Chapter 2: Work with Text and Lists

Overview

How to…

- Designate headings
- Identify text
- Create bulleted lists
- Build numbered lists
- Create definition lists
- Add special characters

In Chapter 1 you learned how to create a basic Web page. In this chapter you'll have the chance to put something more than raw text on your page. You'll also learn about a key issue in using HTML and XHTML: the importance of content versus presentation.

Chances are that if you want to create Web pages, your primary concern is how your Web pages will *look*. In other words, you are interested in *presentation.* However, to be able to work with XHTML, you need to understand that *content* comes first. In fact, HTML was originally developed to assist scientists in linking and transmitting research documents over a computer network. Thus, as you'll see in this chapter, the elements are designed to reflect their content. Working with headings, simple text elements, and lists is a great way to begin learning this aspect of HTML. These elements are simple, straightforward, and uncomplicated, but they give you good practice for working with tags. Creating lists will allow you to experiment with the concept of nesting covered in the last chapter. *Nesting* (placing HTML elements inside one another) is an important practice in HTML; it is essential in XHTML. Lists will also provide the opportunity to learn about another important part of working with XHTML, the problem of *deprecated* elements and attributes. However, before tackling that problem, you'll find it easier to work with something easy, like headings.

Designate Headings with <h#> </h#>

The purpose of the heading element is to indicate different heading levels in a document. The tags are made up of an *h* with a number following it. For example, to specify a level one heading, you would write:

```
<h1>This is a level one heading.</h1>
```

A level two heading would look like this:

```
<h2>This is a level two heading.</h2>
```

HTML includes six heading levels: <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. Try typing the following HTML code to see how the six heading levels will display on a Web browser:

```
<html>
     <head>
          <title>The Heading Element</title>
     </head>
     <body>
          <h1>This is a level one heading.</h1>
          <h2>This is a level two heading.</h2>
          <h3>This is a level three heading.</h3>
          <h4>This is a level four heading.</h4>
          <h5>This is a level five heading.</h5>
          <h6>This is a level six heading.</h6>
     </body>
</html>
```

Shortcut Remember the file you created in Chapter 1 named *template.htm?* That file can save you the trouble of retyping the basic page elements every time you create a new page. Just open template.htm, then save it with a new name and, presto, you have a new HTML page ready to work with.

Now, save this file as *headings.htm* and open it in your Web browser. Figure 2-1 shows you what you should see when you display your page.

Figure 2-1: The six heading levels of HTML

As you can see, the text displays in a range of different sizes, from very large to quite small. In fact, it has become commonplace to use the heading elements as an easy way to control font size. However, as you begin to learn to use HTML and XHTML, you will be wise to avoid this shortcut. The reason for this is that it defeats the purpose of the markup, which is to *describe* the content of your page. The <h#> elements were created to identify different divisions within a document, not to display different font sizes. To understand this concept further, consider how the *text elements* work.

PREVIOUS NEXT
PREVIOUS NEXT

Use Text Elements to Describe Text

Text elements derive their names from the intended function or purpose of the text. For example, the

 element stands for strongly emphasized text. Generally, browsers will display text with a bold typeface, indicating that it is *strongly emphasized.*

Identify Text with Text Elements

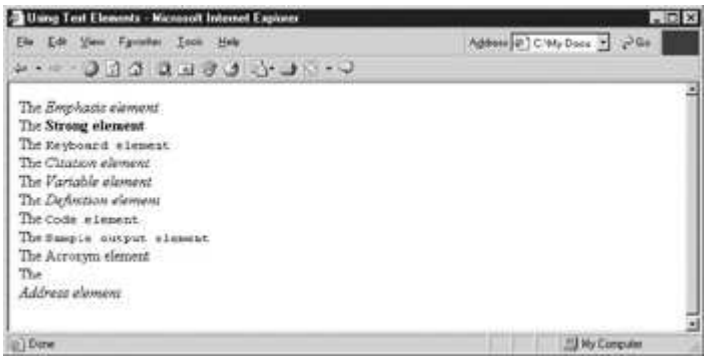
As you will see, each of the following elements not only displays text, but also says something *about* the text it contains:

- ** ** The *emphasis* element is an element used for emphasizing important portions of a document. It generally displays italicized text.
- ** ** The *strong* element indicates stronger emphasis than does and usually displays as bold text.
- **<kbd> </kbd>** Standing for *keyboard*, this element identifies its contents as user input through a keyboard. Generally, it displays as a monospaced font. Some browsers also might display it as bold.
- **<cite> </cite>** The *citation* element identifies a portion of your document as a reference to an outside source.
- **<var> </var>** This element indicates a *variable*, as might be used in computer code.
- **<dfn> </dfn>** The <dfn> element identifies a portion of text as a *defining instance* of a term. It also generally displays in italic.
- **<code> </code>** This element not only displays in a courier, fixed-width font, but also indicates that the text is a portion of computer code.
- **<samp> </samp>** The <samp> element identifies its contents as *sample output*, for example, from a computer program, most often rendering text in a monospaced font.
- **<abbr> </abbr>** Identifies an abbreviation; for example, TX instead of Texas.
- **<acronym> </acronym>** This element identifies text as an acronym, such as S.C.U.B.A. or N.A.S.A.
- **<address> </address>** You would use this element to set apart your address or personal information at the bottom of a Web page. The <address> element also adds a line break before and after the address.

To see each of these elements in action, copy the code listing that follows and save it as *text.htm*. To save time, open template.htm and save it as text.htm. Then just add the lines in between the <body> </body> portion of the code to get the results shown in the following illustration:

```
<html>
     <head>
          <title>Using Text Elements</title>
     </head>
     <body>
          The <em>Emphasis element</em><br />
          The <strong>Strong element</strong><br />
          The <kbd>Keyboard element</kbd><br />
          The <cite>Citation element</cite><br />
          The <var>Variable element</var><br />
```

```
          The <dfn>Definition element</dfn><br />
          The <code>Code element</code><br />
          The<samp>Sample output element</samp><br />
          The <acronym>Acronym element</acronym><br />
          The <address>Address element</address><br />
     </body>
</html>
```



Shortcut The line break <br /> element at the end of each line in the preceding code causes the browser to create a new line of text.

Why are there so many different ways to describe text? Because HTML is not primarily concerned with what someone sees when they view a Web page. Hypertext Markup Language is aimed at defining the content of a document, not its appearance. Do you need to use all of these different elements in your Web pages? Not necessarily, but keep in mind that the World Wide Web didn't get that name for nothing. As your knowledge of HTML and XHTML grows and you become more concerned with making your pages accessible to as many people as possible, you will develop an appreciation for these elements.

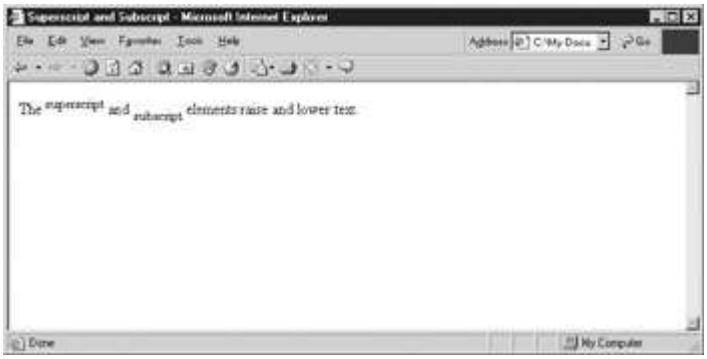
Add Superscripts and Subscripts with <sup> and <sub>

The superscript and subscript elements are reflective of the need to add footnotes for sources as well as the need to be able to write chemical formulas. They are also useful tools for the Web page author. Perhaps you'll need to document a source on your Web site by inserting a footnote reference at the end of a quotation. For this, you'll use the <sup> element. Or perhaps you're a science nut and want to describe the molecular structure of water or carbon dioxide. Then you can create a subscript with the <sub> element.

