js"></script>
<script>
(function ($) {
$.fn.notHate = function (options) {
    var settings = $.extend({},
        ➡ $.fn.notHate.defaults, options);

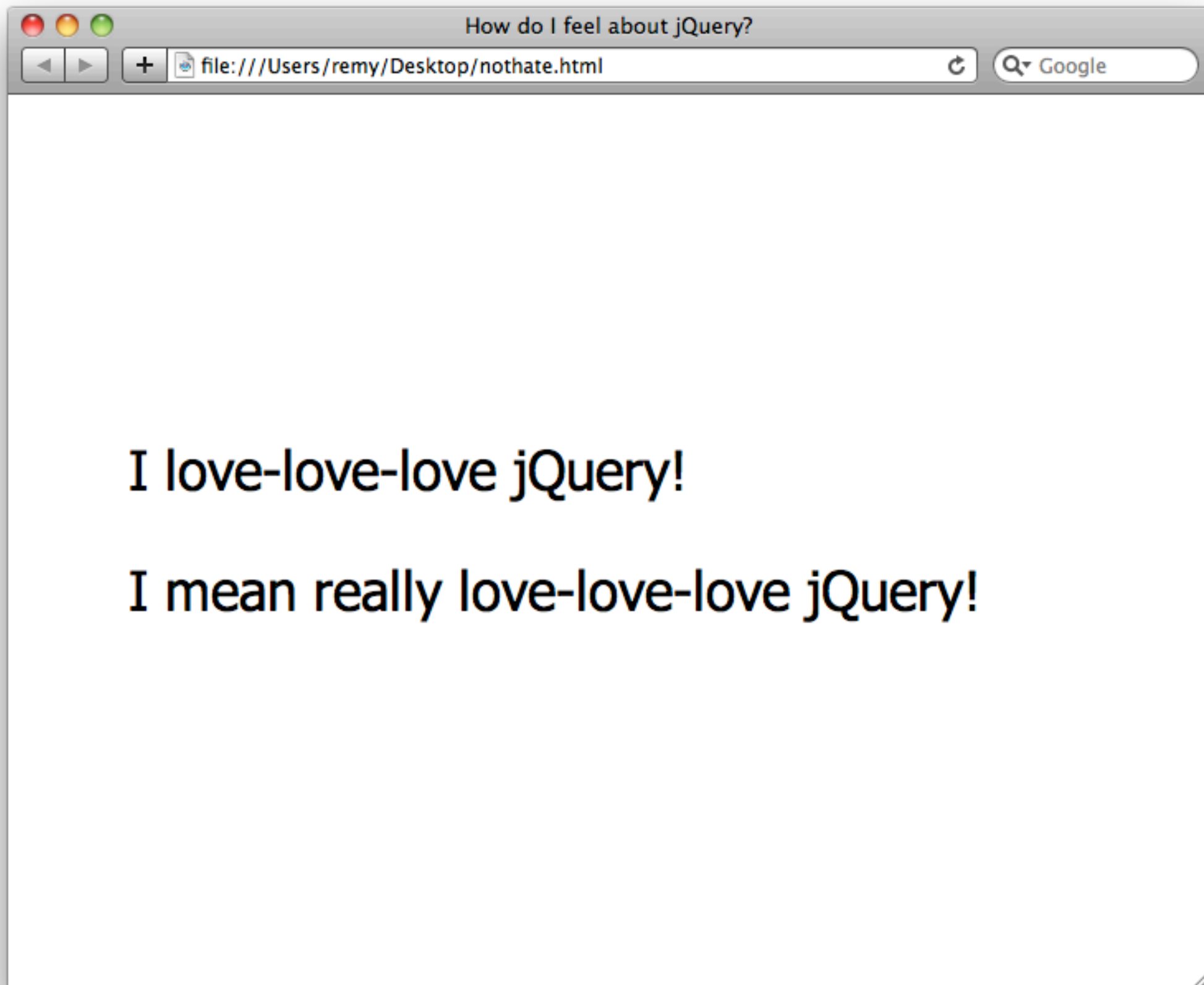
    return this.each(function () {
        $(this).text(
            $(this).text().replace(/hate/g,
                ➡ $.fn.notHate.defaults.text)
        );
    });
}; // end of plugin
$.fn.notHate.defaults = {text: 'love'};
})(jQuery);

$('p').notHate({text: 'love-love-love'});
</script></body></html>
```

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script src="jquery.js"></script>
<script>
(function ($) {
$.fn.notHate = function (options) {
    var settings = $.extend({},
        ➡ $.fn.notHate.defaults, options);

    return this.each(function () {
        $(this).text(
            $(this).text().replace(/hate/g,
                ➡ settings.text)
        );
    });
}; // end of plugin
$.fn.notHate.defaults = {text: 'love'};
})(jQuery);

$('p').notHate({text: 'love-love-love'});
</script></body></html>
```



News

- Four conferences next year:
London, Boston, San Francisco and online
- New plugin site
- jQuery Forum (moving from Google Groups)
- Mobile jQuery

Demo!

Remy Sharp [@rem](#)

jQuery team member

Co-author of O'Reilly

jQuery Cookbook

(due November 20th)

jqueryfordesigners.com

remysharp.com

Remy Sharp [@rem](#)

jQuery team member

Co-author of O'Reilly
jQuery Cookbook
(due November 20th)

jqueryfordesigners.com

remysharp.com



Questions