List markup

Lesson 6

A list is just a list is just a list, right? The biggest list formatting decision you normally have to make is whether a list needs bullets or numbers in front of the items it contains. HTML provides you with three different list types:

- Ordered lists, which are really just numbered lists
- Unordered lists, which are bulleted lists
- Definition lists, which associate a term (such as HTML) with a definition (such as Hypertext Markup Language)

Lists are very easy to create in HTML, and they have another advantage: Web browsers always display lists a bit differently from the text around them. This means that when you add lists to your Web pages you not only add numbers or bullets to your content, but you also change the landscape of your Web page a bit and make it more interesting.

A good example of the difference a list can make in a Web page can be found in the ACME Widgets Web Page you've been building. The current code beneath the Order Your Widgets Today heading uses line breaks (`<br>`) within a paragraph to list three different ways to order widgets, as shown in Figure 6-1.

Figure 6-1: Line breaks can separate different items in a group.

Now see what happens when HTML list code makes the three different ways of ordering into a numbered list, as shown in Figure 6-2:

Figure 6-2: A numbered list groups related items better, and changes the look and feel of the page.

Not only is each of the three ways to order widgets preceded by a number, but the entire list is also indented from the left margin. As you'll learn next, it's truly simple to build this numbered list.

All HTML lists function in more or less the same way:

- They tell the browser you are beginning a list of a particular type
- They identify the different items in the list
- They tell the browser your list is complete

You can do all of this with the few simple HTML elements you'll learn about in the remaining pages of this lesson. In short, lists are a wonderful tool to have in your HTML arsenal and you should use them whenever appropriate.

Ordered lists   Page 2 of 6

Ordered lists are simply numbered lists. Remember that there is a simple equation for building any list in HTML: define the kind of list you want to build and then identify the items in the list. Ordered lists use two different elements to complete this equation:

- The ordered list element (** . . . **) specifies that list type and begins and ends the list.
- The list item element (** . . . **) identifies the items in the list.

So, the code for a simple ordered list looks like this:

```
<ol>
<li>First item</li>
<li>Second item</li>
<li>Third item</li>
</ol>
```

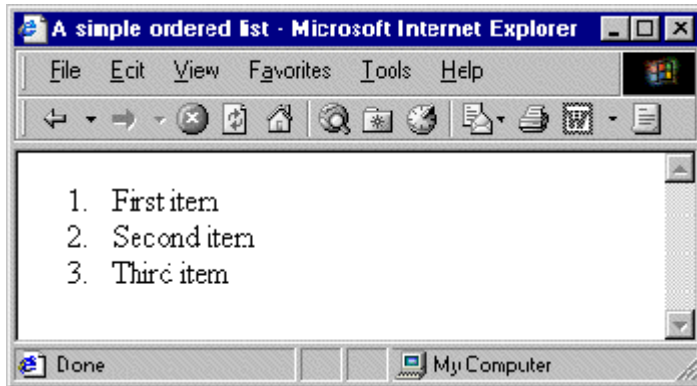Figure 6-3 shows how a Web browser displays this code:

Figure 6-3: A basic ordered list is a numbered list.

Notice that the Web browser automatically numbers the items in the list. The number assigned to each item is based on its order in the HTML code that builds the list. Look at Figure 6-4 to see what happens when you swap the first item with the second item in the list, as in the following bit of code:

```
<ol>
<li>Second item</li>
<li>First item</li>
<li>Third item</li>
</ol>
```
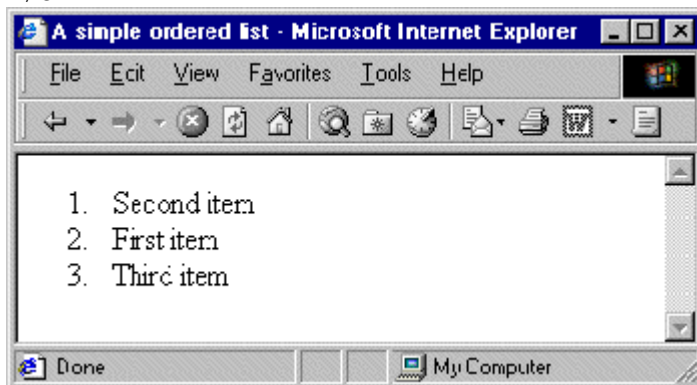

Figure 6-4: Web browsers assign item numbers based on an item's place in the list code.

