



FEWD

Week 4 • Class 8

Learning  **jQuery**
write less, do more.

Quick Review

- Who remembers what the DOM tree is?
- How do you use an attribute as a selector?
- What is the difference between Javascript and jQuery?
- Do you remember what *method* we used to select the element by its id in Javascript?

What We'll Cover

- jQuery Overview
- Introduction to the Javascript Console
- Adding Interactivity with jQuery

jQuery Basics

Objectives: jQuery Overview

- Understand what jQuery does for us
- Add jQuery to a page and check that it is loaded

What is jQuery?

jQuery is a *cross-browser* JavaScript library designed to simplify the client-side scripting of HTML.

Why jQuery?

- Faster and more intuitive to work with
- Robust opensource community and lots of plugins
- Smooths out those differences between how javascript is implemented in different browsers

What's it Good for?

jQuery is especially good at:

- DOM tree traversal & DOM manipulation
- AJAX requests
- Simple animations
- Event Handling
- Handling Forms



Javascript vs. jQuery

What's the Catch?

- Another HTTP request and download of ~25-40K
- You still need to learn Javascript 😞

How Pages are Rendered

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My Page</title>
  <link rel="stylesheet" href="http://example.com/styles.css">
  <link rel="stylesheet" href="css/styles.css">
</head>
<body>
  ...Page Contents...

  <script src="https://code.jquery.com/jquery-3.3.1.js"></script>
</body>
</html>
```

Using jQuery

Loading Scripts

- External Javascript files must be loaded and processed before you use them.
- To do this we use `<script>` tags for Javascript in our HTML
- It's valid to insert script tags **anywhere** on the page.
- But you may only put script tags at the end of the page immediately before the closing `</body>` tag.*

* Meaning in this class, you may only put them there... *Really, don't let me see you put them anywhere else* 🙄

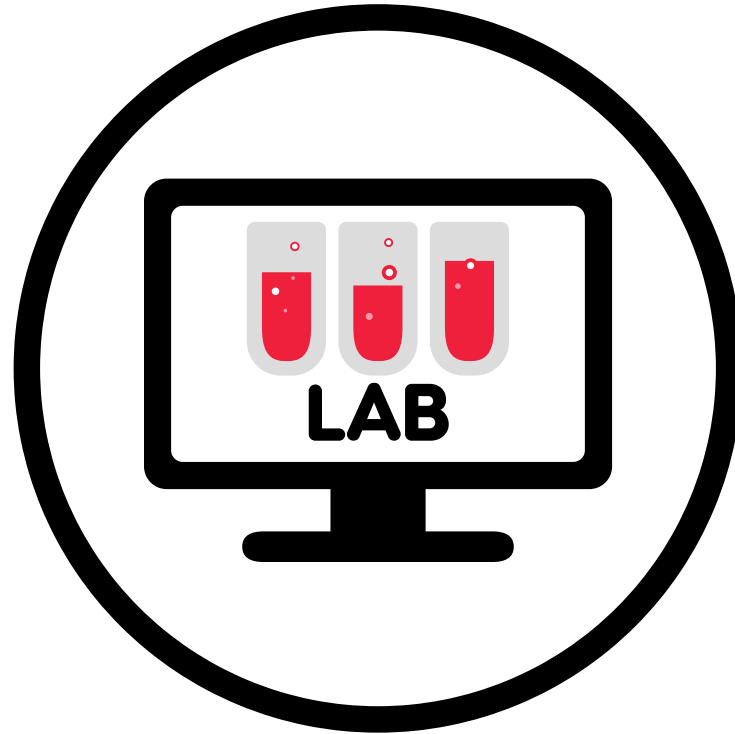
Loading jQuery from a CDN

(Content Delivery Network)

```
<script src="https://code.jquery.com/jquery-3.3.1.min.js"  
integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="<br>  
crossorigin="anonymous"></script>
```

jQuery can also be downloaded and linked with a relative path

But don't do that either 😏



Adding jQuery

The Console is Your Friend

Open the console with any of the following options:

- Right-click in the browser and choose **Inspect**
- ⌘ (command) + ⌥ (alt/option) + I
- **View > Developer > Developer Tools**



Checking for jQuery in the Console

Using the Console

In the console panel, type:

```
> $.fn.jquery
```

If you got back a version number jQuery is loaded!

Syntax

Objectives: Javascript Syntax

- Learn the basic syntax rules for Javascript

Syntax

Like any language, there are formal rules for spelling and grammar. This is called the **syntax** of a language.

CamelCase

Javascript is very much **case-sensitive**! It will nearly always use lowercase or lower camelcase, but there are exceptions.

