

LESSON 2

Markup Tags and Character Entities

OBJECTIVES

You will be able to:

- list the components that make up HTML documents.
- use HTML syntax and apply markup tags.
- create HTML documents.
- format text within HTML documents.
- create and format a variety of lists.
- create and format horizontal rules.
- use character entities.



This lesson provides you with the foundation for understanding HTML. You will learn proper HTML syntax, and how to use markup tags and character entities for structuring and formatting text in HTML documents.

HTML Tags

HTML Elements

HTML provides a system for structuring documents. Originally, the intent was not to format text, but to arrange it into logical sections. These sections, in turn, contain formatting characteristics that may affect the appearance of the text. More recently, HTML tags have been introduced solely for the purpose of formatting the HTML document.

Each section in an HTML document represents an *element*. An element is made up of coded instructions, or *markup tags*. Tags can be *open* or *closed*. (You may sometimes see the term “paired” and “empty” instead of “closed” and “open.”) A closed tag is also known as a *container* because it consists of a pair of matching tags that surround an area of text—it “contains” the text that will be affected by its instructions. A closed tag requires both a beginning tag to turn it *on* and an ending tag to turn it *off*. For example, the instruction to bold a section of text is turned on at one point, and at another location is turned off. An open tag, on the other hand, gives an instruction to a section of the document but does not surround any text and therefore does not need to be concluded.

Some elements also include *attributes*, or properties that provide additional information about the tag’s instruction. An example of an attribute is found within the horizontal line tag. An attribute can be used to describe the line thickness or alignment.

The collection of elements that make up an HTML document is known as its *source code*. HTML source code is plain ASCII text and may be created using standard text editors. As such, the source code for HTML documents can be created or edited in any text editor application, such as Microsoft Notepad for Windows, or SimpleText for Macintosh. Specialized *HTML editors* are also available which automate the creation of HTML tags and are widely available for all computer platforms.

Markup Tag Syntax

A markup tag is defined using a precise syntax. As when using a programming language, it is essential to conform to the required syntax of the markup tags in order for the browser to correctly interpret the “language” of your HTML documents. The syntax for a markup tag is shown in Figure 2-1.

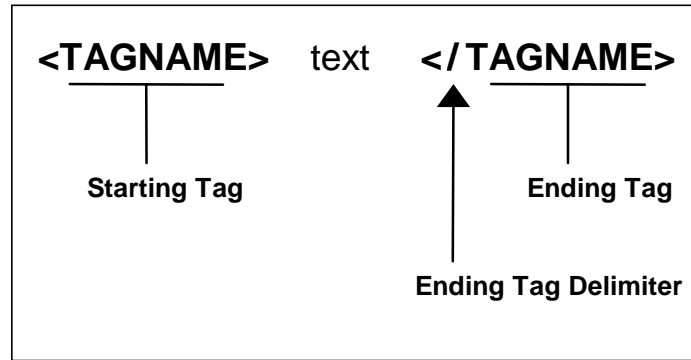


Figure 2-1: Markup tag syntax showing beginning and ending tags

The markup tag is enclosed within a pair of angle brackets (< and >), sometimes referred to as "wickets." A starting tag begins with the left angle bracket (<) followed immediately by the tag name (one word), any attributes, and then a right angle bracket (>) to close the tag. If it is a closed element that contains text, it will also require an ending tag. Ending tags are identical to starting tags, with the exception of a forward slash (/) immediately before the tag name.

Document Structure

Using Tags for Document Structure

In the upcoming exercises, you will learn to use a variety of tags to structure your HTML documents. Four primary tags supply the basic framework of all HTML documents. The <HTML> tag is the only required tag. It declares that the document should be recognized as an HTML document. It is followed with tags to create head and body sections. While the HEAD and BODY tags are not absolutely necessary, it is a good practice to include them in every HTML document. You should use the following tags in your documents:

<i>Markup Tag</i>	<i>Use</i>
HTML tag	Identifies file type as HTML
HEAD tag	Identifies start and end of page header
TITLE tag	Labels the title bar of the Web browser
BODY tag	Identifies the start and end of Web page content

Figure 2-2: Function of HTML Header Tags

Not all tags will have a visible effect in the browser, especially identification tags such as <HEAD> and <BODY>. Figure 2-3 identifies the primary structure tags, their suggested use, and their visibility.

<i>Markup Tag</i>	<i>Required</i>	<i>Strongly Recommended</i>	<i>Visible in Browser</i>
HTML	Yes	Yes	No
HEAD	No	Yes	No
TITLE	No	Yes	Yes
BODY	No	Yes	No

Figure 2-3: Use of HTML Structure Tags

HEAD

The HEAD element provides information about the document itself. The <HEAD> tag will be the first tag after the <HTML> tag in an HTML document. The <TITLE> tag is specified within the <HEAD> tag, to introduce a label for the document. There may be only one TITLE element per document.

TITLE

The <TITLE> tag is particularly important, as the title of the document will appear in the title bar of the web browser. The title will also appear in a “bookmarks” or “favorites” list of the browser, if the page is saved in such a listing. The following guidelines should be kept in mind when determining a title for your document:

- Keep it short—it should fit on the title bar or in a small text box. Longer titles are unwieldy in a bookmark list.
- Make it descriptive—titles often include an author or organization name, but should also be informative out of context (as in search results or another’s browser list).
- Make it unique—each document should have a title that distinguishes it from other documents within the web site or bookmark list.
- Make it text only—the content of the title may not contain any additional tags.

BODY

The BODY element encloses the bulk of your HTML document. While the HEAD element conveys introductory information *about* the document, nothing typed in the HEAD section is visible to the user, except the title. The <BODY> tag, on the other hand, contains all the content material in your document and will be visible in the browser window.

Since the BODY element contains the substance of text and tags in the document, it is a closed tag and must be ended. The only tag you will see following the closing </BODY> tag will be the closing </HTML> tag.

The required source code appears as follows:

```
<HTML>
<HEAD>
```

```

<TITLE> Document Title </TITLE>
</HEAD>
<BODY>
The content of your HTML document . . .
</BODY>
</HTML>

```

Every HTML document should contain these four elements, in the order shown. Most of the additional tags that are used for text formatting, image placement and links will be contained in the BODY of the document.



Exercise 2-1: Creating a Web Document

In this exercise, you will create your first HTML document.

1. Create a new document using your text editor.
 - Type the following HTML source code:

```

<HTML>

<HEAD>
<TITLE> My First HTML Document </TITLE>
</HEAD>

```

Notice how both the <HEAD> and <TITLE> tags have a starting and an ending tag. The information to be displayed in the title bar is entered between the starting and ending <TITLE> tags.

The tag names can be specified in either upper or lower case text. The text between the starting and ending <TITLE> tags will be displayed exactly as you type it. The browser ignores the case of the HTML source code. For clarity in editing your HTML document, it is suggested that you use UPPERCASE for your tags.

- Continue in the same document, and add the following:

```

<BODY>
This text will be displayed in the browser window as the body of the document. It will appear in a default font and size. You will soon learn how to apply formatting to alter the appearance of the text.
</BODY>

</HTML>

```
- Save the HTML file you just created in the folder specified by your instructor. Name it **first.htm**.