The numbers don't change even when the items in the list are rearranged. This means you can add items to a list and rearrange them without having to manually renumber them. And that's all there really is to building a basic numbered list.

The default marker for any ordered list is numbers, and an ordered list always begins with the number 1 by default. The `<ol> . . . </ol>` element takes two different attributes, in case you want a different kind of marker, or want to begin your list items with a number other than 1.

The `type=` attribute allows you to specify what kind of list marker you want to use to mark the items in your list. The possible values are:

- **1**: numbers (1, 2, 3, 4, 5)
- **a**: lowercase letters (a, b, c, d, e)
- **A**: uppercase letters (A, B, C, D, E)
- **i**: lowercase Roman numerals (i, ii, iii, iv, v)
- **I**: uppercase Roman numerals (I, II, III, IV, V)

The `start=` attribute takes a number value so you can specify what number you want your list items to begin with. In the following code the attributes added to this simple list changes the markers to uppercase letters and starts the list at the fifth number -- or, in this case, the fifth letter:

```
<ol type="A" start="5">
<li>First item</li>
<li>Second item</li>
<li>Third item</li>
</ol>
```

Figure 6-5 shows how a browser displays this code. Notice that the list begins with the letter E because it is the fifth character in the alphabet and the value for `start=` is set at 5.
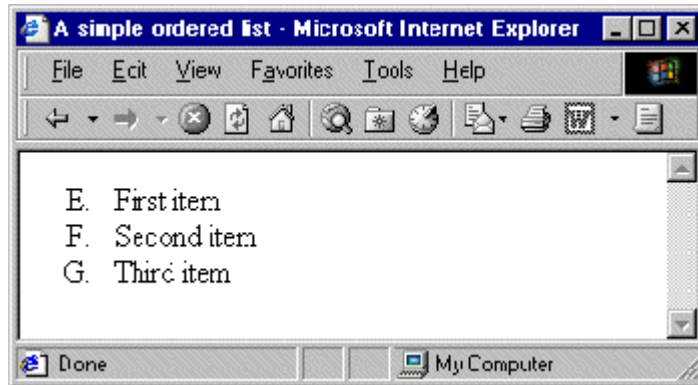

Figure 6-5: The type and start attributes help you specify how numbered lists should be displayed.

Nesting lists

One of the cool things you can do with just about any list is to nest it within another list. The browser indents each nested list further to the right so that you get an outline effect like the one shown in Figure 6-6.
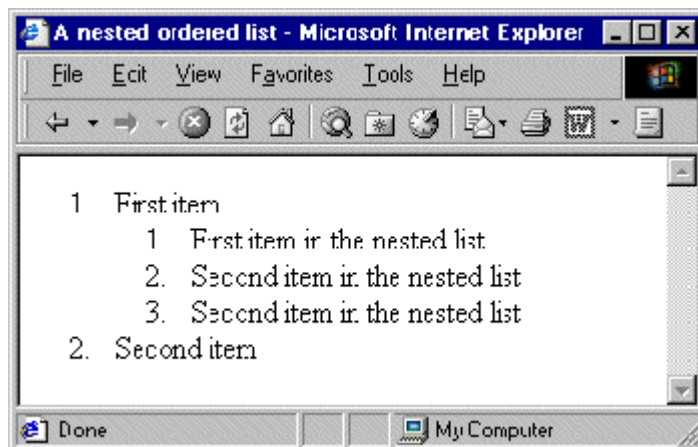
Figure 6-6: Nested lists create an outline effect.

