ASK-PC.com
WE'VE THE ANSWER!

Certified Website Designer

Website Design Fundamentals

Make professional websites

html, Dreamweaver....

Your Key To Pass ACWSD Exam

By
Mohamed N. El-Guindy
Founder & CIO
ASK PC, LLC
Alpharetta, GA USA.
www.askpc.org

Volume 1
March, 2007

ACWSD HML-L1 Study Guide

Prepared and Edited by
Mohamed N. El-Guindy
BSc. CS. Trinity University, USA
MCSE, MCT
IEEE Computer Society Member
British Computer Society Member
IWA & HWG Member
Member of Experts Exchange
Member of E-Learning Guild
Member of WAOE
Listed in "Marquise Who's Who in the World 2007 Publication" by invitation.
Chief Information Officer
ASK PC Computer Academy
Alpharetta, GA
United States of America

ASK-PC
The Largest Arabic Technical Support Community in association with
Culminis
Microsoft
Symantec
Winternals
Sysinternals

ASK PC, LLC is a registered legal name in United States of America.

Contact the Author
admin@ask-pc.com
naguib@computer.org

HTML Level-1

Creating a Web Page

Lesson Objectives

In this lesson, you will create a simple Web page.

You will:

- Create a global structure and save with the correct file extension.
- Add a page title based on page title guidelines.
- Add color and an image to change the appearance of a Web page background.
- Create paragraphs.
- Embed an image.

Introduction

You have previously viewed Web pages on your computer. In order to create a basic Web page or the more complex pages similar to those that you have already seen while surfing the Internet, you will need to know HTML. In this lesson, you will use HTML to create a basic Web page consisting of text and graphics.

When creating Web pages, the main goal is to create a satisfying experience so that users will want to come back. To accomplish this, you need to create pages that appeal to the user's needs and sensibilities, as well as function properly. Users can become frustrated when information is lost because the content can't be viewed or part of a site does not function correctly. Properly using HTML to display the text and graphics is a good starting point to create basic Web pages that won't frustrate your users.

Create a Global Structure

You have previously used a Web browser to view the pages of a site. In order to make this possible, the browser must have been able to recognize not only the fact that the files you viewed were Web pages, but the browser must have also been able to identify defined containers within the file. In this topic, you will create these identifying containers and save the file as a Web page format that the browser will recognize.

Without entering the logical containers that will hold the text and graphic content of your Web page and saving your file with the correct extension, the Web browser won't be able to recognize any of your files as Web pages. To prevent the major failure of your Web pages, you will need to identify the file as a Web page and enter these identifying containers.

The World Wide Web

The *Internet* is a world-wide network of computers. There are numerous ways you can communicate over the Internet: you can send electronic mail (email), transfer files, log in remotely to distant computers, and view hypertext multimedia files (Web pages). The most popular segment of the Internet is the *World Wide Web* (or simply"the Web") which consists of an indescribably large quantity of Web pages. Using a software application called a *Web browser,* you can access and view the contents of these Web pages. A Web browser is an example of a *client*, which is a computer or software application that relies on another computer to complete operations. Other examples of clients that are becoming increasingly popular for accessing Web pages are devices such as cellular phones and personal digital assistants (PDAs).

HTML Elements

Definition

HTML elements are a set of instructions that describe the structure and content of a Web page. These instructions are applied to the content of a Web page through the use of tags. HTML elements can be either container or empty elements. A *container element* contains the data that is intended to be affected by that element. It consists of opening and closing "tags." You will examine empty elements in a later topic.

Tags

HTML container elements consist of two tags. The opening tag marks the beginning of the element and the closing tag marks the end of the element. Anything between these tags is affected by the element.

Example

The following table includes a description of HTML global structure elements and their proper tags:

Description	Opening Tag	Closing Tag
Identifies the file as an HTML file. This is the highest-level set of tags and, therefore, it contains all the other HTML tags of a Web page.	<html>	</html>
Identifies the portion of the file that contains the head elements of the Web page. Most of the content contained in the head section simply describes various aspects of the page and isn't actual text or graphics that are displayed on the Web page.	<head>	</head>
Identifies the portion of the file that is the body of the Web page. This is where you enter the text and graphics that are displayed when your page is viewed in a browser.	<body>	</body>

Container Element Syntax Rules

The syntax for HTML container tags is wonderfully simple. The following list describes the rules for HTML container elements:

- Consist of both an opening and closing tag.
- An opening tag consists of the tag name enclosed within less-than (<) and greater-than (>) brackets (for example: <body>).
- A closing tag also consists of the tag name enclosed in less-than and greater-than brackets, but the less-than bracket of a closing tag is always succeeded with a forward-slash (/) (for example: </body>).
- The content that is intended to be affected is contained within these tags (for example: <body>Text content.</body>). The opening tag alerts the browser to treat the content that follows according to the rules of the tag name, and then the closing tag tells the browser that the application of those rules is finished.

HTML tags are case-independent, meaning that they may be entered in either upper- or lower case (for example: <body></body> is just as valid as <BODY></BODY>). Some HTML coders prefer to enter all of their tags in upper case so that they stand out a bit more from the rest of the page content. Other developers feel that it is simpler to enter HTML code in lower case; saving them from

repeatedly pressing [Shift]. The HTML tags in this course are presented in lower case, although you have the option to enter them in caps.

Create a Global Structure

Procedure Reference

To enter the global structure of an HTML document:

1. In a blank document open in your text editor, enter the <html></html> container tags.
2. Place the insertion point between the opening and closing tags and press [Enter] until there are one or two empty lines between the tags.
3. On the empty line below the opening <html> tag, type <head>, press [Enter] a few times and then type </head>. You should have a couple of empty lines between the <head></head> tags. You will enter elements in this section later in the course.
4. Place the insertion point on the line below the </head> tag.
5. Type <body>, press [Enter] a few times, and type </body>. Again, you will be entering elements on these blank lines later in the course.
6. Save the file using either the .htm or .html file extension.

HTML File Names

In order for your files to be recognized as Web pages, you need to use your text editor to save them with either the .htm or .html file extensions. Once you have saved the files with one of these extensions, you can open them and view them in any browser. If you intend for a file that you are creating to be the *home page* (main page) of the site, then there are also considerations to be made. Most Web servers are set up to automatically accept either index.html (or index.htm) or default.html (or default.htm) as the file that will be used as the site's home page. That means that this will be the first file that a user sees when they go to your main Web address. Some companies that host Web sites will require you to name your site's home page using one of these two file names, while others will let you create your own home page file name.

Add a Page Title

Previously, you performed the necessary steps to identify a file as a Web page and define its document sections. Just like an author likes to title his or her work before beginning to write, you will give the Web page a title. Instead of the title appearing on the cover as it would on a book, a Web page title is displayed in the title bar of the browser. In this topic, you will add a page title that will appear in the title bar of a browser and serve as a bookmark label.

As a Web developer, one of your most important goals is to create a Web site that users will want to visit again and again. To assist the user's ability to create a shortcut to the site, it is a good idea to specify an appropriate page title. By default, a browser uses the page title as a label to identify the page in your Favorites or Bookmarks menu. This allows them to go directly to any page on your site without making it necessary to remember the Web address, which can become quite complicated as the size and sophistication of your site increases.

Page Title Guidelines

It is important to supply users with a useful default page title so that they can easily reference the pages you create from others in a list of bookmarks. To determine useful page titles, adhere to the following guidelines:

Guidelines

The golden rules for page titles:

- Be brief. Keep in mind that this title may end up as a menu item in a user's Favorites menu. Long page titles can easily become annoying menu options.
- Be specific. Instead of “Product Description Page,” give the name of the product that is being described.

Things you may want to include:

- The name of the company, Web site, or organization.
- The highest-level heading or subject of the page if it gives the user a better idea of what the page is about. This is just about the minimum for most page titles.
- Symbols such as a hyphens or colons to denote a topical hierarchy within the title. In many cases, the name of the company or organization is listed first in the page title at the top of the hierarchy. Any subsequent categories or topics are listed afterwards and are separated by some type of symbol.
- If the content of a site is organized by category, include the name of the category that the bookmarked page falls under.

Things to avoid:

- Marketing slogans. If you must include a marketing slogan, be sure to keep it as brief as possible. They don't serve any purpose to someone that wants to create a shortcut to the page.
- Meaningless codes or product numbers. The codes or item numbers that a company uses to internally identify their products isn't as useful to a Web user as the actual name of the product.

Example

The following are generally good examples of useful page titles for an online car parts Web site:

- Bob's Car Parts (This title is brief and specific and includes the name of the company—an ideal title for a home page.)
- Bob's Car Parts: Fenders (This title includes the name of the company listed first in the hierarchy, followed by the category that the user is viewing. The two items are separated by a colon.)
- Bob's Car Parts: Fenders: 1995 Jeep Wrangler (This title includes the elements of the previous title, but also includes a descriptive product name that is listed in the hierarchy and is separated by a symbol.)

Non-Example

The following are examples of page titles for the same company that aren't as useful:

- Bob's Car Parts– Your one-stop source for domestic and foreign auto parts. (This title includes a long marketing slogan.)
- Order Form (This title isn't specific enough and it doesn't include the name of the company or organization.)
- Product Description Page: Part #598828 (This title doesn't include the name of the company or organization and the product's part number isn't as useful to the user as

the actual name of the product.)

Modify the Page Background

Now that most of the legwork is done to identify the file as a Web page and to identify the page to users who choose to bookmark it, it is time to do some coding that will determine how the page will look. Before you add text and graphics to the body of the Web page, it is important to establish the appearance of the Web page background. In this topic, you will modify the background of a Web page.

Some of the most popular sites on the Web use a white background color; such as Amazon.com, Yahoo!, and Google. You will probably find yourself using a white background for some of the designs that you create, but eventually you may want to give your pages a very different feel—perhaps to match a theme. Suppose a client came to you requesting a golf theme. You could really establish the theme by changing the background of the Web page to match the aesthetics of the theme. In this case, a solid green background or an image with a subtle, grassy texture might be appropriate.

Attributes

Definition

An *attribute* is part of an HTML element that modifies the characteristics of that element. Most HTML elements have a defined set of attributes that you can modify to suit your needs.

In this topic, you will discover how attributes of the <body> tag allow you to modify the appearance of a Web page's background.

Example

The following table includes a description of two <body> element attributes that relate to modifying the background of a Web page:

Description	Attribute	Values
This attribute specifies a solid background color for the body of a Web page.	`bgcolor`	Any valid color name or hexadecimal color.
This attributes specifies an image file to be used as the background of the Web page. All text, tables, embedded graphics, etc. will display on top of this image.	`background`	The file path to location of the designated background image. Although the attribute is not case-sensitive, the file path to the image *is* case-sensitive.