If you know that your web pages will be stored on a web server that requires an **html** extension, you are encouraged to use that extension throughout this course.

- Open your web browser.

Once open, you can toggle between your text editor and web browser windows by holding down the **<ALT>** key while tapping the **<TAB>** key.

- From your web browser, open the file **first.htm**.



Figure 2-4: HTML Document As Viewed In Netscape Browser

A Web browser recognizes individual tags by the wickets that mark the start and end tags.

In Exercise 2-1, the placement of the beginning and ending tags were neatly placed on separate lines. However, the HTML tags are read without respect to line breaks entered in the text editor. The following example demonstrates how the tags might be arranged with the document.

```
<HTML> <HEAD> <TITLE> Document Title </TITLE> </HEAD>
</HTML>
```

Entering HTML tags without line breaks, as in the example, will result in identical output by the Web browser.

Comment Tag

The COMMENT tag allows the author to annotate the HTML document to improve its clarity and readability for other authors or editors of the document. For example, you may wish to explain why you used a particular tag, or what part

of the document should be updated, or the status of some content that may require revisions.

Comments are not displayed in the browser; they are only visible when viewing the HTML source code. To insert a comment, the author will use a single, open tag. The opening wicket is followed with an exclamation point, followed by two hyphens. The comment, or message, is entered next, and the tag concludes with two more hyphens and the closing wicket.

The following is an example of a comment tag.

```
<! - - This is a message for others who may edit this document - - >
```



Exercise 2-2: Adding Comments to Your HTML Code

In this exercise, you will add a comment to the body of your document.

2. Switch (if necessary) to **first.htm** in your text editor.
 - Insert the following comment:

```
<HTML>
<HEAD>
<TITLE>My First HTML Document</TITLE>
</HEAD>
<BODY>
```

```
<! - - This document was created by me in HTML class - - >
```

This text will be displayed in the browser window as the body of the document. It will appear in a default font and size. You will soon learn how to apply formatting to alter the appearance of the text.

```
</BODY>
</HTML>
```

Generally, it is a good idea to place HTML comments on a separate line to improve readability of the HTML source code. Comments should be used to add information that is relevant to the author or others who will edit the file.

- Save to update the HTML source file you just modified.
- Switch to your web browser.
- Reload the source file **first.htm** from within your web browser.

Your comment should not be visible. If you wish to see the comment, view the source code from within your browser.

Heading Level Tags

Heading Level tags provide a way for you to divide a page into different level topics. Much as you would use headings and sub-headings within a word processing document, you may use heading level tags within your HTML document.

There are currently six levels of headings specified in HTML: <H1>, <H2>, <H3>, <H4>, <H5>, and <H6> (in order from largest to smallest). Typically, the <H1> tag might be the first tag to follow the opening <BODY> tag, and will be followed, as needed, by an <H2> heading, an <H3> heading, and so on. While this provides good style for your documents, it is not an HTML requirement that you order the tags hierarchically.

Heading tags automatically format the designated text. However, each browser determines exactly what format will be used. A heading tag may appear in a bold font in one browser, but not another. Font size is relative to the tags numeric level, with <H1> appearing in the largest size. The browser will automatically insert a blank line before and after the heading. (Additional tags for paragraph and line breaks will be addressed in detail in the next section.)



Exercise 2-3: Adding Heading Level Tags

In this exercise, you will add the necessary structure and Heading Level tags, <H1> to <H6> to the document, **travel.txt**.

3. Open the file **travel.txt** in your text editor.

- Use **Save As...** to save it as an HTML file called **travel.htm**.

*Recall that an HTML file is an ASCII text file, saved with the **htm** or **html** extension. You may wish to use the HTML extension for your exercise files, if that is the extension specified by your web server administrator.*

- Enter the following HTML code at the beginning of the document.

```
<HTML>
<HEAD>
<TITLE> Travel Tips </TITLE>
</HEAD>
<BODY>
```

- Place a pair of Heading level tags on either end of each line of text in the document. Your code should appear as shown below when you are finished.

```
<H1> Travel Tips (Level 1) </H1>
<H2> Choosing a Destination (Level 2) </H2>
<H3> Accommodations (Level 3) </H3>
<H4> Hotels (Level 4) </H4>
<H5> Price (Level 5) </H5>
```

```

<H6> Inexpensive (Level 6) </H6>
<H6> Moderate (Level 6) </H6>
<H6> Luxury (Level 6) </H6>

</BODY>
</HTML>

```

- Switch to your web browser.
- Open the file **travel.htm** from within your web browser.



Figure 2-5: Heading Levels 1 through 6

Body Text Formatting

Body text is the unformatted text that appears in your HTML documents. Formatting tags can be applied to change the font type, style, amount of white space, placement of line breaks, etc. However, in the absence of additional formatting tags, standard text will appear in your documents as a single continuous paragraph. The only breaks you will see will be the result of word wrap (which adjusts as you resize the width of the browser window) and a single space between words.

Carriage returns, line breaks, tabs, extra spaces and other white space embedded in body text are ignored by Web browsers. In order to format your text, you must explicitly insert an appropriate tag.



Exercise 2-4: Manipulating Body Text

In this exercise, you will change the formatting of a text document.

4. Open the file **voyage.txt** in your text editor.
 - Use the **Save As...** command to save the HTML source file, with the name **voyage.htm**.
 - Insert the cursor at the end of the sentence “We’ll make the plans; you’ll have the fun!” and insert a return or two.
 - Save the file and switch to your web browser.
 - Open the **voyage.htm** document in your web browser.



Figure 2-6: Text Without Paragraph Breaks

The return(s) you entered will not display in the browser display.

Remember, a browser recognizes only single spaces between words; it ignores carriage returns, line breaks, and tabs. The following example of HTML would yield the identical output in a web browser as the figure above.

```
<HTML><HEAD><TITLE>Voyager
Tickets</TITLE></HEAD><BODY>
<H1>Traveling with Voyager Tickets</H1>
Welcome to the Voyager Tickets Web site.
```

**Voyager Tickets is ready to take you anywhere you want to go.
We'll make the plans; you'll have the fun!**

**Fly away to a European castle.
Cruise the globe to a secluded tropical island.
Plan your next meeting at a luxurious conference center.**

Whatever your travel needs may be, Voyager Tickets can help you make the arrangements!
 </BODY></HTML>

In order to force a line wrap or paragraph break, special tags are needed.

Break Tag

The *BREAK* tag,
, is an open tag; only a single tag is required. The
 tag inserts a line break at the location of the
 tag, forcing text to the next line and aligning it to the left.



Exercise 2-5: Adding Line Breaks to Text

In this exercise, you will add a line break to a block of text.

5. Return to your text editor, and the document **voyage.htm**.
 - Insert a
 tag in each of the locations indicated below

```
<HTML>
<HEAD>
<TITLE> Voyager Tickets </TITLE>
</HEAD>
<BODY>
<H1>Traveling with Voyager Tickets</H1>
```

Welcome to the Voyager Tickets Web site. Voyager Tickets is ready to take you anywhere you want to go. We'll make the plans; you'll have the fun!