If you choose to nest lists, be sure you nest one list completely within the other and that you close a nested list before you close the list that holds it, as in this bit of code:

```
<ol>
<li>First item</li>
    <ol>
        <li>First item in the nested list</li>
        <li>Second item in the nested list</li>
        <li>Second item in the nested list</li>
    </ol>
<li>Second item</li>
</ol>
```

An unordered list is just like an ordered list, but without the order. Web browsers display unordered lists as bulleted lists and you use these two elements to build them:

- The unordered list element (** . . . **) begins and ends the list.
- The list item element (** . . . **) identifies items in the list.

If you think these elements resemble the tags for an ordered list, you're right on target. The two kinds of lists share the <li> . . . element to identify list items, and the unordered list begins and ends with <ul> . . . instead of the <ol> . . . . of the ordered list.

A simple unordered list looks like this:

```
<ul>
<li>First item</li>
<li>Second item</li>
<li>Third item</li>
</ul>
```
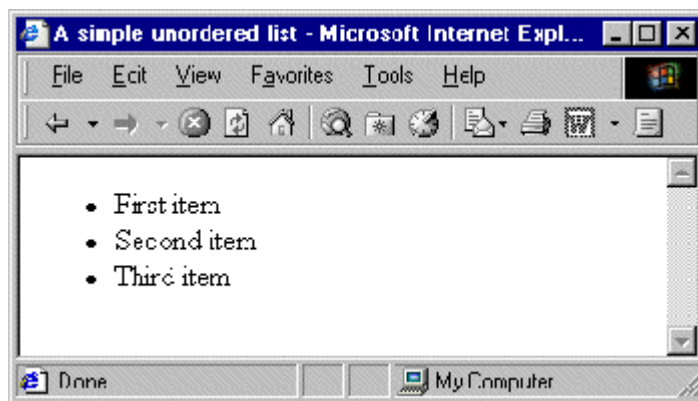
Figure 6-7 shows how a browser displays this code.

Figure 6-7: A simple unordered list with bullets opens with the <ul> tag.

The default marker for an unordered list is a round, filled bullet. You can use the type= attribute with

the `<ul>` . . . `</ul>` element to change the bullet. You have several options:

- **disc** specifies a round filled bullet.
- **square** specifies a square filled bullet.
- **circle** specifies a round hollow bullet.

Figure 6-8 shows how the simple list looks in a Web browser if it has the bullet type square, as in this bit of code:

```
<ul type="square">
<li>First item</li>
<li>Second item</li>
<li>Third item</li>
</ul>
```
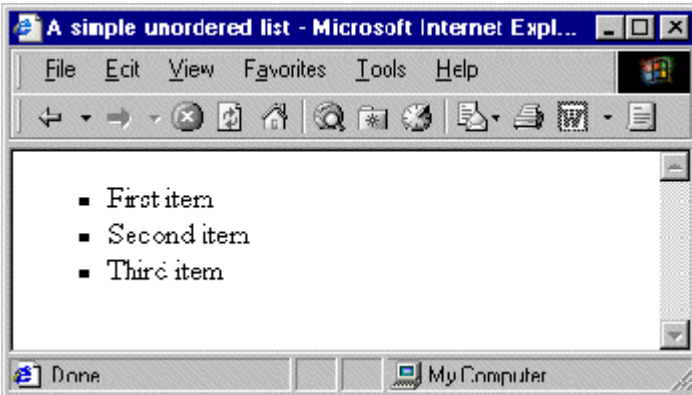

Figure 6-8: This is a simple unordered list marked with square bullets, as specified by the type= attribute.

When you nest unordered lists within other unordered lists, most browsers alternately use one of the three bullet types, as shown in Figure 6-9.
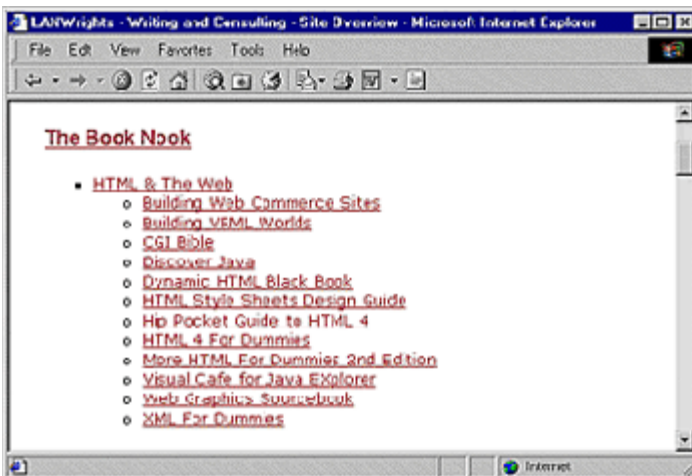

Figure 6-9: Each nested unordered list is marked with different bullet types.

