**The SCRIPT Tag**

It must be evident to you now that event handlers (such as onClick) are inadequate for holding any substantive quantity of code. Most JavaScript code in a JavaScript-enhanced HTML page is placed within a SCRIPT tag. The main SCRIPT tag for a page usually goes in the HEAD section of your HTML document, although it can also be placed within the BODY section. Between the opening and closing SCRIPT tags, you will write out your lines of JavaScript code.

The SCRIPT tag has ONE attribute, TYPE. The TYPE attribute tells the browser which programming language is being coded inside of the SCRIPT tag. There are many different languages that can be coded within a SCRIPT tag; there are even several different versions of JavaScript itself. For this introductory class, all that you are concerned with is the main version of JavaScript.

Tag: **SCRIPT**  
Attribute: **TYPE**  
Value: **text/javascript**  
Description: The TYPE attribute of the SCRIPT tag sets the programming language type to be coded within the tag.

Example (abbreviated):

<script type="text/javascript"> // Here's a couple of lines of JavaScript code: var george = 1; alert("Howdy"); </script>

Example (in context):

<html> <head> <title>Sample JavaScript Page</title> <script type="text/javascript"> // Here's a couple of lines of JavaScript code: var george = 1; alert("Howdy"); </script> </head> <body> <p>Some content...</p> </body> </html>

In the above example, I have placed my SCRIPT tag in the HEAD of my HTML document. Between the opening and closing SCRIPT tags, I have programmed my JavaScript code, creating a variable called george containing a numerical value of 1, and calling the alert() method which is receiving one argument, the string "Howdy".

Notice how I am able to use DOUBLE-QUOTE marks again to mark my string values? This is possible because I am NOT inside an attribute (as I was with onClick).

Remember in our CSS module, when I talked about CSS comments? Single line comments were marked at the beginning of the line with "//", while multiple line comments were marked "/\* comment... \*/". JavaScript, and all of the C-based languages, use the SAME commenting style! Unlike CSS, however, single line comment syntax IS supported in JavaScript, so you may use it with impunity here (as I did immediately following my opening SCRIPT tag).

There is an older, deprecated attribute of the SCRIPT tag: LANGUAGE. You will still see many JavaScript-enhanced web pages using the LANGUAGE attribute of the SCRIPT tag rather than the TYPE attribute.

Example:

<script language="javascript"> <!-- // Here's a couple of lines of JavaScript code: var george = 1; alert("Howdy") //--> </script>

It is no longer necessary to use the LANGUAGE attribute.

Note: The <!-- right after the script tag and the //--> right before </script> are used to comply with older browsers that do not have javascript enabled. For the most part, they may be omitted. Note that some of my examples may include these statements.

As an HTML page loads, the SCRIPT tag is executed, declaring and initializing variables, and executing methods.

Example Code (abbreviated):

<script type="text/javascript"> var george = 1; alert("Howdy"); alert(george); </script>

Example Code (in context):

<html> <head> <title>Sample JavaScript Page</title> <script type="text/javascript"> var george = 1; alert("Howdy"); alert(george); </script> </head> <body> <p>Some content...</p> </body> </html>

In the above example, I first declare and initialize a variable, george. I then call the alert method, passing the string value "Howdy" as an argument to the alert method. Last, I call the alert method again, passing the contents of the variable, george, to the alert method. Note: even though the contents of george are a NUMBER, not a STRING, JavaScript automatically converts the number INTO a string for use by the alert method. Cool, huh?

Here is the above example [**displayed**](http://fog.ccsf.edu/~srubin/js1.html).

Notice how the alert dialog boxes pop up as the page loads in the above example? As I said, the code inside the SCRIPT tag is executed as the HTML page loads.

Of course, you're not usually going to want to have EVERYTHING in your script tag execute all at once; that would be complete chaos. To reserve some actions, and to call those actions at will rather than just as the page loads, requires the use of a FUNCTION. We'll talk about functions in the next section.

**Final Notes**

An HTML page may, in fact, have several SCRIPT tags on it, both in the HEAD and in the BODY of the document. Because SCRIPT tags are executed as an HTML page loads, you may have a SCRIPT tag in the BODY of a page which lays on-the-fly HTML (built from JavaScript code) into the BODY of the HTML page as the page is parsing. The ramifications of this topic are outside the scope of this introductory lecture, but this sort of JavaScript usage is VERY common.

JavaScript code may ALSO be coded into an external text file, called a ".js" file, which is very much like an external ".css" document, only filled with JavaScript code rather than CSS code. A ".js" file, or even several ".js" files, would be attached to an HTML page using the SCRIPT tag, and many HTML pages could share the same ".js" files. For more information, see *"JavaScript: The Definitive Guide"*, section 12.2.2, pp.215-216. For example:  
  
<script src="abc.js"></script>

Do not put the script tags in your external script files!

Note that some scripts won't validate for XHTML unless you include  
  
/\* <![CDATA[ \*/ right after the script statement and /\*]]>\*/ right before the closing script statement.

<script type="text/javascript">  
**/\* <![CDATA[ \*/**  
*// content of your Javascript goes here*  
**/\*]]>**\*/  
</script>

XHTML is subject to the same syntactical rules as XML. Because of this, an XHTML processor treats the characters < and & as markup, even if they reside inside a <script> block.  
  
see:  
  
<http://javascript.about.com/library/blxhtml.htm>

**I suggest for all your scripts that you use:**

<script type="text/javascript">  
**/\* <![CDATA[ \*/**  
*// content of your Javascript goes here*  
**/\*]]>**\*/  
</script>

Avoid this coding of the script tag because IE will not accept it, even though other browsers will accept it.

<script type=" text/javascript">  
**/\* <![CDATA[ \*/**  
*// content of your Javascript goes here*  
**/\*]]>**\*/  
</script>

Notice the space before 'text'.