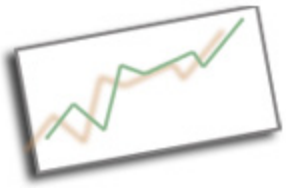

Advanced Online Media
Dr. Cindy Royal
School of Journalism and Mass Communication
Texas State University

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

- Also called a JavaScript Framework.
- Free and open source.
- Makes it easier to create feature-rich JavaScript-enabled Web pages.
- The library functions as an external script that you reference in your document

JQuery has the following types of functionality

- Core Functionality
- Selection and Traversal - Query part
- Manipulation and CSS - editing and changing content
- Events - Simplifies working with DOM events
- Effects - animations, hiding, fading
- AJAX - working with content from remote pages
- User Interface - plugin - slider controls, progress bars, accordions
- Extensibility - add your own functionality

Using JQuery - go to JQuery.com, download the JQuery Library and put it in the folder where you would like to use it. Reference it via the external script method.
<script type="text/javascript" src="jquery.js"></script>

The jQuery library actually comes in 2 forms:

- The uncompressed .js file is easy to read and modify, but it's around 160kb in size (at the time of writing)
- The compressed .js file has all comments, whitespace, and other unnecessary characters removed from the file, squeezing the whole library into a mere 23kb. Although you can't easily read the code, this is the version you'll want to place on your site, as it's much quicker for visitors to download. JQuery lines begin with $ to indicate the use of a function. This is called shorthand notation.

You can use the compressed version for this exercise. Right-click to download. You can rename the file jquery.js. If you don't rename the file, use the proper name of the file in the script tag. This provides all the libraries that the JQuery code needs.

You can also do an external reference to a JQuery library that you can access from a reliable source, like Google. Find the current version (https://developers.google.com/speed/libraries/devguide#jquery).

```
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.0/jquery.min.js"></script>
```

You will link the jquery.js script as well as any other external script you are using. You can use your other scripting either internally or in another external .js script.

```
<script type="text/javascript" src="jquery.js"></script>

<script type="text/javascript" src="script.js"></script>
```

Note that the script tag is not self-closing, even if you don't have anything to put within it.

Basic Syntax

```
$(document).ready(function() {
    $(thingToClick).event(function() {
        $(thingToAffect).effect();
    });
});
```

The $() is a function that turns whatever is inside it into a JQuery object. You always start with $(document).ready(function() { }); to affect the document.

JQuery works with your html and CSS to make interactive changes.

Set up a basic html page with the following:

```
<!DOCTYPE html>
<html>
 <head>
<title>My First JQuery</title>
 </head>
<body>
      <div>Let's do this!</div>
</body>
</html>
```

This is just an html page with head, title, body. The only thing in the body is a div with text in it.

Let's style the div. Add some styles like this into the head. Remember, we usually use an external stylesheet, but since we are working with one file, we can just put the styles in the head.

```
<style>
div {
   height: 50px;
   width: 100px;
   background-color: #FA6900;
   border-radius: 5px;
   text-align: center;
}
</style>
```

Check to see what it looks like in the browser.

Make sure you have the link to your jquery file or the external google api jquery file.

Below that (but still inside the head), include this code.

```
<script>
$(document).ready(function(){
  $('div').hide();

});
</script>
```

This should now hide the div that is in the document. Notice that the word document is not in quotes, but div is. (document) is a jquery object, but a div is a selector, like p, li ul, etc. You can also select by class ('.test ') or id ('#test'). We are only working with for now, but know you can apply these same techniques to any selector.

Let's do a few things with this script. Comment out the line that hides the div.

```
//  $('div').hide();
```

and add :

```
$('div').fadeOut('slow');
```

fadeOut and hide() are effects. There are others show(), fadeIn(), slideUp(), slideDown(). You have to know the arguments of these functions to use them. JQuery.com has full documentation.

You can use JQuery to change classes.

Add to the style section:

```
div.test {
background-color: lightblue;
}
```

Comment out the script that faded the div out. Then add this line to the script:
$('div').addClass('test');

You can use JQuery to change any of the CSS with addClass. There is also a removeClass function.