**
**

Fly away to a European castle.

**
**

Cruise the globe to a secluded tropical island.

**
**

Plan your next meeting at a luxurious conference center.

**
**

Whatever your travel needs may be, Voyager Tickets can help you make the arrangements!

</BODY>

</HTML>

- Save the file.

- Switch to your web browser and reload the **voyage.htm** file.

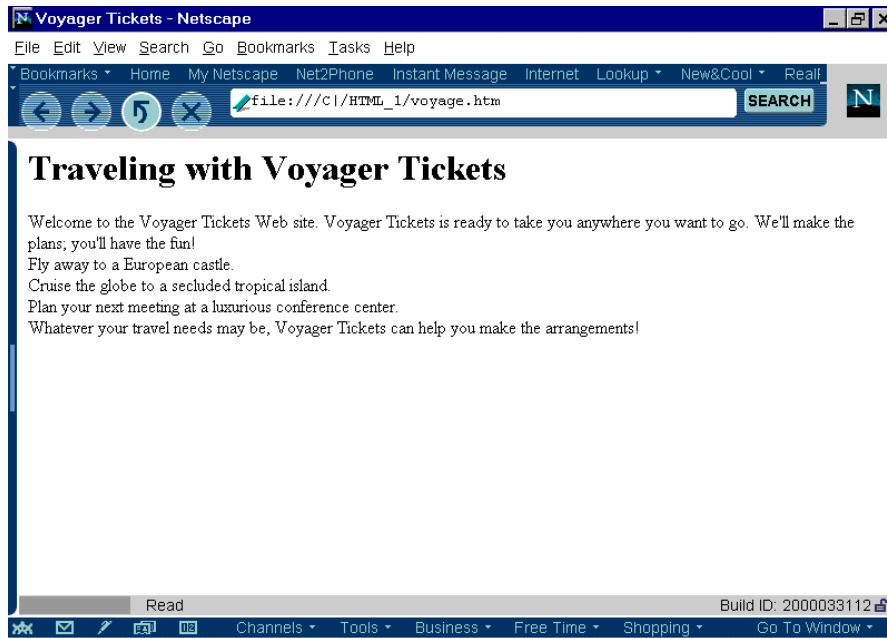


Figure 2-7: Using The BREAK Tag

Adjust the window size of your web browser to see the effects on word wrap and line breaks. The window size will determine word wrap unless the break tag forces a line break.

Paragraph Tag

In the previous example, inserting line breaks improved the visual appearance and readability of the text. If you wish to include a line of blank space between paragraphs, you will use the PARAGRAPH tag.

The *PARAGRAPH* tag <P>, indicates the start of a new paragraph. The paragraph tag inserts a full extra line space at the location of the tag. The difference between the
 and <P> tags is that the
 tag does not insert any additional line space, but simply forces text to next line. The browser *will* display an additional line of space between paragraphs when the <P> tag is used.

The paragraph tag, like the break tag, began as an open tag. Changes in the HTML standards have changed the paragraph tag, adding attributes and allowing its use as a closed tag. However, current browsers will interpret the paragraph tag as an open or closed tag.



Exercise 2-6: Adding Paragraph Breaks to a Block of Text

In this exercise, you will add paragraph spacing between sections of the text.

6. Switch back to **voyage.htm** in your text editor.

- Add a <P> tag after the first sentence, “Welcome to the Voyager Tickets Web site.”
- Change the first and last break
 tags to paragraph <P> tags. Your file should look similar to that shown below.

```

<H1>Traveling with Voyager Tickets</H1>
Welcome to the Voyager Tickets Web site.
<P>
Voyager Tickets is ready to take you anywhere you want to go.
We'll make the plans; you'll have the fun!
<P>
Fly away to a European castle. <BR>
Cruise the globe to a secluded tropical island. <BR>
Plan your next meeting at a luxurious conference center.
<P>
Whatever your travel needs may be, Voyager Tickets can help
you make the arrangements!
    
```

- Save the file and switch to your web browser. Reload the file.

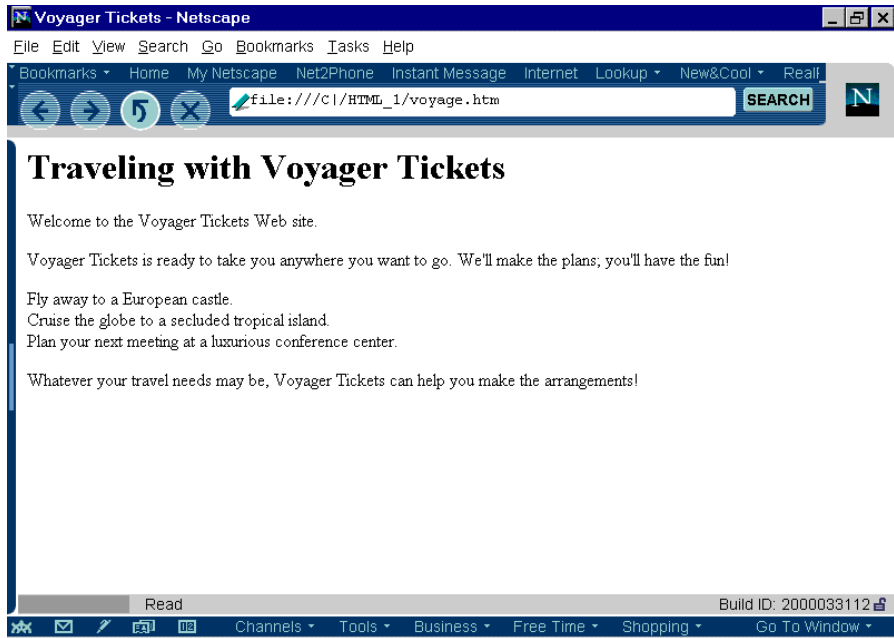


Figure 2-8: Use Of The Paragraph Tag

Centering

One option for formatting is the ability to center text and other objects on a page. There are currently two methods for centering text: the <CENTER> tag and the ALIGN attribute. An attribute is an optional instruction added within a tag to specify a variation to the standard use of the tag. The ALIGN attribute can be used with the <P> tag or any of the Heading Level tags <H1> to <H6>.

A `<CENTER>` tag is opened anywhere within the body of the document, and all text, graphic, table and form elements that follow it will be centered within the browser window. At some point before the ending body tag `</BODY>`, the `CENTER` tag must be closed with `</CENTER>`. The `CENTER` tag is ideal when multiple paragraphs or elements must be centered.

If there is just a single line or paragraph to be aligned, the `ALIGN` attribute may be more efficient. To add an attribute to a tag, the tag is begun in the usual manner, with an open wicket and the tag (name). A space is entered to separate the tag from the attribute, followed by an equal sign and a value, and finally the close wicket.