- **** This element creates a superscript.
- **** This element forces text to display as a subscript.

To see these elements in action, use your template to create a new document, saving it as *sup.htm*. Then insert the following line in between the <body> </body> tags. It will create a line with both superscripts and subscripts, as in the following illustration:

```
<html>
     <head>
          <title>Superscript and Subscript</title>
     </head>
     <body>
     The <sup>superscript</sup> and <sub>subscript</sub> elements
raise and lower text.
     </body>
</html>
```



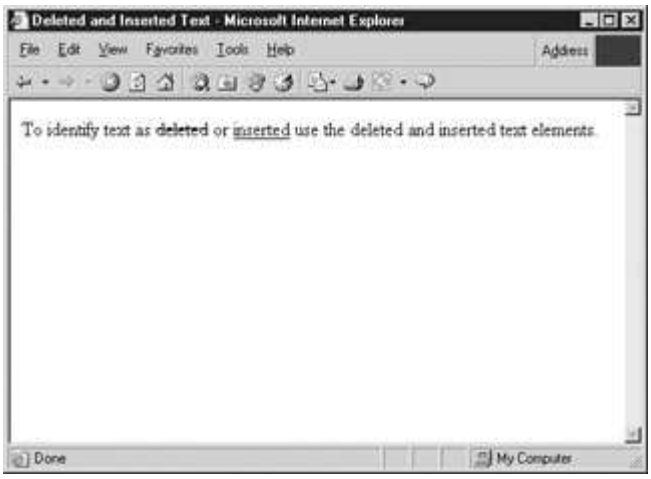
Identify Deleted and Inserted Text with <del> and <ins>

Perhaps no elements better illustrate HTML's document-oriented roots than the <del> and <ins> elements. In the process of editing and rewriting manuscripts, deleted text is displayed with a line through the middle (strikethrough), and newly added text is displayed as underlined. That way, the writers and editors of a document can clearly discern the changes that are being made. HTML's <del> and <ins> perform the same function, causing text to display as strikethrough or as underlined.

- **<del> </del>** The *deleted text* element indicates (by a strikethrough) that the text has been deleted but left in the page.
- **<ins> </ins>** The *inserted text* element is a logical element that indicates new text has been inserted since the document was written.

Create an HTML file like the one that follows and save it as del.htm. Then display it in your browser to see the <del> and <ins> elements in action. Your page should resemble the following illustration:

```
 <html>
     <head>
          <title>Deleted and Inserted Text</title>
     </head>
     <body>
To identify text as <del>deleted</del> or <ins>inserted</ins>
use the deleted and inserted text elements.
     </body>
</html>
```



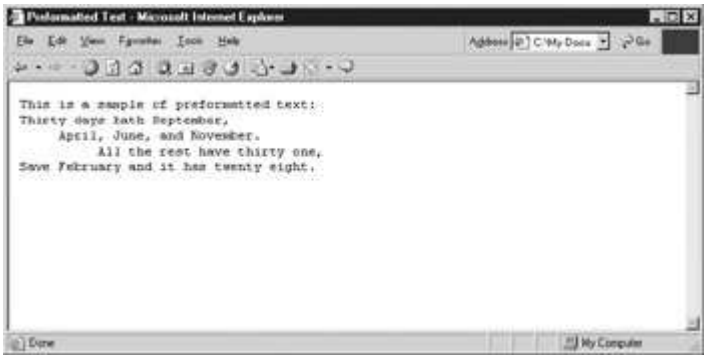
Retain Text Formatting with <pre>

If you try converting large blocks of text from a word processor into HTML, you will notice that all

formatting, spacing, line, and paragraph divisions are lost in the process of conversion. As you'll learn in the next chapter, style sheets will enable you to re-create that formatting. However, HTML does provide an element that preserves line breaks, spacing, and white space. Using the *preformatted text* element, <pre>, you can instruct a browser to display your text exactly as you enter it. Simply enclose the text between the <pre> </pre> tags, and the browser will leave it as is. The downside of this element is that the browser will also display the text in a typewriter-style or monospaced font. However, in the next chapter you'll learn how to use style sheets to override this and instruct the browser to use a different font.