But the magic really happens when you can make these things interactive. Modify your current script to add the items in bold:

```
<script>
$(document).ready(function(){
$('div').click(function() {
  //('div').hide();
//$('div').fadeOut('slow');
$('div').addClass('test');
});
});
</script>
```

You just surrounded the addClass function with a click event. Now, the addClass function only happens when the div is clicked. This is an example of an event – click(). Other events include dblclick(), mouseenter(), mouseleave(), hover(). There are lots more, but this will get you started with interactivity.

Change the word “click” to “hover” to see what happens.

Change the function from addClass to toggleClass to see what happens.

Comment out the addClass line and uncomment the fadeOut line to see what happens.

You can also embed other events within the main function to provide advanced interactivity. For example, you could have the div fadeOut with one action, then fadeIn with another.

Using this
“this” is a special object that allows you to reference the item to which you are already affecting. For example, instead of continuing to use the (‘div’), you can just use (this) after the event. This helps if you are applying actions to relative objects, so that you are affecting what is currently active rather than something that is absolutely referenced.

Variables
You can also use variables in JQuery to store values or results of functions.

```
var divHTML = $('div').text();
```

Changing HTML
You can use JQuery to change or add content to your html. In our page, the has some text in it. But we can create a variable that receives some other text (either explicitly or from form input or the result of another function) and inserts that into the html.

```
<script>
$(document).ready(function() {
var newText = "jQuery magic in action!";
$('div').html(newText);
});
</script>
```

JQuery Basic Interactive Button with Animation Exercise
1. Just as you did above, set up a basic HTML page with the following. Let's be careful to keep our html, css and script separate. This creates a div with the words Hover Over! in it.

```
<!DOCTYPE html>
<html>
<head>
<title>Button Magic</title>
</head>
<body>
<div><strong>Hover Over!</strong></div>
</body>
</html>
```

Check the file in the browser after each step.

2. Next, let's style the div so it looks more like a button. We usually use an external stylesheet, but since we are just working with one file, it will be fine to add these in the head.

This is just a generic div without any special id or class indicated.

```
<style>
   div {
   height: 60px;
   width: 100px;
   border-radius: 5px;
   background-color: #69D2E7;
   text-align: center;
   color: #FFFFFF;
   font-family: Verdana, Arial, Sans-Serif;
   opacity: 0.5;
}
```

```
</style>
```

Check the file in the browser.

3. Now, make sure you have a link to your JQuery file in the head, either one on your own server or the one on the Google API site.

```
<script type="text/javascript" src="jquery.js"></script>
```

4. Finally, we add the javascript to our document. This can be done in the head or in an external .js file.

```
<script type="text/javascript">
$(document).ready(function() {
   $('div').mouseenter(function () {
      $('div').fadeTo('slow', 1);
    });
});
</script>
```

Check the file in the browser. You should see the button and when you hover over it, it fades in.

5. Now we want to fade it back out. Insert this before the final line with }); in it:

```
$('div').mouseleave(function(){
     $('div').fadeTo('fast', 0.5);
  });
```

Check it in the browser. You should now see it fading in when you hover over and out when you leave.

mouseenter and mouseleave are events.
fadeTo is a JQuery function

You have now created a basic script that you can use to apply a variety of interactive effects to buttons.

Automatic table striping exercise

We will be using JQuery to put alternate colored stripes on a table. This is very helpful in allowing users to read and understand table information and can come in handy for situations when you have a dynamically produced table in which you do not know how many rows it will have. See example at http://cindyroyal.com/advanced/jquery/tablestripe.html.

Be careful as you are typing in your text editor that you are using straight quotes and not any smart quotes copied over from the Word doc.

1. Create a table with several rows and at least two columns – something like bands and cities (put you can use anything you want). Use the <thead> and <tbody> tags (this designates your header row of the table) to delineate those sections. Use <th> for table header cells. Put this in a correctly formed html page (<html>, <head>, <body> etc.)

