Web Authoring: Intermediate

Academic Computing Services

A Division of Information Services

www.ku.edu/acs

Abstract: Learn advanced techniques for creating hyperlinks, especially targeted links and relative URLs, as well as advanced `<head>` elements such as `<meta>` and `<base>`. Add images to your Web pages and make non-HTML documents available for download from your Web sites.

Contents

Introduction

This course covers additional Web authoring techniques not covered in the *Web Authoring: Introduction* workshop. Participants learn to place images (graphics), use special characters such as ©, and include advanced `<head>` element features in their Web pages. In addition, advanced techniques for creating and using hypertext links are explained. Participants use two browsers, Netscape and Microsoft Internet Explorer to view their Web pages and learn the subtle differences in rendering between them. Upon completion of this workshop, participants will be prepared to take part in any of the following Web Authoring training sessions offered by ACS: *Tables, Frames, and Imagemaps*; *Cascading Style Sheets Introduction*; and *Forms*.

Objectives

The goal of this workshop is to further the skills of participants as Web authors and prepare them for additional training as Web authors. After this workshop, participants will be able to:

- use these advanced `<head>` element tags: `<meta>` and `<link>`
- create targeted internal and external hyperlinks
- put email links on their Web pages
- work with advanced relative URL concepts
- make non-HTML documents available for download from their Web sites
- place images in their Web documents and understand issues related to graphics such as size and load time.
- attend *Web Authoring: Tables, Frames, and Imagemaps*; *Cascading Style Sheets Introduction*; and *Forms*

Prerequisites

Prior to attending this workshop, participants should attend *Web Authoring: Introduction* or have equivalent skills. The handout for the class (which lists the skills acquired during the class) is available online at http://www.ku.edu/acs/docs/wkshop/webauthor-intro.pdf.

Related Training Available from ACS

All workshops offered by Academic Computing Services (ACS), a division of Information Services, are free to KU students, staff, faculty, and approved affiliates**.** The general public is also welcome to most workshops, but some ACS workshops require a registration fee for them.

To learn more about or register for workshops, receive automatic announcements of upcoming workshops, and track workshops you've registered for and have attended, visit the ACS Web site at www.ku.edu/acs/train. You can also check our online schedule at www.ku.edu/acs/schedule for a list of class offerings and their availability. For further workshop related questions, please email training@ku.edu.

Definitions

Term	Definition
Web robot	Other names for a robot are spiders, web crawlers, worms, ants, wanderers, and gatherers. Simply put, these are programs that automatically traverse the Web's hypertext structure by retrieving a document, and recursively retrieving all documents that are referenced in that document. The documents are then cataloged in a database according to the particular priorities, schemes, and algorithms of the robot maker. Having one's site cataloged by a robot is usually considered desirable, although not always. However, the Web author has *some* influence over how his/her site is cataloged.
Search engine	A search engine is a program that searches the database created by a **Web robot** based on keywords provided by a user. The search engine then provides "hits" (references to Web sites) based on matches between user requests and database entries. With the ever-expanding size of the Web, searching the Web for meaningful information can be very frustrating. Consider attending the ACS workshop *Finding Information on the World Wide Web: Basic* for help with this.

The <a> element

Email links

In the *Web Authoring: Introduction* class, you learned to use the anchor (`<a>`) tag to create links to other HTML documents using the `href` attribute. It is also possible to use this attribute, combined with a special type of URL, to create a link that opens the email program associated with the browser. This is used to offer visitors an easy way to send you feedback via email. (Of course you may simply list your email address so that visitors may copy it down and email you using another email program.) The format for creating an email link is as follows:

```
<a href="mailto:me@somewhere.com">Send me email</a>
```

Replace *me@somewhere.com* with the email address to which you want email sent. The text **Send me email** will be clickable, just as a normal hyperlink is. When the visitor clicks the link, the email program associated with the browser opens, ready for the visitor to compose a new message. The **To:** field automatically contains the email address given in the `href` attribute.