`<P ALIGN=CENTER>` All text will be centered until the `PARAGRAPH` tag is closed or ended with any other applicable tag.`</P>`

Be advised that the `CENTER` tag is deprecated in the HTML 4.0 standard, in favor of Cascading Style Sheets. (Cascading Style Sheets are covered in depth in the *Building Web Pages with HTML, Part 3* course.) A deprecated element or attribute is one that has been outdated by newer constructs and may become obsolete in future versions of HTML. For more information regarding the 4.0 standard and deprecation of tags, visit the World Wide Web Consortium web site at <http://www.w3.org>.



Exercise 2-7: Centering Text

In this exercise, you will learn how to center text using the `ALIGN` attribute and the `<CENTER>` tag.

7. Open the **voyage.htm** document in your text editor.

- Add the `ALIGN` attribute to the first `<H1>` tag as shown.

```
<BODY>
<H1 ALIGN=CENTER>Traveling with Voyager Tickets</H1>
Welcome to the Voyager Tickets Web site.
```

- Add the `CENTER` tag before the text “Fly away to a European castle.”

```
<CENTER>
Fly away to a European castle. <BR>
Cruise the globe to a secluded tropical island. <BR>
```

- Add a closing tag before the last paragraph.

```
Plan your next meeting at a luxurious conference center.
</CENTER>
```

- Save the changes.
- Switch to your web browser.

- Reload the **voyage.htm** document in the browser.

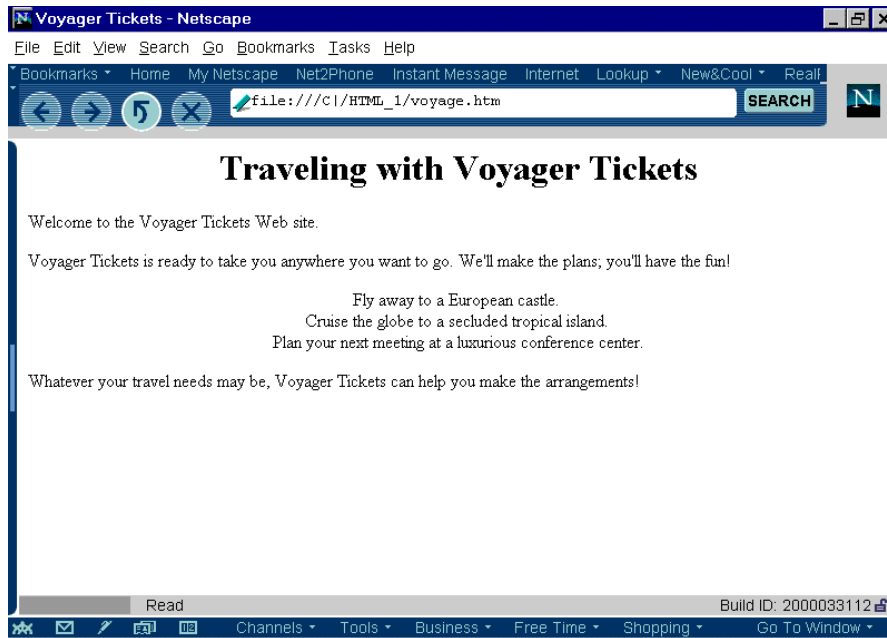


Figure 2-9: Use Of The ALIGN Attribute And CENTER Tag

Blockquote Tag

The *BLOCKQUOTE* element was developed to format lengthy quoted material. Formal style manuals specify that quotations be indented and single-spaced.

Most browsers interpret the `<BLOCKQUOTE>` as a signal to indent the text, generally at the right and left margins. Since the `<BLOCKQUOTE>` tag encloses the section of text being quoted, it is a closed tag and must be ended with a `</BLOCKQUOTE>` tag.

The blockquote element also adds a line space at each tag. (You may view the sample shown below by opening the **blockquote.htm** file in your browser or text editor.)

```
<BODY>
```

```
<H1>What Our Customers Say About Us</H1>
```

```
We at Voyager Tickets take great pride in our customer
satisfaction. Scores of happy clients have written to us to tell us
what they thought of our services. We thought we'd share
excerpts from just a few of those letters:
```

```
<BLOCKQUOTE>
```

```
"I've done a lot of traveling with a lot of travel agencies - but
Voyager is tops in my book!"
```

```
<P>
```

```
"We had been trying to plan a complicated trip but ran into
obstacles at every turn. Once we asked Voyager Tickets to help
us out, our itinerary and reservations were set in no time. It was
```

a great trip and, thanks to Voyager, a perfect anniversary present."

</BLOCKQUOTE>

Any one of our agents can help you with your special plans 24 hours a day. Contact us soon to experience our quality services for yourself.

</BODY>

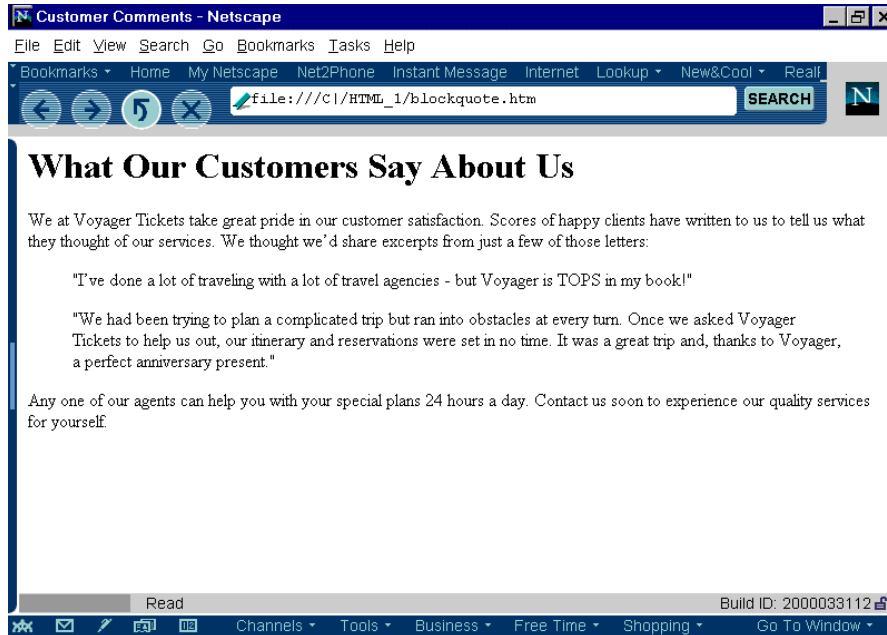


Figure 2-10: Use Of BLOCKQUOTE Tag

Preformatted Tag

The PREFORMATTED tag, <PRE>, instructs the browser to preserve character spacing. This tag is useful for displaying unusual spacing such as text imported from a mainframe computer, or programming code. However the <PRE> tag has the drawback of displaying in a mono-spaced font such as Courier, rather than the standard proportional-spaced font used in most web browsers. At one time, the <PRE> was the only effective way to format tabular material; now there are tags for creating more sophisticated table structures. Note that the <PRE> tag requires the user to insert all line breaks, as the browser will not apply word wrap to preformatted text. (You may view the following sample file by opening **room.htm** in your browser.)

```

<BODY>
<PRE>
                                ACCOMMODATIONS

HOTEL      STARS  PRICE  ROOMS  DINING  CABLE

Paradiso  ****  $150-250  21      3      Yes
The Windmill**  $40-50  35      0      No
Smith's   ***   $60-90   50      1      Yes
</PRE>
</BODY>

```

The file would display in the browser as shown.



