

Web Page Creation II

Advanced HTML



**Utah
Education Network**

Staff Development

Created by: Ray Dahl
Revised: June 2, 2000
Revised by: Jessica Anderson

Table of Contents

Introduction	7
Review of Basic Web Page Creation	7
Working between the Browser and Editor	7
Saving Files	8
A Note on Standards	8
Section 1 - HTML Review	9
HTML The language of the Web	9
Activity	11
Section 2 - The <META> Tag	13
The Meta Tag	13
Indexing your site with the Meta tag	13
Automating Requests of the Server	16
Activity	19
Section 3 - Building a Frameset	21
What is a Frame	21
Welcome to the Suburbs	21
Virtual Surveying	22
Activity.....	26
Section 4 - Navigation in Frames	27
Navigating with Frames	27
Making it All Work	27
Magic TARGET Names	28
Activity	29
Section 5 - Creating a Form	31
Front End	31
Back End	31
The Form Tag	32
The Input Tag	33
The Select Tag	36
The Textarea Tag	37
Activity	39

Section 6 - Image Maps	41
Finding the Coordinates	43
Putting it all together	46
ImageMap Design Issues	47
Activity	48
Appendix A - HTML 4.0	49
HTML 4.0 and you	49
Three Versions of the Standard	49
Unappreciated HTML	50
Deprecated Tags	51
Appendix B - Java Script	53
Using JavaScript	54
Sample JavaScript	56
Appendix C - Web Resources	63
Appendix D - Web Accessibility	65



Web Page Creation II Agenda

Instructor: _____

Date: _____

Phone: _____

E-mail: _____

- _____ 1. Preview the Manual.
- _____ 2. Basic HTML Review
- _____ 3. Activity 1 - Build a basic web page.
- _____ 4. The <META> Tag
- _____ 5. Activity 2 - Add Meta Information to your web pages.
- _____ 6. Frames
- _____ 7. Activity 3 - Build a frameset.
- _____ 8. Navigation in frames.
- _____ 9. Activity 4 - Adding links to the frameset.
- _____ 10. Forms
- _____ 11. Activity 5 - Build a form for your web site.
- _____ 12. Image Maps
- _____ 13. Activity 6 - Create an Image map.
- _____ 14. Fill out a class evaluation.
- _____ 15. Complete and turn in homework.



Advanced HTML Introduction

Web Page Creation: Advanced HTML class is an extension of the Web Page Creation: Basic HTML class that introduces the basics of creating documents for the World Wide Web using Hyper Text Markup Language (HTML). This course introduces some of the more advanced HTML tags and tools that can be used to improve the quality of the pages you create. This course will also introduce other resources that exist for creating greater interactivity in your web site.

Review of Basic Web Creation Concepts

The basic tools of web page creation are a text editor, a web browser, knowledge of HTML and creativity. One of the most important things to remember with writing HTML is keeping the code simple and straightforward. Many tags can be nested within others to create particular effects. Remember write the code clearly so you will be able to de-bug it later. In general, use HTML tags to add impact or value to information. Complex compositions seldom delight anyone but their creators. Keep things simple so that others can appreciate your skills.

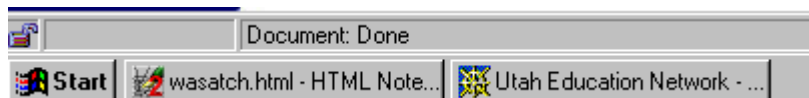
Working between the Browser and the Editor

When you are creating your page you need to have the text editor and the web browser open so that you can preview your work as you create it. There are several different ways to navigate between the windows. The keyboard command is **ALT + Tab**.

Another method is to use the **minimize button**. Use the minimize button to move the text editor out of the way when you are previewing and minimize the browser when you are writing your code.



The easiest way to move between the two programs is by using the start bar at the bottom of the screen.



Be sure to leave both running this will make it easier and faster to move between the programs.

Note: Be sure to save your work before you preview it. The web browser will only display the saved information. To view your changes hit the "Reload" button.

Macintosh computers are equally easy to work with. The easiest way to move between the two programs is to leave a portion of both windows visible and click between them. The Chooser is another equally easy way to shuffle between the programs that you might be using.

Saving Files

HTML files are text files but in order for your browser to recognize the file it must be saved with the correct file extension. In the world of Windows 95, Macintosh and UNIX the .html extension is the norm. In DOS and Windows 3.x the .htm is the valid extension. For a file to be viewed as a web page one of these two extensions must be used.

A Note on Standards

The web is continually evolving to incorporate new media types and capabilities. In order to keep things from getting to disorganized the World Wide Web Consortium (W3C) was established to maintain standards for how HTML should function. This course introduces many concepts that are part of the HTML 4.0 standard. In general version 4.0 browsers and newer support the techniques that you will learn in this course. If your target audience is not using a version 4.0 browser you will want to use the HTML 3.2 standard.

The following links will help you get a better understanding of what the standards are and how best to use HTML.

World Wide Web Consortium - www.w3.org

W3C HTML 3.2 Standard - www.w3.org/TR/REC-html32.html

W3C HTML 4.0 Standard - www.w3.org/TR/REC-html40/

Rob Schluter's HTML Index - www.erinet.com/martyh/html



HTML Review

Section 1

HTML - The Language of the Web

Hyper Text Markup Language is essentially a word processing language used to format text and images for presentation on the web. HTML is a cross platform language that can be used on any type of computer. The language consists of tags or codes used to describe how the text should be displayed.

Tags come in two basic categories, open and closed. A tag consists of the < and > brackets with an HTML command in side. For instance, <BODY> denotes the body section of your web page. Some tags are open or only use one tag to accomplish formatting of the page. is one such tag. This is used to identify where an image needs to be placed. There is no need to say the image starts at this point and ends at another because the image file defines its own limits.

A closed tag is one that has a starting point and an ending point. For instance tells the browser to display the enclosed text with a bold typeface. The closing tag tells the browser to stop bolding the text and return it to the previous size. The forward slash "/" is the ending mark for the tag. When writing your code, be sure to close your tags if they require a back half. Unclosed tags will apply their characteristics to the rest of your page.

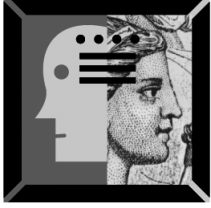
Attributes are ways of describing tags. In language terms a tag is the HTML equivalent of a noun and an attribute, an adjective. Attributes are included as part of the tag.

In this example the tag is the image "picture.jpg." ALIGN is the attribute and is describing how to position the image relative to text in the document.

Note: Attributes don't need to be closed. They are part of the tag and must be contained in the opening portion.

The following table is a list of basic HTML tags and a brief description of how they are used.

Tag	Description
<!--....-->	Comment - This tag can be used to add comments to your page. The browser ignores any text that shows between these tags.
<HTML>...</HTML>	HTML - This tag identifies the starting and ending of an HTML document. Very important!
<HEAD>...</HEAD>	Head - Identifies the beginning and ending of the HTML document's head. Important!
<BODY>...</BODY>	Body - Identifies the beginning and end of the body of an HTML document. Important!
<TITLE>...</TITLE>	Title - Text inside this tag provides the title for the entire document.
<H#>...</H#>	Heading - This is formats the section heading. "#" represents a possibility of 1 - 6. One is the largest size and 6 the smallest. The text displays in a bold face with two carriage returns following the text.
 	Break - Forces a line break in the on screen text display.
<PRE>...</PRE>	Pre - Preserves the original layout of preformatted text. The display is limited to the Times font.
<HR>	Horizontal Rule - This command draws a horizontal line across the screen.
	Image - Inserts an image into the document.
...	Bold - Applies a bold face to the text inside the tags.
<I>...</I>	Italics - Applies an italic face to the text inside the tags.
<TT>...</TT>	Typewriter Text - Displays the text inside the tags in a typewriter font.
	List Item - Marks a member item within a list of any type.
...	Ordered List - Marks a numbered list of elements.
...	Unordered List - Marks a bulleted list of elements.
<A>...	Anchor - Defines a link to another location in the document or another page.
<BLOCKQUOTE>...</BLOCKQUOTE>	Blockquote - Creates a double indented text block for longer quotes.

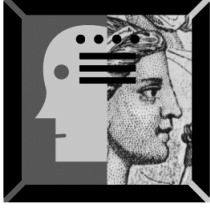


Build a Web Page Activity 1

Build a web page similar to the one displayed below.
Include in your design some of the following:

- School name or organization masthead.
- Contact information.
- Links to other sites.
- Email links
- Images

An alternate activity would be to continue working on a web page that you have already created.



The <META> Tag

Section 2

The <META> Tag

The <META> tag is a unique tag that lives in the HEAD area of an HTML document. This tag is a behind the scene player that provides information about the HTML document or makes automated requests of the server.

The <META> tag is a flexible tag that can be defined using two different attributes, NAME and HTTP-EQUIV. The two attributes can be further defined using the CONTENT attribute to provide a broad range of information or perform various functions. For example:

```
<META name="something" content="something else">  
<META http-equiv="something" content="something else">
```

Note that you don't have a </META> it is not needed. Unlike other tags that can have multiple attributes assigned to an individual tag, the META tag can only be defined in one form per use of the tag. The tag can be used multiple times in the HEAD of the document. The remainder of this section is devoted to showing some of the common uses of this tag.

Indexing your site with the <META> tag.

One of the most common uses of the <META> tag is to index a site for a search engine. The <META> tag is used in the head of an HTML document and gives information to search engines that isn't necessarily included in your web page. The information contained in the <META> tag can help search engines more accurately index your site.