To see how <pre> works, create and save a page, naming it *pre.htm.* Then type in the following code. As you can see in the illustration that follows, the browser retains whatever spacing you gave the characters:

```
<html>
     <head>
          <title>Preformatted Text</title>
     <head>
     <body>
<pre>This is a sample of preformatted text:
Thirty days hath September,
     April, June, and November.
          All the rest have thirty one,
Save February and it has twenty eight.</pre>
     </body>
</html>
```



The text elements give you limited influence over how your text is displayed. In the next chapter you will expand that influence. But before moving into that arena, it will be helpful for you to learn a simple way to structure the information you put on your Web pages.

PREVIOUS NEXT
PREVIOUS NEXT

Organize Your Content with Lists

HTML lists are a great way to organize content on your Web site. Whether you want to list the ingredients of your favorite recipe, create a page of links with their descriptions, or offer step-by-step instructions for washing a dog, you can use list elements to put your material in order. Another benefit that comes from working with lists is that it will give you some experience in nesting your elements.

Create Unordered Lists with <ul> and

An *unordered list* is what you might know as a bulleted list. The items in the list are not organized by number or letter. Rather, each item has a small symbol (bullet) in front of it. In HTML, you can create unordered lists that have a solid disc, a circle, or a square.

Use <ul> and <li> to Create an Unordered List

You need two elements to create an unordered (bulleted) list:

- ** ** The *unordered list* element creates the list.
- ** ** You specify individual items on the list with the *list item* element.

Use your template to create a new HTML document. Save it in your HTML reference directory as *ulist.htm.* Then, in between the <body> </body> tags:

1. Type an opening *unordered list* tag: **<ul>**.
2. Next, add a set of *list item* tags: ** **.
3. For each new list item, add another set of ** ** tags.
4. When the list is complete, type a closing **
** tag.

The code for a simple unordered list might look something like what you see here:

```
<html>
     <head>
          <title>Unordered Lists</title>
     <head>
     <body>
          <ul>
              <li>This is the first item</li>
              <li>This is the second item</li>
              <li>This is the third item</li>
          </ul>
     </body>
</html>
```

When you display your page in a browser, it will produce a bulleted list, like the one in the illustration here:

Did You Know?—W3C Stands for World Wide Web Consortium

The W3C is an international organization responsible for setting the standards concerning the Internet, World Wide Web, HTML, XHTML, and any other related technologies.

You'll notice when you display the list that your browser supplies a solid disc as the bullet for each item. If you would prefer a circle or a square, you can specify it by using the *type* attribute. However, before you begin experimenting with different bullet types, you need to understand something about the type attribute and why the W3C doesn't want you to use it.

Understand Deprecated Elements and Attributes

Over the years since HTML was developed, browser companies (specifically Netscape and Internet Explorer) have *extended* it to include what might be known as *presentational elements*. For example, Netscape introduced the <blink> element, which causes text to blink on and off. Internet Explorer introduced the <marquee> element, which makes text scroll across the screen like a movie marquee. Some of these *proprietary elements* were supported only by their respective browsers; others were eventually incorporated into the HTML standard. For example, frames are a Netscape innovation that ultimately became part of HTML. However, as more and more presentational elements were added, HTML moved farther and farther away from its original purpose—describing document content.

The W3C has been working for several years to eliminate any HTML markup that governs appearance. The reasons for this move, and the alternative method for addressing presentation (Cascading Style Sheets), will be introduced in Chapter 3, but it's important to understand now that certain elements and attributes have been officially *deprecated* by the W3C. Deprecated is just a fancy way of saying that they don't want you to use those elements and attributes anymore. Don't worry, though. Nobody from the "Web Police" will knock on your door if you use these deprecated portions of HTML. However, the possibility exists that browsers may eventually stop supporting them. If your Web pages are built on those elements and attributes, you might find yourself having to redo them down the line.

