

Simple Javascript for the Web

Joe Struss

March 7, 2011

A Cycling Banner with Links



Visit the IT Learning Pod's
Online Web Site at:
<http://css.ait.iastate.edu/>

Copyright © 2011 by Creative Services/ITS

Permission to reproduce all or part of this document for noncommercial purposes is granted, provided the author and Iowa State University are given credit. To copy otherwise requires specific permission. The author's contact information is available at the IT Learning Pods web site.

Setting Up For Class

Downloading Data and Editing

To learn a little something about Javascript (JS) we are going to need a specific desktop setup which isn't overly complicated.

First download the js.html and graphic files for this class (quips.gif, grandc.gif and drivein.gif) at the following location:

<http://css.ait.iastate.edu/Networking/Data1/>

Right-click (Windows) or Ctrl-click (Macintosh) the file **JS1Archive.zip** and download the file to your desktop. After that extract the files from the archive then take the files from the JS1Archive folder and place those files onto your desktop.

Next we are going to open the file js.html with a standard text editor.

Macintosh

On a Macintosh, double-click the **TextEdit** application that is normally within the **Applications** folder. Go under **TextEdit** → **Preferences** then click on **Open and Save**. After that go under **When Opening a File:** and click on the option to **ignore rich text commands in HTML files**.

Then pull down under **File** → **Open** and open up the file **js.html**.

Windows

On a Windows system, open up WordPad by going under **All Programs** → **Accessories** → **WordPad**. Then pull down under **File** → **Open** and open up the file **js.html**.

Browser Setup For Class

Finally just open up your favorite internet browser and point it at the file **js.html** on your desktop. So open up your internet browser, pull down under **File** to **Open** and open up the file **js.html** on your desktop.

So to be ready for class, you should have an editor open, editing a file named **js.html** and a browser open that is pointed at the file **js.html**. We are going to make changes to the **js.html** file, save the those changes with the editor and then reload the page on the internet browser to see the effects take place.

HTML Basics

HTML stands for Hypertext Markup Language. It is the markup language used to create most web pages on the internet. HTML commands known as tags are enclosed in `< >` and mostly come in pairs `<H1>Topic</H1>` which surround the area where the effect is to take place with a `/` preceding the ending tag name. For educational purposes and emphasis, HTML commands are shown within this document in both upper and lowercase but should all be in lowercase to be XHTML compliant.

The HTML Page

A full HTML page is made up of various sections.

`<HTML>`

– marks web page as an HTML file

`<HEAD>` Title stuff `</HEAD>`

– marks header of this web page.

`<BODY>`

`<p>`Everything else on this page.`</p>`

`</BODY>`

– marks body of this web page.

`</HTML>`

- marks end of an HTML file

An initial full HTML page might look something like this:

```
<HTML>
<HEAD>
<TITLE>
Personal Homepage -
Your Name Here</TITLE>
</HEAD>

<BODY>

<H1>Welcome!</H1>

<P>This is a sample homepage.</P>

<P>Text can be shown in <B>bold</B>
and <EM>emphasis</EM>.</P>

<p>My electronic mail address is
<ADDRESS>somewhere@out.there</ADDRESS></P>

</BODY>
</HTML>
```

Simple JS: The Basics

What is Javascript and how does it differ from Java?

Javascript is a scripting language that can add interaction to an HTML Web page. Java is a object-oriented programming language this is most often implemented on a Web page through APPLET tags. Despite the names being similiar, Javascript is very different from Java and the two have almost nothing in common.

Having said that, Javascript is object-based and uses commands referred to as *event handlers* to add Web page interactivity. Some example event handlers are: onMouseOver, onMouseOut, OnClick, onSelect, onBlur and onLoad which all refer to events that might happen on a Web page.

Borrowing Javascript Code for use in class:

A complete set of Javascript codes for use in this class is located at:

`css.ait.iastate.edu/Networking/jsclass1.html`

Follow a link to the Javascript trick that we are doing in class. When you are on the correct page, within your internet browser pull down under **View** to **View Source**. This works for most javascripts.

Next highlight the javascript code that you would like to copy. Then enter `<CMD/C>` (Macintosh) or `<CTRL/C>` (Windows) to copy the selected text. Next go to your Web page and enter `<CMD/V>` (Macintosh) or **Edit** → **Paste Special (Unformatted)** (Windows) and paste the text onto your Web page. Try extending the example by altering the code.

ERRORS: If you get an error, enter `javascript:` in the URL box to get the javascript error messages. **Remember** javascript is case sensitive and somewhat touchy. Try simplifying your code if you run into trouble.

How do I add Javascript to a Web page?

To put Javascript code onto a Web page, enter the following into the HEAD or BODY areas:

```
<SCRIPT LANGUAGE=JAVASCRIPT
  TYPE="TEXT/JAVASCRIPT">
<!-- Hides Javascript from old browsers
document.write("Put js code goes <b>here</b>.")
// Ends hiding of Javascript code -->
</SCRIPT>
```

Or you can create a filename.**js** file and put your javascript code in there:

```
<SCRIPT SRC="java.js" LANGUAGE=JAVASCRIPT
  TYPE="TEXT/JAVASCRIPT">
<!-- The following is a Javascript 1.0 fix
function imgOver() { }
function imgOut() { }
// -->
</SCRIPT>
```

So the java.js file would contain:

```
document.write("Put js code <b>here</b>.")
```

Simple JS: Conditionals, Loops & Functions

If Conditional:

```
if (condition) {  
    statement(s) to execute  
    if condition is true  
}  
else {  
    statement(s) to execute  
    if condition is false  
}
```

The *else* part is optional.