Including a Subject:

It is possible to pre-set the subject of the email sent to you as a result of clicking on the email link. This is helpful for quickly identifying the source of incoming email. To add a

Subject, simply append `?Subject=`*`text-of-subject`* to your email address in the email URL. For example:

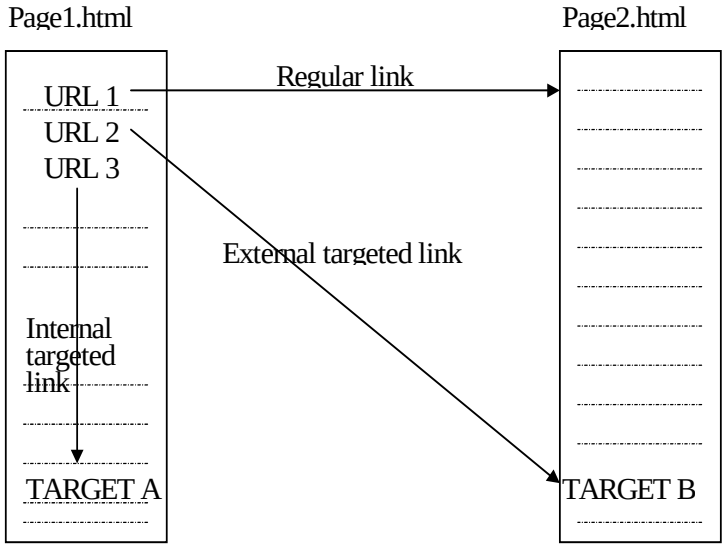
```
<a href="mailto:jugglers@somewhere.com?Subject=Juggling Info
Request">Send email to Jugglers R Us</a>
```

Targeted links

When a user clicks a hyperlink text or image a new document is displayed, normally from the beginning (or top) of the document. However, it is possible to display a document beginning somewhere other than top. This is known as a targeted, or anchored, link. More specifically, sometimes called an *external targeted link*.

In addition, it is possible to jump to a specific location within the same document, rather than opening an entirely new document for display. This is called an *internal targeted link*.

The following diagram illustrates the various types of links.



Regular and targeted links

Creating targeted links is a two step process. First you must identify the location in the document that you would like to reference, thus creating a target. Second, you add the target name to the URL in the hyperlink so that the browser knows to begin the display at the targeted location rather than at the beginning of the specified document.

Naming the target

To make a reference marker or target, skip to the place in the document to which you want to refer and surround that text with an `<a>` tag with a `name` attribute. You may pick the value you assign to name. For example,

```
<a name="s-terms">S Terms</a>
```

Note:
- The name **is** case-sensitive and must match **exactly** when you reference it later.

- The name **must** begin with an alphabetical character or a number
- The name **cannot** have spaces.

This tag goes in the place labeled “TARGET A” in the diagram. Note that the text within the <a> element is actually optional; even if you include text between these tags, it will not appear highlighted, underlined, or sensitive in any way. This anchor tag is not supposed to “do” anything; it simply marks a spot for later reference.

Referencing the target

To reference your target information in a URL (URL 2 or 3 in the diagram), append the pound sign (#) followed by the name you gave the target (in this case, “s-terms”) to the end of the URL.

Internal targeted link

For example, the (relative) URL for creating the internal link shown in the diagram would look like this:

```
<a href="page1.html#s-terms">Go to terms that start with S</a>
```

If the link is within the same document, it is possible to leave out the document name. In that case the (relative) URL for creating the internal could look like this:

```
<a href="#s-terms">Go to terms that start with S</a>
```

When a user activates the link represented by **Go to terms that start with S**, the browser will skip to the <a name="s-terms"> reference in the document and begin the display at that point.

External targeted link

External targeted links work in the same manner as internal targeted links except the target is in another HTML document as opposed to somewhere within the current document.

Thus, if, for example, you name a target “John_Smith,” in a page containing a list of personnel biographies, you can reference it from any other page using a (relative) URL of this type:

```
<a href="personnel.html#John_Smith">John Smith's biography.</a>
```

Of course if **personnel.html** is not in the same location as the document currently being displayed, you would need to use an absolute URL to reference the document, such as:

```
<a href="http://www.server.com/ourcompany/personnel.html#John
Smith">John Smith's biography.</a>
```

When a user selects the link represented by **John Smith's biography**, the browser will present the file **personnel.html** beginning with the text at the “John_Smith” target.