Most search engines gather all the words on a web page and index them accordingly by the number of references on the page. This creates problems for pages that use frames or where the words are graphics on the page (Photoshop graphics with text in them for example). The search engine only sees the ASCII text and these graphical pages won't be represented in the searches. To get around this, several popular search engines have made use of the <META> tag. The <META> tag allows web page developers to index their own web sites by keywords. Not all search engines use the <META> tag, for example Altavista does, but Excite does not. The <META> tag is not a complete cure-all, but you won't miss out by putting the tag in, and it may work well in some cases.

How it all works

The <META> tag has attributes like other HTML tags. These attributes define properties of the <META> tag. There are two main attribute types used when trying to create an index for your site, DESCRIPTION and KEYWORDS.

<META name="description" content="a description of your page">

Depending on the search engine, the description will be displayed along with the title of your page in an index. The "content" could be a word, sentence or even paragraph to describe your page. Keep this reasonably short, concise and to the point. Make sure your description is an appropriate reflection of the page's content. You will want your description to be different than your page title.

For example:

<META name="description" content="UtahLINK. Linking information and Utah schools">

<META name="keywords" content="a, list, of, keywords">

Choose keywords you think are appropriate, separated by commas. Remember to include synonyms, Americanisms and so on. So, if you had a page on cars, you might want to include keywords such as car, cars, vehicles, automobiles and so on.

For example:

<META name="keywords" content="utahlink, school resources, lesson plan, utah schools, education">

Based on our examples when someone searches for utahlink, public, education, lesson, and plan the UtahLINK site will probably be toward the top of the list.

Search in and Display

Tip: To find a bed-time story: "fairy tale" +frog -dragon

The search engines will index the UtahLINK web site under the keywords listed in the keyword content attribute and display the UtahLINK home page as one of the top sites.

Notice how the HTML was written. The description and keywords were put into separate <META> tags. This goes contrary to most HTML syntax. Usually attributes are simply added inside the tag. However in this special case, the content attribute requires a name. In order for the name and content to be matched, they need to be inserted inside their own, unique <META> tag.

What Searches Return

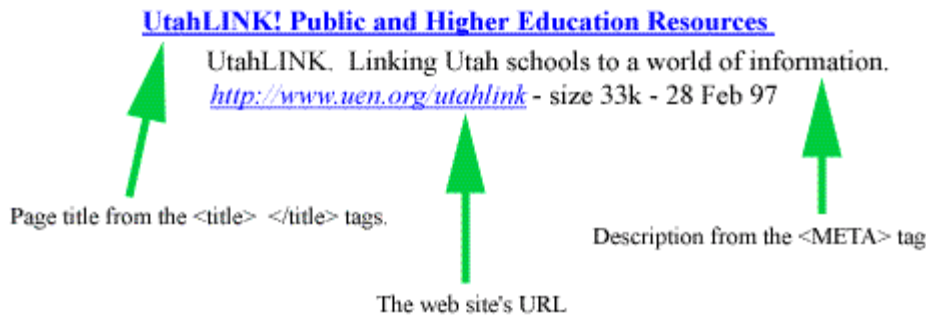
If we performed the search for UtahLINK shown earlier, we might find results like:

[UtahLINK! Public and Higher Education Resources](#)

UtahLINK. Linking Utah schools to a world of information.

<http://www.uen.org/utahlink> - size 33k - 28 Feb 97

The two underlined sections are links to the web page. The text on the middle line is the description in the <META> tag. The size of the file (33K) and the date last revised are listed on the bottom line.



Abusing the System

Some sites get carried away with keywords in the <META> tag. They put in the names of all the states and other words that attract attention but don't have anything to do with the site. Some search engines have started screening for these types of sites. If a word appears in the keywords more than three times, it's automatically dropped from some search engines. This can be a big problem if that keyword is the main search word for your site. Make sure to include the important words but don't overuse them. They may get dropped.

Below is a list of search engines and their use of the <META> tag for indexing.

The most up-to-date version can be found at

<http://searchenginewatch.com/webmasters/features.html>

Search Engine	Alta Vista	Excite	HotBot	InfoSeek	Lycos	Open Text	Web Crawler
Meta Tag Support	Yes	No	Yes	Yes	Yes & No*	No	Yes
Description	Meta tag, or first few lines on page	Sentences grouped by concept; most dominant sentences extracted	Meta tag, or first few lines on page	Meta tag, or first 200 characters after <body> tag	Created based on content	First 100 characters on page	Meta tag, or first 275 characters after <body> tag

<META> Tags Won't Solve All Your Problems

The <META> tag won't magically move your page from the bottom of the search engine index to the top but it can help in some situations. It's a nice tool for SOME search engines but not all use the <META> tag.

<META> tags are not a miracle solution.

Since different search engines use the <META> tag in different ways, there's no sure-fire way to put your web site at the top of the search engine list. However, with careful planning and a little research, it's possible to make sure people looking for your site will be able to find it.

Make sure the title of the page accurately describes the content. If the web page has a descriptive title that people might enter into a search, it'll be closer to the top.

Automating Requests of the Server

Another common use for the <META> tag is to request that the server do something automatically. With the <META> tag you can ask the server to do such things as reroute the user to another URL or reload the page with in a certain amount of time

Automatically Loading New Pages

The <META> tag can be used to move a user automatically from one page to another. This feature is used commonly when a web address changes. The user loads the old page and a message appears.

This page has changed location. Please update your bookmarks.

<http://www.anywhere.com/anything.html>

You may click on the link or wait and you'll be taken there automatically.

The page waits for the user to write down the new link and then automatically connects and downloads the home page at the new location.

This is called a meta client pull. In the <META> client pull, the browser loads, waits a specified number of seconds, and loads and new web address.

```
<META HTTP-EQUIV="refresh" content="0; url=homepage.htm"
```

This can be used in the HEAD section of the index.html file to redirect it to homepage.htm. The figure after content is the time in seconds that the browser waits before moving on.

The amount of time needed before rerouting the page is something that may need some experimentation. It is a good idea to test your page on a 28.8k modem or slower to make sure that all the information loads and the user has time to read the message before the page is automatically rerouted.

The HTML code below is an example of loading a new page and belongs in the head of the document.

The <META> tag can also be used to load a new page after a set number of seconds. This can be very useful when pages change servers (and invariably change URLs).

This page has changed location. Please update your bookmarks.

<http://www.anywhere.com/anything.html>

You may click on the link or wait and you'll be taken there automatically.

Here's the HTML for the auto-loading page above.

```
<TITLE>We've Moved</TITLE>

<META HTTP-EQUIV="Refresh" CONTENT="5;
URL=http://www.uen.org/utahlink">

</HEAD>
<BODY>
<H3 align="center">This page has changed location.
Please update your bookmarks.</H3>
<P>
<A HREF = "http://www.anywhere.com/anything.html">
http://www.anywhere.com/anything.html</A></P>
```

Auto Reload

A simple use of client pull is to cause a document to be automatically reloaded on a regular basis. During the NBA playoffs, nba.com and other sports sites have score that update every minute automatically. They do it with the <META> tag.

```
<TITLE>Auto Re-loading Page</TITLE>

<META HTTP-EQUIV="Refresh" CONTENT=5>

</HEAD>
<BODY>
<h1>Isn't this fun?</h1>
<P>This page will refresh every 5 seconds because the
content attribute was set equal to 5...
```

Here's a full example of HTML for an auto-reload page.

```
<HTML>
<HEAD>
<TITLE>Reload</TITLE>
<META HTTP-EQUIV="Refresh" CONTENT="5">
</HEAD>

<BODY>
<H3 align="center">Auto reload every 5 seconds</H3>
<P>
This page will auto reload to give you the most up to
date information.
</P>
</BODY>
</HTML>
```

The most common <META> tags have been covered in this section, but there are many other <META> tags that can be included inside of your head tag. There are several web sites that are good resources to find additional <META> tags. We suggest using the sites listed in the activity.



Add Meta Information Activity 2

Add Keyword and Description information to your web page(s).

Meta Resources

Build your own Meta Tag wizard.

<http://vancouver-webpages.com/META/mk-metas.html>

<http://www.websitepromote.com/resources/meta/>

Good Meta Definitions

<http://WDVL.com/Authoring/HTML/Head/Meta/>

<http://www.philb.com/metatag.htm>

http://www.webdeveloper.com/html/html_metatags.html

How to promote your site

<http://WWW.WDVL.com/Location/Promotion/>

<http://WWW.WDVL.com/Location/Promotion/Search/>

How to announce FAQ

<http://ep.com/faq/webannounce.html>

Search Engines and Meta tags.

<http://www.stars.com/Location/Meta/Tag.html>

<http://searchenginewatch.com/webmasters/features.html>



Building a Frameset

Section 3

What's in a Frame

Frames have been around the web for a while. They are web pages that are divided into sub-windows, with each window displaying a separate web page. Frames are very similar to the large picture frames that can display multiple pictures all for viewing at the same time. Picture frames like this can be very fun when they display pictures representing important times in ones life. These same picture frames can be a confusing collection of images if there isn't some form of organization or relationship between the images.

Frames can allow web pages to function more efficiently by having a navigation bar or site masthead available at all times. When used well frames can make a good web site even better. Poorly used a good site can become an atrocity.

Accessibility and Frames

Frames are a topic of great debate in the Web Design community. For several years the use of frames was not recognized by the World Wide Web Consortium (www.w3.org) as a standardized method of web page creation. HTML 4.0 includes a recommendation for the use of Frames as standardized method of design. Shortly after HTML 4.0 was introduced the Web Accessibility Initiative (WAI) became a working draft. This document includes recommendations on how to design web pages that offer greater accessibility to individuals with a wide range of disabilities. Part of the WAI discusses the use of frames and notes their use as causing potential access problems. For more information on the Web Accessibility Initiative visit the World Wide Web Consortium web site at www.w3.org . Additional information is available on this topic in Appendix D of this document.

