Managing World Wide Web Information in a Frames Environment: A Guide to Constructing Web Pages Using Frames

Donald L. Gilstrap – Wichita State University

Deposited 11/16/2018

Managing World Wide Web Information in a Frames Environment

A Guide to Constructing Web Pages Using Frames

Donald L. Gilstrap

Abstract

This article serves as a guide for building World Wide Web pages using frames-based HTML. It covers the basic of frames tags and provides descriptions and examples for writing frames HTML code. Advanced concepts are included, as well as additional techniques for home page design.

Introduction

Over the past few years, the rise of World Wide Web-based resources has placed an increasing emphasis on the creation of library home pages that facilitate access to remote and electronic information. As the newer generation of students arrives at academic, public, and school libraries, they bring with them different cognitive styles based on a multimedia approach to learning. Students' learning habits have become more and more centered around visual representations of knowledge, and the World Wide Web has enabled them to navigate in a windows-oriented environment rather than just one window of information. And it's not just the students. Teachers, professors, administrators, and layfolk are now using this multiple windows environment when performing daily tasks on their computers. It is now possible to work on several applications simultaneously while navigating through various forms of electronic information.

This poses a new and exciting challenge for us as librarians, who have always been at the forefront of Internet development. Many librarians across the country are taking on ambitious marketing projects to promote the services and resources we offer. However, it is increasingly difficult to capture the attention of library users via the World Wide Web without placing a more qualitative emphasis on the design elements of library home pages. But, how do we move beyond the work that has been done already to foster on-line research between librarians and the patrons we serve? Perhaps one way to achieve this is to begin managing the information we provide via the World Wide Web through a frames environment.

Many Web designers can attest that using frames in home pages has always been a double-edged sword. On one hand, frames provide an opportunity to create multiple windows within one World Wide Web browser. On the other hand, frames are not the easiest application to understand or master when writing HTML code. Some of the new HTML editors do provide support for frames. However, many librarians have been frustrated by these editors: pages don't load correctly, they load in the wrong frame, links malfunction, or error messages occur where a frame should be. Having a knowledge of frames-based HTML can quickly remedy many of these problems, making it much easier to write HTML code or to work between an HTML editor and code writing. And viewing different documents while remaining in the basic structure of the library's home page makes the library user's Web experience less confusing.

This article focuses on frames-based concepts of Web design and how librarians can reenergize their home pages to reach out to the newer Web surfers. Using frames-based home pages, Web designers can launch multiple files (or rather windows) into one World Wide Web browser, helping librarians to organize the growing amount of information into a frames environment while making it easier for library users to navigate the Web. It is assumed that readers already have a working knowledge of the HyperText Markup Language (HTML) and the World Wide Web. For those who need a quick refresher course on writing HTML code, see Beverly K. Duval and Linda Main's article from the Winter 1995 issue of Library Software Review or locate the National Center for Supercomputing Applications beginning HTML course on the Internet[1].

<center>Working with Frames</center>

You must first set up a master frames page, which will be the page from which all the other documents link. Think of your frames page as a simplified spreadsheet where there are rows and columns for your information. Instead of inserting data into each cell you create in the rows and columns, you will insert an HTML page (or rather a window of a little more flexibility in how the frames will be presented in different browsers. You would write this the same way except you would use the percent (%) symbol after the values, for example, <frameset cols="60%,*"...>. Experiment with these values to fit your own preferences. **Framespacing=10** tells the browser to allow 10 spaces for margins at the top, bottom, left, and right sides of the frame border, and **frameborder=yes** tells the browser to place borders around the frame's edges. The borders will be the gray vertical and horizontal lines that appear between each frame.

