

# Basic Uses of JavaScript and Java: *Modifying Existing Scripts and Applets*

**Overview:** Two popular high-level programming languages used for making Web pages interactive are *JavaScript* and *Java*. Though learning to program with them from scratch is beyond the scope of this activity, a number of useful objectives can be accomplished by copying (and in some cases modifying) scripts written in JavaScript or applets written in Java. It is this idea that we explore today.

## 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.

Though it is not one of the goals of CS16 to learn to program in JavaScript, it is one of the goals of the CS19 class that I told you about. If you're interested, please ask me for details.

## 2. Copying and Modifying JavaScript Scripts

As noted in the overview, we do not have enough time to learn the JavaScript language in total in this lab, but we will begin to gain an understanding of how it 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, 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 alternate versions of the Furman logo. (They are stored in the CS16 *OUT* folder in a folder called *Scripting Lab*. Copy them to the *images* folder of your demonstration Web site. Do not copy the whole folder – just the images themselves.)

- a. Copy the file `slideshow.html` to your demonstration Web site folder. Set up a Dreamweaver *site* that points to this folder. Open the `slideshow` file and click the “Show Code View” button to edit the HTML code directly.
- 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. Look at 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> &nbsp;
<A HREF=Javascript:processNext()> Next </A>
</P>
</H1></CENTER>
```

Let's consider this code. 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 see how to implement these functions.

- c. JavaScript function code is typically placed in the *header* of the Web page. Look now at the code between the `<HEAD>` and `</HEAD>` tags, reproduced on the next 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 briefly. 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, and we give them (collectively) a name – myPix. 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. (Note that I would be happy to go into the gory details with you if you are interested in this. Just ask.)

- d. The list of images has not been entered into the code that was provided to you. Enter the list now, being sure to type the folder name (`images`) and file names correctly. Remember that the computer will not be able to figure out even the simplest typos.
- e. Save the page and preview it. Try out the links. Assuming no typos or other errors, the slide show should be working.
- f. 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. Use more 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. Be sure that the original image (the one that begins the slideshow) is changed appropriately!

This example gives a pretty good feel for what JavaScript is like as a language, but it is basically a “showy” example. Next we'll get more practical.

### 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/wwhtm/bin/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 the lab exercise that you (optionally) completed before this one, you learned about web *forms*, which is 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 JavaScripts 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 box titled “DHTML Script Categories”. (Recall that DHTML stands for “dynamic HTML”.) 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 “Dynamic Clocks and Dates” 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 Windows XP or a program called WinZip. Instructions for how to do this will be given in a class demonstration.

As you can see, it’s not very difficult at all to give visitors to your site the impression that you’re quite a 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://www.webmonkey.com/webmonkey/programming/javascript/tutorials/tutorial11.html>

## 3. Java Overview

Java, unlike JavaScript, is a full-fledged, standalone programming language. That is, it has all the power and flexibility of commonly used languages such as C++, Ada and Pascal, and thus does not need to be used in a Web environment. One of the reasons for its popularity, however, is that it *does* work seamlessly with the Web. Skilled programmers can use Java to create applications — called *applets* — that can access and modify files through Web pages, create dynamic pages, and display visual effects, among other things. Java is differentiated from JavaScript in that it is far more powerful; it can be used apart from the Web, and programs that you write with Java are not contained within Web pages.

A program in Java is called the Java *source*. (It is stored in a file with the extension `.java`.) This is the code that is (more or less) readable by humans. Java source code must be *translated* by a complicated process into the binary language of 0's and 1's that a computer can understand. This translated code is called an *applet* and is stored in a file with the extension `.class`. If you have access to the `.class` file, you can use an applet yourself, simply by embedding it in your Web page using HTML. In order to modify an applet from its original functionality, you'd need to know how to edit the source code appropriately, and then translate the source into the applet (which requires another program, called a Java *compiler* or *interpreter*.)

What is so special about Java is that these applets are *machine independent*. That means it doesn't matter if you're using a Mac, a PC or a UNIX system. As long as your operating system and your browser is current enough, the applet will work.

Next, we'll take a few moments to download and incorporate an applet on a Web page.

#### 4. Copying Java Applets

Fortunately, you can make use of Java applets without learning Java, just as you can with JavaScript scripts. In fact, copying Java applets is just as easy as copying Web pages. As is the case with copying Web pages, though, you can't modify a Java applet without knowing the Java language. For this activity, we'll focus just on copying an applet as-is.

- a. Use your browser to access the site:  
`http://javaboutique.internet.com/cathome.html`
- b. There are many useful, interesting, or just plain entertaining applets at this site. It may not be too practical, but just for practice, let's install one that displays text with a nice effect. Click on "Text Effects", and then look for "CoolScroll", or another that catches your eye. You will then go to a page where the applet is demonstrated.
- c. Scroll down the page. There you will find links to a *ZIP archive* containing the applet source files (which you don't really need, unless you are planning to modify the applet), the `.class` files, and some images. You'll also find sample HTML code for embedding the applet in a Web page.
- d. Download the `.ZIP` file to your web folder. (Just click on it to download it.)

Next, you'll need to *extract* the files from this archive to your Web directory. I'll show you how to do that in a lab demonstration. It involves Windows XP or the program *WinZip*.

**Note:** If you've chosen a different applet to download, it's very possible that the file you have downloaded has `.EXE` on the end of it rather than `.ZIP`. This is good news. It means that the files you need are in a *self-extracting archive*. All you have to do to get them out is double-click the file that you downloaded. The WinZip program isn't needed at all.

- e. Highlight the entire segment of code labeled “HTML Source” on the Javaboutique page and copy it. Then use Dreamweaver to create a new page. Click the “Show Code View” button and paste the code into it.
- f. Return to the Javaboutique site from which you copied this applet. Read over the explanation of what the various parameters (settings) of the applet mean. The most important in this case is the “scrolltext” parameter, which specifies what the applet will display.
- g. Return to Dreamweaver. Click once on the Java applet icon to highlight it. Look in the Properties window and click the **Parameters** button. In the “Applet Parameters” box, find the “scrolltext” parameter. Change its value to MY WEB PROJECT FOR CS16 in all caps. Close this window.
- h. Save and preview the new page. Do you think this would make a good banner for your web site homework project? ☺

A bit of Web searching will turn up hundreds of pre-coded applets that accomplish different tasks. Perhaps there are some that will serve your purposes. The general procedure for acquiring and personalizing an applet is always basically the same. (1) Download the `.class` file(s) and any other supporting files as a single archived, compressed file. (2) Decompress the file using WinZip (if necessary) and make sure the files are in your Web directory. (3) Copy and paste the HTML code that calls and configures the applet. (4) Edit the applet parameters so the applet does what you want.

**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 five links in total: slide show, JavaScript calculator, JavaScript effect from Dynamic Drive, Java applet and 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 (`scripts.html`) from your home page. Upload everything to the server and test it. (You’ll have to upload any extra files for your Java applet as well.)