Teach Yourself Basic HTML Web Page Design

by Amy Lindsay

Teach Yourself Basic HTML Web Page Design

by Amy Lindsay

INTRODUCTION

This book provides the most basic information you need to write a web page. Obviously, there are entire libraries full of more detailed, more extensive, and more advanced information on web pages and HTML, but writing web pages are actually easier than you might think from those “advanced” sources.

“You Should Read This Book Because...”

If you’ve opened this book because you want to learn to make web pages, you’ve come to the right place.

A few basic assumptions about your level of skill and understanding:

- You understand how your computer works (at least enough to know how to turn it on and off, how to start and shut down applications, and how to open and close files.)
- You have been online and know what a web page is (in a general sort of way, we will cover specifics in the next chapter.)
- You want to make a web page for yourself.

After reading this book and doing the tutorials, you should be able to:

- Understand how a web site works
- Create a simple web site with more than one page
- Link your pages to other sites and other pages on your own site
- Use graphics and text on your site
- Put your site on the Internet so others can see it

Tools You Will Need

The few tools you need to make a web page are:

- **A computer** —The brand or operating system of your computer does not make a difference, but the computer you are working on should be able to run the software listed below. Most people have either a PC or a Macintosh.
- **A text editor** —A simple text editor, like Notepad (PC) or SimpleText (Mac), should be part of the standard software that comes with your computer.
- **A browser** —You can use any browser you feel comfortable with. The most popular are Internet Explorer and Netscape. Most are available free for download on the Internet or by disk in various stores. For links, see Appendix C.
- **An Internet Service Provider** (ISP) —
  An ISP is your link to the rest of the Internet. Without one, you will not be able to upload your web pages for other people to see.

- **A place to put your web site so other people can see it** (a web server)—Many ISPs provide their customers with disk space to put web pages as part of their service. If yours does not, there are many free Web Servers or Hosts available (see Appendix C.)

A word about Word and other word processors...

A text editor is not a word processor. A word processor, like Word or WordPerfect adds formatting code you can't see. These codes tells your computer things like the size of the characters, which typeface to display the text, and other formatting choices you've made. When you write your web page, you want to write using text only, the unformatted letters that the computer reads. This is also referred to as ***ASCII text***.

You can use a word processor to write HTML, but you need to remember two things:

- ***You must save the document as "plain text" or "text only."***
  Do not be tempted to "save as HTML"! Not only will your word processor add unwanted codes to your file, but when you test your site in your web browser, what you see will not be what you expect.

- ***When you open your document after you have saved it, you must open it as a web page or HTML document.***
  For most word processors, if you do not tell your application what kind of file you are opening, it will default to a file for that specific application and you may not be able to see or edit your HTML code. To open a file as an HTML document, you will need to select "HTML Document" under the file type list and view it as normal or plain text.

Conventions

Throughout the book, the following conventions will be used:

- **Tags** —HTML codes are usually referred to as Tags. When I discuss the tag in the text, the tag will appear in blue, bold sans-serif type, e.g. **`<head>`**.

- **Sidebars** —You will find additional topics of interest provided in the sidebars. These are set apart from the regular text in blue boxes with sans-serif type.

- **Illustrations** —Illustrations are set aside from the regular type in green boxes and sometimes include explanatory text.

- **Tutorials** —Each chapter contains a Tutorial section. You should type everything that appears in **`green Courier Bold`**. Other important information, such as menu selections or reference text appears in **bold** .

- **Test Results** —After each tutorial, you will find a Test Results section. These sections show the entire text of the HTML file in a green box with black `courier` type.

- **Additional Study** —The Additional Study section contains suggestions for additional exercises you can do to make yourself more comfortable using the information in the chapter.

If a line of code is too long to be shown on a single line, the second line is indented as are any additional lines. You should *not* add either a hard return or an indent.

How to Use the Tutorial and Test Sections

After each chapter, you will find a short tutorial. You can work them as you reach them in the text, return to them after reading the entire book, or skip them entirely.

These tutorials allow you to gain some experience working with HTML code directly and help you understand how to make your words and graphics do what you want them to.

Each tutorial guides you through an HTML file where you will make modifications using the tags you learned in that chapter.

You may use your own graphics/image files, or download the ones from the Internet. You can find a list of sources in Appendix C.

Once the document has been saved, you should test it using your browser. The Test Results section will show you how your file should appear when accessed by a browser.

Some of the more common mistakes will be listed below the test screen.

Where to find help…

There are many HTML references available, both online and from your bookstore. Appendix C lists some links and books that you might find useful.

CHAPTER 1: STARTING OUT

HTML & The Internet

The **Internet** (sometimes called "the Net" for short) is a network of computers linked together. It started in 1969 as a US military experiment to share computer resources more efficiently. Later, it was expanded to include colleges and research facilities.

Today, the Net has grown into a massive public broadcast medium. It is an international network of mixed computer technology with more than 600 million users using several different computer languages called **protocols** . HTML is one protocol.

Where Did HTML Come From?

In March of 1989, 20 years after the Internet was "born," Tim Berners-Lee, a computer scientist at the European Laboratory for Particle Physics (CERN) in Geneva, Switzerland published a paper titled *Information Management: A Proposal.* In it, he suggested a way of managing information by linking related documents and having them all available over a computer network so physicists could share research results with each other.

At the time, **SGML** , or Standard Generalized Markup Language, was the standard format for large-scale documents accessed by researchers using computers. As more computers were networked, more documents were put online in SGML format, but SGML was unwieldy and difficult to use.

HTML, or Hypertext Markup Language, combined the wide acceptance of SGML with ease of use. Instead of many commands, HTML used only a small subset, making it easier to learn.

What was truly different about HTML, however, was the concept of **links** , or references to other documents. Each SGML document was designed to stand alone, but HTML documents are designed to refer to other documents.

These references can appear in any web page on any site, and are not limited to references to the same site. This crisscrossing of references makes the World Wide Web a web, and because of its ability to link documents, HTML became the basic language of that Web.