Links to non-HTML documents

In addition to HTML documents, documents saved in other formats can also be made available via the Web. For example, if you have a form, such as an application or an exam, that you would like visitors to your site to have access to but you do not want to

make it an HTML document, you can provide a link to it and the user can then open or download it from your Web site. Examples of popular file formats are:

PDF Portable Document Format

DOC Microsoft Word

WPD Corel WordPerfect

RTF Rich Text Format

XLS Microsoft Excel

Bear in mind, however, that the visitor to your site must have software that can open the file in order to view its contents. In the list above, PDF requires Adobe Acrobat Reader, a free program available on the Web. RTF, on the other hand can be read by most word processing programs and represents a useful neutral file format that can be widely used without regard to version number or platform of the software that was used to create the file originally.

Note: Some non-graphical browsers, such as Lynx, may not be able to download or display these documents.

To create a link to a non-HTML document, simply use the document (file) name in the URL, just as you would with an HTML document. For example, to include a link to a final exam on your Web site, the HTML might look like this:

```
<a href="finalexam.doc">Final exam in Microsoft Word 97
format</a>
```

Or, if you want to make it more widely available and independent of Microsoft Word 97, save it in RTF format (from within the word processor you used to create it) and include a link like this:

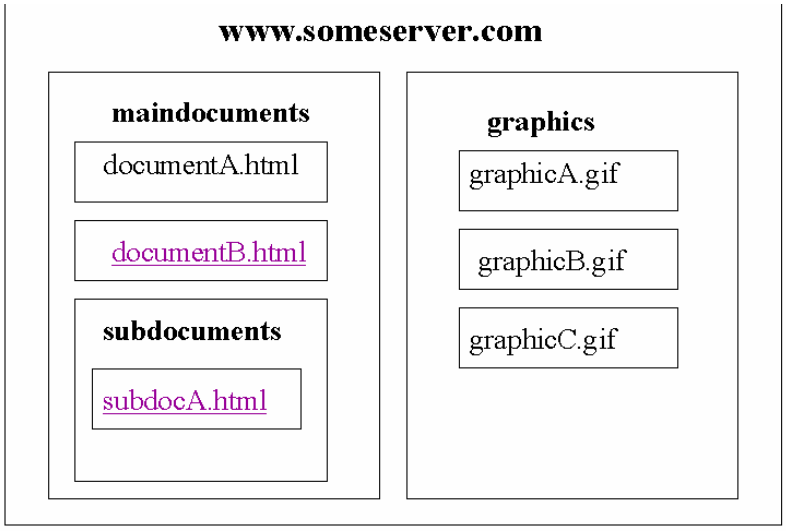
```
<a href="finalexam.rtf">Final exam in Rich Text Format</a>
```

When a visitor to your site clicks a link such as this, the browser will launch the application associated with that file format, or offer to let you save the file locally.

Note: In older browsers it may be necessary to configure the browser to launch a given program when it encounters a particular file type (as denoted by the filename extension).

More about relative URLs

Relative URLs are used when linking to documents on the same server as the current document. In *Web Authoring: Introduction* you learned how to create a hyperlink to a document on your Web site using a relative URL. In that exercise, the document you were linking from and the one you were linking to were in the same directory. The (relative) URL was straightforward: “filename.html.” However, once your site grows more complex, you may organize your files into subdirectories. Consider this schematic of a file structure.



Sample file structure

Referencing subdirectories

If the document you want to link to is in a subdirectory of the current directory, simply reference the document relative to the current document.

For example, if **documentA.html** links to **subdocA.html** the correct relative URL for accessing the file is:

subdocuments/subdocA.html

This URL contains all the information necessary to find the document relative to the location of the document it is currently displaying.

Referencing higher directories

If the resource to be referenced is on the same server but in a different (same level or higher) directory, the relative path identifier **..** can be used.

For example, if **documentA.html** has **graphicA.gif** in it, the correct relative URL for accessing the graphic is:

../graphics/graphicA.gif

Images

Image issues

Formats

Currently the two graphic formats used widely on the Web are GIF and JPEG. *GIF* (pronounced “jif”) stands for Graphics Interchange Format. *JPEG* is pronounced "jay peg" and stands for Joint Photographic Experts Group. Images must be saved in one of these formats in order to be part of an HTML document. Which format should you use?