Use the *type* Attribute to Specify Different Bullets (Deprecated)

The *type* attribute enables you to choose between a disc, circle, or square as the bullet for your list items. Even though this attribute has been deprecated in favor of Cascading Style Sheets, you'll find it helpful to learn how to use it. Just as working with lists will help you understand the concept of nesting, using the type attribute will enable you to become acquainted with attributes and how they work.

There are two different ways you can use the type attribute in a list: globally (the entire list) or individual list items. To specify the bullets for an entire list, put the attribute inside the opening <ul> tag:

- To specify a square: **<ul type="square">**
- To specify a circle: **<ul type="circle">**
- To specify a disc: **<ul type="disc">**

You also can control the type of bullet for each separate item by putting the attribute inside the <li> tag. For example, the following code will supply a different bullet for each item.

```
<ul>
     <li type="square">This supplies a square.</li>
     <li type="circle">This supplies a circle.</li>
     <li type="disc">This supplies a disc.</li>
</ul>
```

Project 2: Create a Multilevel List

To create a list with multiple levels, you simply nest one or more unordered lists inside each other. Save ulist.htm with the new name *ulist2.htm.* Now make a multilevel list from it:

1. Change the <title> to read **Multi-Level Unordered Lists**.
2. In the first <ul> tag, just below the body tag, enter **type="square"**.
3. Inside the first set of <li> tags, type **<ul>**.
4. Inside this new <ul> tag, enter **type="circle"**.
5. Add a line that says **
This is a sub point.
**.
6. Add another line that reads: **
This is a sub point.
**.
7. Close out the sublist with **
**.
8. Add a set of subpoints to the second main point. Use **type="disc"** to specify the disc bullet.
9. Add a set of subpoints to the third main point. Use **type="square"** to specify the square bullet.

Your list should look like this:

```
<html>
     <head>
          <title>Multi-Level Unordered Lists</title>
     <head>
     <body>
          <ul type="square">
               <li>This is the first item
                    <ul type="circle">
                         <li>This is a sub point</li>
                         <li>This is a sub point</li>
                    </ul>
               </li>
               <li>This is the second item
                    <ul type="disc">
                         <li>This is a sub point</li>
                         <li>This is a sub point</li>
                    </ul>
               </li>
               <li>This is the third item
                    <ul type="square">
                         <li>This is a sub point</li>
                         <li>This is a sub point</li>
                    </ul>
               </li>
          </ul>
     </body>
</html>
```

If you want to add another level of sub points, you can do it by nesting another complete list inside each <li> element where you want the next level. A multilevel list such as the one rendered here will resemble the following illustration:

Caution You can get away with putting your sublist below the list item to which it applies instead of inside the tags. Most browsers will still display it properly. However, it's sloppy HTML and will not meet the more stringent standards of XHTML.

Create Ordered Lists with <ol> and

What if you want to display an outline with numbers and letters delineating the various points? Or perhaps you want to create a list of instructions in numbered sequence. Lists that arrange items in sequence by number or letter are called *ordered lists* in HTML and are quite similar in their structure to unordered lists. To create an ordered list, you need the following:

- ** ** This is the ordered list element.
- ** ** This is the list item element.

Use <ol> to Create a Numbered List

You can create a simple numbered list by enclosing a series of list items inside the ordered list element, as in the following sample page. Use your template to open a new HTML page and save it as *olist.htm.* Then type in this code:

```
<html>
     <head>
          <title>Ordered Lists</title>
     </head>
     <body>
          <ol>
               <li>Ordered lists display items with numbers.</li>
               <li>But HTML doesn't sort the items.</li>
               <li>It only numbers them.</li>
               <li>You have to do the arranging yourself.</li>
          </ol>
     </body>
</html>
```

After you save your page, display it in your browser. You should see a numbered list like this:

When you display this in a browser, you will notice that the list is numbered in sequence. The actual format depends on the browser you use, but normally it is a simple list with Arabic numerals. One important difference with numbered lists is that if you nest the lists to create multiple levels, the browser uses the same numbering system throughout; it does not automatically change to a different style with each level. You must use the type attribute or Cascading Style Sheets to specify any changes you want.

