

ASK THE HOMEGURRRR!



More on Interactivity

BY LYNDA WEINMAN

Excerpted and edited with permission from Lynda Weinman's newest book on Web design, <designing web graphics.2>, published by New Riders Publishing and distributed in Canada by Prentice-Hall.



SOME TIME AGO, I PROMISED TO RETURN TO THE SUBJECT OF more advanced interactive techniques and tricks for building your website.

So in this article we will talk about ways to create action and effects through interactive design devices.

MOUSE ROLLOVERS

Mouse rollovers are very popular in CD-ROM-based multimedia, and are being used more and more on the Web as people learn how to make them. On the Web we have a few visual cues for hyperlinks, such as the hand symbol, the bounding box around linked graphics, and underlines below text. These are fairly limited visual signals. Rollovers offer much greater design flexibility.

A rollover can make type appear to glow, change the color of an icon, or make a sound. The possibilities for rollover effects are limited by your imagination only.

In this article, let's evaluate two technologies for rollovers: JavaScript and Macromedia *Director/Shockwave*.

JAVASCRIPT MOUSEOVERS

JavaScript is a programming language developed by Netscape. Unlike Java, C, Perl, or sh, however, you can view the JavaScript code within HTML documents. This makes it very easy to learn and "borrow" from others' sites. Proper netiquette recommends that you always leave references to the originator within the code if you do copy the scripts. These credits are usually left in comment tags, which look like this:

```
//So-and So's Amazing Rollover Script//
```

or like this

```
<!--So-and So's Amazing Rollover Script!-->
```

You'll see these types of credits in the following examples.

JOE MALLER'S JAVASCRIPT ROLLOVER

Joe Maller, a digital designer in New York City and computer imaging instructor at Parsons Institute, uses his site as

an experimental playing field for all types of new Web technologies. Check his stuff out at:

■ <http://www.joemaller.com>.

The following tutorial was written by Joe and is also on his site.

Declaring Images in the Script. In the <SCRIPT> section of the <HEAD> of the document, you need to first declare the images you will later use for your rollover. The JavaScript to set up those images looks like this:

```
Image1 = new Image(100,150)  
Image1.src = "your_image.gif"  
Image2 = new Image(100,150)  
Image2.src = "your_other_image.gif"
```

Each image must have two lines: the first sets its size, and the second sets its source and the location and name of the image. The numbers within the parentheses refer to width and height, in that order. Note that JavaScript is case sensitive; TheDog is not the same as thedog.

Naming Image Objects. JavaScript 1.1, as implemented in Netscape *Navigator* 3.0, defines images as objects. That

joe_blink.jpg



joe open.jpg



Joe's mouseover effect causes his eyes to blink. It's fun—and worth trying at at home!

means the JavaScript now recognizes images as more than just HTML.

To replace an image, you must give JavaScript some way of identifying where you will replace the images. Do this by either inserting **NAME="Rupert"** into the **<IMG...>** tag, or by learning to refer to images in JavaScript's Array syntax: `document.images[2]`

The above refers to the third image tag in the HTML document (you're programming now — zero counts first). I prefer to simply name my images, and that's what I've done in this example.

For JavaScript to recognize a rollover, the object rolled over must be a link; regular text is ignored. Here is the HTML code for the image that will be replaced during a rollover:

```
<A HREF="link.html"
onMouseOver="SwapOut()"
onMouseOut="SwapBack()">
<IMG NAME="Rupert"
SRC="your_image.gif"
WIDTH=100
HEIGHT=150>
</A>
```

Notice the name attribute's appearance in the **<IMG...>** tag.

What's "onMouseOver"? `onMouseOver` and `onMouseOut` are event handlers that tell JavaScript what to do when an event occurs. The command given is the function within the quotes. `onMouseOver="blorg()"` tells JavaScript to execute the function `blorg()`.

Functions. Functions are predefined sets of commands that JavaScript will execute when only the name of the function is called. Functions are usually stored in the **<SCRIPT>** section of the **<HEAD>** of the document, the same place we declared the images earlier. The standard format for defining a function is:

```
function SwapOut() {
document.Rupert.src = Image2.src; return true;
}
```

PUTTING IT ALL TOGETHER

The following is the complete code for a simple document that will change images. Don't believe me? Try it. Here's the code:

```
<HTML>
<HEAD>
<TITLE>
Doubter's page
</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!-- hide from none JavaScript Browsers
Image1 = new Image(121,153)
Image1.src = "joe_open.jpg"
Image2 = new Image(121,153)
Image2.src = "joe_blink.jpg"
function SwapOut() {
document.Rupert.src = Image2.src; return true;
```

```
}
function SwapBack() {
document.Rupert.src = Image1.src; return true;
}
// - stop hiding -->
</SCRIPT>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<CENTER>
<P>
<A HREF="http://www.joemaller.com/"
onmouseover="SwapOut()"
onmouseout="SwapBack()">
<IMG NAME="Rupert"
SRC="joe_open.jpg"
WIDTH=121
HEIGHT=153
BORDER=0>
</A>
</P>
</CENTER>
</BODY>
</HTML>
```

If you use this, please give me credit.