Attribute Syntax

Attributes are always included in the opening tag of a container element and they use a `name="value"` syntax. Using this syntax, the name of the attribute is always followed by the equals sign (=) and the value of the attribute is contained within quotations (" ") (for example `<body **bgcolor="red"**></body>`). Most HTML elements include more than one attribute and the order in which you specify them does not matter. HTML elements include default attribute values which are used if no attributes are specified.

Absolute and Relative File Paths

When specifying the location of files on a Web site, you can direct the browser to a file using either absolute or relative file paths. An *absolute file path* includes the entire directory path of the file all the way back to the root directory, which is usually the hard drive (commonly labeled the C:/ drive on a PC or Macintosh HD on a Mac). An example of an absolute file path to an image in a Web project may be “C:/Windows/Desktop/Web Project/graphics/banner.gif”.

A *relative file path* is one that is more commonly used when doing HTML coding. With a relative file path, you specify the location of the file based on the location of the file you are working with. Comparing it to the absolute file path example, if you had an HTML file that was in the Web Project folder, you would reference the banner.gif graphic using the“graphics/banner.gif” relative file path. It assumes that the Web page is in the same directory as the graphics folder and goes from there.

Background Image Behaviors

The default behavior for a background image is to tile (repeat). This means that if the height or width of the background image are less than the height or width of the browser window, the same background image is rendered adjacent to the original one so that the image constantly covers the entire background no matter what the size of the window is.

Color Methods

Color methods are the means by which you specify the colors of your Web page. Colors are specified as attributes of various HTML elements. Using only HTML, you have two choices for specifying colors: *color names* and *hexadecimal colors*.

Color Method	Description
Color Name	• The easiest method for specifying colors on Web pages. • Simply use the name of the color as an attribute value. If you want the color blue, you specify the color name blue. If you want green, you specify green.
Hexadecimal Color	• A more complicated, yet more powerful and accurate method of specifying color. • Uses hexadecimal notation to describe the RGB (Red, Green, Blue) components of a color. Combining these component colors gives you a palette of millions of colors to choose from.

Web-safe Colors

Unfortunately, life isn't all that easy when you are working with colors on the Web. Browsers like Netscape and Internet Explorer do not use the same palette of colors. Neither do operating systems like Microsoft's Windows and Apple's Mac OS. This puts a good size of the color spectrum in limbo because some color values will render as different shades depending on what browser (IE or Netscape) or platform (PC or Mac) a person uses. Fortunately they do have some colors in common that make up the Web-safe color palette. *Web-safe colors* (also known as browser-safe colors) are a set of 216 colors that, rest assured, will display exactly the same on every computer and in every browser.

Of course, it may not be vital that all of your colors render exactly the same. After all, many of the shading differences of non-Web-safe colors are small enough that not using them may detract from your design more than Web-safe colors can help it. That is why it is important to test your colors in both Netscape and Internet Explorer and, if possible, on both PCs and Macs.

Hexadecimal Notation

A hexadecimal color is made up of three bytes—one each for a red value, a green value, and a blue value. Each of the three component colors is defined by a pair of digits and follows a single pound sign (#). An example of a hexadecimal color is #88ffcc. The first two digits following the pound sign make up the red value, the second two digits make up the green, and the third two digits make up the blue. The range of digits is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, & f. Zero is the least amount of that color while “f” is the greatest. For example, a red value of “00” means that there is actually no red color, while a red value of “ff” is the most amount of red possible (pure red). The same scheme applies for the green and blue colors as well. To put it all together, a hex value of #00ff00 would define no red color (00), the most amount of green possible (ff), and no blue color (00). The resulting color would be pure green.

Web-safe colors are the biggest reason that hexadecimal colors are more powerful than color names. There are only 10 Web-safe color names, while hexadecimal colors allow you to access the entire 216 available colors. It is easy to determine whether a hexadecimal color is Web-safe. A Web-safe hexadecimal color only uses the following pairs of identical digits to define the three component colors: 00, 33, 66, 99, cc, and ff. For example, #9900cc is Web-safe because the red color (99) uses two identical digits, the green color (00) uses two identical digits, and the blue color (cc) uses two identical digits.

The following table lists the 10 Web-safe color names and the hexadecimal equivalents:

Hexadecimal Color	Color Name
#ffffff	white
#000000	black
#00ffff	aqua (or cyan)
#ff0000	red
#00ff00	lime
#0000ff	blue
#ffff00	yellow
#ff00ff	fuschia (or magenta)

Modify the Page Background

Procedure Reference Modifying the Default Background Color

To change the Web page background to a color other than the default of white:

1. Add the following bold attribute along with a color value to the opening body tag:

   `<body` **`bgcolor="value"`**`>` For example: `<body bgcolor="yellow">`
2. Save the file and test your work in at least one browser. The background has changed from the default white color.

If possible it is a good idea to check your colors in both Internet Explorer and Netscape and on both PCs and Macs.

Procedure Reference Adding a Background Image

To add a background image to a Web page:

1. Add the following bold attribute along with a file path referencing the background image to the opening <body> tag:

   `<body bgcolor="yellow"` **`background="value"`**`>` For example: `<body bgcolor="yellow" background="images/bgimage.gif">`

It is good practice to include a background color that is similar to the dominant color of the background image. This will keep the overall appearance of the site consistent if the user happens to have a browser that doesn't display graphics.

2. Save the file and test your work in a browser.

Create Paragraphs

In the previous lesson you modified the background of a Web page by adding a solid color and background image. Now that the appearance of the background is established, you can get a better idea of how the text that will be displayed in the foreground of your page will appear. In this topic, you will add text to the body of the Web page and format it as a paragraph.

Unless you want all of the text in the body of your Web pages to appear as a single, continuous string of text, you need use HTML to tell the browser how to display it. See the following activity for an example:

Create Paragraphs

Procedure Reference

To format text as a paragraph on a Web page:

1. In the body section of the document, identify the block of text that you would like to format as a paragraph.

   This would be text that you would like to display with a line break before and after the paragraph to separate it from the content above and below.
2. At the start of the block of text that you have identified as a paragraph, type .
3. At the end of the same text block, type .
4. Save the file and test your modifications in a browser.

Block-level Elements **Definition**

Block-level elements are elements that cause the browser to automatically render a line break before and after the element.

Example

A paragraph is an example of a block-level HTML element.

Alignment

HTML allows you to align paragraphs three ways: left, right, and center. The default alignment for a paragraph is left. To modify the alignment of a paragraph, you can use the `align` attribute that uses left, right, and center as values. Figure 1-2 shows effects of applying each of the three types of alignments to paragraphs:

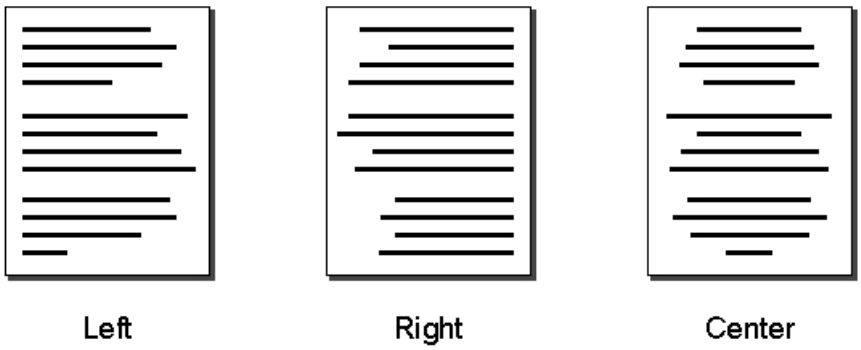


Figure 1-2: ***The affect of applying alignments to the text of paragraphs.***

Embed Images

Next to text, the second most common type of media on a Web page is static graphics. You have already added some paragraphs to the body of a Web page, now you will increase the visual appeal by embedding graphics.

In many cases, images are more effective than words. Most companies like to include their logo on their Web pages because it establishes brand identity. In some cases, such as Apple Computers or Nike, often the identity of the company is so closely associated with the logo that including the name of the company isn't even necessary. On the Web, images are often used to enhance words as well. For example, a user is much more likely to buy an item that they can see a picture of, rather than an item that is only described in words.

Web Image Formats

Compared to the file size of an HTML file, images are typically large and the purpose of a *Web image format* is to compress the files into smaller file sizes so that they can be easily downloaded on the Web. The two main image formats for use on the Web are GIF (pronounced either *gif* or *jif* "depending on who you ask") and JPEG (pronounced *j-peg*). There are many factors involved in choosing the best format for Web images. The correct format depends on the characteristics of the graphic you are working with.

Web Image Format	Description
GIF	• Created in 1987 by Compuserve, licensed by Unisys. • Creates large file sizes, but it decompresses quickly. • Utilizes 8-bit color with a maximum of 256 colors. • Is a lossless format, meaning that the image quality doesn't degrade as the amount of file compression increases. • Good for line art, graphics with large areas of solid color, and pictures with few colors. • Variations include interlaced, non-interlaced, transparent, and animated. • Files of this type can be identified by their ".gif" file extension.
JPEG	• Developed in the early 1990's by C-Cube Microsystems. • Small files, but slow decompression. • Utilizes 24-bit color with a maximum of over 16 million colors. • Is a lossy format, meaning that the image quality degrades as the amount of file compression increases. • Good for photographs; images are more realistic than GIFs. • Variations include regular and progressive. • Files of this type can be identified by their ".jpg" file extension.

Another format called PNG (pronounced *ping*) showed promise as a Web image format in earlier years, but it hasn't picked up too much in popularity other than being used as source files that are later converted to GIFs or JPEGs.

Web Images

A *Web image* is a graphic that has been created or modified to be suitable for the Web. That generally means that it has been created using Web-safe colors, it uses an appropriate resolution, and it has been saved as the proper Web image format. You have already realized in a previous topic the importance of using Web-safe colors to ensure that the colors will appear consistent in multiple browsers and on various platforms like PCs and Macs.

Web Image Resolution

When creating a graphic that is to be used on the Web, it is also important to use an image resolution that is no more than 72 pixels per inch (ppi). That is the maximum resolution that can be displayed on a Web page. A graphic created at a higher resolution will still display in a browser, but still only at 72 ppi. Since higher resolution means larger file size, creating a graphic that uses a resolution higher than 72 ppi will only mean that your users will have to wait longer to view a graphic that will only display at 72 ppi anyway. As a Web designer you will most likely find yourself using a graphics application like Photoshop to convert print graphics or scanned photographs, which typically have a much higher resolution than 72 ppi, down to a "Web-friendly" resolution.

Embed Images

Procedure Reference

To embed an image in a Web page:

1. Identify the location where you want the image to be embedded.
2. In your text editor, enter the `<img>` tag and its vital attributes:

   `<img src="value" height="value" width="value" alt="value">` For example: `<img src="graphics/ourprods.gif" height="75" width="300" alt="Our Products">`
3. Save the file and test your work in a browser.