Use the type Attribute to Specify Numbers or Letters (Deprecated)

Just as you can instruct the browser to use different types of bullets in an unordered list, you can tell the browser what types of letters or numbers to use. This is very useful if you want to produce an outline in HTML, as you have a nice range of choices available. You specify numbers or letters with the *type* attribute, just as you did for unordered lists.

- **<ol type="I">** Capitalized Roman numerals
- **<ol type="i">** Lowercase Roman numerals
- **<ol type="1">** Numbers (default)
- **<ol type="A">** Capital letters
- **<ol type="a">** Lowercase letters

As with unordered lists, if you place the type attribute inside the <ol> tag, you can specify your preferences for the entire list. If you place the attribute inside a <li> tag, it will change only that particular list item.

Use the *start* Attribute to Choose a Starting Number (Deprecated)

What if you want to begin a list at a point other than 1 or A? All you need to do is include the *start* attribute at the point where you want to change the number or letter. The browser will start the list at the number you choose and continue numbering from that point. For example, if you want to start a list at the number 23, you might do it this way:

```
<ol type="1" start="23">
```

Create a new HTML document and save it as *olist2.htm.* Now, try typing the following code and displaying it in your browser to see what a list like this might look like if you did it with Roman numerals, starting at number 10:

```
<html>
     <head>
          <title>Ordered Lists</title>
```

```
     </head>
     <body>
          <ol type="I" start="10">
               <li>Item ten</li>
               <li>Item eleven</li>
               <li>Item twelve</li>
               <li>Item thirteen</li>
          </ol>
     </body>
</html>
```

When you display this in your browser, the list will begin with the Roman numeral "X," as in the following:

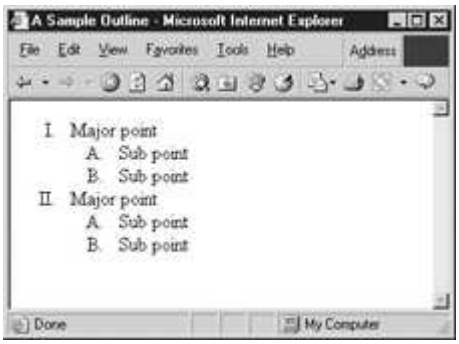
Caution Beware of a little quirk with the start attribute. It makes no difference whether you are using numbers, letters, or Roman numerals in your list. To specify a starting point, you always do it with Arabic numbers (1, 2, 3), not the characters that will display in the outline. If you modify the preceding code to use the Roman numeral for 10, start="X", you'll find the browser ignores the instruction entirely.

Project 3: Create an Outline with

To present material in outline format in HTML takes a bit more thought and planning than it does with an average word processor, but it will produce satisfying results. The following sample HTML code will produce a 2-point Roman numeral outline with lettered subpoints. Save it as *olist3.htm*:

```
<html>
     <head>
          <title>A Sample Outline</title>
     </head>
     <body>
          <ol type="I">
               <li>Major point
                    <ol type="A">
                         <li>Sub point</li>
                         <li>Sub point</li>
                    </ol>
               </li>
               <li>Major point
                    <ol type="A">
                         <li>Sub point</li>
                         <li>Sub point</li>
                    </ol>
               </li>
          </ol>
     </body>
</html>
```

The following illustration shows how an outline would look on your Web browser:



Tip Make sure you have properly nested your tags. Your subpoint list should be placed in between the list item tag it relates to. If you overlap them or leave out a closing tag, your outline will not display the way you want it to.

Create Definition Lists with <dl>, <dt>, and <dd>

Perhaps the most versatile type of list is the definition list. Originally developed as a means of creating glossaries in a document, this type of list lends itself most readily for use on Web sites. You will use a different set of elements to create this type of list:

- **<dl> </dl>** The *definition list* element creates the list.
- **<dt> </dt>** The *definition term* element identifies the term to be defined.
- **<dd> </dd>** The *definition description* element sets off the definition.