This depends on what type of graphic you're working on and what kind of effects you want.

GIF

GIF files are excellent for images with large areas of the same color (supports up to 256 colors) such as logos, line drawings, and graphics. The GIF format also supports transparency and interlacing. A *transparent* GIF image retains only the foreground, so the background of the HTML page shows through it. *Interlacing* is a scheme in which the browser begins drawing the image one line at a time from top to bottom, then starts at the top again, until the image is complete. This way, the user can get an idea of the whole picture right away instead of waiting for the browser to finish the top completely, then the middle, and then the bottom.

JPEG

JPEG works well with photos (supports millions of colors), but it does not support transparency or interlacing. Because they usually include many colors, JPEG files are often larger than GIF files. Generally, use GIF unless you're working with a photo.

File size and load time

If you've ever browsed the Web with a 14.4K modem, you've probably run into a Web site that took so long to load that you gave up and either turned the images feature of your browser off or ignored the page altogether. Large graphics can take an extremely long time to load. The term "large graphics," in this context, means a large *file size* rather than large dimensions. An incorrectly saved, converted, or compressed small graphic can take more time to load than a properly created full-page graphic. As a rule of thumb graphics files should be no larger than 60K. If you must display an image larger than 40K, consider offering users one of these alternatives:

1. Provide a link to the graphic rather than displaying the graphic directly on the page. The link should include a warning about the size of the graphic
2. Use *thumbnails* (small versions of the graphic). These smaller versions (spatially small) load faster and give the user an idea of what the bigger image might look like. Then if users want to see a larger version of the image, they can click the thumbnail and view the larger version.

In addition, proper use of the `height` and `width` attributes can help Web pages load faster, and, conversely, improper use can make loading more sluggish. If you use `height` and `width` to indicate the actual height and width of the graphic, then browsers will reserve the corresponding space for the image and load the text first. However, if you use `height` and `width` to *resize* your image, the graphic will load more slowly because the browser must do the necessary calculations to re-render the image in the new size. Therefore, if you want an image to appear larger or smaller, use a graphics program to alter the size before including it in your document.

Adding images

To include an image in your document, use the `<img>` tag. Note that it requires an attribute in order to function.

Attributes

`src` stands for “source” and identifies the URL of the image to be displayed.

Value: the URL of the image

Example:

```
<img src="http://www.ku.edu/mydept/smith.gif">
```

`alt` stands for “alternate” and provides a text description of the image for use by non-graphical browsers. The text included in the `alt` attribute will display in place of the image in non-graphical browsers or when a graphical browser can’t display the image (either the user has image display turned off or the image is unavailable for some reason.)

Value: descriptive text of your choice.

Example:

```
<img alt="A picture of Prof. Smith."
src="http://www.ku.edu/mydept/smith.gif">
```

Note: *This attribute*, unlike the `height` and `width` attributes that follow, is **required**. In the absence of an `alt` attribute, the browser will display `[INLINE]` in a non-graphical browser. This in turn will be read as “open bracket inline close bracket” on a speech synthesizer used by the vision-impaired. However, if your graphic is not part of the content of your page, rather than an `alt` attribute such as `<img alt="Little blue dot." src="bluedot.gif">`, use the following format:

```
<img alt="" src="bluedot.gif">
```

This “empty” `alt` attribute will cause the speech synthesizer to simply skip the graphic altogether.

`height` specifies the height of the image in pixels, speeding the load time of images.

`width` specifies the width of the image in pixels, speeding the load time of images.

Example:

If your image is 400 pixels tall and 320 pixels wide, the `<img>` tag would look like this:

```
<img height="400" width="320" src="rover.gif" alt="My dog.">
```

Hint: One way to find out the height and width of a graphic in pixels is to open the image in Netscape. The browser window lists the height and width of the graphic at the top of the window in the title bar. Of course any graphics program gives this information as well. Examples of graphics programs include Adobe Photoshop, Microsoft Image Composer, and Microsoft Photo Editor.