Learning HTML

If the origins of the Web seem to be intimidating, you should realize that millions of people create web pages each year. These are not all scientists and engineers. They include people from every industry, every profession, and nearly every age group.

With a little effort, you should be able to learn HTML because:

- **HTML is easy.** You don't need a college degree or any knowledge of computer programming. In fact, most of a web page is written in ordinary text. When you learn HTML, you are simply learning how to format that ordinary text so it can be viewed on the Web.

- **HTML is fast.** You write your page, save it, and *immediately*, you can look at it in your browser.

- **Specialized applications are not needed to write HTML.** The content and commands of an HTML file are written in plain, unformatted text. Any basic text editing software, including SimpleText and Notepad can be used to write web pages.

- **Viewing software is widely available for free.** Applications to view web pages are called **browsers** . They interpret the HTML formatting commands and display the page on your screen. Netscape Communicator and Microsoft's Internet Explorer are currently two of the most popular browsers and are available at no cost from various sites on the Internet.

The Web Development Cycle

Each document you produce in HTML is considered to be one **web page** …no matter how long or short it is. An entire group of web pages collected at one location is called a **web site** (or just "site" for short.)

There are four major stages in the web development cycle:

1. **Planning** — Writing a web page is simple, however, if you want to create a good site, you will need to put some planning into it. You need to decide what you want to say, organize your thoughts, research what you want it to look like and decide what should be linked to what. This should all be done in the planning stage.

2. **Creating** —After the web site is planned, you need to write the text and format it with HTML commands. You will also need to gather or create the graphics you will use in your site.

3. **Testing** —After each page is created, you will need to test it on one or more different browsers to make sure it looks and acts like you want it to. You can do the initial testing off-line, on your computer and correct any mistakes before you publish.

4. **Publishing** —Publishing a web site is similar to publishing a book…you make it available to anyone who wants to see it by **uploading** , or moving your finished page (or pages) from your computer to a web server. A **web server** is simply a computer whose job it is to send the file to any computer asking to look at it.

In creating complicated web pages, it's not unusual for the web page creator to write part of the page, test it, and then write more. The writing and testing stages of the web development cycle usually take the most amount of time.

Tags

What makes HTML documents different than ordinary ones is that formatting commands are written into the file. These commands, called **tags** , tell the browser how to display the document.

You can think of a browser as an obedient, very bright, but very literal, child. It will do what you say *exactly* the way you say to do it. It "knows how" to do many complicated things, but you must tell it explicitly to do something or the results will not be what you think they should be.

An HTML Tag

<	**p**	**align=**	**"center"**	**>**
opening bracket	tag	value	attribute	closing bracket

You can tell which words are tags in HTML because they have angle brackets around them, **< >**. Angle brackets look like “less than” and “greater than” signs. These brackets tell your web browser to use these commands to interpret and display your page.

The angle brackets are followed by the name of the tag itself. While most browsers do not "see" a difference between tags written in uppercase and those written in lowercase, it is considered "good practice" to write your tags in lowercase.

After the tag and within the angle brackets you may find information that modifies the tag or gives specific information that the browser needs to execute the tag. This information is called an **attribute.** Most tags have a few attributes, many of them optional.

Attributes usually need further information, called **values**. Values can be numerical (percentages or measurements) or a specific set of words (“left,” “right,” or “center.”) With a few exceptions, values are always surrounded by double quotes.

For nearly every tag, there is an equivalent **closing tag**. Closing tags tell the browser when to stop doing something. For example, the closing paragraph tag tells the browser where the end of the paragraph is.

A Closing Tag

<	**/**	**p**	**>**
opening bracket	backslash	tag	closing bracket

Closing tags consist of the square brackets, a backslash and the tag itself. You do not need to include the attributes and their values in a closing tag.

As the web has grown and matured, some of the tags have been replaced by other means of doing the same thing. The older tags still work, but the newer tags are preferred. Therefore, the World Web Web Consortium (or W3C) labels those older tags **deprecated**.

In earlier version of HTML, web designers were strongly encouraged to write tags in all capitals so the HTML formatting would stand out from the content. Current convention is the opposite: all tags, attributes, and values should in lowercase.

Whichever way you choose to write your tags, you must be consistent. Some browsers will not recognize a closing tag in all capitals as being related to an opening tag in lowercase.

Document Tags

The first tag that appears in any web page document is the opening HTML tag, **<html>** . This tells the browser to interpret everything after it as HTML.

The HTML Tags

<html>

</html>

The opening and closing HTML tag surround the text you wish browsers to interpret as HTML.

When you have finished writing the contents of the page, signal the browser that the HTML contents of the document is complete by using the closing HTML tag, **</html>** . Anything you put after the **</html>** tag will not be shown by the browser.

The Head Tags

<head>

</head>

The opening and closing head tag surround the text that provides information for the computers accessing your site.

The next tag in most web page document is the opening head tag, **<head>**. The head of an HTML document contains information used by other computers.

Like the **<html>** tag, you will not actually see the results of the **<head>** tag with one exception: the title.

Every web page needs to have a title to identify it to users. This title appears at the top of the screen but not actually on it.

The Title Tags

<title>Title text goes here</title>

Text surrounded by the opening and closing title tags appears in the top bar of the browser.

The opening title tag, **<title>** tells the browser to place these words in the gray bar at the top of the browser screen. The closing title tag, **</title>** tells the browser to stop putting words in that bar.

When using the title tag, remember to keep the title as short as possible but be as specific as you can. Remember, these words may be used in by other sites and bookmarks your visitors make while at your site to describe your page.

The opening body tag, **<body>**, signals the browser that what follows should appear in the main part of the browser screen. All the words, graphics and image tags will appear in this section of your document. The **<body>** tag has several attributes that will be discussed in Chapter 3.

The closing body tag is **</body>**.