Figure 2-11: Use Of Preformatted Tag

Character Formats

There are a variety of markup tags available in HTML for changing the format of characters within a block of text. These tags are known as character style tags. The character style tags are divided into two groups, often identified as *physical* tags and *logical* tags. Physical tags describe the appearance of text, such as the bold and italic. Logical tags describe the purpose of the text, such as a citation or definition.

The table that follows lists the most commonly used physical and logical tags.

Physical Character Tags		Logical Character Tags	
<code> </code>	Bold	<code> </code>	Emphatic
<code><I> </I></code>	Italic	<code> </code>	Stronger than
<code><STRIKE> </STRIKE></code>	Strike-through	<code><CODE> </CODE></code>	Monospaced for example of code
<code><TT> </TT></code>	Typewriter text	<code><CITE> </CITE></code>	Citation for a quoted passage.

Figure 2-12: Common Logical and Physical Character Style Tags

Logical tags were most common in earlier HTML files. They are used less frequently today as they are not consistently interpreted by the various browsers, and can lead to unexpected formats. Physical style tags, on the other hand, are most prevalent in HTML documents today.

Bold, Italic, and Strike-Through Tags

Three common character style tags include bold ``, italic `<I>`, and strike-through `<STRIKE>` tags. All three of these tags are closed. With modern browsers, these tags can be combined or nested; e.g., you can create text that is both bold and italic.

It is important when nesting one tag within another that you “mirror” the tags. For instance, if you begin with the opening tag for BOLD, you must end with the closing tag for BOLD.

Incorrect: `<I>quality services </I>`

Correct: `<I>quality services </I>`

Incorrectly nesting tags may cause the text to display with incorrect formatting. In some browsers, the enclosed text might not even display. You should be aware of the capabilities and limitations of the browsers used by your target audience.



Exercise 2-8: Formatting Text with Bold, Italic, and Strike-through Tags

In this exercise, you will enhance a document using character format tags.

8. Open the **quotes.htm** document in your text editor.
 - Insert the changes shown below, or apply format tags as you wish.

`<H1>What <I>Our Customers</I> Say About Us</H1>`

We at **Voyager Tickets** take great pride in our customer satisfaction. Scores of happy clients have written ~~to us~~ to tell us what they thought of our services. We thought we'd share excerpts from just a few of those letters:

<BLOCKQUOTE><I>

"I've done a lot of traveling with a lot of travel agencies - but **Voyager** is great in my book!"

<P>

"We had been trying to plan a complicated trip but ran into obstacles at every turn. Once we asked **Voyager Tickets** to help us out, our itinerary and reservations were set in no time. It was a great trip and, thanks to **Voyager**, a perfect anniversary present."

</I></BLOCKQUOTE>

Any one of our agents can help you with your special plans **24 hours a day**. Contact us soon to experience our quality services for yourself.

- Save the HTML source file.
- Switch to your web browser and open the **quotes.htm** document.

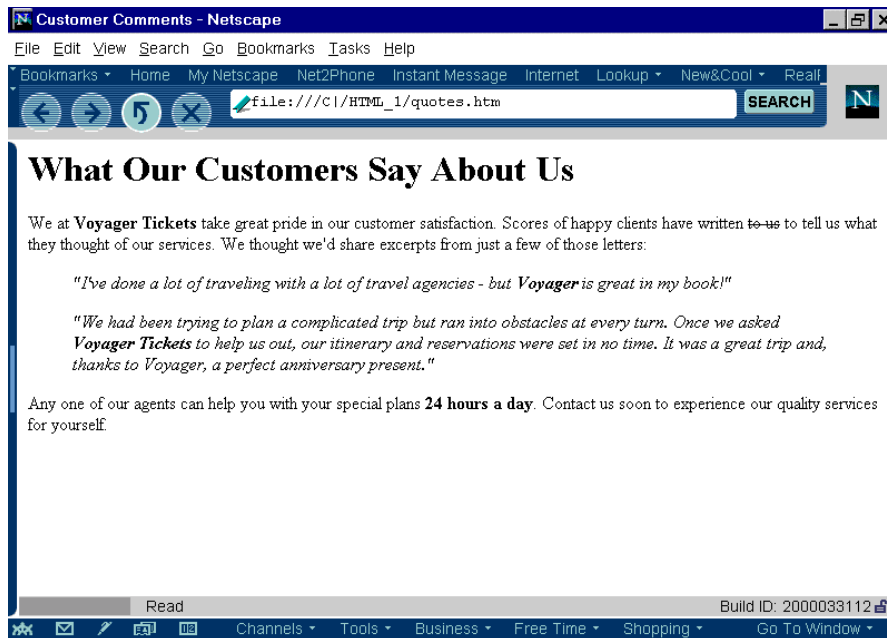


Figure 2-13: Formatted Text

List Structures

There is a variety of markup tag styles available in HTML for creating lists in your documents. The two most common types of lists are ordered and unordered lists.

Unordered and Ordered Lists

The first type of list is an unordered list, commonly referred to as a "bullet" list. Items in the list are displayed indented, with a bullet placed before each item. The second type of list is an ordered list, commonly referred to as a "numbered" list. It is just like an unordered list, except the items in the list are displayed with numbers instead of bullets.

The markup tag indicates the type of list to be display. The UNORDERED LIST tag is , while the ORDERED LIST tag is . The LIST ITEM tag is inserted before each item, in either type of list. The tag is an open tag, whereas the and tags are closed tags.

The source code for an unordered list might resemble the example below.

```
<UL>
<LI> Clothing
<LI> Climate
<LI> Money
<LI> Food
<LI> Entertainment
</UL>
```

The above code would display a bulleted list as shown.

- **Clothing**
- **Climate**
- **Money**
- **Food**
- **Entertainment**

The source code for an ordered list would be similar to the unordered list, except that you would replace the and tags with the and tags.

The source code for an ordered list might appear as follows:

```
<OL>
<LI> Pack For A Variety Of Weather Possibilities
<LI> Include Convertible Outerwear
<LI> Take Layers Of Clothing
<LI> Look For Clothing That Is Easily Hand Washed
</OL>
```

The resulting text would appear in the browser as follows:

1. **Pack For A Variety Of Weather Possibilities**
2. **Include Convertible Outerwear**
3. **Take Layers Of Clothing**
4. **Look For Clothing That Is Easily Hand Washed**



Exercise 2-9: Adding Lists to Your Documents

In this exercise, you will add HTML tags to create an unordered and an ordered list.