Images as links

You already know how to include text in anchor tags such that when a user selects the text, the URL specified in the anchor tag is accessed. It is similarly easy to include an image inside an anchor element so that is the image that is to be invoked in order to access a new Web document (or other Web resource). In the first example, a user selects the word "back" then the document **file1.htm** will be accessed:

```
<a href="http://www.foo.com/file1.htm">back</a>
```

Similarly, in the following example the image named **back.gif** will be sensitive, and if a user selects the image, then the document **file1.htm** will be accessed:

```
<a href="http://www.foo.com/file1.htm"><img src="back.gif"></a>
```

Using images in anchor tags simply involves replacing the text in the first example with an `<img>` tag in the second.

Retrieving graphics from the Web

Any graphic that is not otherwise copyrighted or trademarked may be downloaded from the Web. To retrieve a graphic from a Web site,

1. Mac users: Place your mouse icon on the graphic. Click and hold the mouse button until a pop-up window appears.
   PC users: Place your mouse icon on the graphic. Click and hold the **right** mouse button until a pop-up window appears.
2. Choose **Save picture as…** or **Save image as…** and save the image as you would any document, into the directory of your choice.

The <head> Element

In addition to `<title>`, other tags and their attributes may appear in the `head` section (between `<head>` and `</head>`) of HTML documents.

<base>

The `<base>` tag designates a URL from which all other relative URLs in the HTML document are interpreted. `<base>` can simplify the use of URLs in HTML documents. Suppose you are composing a document in which you plan to include several images and text files that are stored in a location other than that of the document you are currently working on (e.g., in another directory). Normally, you would have to type in the complete URL of each image and text file you reference. However, by using the `<base>` tag to specify the location of the remote directory holding the resources you plan to use, you may use relative URLs throughout your document. The `<base>` tag must include the href attribute, which specifies the URL from which all relative URLs in the document will be referenced.

EXAMPLE:

Let's assume you are modifying a Web document called **deptintro.html**. The original version is online at http://www.ku.edu/~departmentx and has a few graphics in it. You plan to use those same graphics in this modified version but haven't downloaded them to

your desktop while you're modifying the text portion of the document. However, you do want to see what the modified version looks like, complete with the graphics. You have two choices. You can use absolute URLs to reference the images (e.g., http://www.ku.edu/~departmentx/image1.gif). However, by including the `<base>` tag in the `<head>` section of your draft document, you can identify a base URL and then use a relative URL to reference the image (and any other items on www.ku.edu/~departmentx to which your Web page links). The <head> section of the document would look something like this:

```
<head>
<title>Stuff I'm working on</title>
<base href="http://www.ku.edu/~departmentx">
</head>
```

Since you've included the `<base>` tag, you can refer to your files on Raven as if they actually resided on your desktop computer in the same location as **deptintro.html**. In this example, the graphic would be included on the page like this:

```
<img src="image1.gif">
```

Note: This example assumes you have Internet access during this drafting stage. If you're working offline, the benefits of the `<base>` tag will only be available while you're connected (online).

<meta>

The `<meta>` tag allows Web authors to specify metadata—information *about* a document rather than document *content*—in a variety of ways. An HTML document can contain multiple `<meta>` elements.

Many Web robots such as Lycos and Yahoo use this meta-information for indexing Web sites.

name

The `name` attribute is used in conjunction with the `content` attribute to define the nature of the information being provided. The HTML specification does not list legal values for the `name` attribute. Common usages, however, are:

```
name="description"
name="keywords"
name="author"
name="copyright"
```

(Others have been incorporated into an independent, formal standard for metadata created by an organization called the Dublin Core Metadata Initiative. For information, see http://dublincore.org/.)

content

The `content` attribute is used in conjunction with the `name` attribute to provide content for the property defined with the `name` attribute. The HTML specification does not list legal values for the `content` attribute. Instead, provide freeform information relevant to the property defined with the `name` attribute.

EXAMPLE:

If the document you're working on includes information about vegetarianism, your `<meta>` elements might look like this:

```
<head>
<meta name="description" content="An informational site about
vegetarianism">
<meta name="keywords" content="vegetarian, vegan, health,
environment">
<meta name="author" content="Jane Smith">
<meta name="copyright" content="&copy; Vegetarians R Us, Inc.">
</head>
```

meta and search engines

A common use for `meta` is to specify keywords that a search engine may use to improve the quality of search results.

