



Advanced Online Media

Dr. Cindy Royal

Texas State University - San Marcos

School of Journalism and Mass Communication

Javascript, AJAX and JQuery

Javascript - Client side scripting language that can be used to add interactivity to Web interfaces. Written in a text editor, just like html/css and no additional support needed other than a Javascript-enabled browser. Javascript is not Java. It doesn't require a compiler or development environment like other programming languages: Java, PHP, Ruby, Python - just text on pages, run by browsers.

You can use JavaScript to add variables, detect events (like MouseOver), validate data, detect browsers and create cookies (retrieve info from users' computers).

AJAX - Asynchronous Javascript and XML - It uses the standard concepts of HTML, CSS, Javascript and XML in creating dynamic Web applications. The main benefit is that it allows you to communicate with the server without having to refresh the entire Web page. Based on the XMLHttpRequest - added to Microsoft IE originally. Gets data from a server which is used by javascript, but page can be updated without going to server, without changing urls (ie Google Maps).

XML stands for Extensible Markup Language. It is similar to HTML and is used to describe, share, and store data. Tags are not predefined, however. You must define your own tags. RSS feeds are based on XML. Like XHTML, XML elements must have a closing tag, are case sensitive, and must be properly nested.

JQuery -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.

Document Object Model - DOM - a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents. We will be using Javascript and JQuery to manipulate DOM elements. Can download Firefox DOM inspector to look at the DOM of any Web document.

Basic Javascript

- Objects - things in the Document Object Model (DOM) document, window, etc.
- Methods - things you can do like write, open, etc. ends with parentheses.

Syntax object.method();
ex. document.write(); window.open();

End javascript lines with semi-colon ;

- Functions - routines that can be performed in javascript that you define.
- Variables - var cat = "Kiley"; allows you to store things.
- Numbers, strings, booleans (true or false), null- whether variable exists or not.

Can apply math operators (+, -, *, /) or concatenation (for text strings) to variables.

Counters – these all do the same thing, used to count through loops

```
var myCounter = myCounter + 1;  
var myCounter + 1;  
var myCounter++;
```

- Comparisons
Equal ==, not equal !=, greater >, less <
Evaluates to Boolean - either true or false

Usually handled with a conditional
if else

- Event handlers
key presses, mouse clicks, etc.

Simple Javascript Example with html, internal JavaScript and Javascript from an external file
script.html

```
<html>  
<head>  
<title>My first Javascript</title>  
<script type="text/javascript" src="script.js"></script>  
</head>  
<body>  
<h1>Hello World in html</h1>  
  
<h1>  
<script type="text/javascript">  
  document.write("Hello World in Javascript");  
</script>  
</h1>  
  
<div id="helloMessage">  
</div>  
</body>  
</html>
```

script.js

```
window.onload = writeMessage;  
function writeMessage() {  
  document.getElementById("helloMessage").innerHTML = "Hello, world! from an external  
script";  
}
```

The script.js file calls the function writeMessage when the window loads. The function writeMessage consists of the getElementById method, which gets passed an id. That id "helloMessage" is the id given to the third <h1> element. The innerHTML property sets the html code for the element that has that ID.

Comments

It is important to comment your code. If someone else picks up your script, they can easily figure out what you are doing and why.

In html, comments are placed within the following:

```
<!-- Comments go here -->
```

For javascript, and most other programming languages, there are two types of comments.

```
// - for single line comments
```

```
/* */ - for multiline comments
```

For browsers that don't support Javascript, sometimes coders like to include the <noscript> tag.

```
<noscript>Your browser does not support JavaScript!</noscript>
```

Add these one by one to the top of your script.js file

Alerts

```
alert("Welcome to my JavaScript page!");
```

Confirm

```
confirm("Are you sure?");
```

Prompt

In the Hello World example, put the following in your script.js file after window.onload = writeMessage;

```
var ans = prompt("What is your name?", "");
```

Then change your scripts to concatenate ans to the "Hello" string.

```
<script type="text/javascript">
```

```
    document.write("Hello " + ans);
```

```
</script>
```

or in script.js

```
function writeMessage() {
```

```
    document.getElementById("helloMessage").innerHTML = "hello " + ans;
```

```
}
```

or

```
var ans = prompt("Are you sure you want to do that?", "");
```

```
if (ans) {
```

```
    alert("You said " + ans);
```

```
}
```

```
else {
```

```
    alert("You refused to answer");
```

```
}
```

Function that redirects to a javascript page, if the browser supports javascript

```
<html>
<head>
  <title>Welcome to our site</title>
  <script type="text/javascript" src="script.js">
  </script>
</head>
<body>
  <h2 align="center">
    <a href="index.html" id="redirect">Welcome to our site... c'mon in!</a>
  </h2>
</body>
</html>
```

in the script.js file

```
window.onload = initAll;
```

```
function initAll() {
  document.getElementById("redirect").onclick = clickHandler;
}
```

```
function clickHandler() {
  window.location = "jswelcome.html";
  return false;
}
```

Then you'd have a page called jswelcome.html that would redirect if the user clicked. If they didn't have javascript, they'd never see the alert and would go directly to the index.html page.