Welcome to the Suburbs

Frames create some very interesting design challenges. You have limited screen real estate to begin with. Tool bars and scroll bars eat a portion of the available screen space so that you are limited to about 600x400 pixels of working space on an average sized monitor, slightly less on a small Macintosh monitor.

When frames are added to the design, the screen is further subdivided. If the information you are trying to display exceeds the size lot you have designated, scroll bars are added to the display so that the viewer can move around in the frame to see all that is available. Unfortunately when the browser adds scroll bars less of the screen is available to display information, compounding the problem. Careful planning must be used to avoid the maze of scroll bars that can develop if a page is poorly organized.

How to Build a Better Frame

A web page that utilizes frames requires a great amount of planning. Frames are similar to tables in how they are organized and displayed. The primary difference is, with a table you are working with cells that contain data. With frames you are working with “cells” that contain different web pages.

The Shape of Things

When a *frameset* is organized, you are required to identify how the page will be divided. The first part that needs to be identified is how many rows, columns or combination of the two will be created. For instance the following example can be described as two horizontal rows, with the second row divided into two columns.

HEADING	
Index	Page Content

The next part of the puzzle is to identify the proportions of the frames. This can be done proportionally or literally. Proportional measurements are represented in percentages. Literal measurements are noted in pixels.

Represented proportionally the previous example could be described: Row 1 is 30% of the height of the screen, Row two is 70% of the height of the screen. Row 2 is divided into two columns. Column 1 is 30% of the width of the screen with Column 2, 70% of the screen.

A literal representation would read something like this: Row 1 is 75 pixels high, Row 2 will fill the balance of the screen. Row 2 is divided into two columns, column 1 is 150 pixels wide, column 2 will fill the balance of the screen.

Virtual Surveying

When you embark on creating a frame page determine what information will be placed in which frame and how much space each item will require. Once the required space is calculated you can use those values to define your frames. Be




sure to give yourself a modest buffer to account for the way different browsers display information.

Using literal measurements for your frame sizing will ensure that the important information will be displayed in uninterrupted form. This can eliminate many of the annoying scroll bars that clutter so many frame style pages.



Note: When defining frames it is only necessary to define the dimension of one frame the browser will assign all of the remaining space to the undefined frame.

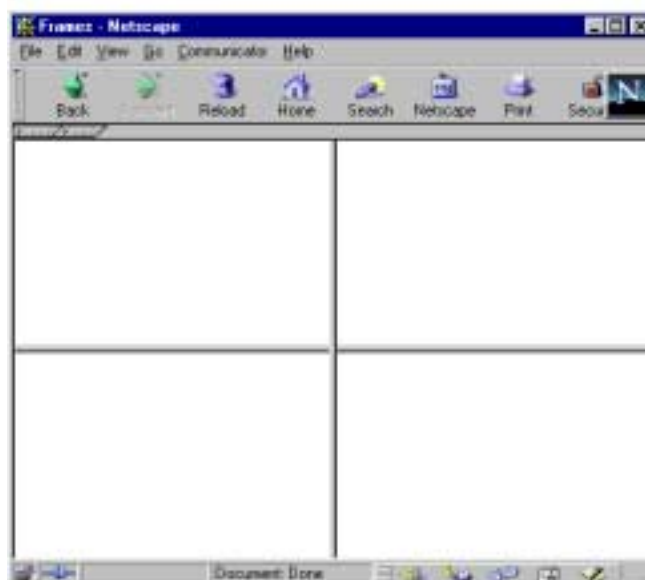
The <FRAMESET>...</FRAMESET> Tag

 <frameset>	<FRAMESET> is the tag used to establish a page that will be using frames. This tag replaces the <BODY> tag used in more conventional designs.
 rows	ROWS establishes that a frameset will be divided into horizontal rows.
 cols	COLS establishes that a frameset will be divided into vertical columns.









Below is an example of an HTML document that contains frames.

```
<HTML>
<HEAD>
<TITLE>Frames</TITLE>
</HEAD>
<FRAMESET ROWS="50%, 50%" COLS="50%, 50%">
</FRAMESET>
</HTML>
```

This page will look like:



The <FRAME> Tag

 <p><frame></p>	<p>This tag is used to identify the document displayed inside each frame of the <frameset>. It has six possible attributes.</p>
 <p>src</p>	<p>SRC="url"</p> <p>The SRC attribute identifies which HTML document will be displayed inside the frame. If this attribute is not applied the frame will be empty.</p>
 <p>name</p>	<p>NAME="window_name"</p> <p>The NAME attribute is used to identify a specific frame within a <frameset>. This attribute is applied so that other documents may be targeted to that frame. Names must start with an alpha numeric value, all other values will be ignored.</p>
 <p>frameborder</p>	<p>FRAMEBORDER="0" or "1"</p> <p>1 means a border a border will be displayed between this frame and the other frames. This is the default. 0 means no border will be displayed.</p>
 <p>scrolling</p>	<p>SCROLLING="yes no auto"</p> <p>The SCROLLING attribute defines whether a frame should have a scroll bar. The default is auto, where the computer decides if a scroll bar is needed.</p>
 <p>marginwidth</p>	<p>MARGINWIDTH="value"</p> <p>The MARGINWIDTH attribute is used to determine the width of margins in a frame. If specified, the value is expressed in pixels. The MARGINWIDTH attribute is optional.</p>
 <p>marginheight</p>	<p>MARGINHEIGHT="value"</p> <p>The MARGINHEIGHT attribute is similar to MARGINWIDTH except it controls the upper and lower margins instead of the left and right margins.</p>
 <p>noresize</p>	<p>The NORESIZE attribute has no value. It is a flag that indicates that the frame is not resizable by the user.</p>

Framesets can be nested together to make more complex pages with multiple rows and columns. It is difficult to make a good site with three or more frames. So be cautious in how you design framesets and the type of information you want to display.

The HTML for a page using rows and columns will look like this.

```
<html>
<head>
<title>Rows and Columns</title>
</head>
<frameset cols="200,*" rows="*">
  <frame src="navlist.html" name="nav">

</frameset>
<frameset rows="150,*" rows="*">
  <frame src="masthead.html" name="header">
  <frame src="intro.html" name="info">
</frameset>
</frameset>
</html>
```

The HTML document displayed in a browser should look like this:

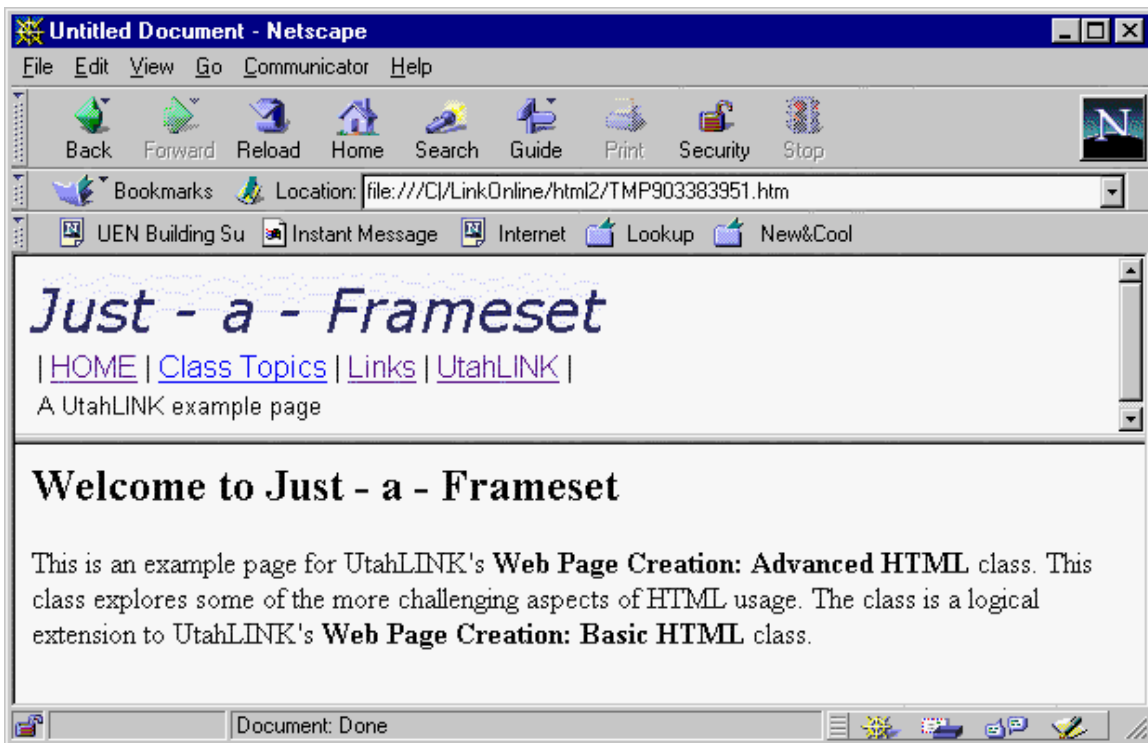




Build a Frameset

Activity 3

Build a Frameset. The Assignments diskette has pages already to load into the frameset or add some of your own to the frameset. Your page should look something like the following.









Navigation in Frames

Section 4

Navigating with Frames

Simplifying navigation around a site is the primary use of frames. Building a basic frameset is an easy task but putting the frames to work for you can be a little more challenging. This calls for a slightly complicated interaction between the Frameset and the Anchor tag.

 <a>	Links are created with the anchor <a> tags. The <a> is not a standalone tag. It needs attributes to work. Anchor attributes are discussed in this section.
 href	href = "http://www.uen.org/utahlink" The 'href' stands for hyper reference. Href is followed by an equal sign and the address where the link connects enclosed in quotation marks. The front half of the tag is then closed off with >.
 name	The name is set equal to some descriptive text enclosed in quotes. The text in the quotes can be almost anything. Then the anchor is closed with an tag. This is used to name a location within a webpage.
 target	Target is used to assign a frame for the link to open into. The value of the target attribute in the value of the name attribute used in the <FRAME> tag.

