Objects, Events and Actions in JavaScript

Overview: We've learned several of the basic tools of JavaScript (variables, assignment, arrays, etc.), but JavaScript is most useful in creating web page applications when you understand how it manipulates the properties of *objects* on a page

1. Javascript Overview

Recall that JavaScript allows the creation of programs that are placed right in the HTML of your Web page and are referred to as *scripts*. The basic functionality of JavaScript is in the manipulation of ***objects*** on a web page – images, text, tables, data in form fields, etc. – while the page is being displayed. JavaScript scripts are typically written to respond to *user events*, such as mouse clicks, mouseovers, loading of a Web page, etc.

Remember once again that the Dreamweaver *behaviors* that you have experimented with all work due to JavaScript. Back when we worked on behaviors, you always defined an *event* that would trigger the behavior, an *action* that would then take place, and you applied the behavior to an *object* on the web page. For example, do you remember the effect of having a welcome message appear when you moused over the Bell Tower in the Furman map? You were defining:

- **Event**: mouse over
- **Action**: show the layer that was previously hidden
- **Object**: the hotspot over the Bell Tower that you moused over

Today we will learn how to work with these things from scratch.

2. Dealing with objects, events and actions

Suppose you wish to display an image on a web page and have that image change as you move your mouse over different images (or *objects*) on the page. As noted above, JavaScript provides the capability of manipulating the various objects on a web page (including images) while the page is being displayed. In this case, we could use it to dynamically change images without changing the HTML page itself. What is needed is a relatively simple **algorithm** and the corresponding JavaScript code.

Our project will make use of some images. You can find them in the class *OUT* folder . Copy them to the *images* folder of your demonstration web site. (Don't copy the whole folder, just the images!)

a. Use the Dreamweaver editor to open a new page within your demonstration web site. (Don't forget to set up a Dreamweaver *site* first.) Name this file: *lastname_sample.html.* Position the cursor somewhere in the page, then click the "HTML Source" button to edit the HTML code.
b. Create a simple 2-row by 4 column table (for formatting purposes). Also make the table rather small 25-30% of browser width.
c. In the top row, we will place four images that act as our "navigation" images – like the navigation bars that we discussed earlier in the semester. Do that now with the images for: spring, summer, fall, and winter.
d. In the second row, we will merge the leftmost two cells and place the picture *dogwood.jpg* in the combined cell.
e. In order to manipulate this image/object (change it's attributes), we must give it a name. Call this image *mainpict* – by selecting the image and typing the name in the property panel: look for the textbox immediately below the words "Image, 15K" in the upper-left.
f. Now we can add an *event handler* to each of the four images using JavaScript, so they will know what action to take when they are moused over. In code view, find the tag for the 'spring' image. Inside of the ***img*** tag, we will add the code below: (do this after the width and height but before the />)

```
onMouseover="mainpict.src='dogwood.jpg'"
```

g. In English: if the mouse moves over this image, set the *mainpict's* source to *dogwood.jpg*. This should seem familiar – it's the exact same principle on which the slideshow script works.

h. *If your files are in a folder, you will need to adjust the filename to include that path. For example: if your image files are in the* ***images*** *folder, it should be:* `mainpict.src='images/dogwood.jpg'`

i. Add similar code (using pictures: *beach*, *wood*, *snow*) for each of the other images and then test.

j. Before we move on, combine the other two cells on the lower row and type in the words "Spring is here". If we wish to change any text using JavaScript we must be able to give it a name. Right now, it is just text without any *object* that we can identify.

k. Highlight the text you entered and select the menu option: ***Insert > Layout Objects > Div Tag***

l. This is a "division" or section of the document and we can give it an identifying name using the properties panel Div Id textbox. Call this text : *maintext*

3. A word or two on *functions*

At times, the onMouseover event handler code becomes too much to keep track of – perhaps we want to change some text *and* the image *and* we want to change the background color too. It may be easier to package this code separately in a different "module". These modules are called ***functions*** and have many benefits, including making programs easier to read. Functions are basically sub-programs that can be "run" by simply using the function name. (You've seen this concept before too – *processNext()* and *processPrevious()* in the slideshow are functions.)

The code below resides in the <**head**> section of an HTML document and defines a function named winterTextChange. Add this code now. [Note: the <script> tag around this snippet of JavaScript code]

```
<script language="javascript">
      var  t;
      function winterTextChange(){
            t = document.getElementById('maintext');
            t.innerHTML='Snowy';
            document.mainpict.src='snow.jpg';
      }
</script>
```

m. When run, this code will find a document element called 'maintext' and then will change it to 'Snowy' [These two lines are the Javascript way to change text – keep this reference handy if you need to do this!] and finally will change the document's *mainpict* image to *snow.jpg*.

n. To run the function, modify the 'winter' image JavaScript to use this function *in place of the previous JavaScript code.*

```
onMouseover="winterTextChange( )"
```

Save and preview your page. When you mouse-over the winter image, the image changes and the text changes. This is really useful!

In this example, we change the text to "Snowy" and it stays – since we never change it to anything else. As a final exercise, try adding this code just below the line: *'document.mainpict.src='snow.jpg'* in your function:

```
setInterval ("t.innerHTML='Back to Normal'",2000);
```

Test your web page and notice the new activity!

For extra credit, change the text in a logical way for *all four seasons*. When you're finished, put your file in the IN folder.