Basic Uses of JavaScript:
Modifying Existing Scripts

Overview: A popular high-level programming languages used for making Web pages interactive is *JavaScript*. Before we learn to program with JavaScript from scratch we will learn some of the basics by copying (and in some cases modifying) scripts written in JavaScript. Throughout the activity, make notes about the details of the programs, and ask lots of questions. Later you'll be asked to write simple JavaScript programs on your own.

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. We will discuss some of these in this activity.

In CSC-101 you'll learn the basics of turning algorithms into JavaScript programs. The course CSC-111 delves into this topic in considerably more detail. If you're interested, please ask me for details.

2. Copying and Modifying JavaScript Scripts

We will begin to gain an understanding of how JavaScript can be used effectively by borrowing and modifying existing scripts from elsewhere on the web. And if you have some task in mind that you'd like to explore using JavaScript (perhaps for your web site homework), just ask and I'll help you figure it out.

Case study #1: An on-line slide show

Suppose you wish to display a sequence of charts on a web page, or a series of product images, or a collection of artwork. In short, any series of images. One thing you could do is put each image on a separate web page and construct links between the pages, but that's a lot of work.

JavaScript provides a perfect 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 swap images without changing the HTML page itself. What is needed is a relatively simple program that is embedded with the HTML code.

Dreamweaver provides a tool for working with JavaScript scripts, but it isn't as convenient as some of the other features of Dreamweaver. (It actually presumes a

reasonable amount of knowledge about scripting ahead of time.) So we'll regress a little bit and return to editing HTML directly. Our project will be to make a slide show of some amusing versions of the Furman logo. (They should be stored in the CSC-101 *OUT* folder. Copy them to the *images* folder of your demonstration web site. Do not copy the whole folder – just the images themselves.)

a. Use the Dreamweaver editor to open a new page within your demonstration Web site. Save it with the file name `slideshow.html`. **(Don't forget to set up a Dreamweaver *site* first.)** Click the "Show Code View" button to edit the HTML code.

b. What we need is a place for the first image of our sequence to be displayed, and buttons for moving to the next picture and returning to the previous picture. Enter the following code in the `<BODY>` section of the page:

```
<CENTER>
<H1>Slide Show
<P><IMG SRC="images/furman3.gif" NAME="myPicture"></P>
<P>
<A HREF=javascript:processPrevious()> Previous </A>  
<A HREF=javascript:processNext()> Next </A>
</P>
</H1></CENTER>
```

Take a look at this code for a moment. Much of it (the HTML stuff) you should be able to interpret. What we basically have done is inserted one of the Furman images on the page, with two hyperlinks over the words "Previous" and "Next". One new item is the inclusion of the name "myPicture" for the image. This is called a *variable* in programming. It is an abstract name given (in this case) to the image in this particular position. Right now the image is **furman3.gif**, but later ***we can change this*** using the variable "myPicture".

Also new are the URLs for the "Previous" and "Next" links. Rather than refer to different pages on the Web, they specify that particular Javascript *functions* are to be called. A function in programming is a segment of code that may be called upon to perform a specific task. In this case, we need these functions to assign new images to the variable "myPicture". Next, we'll implement these functions.

c. Javascript function code is typically placed in the header of the Web page. Position the cursor just before the `</HEAD>` tag. Then type the code on the following page.

```
<SCRIPT LANGUAGE=JAVASCRIPT>
var myPix = new Array("images/furman3.gif",
        "images/furman2_apfl.gif",
        "images/furman2_ghog.gif",
        "images/furman2_val.gif",
        "images/furman2_newyear.gif")
var thisPic = 0

function processPrevious() {
        if (thisPic > 0) {
                thisPic--
                document.myPicture.src=myPix[thisPic]
        }
}

function processNext() {
        if (thisPic < (myPix.length-1)) {
                thisPic++
                document.myPicture.src=myPix[thisPic]
        }
}
</SCRIPT>
```

Let's take a look at this code. What the SCRIPT tags do is specify to your browser that JavaScript is the language we are using. Next, we indicate the list of images that will be used in the order we want them to appear. They are stored in an *array*, which you should remember from our algorithm discussion. Finally, we have the two *functions* that will be executed when the "Previous" and "Next" buttons are clicked.

Without going into too much gory detail, what we have here is the "processPrevious" function setting the variable "myPicture" to indicate the image before the current one in the image sequence, unless the current image is the first one in the sequence. The "processNext" function advances the value of "myPicture" to the next image in the sequence, unless it is at the end.

d. Save the page and preview it. Try out the links. Assuming no typos or other errors, the slide show should be working.

e. Now modify the script to create a slide show with images of your own choosing. Use Furman postcard images from `http://www.furman.edu` or some pictures from the class photo roster if you don't have any others to use. Try using more or less than the five images used in the original slide show. You'll have to have at least a basic understanding of how this works to make these changes.