In example 3, the statement <frame name...> lets you name this particular frame. For this demonstration, we will name it "top," being that it is the top row of the frames page. However, when you write your own HTML code, you can name these frames whatever you like. Notice that there is only one frame name in this row, because the next row is going to be divided into two columns (see example 4). The headings **marginwidth=10** and **marginheight=5** are variations of the framespacing tag. They set the frame to have margins of 10 spaces on the left and right side of the frame and 5 spaces on the top and bottom of the frame. **Scrolling=auto** means that, if the physical size of your HTML page can't fit into the size of the frame, the browser will automatically add scroll bars to the bottom and side of the frame. If you prefer not to have scroll bars in your frame, you would write **scrolling=no**. **Frameborder=yes** again places a visual border around the frame's edges. If you want the border to appear, you must do this for each frame. If you decide later that you want a seamless frames page, you would want to change this setting to **frameborder=no**. Experiment to find out which layout works best for you. The **noresize** command is optional. If you haven't noticed before, in a frames page you can click on the border of the frame with your mouse and drag it wherever you want the border to go. Using the **noresize** command disables users of your home page from resizing your frames. This is a particularly good command to use if you are worried about the information in a specific frame not having enough room to be displayed correctly. Finally, **target="top"** tells the browser to place the frame in the area you have already named "top." And **src="frame1.html"** is the source name of the HTML file you are going to place in the top frame.

The next lines of HTML code found in example 4 are for the columns section. If you have one long row at the top of the page with an HTML file in it, then you might want to divide the next

row into two columns (see example 8). Notice that this <frameset...> has two <frame name...> sections, because there are going to be two columns on the page. Just as you declared the size of each row, the HTML tag <frameset cols="155,*"...>defines the size of the columns on the page. The left column will have a pixel size of 155, and the right column will have the value of the remaining space by using the asterisk (*) symbol. Next, you will name each of these columns as **frame name="right"** and **frame name="left"**. Again, for each column you should declare the sizes of the margins, whether you want scrolling bars and borders, the name of the HTML file you will place in each frame, and the target for each of the frames. After you have completed these steps, save this HTML code as the file **test.html**.

You will then need to create three different files to fit into the **test.html** page. Open a new page in your text editor and type the lines of code from example 5. Save this code as the HTML file **frame1.html**.

Do the same with the next examples, saving example 6 as **frame2.html** and example 7 as **frame3.html**. Make sure you save all of these files, including **test.html**, in the same directory on your hard drive, or the browser won't be able to find them. Then open your World Wide Web browser, select the File menu option, choose Open Page, and select the **test.html** file. Your page should look like the picture in example 8. If not, go back through the steps and make sure you didn't leave out any punctuation marks or make any typos.

Now you can really begin to use the frames' capabilities. Let's say you want to put your library logo or title in the top frame (frame1.html). You could change your code from example 5 to read "The Virtual Library" instead of "Top Frame," or you could place an image there instead with the <img src="logo.gif" alt="library title"> tag, where **logo.gif** is the picture file you have created as your library logo. Now you can create an index in the left frame (frame2.html) that will remain on the screen while the information to which you link is presented in the right frame (frame3.html). Open example 6 (frame2.html) in your text editor one more time. Now change the code to that found in example 9.

As you may have already discovered, the secret to navigating in frames is to declare in which window you want the link to send the information. This is done by using the **target="right"** statement inside the <a href=...> tag. We have stated **target="right"** because that is the name of the right frame where the information will appear. Again, you can name the frame whatever you like, but make sure you link to that frame name with the target option. When these files appear in your browser, the URLs will appear in the left column, and the information will be displayed in the right column after clicking on the individual links. If you don't add the **target=**statement, the link will appear in the same frame from which you linked. Save the HTML code in example 9 to the **file frame2.html**. Now open up the **test.html** file in your browser and click on the first link in the left frame. Your screen should appear similar to the picture in example 10. Again, the links on the left will be presented in the right frame once you click on them.