Empty Elements **Definition**

You have already seen quite a few container elements so you know that they use and opening and closing tag. An *empty element* uses only a single tag and a closing tag isn't necessary. In some cases, the attempt to add a closing tag to an empty element will cause unexpected results so it is important that you don't mistakenly attempt to close them. Just like container elements, empty elements have sets of attributes that can be applied in the same manner.

Example

The `<img>` tag is an empty element because it doesn't use a closing tag. The following table lists the characteristics of the `<img>` element.

Element	Attribute	Description
`<img>`	`src="value"`	Accepts an absolute or relative file path that references the location of an image file.
	`height="value" width="value"`	The height and width attributes match the height and width of the image. It is good practice to specify the height and width attributes for all of your images—it allows your pages to load faster because you tell the browser the dimensions of the graphic rather than the browser needing to figure it out on its own.
	`alt="value"`	The function of the alt attribute is to display alternate text that describes the graphic. In Internet Explorer, this alternate text is typically displayed as a tooltip that pops up when a user hovers their mouse pointer over

	the graphic. In Netscape, this text is displayed in place of the graphic until the image file loads.
`border="value"`	The border attribute displays a border of a specified width around an embedded image. The default border width is 0.

Alternate Text

While alternate text is a simply a nice feature for those of us that use browsers that display graphics, it is vital for people who use devices that are text-only. It is the only possibility a user will have to be able to use aspects of your Web page. Examples of these devices are cell phones and PDAs (although many of these are becoming graphic-enabled), synthesized screen readers, and braille displays. Screen readers and braille displays are browsers designed to allow visually impaired people to use the Web.

While alternate text may sound like a "nice feature," others feel that it is a requirement. As a matter of fact, part of the United States Federal Government's Americans with Disabilities Act (ADA) requires that state and local governments and the business sector must be prepared to offer communications over the Internet via an accessible medium. The guidelines to meet the criteria of the ADA includes the use of alternate text for images. Taking steps to make your Web pages more accessible will not only allow you to meet the requirements of the ADA, but it will also give you the benefit of a loyal user base.

Replaced Elements

Unrelated to empty elements, *replaced elements* are somewhat unique to HTML. Instead of affecting content within the source code, they are merely references to content that is located outside of the document. The `<img>` tag is an example of a replaced element because the physical image file isn't contained within the source code. The tag merely creates a placeholder for the graphic, which references an external file.

Structuring Content

Lesson Objectives

In this lesson, you will add structure to content.

You will:

- Create multiple levels of headings.
- Create a list.
- Notate your code using HTML comments.
- Insert a horizontal rule.
- Create and prevent line breaks.
- Group elements.

Introduction

In the first lesson, you got a taste of what structuring content with HTML is all about. You used HTML to create a simple Web page that consisted of text and graphics. You can use additional HTML elements to structure content, which gives users a better idea of the purpose of sections on your page. In this lesson, you will use HTML to categorize sections, which will make your content more meaningful as well as give you more formatting options.

The HTML elements that are presented in this lesson give you a host of formatting options that will allow you to come closer to creating the type of professional Web pages you typically see while surfing the Web. They provide you with more formatting options than you had available to you in the previous lesson so that users

won't have a difficult time interpreting the meaning of your Web page. These options include creating multiple levels of headings, creating lists, easing your workflow, physically dividing content, creating and preventing line breaks, and modifying the alignment of segments of your Web page.

Create Headings

You have already created a simple Web page. As the pages that you create become more complex, it will become increasingly important to categorize the information so that it is easier to locate. In this topic, you will create headings that will categorize information on a Web page.

Imagine that you were viewing a Web page that was titled Everything You Could Ever Want to Know About Coffee and you were trying to find a detail about a specific type of coffee filter. This page contains a ton of information about the broad topic of coffee including origins, types, grinds, and brewing methods. Thankfully, the Web developer who created the page used headings to properly categorize the content on the page. As a result, you can quickly locate the heading for the topic that contains information about filters without being forced to read a lot of unrelated information.

Heading Sizes

On a Web page, a heading is a block-level element used to briefly describe the section that follows it. There are six levels of headings; h1 being the most important, to h6 being the least important. A browser will render the more important headings using a larger font size, while the less important headings use a smaller font size. Table 2-1 shows the tags for the six levels of headings.

Table 2-1: *The opening and closing tags for the six levels of headings*

Description	Opening Tag	Closing Tag
First (highest) level heading	<h1>	</h1>
Second level heading	<h2>	</h2>
Third level heading	<h3>	</h3>
Fourth level heading	<h4>	</h4>
Fifth level heading	<h5>	</h5>
Sixth (lowest) level heading	<h6>	</h6>

Figure 2-1 shows the relationship of the heading sizes based on their importance.

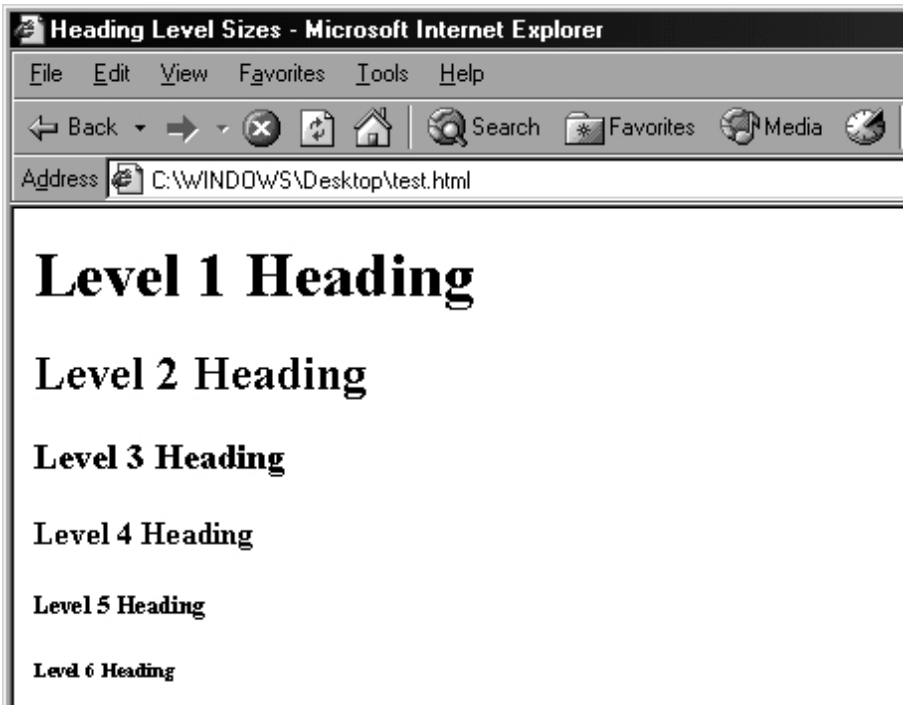


Figure 2-1: ***The relationship of the heading sizes based on their importance.***

Create Headings

Procedure Reference

To create a heading on a Web page:

1. Determine where you want the heading to be located.
2. Determine which level of heading you would like to use. This decision can be based either on the desired font size, or on the hierarchy of importance in the heading structure that it falls into.
3. Enter the opening heading tag for the chosen heading level.

   For example: <h2>
4. Directly following the opening heading tag, enter the heading text.

   For example: <h2>**The Hypocratic Oath**
5. Directly following the heading text, enter the closing tag that corresponds with the heading level that you entered for your opening tag.

   For example: <h2>The Hypocratic Oath**</h2>**
6. Save your work and test it in a browser.

Create Lists

You have used HTML to better organize information on a Web page by categorizing the information under headings. In many cases, headings won't be an appropriate method of categorization. HTML also provides you with another method of organizing informationby displaying it as a numbered or bulleted list of items. In this topic, you will create lists so that content is formatted as an ordered or unordered list.

Lists are commonly used on Web sites. A client may ask you to create a bulleted list of features for some of their products, or maybe you need to create a numbered list to show ordered steps for the procedure of filling out a product order form. With the ability to create unordered and ordered lists, HTML has you covered.

Create Lists

Procedure Reference

To create a list on a Web page:

1. Determine where you want the list to be located.
2. Determine whether you want an ordered (numbered) or unordered (bulleted) list.
3. Enter the appropriate opening tag for the list that you chose (`<ul>` for a bulleted list or `<ol>` for an numbered list).
4. Below the list's opening tag, enter the opening tag for a list item (`<li>`).

   For example:

```
<ul>
  <li>
```

6. Enter the text for the first list item, followed by the closing list item tag (`</li>`).

   For example:

```
<ul>
  <li>George</li>
```

8. Continue adding list items between `<li></li>` container tags.

   For example:

```
<ul>
  <li>George</li>
  <li>John</li>
  <li>Paul</li>
  <li>Ringo</li>
```

13. Add the closing tag that matches the list type you chose (`</ul>` for an bulleted list or `</ol>` for an numbered list).

    For example:

```
<ul>
  <li>George</li>
  <li>John</li>
  <li>Paul</li>
  <li>Ringo</li>
</ul>
```

19. Save the file and test your work in a browser.

Bullet and Numbering Options

When creating lists, you have a few options regarding the styling of bullets in an unordered list and

the numbers of an ordered list. These styles can be modified by applying the `type` attribute to either the `<ul>` or `<ol>` tag, along with an appropriate value (for example: `<ul type="circle">`). Table 2-2 shows the options for an unordered list, while Table 2-3 shows the options for an ordered list.

Table 2-2: *The options for unordered list bullet styles*

Attribute Value	Renders As:
disc (default)	•
circle	○
square	■

Table 2-3: *The options for ordered list numbering styles*

Attribute Value	Renders As:
1	1, 2, 3, 4
a	a, b, c, d
A	A, B, C, D
i	i, ii, iii, iv
I	I, II, III, IV

Notate Code

As you have surfed the Web, you have undoubtedly viewed Web pages that were quite large and complicated. Those types of pages often involve a great deal of complex and fairly ambiguous HTML code to create them. A way to add structure and organization to HTML code is to use comments. In this topic, you will add comments to code so that the purpose of the code is clearly illustrated and the comments do not display when the page is viewed in a browser.

Imagine developing a large Web site for a month or so, and then your supervisor assigns you to a different project for a couple of months. Once work is finished on that project you are asked to return to work on the previous project. It is not likely that you are going to remember what you were thinking when you created some of the site's pages. Luckily, you notated the potentially confusing pages with HTML comments so that you can jump right back into working on the site with little or no hesitation.