Loops:

```
for (counter; limit condition; increment) {  
    stuff in the loop  
}
```

Functions:

Functions usually occur in the HEAD area and then are used in the BODY area.

```
function nameoffunction(parameters){  
    function stuff  
}
```

Function Example- Put this function in the HEAD area as a javascript:

```
<SCRIPT LANGUAGE=JAVASCRIPT TYPE="TEXT/JAVASCRIPT">  
<!--  
function speak(msg){  
    alert(msg)  
}  
// -->  
</SCRIPT>
```

Put the call to the function in the BODY area in this case as standard HTML:

```
<FORM>  
<INPUT TYPE=BUTTON VALUE="Don't push this!"  
    onClick="speak('Do NOT push me again!')">  
<INPUT TYPE=BUTTON VALUE="Don't push this either!"  
    onClick="speak('You are PUSHING ME!')">  
</FORM>
```

From here on in, this handout will just show the plain javascript code and **not** the wrapper code that goes around it.

So the handout will leave out the following **HEAD** portion of the code as it is always the same:

```
<SCRIPT LANGUAGE=JAVASCRIPT TYPE="TEXT/JAVASCRIPT">  
<!--  
  (JS Code that is shown.)  
  // -->  
</SCRIPT>
```

Comments and Alerts:

Try out the following in the **HEAD** portion of your Web page:

Comments:

```
/* starts a javascript comment  
*/ stops a javascript comment  
/* Here is my comment */
```

Alerts:

```
alert("Put a Message here.")
```

pops up an alert window when a Web page is loaded.

Simple JS: Messages and Menus

Changing the Status Bar:

You can change the status bar by altering the *window.status*. Try the following in the **BODY** portion of your page:

```
<A HREF="index.html"  
  onMouseover="window.status='  ';  
  return true"  
  onMouseout="window.status='Click now!';  
  return true">  
  Return to my homepage</A>
```

Note that when you rollover the index.html link that the status bar changes to a blank using the *onMouseover* event handler. The *onMouseout* event handler puts a message into the status bar when you exit the rollover. You can change the link from index.html to any web page like <http://www.google.com> to make it work properly from you desktop.

Scrolling Title Bar:

To do a scrolling title bar, you just need to alter the document.title property on a rotating basis. Place the following as a javascript into the **HEAD** area of your Web page:

```
<SCRIPT LANGUAGE=JAVASCRIPT TYPE="TEXT/JAVASCRIPT">
<!-- Hide script from older browsers
msg = "Visit the Computer Survival School!";
msg = "...    ..." + msg;
pos = 0;
function scrollMSG() {
document.title = msg.substring(pos, msg.length)
    + msg.substring(0, pos);
pos++;
if (pos > msg.length) pos = 0
window.setTimeout("scrollMSG()",200);
}
// End hiding script from older browsers -->
</SCRIPT>
```

Call the function by altering the **BODY** tag:

```
<BODY onLoad="scrollMSG()">
```

This is also the last time I will show the first two lines and last two lines of a javascript.

Warning- Scrolling title bar messages are currently considered to be in bad taste and do not work on all internet browsers.

The scrollMSG runs itself as a loop, cutting up the document title into two parts. The part after where POS is located and then the part before POS starting at the beginning. The program then increments POS by 1 and then if POS is over the length of the message— it then resets POS to zero. After that scrollMSG statement, gives the function a time out for a certain number of thousandths of a second before continuing on. The program itself is started by the onLoad option on the BODY tag.

Play around with this script by changing the `msg = "Visit the";` line and also trying altering the number on the `setTimeout` line (`"scrollMSG()",200`). You should be able to produce your own scrolling message that is either faster or slower than the current message.

This is a very simple javascript but it does show the basics rather well.

The Scroll-And-Go Menu:

The Scroll-And-Go javascript is a nice one that allows the user a scrolling menu choice as to where to jump to next via a nice pulldown list. This is the HEAD part of this javascript.

```
function jumpPage(nLoc) {
newPage = nLoc.options[nLoc.selectedIndex].value
if (newPage != "") {
    window.location.href = newPage
}
}
```

The part that actually goes in the BODY portion of a web page is listed on the top of the next page. Simply enter a menu name after each OPTION tag. Within the double-quotes of each OPTION tag except for the first one, put in the URL or local file to advance to when that menu option is selected. You can add or subtract as many OPTION tags as you want.

```
<DIV ALIGN=CENTER>
  <FORM ACTION="gotoLocation.cgi" METHOD=GET>
    <SELECT NAME="newLocation"
      onChange="jumpPage(this.form.newLocation)">
      <OPTION VALUE="" SELECTED>Where to next?
      <OPTION VALUE="http://www.yahoo.com/">Yahoo
      <OPTION VALUE="http://css.ait.iastate.edu/">CSS
      <OPTION VALUE="http://www.its.iastate.edu/">ITS
      <OPTION VALUE="http://www.ebay.com">Ebay
    </SELECT>
    <NOSCRIPT>
      <INPUT TYPE=SUBMIT VALUE="Go There!">
    </NOSCRIPT>
  </FORM>
</DIV>
```

The NOSCRIPT area in the BODY portion of the Javascript, adds a 'Go There' button for non-Javascript browsers to push- once a menu item is selected.