BILL WEINMAN'S JAVASCRIPT ROLLOVER

This example demonstrates that rollovers, like many things, can be done more ways than one. This is Bill's script:

```
// bill's suave mouseover javascript
// (c) 1996 wew http://www.weinman.com/wew/
//
// If you want to use this, go ahead;
but please leave
// this notice here so that other people
know where it
// came from. —wew.
//
// how many items in the array
numitems = 6
// the offset to the first GIF that's
going to change. This is
// the number of GIFs on the page before
the first one that
// will swap, minus 1.
offset = 7
// creates the array for the GIFs.
don't touch this.
Atitles = new Array(numitems)
// this initializes the array.
for(i = 0; i < numitems; i++)
{
Atitles[i] = new Image()
}
// blankgif is the default GIF.
this one's white 1 pixel x 1 pixel.
// WARNING: transparent GIFs create problems on Macintoshim!
blankgif = "white.gif"
// These are the GIFs that will overlay
the white ones when the
// mouse is over them.
```

```

Atitles[0].src = "tbio.gif"
Atitles[1].src = "tproj.gif"
Atitles[2].src = "tbook.gif"
Atitles[3].src = "tbear.gif" Atitles[4].src = "tlinks.gif"
Atitles[5].src = "tmail.gif"
// this function swaps out the GIFs,
and updates the status line.
function description(m,i) {
status = m
imgtmp = document.images[offset + i].src
document.images[offset + i].src = Atitles[i].src
return true;
}
// this function restores the blankgif and
the status line.
function clearstat(i) {
status=""
document.images[offset + i].src = blankgif
return true;
}
}
</script>

```

DIRECTOR/SHOCKWAVE/LINGO-BASED INTERACTIVE WEB PAGES

Shockwave is the name of a plug-in that enables *Director*-based multimedia projects to be played on the Web. Multimedia can be described as anything that combines sound, animation, images, text, and interactivity.

■ note

Shockwave Versus JavaScript

Director projects (which later get converted to Shockwave) can be much more robust than what is possible through JavaScript. With the added power (a paint and animation interface, and a proprietary programming language called Lingo) comes a steeper learning curve. You can't see the source code for a Director project like you can a JavaScript, making it impossible to reverse engineer what others have done. Learning Shockwave and Director requires buying the program, reading the manual and/or third-party books, and/or joining users groups or discussion groups. The power is definitely greater, but the learning curve may not be worth it to you. For more information, visit ■ <http://www.macromedia.com>.

Unlike JavaScript, Macromedia *Director* is an authoring program for creating interactive multimedia for kiosks and CD-ROMs, and it has been around many years longer than the Web itself. The Shockwave file format boasts an impressive compression ratio of three to one, making the *Director* files one third of their usual size.

Afterburner is the free post-processing tool that offers authors who create *Director* multimedia projects the ability to convert them so that they can be distributed over the Web. Once *Afterburner* has converted a *Director* project, that project becomes a Shockwave file. You can download

Afterburner from Macromedia's site at

■ <http://www.macromedia.com/shockwave/devtools.html#director>.

LEROY'S CLICK AND DRAG

Bruce Heavin has this simple Shockwave piece on his site.



Leroy's Click and Drag is at ■ <http://www.stink.com/leroy/leroy.html>. Every graphic on this page is click and draggable.

DECONSTRUCTING THE LINGO FOR SHOCKWAVE

Adding interactivity to *Director* projects is accomplished through using a proprietary programming language called Lingo. Lingo is touted as an easy programming language — although non-programmers might not necessarily agree with that analysis. Learning Lingo is possible by studying the manuals that are shipped with Macromedia *Director* or through third party books and Usenet groups. *

LYNDA WEINMAN IS AN ACCOMPLISHED INSTRUCTOR, DESIGNER, INTERNATIONALLY KNOWN SPEAKER, AND BEST-SELLING AUTHOR. SHE CAN BE REACHED AT LYNDA@LYNDA.COM.