KU's search engine

The University of Kansas Web site uses AltaVista Intranet Search technology to index Web pages and provide searching capabilities. Once your page is linked into The KU Web site, it will be indexed in the search engine database and visitors can use keywords to find it. Thus, even if a visitor doesn't have the URL of your web page, if he/she finds the KU Web site, a simple search from within the KU Web site should bring up your Web site.

The search engine catalogs web pages based on occurrences of words in the `<title>` tag and within the `<body>` element. In addition, you can specify keywords and text for describing your site through `<meta>` tags. Information about influencing how Alta Vista Search indexes your site is available at: http://www.ku.edu/cgi-bin/search?pg=h#meta.

Avoiding the search engines

At times it may be desirable *not* to allow a search engine (web crawler, web worm, etc.) to gather information about your page or other pages to which it links. Details on how to do this are available at:

http://info.webcrawler.com/mak/projects/robots/meta-user.html

<link>

The `<link>` tag defines the HTML document's relationship to another document or object and can be used more than once to define multiple relationships. `<link>` can establish a connection to related indices and glossaries, to different versions of your current document (for example, graphical versus text versions), to a help file for your document, and more.

`href` defines the URL of the document or object to be referenced. Note: `href` must be present in `<link>`. It is required and is used in conjunction with either `rel` or `rev` as defined below.

Values: a relative or absolute URL

Note: The following attributes all take these values: *alternate*, *stylesheet*, *start*, *contents*, *index*, *glossary*, *copyright*, *chapter*, *next*, *previous*, *help*, *section*, *subsection*, *appendix*, and *bookmark*.

`rel` defines the type of relationship between the current document and the one defined in the `href` attribute.

`rev` defines the type of relationship between the one defined in the `href` attribute and the current document. This is the reverse of `rel`.

EXAMPLES:

```
<link title="Table of Contents" rel=contents href="toc.html">
<link title="Chapter Two" rel=next href="chapter2.html">
<link title="French version" type="text/html" rel="alternate"
hreflang="fr" href="http://www.ku.edu/frenchdep/frenchver.html">
```

Special Characters

Occasionally you will need to include characters like ©, ¢, or diacritical marks in your HTML document. In addition, you may also wish to include characters that would normally be considered markup tags within the text of your document. For example, if you are writing about HTML and you want to talk about the <a> tag, you can't just write `the <a> tag` in your document, because the <a> will be interpreted as an anchor tag and will not be rendered visible to the reader.

HTML refers to these special characters as text entities (they may also be referred to as escape sequences). The format for inserting an entity into your HTML is

`&`*`name`*`;` or `&#`*`number`*`;`

name the HTML entity name or label

number the ISO Latin-1 character number

EXAMPLES:

`<a>`	<a>
`&Auml;`	Ä
`&#196;`	Ä
`&copy;`	©
`&#169;`	©
`&#162;`	¢
`&reg;`	®

Lists of character numbers and their entity names are located at:

http://www.december.com/html/spec/latin1.html

http://www.pemberley.com/janeinfo/latin1.html

http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/iso_table.html

http://www.w3.org/TR/html401/sgml/entities.html

Most HTML reference books will have an appendix that contains symbols and their corresponding codes.

Tips & Tricks

View Source

The **view source** feature of most browsers will allow you to see the HTML used to create the document currently being displayed. Study the source code to learn how others have created their special effects.

Comments in HTML

Occasionally you may want to leave a note to yourself within your HTML code. This is called a comment and is created as follows:

```
<!-- This is a comment and will not be rendered by the
browser. -->
```

Note: Avoid putting a string of hyphens (-----) in your comments, as this is likely to result in confusion for the browser.

Appendix A—Web design principles

The Web today is filled with poorly designed pages. Overuse of big graphics and flashy HTML elements make documents slow and difficult to read. A few tips will help you avoid driving visitors away before they have read your information.