There are times when you will want to be able to leave the frames environment for one reason or another: the window to which you are linking gets too cluttered, you are returning to a parent page within your domain, and so on. If this is the case, you would want to use the **target="_parent"** statement in your <a href=…>tag as is shown in example 11. The

target="_parent" statement is a default HTML command that allows you to leave the test.html frame without launching an additional window as you might using JavaScript. Make sure to put the underscore symbol (_) in front of the word parent, or the command won't work.

You might have been asking yourself, "what does the HTML tag <noframes> mean in example 1?" This tag is optional, but you might want to include it for patrons who are still using text-based World Wide Web browsers such as LYNX. When a text-based browser encounters a frames page, it doesn't know how to display the information. By using the <noframes> tag, you can place links you want people with non-frames-capable browsers to see. When the text-based browser loads the **test.html** page, it will bypass the information in the frames section of the HTML code and go directly to the information found between the <noframes> tags. You can manually type in the series of links you have already included in **frame2.html** between the <noframes> and the </noframes> tags as seen in example 12. Notice also that the <body> tags go between the <noframes> tags, but the last </frameset> tag goes after the last </noframes> tag.

Adding a Bottom Frame

After you have tried the previous examples, you might want to add another frame at the bottom of the screen that gives information about the URL the user is currently seeing. Or, you could add a glossary of terms, a directory of contact numbers for people at your organization, or an index to the links you have in your frames page. In this example, we will place links to some common search engines in the bottom frame. Look at the HTML code in example 13 and compare it to the code in example 1. Notice that in the first <frameset rows...> tag, we have added an additional dimension size. The line now reads <frameset rows=40,* ,40>, where the top row is 40 pixels, the bottom row is 40 pixels, and the columns in between receive whatever pixel size is left for that row, using the asterisk (*) symbol. In this example, we have added an additional frame row after the </frameset> tag for the columns. It must go after the columns tag, or the browser will confuse it as a row within the columns. You will also see the name of this frame is "bottom," and it links to the page **frame4.html**, as seen in example 14.

In example 14, you need to set up your URLs to link to whichever frame you want the information to appear in. In this case, the URLs are set up to link to the right frame. Notice also the use of the ** ** command. This is used to place a single blank space in between each one of the links to keep the frame from being cluttered. When you have changed the HTML code in example 13, save this file as **test.html**. Then, save the code from example 14 as **frame4.html**. Open the **test.html** file in your browser once more, and it should appear like the screen in example 15.

Conclusion

Once you have practiced using a few different frames, you should be well on your way to writing more advanced HTML code and designing high-caliber library home pages. You might also want to experiment with placing graphics files in your frames page. Currently, there are a number of graphics editors and draw programs that can make your Web site look even more professional. Adobe Photoshop, Paint Shop Pro, and Freehand are just a few of these applications, and they are well worth the investment in the long run.[2] With the right knowledge content, enhanced with

frames HTML code, your patrons will quickly recognize your library as one of the movers and shakers in the realm of World Wide Web information management.

Notes

1. Beverly K. Duval and Linda Main, "Microcomputer Applications in the Library: Building Home Pages," Library Software Review 14 (Winter 1995): 218-27; National Center for Supercomputing Applications, "A Beginner's Guide to HTML Home Page," March 12, 1998, at http:llwww.ncsa.uiuc.eduiGeneraliinternetiWWWI.
2. For more information on Photoshop, see Beverly K. Duval and Linda Main, "Adding Images to Web Pages Using Adobe Photoshop 3.0 and Mapedit," Library Software Review 15 (Summer 1996): 88-102. You might also want to download some of the common graphics editors and draw programs by visiting http://www.download.com and http://www.shareware.com. For examples of how you might use a combination of informational links and graphics in a frames environment, see author's home page for the Connecticut Information Technology Institute Library Liaison project at http://www.lib. uconn.edu/-dgilstrap/it1.htm or the Cornell University Library home page at http://campusgw.library.cornell.edu/.