Making it All Work

The target attribute plays a large role in making the navigation of a frameset functional. Without out proper targeting the link you select will open inside the frame of the link. Most navigation frames are too small for effectively displaying a full web page so it is important to target a large enough frame to display the content of the web page.

Sometimes a web page just doesn't display well within a frameset. In a situation like this is best to tell the browser to display the linked page in a new window. To accomplish this an additional command needs to be added to the to the link address.

Magic TARGET names

These names all begin with the underscore character. Any targeted window name beginning with underscore which is not one of these names, will be ignored.

Value	Description
<code>_blank</code>	This target will cause the link to always be loaded in a new blank window.
<code>_top</code>	This target makes the link load in the full body of the window. It is useful for breaking out of an arbitrarily deep FRAME nesting.

Examples:

This is a typical form for a link in a frameset. In this case the link is telling the browser to open in a frame called "view."

```
<A href="newpage.html" target="view">My New Page</A>
```

In this case the link will open the linked document into a blank browser window.

```
<A href=http://www.newsite.com/index.html _blank>New Site</A>
```



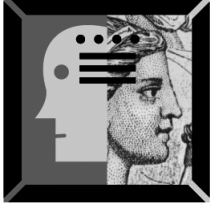
Adding Links to Frames

Activity 4

Take the existing frameset and add a link to an external web site that uses the “_blank” target and add another page that uses the “_top” target.

Online Help

http://home.netscape.com/assist/net_sites/frames.html is Netscape's online tutorial on how to make and use frames.



Creating a Form

Section 5

Forms are popping up all over the Internet. They ask for names, e-mail addresses, and many other things. Some forms have pull down menus and some have checkboxes and radio buttons. Some forms may have all three. Forms are tricky because they involve two different processes, the front and back ends.

Front End

The front end of a form is the HTML. The tags and attributes tell the browser how to display the different elements of the tag. They set up the page and tell the browser how to package up the information and send it off to a computer to be processed.

Back End

The back end of the form is the complicated part. When the information (anything submitted in the form) is sent out of the browser it looks like garbage. The file which is sent is difficult to read because of its formatting. The best way to make sense of the form is to use a small computer program that takes the information, parse it, and put it in an understandable format. The back end is a program written in a programming language like PERL, C, or Applescript. This program is called a CGI script and you guessed it, a computer programmer writes it. There are pre-written CGI scripts available on the Internet but they can require some pretty intense manipulation to get them to work correctly.

If you do not have access to a computer program the form can be submitted to an email address. The email will be difficult to read and may need to be cleaned up in a word processor using the find and replace function. Obviously this is the more difficult and time-consuming way to handle the forms but for some it may be the only option.

Half is better than nothing, isn't it?




In this class, we'll only be dealing with the front end of the HTML forms. There are many resources on the web for CGI scripts but knowledge of C, PERL, or Applescript is essential to understand them.

One resource that is available to help on the hard part is a program called WebForms 2.7. This is a program designed to help create a form and then generate the Perl Script for you. WebForms is available from Q&D Software Development at:
<http://www.q-d.com/wf.htm>.

The <FORM> Tag







The FORM tag specifies an input form or “fill-in-the-form” within an HTML document. More than one form can be placed in a single document, but forms cannot be nested with in other form elements.


```
<FORM ACTION="the-url-for-the-action" method="post">
</FORM>
```

	<p>The FORM tag is used to place a form within a web page. FORM fields are defined using INPUT, SELECT and TEXTAREA.</p>
<p>action</p> 	<p>ACTION = “some-url or email address” ACTION destination where the submitted form will go. It can either be a URL of the query server to which the form contents will be submitted; or an email address (mailto:someone@someplace). If this attribute is absent, then the current document URL will be used.</p>
<p>method</p> 	<p>METHOD = “post” is the HTTP/1.0 method used to submit the form to a query server. Which method you use depends on how your particular server works; we strongly recommend use of POST.</p> <p>POST -- this method causes the form contents to be sent to the server in a data body rather than as part of the URL.</p>

Inside a FORM you can have anything except another FORM. Specifically, INPUT, SELECT, and TEXTAREA tags are used to specify interface elements within the form.

The <input> Tag

	<p>The INPUT tag is used to specify a simple input element inside a FORM. It is a standalone tag; it does not surround anything and there is no terminating tag -- i.e., it is used in much the same way as IMG.</p>
<p>type</p> 	<p>Type = "text" TYPE must be one of: "text" (text entry field; this is the default) "password" (text entry field; entered characters are represented as asterisks) "checkbox" (a single toggle button; on or off) "radio" (a single toggle button; on or off; other toggles with the same) "submit" (a pushbutton that causes the current form to be packaged up into a query URL and sent to a remote server) "reset" (a pushbutton that causes the various input elements in the form to be reset to their default values)</p>
<p>name</p> 	<p>name = "example" NAME is the symbolic name (not a displayed name -- normal HTML within the form is used for that) for this input field. This must be present for all types but "submit" and "reset", as it is used when putting together the query string that gets sent to the remote server when the form is submitted.</p>
<p>value</p> 	<p>value = "work" VALUE, contains the initial value displayed to users. For a text or password entry field, can be used to specify the default contents of the field. For a checkbox or a radio button, VALUE specifies the value of the button when it is checked (unchecked checkboxes are disregarded when submitting queries); the default value for a checkbox or radio button is "on". For types "submit" and "reset", VALUE can be used to specify the label for the pushbutton.</p>
<p>checked</p> 	<p>checked CHECKED (no value needed) specifies that this checkbox or radio button is checked by default; this is only appropriate for checkboxes and radio buttons.</p>
<p>size</p> 	<p>size = 30 SIZE is the physical size of the input field in characters; this is only appropriate for text entry fields and password entry fields. If this is not present, the default is 20. Multi-line text entry fields can be specified as SIZE=width,height; e.g. SIZE=60,12. Note: the SIZE attribute should not be used to specify multi-line text entry fields now that the</p>

	TEXTAREA tag is available.
maxlength 	maxlength = 10 MAXLENGTH is the maximum number of characters that are accepted as input; this is only appropriate for text entry fields and password entry fields (and only for single-line text entry fields at that). If this is not present, the default will be unlimited. The text entry field is assumed to scroll appropriately if LENGTH is greater than SIZE.

A Note on Name-Value Pairs

As a general rule each field on your form needs to have a distinct name. This is so the person or program that receives the information will know what kind of information is submitted with each field. The name attribute describes what kind of information is in the field. The value portion of the equation is the data that your guest enters into the field.

Radio Buttons are the exception to that rule. Radio buttons work in sets. Each set of buttons must have a name and each button must have a distinct value. Look at the following HTML example to see this in practice.

```

form.html - Notepad
File Edit Search Help
<!-- This is just an excerpt from the sample page -->
<form method="post" action="mailto:rdahl@uen.org" name="justaForm">
Name:<input type="text" name="name" size="40"><br>
E-mail:<input type="text" name="email" size="45"><br>
Phone:<input type="text" name="phone">
Fax:<input type="text" name="fax">

<h3>Build Your Sandwich:</h3>

<b>Pick a bread:</b>
  <select name="bread">
    <option>Sour Dough</option>
    <option>Wheat</option>
    <option>Rye</option></select>

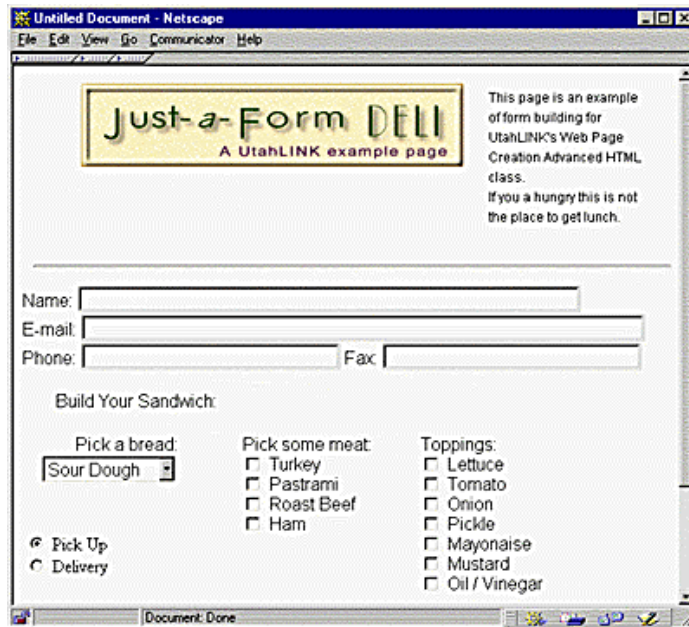
<input type="radio" name="service" value="pickup" checked>Pick Up <br>
<input type="radio" name="service" value="deliver">Delivery</td>

<td width="157" rowspan="3" valign="top"> <b>Pick some meat:</b><br>
<input type="checkbox" name="meat" value="turkey">Turkey<br>
<input type="checkbox" name="meat" value="pastrani">Pastrani<br>

<input type="submit" name="submit" value="Make It!">
<input type="reset" name="submit2" value="Start Over">





```

Here is how the HTML appears in the browser.



The <select> Tag

The <select> tag goes inside the form and offers choices for the person viewing the page much like multiple choice exams.

