

FEWD WEEK 5 • CLASS 9:

Intro to jQuery

<https://slides.com/jennifermeade/fewd-5-9/live>



WHAT IS JQUERY

WEB PAGE COMPONENTS



`<h1>Title</h1>`

HTML

`h1 {color: red;}`

CSS

`insertTitle();`

JS

SO, WHAT'S JQUERY THEN?

Under the covers, jQuery
is Javascript

WAIT, JQUERY IS JAVASCRIPT?

Yes, the jQuery library is **written in Javascript**, and we must include it with our webpage. You can think of the jQuery Library as a kind of **glossary for the browser**. All of the *definitions* in the glossary are in Javascript so the browser can make sense of the jQuery (which it doesn't natively understand) in our code.

WHY USE JQUERY?

- More **concise** and **intuitive** way of writing Javascript
- A great way to **start learning Javascript** and programming
- Became popular because it smoothed out the differences between how browsers interpreted Javascript

GETTING STARTED

OBJECTIVES

- Adding jQuery to your page
- jQuery Syntax
- Using jQuery Methods
- Events

ADDING JQUERY TO YOUR PAGE

MEET THE SCRIPT TAG

- Javascript can be embedded or linked in HTML
- The `<script>` tag is used to **either** link an external file **or** wrap embedded code
- The **best** place for all of your script tags is in the body just before the closing `</body>` tag

```
<!-- Place scripts before the closing body tag -->
<script>
    /* Your code here */
</script>
</body>
</html>
```

TRY IT!

```
<!DOCTYPE html>
<html>
  <head>

    <meta charset="UTF-8">
    <title>Hello World!</title>

  </head>
  <body>

    <!-- Where your content ends -->
    <script>
      alert('Hello World');
    </script>
  </body>
</html>
```



**CONGRATS! YOU
JUST WROTE
JAVASCRIPT CODE**

Let's break down what you did...

JAVASCRIPT STATEMENTS

- Statements are the basic building blocks
- A semi-colon ; ends a statement, but line-breaks work **most** of the time as well

```
alert('Hello World');  
alert('Alerts can be annoying');
```



Javascript has some ~~confusing~~ **STUPID** rules about when to use semicolons. As a general rule, use them at the end of a line except if the line ends with a closing curly brace `}`.

COMMENTS

- Single line or inline comments use `//`
- Multi-line comments use `/* comment */`

```
alert('Hello World'); //inline comment
/*
  MULTILINE COMMENT
  This one spans multiple lines
*/
```



Toggle comments with shortcut keys in most editors:

