

# jQuery & Web Services

Logan Allred

Sponsored by:



NEW ENGLAND HISTORIC  
GENEALOGICAL SOCIETY



rootstech 

# What is jQuery?

- Lightweight cross-browser JavaScript library designed to simplify the process of dynamically changing web pages



# What can it do?

- Access part of a page
- Modify the appearance of a page
- Alter the content of a page
- Respond to a users interaction with a page
- Add animation to a page
- Retrieve information from a server without refreshing a page (Ajax)
- Simplify common JavaScript tasks

# Other JQuery Benefits

- Leverages knowledge of CSS
- Extensible through plugins
- Abstract away most browser quirks
- Always work with sets – implicit iteration
- Allow multiple actions in one line – chaining
- Small, open source, popular

# DEMOS

# Getting jQuery

- Download from [jquery.com](http://jquery.com)

```
<head>
  <title>Hello jQuery world!</title>
  <script type='text/javascript' src='jquery-1.4.2-min.js'></script>
  <script type='text/javascript' src='myscript.js'></script>
</head>
```

- Use a CDN (Google, Microsoft, ...)

<http://code.google.com/apis/libraries/devguide.html>

```
<head>
  <title>Hello jQuery world!</title>
  <script type='text/javascript'
    src='http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/
jquery.min.js'></script>
  <script type='text/javascript' src='myscript.js'></script>
</head>
```

# The jQuery Function

- `jQuery( )` function is the main gateway to most functionality
- Returns a `jQuery` object, which is used to perform further operations
- By default, aliased to `$( )`

```
// both statements below are equivalent  
jQuery('#intro').hide();  
$('#intro').hide();
```

# jQuery Object

- Wraps a set of DOM elements, either existing on the page, or newly created elements

```
// returns jQuery object containing the set  
// of elements on the page with a class named  
// 'intro' – may be empty if none exist  
$('.intro')
```

```
// returns jQuery object containing newly  
// created elements matching the html given  
$('<p>Some <em>new</em> content</p>')
```



# Selectors

Selector	CSS	jQuery	Description
tag	p div	\$('p') \$('div')	Selects elements with the given tag name
class	.some-class .highlight	\$('.some-class') \$('.highlight')	Selects elements with the given class name
id	#my-id #main-content	\$('#my-id') \$('#main-content')	Selects elements with the given id
hierarchy	#my-id > span table a	\$('#my-id > span') \$('table a')	Selects descendants or children matching the expression
attribute	a[href^=mailto:]	\$('a[href^=mailto:]')	Selects 'a' elements whose href attribute begins with "mailto:"
multiple	h1,h2,h3	\$(h1,h2,h3')	Selects all h1,h2, & h3 elements
filter/ pseudo	tr:even n/a n/a	\$('tr:even') \$('p:eq(3)') \$(':radio:checked')	Selects even rows of tables Selects 4 <sup>th</sup> paragraph element Selects checked radio buttons

# Ready?

- You generally want to wait until the document is ready, meaning that the DOM has been loaded

```
$(document).ready(function() {  
    // DOM is ready to select and manipulate  
    $('#myelement').addClass('ready');  
    // ...  
});
```

# Common jQuery Functions

- `addClass()`, `removeClass()`, `css()`
- `length`, `get()`, `[]`, `is()`, `has()`
- `toggle()`, `toggleClass()`, `hasClass()`
- `html()` `text()`