Something like this within a correctly formed html page:

```
<table>
<thead>
<tr>
    <th>Band</th>
    <th>City</th>
</tr>
</thead>
<tbody>
<tr><td>Old 97's</td><td>Dallas</td></tr>
<tr><td>Quiet Company</td><td>Austin</td></tr>
<tr><td>Buttercup</td><td>San Antonio</td></tr>
<tr><td>Spoon</td><td>Austin</td></tr>
</tbody>
</table>
```

2. Create some basic styles for the page & table. You can do this in an external stylesheet or in the head of the document in a <style> section. Feel free to use any colors, typefaces or sizes you want. When you are done, check your table in a browser to make sure it looks correct.

```
<style>
    body {
        font-family: Arial;
        font-size: 12px;
        color: #000000;
        }

        table {
        border: 1pt solid gray;
        text-align: left;
        width: 400px;
```

```
        }

        th {
        background-color: lightblue;
        color: white;
        }
    </style>
```

3. In the table tag, give it an id of "theList". (in the table tag – id="theList")
4. In the head of the document, insert the reference to the jquery script (use the correct version).

   ```
   <script type="text/javascript" src="jquery.js"></script>
   ```

   You can download the file and put it in the proper folder from Jquery.com.
5. Create the following script. Comments explain each line:

   ```
   <script type="text/javascript">
   $(function() {
   $("#theList tr:even").addClass("stripe1");
   $("#theList tr:odd").addClass("stripe2");
       });
   </script>
   ```
6. Now you just have to create those two styles and add to your style section.

   ```
   .stripe1 {
      background-color:gray;
   }

   .stripe2 {
      background-color:lightgray;
   }
   ```

The JQuery UI
JQuery takes things one step further by offering an easy User Interface that allows you to create simple animations and add effects to your site. It offers a ThemeRoller that allows you to customize and download your own theme, giving you a head start on design. Simply visit jqueryui.com/themeroller/, find a theme you like, start with one and customize it further, or use the Roll Your Own section to create one from scratch. Then choose download theme to get the file structure that you will use for your site.

Look for the script folder, probably named "js" or "script". It will contain the js and ui files you will need to include on your page. You will need to use the script tag to include both jquery and jqueryui files on your page. Something like:

```
<script type="text/javascript" src="js/jquery-1.9.0.js"></script>
<script type="text/javascript" src="js/jquery-ui-1.10.0.custom.min.js"></script>
```

You will also need to reference the theme's css in your document. Something like:

```
<link rel="stylesheet" href="css/sunny/jquery-ui-1.10.0.custom.css"
type="text/css" />
```

Be careful that you are referencing things properly with the proper names and folder locations.

Using the JQuery UI to make an Accordion Widget
There are many things you can do with the JQuery UI, once you know how to structure and use the files. Visit jqueryui.com/ to see the galleries of effects and widgets, including accordion menus, progressbars, sliders, datepickers, etc.

We will be working with the Accordion widget. See the example at http://cindyroyal.com/advanced/jquery/accordion/accordion.html.

First, find a theme that you want to use or customize. Put the accordion code in the body.

```
<div id="accordion">
        <h3><a href="#">Item 1</a></h3>
        <div><p>Item 1 Content</p></div>
        <h3><a href="#">Item 2</a></h3>
        <div><p>Item 2 Content</p></div>
        <h3><a href="#">Item 3</a></h3>
        <div><p>Item 3 Content</p></div>
        <h3><a href="#">Item 4</a></h3>
        <div><p>Item 4 Content</p></div>
</div>
```

You can add styles to the document to override body elements from the theme css and to give the #accordion div a specific width. This will prevent it from expanding the entire length of the window.

```
<style type="text/css">
body {
font-size: 12px;
}

#accordion {
width: 400px;
}
</style>
```

Then, all you have to do is add the following script to the head of the document and the UI does the rest!

```
<script type="text/javascript">
	$(function() {
		$( "#accordion" ).accordion();
	});
</script>
```

Change the content in the accordion menu so you can see something other than Item 1, Item 1 Content, etc.