**Event Handlers and the javascript Absolute URL**

JavaScript is an event-driven programming language, where JavaScript commands are triggered by events caused by user interaction.

JavaScript uses special EVENT HANDLERS inserted as ATTRIBUTES into ordinary HTML tags to access user events, such as clicking the mouse.

Here is the event handler which deals with mouse clicks:

**onclick**

As long as we are supporting older browsers, most event handlers MUST be placed into tags which can ordinarily accept mouse events, such as the A (anchor) tag, or the INPUT tag in a FORM.

Here is the **onclick event handler** inserted into a hyper-reference:

<a href="#" onclick="var george = 1;">Link Word</a>

The value of the onclick event handler attribute would be a JavaScript command of some sort. More than one command may be placed into an event handler in this fashion simply by appending commands onto the end of the previous command (don't forget your semi-colon end-of-line markers after each command!).

Example:

<a href="#" onclick="var george = 1; var fred = 2; var answer = george + fred;">Link Word</a>

Before we get any deeper into this material, we need to look at the JavaScript Window object, which is built into every JavaScript-enabled web browser.

**The Window Object**

When you open a JavaScript-enabled web browser, such as Firefox or Internet Explorer, certain built-in objects in JavaScript are declared and initialized automatically. One of these built-in objects is an instance of the Window object called window (with a lower-case "w").

Again, you do NOT need to declare the window instance of the Window object; this is done for you automatically when a web browser window is opened. The window instance refers to the current browser window, and has many properties and methods associated with it. A complete list of properties and methods of the Window object are printed in *"JavaScript: The Definitive Guide"*, if you are interested. For the moment, I am only concerned with the alert() method of the Window object.

**The alert() method** of the Window object causes an alert dialog box to open. Here is an example of the alert() method in action:

[Alert Me](javascript:window.alert('Hi%20There,%20Baby!');)

The alert() method REQUIRES one argument, a string, which will become the text displayed in the alert dialog box.

Technically, the alert() method is invoked with the following complete syntax:

window.alert("Hi");

Because everything on an HTML page is displayed within the web browser window, however, it is NOT necessary to state the window instance name directly; window is automatically implied as the parent object for everything on an HTML page. It is perfectly safe to state the alert() method by itself, without using dot syntax to explicitly state the parent/child relationship between it and the window instance. The following syntax for calling the alert() method is perfectly safe and correct:

alert("Hi");

Here is an example of the alert() method placed into an onClick event handler:

<a href="#" onclick="alert('Hi');">Link Word</a>

Although in the previous examples, the string value passed as the argument to the alert() method was enclosed in**DOUBLE-quotes**, you can see that I have marked the string value in the above example using **SINGLE-quote**marks. I believe that we have encountered this problem in previous modules. Attributes may NOT have double-quote marks inside of the double-quote marks beginning and ending the value for the attribute, as this will break your HTML code. Because we NEED quote-marks of some kind to delimit the string value inside the function call operator for the alert method, we MUST use single-quote marks within the bounds of the double-quote marks marking the attribute value. When we write commands within ordinary JavaScript scripts, however, we will be able to go back to using double-quote marks to delineate string values; we'll talk about this later.

Again, the following example would be WRONG:

<a href="#" onclick="alert("Hi");">Link Word</a>

Those extra double-quote marks are EVIL.

Corrected Example:

<a href="#" onclick="alert('Hi');">Link Word</a>

You don't have to use the onclick event handler from within an A (anchor) tag. Instead, you can use the javascript absolute URL as the value for the HREF attribute in order to trigger JavaScript commands.

Example:

<a href="javascript:alert('Hi');">Link Word</a>

In the above example, I have stated the SCHEME for the absolute URL (javascript), followed by the colon character, followed by the "scheme\_specific\_part", the actual JavaScript command desired. **Note:** When using the javascript absolute URL, you must NEVER have any spaces in your JavaScript code; otherwise, the javascript URL will fail to operate.

The following example is WRONG because there are spaces in the URL:

<a href="javascript:var fred = 1; alert(fred);">Link Word</a>

If your JavaScript command requires spaces (as in the above example), you MUST then use the onClick event handler rather than the javascript absolute URL.

As I mentioned earlier, you do NOT have to put your onclick event handlers ONLY in A (anchor) tags; you may place the onclick event handler in ANY clickable tag, such as the INPUT tag in a form.

Example:

<form>  
<input type="button" value="Alert Me" onclick="alert('Hi There');" />   
</form>

[Displayed](http://fog.ccsf.edu/~srubin/buttalert.html)