Examples

Example 1

```
<html>
<frameset rows="40,*" framespacing=10 frameborder=yes>
     <frame name="top" marginwidth=10 marginheight=5 scrolling=auto
         frameborder=yes noresize target="top" src="frame1.html">
<frameset cols="155,*" framespacing=10 frameborder=yes>
     <frame name="left" marginwidth=5 marginheight=5 target="left"
         src="frame2.html" scrolling=auto frameborder=yes>
     <frame name="right" marginwidth=5 marginheight=5 src="frame3.html"
         target="right" scrolling=auto frameborder=yes>
</frameset>
<noframes>
<body>
<a href="http://www.lib.uconn.edu">
</body>
</noframes>
</frameset>
</html>
```

Example 2

```
     <frameset rows="40,*"
framespacing=10 frameborder=yes>
```

Example 3

```
     <frame name="top" marginwidth=10
marginheight=5 scrolling=auto
     frameborder=yes noresize
target="top" src="frame1.html">
```

Example 4

```
<frameset cols="155,*"
framespacing=10 border=0>
<frame name="left" marginwidth=5
marginheight=5 target="left"
    src="frame2.html" scrolling="auto"
frameborder=0>
<frame name="right" marginwidth=5
marginheight=5
    src="frame3.html" target="right"
scrolling=auto frameborder=0>
```

Example 5-7

Example 5	Example 6	Example 7
<html> <body> Top Frame </body> </html>	<html> <body> Left Frame </body> </html>	<html> <body> Right Frame </body> </html>

Example 8

Example 9

```
<html>
<body>
Click on the links below to access individual library pages!<br>
<a href="http://www.lib.uconn.edu/~dgilstrap/it1.htm"
        target="right">CITI Homepage</a><br>
<a href="http://www.lib.uconn.edu" target="right">UConn
        Libraries</a><br>
    <a href="http://www.sagepub.co.uk/journals/usdetails/j0016.html"
 target="right">
    Libr Soft Rev</a><br>
    </body>
    </html>
```

Example 10

Example 11

```
<a href="http://www.uconn.edu"
target="_parent">
```

Example 12

```
<noframes>
<body>
Click on the links below to access individual library pages!<br>
<a href="http://www.lib.uconn.edu/~dgilstrap /it1.htm">CITI Pages</a>
<a href="http://www.lib.uconn.edu">UConn Libraries</a>
<a href="http://www.sagepub.co.uk/journals/usdetails/j0016.html">
Library Software Review</a>
</body>
</noframes>
```