The Body Tags

<body>

</body>

The opening and closing body tags surround everything you want to see in the main part of the screen.

Comments

You may find that you need to make notes to yourself, or leave information for anyone else who may be editing your web page in the future. You can do this using comment tags.

The opening comment tag is an opening bracket, followed by an exclamation mark and two dashes, **<!--** . Notice that the opening comment tag has no ending bracket.

After you have typed your comments, signal the end of your comments by using the closing comment tag, **-->** . Like the opening comment tag, the closing comment tag seems to be missing a bracket. This is because, technically, all the text between the opening and closing brackets is included in "one big tag."

The Comment Tags

<!-- Make your comment here -->

Surround your comments with the opening and closing comment tags.

You can make as many lines of comments as you like between the opening and closing comment tags, the browsers will ignore everything between them.

Tutorial

In this tutorial, you are going to make a very simple web page to help you identify the parts of a page and how the tags work.

Lines of code you will type appear in `green Courier` text. Press the **Enter** key after each line. Press the **Enter** key twice for a blank line between lines of code.

1. Open your text editor.

2. If one is not already open, open a blank document.

3. Type the following lines of code into your document:

```
 <html>

</html>
```

4. Move your cursor to the blank line between the opening and closing HTML tags.

5. Type

```
<head>
<!-- Info in this section is used by the browser-->

 </head>

<body>
 <!-- Info in this section can be seen in the window -->

 </body>
```

6. Move your cursor to the line after the comment in the head section.

7. Type `<title>This is my first web page</title>`.

8. Move your cursor to the line after the comment in the body section.

9. Type `Welcome to my first web page.`

10. Save your document as `first.html`. (This should be save as a plain text file.)

11. Open your browser.

12. Under the **File** pull-down menu, choose **Open File** .

You should see a dialogue box with a list of files.

13. Select your file, **first.html** .

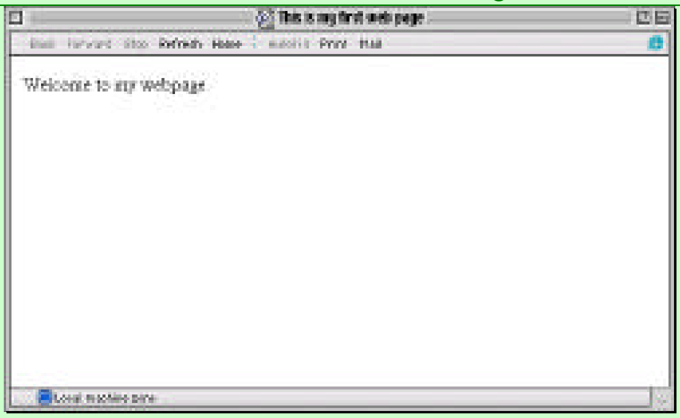
Your file should open on the screen.

Test Results

Your saved document should look like this:

```
<html>

<head>

<!-- Info in the section is used by the browser -->

<title>This is my first web page</title>

</head>


<body>

<!-- Info in the section can be seen in the window -->

Welcome to my first web page.

</body>

</html>
```

Your file should look like this when opened using your browser:


This is my first web page
Back Forward Stop Refresh Home AutoFill Print Mail
Welcome to my webpage
Local machine zone

Possible errors:

- If you can see any of the comments on the screen, you may have forgotten to include the exclamation point in the beginning of the comment tag.
- If the text has not appeared in the title bar, you may have forgotten to use the closing title tag.

Additional Study

To see these tags in action:

1. Go online.
2. Go to any web site.
3. View the source code.
   - Using Internet Explorer; select **Source** under **View** on the menu bar.
   - Using Netscape; select **Page Source** under **View** on the menu bar.
4. Look for the tags introduced in this chapter and see how they are used:
   - **<html> </html>**
   - **<head> </head>**
   - **<title> </title>**
   - **<body> </body>**
   - Comments **<!-- -->**

CHAPTER 2 : WORKING WITH TEXT

As you saw in the last tutorial, any text typed into an HTML document without a tag is considered to be a paragraph. If you didn't use any tags at all, all of the content you included in your HTML file would be shown as a single paragraph.

To break the content into separate text items, you need to use tags that mark the beginning and ended of each item. In this chapter you will learn how to use paragraph and heading tags to label and break up the text on a web page.

Each time you use the tags that create headings or paragraphs, the browser will start that element on the next available line and allow extra space between the heading and/or paragraph and the one before it.

Headings

A **heading** identifies the text after it and allows visitors to scan your document looking for something that interests them. Headings enable you to organize the content of your page so your visitors can quickly figure out where to go and then go there.

Use headings to:

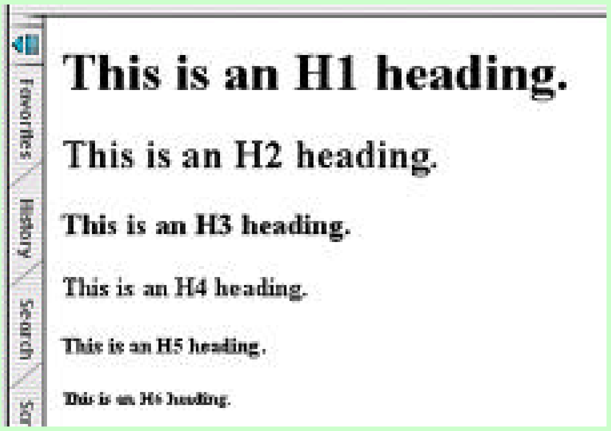
- State the topic of the following text
- Indicate the scope and purpose of the following text
- Signal a break between topics
- Divide the content into smaller sections

In addition, some web designers use heading tags to highlight text since these tags tell the browser to make the text following the tags large and bold.

Heading Tags

<h1>This is a heading.</h1>

Use the same level opening and closing heading tag.



This is an H1 heading.

This is an H2 heading.

This is an H3 heading.

This is an H4 heading.

This is an H5 heading.