9. Open the document **guides.htm** in your text editor.

- Insert the following tags into the existing text to create an Unordered List.

```
<H2>Currently Available Guides</H2>
```

```
<UL>
```

```
<LI>Clothing
```

```
<LI>Climate
```

```
<LI>Money
```

```
<LI>Food
```

```
<LI>Entertainment
```

```
</UL>
```

What type of information will you find in our famous guides?
Check out the sampling of the pointers from our clothing guide.

- Insert the following tags into the existing text to create an Ordered List.

```
<H2>Principles of Packing</H2>
```

```
<OL>
```

```
<LI>Pack For A Variety Of Weather Conditions
```

```
<LI>Take Layers Of Clothing
```

```
<LI>Pack As Little Possible
```

```
<LI>Take Out Insurance
```

```
</OL>
```

```
</BODY>
```

```
</HTML>
```

- Switch to your web browser.
- Open the **guides.htm** document in your web browser.

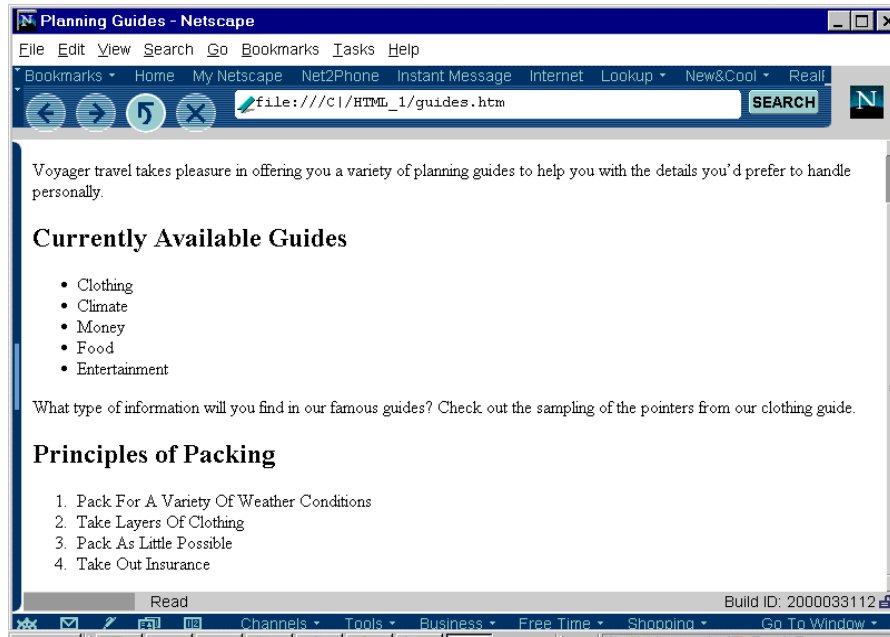


Figure 2-14: Example Of Ordered And Unordered Lists

- Return to the **guides.htm** document in your text editor.

It is possible to “nest” lists; that is, put one list inside another. This creates indented text such as might be seen in an outline. A later exercise will show you how to create numbers and letters in ordered lists to enable you to create such an outline.
- Copy and paste the HTML source code for the Ordered List into the Unordered List as shown.

```

<UL>
<LI>Clothing
<OL>
<LI>Pack For A Variety Of Weather Conditions
<LI>Take Layers Of Clothing
<LI>Pack As Little Possible
<LI>Take Out Insurance
</OL>
<LI>Climate
<LI>Money
<LI>Food
<LI>Entertainment
</UL>

```

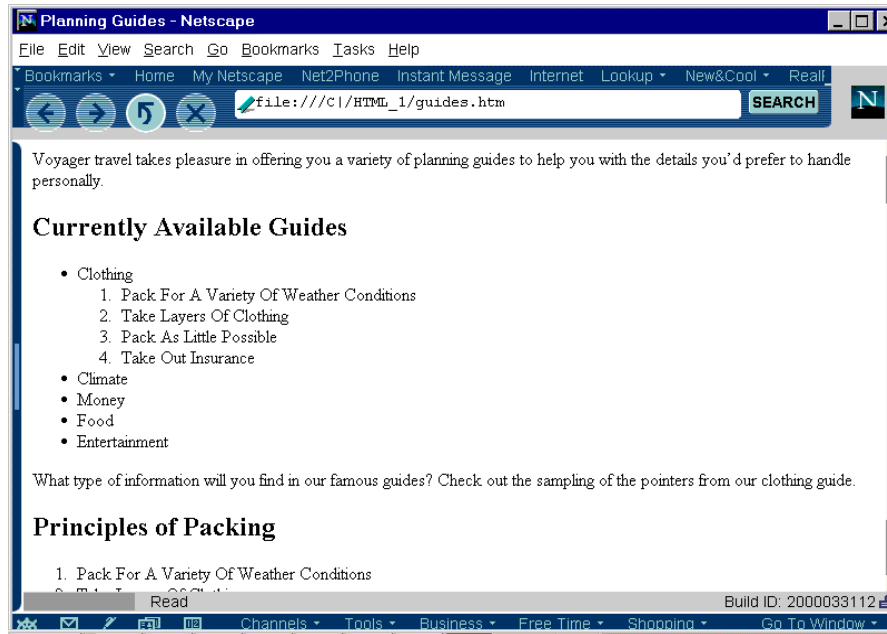


Figure 2-15: Example Of A “Nested” List

List Extensions

As part of enhancements to HTML, a set of attributes is available for modifying the appearance of ordered and unordered lists. These attributes allow you more control over the bullet and number styles in lists.

Unordered Lists

For unordered lists, there are three different bullet symbols available. These are entered as a value for the TYPE attribute, within the LIST ITEM tag. Available types include DISC, CIRCLE, and SQUARE. The DISC is the default bullet shape.

The TYPE attribute was not available in earlier browsers, and in those browsers, the default bullet shape will be displayed. Newer browsers require that the TYPE attribute be included with every LIST ITEM tag for a special bullet shape.



Exercise 2-10: Using Attributes with Unordered Lists

In this exercise, you will use different bullet shapes for an unordered list.

10. Open the **list.htm** document in your text editor.

- Edit the source code to include the TYPE attribute to change the bullet shape.

```
<UL>
```

```
<LI TYPE=square> SQUARES are things like...
```

```

<LI TYPE=square> Stepping stones
<LI TYPE=square> Garden walls
<P>
<LI TYPE=circle> CIRCLES are things like...
<LI TYPE=circle> Ripples in a pond
<LI TYPE=circle> Bubbles in the water
<P>
<LI TYPE=disc> DISCS are things like...
<LI TYPE=disc> Bird baths
<LI TYPE=disc> Crabapples
</UL>

```

- Save the changes.
- Switch to your web browser and open the **list.htm** document.