Example 13

```
<html>
<frameset rows="40,*,40" framespacing=10 frameborder=yes>
<frame name="top" marginwidth=10 marginheight=5 scrolling=auto
        frameborder=yes noresize target="top" src="frame1.html">
<frameset cols="155,*" framespacing=10 frameborder=yes>
<frame name="left" marginwidth=5 marginheight=5 target="left"
        src="frame2.html" scrolling=auto frameborder=yes>
<frame name="right" marginwidth=5 marginheight=5 src="frame3.html"
        target="right" scrolling=auto frameborder=yes>
</frameset>
<frame name="bottom" marginwidth=10 marginheight=5 scrolling=auto
        frameborder=yes noresize target="bottom" src="frame4.html">
<noframes>
<body>
<a href="http://www.lib.uconn.edu">
</body>
</noframes>
</frameset>
</html>
```
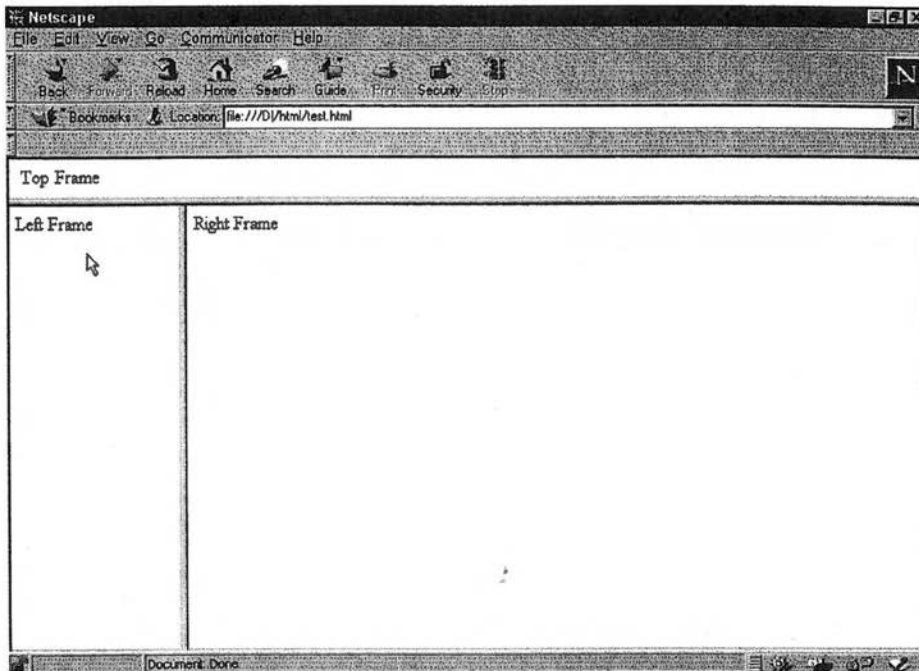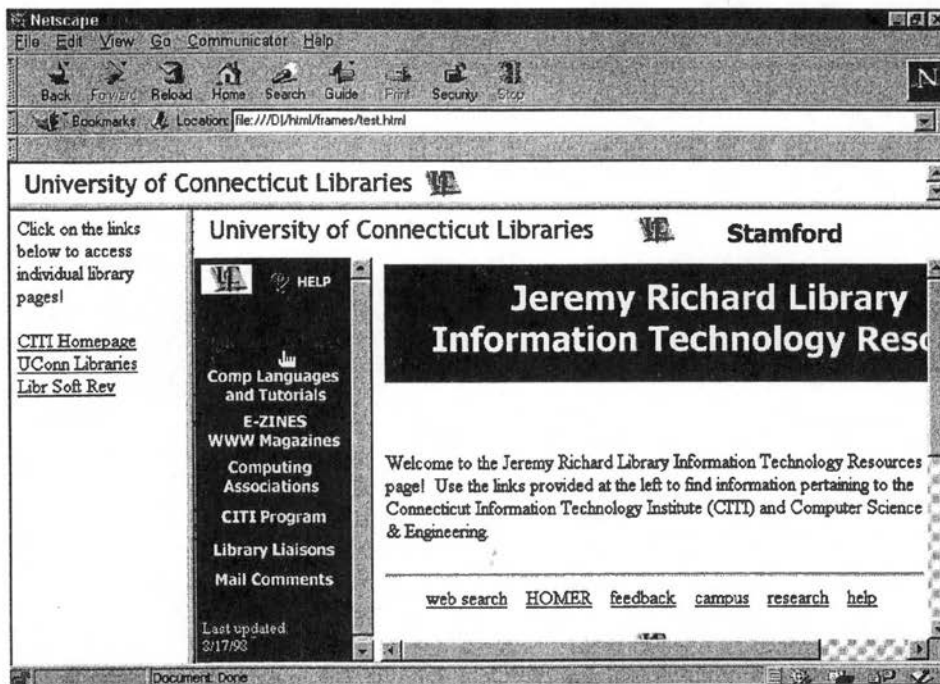
Example 14

```
<html>
<body>
<center>
<a href="http://www.altavista.digital.com" target="right">AltaVista</a>
       
<a href="http://www.infoseek.com" target="right">Infoseek</a>
       
<a href="http://www.northernlight.com" target ="right">Northern Light</a>
       
<a href="http://www.yahoo.com" target="right">Yahoo</a>
</center>
</body>
</html>
```

Example 15