The most important thing to remember when designing Web pages is that you are designing a Web page because you have something to say (content to provide). The principles discussed here can help you present your information clearly and in a manner that makes it accessible to all.

Backgrounds

Backgrounds have become popular, as evidenced by the many collections of colors and textures on the Web today. Judicious use of backgrounds and color can spice up an otherwise unattractive or dull Web page.

If you decide to use a plain colored background, consider using a light background rather than a dark one, as dark text on a light background is much easier to read than light text on a dark background.

If you choose to use a patterned or textured background, try to choose one that isn't too bright or highly contrasted. Be sure to choose one that doesn't obscure your text. Most importantly, remember to check your background on several different platforms; textures that look great on a PC may look awful on a Macintosh, and vice versa.

Flash and gimmicks

Have you been to a Web site with four different frames that you couldn't figure out how to escape from, with words scrolling across the bottom, top, or even middle of your

screen, with pulsating stars or flying planes zipping across your screen, or with all of the above? Most flash on the Web has no useful purpose; if you decide to use any of these tricks, use them with care and offer a way to "turn them off" so that visitors who prefer not to be subjected to those gimmicks can avoid them without leaving your site.

Frames

Frames are covered in the *Web Authoring: Tables, Frames, and Image Maps* class. Be careful to learn to use frames properly or they can be a hindrance rather than a help to your site. In particular, people with small monitors sometimes have difficulty with pages designed with non-scrollable or non-resizable frame windows. Also, non-graphical browsers cannot implement frames. If you decide to use frames, be sure to include `<noframes>` information for those who can't use frames.

Accessibility

HTML is constantly evolving to make Web pages easier for those with disabilities (especially visual impairments) to access. Strive to be aware of how your site "works" for someone using a non-graphical browser or a speech synthesizer to access your content. Attend our *Web Authoring: Improving Accessibility* class for in-depth assistance with this topic.

More help with design

Additional information on good Web site design is available at Web sites listed in Appendix B.

What about animated graphics? Where do I get them? Can I make my own?

Animated graphics can be retrieved from the Internet in the same fashion as other graphics (described in the question above). See Appendix B for URLs regarding animated graphics download sites and sites with info on how to make your own animated images.

Appendix B—Helpful Web sites

Official HTML 4.01 specifications

http://www.w3.org/TR/html401/

Computer training offered by Academic Computing Services

http://www.ku.edu/acs/training/

Documentation available from Academic Computing Services

http://www.ku.edu/acs/docs

HTML Info & Tutorials

http://www.kumc.edu/webdev/html/

http://builder.cnet.com/webbuilding/0-7250.html

http://hotwired.lycos.com/webmonkey/authoring/

http://www.mcli.dist.maricopa.edu/tut/

http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html

http://www.htmlgoodies.com/tutors/

Cascading Style Sheets

http://www.w3.org/TR/REC-CSS2/

http://www.htmlhelp.com/reference/css/

Buttons, bullets, lines, and stamps

http://www.theshockzone.com

Color palettes

http://www.pawprint.net/designresources/web-colours.php

Clip Art, backgrounds, animated graphics, etc.

http://www.clip-art.com/

Animated graphics (animated gifs) gallery

http://members.aol.com/royalef/galframe.htm

Animated graphics (animated gifs) how-to

http://members.aol.com/royalef/gifanim.htm

HTML Validation services

http://validator.w3.org

Good help on creating web pages

http://wp.netscape.com/browsers/createsites/index.html

Check your web page for accessibility for those with disabilities (free service)

http://bobby.watchfire.com/bobby/html/en/index.jsp

How to Use Metatags

http://www.searchenginewatch.com/webmasters/meta.html

Meta Resources

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

Dublin Core Metadata Initiative

http://dublincore.org/

Search Engine Features for Webmasters

http://www.searchenginewatch.com/webmasters/features.html

More info on Robots metatags

http://www.webdeveloper.com/html/html_metatags_part2.html

http://www.w3.org/Search/9605-Indexing-Workshop/ReportOutcomes/Spidering.txt

Getting Additional Help

ACS provides consulting and Q&A help in a variety of ways:

785/864-0200
question@ku.edu
www.ku.edu/acs/help

Last Update: 06/23/2004