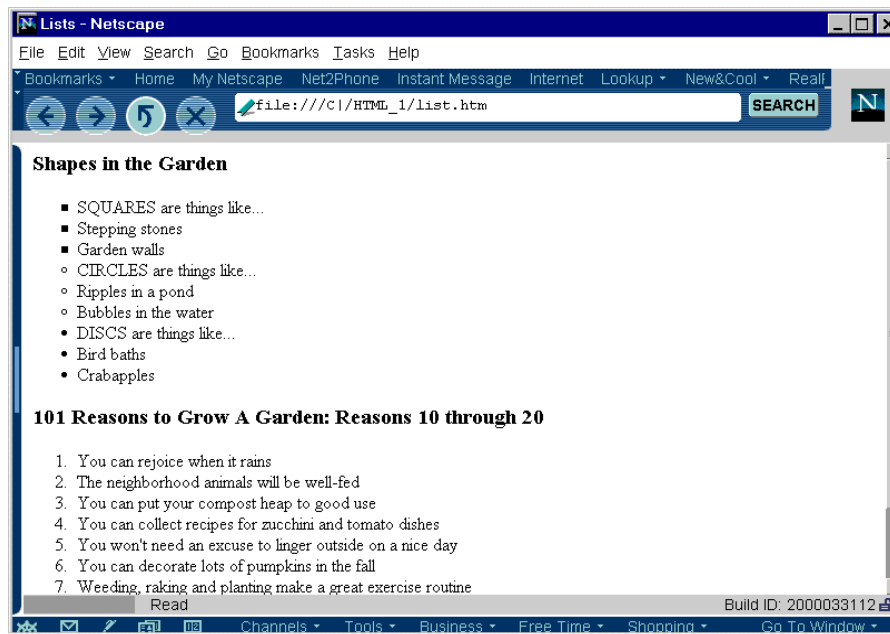


Figure 2-16: Attributes of an Unordered List

Ordered Lists

Ordered Lists can take on the following five attribute to modify the numbering scheme:

- Type=1 is the default numbering scheme (1, 2, 3)
- Type=A orders list items alphabetically with capital letters (A, B, C)
- Type=a orders list items using lower case letters (a, b, c)
- Type=I orders list items using capital Roman numerals (I,II,III)

Type=i orders list items using lower case Roman numerals (i,ii,iii)

Ordered Lists can also be given a starting value, using a START attribute within the tag. This attribute allows you to start your ordered lists at a number other than (the default value of) one. Although you may specify a list to be ordered with letters or Roman numerals, the START value is always entered as an Arabic number (1, 2, 3, etc.)



Exercise 2-11: Using Attributes with Ordered Lists

In this exercise, you will use different notation and starting values to modify an ordered list. Again, the TYPE attribute should be included with every LIST ITEM tag for the specified numbering scheme.

11. Return to the **list.htm** document in your text editor.

- Edit the source code for the ordered list to set the starting value at 10, and change the numbering format as follows:

```
<OL START=10>
<LI TYPE=A>You can rejoice when it rains.
<LI TYPE=A>The neighborhood animals will be well-fed.
<LI TYPE=a>You can put your compost heap to good use.
<LI TYPE=a>You can collect recipes for zucchini and tomato
dishes.
<LI TYPE=I>You won't need an excuse to linger outside on a
nice day.
<LI TYPE=I>You can decorate lots of pumpkins in the fall.
<LI TYPE=i>Weeding, raking and planting make a great exercise
routine.
<LI TYPE=i>You'll eat more vegetables.
<LI TYPE=1>At harvest time, you will have lots of friends.
<LI TYPE=1>You can have fresh flower arrangements all
summer long.
</OL>
```

- Save the changes.
- Switch to the web browser and reload the **list.htm** document.

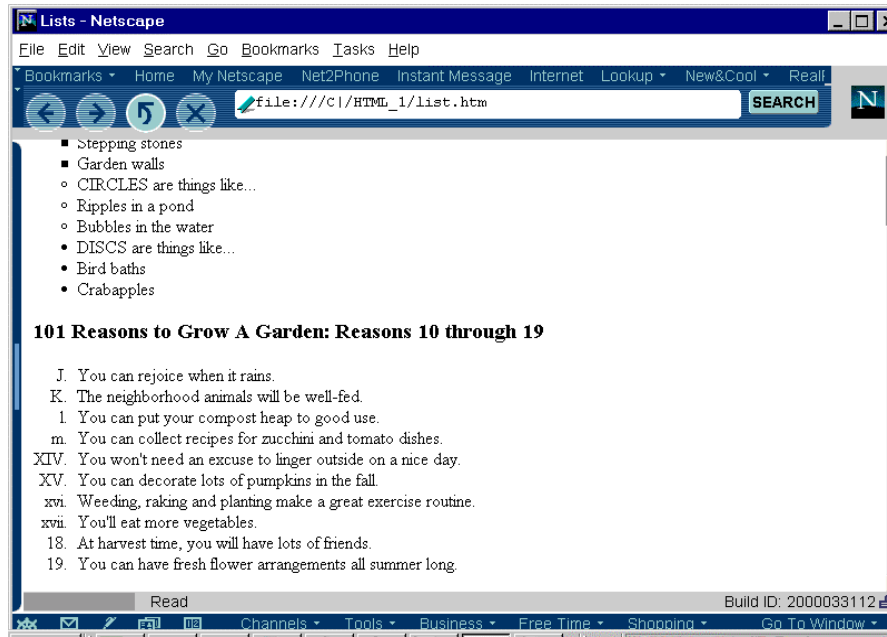


Figure 2-17: Attributes Of An Ordered List

Definition Lists

A Definition List allows you to create the look of a glossary listing with a term followed by its definition. It differs from the previous list types in that it requires three tags. The first is the declaration of the list using the Definition List `<DL>` tag. The term to be defined is identified with a Definition Term `<DT>` tag. The `<DD>` tag declares the Definition Description. The list is closed with the `</DL>` tag. The following shows an example of a Definition List.

```
<DL>
<DT>Hypermedia
<DD>Any one of dozens of audio, video, and image file formats
available via anchors in an HTML document.
```

```
<DT>Hypertext
<DD>Text which contains links to other pieces of text, or to
various types of media, including sound, video, animation, and
graphical images.
```

```
<DT>Hypertext Markup Language (HTML)
<DD>The markup language in which Web pages are written.
</DL>
```