Definition lists   Page 5 of 6

You build a definition list a bit differently than you do an ordered or unordered list. The equation for building the list is the same, but you need two elements to identify each item in the list instead of one. This brings the total number of elements you need to

The many uses of definition lists

build any definition list to three:

- The definition list element (**<dl> . . . </dl>**) begins and ends the list.
- The definition term element (**<dt> . . . </dt>**) identifies terms in the list.
- The definition description element (**<dd> . . . </dd>**) identifies a description to go along with the term.

The basic syntax for a definition list looks like this:

```
<dl>
<dt>term</dt>
<dd>description</dd>
</dl>
```

You can have more than one description (<dd> . . . </dd>) for any term (<dt> . . . </dt>). The browser just assumes that all descriptions it sees after a term go with that term. This bit of code uses definition list markup to describe glossary terms:

```
<dl>
<dt>HTTP</dt>
<dd>Hypertext Transfer Protocol</dd>
<dd>The protocol that allows Web browsers and Web
    servers to communicate with one another.</dd>
<dt>SMTP</dt>
<dd>Simple Mail Transfer Protocol</dd>
<dd>The protocol that allows e-mail clients and
    e-mail servers to communicate with one another
    when sending mail.</dd>
</dl>
```

Figure 6-10 shows how a Web browser interprets this simple definition list.
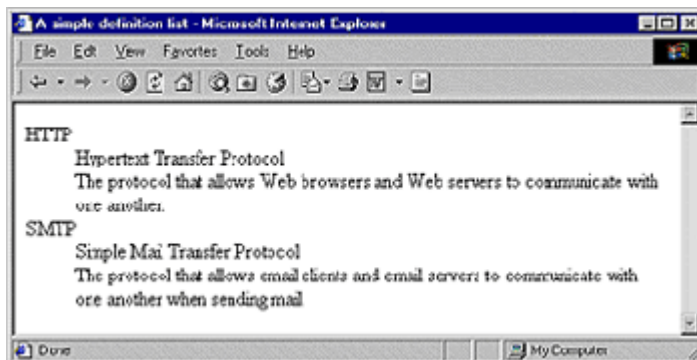

Figure 6-10: A simple definition list can be used to build a glossary, among other things.

In general, lists are very useful page components that are easy to create and maintain. Nested lists can be a bit more complex, but they give you a way to include outlines in your Web pages and show several levels of relationships between different bits of content. Use lists and use them often.

You can use definition lists to build more than glossaries. A definition list really has two levels of information. The topmost level uses definition terms to group related information, while the next level down uses definition descriptions to group information related to a single term in the larger definition list. You can use definition lists to describe articles and the magazines they came from, merchandise part numbers and descriptions, or any other kinds of data that are related to one another. In the assignment for this lesson you'll use a definition list to describe contact information.

Lists are some of the easier constructs you'll come across in HTML and they are some of the most useful. Any time you need to group like items, look to a list to do so. Also remember that something that might not

immediately seem like a list -- a group of magazine articles, for example -- may actually fit well into the list and help users process the information better.

In the assignment for this lesson, you'll add two lists to the Web page you've been building. Be sure to take this opportunity to practice your list building before you move onto the next lesson. In Lesson 7, you'll learn how to add a little pizzazz to your site by changing your font colors, faces, and sizes. It's tedious work but the effort is well worth it.

As always, if you have questions about the HTML you're working with, or about any other aspect of Web page building, be sure to post to the message board so that your instructor and fellow students can help you along. If you find that you're really catching onto this HTML stuff, be sure to share your understanding on the message board as well.