**Opening a New Window**

In this section, I'm going to write a function which throws up an alert dialog box, then opens a new browser window. To do this, I am going to call two methods of the window instance of the Window object: alert() (which you already know) and open() (which opens a new browser window).

A call to the **open() method** is usually written out like this:

window.open();

Even though the reference to the window object is understood implicitly, the conservative syntax given above looks much more like regular English, and is always used.

The open() method requires three arguments, all strings. The first argument is the URL (absolute or relative) that will appear in the new window. The second argument is the NAME of the new window (which will NOT be visible to the user, but rather identifies the window in JavaScript). The third argument is a series of parameters, describing the appearance of the new window; this third argument may NOT contain any spaces, nor will it contain any additional quotation marks.

Here is an example call to the open() method:

window.open("http://www.yahoo.com/", "marian", "width=500,height=400");

Here is a [**link**](http://fog.ccsf.edu/~srubin/js6nw.html) which demonstrates this code in action. **Note:** you MUST close this new window when you have finished looking at it; don't just hide it behind the current browser window, or some of my later code examples will not operate correctly.

Notice how the new window in the above example had NO status bar, NO location bar, possibly even NO menubar or scrollbars; it wasn't resizable, either. The window was 500 pixels wide and 400 pixels high, and nothing else, because that was all I defined in the third argument of the open() method. To get a new browser window with more features, I have to state ALL of the features that I want, explicitly, in the third argument. Again, each of these features MUST be separated with a comma within the third argument string, and there must be NO SPACES anywhere or it won't work; this is NOT HTML, it just resembles it, so beware.

A complete list of all of the window features available with the window.open() method is printed in *"JavaScript: The Definitive Guide"*. Because these features are not pertinent to my demonstration, I am not going to go into all of them here.

Here is a full-featured window called from JavaScript:

window.open("http://www.yahoo.com/", "marian","top=100,left=300,width=500,height=400,resizable,scrollbars,menubar,status,location,toolbar");

Here is a [**link**](http://fog.ccsf.edu/~srubin/js7nw.html) which demonstrates this code in action. Notice that the left upper corner of the new window is 100 pixels from the top and 300 pixels from the left.

**Note:** again, you MUST close this new window when you have finished looking at it; don't just hide it behind the current browser window, or some of my later code examples will not operate correctly.

Note also that the following will cause a SYNTAX ERROR because you may not separate the string between the last set of quotes, with a carriage return.  
  
window.open("http://www.yahoo.com/", "marian", "top=100,left=300,width=500,height=400,resizable,scrollbars,  
menubar,status,location,toolbar");

Now that I've talked about the window.open() method, let's add that method to our function declaration.

Example:

function doStuff() {

alert("Here is Yahoo!");

window.open("http://www.yahoo.com", "marian", "width=500,height=400");

}

Example (in context):

<html>

<head>

<title>Sample JavaScript</title>

<script type="text/javascript">

/\* <![CDATA[ \*/

function doStuff() {

alert("Here is Yahoo!");

window.open("http://www.yahoo.com", "marian", "width=500,height=400");

}

/\*]]>\*/

</script>

</head>

<body>

<p><a href="#" onClick="doStuff();">Calling doStuff()</a></p>

</body>

</html>

Here is the above example [**displayed**](http://fog.ccsf.edu/~srubin/js8nw.html). Again, PLEASE CLOSE THE NEW WINDOW when you have finished with it!

Obviously, if we hard-coded everything into our functions, we would have to write a different function for each message in the alert dialog box and each URL in the new window. This is an example of someplace where you would want to create a GENERALIZED function by declaring ARGUMENTS for the function (as we discussed previously).

Example:

function doStuff(myMSG, myURL) {

alert(myMSG);

window.open(myURL, "marian", "width=500,height=400");

}

By creating a generalized function as I have in the example above, I can now simply pass arguments to doStuff, passing the string in myMSG to the alert() method, while passing the string in myURL to the window.open() method.

Example (in context):

<html>

<head>

<title>Sample JavaScript</title>

<script type="text/javascript">

/\* <![CDATA[ \*/

function doStuff(myMSG, myURL) {

alert(myMSG);

window.open(myURL, "marian", "width=500,height=400");

}

/\*]]>\*/

</script>

</head>

<body>

<p><a href="#" onClick=

"doStuff('Here is Yahoo', 'http://www.yahoo.com/');">Yahoo Link</a></p>

<p><a href="#" onClick=

"doStuff('Here is Macromedia', 'http://www.macromedia.com/');">

Macromedia Link</a></p>

</body>

</html>

Here is the above example [**displayed**](http://fog.ccsf.edu/~srubin/js9nw.html).

I can call this new generalized function from as many places as I wish. Again, there is NO correlation between what sort of information I pass into an argument variable, and the name of that variable; arguments are always passed into a function from left to right, the first argument going into the first variable, the second argument going into the second variable, etc. The order of the arguments being passed into a function, then, is critical.

Variable names are also important, but mostly to help you, as the programmer, understand what the code is doing. To prevent confusion, the names of the local argument variables should be utilitarian, simple, and clear, which will make your code more comprehensible, as well as easier to read.