```
CamelCase //upper camel case  
camelCase //lower camel case
```

Javascript Punctuation

- Statements end with a `;`
- Blocks are denoted with curly braces `{ }`
- Parentheses surround function and method parameters `()`
- String values are surrounded with quotes `' '`

Semicolons

Semicolons in JavaScript are used to separate statements. A statement is a piece of code that tells the computer to do something.

YES

```
var x = 9;  
var fun = function() {...};  
alert("hi");  
do {...} while (...);
```

NO

```
if (...) {...} else {...}  
for (...) {...}  
while (...) {...}  
function (arg) { }
```


Curly Braces

Code that is meant to be run together (known as a ***code block***), are surrounded by curly braces. We also use curly braces to surround object keys and values (but more on that later).

```
if (...) {  
    //run all the code here if true  
} else {  
    //run all the code here if false  
}  
  
function () {  
    //run all the code here if called  
}
```

Parentheses

Parentheses surround the data that we pass to functions and methods. For example, if we have a function or a method that sets the height of an element, we might pass it the height that we want the element to be. Parentheses are also used in math to control order of operations.

```
function setHeight(height, element) {  
  element.style.height = height;  
}
```

```
9 + 1 * 3 // 12  
(9 + 1) * 3 // 30
```

String Values

In Javascript, there are different data types (like numbers, arrays, boolean). String types (or string values) is one that you will work with a lot. Strings are surrounded by quotes. In Javascript, you can use double or single quotes interchangeably.

```
var name = 'Jennifer';
```

Convention is to use single quotes in JS and double in HTML.

Comments in JS

Comments in javascript are denoted as they are in CSS with `//` before a single line comment or `/* */` surrounding the comment

```
// here is a single line comment
```

```
/* this is a multi-line or an  
   inline comment*/
```



Grammar Quiz

jQuery Syntax

jQuery Syntax

jQuery abides by the basic syntax rules for Javascript, but it has it's own syntax to set it apart.

Parts of the jQuery Sentence

The significance of the \$

The dollar sign in jQuery is just a shorter form of writing jQuery.

```
var divs = $('div');           // find all divs
var divs = jQuery('div');      // also find all divs, because
                                // $ is just an alias of jQuery
console.log($ === jQuery);     // === in JS means "strictly equal to"
```

\$ Signals jQuery is Needed

The `$` or `jQuery` lets the Javascript engine know that it needs the jQuery library to run this bit of code.

► **NOTE:** Other code can specify that it wants to use the dollar sign for it's own purposes as an alias. In this case, we can run jQuery in no-conflict mode and just use the written out version of jQuery. We cannot omit it though.

jQuery Selectors

Selectors in jQuery work like in CSS, with one exception: No pseudo-elements.

Selectors are like the nouns referring to things in the Document Object Model.

```
var firstDiv = $('div:first-of-type'); // get the first div only

var firstChildPs = $('p:first-child'); // get all the p that are the
                                         // first child of their parent
var notFirstChildPs = $('p:not(:first-child)');
```

jQuery Methods

Methods take action. They are like the verb in a sentence. Methods often have parameters that allow us to pass data into the method to affect the result.

```
$( 'body' ).addClass( 'special' ) //Adds the class special to the body

$( '.box' ).height( '10vh' ); //Sets the height of the element(s)
                             //with the box class
                             //Gets the height if no value is passed to it
```

jQuery Event Handlers

Events are something that happens outside of your program, such as a click, scroll, swipe, or resize of the browser window.

We can '*listen*' for events and respond to them with an event handler.

```
$( '#goButton' ).click(function(){  
    $( 'body' ).css( 'backgroundColor', 'green' );  
    $( '#goButton' ).hide();  
    $( '#stopButton' ).show();  
});
```

Ready, Set, Go