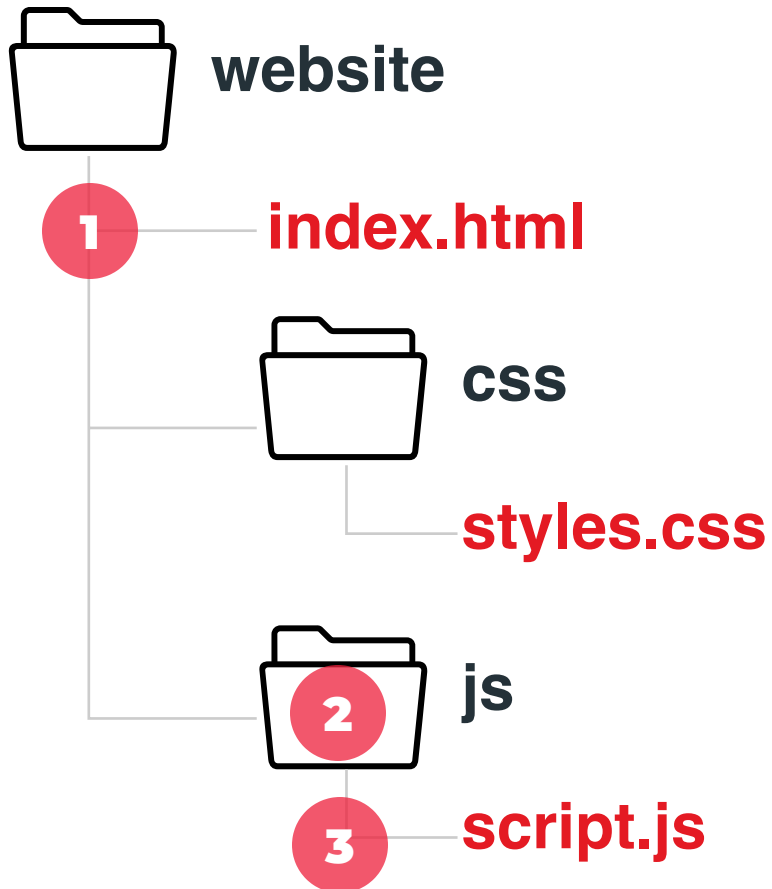
MAC: `⌘` Command + `/` Forward-slash

WIN: `CTRL` Control + `/` Forward-slash

EMBEDDED VS. EXTERNAL CODE

- Just like **embedding** styles in HTML using the style tags makes your code more difficult to maintain, writing your code inside the script tags **isn't optimal**.
- Ideally, you'll want to create one or more separate files and **link your scripts** so you can reuse them on multiple pages in your site.

LINKING FILES RELATIVELY



index.html → script.js

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

- 1** Start in the folder where the current file is
- 2** Go into the folder called **js**
- 3** Get the file called **script.js**

GETTING JQUERY

1. Go to **code.jquery.com**
2. Click on the link for the **minified** version.
3. **Copy** the script tag from the modal window.
4. **Paste** the tag in your page at the bottom of the page, before the closing `</body>` tag and any other script tags.

TRY IT!



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Is jQuery Loaded?</title>
  </head>
  <body>

    <!-- Where your content ends -->
    <script src="https://code.jquery.com/jquery-3.3.1.min.js"
    integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
    crossorigin="anonymous"></script>
    <script>
      alert('jQuery is loaded: ' + $ === jQuery);
    </script>
  </body>
</html>
```

WHAT'S THAT?

```
alert('jQuery is loaded: ' + $===jQuery)
```

- This bit of code uses **concatenation**. That's when we take strings (bits of text) and combine them together.
- The second part checks if jQuery is loaded. If it is, it returns **true**, if it isn't it would return **false**.
- The entire result is then displayed inside the alert!

BEFORE WE MOVE ON...

```
// The jQuery script tag
```

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

- What's up with the crazy long **integrity** attribute?
- And, what does **crossorigin** refer to?

JQUERY SYNTAX

JQUERY SYNTAX

```
$ ( 'selector' ) .method ( ) ;
```

1

2

3

- 1 The dollar sign is actually shorthand for **jQuery**
- 2 The selector targets the things in the DOM we want to operate on. Selectors are identical to **CSS** selectors (note that **they are always in quotes**)
- 3 The **method** is the operation we want to perform

TRY IT!



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello World!</title>
  </head>
  <body>
    <h1></h1>
    <!-- Where your content ends -->
    <script src="https://code.jquery.com/jquery-3.3.1.min.js"
    integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
    crossorigin="anonymous"></script>
    <script>
      $('h1').text('Hello World');
    </script>
  </body>
</html>
```

ADD AN H1 ELEMENT

USING JQUERY METHODS

JQUERY TEXT METHOD

```
//replaces the contents of every h2 element  
$('h2').text("Here's some new text");  
  
//replaces the contents of every element with a class of next  
$('.next').text('Go To Next');  
  
//replaces the contents of the element with an id of title  
$('#title').text('New Title');
```

- The text method **replaces all of the contents** of each element that matches the selector with the text inside the method parentheses.
- Notice that the text value is surrounded in quotes

WHAT WILL HAPPEN?



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Text Method</title>
  </head>
  <body>
    <h1>My Title</h1>
    <p>Paragraph 1 content.</p>
    <p>Paragraph 2 content.</p>
    <!-- Where your content ends -->
    <script src="https://code.jquery.com/jquery-3.3.1.min.js"
    integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
    crossorigin="anonymous"></script>
    <script>
      $('h1').text('This is an h1');
      $('p').text('This is a paragraph');
    </script>
  </body>
</html>
```

JQUERY HTML METHOD