This is an H6 heading.

The relative sizes of headings using heading tags <h1> through <h6>

HTML provides six levels of headings. The largest is h1. The smallest is h6.

If you want your heading to be the largest possible, as for a chapter title, use the first heading tag, **<h1>** . For less prominent headings, like section headings or subheads, use the next tag, **<h2>** and so on.

At the end of your heading text, you must use the appropriate closing tag. The closing tag is the almost identical to the opening tags except it has with a backslash after the opening angle bracket. For example, if you have a level 1 heading, **<h1>** , the closing tag will also be a level 1, **</h1>** .

Remember, in order for a heading to be most effective, keep headings relevant and on one line, if possible.

Paragraphs

In print or on the web, **paragraphs** are the basic building blocks of a page. They allow you to convey complex ideas requiring description or definition.

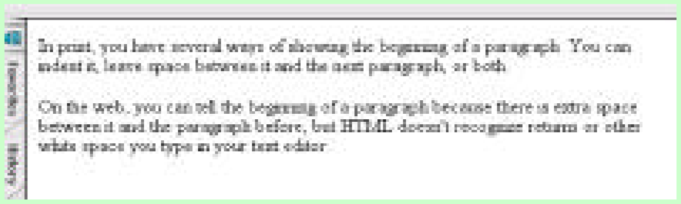
In print, you have several ways of showing the beginning of a paragraph. You can indent it, leave space between it and the next paragraph, or both, but on the web, the beginning of a paragraph is marked by extra space between it and the preceding paragraph.

To get this space, mark each paragraph using the opening paragraph tag **<p>** . and when you have finished your paragraph, use the closing paragraph tag, **</p>** .

Paragraph Tags

<p>This is paragraph text.</p>

Place an opening paragraph tag at the beginning of each paragraph and a closing paragraph tag at the end.



In print, you have several ways of showing the beginning of a paragraph. You can indent it, leave space between it and the next paragraph, or both.

On the web, you can tell the beginning of a paragraph because there is extra space between it and the paragraph before, but HTML doesn't recognize returns or other white space you type in your text editor.

Each <p> tag signals the broswer to show extra space between paragraphs.

As you look at the source code of other web pages, you will find some of their creators did not use ending paragraph tags, yet the pages appear the same as if they did. This is because the closing tag is optional is most versions of HTML.

Using **</p>** at the end of each of your paragraphs is a good practice to get into if you plan update your web pages to more advanced versions of HTML in the future.

Align Attribute
Attributes in the heading and paragraph tags are optional, but similar attributes are available for each. For most purposes, you will only need one: **align** .

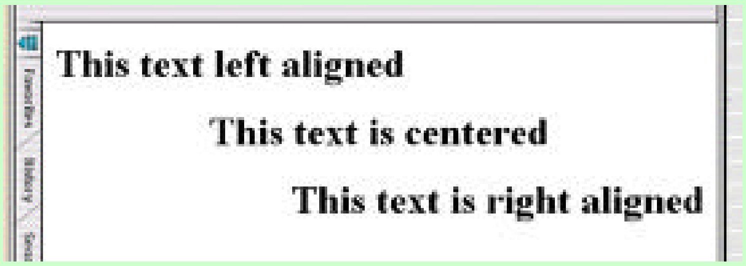
If you do not use the align attribute, your paragraphs and headings will line up on the left side of your screen. At times, however you may want to center your headings or align your paragraphs to the right side of your page. You can do this using the **align** attribute.

To align your paragraph or heading to the right side of your browser's screen, type **align="right"** after the tag.

The Align Attribute

```
<h1 align="left">This text left aligned</h1>
<h1 align="center">This text is centered</h1>
<h1 align="right">This text is right aligned</h1>
```

The align attribute is placed after the tag.


This text left aligned
This text is centered
This text is right aligned

The align attribute tells the browser how to align the heading or paragraph.

To center your paragraph or heading, use **align="center"** after the tag.

Forcing Line Breaks

When you start a new paragraph or heading, the browser inserts space between the paragraph and the item above it. This is useful when separating different thoughts into paragraphs, however, there may be times when you want to start a new line without that extra space, as in poetry or addresses.

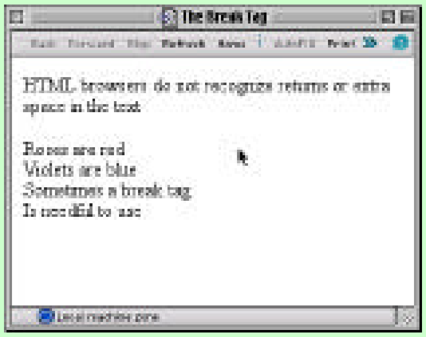
To start a new line without starting a new heading or paragraph, use break tag, **<br>** .

There is no closing tag for the **<br>** tag.

The Break Tag

...are blue<br>Sometimes...

Use the break tag to force a line break within a heading or a paragraph.



HTML browsers do not recognize returns or extra space in the text

Roses are red
Violets are blue
Sometimes a break tag
Is needed to use

“Non-Standard” Characters

Non-standard characters are those that are not normally found on the keyboard or those that you use for HTML commands. These include control characters (spaces, angle brackets and some punctuation), graphics (em dashes and mathematical symbols), and accented characters.

You can make non-standard characters appear on your web page by using the character’s **code name** or the **code** itself.

To alert the browser that the next several digits are a code, all codes and code names start with an ampersand (&)

Some Non-Standard Character Code Names (See Appendix A for entire list)

Code	*Code Name*	*Description*	*Code*	*Code Name*	*Description*
&#34	&quot	Double Quotation Mark	&#188	&frac14	1/4
&#38	&amp	Ampersand &	&#189	&frac12	1/2
&#60	&lt	Less than sign <	&#190	&frac34	3/4
&#62	&gt	Greater than sign >	&#199	&Ccedil	Captial C, cedilla Ç
&#160	&nbsp	Non-breaking space	&#201	&Eacute	Capital E, acute accent É
&#162	&cent	Cent sign ¢	&#214	&Ouml	Capital O, umlaut Ö
&#163	&pound	Pound sterling £	&#231	&ccedil	Small c, cedilla ç
&#169	&copy	Copyright symbol ©	&#233	&eacute	Small e, acute accent é
&#174	&reg	Registerd trademark ®	&#246	&ouml	Small o, umlaut ö

Nesting Tags

When you insert one tag completely within another, you are **nesting** the tags.