```
<script>
  $(document).ready(function(){
    //all of your jQuery goes in here
    //you don't repeat this for every piece of code
  });
</script>
```

Effects & CSS Methods

jQuery Effects

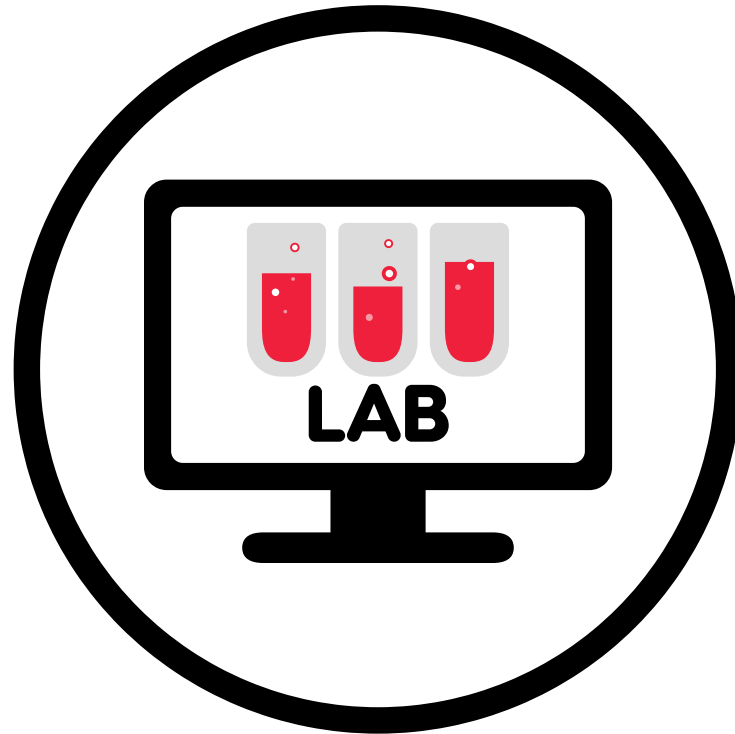
- fadeIn()
- fadeOut()
- fadeToggle()
- fadeTo()
- slideUp() **hides*
- slideDown() **shows*
- slideToggle()
- hide()
- show()

jQuery CSS

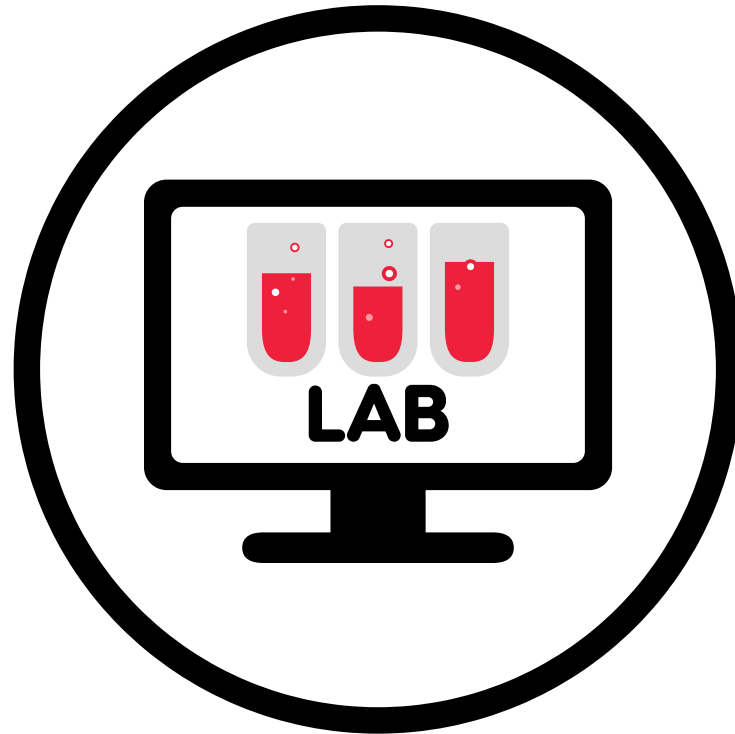
- `css('property', 'value')`
- `addClass()`
- `toggleClass()`
- `height()`
- `width()`



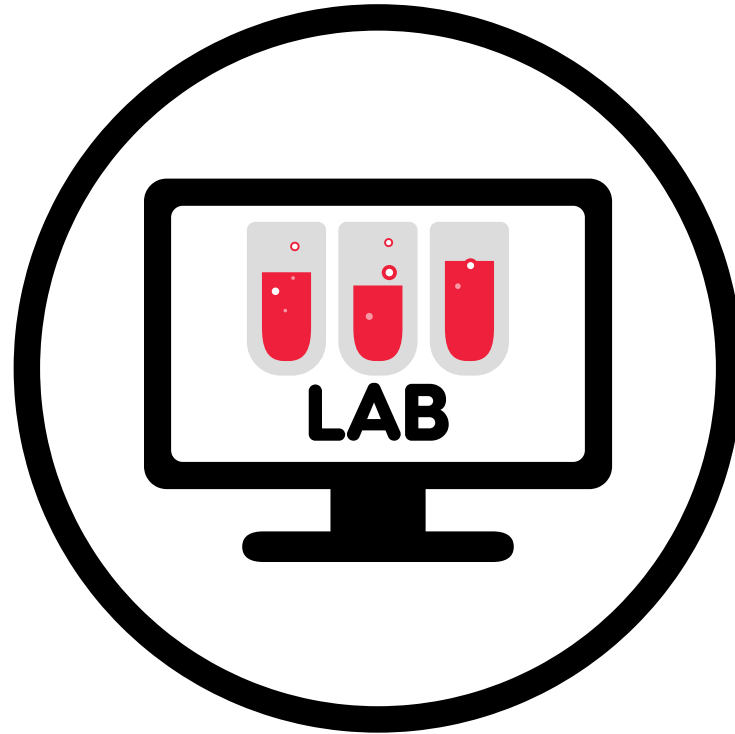
**Practice with Selectors,
Methods, & Events**



FAQs Exercise



Boxes Exercise



Start Your Homework

Go Build Awesome Things!