Comments

As you know, most of the text that you enter into the code of an HTML document is displayed on a Web page. By using HTML *comments*, you can control what text that the browser will display. HTML comments work like a light switch. When the browser reads the start of a comment (`<!--`), it turns off its ability to print any of the text that follows until it reaches the end of the comment (`-->`), where it begins to print the text once again. Reasons for using comments can include:

- Adding information about the page that you don't want to be shown on the actual Web page such as the developer's name and contact information, or the date that the page was created.
- Adding notes to yourself or another developer about past, present, or future plans for a section of the HTML code.

- As a labeling mechanism for quick and easy reference about the purpose or function of segments of the code.
- As a testing tool to comment out a section or sections of existing code that you temporarily don't want the browser to display.

Notate Code

Procedure Reference

To add HTML comments to your source code:

1. Determine where you want the comment to be located.
2. Enter the opening comment tag (`<!--`), followed by the comment.

   For example: `<!-- This text will not be displayed by the browser.`
3. Directly following the comment, enter the closing comment tag (`-->`).

   For example: `<!-- This text will not be displayed by the browser. -->`
4. Save the file and test your work in a browser to verify that the comment did not display on the Web page.

Page Footers **Definition**

A *page footer* is located at the bottom of the Web page and usually contains information about the page, or the company or organization that created, maintains, or owns the site. It may also include navigation, notices, copyright information, or links pertaining to privacy and terms of service.

When developing pages that are going to be used by a large audience, it is a good idea to include sections that users are familiar with from their past experiences. The types of sections that are included on most Web pages and are easy to identify are headers, navigation bars, main content areas, and page footers.

Example

Figure 2-2 shows examples of page footers from some popular sites on the Web. Although they are styled differently, you will notice that they include many similar attributes.

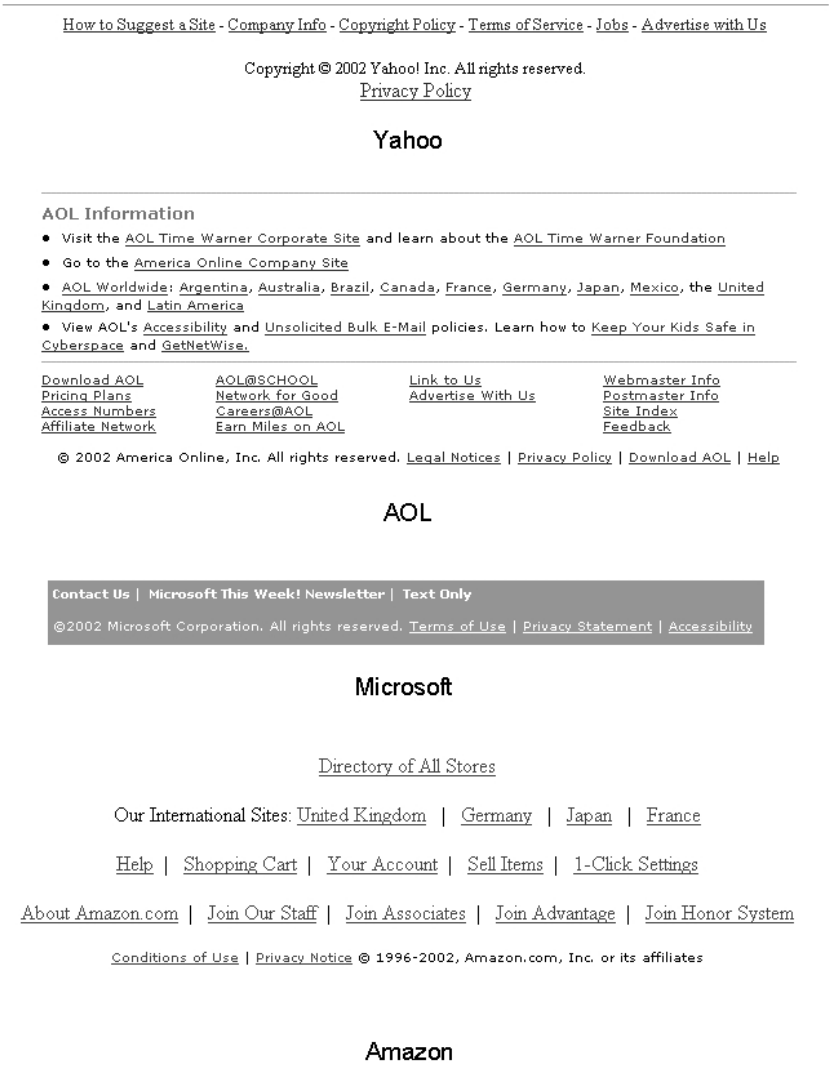


Figure 2-2: ***Examples of page footers from some popular sites on the Web.***

Insert Horizontal Rules

As you add more text and graphics to your Web pages, you may notice that things get a little crowded. To structure Web pages so that they aren't cluttered and messy, it is a good idea to separate a page into sections. In this topic, you will visually separate a section of a Web page using a horizontal rule.

Most Web pages have multiple sections that serve different purposes. Examples of these types of sections that commonly exist on the same page are navigation bars, headers and footers, sidebars, and main content areas. You can insert a horizontal rule to create visual separation between sections of a Web page that are stacked too close together.

Insert Horizontal Rules

In the previous topic, you noted that many Web pages contain sections, such as navigation bars and page footers, that have common characteristics so that users will feel comfortable with them when they are encountered. On a Web page, especially a page that is crowded, it is important to visually separate these sections to clearly define them and to prevent them from interfering with each other. Horizontal rules provide the type of visual separation that makes the sections of a Web page clearly defined.

Procedure Reference

To add a horizontal rule to a Web page:

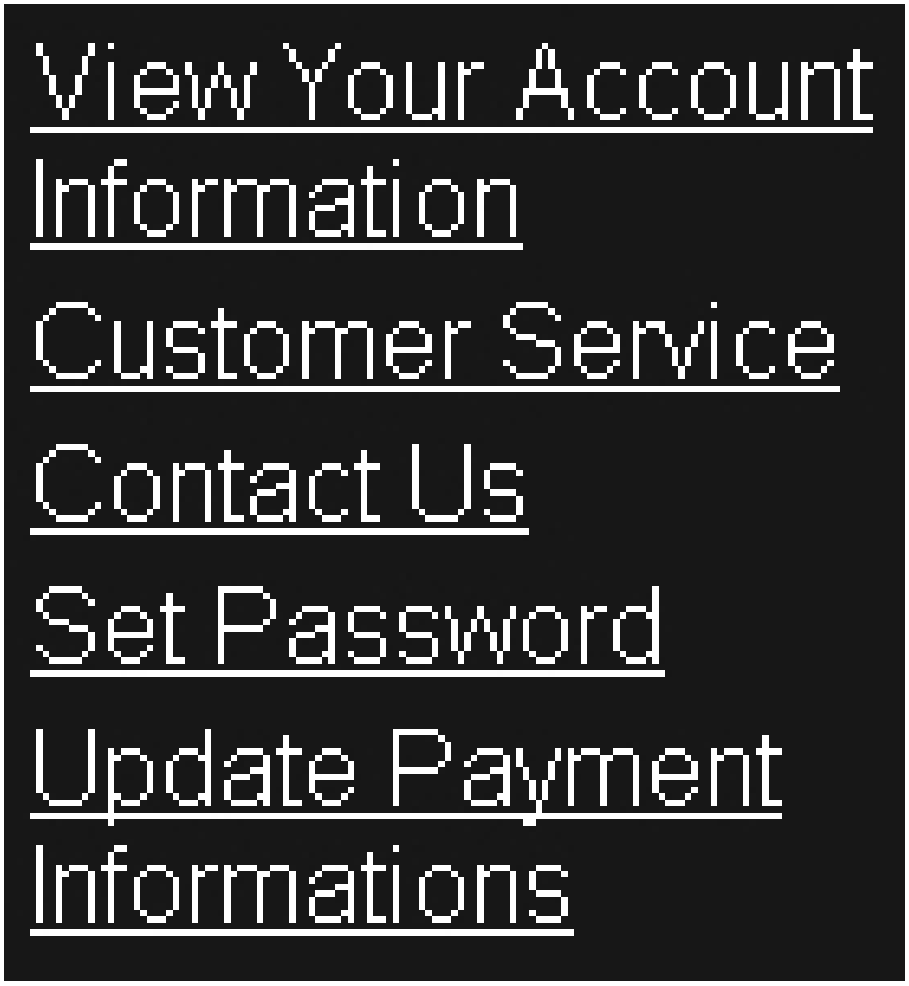
1. Determine where you would like the horizontal rule to be located.
2. In the chosen location, enter one <hr> tag for every horizontal rule that you want to create.
3. Save the file and test your work in a browser.

Control Line Breaks

As you noticed when you created paragraphs in the previous lesson, HTML does not recognize extra spaces or line breaks in your text editor when you add code to the body of a Web page like a word processing program does. This often results in a case where the default structure of the line breaks on your Web page may not suit your design. HTML offers ways to override the default line breaking behaviors. In this topic, you will use HTML markup to create and prevent line breaks so that the default behaviors are overruled.

Suppose that you have all your text and images marked up with HTML just the way you want them. When you actually take a look at the results in the browser, you may find that certain sections are too close together. HTML allows you to add line breaks wherever you need them.

The following graphic shows an example of where the browser has created line breaks where you don't want it to. This commonly occurs when you decrease the size of the browser window. Undesirable line breaks in this navigation bar has caused these five links, at first glance, to appear as seven links. You will see how HTML allows you to gain more control over your layout by preventing undesirable line breaks from occurring.



Control Line Breaks

Procedure Reference Creating Line Breaks

To create a line break:

1. Determine the location where you want to create page breaks.
2. At the location that you determine, enter the <br> empty element once for every line break that you want to create.
3. Save the file and test your work in a browser.

Procedure Reference Preventing Line Breaks

To prevent a line break from occurring:

1. Determine the location where you want to prevent the browser from creating a line break.
2. At the location that you determine, type   to enter a non-breaking space.

Although, in this case, non-breaking spaces are being used to prevent a line from breaking, they can also be used to enter more than one space between text and elements.

3. Save the file and test your work in a browser.

Group Elements

You have created Web pages that have sections made up of multiple HTML elements that work together for a common purpose. Some of these types of sections include navigation bars, headers, and footers. Using HTML, you can group these types of sections together to create a logical structure of elements so that you can manipulate them as one entity. In this topic, you will group a section of elements in the source code so that an applied attribute affects all the grouped elements.

Suppose that you wanted to center align all of the content within the page footer. One way to do this would be to apply center alignment to each of the individual elements that make up the footer. This can be somewhat tedious considering the number of elements that make up the footer.

