JavaScript/jQuery

Three Parts of a Web Page

- HTML: provides content and structure
- CSS: provides formatting/presentation
- JavaScript/jQuery: provides behavior/interaction

JavaScript

- a programming language that is *not* Java!
- programs run in the web browser
- programs can be just as sophisticated as those written in other programming languages

jQuery

- a collection of functions written in JavaScript
- far fewer lines of code are required than writing in JavaScript alone
- takes care of browser inconsistencies so we don't have to "browser sniff"

Document Object Model (DOM)

- the DOM is a JavaScript version of all of the document's elements
- we can access all HTML elements in a document
- jQuery can add, modify, and delete elements
- this page has more about the DOM

Using JavaScript

- any scripts (i.e. code) we write must be placed within the <script> tag
- the best place to put our code is right before the </body> tag

```
    <script>
        /* Open a small dialog box to greet the
        user when the page is loaded */
        alert("Hello World");
    </script>
</body>
</html>
```

What's a JavaScript Function?

- a *function* is a group of commands that performs some useful task
- some functions return a value they have computed
- we may want to save that value, so we store it in a *variable*

```
/*
Call the function 'sqrt' to compute the square root of 1234.
Store the result in a variable called 'answer' for later use.
*/
var answer = sqrt(1234);

/* Show the answer in a popup dialog box */
alert("The square root of 1234 is " + answer);
```

This page shows the code in action...

Using jQuery

- to use jQuery, we first need to load it from an external file
- instead of keeping a copy locally, we can refer to a Content Delivery Network (CDN) to get it

```
/* Load jQuery */
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jque
    <script>
        /* (Do our stuff in a separate <script> element...) */
    </script>
</body>
```

jQuery Selectors

- we use selectors to select the HTML elements we want to work on
- lucky for us, jQuery uses the same syntax as CSS to select elements
- (please brace yourself for the following cryptic code...)

```
/* find all paragraphs */
$( "p" );

/* find all elements belonging to class 'warning' */
$( ".warning" );

/* find the element with ID 'site-title' */
$( "#site-title" );
```

jQuery Functions

- once we have selected elements, we do something to them with jQuery functions

```
/* hide all paragraphs */
$( "p" ).hide();

/* show all elements in class 'warning' (in case any are hidden) */
$( ".warning" ).show();

/* set the text color of all paragraphs to red */
$( "p" ).css("color", "red");

/* set the content of the level-1 heading to "<em>Hello World</em>" *
$( "h1" ).html("<em>Hello World</em>");
```

Exercise J1: jQuery Selectors

1. download file J-starter.html and save it in your folder as J-selector.html
2. add three level-2 headings, with one belonging to class 'fooclass' and one with the ID 'fooid'
3. with jQuery, do the following
    1. change the color of all level-2 headings to 'green'
    2. add a 1px black solid border to all elements in class 'fooclass'
    3. hide the element with ID 'fooid'

jQuery Events

- many kinds of interaction happen in response to user actions
- these user actions are called *events*
- typical jQuery events include: mouseenter, click, keypress, scroll
- we can tell jQuery to notify us when one of these events occurs
- we do this with functions that are called *event handlers*
- we tell jQuery we'd like to react to events by *registering* an event handler that will be called for those events

jQuery Event Handlers

```
// Define a function that sets an element's background to red
// $(this) specifies the element that received an event
function turnRed() {
    $(this).css("background-color", "red");
}

// Register an event handler for the element with ID 'para1'
// When 'para1' is clicked on, the 'turnRed' handler will be called
$( "#para1" ).click(turnRed);
```

This file shows a complete example.

Exercise J2: jQuery Events

1. download file J-events.html and save it in your folder as J-events1.html
2. replace the turnRed/turnWhite handlers with addBorder/removeBorder handlers that add/remove a 1px green border for mouseenter/mouseleave events (to remove a border set the property to 'none')
3. modify the 'para2' handlers to set the text to all caps on mouseenter and back to normal on mouseleave (the property is 'text-transform' and the values you need are 'uppercase' and 'none')

jQuery Hide/Show

jQuery lets you hide/show elements, or toggle back and forth

```
// Hide all elements in class 'myclass'
$( ".myclass" ).hide();

// Show all paragraph elements
$( "p" ).show();

// For element with ID 'fooid', toggle between hide and show,
// and do it slowly
$( "#fooid" ).toggle("slow");
```

This file shows a complete example.

Here's another nice example in the form of a FAQ.

Exercise J3: jQuery Toggle

1. download file J-starter.html and save it in your folder as J-toggle1.html
2. create an unordered list with at least four items
3. write an event handler that will use jQuery's 'toggle' function to hide/show the list
4. register the event handler with the level-1 heading
5. in other words, clicking on the heading should hide/show the list
6. you may want to look at the two examples from the prior slide for help

jQuery & AJAX

- *AJAX* stands for Asynchronous JavaScript And XML
- (However we don't need to use XML.)
- AJAX lets us load something into a part of our page without reloading the entire page
- it is useful when we need info that only the server has and that JavaScript can't provide in the browser
- this example shows how it works

Workshop Summary/Demo

Here is a summary/demo of the different things we have talked about in the workshop, as well as a few new things to study.

Questions?

If you have questions about the demo or other things we have covered in class, send me an email at mail@jimrudolf.org.