```
//replaces the contents of every p element
```

```
$( 'p' ).html( "Here's some <strong>new</strong> text." );
```

- The html method **replaces all of the contents** of the elements that match the selector with the ***string passed to the method.***
- Unlike the text method, it **can include html** tags.
- Note that the value is always surrounded in quotes

TRY IT!



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>HTML Method</title>
  </head>
  <body>
    <h1>My Title</h1>
    <p>Paragraph 1 content.</p>
    <p>Paragraph 2 content.</p>
    <!-- Where your content ends -->
    <script src="https://code.jquery.com/jquery-3.3.1.min.js"
    integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
    crossorigin="anonymous"></script>
    <script>
      $('h1').html('This is an <em>h1</em> element');
      $('p').html('This is a <em>paragraph</em>');
    </script>
  </body>
</html>
```

JQUERY & DOM MANIPULATION

- jQuery is great at manipulating the DOM
- The `.text()` and `.html()` are examples of DOM manipulation.
- Let's look at some others...

<https://api.jquery.com/>

JQUERY CSS METHOD

```
//adds an inline style to every p element  
$('p').css('background-color', 'deeppink');
```

- The css method will add an inline style to every element that matches the selector.
- This method takes **2** values separated by a **comma**.
- Note that **each** value is always surrounded in quotes

TRY IT!



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>CSS Method</title>
  </head>
  <body>
    <p>Paragraph 1 content.</p>
    <p>Paragraph 2 content.</p>

    <!-- Where your content ends -->
    <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
    <script>
      $('p').html('This is a <em>paragraph</em>');
      $('p:first-of-type').css('color', 'red');
    </script>
  </body>
</html>
```

JQUERY ADDCLASS METHOD

```
//adds a class to every p element  
$('p').addClass('pink');
```

- The `.addClass()` method adds a class, **if it doesn't already exist**.
- This method takes the name of the class
- Note that the class name doesn't have a dot before it!
- As usual, the class name is always in quotes

JQUERY REMOVECLASS METHOD

```
//adds a class to every p element  
$('p').removeClass('pink');
```

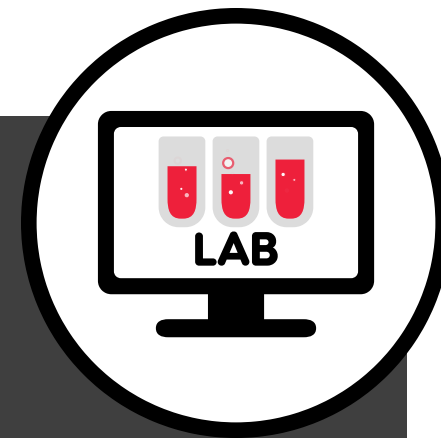
- The `.removeClass()` method removes a class, **if it already exists**.
- This method takes the name of the class
- Don't forget that the class name doesn't have a dot before it!
- As usual, the class name is always in quotes

JQUERY TOGGLECLASS METHOD

```
//adds a class to every p element  
$('p').toggleClass('pink');
```

- The `.toggleClass()` method adds a class to each matched element if it **doesn't already exist**, and removes the class if it **does exist**.
- This method takes the name of the class
- Don't forget that the class name doesn't have a dot before it!
- Of course, the class name is always in quotes

TRY ON YOUR OWN...



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Swap Classes</title>
    <style>
      .red { color: red; }
    </style>
  </head>
  <body>
    <p>Give me the red class!</p>
    <p class="red">Remove the red class from me!</p>

    <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
    <script>
      // Add the red class to the first p and remove it from the second p
    </script>
  </body>
</html>
```

SO WHAT?

jQuery has an easy syntax that allows us to manipulate the DOM, but so far everything we've done could be done without Javascript at all, just by manually changing our html or css.

The real power of JS however is in its ability to **listen for and respond to what the user is doing**. Let's find out how...

LISTENING FOR EVENTS

WHAT ARE EVENTS?

- Events are things that happen when a user interacts with our page
- Some events are browser events like **scroll** and **resize**
- Some events are keyboard events like **keydown** and **keyup**
- Some events are mouse events like **click**

LISTENING FOR EVENTS

- With Javascript, we can listen for when these events happen and then respond to them
- To respond to events in Javascript we use an **event handler**

JQUERY EVENT HANDLER

```
1      2      3  
$( 'selector' ).eventmethod(function() {  
    //code to run when the event is observed  
});
```

- 1 The selector identifies the element we're listening for the event to happen on
- 2 The event method is the specific type of event we're listening for (such as **click**)
- 3 The function part is called an **anonymous function** because it has no name.