Logical Divisions

The intent of the `<div>` container element is to enable HTML authors to logically group elements. In other words, it is used to divide one set of related content from other content or divisions of content. The only visible effect of using the `<div>` element on its own is a line break at both the start and end of the division. Its most common use is to control the alignment of a section of the Web page by applying the `align` attribute. Just as with paragraphs, the `align` attribute accepts left, right, and center as values. The more advanced uses for the `<div>` element are outside the scope of this course.

Group Elements

Procedure Reference

To group elements:

1. Identify the elements that you would like to group together.
2. Directly before the first element that you would like to group, enter the `<div>` tag along with any attributes that you want to apply to the group.

   For example:

```
<div align="right">
<img src="images/sm_thnx.jpg" height="45" width="60"
alt="Thank You">
<p>SomeCo Inc. would like to thank everyone who attended last
week's charity event for making it a huge success!</p>
```

5. Directly after the last element that you would like to group, enter the `</div>` tag.

   For example:

```
<div align="right">
<img src="images/sm_thnx.jpg" height="45" width="60"
alt="Thank You">
<p>SomeCo Inc. would like to thank everyone who attended last
week's charity event for making it a huge success!</p>
```

```
</div>
```

9. Save the file and test your work in a browser to verify that any attributes that you applied to the tag have been applied to the grouped elements.

Linking Web Pages

Lesson Objectives

In this lesson, you will create links to Web pages.

You will:

- Create links to other locations on your Web site.
- Create links to locations outside of your Web site.
- Create an image link.

Introduction

In the previous lessons, you added text and graphic content to create Web pages. Now that you know how to create them, the next logical step is to link them together. In this lesson, you will create HTML links, which allow you to connect to other files within your site and out to the entire World Wide Web.

Unless you want to attempt to fit the entire contents of your Web sites onto a single page, you are going to want to divide your sites into separate, logical pages. Like turning a page in a book, links provide your users with the ability to navigate between Web pages. Including links from your site to other Web sites will also give your users a place to go once they are finished viewing your site. This idea of jumping from page-to-page is what surfing the Web is all about.

Create Local Links

Now that you have the ability to create Web pages, the next logical step is to create links that allow users to navigate between them. Since HTML allows you to create links with text, you are halfway to creating site navigation. In this topic, you will transform the static text into hypertext links, so that a different page from the same site is displayed when a user clicks the link.

It won't be too often that a client comes to you looking for a one-page Web site—especially since, by definition, a Web “site” must contain multiple pages. In order for a user to navigate from one page to another on the same site, you will need to create links.

Hypertext Links

Links are hotspots for your mouse that, when clicked, connect you to the file that the link references. The text that is used to create the link is called *hypertext*. It is highlighted, by default, using blue underlined characters. The highlighting is a cue to the user that the text is an available link. When hovered over a link, a mouse pointer will typically change into a hand with its index finger extended as if to press a button on the Web page. A *local link* is a type of link that is used when the linked files are both located on the same computer and, typically, the same site.

Description	Opening Tag	Closing Tag

Transforms the text between the tags into a link (hotspot) that references a named file that is located on the same computer. The value of the `href` attribute is the filepath to the document you want to link to.	`<a href="value">`	`</a>`

Create Local Hypertext Links

Procedure Reference

To create a local hypertext link:

1. Determine which text you want to use for the link.
2. Directly before the link text, enter the <a> tag. Be sure to include the `href` attribute with the file path to the linked file.

   For example: **`<a href="mssgbrd.html">`**`Message Board`
3. Immediately following the link text, enter the tag.

   For example: `<a href="mssgbrd.html">Message Board`**`</a>`**
4. Save the file and test your work in a browser.

Site Maps **Definition**

A *site map* is a representation of the document structure of a Web site. Some site maps are graphical depictions of the hierarchical structure, much like a flowchart, while others are merely text links organized into categories. A simple site with only a few pages and a simple navigation scheme will generally not require a site map, but as your sites become more complex they can be a useful tool for you and your users. Creating a site map involves identifying the major directories and then identifying the pages that will make up these directories. Once you have these directories organized into categories, simply create a Web page that links users to each category.

Example

Lycos has an example of a site map containing links to the various directories of their Web site. The site map can be found for at ***http://www.lycos.com/sitemap.asp***. Figure 3-1 shows an example of a graphical site map for a site with four pages.

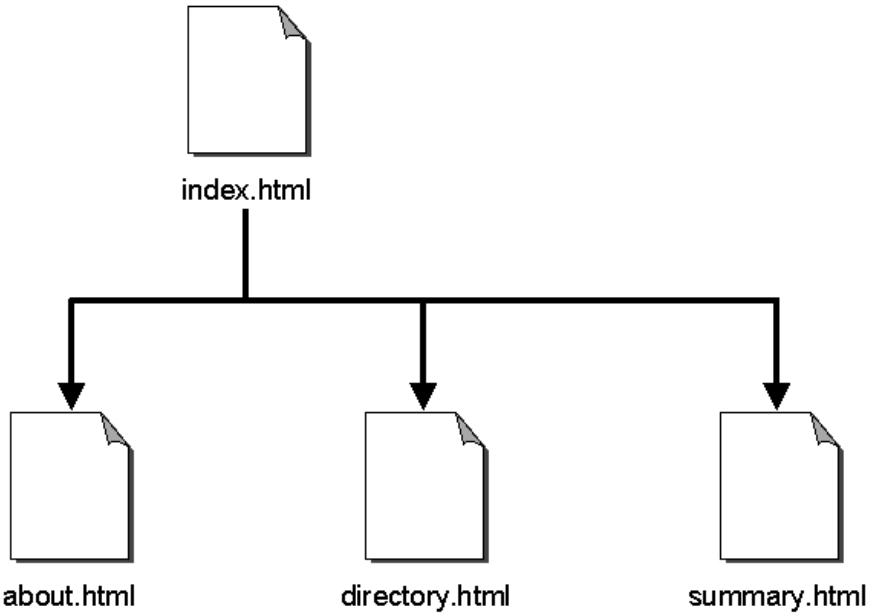


Figure 3-1: ***A graphical site map for a Web site with four pages.***

Create Image Links

You have transformed static text into hypertext links that transport users from page to page on the Web, but you may not always want to use text. Instead of text, you have the option to create a link using an image. In this topic, you will create an image link so that a different page is displayed when a user clicks the image.

Creating links with hypertext will always be an available option, but using a graphic as a link can offer the same functionality with increased aesthetic appeal. This technique is commonly used to create graphical navigation bars and banner ads. Many users think that banner ads are ugly and annoying, but advertisers love them and there is no doubt that if you do professional development work you will be adding them to your sites in one form or another.

Create Image Links

Procedure Reference

To create an image link:

1. Determine which graphic you want to use for the image link.
2. Directly before the <img> tag, enter the <a> tag. Be sure to include the href attribute with the name of the local file or the address of the remote site that you want to link to.

   For example: `<a href="http://www.books-2-you.com/"><img src="b2ubanner.gif" width="300" height="90" alt="Books-2-You.com">`
3. Immediately following the <img> tag, enter the tag.

   For example: `<a href="http://www.books-2-you.com/"><img src="b2ubanner.gif" width="300" height="90" alt="Books-2-You.com"></a>`
4. If you would like to remove the blue border around the image link, add a border attribute to the image that has a value of 0.

   For example: `<a href="http://www.books-2-you.com/"><img src="b2ubanner.gif" width="300" height="90" alt="Books-2-You.com" border="0"></a>`
5. Save the file and test your work in a browser.

Formatting Text

Lesson Objectives

In this lesson, you will modify the appearance of text.

You will:

- Modify text so that it appears bold or italic.
- Modify the font face, size, and color of text.
- Create a style sheet embedded in a Web page.
- Enter designated HTML entity codes that render as characters when viewed in a browser.

Introduction

In the previous lessons, you added a good deal of text content to your pages that is formatted using the browser's default styles. Although text that is formatted with the default styles is typically legible, you have many options to change them to your liking. In this lesson, you will modify the appearance of text so that it is different than the default settings.

While the browser's default formatting styles are generally relevant to their function on a page, they are often inappropriate for your particular design. HTML styles and an additional technology give you the flexibility to modify the appearance of text such as its size, font-face, and color so that it fits your design needs.

Apply Bold and Italics

You have added text to a Web page that is formatted using the browser's default styles. To bring attention to specified text, HTML offers options to format it so that it stands out from the default style. In this topic, you will modify text so that it displays in a browser as bold or italic.

Suppose that you wanted to make some text stand out on the Web page that you were designing. To draw the users eye to certain significant words or sentences, you can make modifications to the formatting of the text to differentiate it from the rest of the text content on a page. HTML offers the ability to do this by applying a bold or italic font style.

Apply Bold and Italic Text Styles

Procedure Reference Applying a Bold Text Style

To modify text so that it displays as bold:

1. Determine which text that you want to display as bold.
2. Directly before the chosen text, add the opening bold tag ().

   For example: ****The Daily Newsletter
3. Following the chosen text, add the closing tag ().

   For example: The Daily Newsletter****
4. Save the file and test your work in a browser.

Procedure Reference Applying an Italic Text Style

To modify text so that it displays as italic:

1. Determine which text that you want to display as italic.
2. Directly before the chosen text, add the opening bold tag ().

   For example: ****Monday, November 13
3. Following the chosen text, add the closing tag ().

   For example: Monday, November 13****
4. Save the file and test your work in a browser.

Nesting Tags

As you write HTML code, you will find occasions when you need to apply more than one tag to a block of text. This can be accomplished by nesting tags. *Nesting* is the practice of placing one set of tags within a set of container tags. Properly nested tags need to be ordered correctly to achieve consistent results. An easy way to check for correct nesting is to draw lines (either mentally, or on paper) from all the opening tags in the nested block of content to its corresponding closing tag. In a properly nested block of content, none of the lines should cross. Figure 4-1 shows examples of properly and improperly nested HTML container tags.

Figure 4-1: ***Properly and improperly nested HTML container tags.***

In the first example, the lines do not cross because the <I> and </I> tags remain on the outside, containing the <B> and </B> tags. Not so in the second example! Neither the outermost nor innermost tags match; lines drawn between corresponding tags cross over each other. It may seem unnecessary to use this method to check for proper nesting with an example that is this simple, but as the amount of code you use grows, so does the need to keep organized.

Modify Font Styles

You have seen how to format text as bold and italics, but HTML offers more options to modify the appearance of text. Applying these options can really make a noticeable difference in the appearance of text on a Web page. In this topic, you will modify the font face, size, and color of text.

Image that you have created a Web page for an older audience. This page also happens to have a dark background color applied. HTML styles will allow you to make the text of the page easier to read by increasing the font size and applying a font face and color that is easier to view against the dark page background.