You have already done this without understanding it when you included the **<body>** and **<head>** tags inside the opening and closing **<html>** tags, and the **<title>** tags inside the opening and closing **<head>** tags.

Nesting can become very complicated when you start using text formatting tags. You should remember "**FILO**" or first in, last out. If you use a paragraph tag first, and then another tag inside it, you must close the second tag *first*, and then close the paragraph tag.

If you don't close your tags in the reverse order of opening them, results can be unpredictable, especially when using more advanced tags, like lists and tables. Some browsers will not display your formatting if your tags are not nested correctly.

You can think of nesting tags as if you were putting text inside other envelopes. You must address and close the inside envelopes before you can insert them into a larger envelope but you can enclose more than one smaller envelope in the larger one.

Text Formatting Tags

HTML has many tags which change the way text looks. The most commonly used are the bold and italic tags. These tags are almost always used within the paragraph or heading tags.

Making Text Stand Out

The bold tag, **<b>** makes text stand out from the other text around it by using a typeface made up of thicker lines. It is used when you want to give a word or phrase within a paragraph or header a special emphasis.

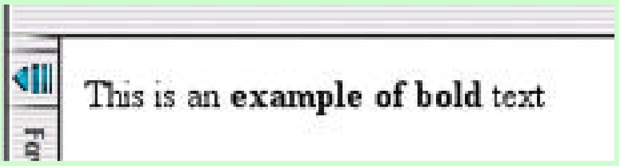
The corresponding closing tag for the bold tag is **</b>** . The closing tag tells the browser to stop making the text bold.

To use the bold tag, place the opening tag **<b>** in front of the text you want in bold and the closing tag **</b>** after the text.

Bold Tags

This is an <b>example of bold</b> text

Place an opening bold tag in front of the text you want in bold and a closing bold tag after it.



This is an **example of bold** text

An example of how text formattted with the <b> and </b> tags will appear in a browser.

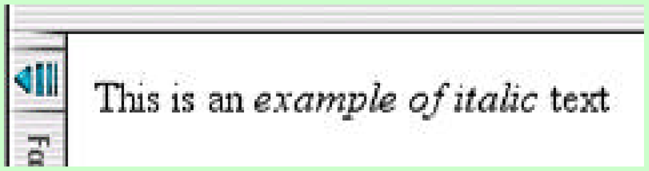
Like the bold tags, the italic tags **<i>** make text stand out. Italic text is slanted or oblique, and generally does not give as much emphasis as bold text.

The corresponding closing tag is **</i>** . Like the closing tag for bold text, this tag is required.

Italic Tags

This is <i>italic</i> text.

The opening and closing italic tags surround the text you want in italic type.


This is an *example of italic* text

An example of how text formattted with the <i> and </i> tags will appear in a browser.

To use the italic tag, place the opening tag **<i>** in front of the text you want in bold and the closing tag **</i>** after the text.

Both the bold and italic tags nest inside paragraph and heading tags, but you can make your text bold and italic at the same time by nesting the italic tag within the bold tag. Just remember that the *first* opening tag you use, must be the *last* closing tag.

Other Formatting tags

As HTML has grown and changed, other formatting tags have been added. Many of these are now considered **deprecated** , or "not recommended" because newer versions of HTML have better ways of formatting text. These new tags require more advanced knowledge of HTML and the deprecated tags will continue to work for some time, so if you don't want to learn advanced HTML, you can continue to use them.

Other Formatting tags

Name	*Use it for...*	*Open/Close tag*	*Example*
Superscript	Footnotes, foreign language abbreviations, math formulas, etc.	<sup> ... </sup>	M^{lle} 3^{rd} 10^{12}
Subscript	Chemical formulas, etc.	<sub> ... </sub>	H_2O
Typewriter Text (monospace)	Indicating what you need to type at the keyboard	<tt> ... </tt>	`Typewriter text`
Code (monospace)	Same as Typewriter Text, but implies computer codes.	<code> ... </code>	`Code text`

Changing Fonts

All the other tags covered so far have relied on the default typeface of the browser, but you can change the typeface by using the opening and closing font tags.

Like the other text formatting tags, the opening font tag, **<font>** , is used to show where you want the text to appear in a specific type style, and the closing font tag, **</font>** marks the end of the font change.

Unlike the other text formatting tags, there are a number of different attributes that can be used with the **<font>** tag. You must use at least one of them. You can use more than one of the attributes in a single **<font>** tag as long as you want to apply them to the same text. You do not need to include the attributes in any particular order.

The most commonly used attribute of the **<font>** tag is the **size** attribute. To use it, follow the tag name with **size="** ***x*****"** where ***x*** is the desired font size. The sizes you will most often use are the same as the heading sizes, 1-6. You can also use measurements like inches ("), picas (p), or pixels (px), but these will appear differently on different screens and are supported differently by different browsers. To avoid confusion, using specific measurements is not recommended.

Another attribute that is often used within the **<font>** tag is the **face** attribute. This allows you to name the actual typeface the browser will use to show the text within the **<font>** tags.

There are some drawbacks to the **face** attribute. Not everyone has the same typefaces available on their computer. The **face** attribute will allow you to name as many substitutes as you'd like, but if none are available to your page's visitor, all text will appear as the default.

To use the **face** attribute, type **face="typeface1, typeface2, typeface3"** after the tag and before the closing angle bracket. You should replace "typeface1", etc with the names of the typefaces you are specifying in preferential order.