<p><select></p> 	<p>The SELECT tag allows users to select one or more items from a list generally displayed in a pull down menu. The items are generated using the <OPTION> tag. The select tag is put between the <FORM> ... </FORM>, any number of SELECT tags are allowed.</p>
<p>name</p> 	<p>name = "occupation" NAME is the symbolic name for this SELECT element. It's the name given to the output of this field.</p>
<p>size</p> 	<p>size = "5" If SIZE is 1 or if the SIZE attribute is missing, by default the SELECT will be represented as a Motif option menu. If SIZE is 2 or more, the SELECT will be represented as a Motif scrolled list; the value of SIZE then determines how many items will be visible.</p>
<p>multiple</p> 	<p>multiple MULTIPLE, if present (no value), specifies that the SELECT should allow multiple selections (n of many behavior). The presence of MULTIPLE forces the SELECT to be represented as a Motif scrolled list, regardless of the value of SIZE.</p>





Unlike INPUT, SELECT has both opening and closing tags. Inside SELECT, only a sequence of OPTION tags -- each followed by an arbitrary amount of plain text (no HTML markup) -- See the following example.

```

Select - Notepad
File Edit Search Help
<HTML>
<HEAD>
  <TITLE>Comment Form</TITLE>
</HEAD>
<BODY>
<FORM ACTION="mailto:janderson@media.utah.edu" METHOD="post">
<P><B>Please select your occupation.</B></P><BR>
  <SELECT NAME="occupation">
    <OPTION>Teacher
    <OPTION>Administrative
    <OPTION>Student
  </SELECT>
<P>
<INPUT TYPE="submit" NAME="Submit">
<INPUT TYPE="reset" NAME="Reset"
</BODY>
</HTML>

```

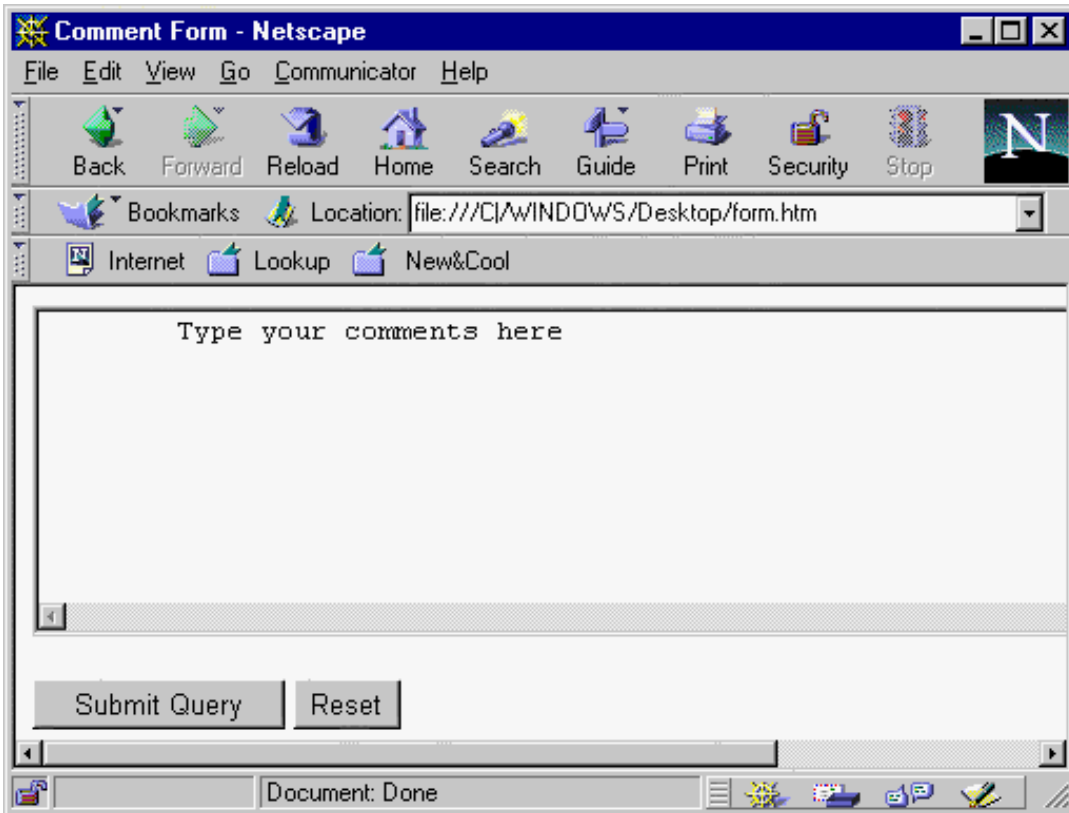
The <textarea> Tag

 <textarea>	TEXTAREA fields automatically have scrollbars; any amount of text can be entered in them.
name 	name = "somename" This is the name which is attached to the material in the form.
rows 	rows = "number" Specifies how many rows are displayed in the text area.
cols 	cols = "number" Specifies how many columns are displayed in the text area.

The TEXTAREA element requires both an opening and a closing tag. Here is an example of the <textarea> tag.

```
form.htm - Notepad
File Edit Search Help
<HTML>
<HEAD>
  <TITLE>Comment Form</TITLE>
</HEAD>
<BODY>
  <FORM action="mailto:rdahl@media.utah.edu" method="post">
    <TEXTAREA name="comment" rows=8 cols=80>
      Type your comments here
    </TEXTAREA>
  <P>
    <INPUT type=submit name="Submit Query">
    <INPUT type=reset name="Reset">
  </BODY>
</HTML>
```

Here is how the HTML looks in the browser:





Build a Form

Activity 5

Create a form that has text fields for: Name, E-Mail Address, School and Phone Number; radio buttons to identify whether the phone number is a work or home number and a comment box. Have this form got to your email address.

The following image is an example of how your form could look.

A screenshot of a Netscape browser window titled "Demo Form - Netscape". The browser's menu bar includes "File", "Edit", "View", "Go", "Communicator", and "Help". The address bar is empty. The main content area displays a form titled "Getting to know you". The form contains the following elements:

- A horizontal line separator.
- A text label "What is your name:" followed by a single-line text input field.
- A text label "What is your E-mail Address:" followed by a single-line text input field.
- A text label "What is your phone number:" followed by a single-line text input field and two radio buttons labeled "Work" and "Home". The "Home" radio button is selected.
- A large text area with the prompt "Tell us what you think." and a vertical scrollbar on the right.
- Two buttons at the bottom: "Send It!" and "Start Over".

The status bar at the bottom of the browser window shows "Document: Done" and several system icons.

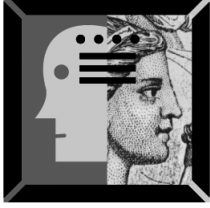


Image Maps

Section 6

Image Maps

An image map allows you to break up a single image into “hot spots” that when clicked will transport you to another page depending on where you clicked on the image. Rather than an entire image link to a single location and image map makes it possible for one image to link to multiple locations. Image maps are great for button bars and other graphics where different parts of the image should lead to different places, such as the image map that you will find on the Utah Education Network web site.








There are two methods of creating an image map. The older method server-side image map called for the use of a CGI script and required extra work on the server to make the system work. Most web designers use the easier and faster client-side image map technique.

Client-Side Image Maps

A client-side image map has two parts, an image and a map. The image can be any image. GIF and JPEG are most common. The map is created using the following group of HTML tags.

Part 1: The Image

 <A>	The anchor tag is used to identify the image as a link to another site. Attributes within the tag further define the image as an image map.
 	The name attribute in the map tag gives a name to the map section of the HTML document. This is the name which is linked to from the image tag which contains the image map.
 src	src = “the URL of the image for the map” This is the URL for the image which is to be used as the picture for the image map.
 usemap	usemap = “the relative link to the map” The usemap attribute links to the map section of the document. Remember to use a # sign before the name of the map given below in the <map> tag.
 ISMAP	ISMAP The ismap command lets the browser know the image is an image map. ISMAP stands all alone and isn’t set equal to anything.



```
<a>  </a>
```





This will load the image on the screen and make it an image map.

- The src attribute gives the URL of the image.
- The usemap attribute tells the name of the map we’re going to use with the image map, our map of the coordinates.
- The ISMAP attribute tells the browser this image is an image map.

Part 2: The Map

The map is completely enclosed with the <map> and </map> tags. Just like a table, several tags go in-between the <map> and </map> . These tags make up the link and coordinate data of the map.

 <MAP>	This tag is the holder for the image map area information.
 name	name = “mymap” The name attribute in the map tag gives a name to the map section of the HTML document. This is the name which is linked to from the image tag which contains the image map.

	<p>The area tag is one sided and used to define the characteristics of each link of the image map.</p>
<p>shape</p> 	<p>shape = "rect" The shape attribute designates the shape of the mapped regions. Other choices are circle, point, or poly.</p>
<p>coords</p> 	<p>coords = "number, number, number, number" The coordinates are the x,y for the top left of the section and the x,y for the bottom right. These four numbers work for the rect shape. The circle tag needs two sets coordinates, one for the center point and the other for a point on the outer circle. The point tag needs a single coordinate for a point. The poly shape can take as multiple coordinates and the browser assumes a line connects all of them much like connect-the-dots.</p>
<p>href</p> 	<p>href = "the URL of the link" The href attribute designates the link for a particular section. This is the page Netscape pulls up if the area described in the coords attribute is chosen.</p>

```
<MAP NAME="kayak" >
<AREA SHAPE=RECT COORDS="105,7,188,122"
HREF="http://www.uen.org" ALT="Utah Education
Network" >
</MAP>
```

This will create the clickable "hot spot" over the image

- The name distinguishes the map that goes with the image.
- The area attributes tell where on the screen (X,Y coordinates) where the "hot spots" should be.
- The HREF give the URL for that particular "hot spot".

The HTML to this point is fairly straight forward. The point that is likely to raise some concern is the coordinate pairs that must be entered to define the "hot spot". If you remember your Cartesian Geometry this will be easy. The numbers are geometrical reference points from the image identified as a X, Y pair. The X reference is the horizontal distance (in pixels) across the image from the left. The Y reference is vertical measure (in pixels) from the top of the image.