ANONYMOUS FUNCTION

```
$( 'p' ).text( 'substitute content' );
```

```
$( 'p' ).click( function(){} );
```



Notice that the only difference between the methods we used earlier and the event handler is the function given to the method.

FUNCTION SYNTAX

```
$('h1').click( function(){} );
```

- The function part is easy to mess up, so here's an easy way to get it right
- Say it out loud as we type it:
function - **open paren** - **close paren** - **squiggly** - **squiggly**
- Once you've typed your function on one line, place the cursor inside the curly braces and hit return a couple of times so you have room to type the code that'll run when the event is observed.

CODEPEN

HTML

CSS

Result

FORK ON
CODEPEN

1 ▾ <h1>CODEPEN EXAMPLE</h1>

2 ▾ <h3>Use the link below to visit this example</h3>

LIVE

CODEPEN
EXAMPLE

Use the link below to visit this
example

<https://codepen.io/jme11/pen/mQJeVv>

ONE MORE TIME

```
$('h1').click(function() {  });
```

function

open paren

close paren

squiggly

squiggly

WHAT WILL HAPPEN?



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello World!</title>
  </head>
  <body>
    <h1>My Title</h1>
    <!-- Where your content ends -->
    <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
    <script>
      $('h1').click(function(){

        $('h1').css('color', 'red');

      });
    </script>
  </body>
</html>
```

JQUERY TRAFFIC LIGHT

HTML

CSS

Result

FORK ON
CODEPEN

1 ▾ <h1>CODEPEN EXAMPLE</h1>

2 ▾ <h3>Use the link below to visit this example</h3>

LIVE

CODEPEN
EXAMPLE

Use the link below to visit this
example

<https://codepen.io/jme11/pen/KeLOgP/>

TRAFFIC LIGHT SOLUTION



```
$('#stop-btn').click(function(){  
  
    $('#traffic-light > div').removeClass('red yellow green'); //BONUS  
    $('#stop-light').addClass('red');  
  
});  
  
$('#slow-btn').click(function(){  
  
    $('#traffic-light > div').removeClass('red yellow green'); //BONUS  
    $('#slow-light').addClass('yellow');  
  
});  
  
$('#go-btn').click(function(){  
  
    $('#traffic-light > div').removeClass('red yellow green'); //BONUS  
    $('#go-light').addClass('green');  
  