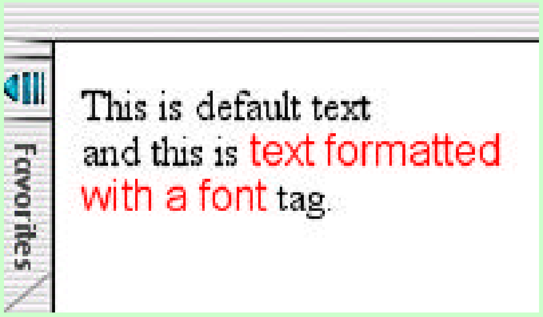
The **<font>** tag will also allow you to change the color of the text it surrounds. The attribute you use for this is **color** .

To use the color attribute, type **color="colorname"** after the tag name and before the closing angle bracket. You can use either a color name or a hexadecimal number. More information on color is available in chapter 3.

Font Tags

<font size="3" typeface="Arial, Helvetica, sans-serif" color="red">sample</font>

You may use as many attributes in a <font> tag as you like.



This is default text and this is text formatted with a font tag.

The <font> tag allows you to specifiy the size, typeface, and color of text.

Links & Anchors

Links are words or phrases you can click on to view another topic or web page. You can tell when text is a link because it is usually shown in another color with a line underneath it.

Before you can link text, you need to know:

- What text you want to link
- Where you want to link it to
- Whether you want the new page to open in a new window

The link tag, **<a>** , is probably the most important tag you can learn in HTML. As mentioned in Chapter 1, linking documents together is what makes HTML Hypertext.

Unlike the tags you have seen before now, the tag used to create links is used for more than one purpose. That purpose is defined by its attributes.

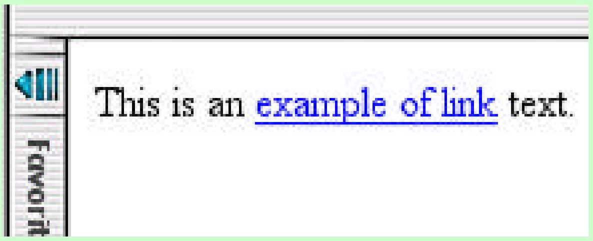
The list tag has two three commonly used attributes:

- **href** , or hyper-reference
- **name**
- **target**

The Hypertext Reference Attribute

<a href="newpage.html"></a>

Linking text to a different web page at the same web site



This is an example of link text.

Linked text will usually be another color and have a line underneath it.

Linking Text

When choosing text for your link, keep your visitor's attention focused by using only a few words only for your link text. You should never use more than a sentence as a link.

You should also avoid using the term "click here" because it breaks up the flow of your content. Instead, choose appropriate words you have already written. As soon as your visitors see the colored, underlined text, they will know to click on them.

After deciding what text to link, place the opening link tag, **<a>** in front of the text you have chosen and the closing link tag **</a>** after it.

Link Destinations

Now that you've told the browser what text is a link, you need to tell it the link's destination. HTML does this with the attribute **href** , or hyper-reference.

To make a page the destination of a link, you will need to know its URL. The **URL** specifies a precise location on the Web for a file.

You can think of the URL like an address. If I were to come and visit you at your home, I would need to know where you lived. You could give me your address, and I could drive over.

If the link destination page is on the same computer and in the same folder as the page you are linking, the URL can be as simple as just a file name. Documents on other computers and in other folders can be more complicated to specify.

To make things a little easier, keep your web pages in the same folder.

To link to a page on the same computer in the same directory, you would use the **href** attribute and set its value to the filename of the other document, **href="localfile.html"** .

If the link destination document is not in the same directory...

Some web designers organize their local files so that each web page and the it uses are in different directories. If you want to do this, you will need to specify the file pathname for the destination page as well as the name as of the file itself. A ***relative pathname*** is like driving directions for your computer.

To link to a local document which is not in the same directory, use a pathname which specifies how to get there, such as **href="../foldername/localfile.html"** .

In this pathname, the two periods and the slash indicate that the destination page's document is in a folder one level up from the folder. The text between the slashes is the directory. The filename of the other document is at the end.

In essence, this pathname tells the browser to "go up one directory, over to the folder named **foldername** , and then down to **localfile.html** ."

You can create connections to web pages designed by other people on other computers by using the http protocol and using the same **href** attribute. To do this, you will need to URL of that "outside" web page.

To find the URL of a web page that is not your own, look in the Location or Address box of your browser window (usually in the control bar at the top.) You can select all of that text, and copy it into your web page.

To link to a page using the http protocol, use the **href** attribute and set its value to the text you found in the address box, **href="http://www.umass-lowell.edu/username/filename.html"** .

The Anchor Attribute

<a name="anchor"></a>

Naming an achor

<a href="#anchor"></a>

Linking to an achor

Anchoring Text

The reason the link tag is an "a" rather than an "l" is because it is used to create **anchors** , or places in the text specified as destinations for links. You create the anchors first by using the **name** attribute and then you link to them.

When you link text to another web page, the browser will open the web page at the top. If you want the browser to open to some other part of the web page, or to "jump" from the link to another area on the same page (for footnotes or annotations), you need to include anchors in your text.

Creating anchors are the same as creating links except you use the **name** attribute rather than the **href** attribute. You can name your anchors anything you want, but each name should be unique on the page.

After opening the anchor with the opening link tag and the name attribute, you need to close it with **</a>** . You do not need to include any text between the opening and closing link tags in this instance.

If you wrote a web page that was book report and you wanted to include footnotes at the bottom of the page, you could use a series of anchors, named "foot1", "foot2", etc. The anchor for the first one would be **<a name="foot1"></a>** .

To link to an anchor, you must tell the browser that the text in the quotes is not a filename, but an anchor name. To do this, you use the command character **#**. A link to foot1 would be **<a href="#foot1">** .

To link to an anchor on another page, include the name of that page before the anchor, **<a href="bookreport.html#foot1">** .