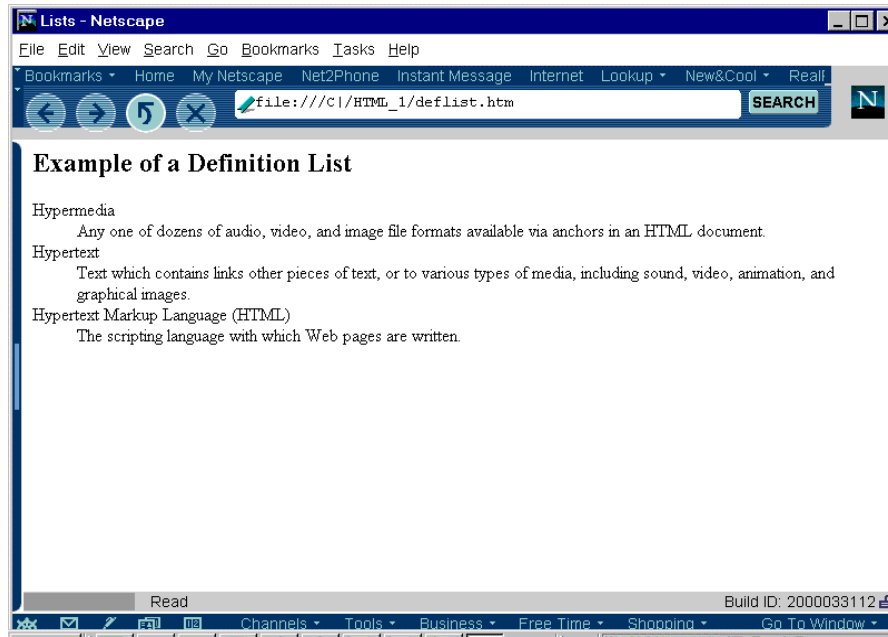


Figure 2-18: Example of a Definition list

Horizontal Rule Tag

The HORIZONTAL RULE tag, `<HR>`, allows you to place horizontal lines in your documents to help organize the document and make it more pleasing to the eye. The `<HR>` tag is an open tag and has optional attributes to allow formatting of the line length (WIDTH), the thickness of the line (SIZE), its alignment on the page (ALIGN), and whether it is flat (NOSHADA) or has a three-dimensional appearance.

The figure below shows the values that each of the attributes may take. Note that the attributes associated with the `<HR>` tag may not be available in early web browsers.

<i>Attribute</i>	<i>Values</i>
Width	0-100% page width, or a pixel value
Size	0-200 Pixels
Align	Left, Right (default=center)
Noshade	none

Figure 2-19: Attributes of the `<HR>` Tag



Exercise 2-12: Adding Horizontal Rules to Documents

In this exercise, you will learn to use the HORIZONTAL RULE tag and its attributes.

12. Return to the **list.htm** document in your text editor.
- At any location in the body of the document, enter the following to create three horizontal rules:


```
<HR>
```

```
<HR WIDTH=50% SIZE=10 ALIGN=left>
```

```
<HR WIDTH=75% SIZE=15 NOSHADE>
```
 - Save the file, switch to your web browser, and reload the **list.htm** document in the Web browser.

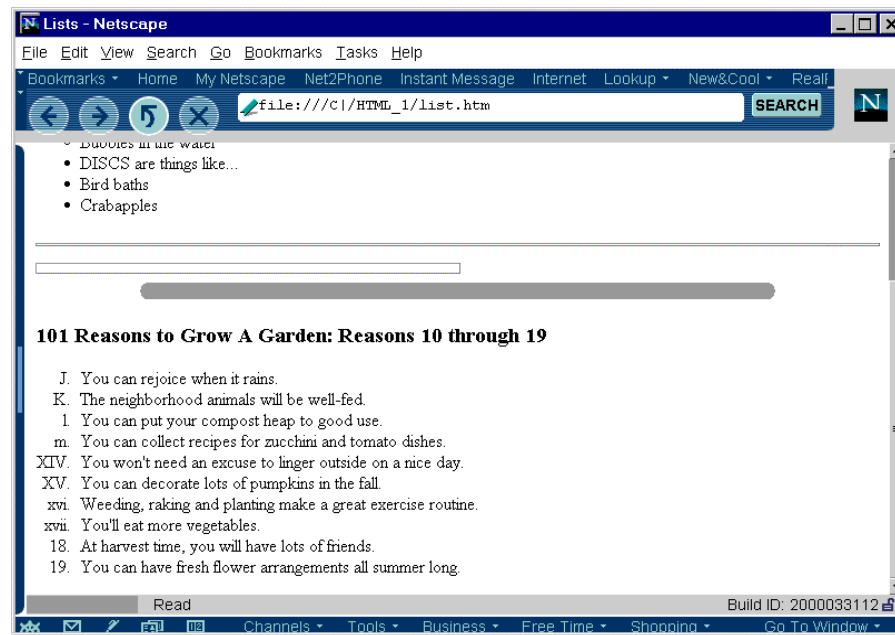


Figure 2-20: Example of Horizontal Rules

Character Entities

Character entities are elements that allow special characters and symbols to be incorporated into your HTML documents. For example, you may want to include a copyright or a trademark symbol. Character entities are always preceded by the ampersand character (&), and followed by a semicolon (;).

Not all browsers support every character entity. Many variations may occur if the user's browser is mapped to a language other than U.S. English. If you have a document that contains a wide variety or large number of special characters, you may consider converting it to a Portable Document Format (PDF) and posting a link to it on another page.

All character entities may be accessed using a numeric, or decimal, value, although most can be accessed with a mnemonic instead. When using the decimal value, the number must be preceded with the hash mark (#).

A list of these values is located in Appendix B of this book, and can also be found at numerous web sites, including

<http://www.natural-innovations.com/boo/doc-charset.html>.

© or **©** produces a **copyright symbol ©**
® or **®** produces a **registered symbol ®**
¥ or **¥** produces a **Yen symbol ¥**
Ç or **Ç** produces a **capital C with a cedilla Ç**

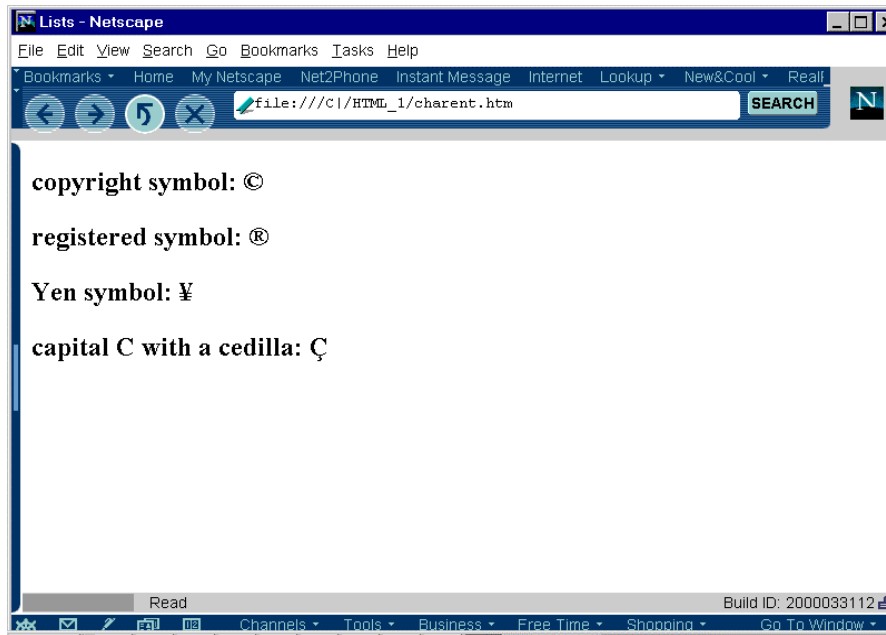


Figure 2-21: Example of Character Entities

You may wish to experiment with a few of the character entities listed in the Appendix.

Review

In this lesson, you have learned how to use HTML markup tags to format and place text in a document. You should be able to construct a basic web page, structure the text using heading levels and lists, and change the text style using formatting tags. You have also been introduced to character entities and the use of the horizontal rule tag.