Simple Rollover

```
<a href="next.html" onmouseover="document.arrow.src='images/arrow_on.gif'"
onmouseout="document.arrow.src='images/arrow_off.gif'"></a>
```

Make sure you have all the images named properly in the correct folder.

Not really used - doesn't preload images, might cause a delay. There is a much more detailed script that handles this issue (like the rollover functions created in Dreamweaver). But gives you the idea of how the event handlers work.

Javascript for Size and Position of New Window – this is helpful if you want to open a popup window in a specific size and location on the screen.

```
<a href="filename.html"onclick="javascript:window.open('filename.html','Listin
gs','width=650,height=600,scrollbars=yes,resizable=yes,screenX=50,screen
Y=250,top=50,left=250'); return false ;" >Link Text</a>
```

You can do lots more with javascript, with logic using functions and loops. But JQuery was developed to make the use of javascript for routine features a lot easier.

JQuery

JQuery Features

- 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 your Web folder. Reference it via the external script method. `<script type="text/javascript" src="script.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 minified .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 shorthand notation

Selectors - returns collection of objects (array) that match selection criteria
can pass tagname, #identifier or .classname

```
.className  
tag.className  
tag#id.className  
$("p");  
$("#list1");  
$("li.a");  
$("ul .b");
```

Filters - further refine results of selection

Use with a dom element to further refine a selection. Then you can apply a modification (like css) to it.

```
<script>$("#tr:first").css("font-style", "italic");</script>
```

```
:first  
:last  
:even  
:odd  
:animated
```

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>.

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

```
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;
}
```

3. In the table tag, give it an id of "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-1.4.4.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() { //run this function when page opens
    $("#theList tr:even").addClass("stripe1"); //for the even rows in theList add class
"stripe1"
    $("#theList tr:odd").addClass("stripe2"); //for the odd rows in theList add class
"stripe2"

    $("#theList tr").hover( //when you hover over rows in theList
      function() { // run this function
        $(this).toggleClass("highlight"); //toggle the "highlight" class
      },
      function() {
        $(this).toggleClass("highlight");
      }
    );
  });
</script>
```

6. Now you just have to create those three styles and add to your style section.

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

```

.stripe2 {
    background-color:lightgray;
}

.highlight {
    background-color: red;
}

```

Image Selector – We made a slideshow with a fading effect in Flash last semester. This time, we'll do the same with JQuery. See the example at <http://cindyroyal.com/advanced/jquery/imageselect.html>.

Exercise

1. Create a page that has two divs. Name one with id="big" and one id="small". Create styles that size and position the divs as follows. This can be as a style in the document, since these styles will only apply to this page. You can add additional body styles to control typeface, size, color, etc.

```
<style type="text/css">
```

```

#big {
    position: relative;
    height: 500px;
    width: 1000px;
}

```

```

#small {
    position: relative;
    height: 60px;
    width: 1000px;
}

```

```
</style>
```

2. In the big div, insert an tag and a <p> tag with a caption for the first image. The should have an id of "photo_large" and the <p> should have an id of "caption1".

```

<div id="big">

<p id="caption1">Fun in the Park</p>
</div>

```

3. In the small div, insert a tag within an <a> tag. Do this with the thumbnail version of each image you have. Make sure you use the proper path for each image. You can use a smaller sized image, or you can simply use the width property to size the original image. Make sure all images are optimized in Photoshop. Use the title attribute to set up the caption. Each image must have the class "gallery."

```

<div id="photoSmall">
<a href="aclphotos/acl1.jpg" title="Fun in the Park"></a>
</div>

```

4. Insert the reference to the script file in the head of the document.

```
<script type="text/javascript" src="jquery-1.4.4.js"></script>
```
5. Insert the following JQuery script in the head of the document. The comments explain each line:

```

<script type="text/javascript">
  $(function() { //run this function when the page loads
    $("a:has(img.gallery)").click(function() { //for a tags that have imgs of the class "gallery" - on
click
      var largePath = $(this).attr("href"); //set a variable that finds the href of the image
      var caption = $(this).attr("title"); //set a variable that finds the caption in the title.

      $("#photo_large").attr({ src: largePath }); //for the #photo_large div, get the src from the
variable
      $("#photo_large").css({opacity: 0.0}).animate({ opacity: 1.0}, 1000 ); //sets up the animation of
the large picture
      $("#caption1").text(caption); //finds the caption from the variable
      return false; //tells browser not to follow link that came with the
image
    });

    $("img.gallery").hover(function() { //for images with the class "gallery" on hover
      $(this).css({opacity: 1.0}).animate({ opacity: 0.7}, 1000 ); //do this animation on mouseover
    }, function() {
      $(this).css({opacity: .7}).animate({ opacity: 1.0}, 1000 ); //do this animation on mouseout
    });
  });
</script>

```

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.

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.

```
<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>
```