Finding the coordinates

Finding the coordinates to put in the <area> tag can be a little tricky. If you use a high powered HTML editor this process is typically automated for you. If not you will need to use image editing software or a freeware program that will create image maps.

For our purposes I'll explain the process using a freeware program Map This!.

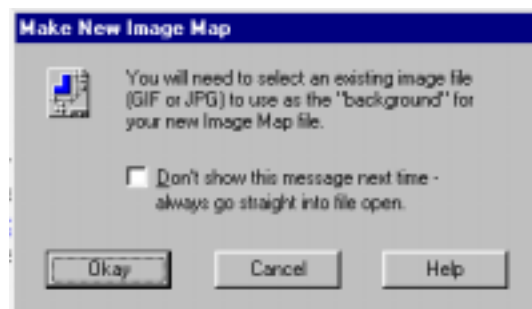
Map This!

Map This! is a freeware program by Todd Wilson that automates the process of image map creation. You can download a free copy at:

<http://hotfiles.zdnet.com/cgi-bin/texis/swlib/hotfiles/info.html?fcode=0007SG>

This is an easy to use program that allows you to create hotspot outlines for rectangles, circles and polygons on top of your image and then link URLs to them. We will go through a step by step process of how to use Map This!. Make sure that you have installed map this on you computer.

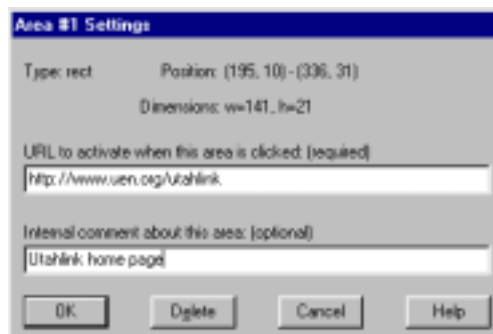
1. Go to the START menu and find Map This! Click on it to start the program.
2. Go to the FILE menu, Click on NEW.



3. Click on okay. This will take you to a file explorer menu. Brows to find the image you want to make into an image map. Once the image is loaded into Map This! you are ready to make your image map.



4. Chose the appropriate hot spot tool, in this case the rectangle, and click and drag where you want the hot spot to be on your image.
5. Click on the arrow tool once all of your hot spots have been defined and double click on one of them to set the URL and ALT information.



6. After you have finished "Save" the file and fill in the requested information. The title becomes the map name. Make sure you save using the file format as CSIM.
7. Chose the location to save your file. Use the save as type .html.
8. Your image map information is now saved in an HTML file. You can either copy the information into your own HTML document or you can edit the document containing the map information. Your final HTML document should look something like this:

```

<HTML>
<HEAD>
<TITLE>This is a Image Map</TITLE>
</HEAD>
<BODY>
Here is and image map.<BR>

<a><IMG src="kayakmenubar.GIF" usemap="#kayak" ISMAP></a>

<MAP NAME="kayak">
<!-- #\$-:Image Map file created by Map THIS! -->
<!-- #\$-:Map THIS! free image map editor by Todd C. Wilson
-->
<!-- #\$-:Please do not edit lines starting with "#\$" -->
<!-- #\$VERSION:1.30 -->
<!-- #\$DESCRIPTION:this is an image map -->
<!-- #\$AUTHOR:Jessica Anderson -->
<!-- #\$DATE:Wed Nov 10 21:20:57 1999 -->
<!-- #\$PATH:D:\HTML Document Treasure Chest\Images\Map-
Files\ -->
<!-- #\$GIF:kayakmenubar.GIF -->
<AREA SHAPE=RECT COORDS="6,6,98,122"
HREF="http://www.cnn.com" ALT="CNN News">
<AREA SHAPE=RECT COORDS="105,7,188,122"
HREF="http://www.uen.org" ALT="Utah Education Network">
<AREA SHAPE=RECT COORDS="195,7,281,122"
HREF="http://www.utah.edu" ALT="University of Utah">
<AREA SHAPE=default HREF="http://www.uen.org/utahlink">
</MAP>

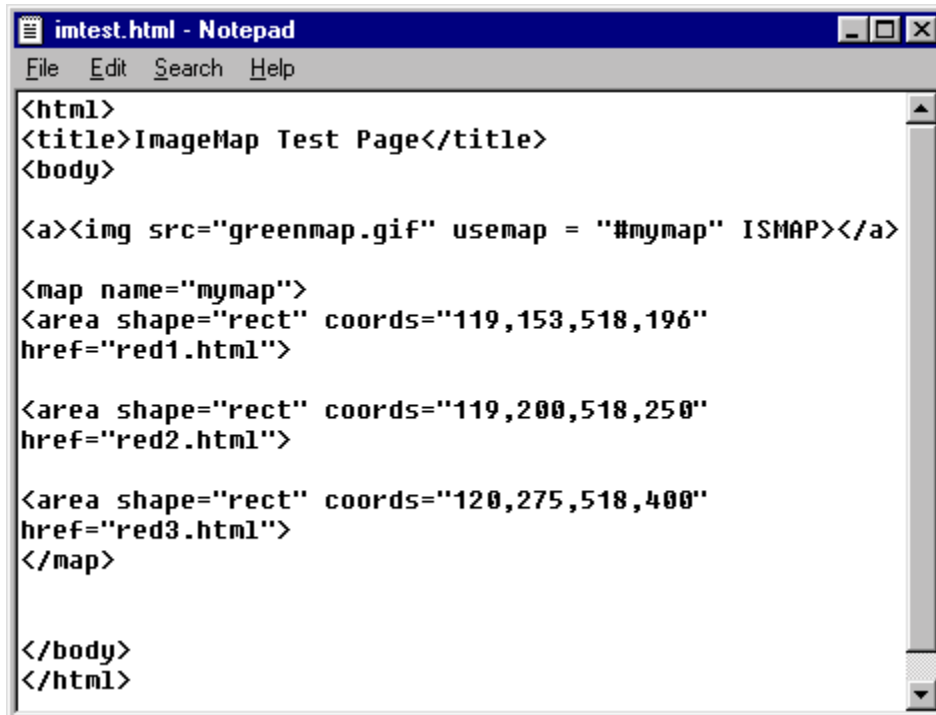
</body>
</HTML>

```

Putting it all together

Now that you have the basic pieces its time to put them together and create a functioning image map.

Here is a sample of the code for a client side image map.

A screenshot of a Notepad window titled "imtest.html - Notepad". The window contains the following HTML code:

```
<html>
<title>ImageMap Test Page</title>
<body>

<a></a>

<map name="mymap">
<area shape="rect" coords="119,153,518,196"
href="red1.html">

<area shape="rect" coords="119,200,518,250"
href="red2.html">

<area shape="rect" coords="120,275,518,400"
href="red3.html">
</map>

</body>
</html>
```

- The <a> tag has no attributes and surrounds the tag's contents.
- The name of the image I'm using for the image map is greenmap.gif.
- The name of the map is mymap but I put a # sign in front to tell the browser to find the map file in this document.
- I named the map section mymap inside the <map> tag.
- I have three different regions which point to different pages inside the <map> and </map> tags.
- I used a rectangular shape to mark sections of the image.

ISMAP

The ISMAP tells the browser "Hey, this is an IMAGEMAP" and is included in the tag. The browser then treats the image differently in the browser. It waits for a click on the image and records the coordinates for comparison with the <map> tag within the HTML document.

ImageMap Design Issues

Plan out the Image

Before you create the image, you should consider what type of information is going to be collected and how it should be organized on screen. Also think about how the image will fit into the page, perhaps as a navigation bar or as a large title screen. Finally, you should decide what colors and layout you will use to get the best visual effect while maintaining ease of use.

Have alternate text links

Its important to remember that many users will not be able to see your images or use ImageMaps -- either by choice or by software limitations. For these users, you should provide alternate description and access to linked resources.

Note:

Here is a web site that can give you additional information about image map programs.

http://www.zdnet.com/pcmag/features/webgraph/_open.htm



Create an Image Map

Activity 6

Take the supplied image and use it to create an image map.



Create a new web page for the image map. (Have this page replace the navigation page of your frameset.). Use the freeware program Map This! to come up with the image map HTML codes.



HTML 4.0

Appendix A

HTML 4.0 and You

HTML 4.0 is the latest version of HyperText Markup Language supported by the World Wide Web Consortium (W3C - www.w3c.org). As of 18 December 1997 it is the standard that is *recommended* for creating web documents and developing applications for the web. HTML 4.0 introduces an interesting shift from previous versions of the standard. These differences are important to understand as this standard is the transition link between older methods of creating documents for the web and the future of web page creation.

In the beginning HTML was developed as a tool for formatting text so that it could be presented over the Internet. As HTML was revised tags and attributes were added to facilitate the layout or presentation of the content. With the introduction of HTML 4.0 there is an active movement to return HTML to its original function of formatting text. Other tools are used in conjunction with the current standard to accomplish page layout and further define the kind of information being presented.

HTML 4.0 will be used as a transitional standard until sometime in late 1999 when the next edition of HTML is introduced. The next generation of HTML will be a "suite of XML tag sets". (XML stands for eXtensible Markup Language.) In addition to the HTML/XML tag sets page layout will be accomplished using Cascading Style Sheets and eXtensible Stylesheet Language.

The purpose for the migration from the simple HTML 3.2 standard is to add support for style sheets, internationalization of web documents, and increased accessibility to the Web for individuals with disabilities.

Three Versions of the Standard

HTML 4.0 is specified in three forms. Each of these forms are acceptable for use in web page design. Depending on your target audience one version of the standard may be more applicable than the next. It is interesting to note that many of the tags and attributes introduced by HTML 4.0 are still not fully supported by the version 4 browsers. All new tags and attributes introduced in this manual have reported support by the version 4 browsers.