To see how a definition list works, try typing in the following code and displaying it in your browser. Save the file as *dlist.htm.* Notice that the code also includes some of the text formatting elements covered earlier in this chapter:

```
<html>
     <head>
          <title>Definition Lists</title>
     </head>
     <body>
          <h1>HTML Terms</h1>
               <dl>
                    <dt>HTML</dt>
                    <dd>HyperText Markup Language</dd>
                    <dt>Element</dt>
                    <dd>Describes document content.</dd>
                    <dt>Attribute</dt>
                    <dd>Identifies a characteristic of an element</dd>
                    <dt>Value</dt>
                    <dd> Gives details of a specific attribute</dd>
               </dl>
     </body>
</html>
```

Displayed in a browser, a definition list will look similar to the one in the following illustration:

As you can see, this type of list can be very useful; you could use it to provide a page of links, with a special paragraph explaining each link. It also might be used to create a directory (often called a *site map*) of your Web site. The possible uses of the definition list are limited only by your imagination.

PREVIOUS NEXT
PREVIOUS NEXT

Insert Special Characters on Your Page

Have you ever seen the copyright notice at the bottom of a Web page and wondered where they found the © symbol? After all, it's not on the keyboard. How did they get it onto the page? Let's say you're doing a mathematical formula and you want to place a < or > sign on your page. If you just type in the characters, the Web browser will interpret them as HTML markup. Not only will the characters not appear, they very likely will cause your page to display incorrectly. The same holds true with marks like the ampersand (&) and quotation marks. If you have JavaScript or another scripting language on your page, browsers can misinterpret these characters, too. So how do you instruct a Web browser to display special characters rather than ignoring them or treating them as code? Perhaps you want to include a word from a foreign language and you need a special symbol or accent. What do you do? The way around this and similar problems is through the use of *entities*.

Understand Entities

Simply put, an *entity* is a character that either is not accessible through your keyboard or one that will be incorrectly interpreted by the browser. However, these characters are resident in your computer's system, and you can access them through the use of special codes.

The source for entity codes is the ISO-Latin1 character set. ISO stands for the *International Standards Organization*. The rest of the term denotes that this character set is derived from the Latin (or Roman) alphabet. Of course, there are many other character sets, derived from different alphabets. But the one you are most likely to use is ISO Latin-1; this also is the default character set for the Web.

Note Entities described by numbers are called numeric entities. Entities described by descriptive terms are called character entities. Many special characters are represented by both types. For example, the copyright symbol © can be written either as a numeric entity, ©, or as a character entity, ©. Whichever you use, a Web browser will recognize it as the entity for the copyright symbol and display the symbol in its place.

An entity must be constructed properly for the browser to recognize it. It always begins with an

ampersand (&) and closes with a semicolon (;). In between, you insert either a numeric code or a logical descriptive term. In addition, numeric entities must have the number symbol (#) preceding the entity number. Also, entities are case sensitive; always type them in exactly as you see them on a reference chart. Incidentally, some older browsers might not recognize particular character entities. It's always a good idea to test your page in different browsers to be sure of its compatibility.

Tip For an extensive chart of numeric and character entities, download Appendix C from the author's Web site at www.jamespence.com.

Insert an Entity in a Web Page

A practical way to experiment with entities is by adding a copyright notice to your Web page. To use the entity for the copyright © symbol, follow these steps:

1. Open template.htm and save it as *entity.htm*.
2. In the <body> section of the document, type **Copyright 2015**.
3. After the word *Copyright*, type the ampersand character, &.
4. Enter either the numeric or character code. For example, for the copyright symbol you would type **copy** or **#169**.
5. Close out the code by typing a semicolon. Your text should look like this:

```
Copyright &copy; 2015 or Copyright &#169; 2015
```