This example gives a pretty good feel for what JavaScript is like as a language, but it is basically a "showy" example. ***Before moving on, take as much time as necessary to understand the code and ask questions about it.***

Case study #2: On-line calculators

The ability to do various kinds of calculations on a web page can be very convenient for both commercial and educational sites. JavaScript calculators for many different problems already exist, and the creation of tailor-made scripts is not difficult, even without a complete mastery of JavaScript. They serve as productive examples for how JavaScript can be used to manipulate data entered into *HTML forms* (one of the strengths of JavaScript.)

Rather than write a script from scratch this time, we'll see how easy it is to "borrow" an existing one from the web.

A very good archive of different Javascript calculators can be found at:
`http://www.webwinder.com/wwhtmbin/javacalc.html`
Copying one is a relatively easy task.

a. Access the URL given above. Then select the calculator that you would like to use. The temperature conversion example is a good one, but you may choose any that you like.

b. This particular site gives relatively clear instructions for copying their code. The process typically involves one or two copy-and-paste operations. As we saw with the slide show, there is usually some JavaScript code for the <HEAD> part of a page, and some JavaScript mixed with HTML for the <BODY> part. Give this a try with a new blank page in your demonstration site. Remember to click the "Show Code View" button first.

NOTE: If you find some JavaScript that you want to copy from a different site, it's still pretty easy. Since the script code is contained in the HTML for the page, what you do is simply execute **File**:*Save As* and save the entire page somewhere in the directory for your site. Then open the page that you saved in Dreamweaver and edit it.

c. Click "Show Design View" for your new page and you can do some basic editing to get rid of text, images and ads that you don't want. Then you'll have a working calculator as part of your web site.

The reason for stepping you through an example like this is to show you how an interesting calculation that you require might be implemented with only a minimal understanding of JavaScript. If you have an idea for a calculator that doesn't appear on this web site, let me know and we can work on implementing it. In another (optional) lab exercise you will have the chance to learn about web *forms*, which are the last piece of the puzzle.

Another source of pre-written JavaScript scripts

We have seen several ways that Dreamweaver *Behaviors* can be used to add interactive features to Web pages in the last few weeks. Even more impressive features are possible, and you still don't have to learn JavaScript to use them. A brief look at the *Dynamic Drive* Web site will show you what I mean.

Dynamic Drive is one (of many) sites that stores pre-written JavaScript programs that you can copy and modify for your own use for free. As you'll see, doing this is as simple as copying and pasting some JavaScript code. Let's give it a try.

a. Use your browser to go to `http://www.dynamicdrive.com`. Look in the section titled "Script Categories". Select a category that looks interesting.

b. From the category you've selected, choose one of the effects to see what it looks like. If you'd like to give it a try, just follow the directions.

c. For example, suppose you select "Live Clock" from the "Date and Time" category. To use it, you would first copy the code in the first window and paste it into the HTML code of your web page. (Note that you need to use "Show Code View" here. If you paste into your WYSIWYG Dreamweaver window, the *JavaScript* is what will appear, not the clock.) Then you would go to the BODY tag of your page and add the `onLoad` code given at Dynamic Drive. Next, you would download the file `liveclock.js` simply by clicking on it. You'd store it in the folder that contains your web page. Finally, you would open this file to see what aspects of the clock (how it looks, etc.) you can change with some simple edits.

Note: It is possible that a file that you download from Dynamic Drive will have the extension `.ZIP` on the end of it. This means that multiple files have been stored in a single *archive*, and you will have to *extract* the files from the archive using a program called WinZip, or the Windows operating system itself. Doing this is easy. Find the ZIP file that you downloaded and right-click on it. Choose the option "Extract to here" and the archived files will appear in your folder.

d. Experiment a little bit and pick a script that you like. Save it to a file with your other web pages (or ***on*** one of your other web pages).

As you can see, it's not very difficult at all to give visitors to your site the impression that you're quite the web authoring expert!

A JavaScript tutorial

The preceding examples (plus your work with Dreamweaver behaviors and rollovers) should give you a glimpse of what JavaScript can do, and how you can use it to accomplish interesting tasks with images and form information. At this point, you can focus on a specific JavaScript project and use the web or your professor to figure out how to do it, or you might prefer to roll up your sleeves and dig in further. To do that, use your browser to access this excellent tutorial:

`http://echoecho.com/javascript.htm`

In class and subsequent labs we'll be learning more about how to create our own JavaScript programs.

FINALLY: Create a new page in your existing site with the filename `scripts.html` and put on it links to all the new pages you've created – and a link to your home page. (There should be four links in total: slide show, JavaScript calculator, JavaScript effect from Dynamic Drive, and a link to the home page. Even if you put multiple things on the same page, please create separate links so I'll know you did everything.) Then make a link to this page from your home page. Upload everything to the server and test it.