HTML 4.0 Strict - Use this standard when you want to use bare essential tagging for your document. This version should be used in conjunction with Cascading Style Sheet language (CSS) to handle the page layout issues. It is important to note that only visitors using version 4.0 browsers or newer will be able to take advantage of the CSS page layout. All other guests will see the page via the unadorned HTML. In this sense *HTML 4.0 Strict* is downward compatible but better suited for the future editions of HTML.

HTML 4.0 Transitional - Use this version of the standard when you need to take advantage of HTML's presentational features because your target audience doesn't have the latest browsers that support CSS. In many respects this version is little different than HTML 3.2. The primary distinction is some tags and attributes have "lost favor" and should be avoided when possible.

HTML 4.0 Frameset - Use this version of the standard when you want to use Frames to partition your browser window. The use of Frames predates HTML 4.0, however this technique has only been officially recognized as an HTML tag set with this version of HTML.

Unappreciated HTML

Deprecate: v, To express disapproval or disparagement of. With each revision of the HTML standard the tags and their various attributes are reviewed. Some times tags are added and some are "deprecated." Basically tags are difficult to remove from the HTML standard because downward compatibility is an important attribute of HTML. However, tags can fall into official disapproval in the hopes that they will stop being used. It is important to understand which tags are favorable because this version of the HTML standard represents the transition phase to a new method (XML) of publishing for the Web.

The current plans for the next generation of markup language will break free of the current restrictions imposed by the current HTML standards. The W3C has stated,

"The new version of HTML will be informed by HTML 4.0 but not bound by it. There is no strict requirement for strict upward compatibility, although the migration path will be carefully considered."

In light of the movement away from the traditional methods of publishing for the web it is important to be informed on which tags will likely continue to function under the future standard. Continued downward compatibility is not guaranteed by the W3C. Adherence to the HTML 4.0 standard will minimize the amount of re-engineering required when the next generation of markup and the ensuing browsers arrive.

Deprecated Tags

The following is a list of tags that have officially fallen into disfavor with the W3C. Use of these tags will still work with current browsers and they represent tags that fall into the "Transitional" version of the standard. It is highly unlikely that these will be supported in the future as the majority of these tags are made redundant through the use of CSS or the creation of new broad based tags.

APPLET - This tag is used to hold Java Applet code within an HTML document. In the future the OBJECT tag will be used to identify any embedded objects.

BASEFONT - This tag is used to establish a default font for a web page. The font property within CSS eliminates the need for BASEFONT.

CENTER - Center was used to give a center alignment to items within the tag. This can be accomplished using CSS or the align="center" attributed within most HTML tags.

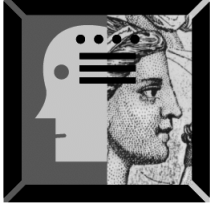
DIR - A five space indented left margin was created using this tag. The CSS property margin-left is the replacement for DIR.

FONT - Font was used to control the font face, size and color of text displayed on a web page. The family of CSS font properties as used to control the display of text.

MENU - A container for a list. is the designated replacement for this tag.

STRIKE - Render text as strikethrough. This can be used to show text of a document that has been abandoned. Using CSS controls to format the text is the preferred method now. Also two new tags will be available soon <INS> newly inserted text. deleted text.

U - The U tag was used to underline text. The CSS border-bottom properties control this type of display.



JavaScript Appendix B

Introduction to JavaScript

JavaScript is a scripting language developed by Netscape Communications tailored specifically for use on the web. It's an event driven language. It waits for the user to trigger an event and then performs some sort of action. For example, a popular JavaScript waits for the user to move the mouse over an image. When the mouse moves over the image the picture changes. JavaScript instructs Netscape to wait for the mouse to move over the image and when it does, change the image.

JavaScript can also take data entered by the user and incorporate it into the web page. For example, a web page could ask a user for his or her name. Then, using JavaScript, Netscape could insert the person's name anywhere it was called for in the document.

Obviously JavaScript can be used to spruce up web pages and create a more interactive environment for the user.

JavaScript vs. Java

Much of the confusion between Java and JavaScript stems from their similar names. They are actually two different languages. JavaScript is a scripting language built for the web and Java is more of a general programming language for building applications.

JavaScript is a subset of the Sun Microsystems' Java programming language developed by Netscape Communications. JavaScript's syntax is much like Java but is more forgiving than the full-scale programming language. Luckily for us, this makes learning JavaScript much easier than Java.

JavaScript is Object-based

JavaScript is not an "object-oriented" but rather an "object-based" language. Java is an "object-oriented" language. Objects are chunks of code which can be fit together to create a program. They represent a small portion of code that can be re-used over and over in several different programs much like articles in an encyclopedia. The encyclopedia might have information about CD players listed

several different places. They might be found under a section entitled electronics, or in an entry on Japan. CDs probably have their own entry as well. Here the same information can be used over and over.

In JavaScript, different parts of code can be used over and over the same way it can in Java. However, in Java, the objects can be built from scratch. In JavaScript, the user is limited to pre-defined or "pre-built" objects already built into the web browser that can be expanded and fitted for different tasks on the web.

Using JavaScript

JavaScript is inserted into the head, body, or both of an HTML document. Sometimes, both the head and body contain JavaScript.

<SCRIPT> Tag

JavaScript is embedded into the HTML two different ways. The most common is the <script> tag. All the JavaScript commands go between the <script> and </script> tags. The <script> tags can go in either the <head> or the <body> of the document.

Older browsers may not recognize the <script> tag and won't be able to execute the JavaScript commands. In fact, when a web browser doesn't understand a tag, it skips right over it and moves on. This can be a big problem in JavaScript.

If the browser doesn't know the <script> tag, it will proceed to the next set of instructions in the HTML. Ideally, the <script> tags enclose the JavaScript and keep the command from being displayed in the browser. However, if the browser doesn't understand the <script> tag, it will proceed to display the JavaScript commands on the screen instead of executing them.

To prevent this, the JavaScript commands should be enclosed with HTML comment tags to prevent them from being shown in the browser. Remember, comment tags enclose text and keep it from appearing in the browser window. The HTML comment tag looks like this:

```
<!--      Everything in here will be hidden      -->
```

JavaScript also has a way to comment out lines of the script. It uses two forward slashes.

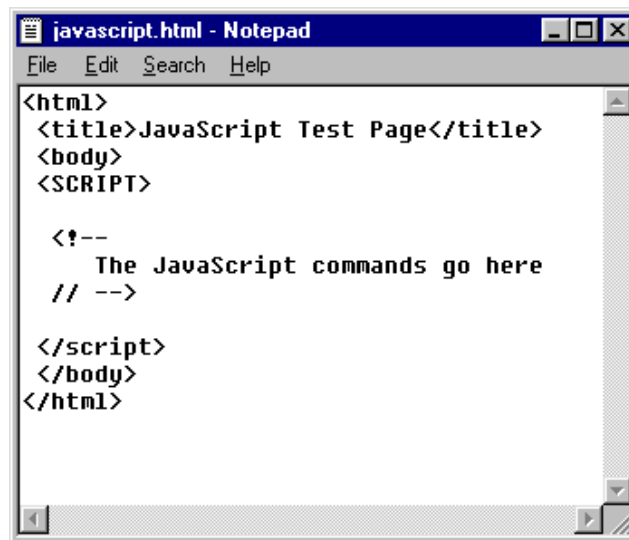
```
// This text will not be read by JavaScript.
```

The following table shows how comments are written in different programming environments.

Language	Comment	Example
JavaScript	//	// Everything here will be commented out if on the same line.
HTML	<!-- -->	<!-- The browser will ignore everything in here -->
DOS	REM	REM This line will be ignored in DOS

In our HTML document, we need to use both the HTML and JavaScript comments.

Notice the JavaScript comment before the last half of the HTML comment. It's necessary for the script to work. Otherwise, JavaScript will misinterpret the end of the tag and the script will produce an error.



Older Browsers

Now when an older browser accesses the page above:

- It will first ignore the <script> tag because it doesn't understand it.
- It will skip over the HTML comment that contains our JavaScript commands.
- It will ignore the </script> tag.
- We have succeeded in hiding the JavaScript from *less-fortunate* browsers. Had we not taken these precautions, all the JavaScript commands would have appeared on the screen and made absolutely no sense to the user.

Newer Browsers

Now if a JavaScript-capable browser accesses the same page:

- It recognizes the first <script> tag.
- It knows to read the commands from inside the HTML comment tag so we don't have a problem.
- It executes the commands and finishes the script.
- Remember that we had to include the JavaScript comment to hide the end of the HTML comment tag.

Sample JavaScripts

JavaScript to print the date of the last change.

```
<html>
<title>Fun Date Modified</title>
<body>
  <script language="JavaScript">

    <!-- hide the script from old browsers --

    //The function to write the date in the browser.
    //The function is called LastUpdate and we
    //can call it up at anytime in the body.

    function LastUpdate(){
      document.write("Last Updated:
"+document.lastModified);
      document.write();
    }
    //--end hiding here -->
  </script>

  <!-- Here is where we call up the function
      to print the date -->

  <strong><script language="JavaScript"> {LastUpdate();}
</script></strong>

</body>
</html>
```

This script will count the number of pages the user has visited in a session. It uses the Netscape History to find the number.