However, on a Web browser it will display this way: Copyright © 2015. Try it out on a sample page. Remember to put the entities in the actual text of your Web page, not in the tags. It would be incorrect to write <h6 ©>Copyright 2015</h6>. If you do put the entity in the wrong place, the browser will just ignore it. Type in the following code for a demonstration of how entities work and for a sample of what shows up when they are entered correctly:

```
<html>
     <head>
          <title>Sample Entity Display</title>
     </head>
     <body>
          Copyright &copy; 2015
          My Trademark &#174; is a registered trademark.
     </body>
</html>
```

When you save and display this page, you'll notice, as in the illustration that follows, that the code for your entity is replaced by the special character that it represents:

Quick Reference: Text, Lists, and Entities

In this chapter you have learned how to use headings, text elements, lists, and entities. More important, you have mastered the concept of nesting and have learned how to use elements, attributes, and values in a Web page. As you practice these techniques, use the following reference charts to help you remember what you have learned.

Work with Text

The elements which allow you to work with text are straightforward and easy to use. To cause text to display using the elements listed in the table below, simply enclose it between the proper set of tags. For example, to emphasize text, place it inside the element like this: Emphasized Text.

To Do This	Use This
Indicate a heading	<h#> </h#> (may use #1-6)
Emphasize text	
Strongly emphasize text	
Identify text as keyboard input	<kbd> </kbd>
Identify text as a citation	<cite> </cite>
Identify text as a variable (as in a computer program)	<var> </var>
Identify text as a defining instance (of a term)	<dfn> </dfn>
Identify text as sample output	<samp> </samp>
Identify text as an acronym	<acronym> </acronym>
Identify text as an abbreviated term	<abbr> </abbr>
Set apart a portion of text that contains author information	<address> </address>
Identify text as computer code	<code> </code>
Create a superscript	
Create a subscript	
Preserve your spacing and formatting	<pre> </pre>
Indicate inserted text	<ins> </ins>
Indicate deleted text	<del> </del>

Create Lists

Lists are easy to create and provide a useful tool for menus, site directories, lists of links, and outlines. There are three different types of lists to choose from: ordered (numbered), unordered (bulleted), and definition (glossary). The trick to doing lists well is remembering to nest your elements properly. The elements necessary for creating lists are included in the following table:

To Do This	Use This
Create an ordered (numbered) list	
Create an unordered (bulleted) list	

Create a definition list	`<dl> </dl>`
Add a list item	`<li> </li>`
Add a definition term (definition list only)	`<dt> </dt>`
Add a definition description (definition list only)	`<dd> </dd>`
Specify a starting number (ordered list only)	`start="#"` (Deprecated)
Specify a bullet type (unordered list only)	`type="disc | circle | square"` (Deprecated)
Specify a number type (ordered list only)	`type="a | A | i | I | #"` (Deprecated)
Set the value for a particular list item	`value="#"` (Use Arabic numerals only, no matter what type of numbering is used in the list.) (Deprecated)

Insert Entities

In the following table are some entities that you might be likely to use. Notice that some have both numeric and character versions, while others only are represented by numeric codes. This list is not complete, but includes some of the most common characters. Download Appendix C for a more extensive list (www.jamespence.com).

To Display This	Use This Numeric Entity	Use This Character Entity	This Character Will Display
Exclamation	&#33;		!
Quotation	&#34;	&quot;	“
Number	&#35;		#
Dollar	&#36;		$
Percent	&#37;		%
Ampersand	&#38;	&amp;	&
Apostrophe	&#39;		‘
Asterisk	&#42;		*
Plus sign	&#43;		+
Comma	&#44;		,
Hyphen	&#45;		-
Period	&#46;		.
Slash	&#47;		/
Colon	&#58;		:
Semicolon	&#59;		;
Less than	&#60;	&lt;	<

Equal sign	&#61;		=
Greater than	&#62;	&gt;	>
Question	&#63;		?
“At” sign	&#64;		@
Left bracket	&#91;		[
Backslash	&#92;		\
Right bracket	&#93;		]
Left curly brace	&#123;		{
Right curly brace	&#125;		}
Tilde	&#126;		~
Copyright	&#169;	&copy;	©
Trademark	&#174;	&reg;	®

PREVIOUS NEXT