Font Families

Using HTML font styles, you have the ability to specify font families. A *font family* is the name of a specific font, such as Times New Roman or Arial, and a list of all of its variations (bold, italic, etc.). When applying a font family to the text of your Web pages, you have the option to specify a list of fonts. If the browser is unable to locate a specifically-named font on the user's system, the browser will go down the list until it finds a font family that the user's computer does have.

To achieve a consistent appearance for all potential viewers, it's important to include a generic font family at the end of the font family list in case none of the listed font families are located.

In addition, it's a good idea to choose fonts that are common on the Windows and Macintosh platforms. Since many Windows fonts aren't likely to be found on a Macintosh, and vice-versa, you should include fonts that are similar to each other. This ensures that your intended font, or at least something similar to it, will display on different platforms.

Serif and Sans-Serif Fonts

Serif fonts are fonts with letters that have “decorations” (often called flourishes) that help to distinguish the letter. Serifs are usually applied to the ends of a letter. Examples of serif fonts are Times New Roman, and Georgia. *Sans-serif fonts* are fonts without decorations. Examples of sans-serif fonts include Helvetica, Arial, and Verdana. Figure 4-2 shows the decoration differences that distinguish serif and sans-serif fonts.

Figure 4-2: ***The decoration differences that distinguish serif and sans-serif fonts.***

Generic Families

A *generic family* describes a type of font, rather than a specific font face. Generic font families include monospace, serif, sans-serif, cursive, and fantasy.

Font Sizes

Font sizes can be specified as either absolute or relative. An *absolute font size* can be any integer between 1 and 7. The default absolute font size is 3. Specifying a number higher than 3 will increase the size of the text on your Web page, while specifying a number less than 3 will result in smaller text. Figure 4-3 shows the relationship between absolute font size values.

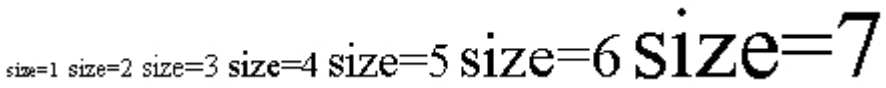


Figure 4-3: ***The relationship between absolute font size values.***

Relative font sizes determine the size of the text relative to the default font size. Relative font sizes values are based on a scale from -4 to +4 with 0 being the equivalent of the default absolute font size value. A relative font size value of +1 will result in text that is one increment larger than default. A font size of -2 will result in a font size that is two increments smaller than default. Figure 4-4 shows the relationship between relative font size values.

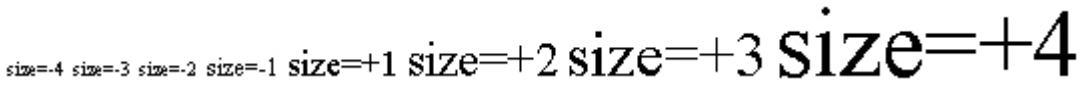


Figure 4-4: ***The relationship between relative font size values.***

Modify Font Styles

Procedure Reference Modifying Font Face

To modify the default font face:

1. Determine the fonts that you would like to apply.
2. Locate the text that you want to modify.
3. Directly before the text, enter the `<font face="value">` tag.

   For example: **`<font face="Verdana, Arial, sans-serif">`**`Web Design`

   
   When specifying specific font names, it is important to spell the font name exactly as it is displayed on the system or else that font name will be ignored.

4. Following the text that you want to modify, enter the `</font>` tag.

   For example: `<font face="Verdana, Arial, sans-serif">Web Design`**`</font>`**
5. Save the file and check your work in a browser.

Procedure Reference Modifying Font Size

To modify the default text size:

1. Determine the size that you would like to apply.
2. Locate the text that you want to modify.
3. Directly before the text, enter the `<font size="value">` tag.

   For example: **`<font size="5">`**`Web Design`

   
   Remember that you can specify font size using either absolute or relative values.

4. Following the text that you want to modify, enter the tag.

   For example: Web Design****
5. Save the file and check your work in a browser.

Procedure Reference Modifying Font Color

To modify the default text color:

1. Determine the color that you would like to apply.
2. Locate the text that you want to modify.
3. Directly before the text, enter the tag.

   For example: ****Web Design

Remember that you can specify colors using color names or hexadecimal notation.

4. Following the text that you want to modify, enter the tag.

   Web Design****
5. Save the file and check your work in a browser.

Create an Embedded Style Sheet

You have seen how to do quite a bit of formatting strictly using HTML. Although this can have a profound affect on the appearance of your Web pages, there are more powerful styling options available to you. In this topic, you will discover ways to format text using Cascading Style Sheets.

Imagine that you have used HTML styles to apply formatting to a large amount of text on a Web page. This requires that you apply the <font> tag directly to every instance of modified text. With a lot of text, this can become pretty tedious and it adds a lot of extra code. Creating an embedded style sheet gives you the ability to make more powerful styling changes with less coding than HTML styles.

Cascading Style Sheets

Cascading Style Sheets (CSS) is a set of formatting rules that allows you to define the presentation of a Web page. It does not require any additional hardware or software other than what you already use to code HTML. CSS is a technology that is complementary to HTML and is **NOT** part of the original documentation for HTML 4.01 (other than an occasional reference to it).

Cascading Style Sheets is complementary to HTML and is a more complex subject than what is being introduced to you in this lesson. Cascading Style Sheets (Second Edition) is a separate course you can take if you are interested in learning more about CSS.

Why Use CSS?

CSS gives the creator of the site a type of control once reserved for typesetters and layout technicians. It eliminates many of the formatting-related HTML tags that clutter the source code and

lets you, the author, easily concentrate on the content structuring elements that remain. Using CSS is a more powerful method than relying on HTML styles to dictate where elements should go and how they should be rendered. The main advantages of using CSS are the following:

- It separates the style from the structure
- It has the ability to globally manipulate elements
- Its layout capabilities (single-pixel accuracy)

Since the inception of CSS, it has been used in many different ways to do things with HTML documents that could either not be done previously, or normally used large amounts of HTML coding to “hack out” the same solution that a few lines of CSS code can do. This is where most of CSS's power lies—in its efficiency. A few simple line of CSS code can affect unlimited quantities of HTML code.

Browser Support for CSS

Now here's a little bit of bad news. Although the majority of CSS is supported by the most popular browsers, some of it is interpreted differently. Even subtle rendering differences can cause problems for Web developers, who need to consider all audiences. The good news is that as time goes on, support for these features will increase and become more consistent, making CSS an even more powerful tool. Version 6 of both Internet Explorer and Netscape do a terrific job of supporting CSS styles and, as these (and newer) browsers become increasingly prevalent, so should your confidence in utilizing CSS.

Since style sheets are fairly complex and are not the focus of this course, you will only be touching on a portion of their capabilities.

Deprecated Elements and Attributes

In its earliest form, HTML was strictly a means to re-create simple hardcopy document components, such as paragraphs and tables. As time passed, authors found the need to increasingly enhance the presentation of their Web documents; therefore, many elements and attributes were added to the HTML specification that allowed Web authors to apply formatting to their documents. As the specification became cluttered with formatting elements, the group that governs HTML realized that there was a need to develop a separate formatting method. They responded to this need by deprecating many of the elements used for formatting in favor of Cascading Style Sheets (CSS).

Deprecating an element or attribute is discouraging its use in favor of newer formatting practices. An item designated as being deprecated may eventually become obsolete. The reason for the deprecation of formatting in HTML is to get it back to its intended function—document structure. As you continue developing your Web authoring skills, you should attempt to reduce the number of deprecated elements and attributes that you use in favor of style sheets.

Style Sheet Options

Your main style sheet options are inline, embedded, linked, and imported. In this course, you will be focusing on embedded style sheets. An *embedded style sheet* (also known as an internal style sheet) is a rule or set of rules that is contained within the head section of an HTML document and that strictly pertains to that document. They are easily accessible and can be viewed in the source code of a selected page.

The main advantage of an embedded style sheet over HTML font styles is that you can change the value of a property throughout an entire document by changing one style rule. For example, you have already seen that you can specify the font of an element by using the `<font face="value">` tag (where “value” is the name of the font). Every element having a non-default, specified font needs to have one of these font tags attached to it. The biggest drawback to this method comes into play when you want to make changes. If you want to change all the text on a Web page from Arial to Verdana, you must go into the code and adjust every font tag to specify the new font name. This can be quite time consuming if you have to change hundreds of font tags.

A solution to this is style rules embedded within the head section of a Web page. One style rule in a page can change the properties (such as the font face) of multiple elements within the same page. Use the previous example—if the style rule has been defined effectively, then all the elements

containing text that needs to be changed to the Verdana font will be changed by modifying only that one style rule, thus saving you time.

Embedded Style Sheet Container Syntax

An embedded style sheet container contains the style rules that you want to apply to the elements of a Web page. The bold text in the following block of code is the syntax for an embedded style sheet container:

```
<head>
  <style type="text/css">
  <!--

  -->
  </style>
</head>
```

An embedded style sheet container will always be held within the head section of a document. Notice the `<style type="text/css">` opening tag in the previous example. This tag tells the browser to expect some style rules to follow. Since older browsers do not recognize the `<style>` element, it is a good idea to place style rules within HTML comment tags otherwise, older browsers might display style rules as literal text.

Style Rules

If you're new to CSS, you will soon find out that learning its syntax is reasonably simple. Style sheets establish rules that declare how certain elements are rendered on a page.

Every style rule begins with a *CSS selector*, which is the element that is connected with a particular style. Any valid HTML element (such as `body`, `p`, or `h3`) can be a selector. You may also apply the same styles to multiple selectors by separating them in the style rule with a comma.

A declaration applies the style to the selector; it consists of braces that contain a property or series of properties. Each property has an associated value, an exact specification of the property. The selector proceeds the declaration, which consists of everything inside the braces (see Figure 4-5).