Performing Actions Using Links

You can use the **href** attribute to do things other than open web pages by using different protocols.

Linking With The Mail Protocol

<a href="mailto:xxx@thisdomain.com"></a>

Using the link tags with the mailto protocol allows the reader to clink on a link and send you mail.

If you want the browser to open the visitor's mail program and send mail, you would use the **mailto:** protocol followed by the address you want the visitor to send the mail to.

You can also use a link to allow your visitors to download files by using the **ftp:** protocol followed by the file you want them to download, including the pathname.

Other protocols are less common, but include **news:** for opening newsgroups, **telnet:** for starting a direct connection to the destination computer with a telnet program, and gopher: which acts much like **ftp:** .

Most of these protocols are probably unfamiliar to you, but they all work the same, instead of **http://** after the **href** attribute, you use the other protocol with its accompanying file, newsgroup or mailing address.

The Target Attribute

If you want your visitors to see another page of information, but still keep the original page on their screen, you need to use the **target** attribute.

The Target Attribute

```
<a href="http://www.google.com></a>
```

Linking text to a web page at a different web site.

```
<a href="newpage.html" target="newpage"></a>
```

Opening another page in a new window.

Think of the window that a browser opens up as a television set. No matter how many channels you switch your television set to, it is still the same television set. Each time you click on a link without a target attribute, you use that window like your television set: the new link is shown in the same window.

But if you wanted to watch two different television shows at the same time, you would need another television set or a split screen. If you your visitors to look at another web page while the original page is still on the screen, you need to tell the browser to open another window. You can do this by using the attribute **target** .

The **target** attribute's default value is **_self** , which tells the browser to load all new web pages in the window where the link was clicked. To open another window, simply set the value of the **target** attribute to something else.

If you have a few small files that contain definitions of words used in your document, you could have them open a new window called definition by using the link tag: **<a href="window.html" target="definition">** .

Each time the visitor clicked on one of the definition links the linked file would load in the same window, in this case, called **definition** internally by the browser.

If you had a different window open up each time one of these definitions was clicked, your visitor would have to close the window when finished or multiple windows would be open by the visitor's browser.

The **target** attribute has several values that are used by more advanced HTML functions. Like, `_self`, these reserved values all start with an underscore.

Reserved Target Values

`_blank` — Loads the web page into a new, unnamed window

`_parent` — Used with web pages formatted with frames

`_self` — Loads the web page into the same frame or window that contains the hypertext link tag

`_top` — Loads the web page into the full diplay area, replacing the current frame layout

Tutorial

In this tutorial you will create a web page for the William Family Reunion.

Lines of code you will type appear in **`green Courier`** text. Press the **Enter** key after each line. Press the **Enter** key twice for a blank line between lines of code.

1. Open your text editor.

2. If one is not already open, open a blank document.

3. Type the following lines of code into your document:
```
 <html>

</html>
```

4. Move your cursor to the blank line between the opening and closing HTML tags and type the following lines of code:
```
<head>

 </head>

<body>

 </body>
```

5. Move your cursor to the line between the opening and closing head tags and type `<title>Williams Family BBQ</title>`.

6. Move your cursor to the line between the opening and closing body tags and type the following:
```
<!-- Basic Information -->

<!-- Who is Bringing What -->

<!-- Directions -->
```

7. Move your cursor to the line after the opening body tag, before the **Basic Information** comment, and type the following:
```
<h1 align="center">Williams Family BBQ</h1>
<h2 align="center">Sunday, <font color="red">August
29<sup>th</sup></font><h2>
<h2 align="center">at <font color="blue">Elliot Lake
Park</font>Picnic Area C</h2>
```

8. Move your cursor to the line after the **Basic Information** comment and type the following:
```
<p>It's my turn to plan the family reunion, so I have
signed up for the usual picnic area at Elliot Lake Park.
Since Elliot Lake Park was featured in <i>Where to Go With
Kids</i>magazine, there has been lots of competition for
the picnic areas. Finally, I was able to get Area C for the
last Sunday in August.</p>
```

```
<p>Send me <a href="mailto:ggw@temptide.com">email</a>if
you can come. Please let me know what dish you plan to
bring for the potluck dinner, so I can coordinate the
dishes.</p>
```

9. Move your cursor to the line after the comment **Who is Bringing What** and type the following:

```
<p>I will be bringing tablecloths, plates, silverware, and
drinking glasses. If cooking is not your "thing," please
feel free to volunteer to bring soda, chips or raw
veggies.</p>
```

10. Move your cursor to the line after the **Directions** comment and type the following:

```
<h2>Directions</h2>
 <h3>From the City</h3>
<p>Take US3 North to <b>Exit 34</b>. <br> Go &frac14; of
the way around the rotary to <b>Rt 4 North</b>. </p>
 <p>&dagger;Stay on Rt 4 North through <b>3 stoplights</b>
(approx. 12 miles.)<br> Elliot Lake Park is about <b>1
mile</b> after the last stoplight, on your <b>left.</b></p>

<h3>From the North</h3>
 <p>Take your best route to <b>US3 South</b>.<br>Take US3
South to <b>Exit 34A</b>. <br>To &frac34; around the rotary
to <b>Rt 4 North</b>. <br>Follow the directions from the
&dagger; above.</p>
```

11. Save your document as `williams.html`.

Remember to save **williams.html** as a *plain text* file.

12. Open your browser.

13. Under the **File** pull-down menu, choose **Open File** .

You should see a dialogue box with a list of files.

