

his month we're going to start looking at JavaScript. JavaScript is a lightweight way to add active content to your web pages. Since JavaScript is very similar to the C and C++ programming languages, those who have a background in those languages will have a head start. However, JavaScript is not as daunting as it may seem. I'll take it easy on those of you who have never programmed a lick of C++... We'll start with the simplest application of JavaScript and work our way to more difficult things.

From Static to Active Design

Sooner or later, every web page designer has the urge to create pages which grab the web surfer's attention. I'm not simply talking about snazzy graphics and a nice layout for the page. I'm talking about active content: animated graphics, movies, background sounds, and most importantly, JavaScript.

Animated GIFs (an image format) are produced using a program like GIFBuilder. An animated GIF is a sequence of single GIF images.

Basically, it's the electronic counterpart to those flip-book animations — you know, draw something on the corner of each page of a book, and as you flip through the pages you have the illusion of motion. Though many animated GIF tools are available, my favorite is GIFBuilder, <http://iawww.epfl.ch/Staff/Yves.Piguet/clip2gif-home/GifBuilder.html> .

As far as movies and background sounds are concerned, you're a little more limited in your choices. Many times background

music

for web pages is a MIDI file — a MIDI file is basically like the sheet music for a song. The computer reads the "sheet music" and uses electronic versions of the instruments to play the song. MIDI files have the benefit of being small compared to most other media formats. Movie files can become large quickly; plus, they're not as easy to create as the simple animated GIF images. Many people dislike having their web browser spurting out the MIDI rendition of Hotel California, so think carefully before implementing background music.

What's this JavaScript Stuff?

So what the heck am I talking about when I say "JavaScript?" Well, anyone who has played around with AppleScript on his or her Mac knows what scripting is about: a programmable interface to each and every application. With a little bit of knowledge, you can control the internal aspects of a program and add new functionality. In the same respect, JavaScript gives your web page more control over its interface. For instance, have you ever wondered how a web page designer makes the buttons on a page seem to glow or depress when the mouse moves over them? Or even how he does something so simple as putting "Jump to the Main Page" in the browser's status bar (generally the bottom portion of the browser window) instead of "<http://www.nowhere.onearth.com/home.html>?" JavaScript is the answer.

The Magic Status Bar Modifier

Before we jump into active web pages which have glowing text, we should start simple. Let's take a look at how we modify the text in the status bar of the browser.

The first question to ask is when do we need to put text in the status bar? The functionality we wish to modify is when the mouse is on top of a link, we wish to display our own text, not the URL of the link. Therefore, to answer the when: we want to display our text as the mouse moves over a link.

JavaScript works like many programming methods: it reacts to events, or a user interaction of interest. Events include mouse clicks, the mouse moving into an area, the mouse moving out of an area, and the user typing on the keyboard. The event that corresponds to our when is when is when the

mouse moves onto the link. In JavaScript such an event is labeled by the "onMouseOver" event type.

When the mouse moves over our link, the "onMouseOver" event is raised. To react to that event, we need to write some kind of JavaScript command and associate it with the "onMouseOver" event. We do this the same way we add any option to an HTML tag:

```
<A HREF="index.html" onMouseOver="window.status='Back to main page'; return true;">
```

As you can see, we've simply assigned an action to the `onMouseOver` tag in our link. Now when the mouse moves onto the link, the status bar displays the text "Back to main page" instead of a URL. We have to add the "return;" to our action to tell the browser that it should not print the URL in the status bar. However, for Internet Explorer users, a slight problem crops up — when the mouse moves off of the link, the status line remains! If you feel like typing some extra code, you can alleviate this problem. But how?

Thinking in terms of events, we need to intercept an event that indicates we're outside of the area in question. Can you guess the name of that event? That's right, "onMouseOut" is the event which is raised when the mouse moves off of our link. Since we want to get rid of the text we displayed when the mouse was over the link, we simply tell the status bar to display a blank string:

```
<A HREF="index.html" onMouseOver="window.status='Back to main page.';return true;"  
onMouseOut="window.status="";return true;">
```

That's all there is to it! When the mouse is over the link, the "onMouseOver" event takes place, which then displays "Back to main page." Then as the mouse is moved off of the link, the "onMouseOut" event takes place, and the text is cleared from the status bar.

or Next Month...

Next month we will discuss more of the events available in JavaScript. We will be working our way toward the creation of an interactive web site menu — for an example, check out <http://indigo.lvc.edu/~sg/> .

Good luck, and happy coding!

Jeff Frey
jeff@applewizards.net

<http://applewizards.net/>