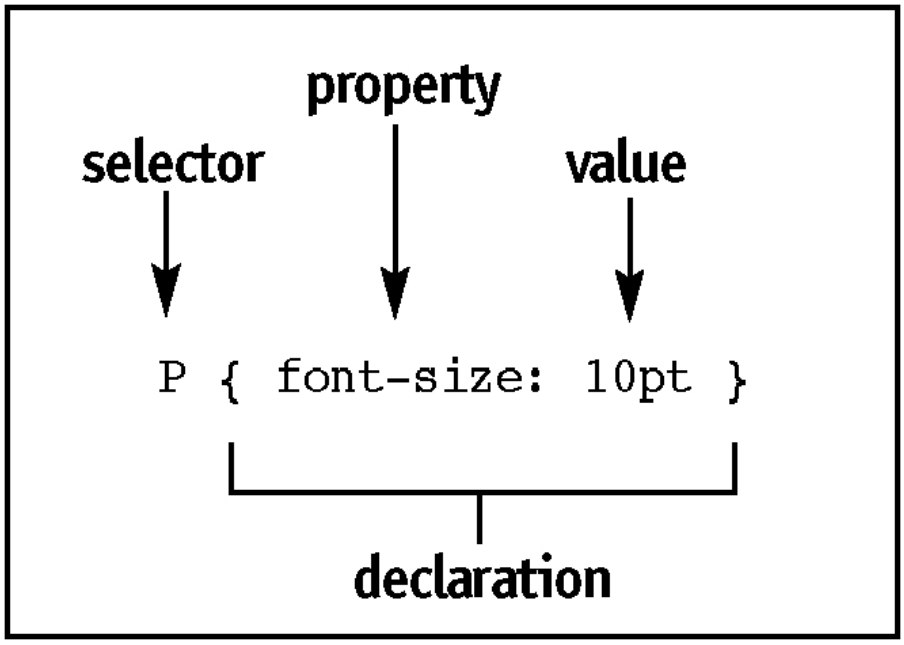


Figure 4-5: ***The components of a CSS style rule.***

When reading a style rule, compare it to identifying the grammatical components of a sentence. Take the following style rule for example:

```
h2 { font-family: Arial, sans-serif; }
```

The h2 represents the second-level heading element, which is the noun. The noun is being described by the value of the font-family property as having either an Arial or sans-serif family. Therefore, the Arial and sans-serif families are the adjectives. The verb is implied because you know that the purpose of the rule is to apply a style. Look at how this rule is applied in the embedded style sheet container:

```
<head>
<style type="text/css">
<!--
    h2 { font-family: Arial, sans-serif; }
-->
</style>
</head>
```

In plain terms, applying this style rule will cause all h2 level headings to be displayed with the Arial font if the browser can find it on the user's computer, otherwise the generic sans-serif font will be used.

Create an Embedded Style Sheet

Procedure Reference

To create an embedded style sheet:

1. In your text editor, place your text insertion point within the head section of the document.
2. Type `<style type="text/css">` and press [Enter].
3. Type `<!--` and press [Enter] twice.
4. Type `-->` and press [Enter].
5. Type `</style>`.
6. Between the HTML comment, add any valid CSS style rules.

   For example:

```
  <head>
  <style type="text/css">
  <!--
    p { font-family: Georgia, Times New Roman, serif; }
  -->
  </style>
</head>
```

> Despite the HTML comment tags, the browser still processes the text that is recognized as a style rule even though it won't print the rule to the Web page as literal text.

13. Save the file and test your work in a browser.

Insert HTML Entities

You have created and formatted text on a Web page. To enhance your Web pages with symbols and characters that aren't available as keyboard commands, you can use HTML entities. In this topic, you will enter HTML entity codes so that characters that do not have keyboard commands can be rendered on your Web page.

Not all of the characters that you can create on a Web page are available using keyboard commands. Imagine that you wanted to add a bullet or copyright symbol. Can you find the bullet or copyright key on your keyboard? I don't think so. HTML entity codes allow you to add symbols such as these to your Web pages.

HTML Entities

Definition

HTML entities are a set of symbols that are defined by either character or numeric codes. They allow you to display a greater range of characters on your Web pages than what is available as keyboard commands. The also allow you to display reserved characters that the browser would normally interpret as HTML.

Strategically replacing HTML's reserved characters (such as <, >, and /) with entity codes enables you to do things like display HTML code examples on your Web pages without the browser processing the text as actual code.

Example

The following table lists some of the most common entities and their character and numeric codes:

Character	Description	Char. Code	Numeric Code
©	Copyright	©	©
™	Trademark	™	™
®	Registered Trademark	®	®
•	Bullet	•	•
<	Less than	<	<
>	Greater than	>	>
¢	Cent	¢	¢

Character and Numeric Entity Codes

Virtually all HTML entities have a numeric entity code. Typically the numeric symbols are better-supported in browsers. Unfortunately, it is much more difficult to remember the numbers than it is to remember the character codes. Character codes are easier to remember because they typically resemble the name of the character for which they represent. For example, the character code for the cent (¢) symbol is `¢`, while the numeric code for the same symbol is `¢`. The good news is that you probably won't encounter too many browser-support issues with the character codes for the more common entities. As always, be sure to test your HTML entity codes in both Internet Explorer and Netscape browsers.

Insert HTML Entities

Procedure Reference

To display an HTML entity on a Web page:

1. Determine the location that you want the entity to display.
2. Either from memory or from reference, enter either the character or numeric code for the desired entity.

   For example: `The price of the football card is 50¢.`
3. Save the file and test your work in a browser.

Creating Tables

Lesson Objectives

In this lesson, you will create a data table.

You will:

- Create the table body by adding rows and cells.
- Create table headers.
- Modify the structure of a table by merging columns and rows.
- Align content held within a table.
- Change the background color of a table row.

Introduction

If you have seen a table in a document or spreadsheet, then you know that it is an excellent way to display many types of data. Like a spreadsheet, HTML tables use rows and cells to organize data. Also like spreadsheets, you are offered formatting options to make the data easier to identify and understand, such as bold headers, alignment, and shading. In this lesson, you will create a formatted table that contains data.

Imagine that a transit company has hired you to put their bus schedule on their Web site. Would you describe the pick-up times and locations using text in paragraphs? Probably not. Most likely you would display this type of data in a clearly organized and easy-to-reference table.

Create the Table Body

Tables consist of rows and columns with data contained inside of the cells that are created where these rows and columns intersect. Before you can display data in a table, you need to create the table body. In this topic, you will add rows and cells to create a table body that accurately structures data.

You have been handed a table containing multiple rows and columns of data. HTML allows you to reproduce the organization of this data on a Web page by defining rows and cells in the body of a table.

HTML Table Bodies

The components of the body of an HTML table are rows and cells, which are all contained within tags that define many of the attributes of the table. Table 5-1 shows the tags and attributes used to construct an HTML table body.

Table 5-1: *The HTML tags used to construct the body of a table*

Description	Opening Tag	Closing Tag
Instantiates a table. • The default border width is 0. • Table width may be specified as an absolute value or as a percentage of the width of the browser window.	`<table border="value" width="value">`	`</table>`
Defines a table row.	`<tr>`	`</tr>`
Defines a table cell.	`<td>`	`</td>`

Create the Table Body

Procedure Reference

There are many variations on the order of operations when creating an HTML table body. Over time you will find one or two methods that suit you. The following is one method to create a table body:

1. Determine where you want the table to be located.
2. In your text editor, at the chosen location, type `<table border="value">` and press [Enter] twice. Add any additional attributes to this that you may need.
3. To close the table container, type `</table>`.
4. On the blank line between the tags that you entered previously, create the first row by typing `<tr>`, pressing [Enter] twice, and then typing `</tr>`.

   For example:

```
<table border="1">
<tr>

</tr>
</table>
```

9. On the blank line between the table row tags, press the spacebar twice and create the first table cell by typing `<td></td>`.

   For example:

```
<table border="1">
<tr>
  <td></td>
</tr>
</table>
```

Adding the extra spaces before the cell isn't required, but it may help you to

visualize the structure of the table. Since extra spaces won't affect the appearance of the final table, feel free to use them as a mechanism to make your table code easier for you to read.

14. Add a second cell to the first row.

For example:

```
  <table border="1">
  <tr>
    <td></td><td></td>
  </tr>
</table>
```

19. Repeat the steps to create rows and cells until the table fits the structure and quantity of the data that you want to display.

For example, this table has three rows, with three cells per row:

```
  <table border="1">
  <!-- Row 1 -->
  <tr>
    <td></td><td></td><td></td>
  </tr>
  <!-- Row 2 -->
  <tr>
    <td></td><td></td><td></td>
  </tr>
  <!-- Row 3 -->
  <tr>
    <td></td><td></td><td></td>
  </tr>
</table>
```

The HTML comments have been added to label the individual table rows. They are not required, but they can help you to read the table code.

33. Add your data to the table cells.

For example:

```
  <table border="1">
  <!-- Row 1 -->
  <tr>
    <td>March</td><td>14</td><td>$2400.00</td>
  </tr>
  <!-- Row 2 -->
  <tr>
    <td>April</td><td>5</td><td>$950.00</td>
  </tr>
  <!-- Row 3 -->
  <tr>
    <td>May</td><td>9</td><td>$1400.00</td>
  </tr>
</table>
```

47. Save the file and test your work in a browser.

Create Table Headers

Although you have created the body of a data table, it is still important to ensure that it is easy to understand. HTML includes the ability to create headers that can be used to categorize the data held in the table. In this topic, you will add header cells to a table body.

Imagine trying to read a complex scientific table of data. Without labels at the start of a column or row of data, the meaning of the data may not be clear. You can add table header cells that will clearly categorize the data content. The following graphic is an example of where including table headers brings meaning to the data in a table.

acres	hectacres	.4047
acres	square feet	43,560
acres	square miles	.001562
atmospheres	cms. of mercury	76

Conversion Factors		
From	**To**	**Multiply By**
acres	hectacres	.4047
acres	square feet	43,560

Create Table Headers

Procedure Reference

To create headers in a table:

1. Determine where you want the header to be located.
2. At the chosen header location, add the tags to create a table row. Don't create any cells.

   For example:

```
  <table border="1">
  <tr>

  </tr>
  <!-- Row 1 -->
  <tr>
    <td>March</td><td>14</td><td>$2400.00</td>
  </tr>
  <!-- Row 2 -->
  <tr>
    <td>April</td><td>5</td><td>$950.00</td>
  </tr>
  <!-- Row 3 -->
  <tr>
    <td>May</td><td>9</td><td>$1400.00</td>
  </tr>
</table>
```

19. Between the tags for this new row, create a table header cell by pressing the [spacebar] twice and typing `<th></th>`. Enter these tags for each header cell that you need in the row.

    For example:

```
  <table border="1">
  <tr>
    <th></th><th></th><th></th>
  </tr>
  <!-- Row 1 -->
  <tr>
    <td>March</td><td>14</td><td>$2400.00</td>
  </tr>
  <!-- Row 2 -->
  <tr>
```

```
30.    <td>April</td><td>5</td><td>$950.00</td>
31.  </tr>
32.  <!-- Row 3 -->
33.  <tr>
34.    <td>May</td><td>9</td><td>$1400.00</td>
35.  </tr>
   </table>
```

Header cells can be entered anywhere within the structure of a table. Not just the first row.

36. Add the text for the header. By default, this text will be center aligned and bold when the table is viewed in a browser.