Try altering this javascript to create your own site menu. You can also easily make pull down menus and other menu-like items with javascript.

Simple JS: Rollovers

A Simple Rollover:

This simple rollover uses the `onmouseover` and `onmouseout` event handlers to produce a rollover effect when the user rolls over the image. Try this out in the **BODY** part of your page:

```
<A HREF="index.html"
  onmouseover="document.bldg.src='quips.gif'"
  onmouseout="document.bldg.src='drivein.gif'">
<IMG SRC='drivein.gif' WIDTH=184 HEIGHT=119
  BORDER=0 NAME="bldg"></A>
```

Simply put, when the image is rolled over- `onmouseover` switches in the new image; and when the image is rolled off of- `onmouseout` puts the image back to its original form. It's normally a good idea to use two images that are about the same size for a rollover.

Now this effect is o.k. but it takes some time for the second image to load when the rollover begins. So it would be nice to pre-load the rollover image.

A Better Rollover:

You can preload images in a javascript in the **HEAD** area of your Web page:

```
/* document.images cond used for older browsers */
if (document.images){
    quips = new Image
    drivein = new Image
    quips.src = "quips.gif"
    drivein.src = "drivein.gif"}
else { quips = ""
    drivein = ""
    document.blck = ""}
```

Then create the rollover with your preloaded images in the **BODY** portion of your page:

```
<A HREF="index.html"
    onMouseover="document.blck.src=quips.src"
    onMouseout="document.blck.src=drivein.src">
<IMG SRC='drivein.gif' WIDTH=184 HEIGHT=119
    BORDER=0 NAME="blck"></A>
```

Try moving the IMG tag below the anchor tag and put a word in for the link. Neat effect if the first image is invisible.

Simple JS: Cycling Banners

Many pages use animated gifs for cycling banners but you can also create them using javascript. Put this portion of the example as a javascript into the **HEAD** portion of your page:

```
adimgs = new Array
  ("quips.gif","grandc.gif","drivein.gif")
adv = -1
advct = adimgs.length
function switchit(){
if (document.images) {
  adv++
  if (adv == advct) {
    adv = 0 }
  document.cycle.src = adimgs[adv]
  setTimeout("switchit()", 5*1000) } }
```

Then in the **BODY** part of your page, alter the BODY tag and put in the initial image:

```
<BODY onLoad="switchit()">
<IMG SRC='quips.gif' WIDTH=184 HEIGHT=119
  BORDER=0 NAME="cycle">
```

Adding a Link to a Cycling Banner:

Add the following in the **HEAD** area after the `adimgs` line:

```
adhttp = new Array  
  ("quips.html", "grand.html", "drivein.html")
```

and add the following `newloc` function after the `switchit` function:

```
function newloc() {  
  document.location.href =  
    "http://css.ait.iastate.edu/Midland/"  
  + adhttp[adv] }  
}
```

Then in the **BODY** portion, use the following image link:

```
<A HREF="javascript:newloc()">  
<IMG SRC='quips.gif' WIDTH=184 HEIGHT=119  
  BORDER=0 NAME="cycle"></A>
```

A perhaps easier way to do this one is to clear out all the previous scripts and simply copy and paste the new header script and body script from the example file.

Automatically Changing a Background Color:

You can automatically change your background color with a javascript. Just place the following javascript into the HEAD portion of your Web page:

```
bcolors = new Array("red","green","blue","orange")
bgit = -1
bgout = bcolors.length
function colorit(){
  if (document.images) {
    bgit++
    if (bgit == bgout) {
      bgit = 0}
    document.bgColor=bcolors[bgit]
    setTimeout("colorit()", 5 * 1000) } }
```

Then alter your BODY tag as follows:

```
<BODY BGCOLOR=RED onLoad="colorit()">
```

I'm not exactly sure why you would want to do this but it is kind of fun. **Note:** You can combine this with the previous example by using:
onLoad="switchit();colorit()"

Simple JS: References

The Javascript Source is a great online source for a diverse set of javascript examples:

<http://javascript.internet.com/>

Javascript Kit also has a fine selection of free Javascripts and some javascript tutorials:

<http://www.javascriptkit.com/>

Javascript.Com is a great javascript resource and includes an free e-mail javascript Newsletter to which you can subscribe:

<http://www.javascript.com/>

Also *EchoEcho.Com* has a fine set of javascript related tutorials:

<http://www.echoecho.com/javascript.htm>

Currently recommended Javascript text:

Javascript & Ajax (2008) by Tom Negrino & Dori Smith; Peachpit Press.