14. Select your file, **williams.html** .

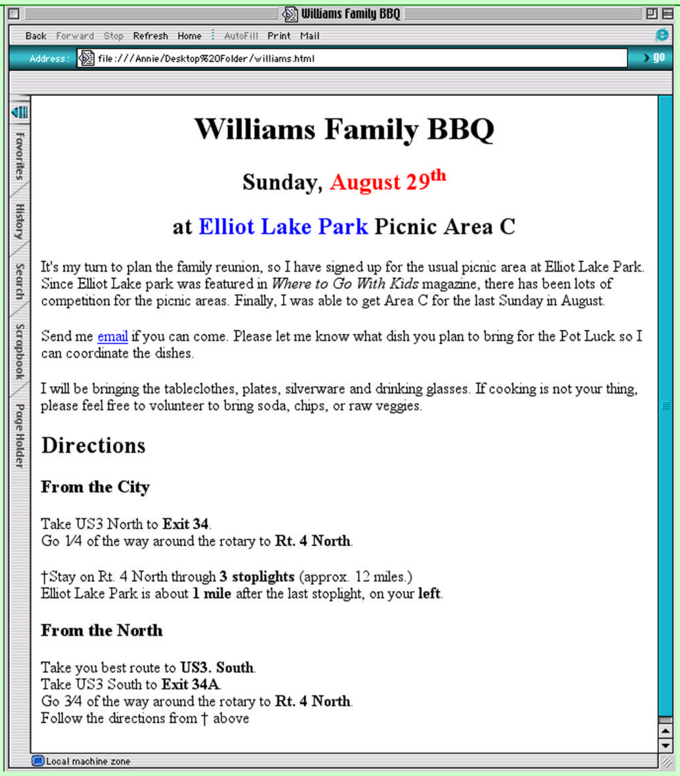
Your file should open on the screen.

Test Results

Your saved document should look like this:

```
<html>

<head>

<title>Williams Family BBQ</title>

</head>

<body>

<h1 align="center">Williams Family BBQ</h1>

<h2 align="center">Sunday, <font color="red">August
    29<sup>th</sup></font></h2>

<h2 align="center">at <font color="blue">Elliot Lake Park</font> Picnic Area
    C</h2>

<!-- Basic Information-->

<p>It's my turn to plan the family reunion, so I have signed up for the usual
    picnic area at Elliot Lake Park. Since Elliot Lake park was featured in
    <i>Where to Go With Kids</i> magazine, there has been lots of competition
    for the picnic areas. Finally, I was able to get Area C for the last
    Sunday in August.</p>

<p>Send me <a href="mailto:ggw@temptide.com">email</a> if you can come. Please
    let me know what dish you plan to bring for the Pot Luck so I can
    coordinate the dishes.</p>

<!-- Who is Bringing What-->

<p>I will be bringing the tableclothes, plates, silverware and drinking
    glasses. If cooking is not your thing, please feel free to volunteer to
    bring soda, chips, or raw veggies.</p>

<!-- Directions-->

<h2>Directions</h2>

<h3>From the City</h3>

<p>Take US3 North to <b>Exit 34</b>.<br> Go &frac14; of the way around the
    rotary to <b>Rt. 4 North</b>. </p>

<p>&dagger;Stay on Rt. 4 North through <b>3 stoplights</b> (approx. 12
    miles.)<br> Elliot Lake Park is about <b>1 mile</b> after the last
    stoplight, on your <b>left</b>.</p>

<h3>From the North</h3>

<p>Take you best route to <b>US3. South</b>. <br> Take US3 South to <b>Exit
    34A</b>.<br> Go &frac34; of the way around the rotary to <b>Rt. 4
    North</b>.<br> Follow the directions from &dagger; above</p>

</body>

</html>
```

Your file should look like this when opened by a browser:


Williams Family BBQ
Back Forward Stop Refresh Home AutoFill Print Mail
Address: file:///Annie/Desktop%20Folder/williams.html
go
Favorites
History
Search
Scrapbook
Page Holder
Local machine zone

Williams Family BBQ

Sunday, August 29th

at Elliot Lake Park Picnic Area C

It's my turn to plan the family reunion, so I have signed up for the usual picnic area at Elliot Lake Park. Since Elliot Lake park was featured in *Where to Go With Kids* magazine, there has been lots of competition for the picnic areas. Finally, I was able to get Area C for the last Sunday in August.

Send me email if you can come. Please let me know what dish you plan to bring for the Pot Luck so I can coordinate the dishes.

I will be bringing the tablecloths, plates, silverware and drinking glasses. If cooking is not your thing, please feel free to volunteer to bring soda, chips, or raw veggies.

Directions

From the City

Take US3 North to **Exit 34**.
Go 1/4 of the way around the rotary to **Rt. 4 North**.

†Stay on Rt. 4 North through **3 stoplights** (approx. 12 miles.)
Elliot Lake Park is about **1 mile** after the last stoplight, on your **left**.

From the North

Take you best route to **US3. South**.
Take US3 South to **Exit 34A**.
Go 3/4 of the way around the rotary to **Rt. 4 North**.
Follow the directions from † above

You should notice:

- Typing the comments before writing the content of the site is one way of organizing the information on your web page.
- Notice the different in sizes between the **<h1>**, **<h2>** and **<h3>** headings.
- When you gave no **align** attribute the heading tags after the directions, the default was to align to the left.

Some possible errors:

- If you have one big paragraph after one or each of the headings, you may have forgotten to type **<p>** at the beginning of the paragraphs.
- If words in addition to "August 29th" are in red, or in addition to "Elliot Lake Park" are in blue, you may have forgotten to type in the ending font tag **</font>** .
- If words in addition to "Where to Go With Kids" are in italic, you may have forgotten the ending italic tag, **</i>** .
- If words in addition to "Exit 34", "Rt. 4 North", "3 stoplights","1 mile", "left", "US3 South", "Exit 34", or "Rt. 4 North" are in bold, you may have forgotten to type in the ending bold tag, **</b>** .
- If words in addition to "email" are underlined, you may have forgotten to type the closing anchor/link tag, **</a>** .
- If you cannot see the dagger or the fractions, check to see that there are no spaces in the extended character code names for those characters.
- If you cannot see the dagger and the fractions and there are no spaces in the code names, you may be using a browser that does not support the code names, and you will need to use the code instead.