For example:

```
37.  <table border="1">
38.  <tr>
39.    <th>Month</th><th>Units Sold</th><th>Net Profit</th>
40.  </tr>
41.  <!-- Row 1 -->
42.  <tr>
43.    <td>March</td><td>14</td><td>$2400.00</td>
44.  </tr>
45.  <!-- Row 2 -->
46.  <tr>
47.    <td>April</td><td>5</td><td>$950.00</td>
48.  </tr>
49.  <!-- Row 3 -->
50.  <tr>
51.    <td>May</td><td>9</td><td>$1400.00</td>
52.  </tr>
   </table>
```

53. Save the file and test your work in a browser.

Modify the Table Structure

You have created a table body that has an equal number of cells in each row and an equal number of rows in each column. In some cases, you may want a table row or cell to extend into an adjacent row or cell without disrupting the structure of the table. In this topic, you will modify the structure of a data table so that cells and rows are merged.

Imagine that you had a table of data that had a large amount of identical, adjacent data. Instead of entering the same data into each adjacent table cell, you can show that the same data applies to multiple rows or columns by spanning the rows and cells. This will present the data more clearly, while decreasing the amount of required HTML code.

Modify the Table Structure

Procedure Reference

The structure of a table can be modified by merging cells across rows or columns. To modify the structure of a table:

1. Determine which cells or rows that you would like to extend.
2. To extend cells horizontally between columns, add the `colspan="value"` attribute to the cell that you would like to keep.

   For example:

```
  <table border="1">
  <tr>
    <th>Month</th><th>Start Date</th><th>End Date</th><th>Net
Profit</th>
  </tr>
  <!-- Row 1 -->
  <tr>
    <td>March</td><td
colspan="2">13</td><td>13</td><td>$400.00</td>
  </tr>
  <!-- Row 2 -->
  <tr>
    <td>April</td><td>7</td><td>12</td><td>$2000.00</td>
  </tr>
  <!-- Row 3 -->
  <tr>
    <td>April</td><td>16</td><td>23</td><td>$2700.00</td>
  </tr>
  <!-- Row 4 -->
  <tr>
    <td>April</td><td>25</td><td>28</td><td>$950.00</td>
  </tr>
</table>
```

23. Delete the code for the cells that will be replaced by the extended cell.

    For example, the grayed out code should be deleted:

```
  <table border="1">
  <tr>
    <th>Month</th><th>Start Date</th><th>End Date</th><th>Net
Profit</th>
  </tr>
  <!-- Row 1 -->
  <tr>
    <td>March</td><td
```

```
   colspan="2">13</td><td>13</td><td>$400.00</td>
31.  </tr>
32.  <!-- Row 2 -->
33.  <tr>
34.    <td>April</td><td>7</td><td>12</td><td>$2000.00</td>
35.  </tr>
36.  <!-- Row 3 -->
37.  <tr>
38.    <td>April</td><td>16</td><td>23</td><td>$2700.00</td>
39.  </tr>
40.  <!-- Row 4 -->
41.  <tr>
42.    <td>April</td><td>25</td><td>28</td><td>$950.00</td>
43.  </tr>
   </table>
```

44. To extend cells vertically between rows, add the rowspan="value" attribute to the cell that you would like to keep.

    For example:

```
45.  <table border="1">
46.  <tr>
47.    <th>Month</th><th>Start Date</th><th>End Date</th><th>Net
   Profit</th>
48.  </tr>
49.  <!-- Row 1 -->
50.  <tr>
51.    <td>March</td><td colspan="2">13</td><td>$400.00</td>
52.  </tr>
53.  <!-- Row 2 -->
54.  <tr>
55.    <td
   rowspan="3">April</td><td>7</td><td>12</td><td>$2000.00</td>
56.  </tr>
57.  <!-- Row 3 -->
58.  <tr>
59.    <td>April</td><td>16</td><td>23</td><td>$2700.00</td>
60.  </tr>
61.  <!-- Row 4 -->
62.  <tr>
63.    <td>April</td><td>25</td><td>28</td><td>$950.00</td>
64.  </tr>
   </table>
```

65. Delete the code for the cells that will be replaced by the extended cell.

    For example, the grayed out code should be deleted:

```
66.  <table border="1">
67.  <tr>
68.    <th>Month</th><th>Start Date</th><th>End Date</th><th>Net
   Profit</th>
69.  </tr>
70.  <!-- Row 1 -->
71.  <tr>
72.    <td>March</td><td colspan="2">13</td><td>$400.00</td>
73.  </tr>
74.  <!-- Row 2 -->
75.  <tr>
76.    <td
   rowspan="3">April</td><td>7</td><td>12</td><td>$2000.00</td>
77.  </tr>
```

```
78.  <!-- Row 3 -->
79.  <tr>
80.    <td>April</td><td>16</td><td>23</td><td>$2700.00</td>
81.  </tr>
82.  <!-- Row 4 -->
83.  <tr>
84.    <td>April</td><td>25</td><td>28</td><td>$950.00</td>
85.  </tr>
   </table>
```

86. Save the file and test your work in a browser.

Align Table Content

You have created the table body and the data has been entered, but you notice that the data in one column is too close to data in another column. You can improve the presentation of the table by changing its alignment to create more separation between the data in columns. In this topic, you will align content held within a table so that the data is more clearly presented.

The following example shows two tables containing data. The data in the first two columns of the top table is unnecessarily bunched together compared to the data in the third column. The bottom table applies center alignment to the middle column to balance the space between the data, thus improving the presentation and readability of the data.

Conversion Table		
1 US pint	is equal to	0.1250 US gallons

Conversion Table		
1 US pint	is equal to	0.1250 US gallons

Align Table Content

Procedure Reference

To align data in a table:

1. Determine which cells or rows that you would like to modify.
2. To modify the alignment of an entire row, add the `align="value"` attribute to the `<tr>` tag for that row.

   For example:

```
  <table border="1">
  <tr>
    <th>Month</th><th>Start Date</th><th>End Date</th><th>Net Profit</th>
  </tr>
  <!-- Row 1 -->
  <tr align="center">
    <td>March</td><td colspan="2">13</td><td>$400.00</td>
  </tr>
  <!-- Row 2 -->
  <tr>
    <td rowspan="3">April</td><td>7</td><td>12</td><td>$2000.00</td>
  </tr>
  <!-- Row 3 -->
  <tr>
    <td>16</td><td>23</td><td>$2700.00</td>
  </tr>
  <!-- Row 4 -->
  <tr>
    <td>25</td><td>28</td><td>$950.00</td>
  </tr>
</table>
```

23. To modify the alignment of individual cells, add the `align="value"` attribute to the `<td>` tag for that cell.

For example:

```
<table border="1">
<tr>
  <th>Month</th><th>Start Date</th><th>End Date</th><th>Net Profit</th>
</tr>
<!-- Row 1 -->
<tr align="center">
  <td>March</td><td colspan="2">13</td><td>$400.00</td>
</tr>
<!-- Row 2 -->
<tr>
  <td rowspan="3">April</td><td>7</td><td>12</td><td align="right">$2000.00</td>
</tr>
<!-- Row 3 -->
<tr>
  <td>16</td><td>23</td><td>$2700.00</td>
</tr>
<!-- Row 4 -->
<tr>
  <td>25</td><td>28</td><td>$950.00</td>
</tr>
</table>
```

44. To modify the position of the entire table on the Web page, add the `align="value"` attribute to the opening `<table>` tag.

For example:

```
<table border="1" align="center">
<tr>
  <th>Month</th><th>Start Date</th><th>End Date</th><th>Net Profit</th>
</tr>
<!-- Row 1 -->
<tr align="center">
  <td>March</td><td colspan="2">13</td><td>$400.00</td>
</tr>
<!-- Row 2 -->
<tr>
  <td rowspan="3">April</td><td>7</td><td>12</td><td align="right">$2000.00</td>
</tr>
<!-- Row 3 -->
<tr>
  <td>16</td><td>23</td><td>$2700.00</td>
</tr>
<!-- Row 4 -->
<tr>
  <td>25</td><td>28</td><td>$950.00</td>
</tr>
</table>
```

65. Save the file and test your work in a browser.

Modify Table Background Colors

You have already created an HTML data table and you want to make the data easier to read. A technique that allows you to present the content of a table more clearly is to strategically modify the background color of its rows or cells. In this topic, you will change the background color of table rows so that the rows are highlighted.

Imagine that you have a large table containing data. You can apply colors to the background of this type of table to make its contents easier to understand and reference. Applying a background color to header cells increases their separation from the data, while applying shades to every other row makes it easier to navigate from one row to another without losing your place.

The Fall Semester Academic Calendar	
Trigonometry: Off on a Tangent	MAT
The Lost Comedies of Edgar Allan Poe	ENG
James Joyce for Freshmen	ENG
Deconstructing Derrida	ENG
Keats: The Later Years	ENG
Kafka: A Bug's Life	ENG
English Comp: Avoiding Cliches	ENG
W. Shakespeare Vs. J. Blume	ENG
Existentialism: The Key to Happiness	PHI

Modify Table Background Colors

Procedure Reference

To modify the background color of table cells:

1. Determine which cells you want to modify and the background colors that you want to apply.
2. Apply the `bgcolor="value"` attribute to the table, rows, or cells that you want to modify.

   In the following example, the header row will be shaded dark gray, while a single table cell in a separate row will be shaded red:

```
  <table border="1" align="center">
  <tr bgcolor="#999999">
    <th>Month</th><th>Start Date</th><th>End Date</th><th>Net Profit</th>
  </tr>
  <!-- Row 1 -->
  <tr align="center">
    <td>March</td><td colspan="2">13</td><td>$400.00</td>
  </tr>
  <!-- Row 2 -->
  <tr>
    <td rowspan="3">April</td><td>7</td><td>12</td><td align="right">$2000.00</td>
  </tr>
  <!-- Row 3 -->
  <tr>
    <td>16</td><td>23</td><td bgcolor="#ff0000">$2700.00</td>
```

```
18.  </tr>
19.  <!-- Row 4 -->
20.  <tr>
21.    <td>25</td><td>28</td><td>$950.00</td>
22.  </tr>
   </table>
```

23. Save the file and test your work in a browser.

Modifying Background Colors with CSS

The `bgcolor` attribute is one of the many HTML attributes that is being phased out of the language with preference being given to Cascading Style Sheets. Now that you know a little about CSS and the fact that it replaces elements and attributes which are being phased out of HTML, it is a good time to examine an alternate approach to modifying the background color of a table. If you remember, any HTML element can be a CSS selector. By applying the `background-color` CSS property, you could easily modify the background color of the header cells by simply specifying the `th` element as the selector as in the following example:

```
<style type="text/css">
<!--
  th { background-color: #999999; }
-->
</style>
```

This style rule causes any table header cell defined with the `<th></th>` container tags to be rendered with a dark gray background color. One important thing to note is that applying the color using this method will affect all of the header cells in any table on a Web page. If you want to apply different colors to separate tables or to separate cells within a table, you should either use the `bgcolor` attribute, or learn more about ways to apply styles with CSS.