Script to count the number of pages you've visited in a session.

```
<html>
<title>Pages Visited</title>
<body>
<script>

    // The variable called times is set to the
    // history number Netscape has been keeping.

var times=history.length;

    // The script write out to the window using the
    // variable times in the middle of the
    // sentence.

document.write('You\'ve visited '+times+' pages(s) in this
browser session. ');
</script>
</body>
</html>
```

Forward and Back buttons

This goes in the head

```
<script language="LiveScript">
  <!-- hide the script from old browsers --
  function gohist(gox) {
    history.go(gox);
  }
  //--end hiding here -->
</script>
```

This goes in the body of the document where you want the forward and back buttons.

```
<html>
<head>
<title>Pages Visited</title>
<script language="LiveScript">
  <!-- hide the script from old browsers --

  // Function defined here in the head but
  // called up inside the body.

  function gohist(gox) {
    history.go(gox);
  }

  --end hiding here -->
</script>
</head>
<body>

<!-- Build a form where we can monitor clicks on buttons.
      Wait for a click on the buttons and then go respective
      place in the history list. -->

<form method="post">
  <input type="button" value="<- Back" onclick="gohist(-
1)">
  <input type="button" value="Fwrd ->"
onclick="gohist(1)">
</form>

</body>
</html>
```

Put a clock on the page.

```
<HTML>
<HEAD>
<TITLE> Showing date and time on a document</TITLE>
<SCRIPT LANGUAGE="JAVASCRIPT">
<!--Hiding the code
var show_time=false;
var timerID=null;

function stop(){
    if (show_time){
        clearTimeout(timerID);
        document.clock.date_time.value=" ";
    }
    show_time=false;
}

function start(form){
    var today=new Date();
    var display_value = " Time= " + today.getHours()
    if(today.getMinutes() < 10){
        display_value+=":0" + today.getMinutes();
    }
    else{
        display_value+=":" + today.getMinutes();
    }
    if (today.getSeconds() < 10){
        display_value+=":0" + today.getSeconds();
    }
    else{
        display_value+=":" + today.getSeconds();
    }
    if(today.getHours()>=12){
        display_value+=" P.M."
        /* Here we have a variable called mygreeting where
        you can store something like this:
        mygreeting ="Good Morning!"; */
    }
    else{
        display_value+=" A.M."
        /* Now set mygreeting to:
        mygreeting ="Good Afternoon!"; */
    }
    display_value += " Date= " + (today.getMonth()+1) + "/"
+
    today.getDate() + "/" + today.getYear();
    document.clock.date_time.value=display_value;
```

```
        timerID=setTimeout("start()",100);
        show_time=true;
        /* Here have an alert() method do the following:
           alert(mygreeting); */
    }
    //done hiding-->
</SCRIPT>
</HEAD>
<BODY BGCOLOR=white Text=Red Link=Green onLoad=stop()>
<center>
<H2>Displaying Date and Time</H2>
<FORM name=clock>
<INPUT type="text" name="date_time" size=35 value=" "><br>
<INPUT type="button" name="show_now" value="Display"
onClick=start()>
<INPUT type="button" name="clear_now" value=" Clear "
onClick=stop()>
</center>
</FORM>
</BODY>
</HTML>
```

Click a button to load and play an audio file

```
<HTML>
<HEAD>
<TITLE>Playing on-demand sound</TITLE>
<SCRIPT LANGUAGE="JavaScript">

    // The function play will start the audio file walker.au
    // when called.

function play(){
window.location = "walker.au"
}

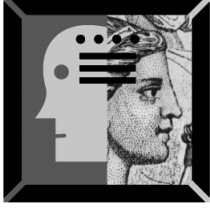
</SCRIPT>
</HEAD>

<body>
<b>Click on the button below to hear Walker Texas
Ranger.</b><br>

    <!-- Here is the form for the button.  It will run the
    function play() when clicked -->

<form>
<input type = "button" name = "Play" value = "Play" onclick
= "play()"
</form>
</body>
</HTML>
```

JavaScript runs on the client and has no interaction with the server. Accessing the server requires a CGI script. All the commands are contained inside the HTML of the web page. Because of this, anyone can read, copy, and use the JavaScript in your page simply by viewing the document source.



Web Resources

Appendix C

The Web is a great resource for learning more about HTML and Web Site design. The following is a list of some of the on-line resources available for web page creators.

HTML Resources

HTML Check: <http://www.netmechanic.com/cobrands/zdnet/htmlcheck/>

Essential HTML 4.0:

<http://www.zdnet.com/devhead/stories/articles/0,4413,2143797,00.html>

HTML home page <http://html.miningco.com/internet/design/html/mbody.htm>

HTML Tag list <http://www.erinet.com/martyh/html/>

Site Inspector <http://siteinspector.linkexchange.com/>

Web Design Resources

CNET.com – Web Building <http://home.cnet.com/category/0-3880.html>

Devhead <http://www.zdnet.com/devhead/>

Devhead – Accessibility

<http://www.zdnet.com/devhead/filters/0,,2133228,00.html>

Accessibility for All

<http://www.zdnet.com/devhead/stories/articles/0,4413,1600188,00.html>

HTML Writers Guild Resources <http://www.hwg.org/resources/>

PERL and CGI Resources

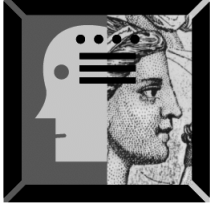
Matt's Script Archive <http://www.worldwidemart.com/scripts/>

Matt Kruse's Scripts <http://mkruse.netexpress.net/scripts/>

Devhead CGI/PERL/TCL <http://www.zdnet.com/devhead/filters/cgiperltcl/>

CGI Resource Index http://www.cgi-resources.com/Programs_and_Scripts/Perl/

The PERL Journal <http://www.itknowledge.com/tpj/>



Web Accessibility

Appendix D

The Web Accessibility Initiative is a program implemented by the World Wide Web Consortium (www.w3c.org). The W3C's commitment is to "lead the Web to its full potential including promoting a high degree of usability for people with disabilities. The Web Accessibility Initiative (WAI), in coordination with organizations around the world, is pursuing accessibility of the Web through five primary areas of work: technology, guidelines, tools, education & outreach, and research & development."

What is Accessibility

The purpose behind the WAI is to make the web more accessible to individuals with disabilities and more useful for clients using new 'page viewing' technologies. Accessible design is not minimalist design, but thoughtful design. Pages should be designed such that the widest possible audience can access the information.

General concepts for accessible design are drawn from the W3C document "**Web Content Accessibility Guidelines**" (<http://www.w3.org/TR/WAI-WEBCONTENT/>).

Quick Tips for Accessible Web Design

1. Create pages that conform to accepted standards. Standards for the web are governed by the World Wide Web Consortium. www.w3.org
2. Know the difference between structural and presentational elements; use style sheets when appropriate (CSS).
3. Use HTML 4 features to provide information about the purpose and function of elements. HTML 4 has added several attributes to the language to help add extra information to your web page.
4. Make sure your pages can be navigated by keyboard.
5. Provide alternative methods to access non-textual content, including images, scripts, multimedia, tables, forms, and frames for user agents (browsers) that do not display them.
6. Be wary of the common pitfalls that can reduce the accessibility of your site.

An extended discussion of these concepts is available at www.hwq.org/resources/accessibility/sixprinciples.html

Check Your Work

When you are done creating your web page, Validate the HTML and use evaluation tools and text- only browsers to verify accessibility.

Lynx – Text based browser. <http://lynx.browser.org/>

Bobby – Accessibility test. <http://www.cast.org/bobby>

Homework

Credit is available in two forms: university credit from Southern Utah University for recertification or inservice credit from the State Office of Education. To receive credit, return the following assignment with in one month of final class date to:

UEN Staff Development
201 Milton Bennion Hall
1705 Campus Central Drive
University of Utah
Salt Lake City, UT 84112-9250

or via fax at (801)585-9040. If you request State credit, your request will be forwarded to the State Office of Education for final approval. If you request university credit, an application form will be returned to you which you can fill out and send back with the fee to SUU. Forms must be returned by the following deadlines: November 1st (fall semester), March 10th (spring semester), August 1st (summer semester).

Name _____ SS# _____

Address _____ Class Title _____

_____ School _____

Date Attended _____ District _____

Instructor _____

Type of Credit:

University (SUU)

500 Level – Graded

\$15.00/per credit

State Office

No Charge

Assignment:

Create a web page that includes at least two of the following elements:

- META Tags
- Frameset
- Form
- Image Map

Complete pages should look professional and include at least 40 lines of code. When finished, send either the URL (if the site is on the web) or a diskette with all of the appropriate HTML files and graphics to the above address.

Information Sheet

Your Name

School or Employer

Email Address

Phone Number

Mailing Address

1. What a types of training would you like to see offered by UEN Staff Development.

2. What areas do you feel like you need additional training?

Course Evaluation

Course Title

Instructor

Dates

Please give both a numerical rating and an explanation. On numerical items, 1 means poor, 2 means fair, 3 means average, 4 means good, and 5 means very strong.

Overall, how would you rate your course experience as compared to other inservice you have received? 1 2 3 4 5

Compared to other UtahLINK Training you've taken, how would you rate this course?
1 2 3 4 5

What was the **best part of this class**?

List three **ways this class could be improved**.

What should the course **spend more time on** or cover that it doesn't?

In terms of curriculum, what should the course **spend less time on**?

How would you rate the **clarity** of the curriculum?

1 2 3 4 5

How would you rate the **delivery** of instruction?

1 2 3 4 5

How would you rate the **knowledge of your instructor/s**?

1 2 3 4 5

How would you rate the **accessibility of instructors and level of personal attention**?

1 2 3 4 5

How would you rate the **manual quality**?

1 2 3 4 5

How would you rate the **pace of instruction**?

Too Fast Good Too Slow

What will be your biggest **obstacles to implementation** of what you learned?

Please rate your past experience with each of the following aspects of UtahLINK (leave an item blank if you have not experienced it)

Overall _____

Help Desk _____

Web Design _____

Training _____

Web Content _____

Installation Team _____

Please clarify!