# Chaining

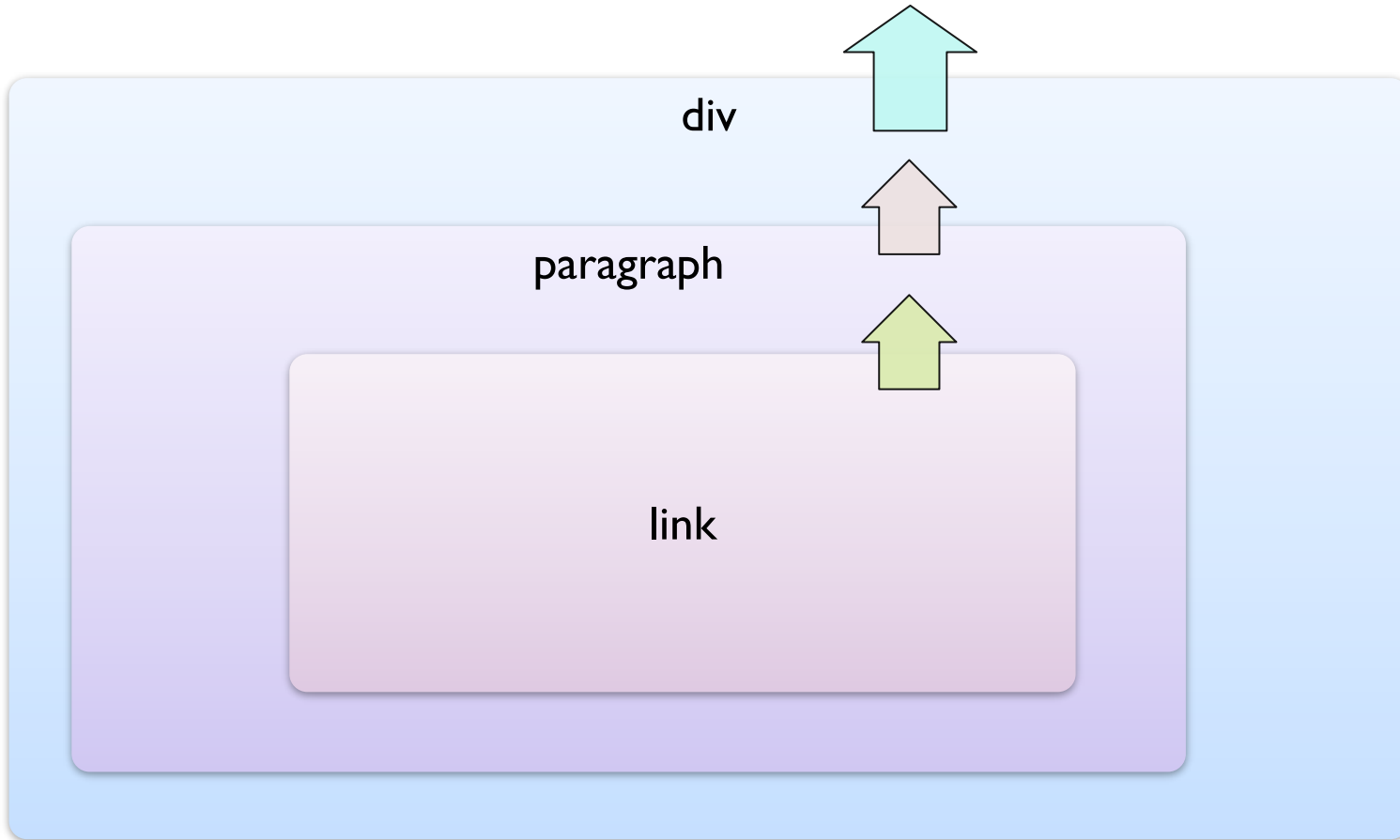
- Most jQuery functions return the same object, so you can continue to work with that object and chain several actions together

```
// chain several actions together on the same object
$('.hidden').removeClass('hidden').addClass('updated').show();
// formatting can help readability of long chains
$('#message')
    .addClass('warning')
    .toggle().text('Could not complete operation')
    .parent().toggleClass('highlight').show();
```

# Events

- Registering Events
  - `bind()` and `unbind()`
- Shorthand methods
  - `click()`, `toggle()`, `hover()`

# Event Bubbling



# Web Services

- Single Origin Policy/Cross Domain Request
- OAuth (for FamilySearch)
  - <http://code.google.com/p/oauth-php/>
- Formats
  - XML
  - JSON
  - HTML

# Cross Domain Requests

- Server side proxy: PHP, Ruby, Python, Apache
  - <https://github.com/cowboy/php-simple-proxy>
- CORS (requires modern browser)
- JSONP
- CSSHttpRequest (like JSONP but with CSS)
- flXHR (uses Flash cross-domain support)



# Ajax

- `load()`
- `$.get()` `$.post()`
- `$.getJSON()`
- `$.ajax()`

# Q & A

- <https://github.com/redbugz/rootstech2011>

# Thank You.

Sponsored by:



NEW ENGLAND HISTORIC  
GENEALOGICAL SOCIETY



rootstech 