

# Basic Exercise with Javascript:

## *Basic Scripting*

**Overview:** A popular programming languages used for making Web pages interactive is called *Javascript*. Though learning to program large projects from scratch is beyond the scope of this class, a number of simple actions can be accomplished relatively easily.

### 1. Javascript Overview

*Javascript* is what is known as a *scripting language*. It 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, 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.

Sound familiar? It should. The Dreamweaver *behaviors* that you have experimented with all work due to Javascript. Every behavior that you define is implemented by a Javascript script, which you can see by viewing the HTML code of one of your old pages that includes a behavior or rollover image. Javascript scripts are extremely useful for a variety of activities, many of which have clear practical applications.

### 2. Copying and Modifying Javascript Scripts

As noted in the overview, we do not have enough time to learn the entire Javascript language in this lab, but we will begin to gain an understanding of how it can be used effectively by borrowing and modifying existing scripts. And if you have some task in mind that you'd like to explore using Javascript, just ask and I'll help you figure it out.

#### *Case study #1: Changing Image Attributes*

Suppose you wish to display an image on a Web page and have that image change as you move your mouse over the navigation choices. Javascript provides a simple solution. As noted above, Javascript provides the capability of manipulating the various objects on a Web page while the page is being displayed. In this case, we could use it to dynamically changing 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. They should be stored in a folder in the class *OUT* folder . Copy them to the *images* folder of your demonstration Web site, or of a new site that you set up.

- 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: *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 navbar from previous labs. 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 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 the interactivity to each of the 4 images using Javascript. 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*
- 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 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

## Functions

At times, the onmouseover 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**. Functions are simply sub-programs that can be "run" by simply using the function name.

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='Snowwy';
    document.mainpict.src='snow.jpg';
  }
</script>
```

- m. When run, this code will find a document element called 'maintext' and then will change it to 'Snowwy' [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's 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 "Snowwy" 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!

Now go out and think of how you might add this type of  
interactivity to your project!