});
```

BONUS SOLUTION EXPLAINED

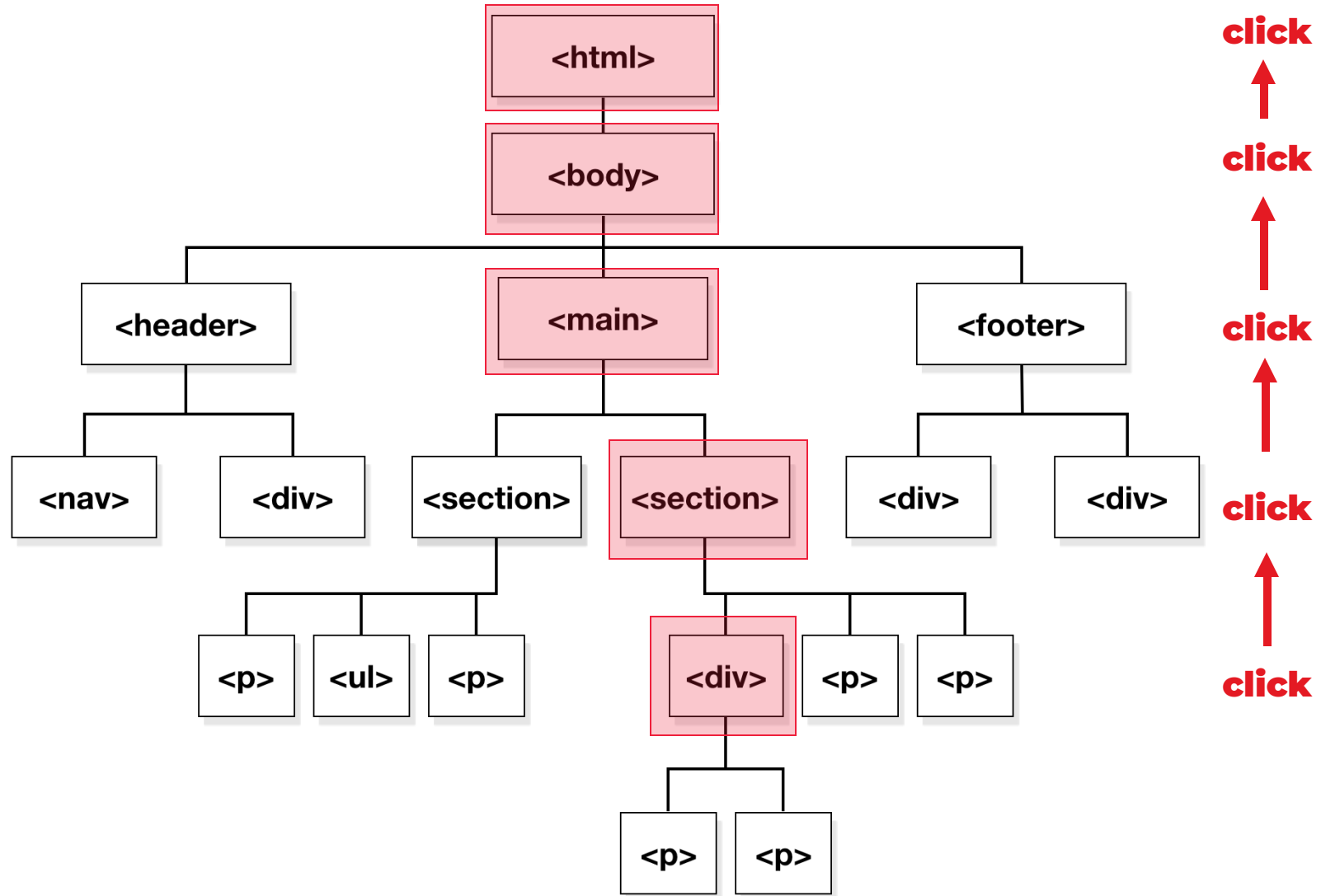
```
/* Remove all of the color classes from all of the lights */  
$('#traffic-light > div').removeClass('red yellow green');
```

- In the HTML, the lights are wrapped in an outer div with with an id of traffic light.
- The selector `#traffic-light>div` targets all of the divs that are children of the outer div, thus, targeting all of the lights.
- We simply need to remove all of the color classes before we apply the new class to the currently selected lamp.
- Remember, `.removeClass()` only removes the class if it exists! And, just like in HTML, we can give it multiple classes separated with spaces.

THE TRICKY PART OF EVENTS

- Events **bubble** meaning they travel from the element all the way up through the up through the element's ancestors.
- So... if you've got another event handler listening somewhere up the chain, it will fire!

BUBBLING EVENTS



STOPPING ROGUE EVENTS

```
$( 'selector' ).eventmethod( function( event ) {  
    event.stopPropagation();  
    // do other stuff...  
});
```

- The stopPropagation method is used to capture and prevent the event from bubbling beyond the current element
- To use it, we need to pass the **event** to our anonymous function
- By convention, we name the event either **event** or **e**

BUBBLING IN ACTION

HTML

CSS

Result

FORK ON
CODEPEN

1 ▾ <h1>CODEPEN EXAMPLE</h1>

2 ▾ <h3>Use the link below to visit this example</h3>

LIVE

CODEPEN
EXAMPLE

Use the link below to visit this
example

<https://codepen.io/jme11/pen/yRaEPL>

THE READY EVENT

```
$(document).ready(function(){  
    // ALL of our jQuery scripts inside here  
  
});
```

- To be safe, it is convention to wrap all of your jQuery inside of a document ready handler.
- The **ready event will fire only once** when the entire DOM has been parsed by the browser.
- Note that in this case the document selector **does not** get quotes.

**GO BUILD
AWESOME THINGS!**