



Using Semantic, Structured HTML to Create Web Pages

Web pages can be created in many ways using anything from Notepad to online site builders to WYSIWYG (what you see is what you get) web editors, but a great web page requires more. Your choice of Expression Web can make creating web pages that work well in a variety of web browsers easy.

HTML is short for HyperText Markup Language, which you use to create ordered, structured pages with headings, body text, lists, and data tables. This structure forms the basis for not only how your page will display but also tells the visitor what is important on your page.

Web pages consist of a doctype declaration that tells the browser which version of HTML to apply to your page so it knows how to display the page in your browser; this is followed by the first HTML element, `<html>`, telling the browser that everything between the opening tag and the closing tag is your web page. The area inside the HTML element is divided into two sections: the head section and the body section.

The head section contains information for the web browser for the proper display of your web page. Earlier in this book, I briefly touched on doctype and charset. These are a few of the hidden, or head section, elements that will affect how your page will render and rank in search engines, something most people are concerned about.

In this chapter, I will focus on writing the HTML of your page, including the required elements, and on structuring the content in the body of your web page, separating content from presentation. This will help your pages work well in a variety of devices, including modern browsers such as Firefox 2, Internet Explorer 7, Opera 9, and Safari; Pocket PCs; television-based browsers such as MSNTV, Kox, PlayStation, and so on; and cell phone browsers. At the same time, using Cascading Style Sheets (CSS) for presentation will make it easier for your website to maintain a consistent look and feel.

Required Elements

Every web page must have at minimum three elements before your browser will be able to render a page:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```

    <head>
    </head>
    <body>
    </body>
</html>

```

A document type declaration, a.k.a. the doctype, tells the browser what version of HTML it should use to interpret the HTML on the page. By default, Expression Web adds the HTML 1.0 Transitional doctype to every web page you create. Unless you are working with existing or legacy pages, this is the doctype I recommend. If you are working with existing web pages created with an earlier version of HTML, it is your choice whether to convert them to HTML 1.0 Transitional or leave them with their current doctype. If there is no current doctype or if the doctype is HTML 3.2 or earlier, I recommend you convert them as soon as possible to HTML 1.0 Transitional. So, the recommended doctypes are as follows:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

```

The `<html>` element immediately follows your doctype declaration. If you do not have an HTML element following your doctype, the browser will display your HTML code as plain text. Although you can include some attributes in your `<html>` element, they are not required.

The `<head>` element encloses the section of your web page that provides instructions to the browser on how to display the contents of the `<body>` element as well as information for search engines, scripting, page-level styles, and the title that will display at the top of your visitor's browser.

The `<body>` element contains the page's text. Only content within the opening and closing `<body>` element will be displayed in your visitor's web browser.

Structuring Your Web Page

Web pages are written using HTML. This means a web page is nothing more than structured text with images that in some cases incorporate sound or movies. When you are creating a web page, think of an outline. That is the ideal structure of a web page. Your page should look like the outlines you remember from your school days when you view them with images and styles turned off. This does not mean your page must look like an outline when viewed in a web browser. The page structure gives a "hook" to put visual styles on, as you'll do before you're done with our site.

When you add text in the design window, Expression Web defaults to paragraphs for your typed text. What you'll see in the design section as you type in the code section of Split view is the same text that's enclosed in paragraph tags, `<p></p>`. In the design section, the text is out-lined with a *p* in a tab near the top-left side, as shown in Figure 4-1.



Figure 4-1. The HTML element for each block will be shown by default when your cursor is inside the element.

This is a good choice since most of the text on your page should be body text, but it doesn't add much structure to your page. When you think of an outline or any other document, there is always visible title, body text, and/or list items with subtitles. Your web page also should have a title the visitor sees and recognizes as a page title, usually by making it larger and more prominent. The title should tell the visitor something about your site. This means giving the title structure with what is called *header markup*, as in an outline where you have headers of various levels and body text. By using headers on your page effectively, the visitor can quickly see that the page contains content worth reading. In an attempt to make a section of text recognizable as a title, many old-school web designers would use presentational elements in their HTML. This is why you sometimes see markup that looks like this:

```
<p><font size="6"><b>Structuring Your Page Exercise</b></font></p>
```

instead of seeing the simple heading 1 element that is part of HTML:

```
<h1>Structuring Your Page Exercise</h1>
```

These are visually equivalent, as you can see in Figure 4-2, but the one using `<h1>` provides valuable information about the structure of your web page and about what you think are the important points to search engines and those visitors who are using assistive devices to read the page to them. The other one uses font size and bold, which provide visual clues only for those viewing the web page in a browser that supports the `` element. Search engines and others who cannot see the visual styles will not recognize the line as a heading.

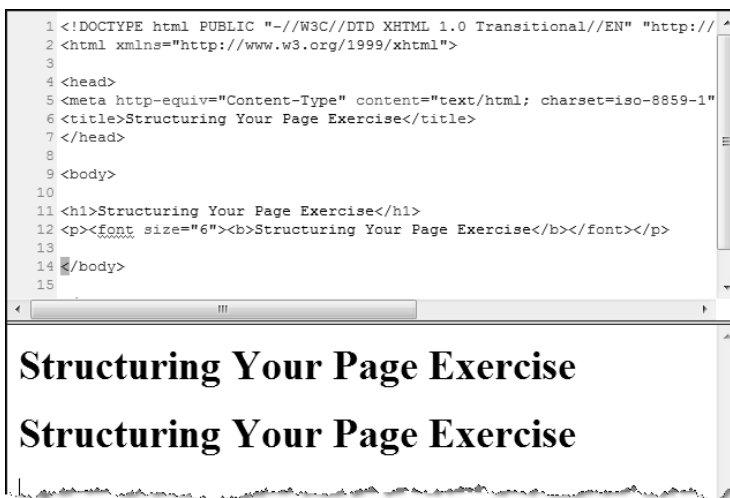


Figure 4-2. By default, HTML heading elements will be bold, and the size will be proportionate to the heading level.

Note Did you notice the jagged line/squiggle under the font element in Figure 4-2? Expression Web is alerting you to invalid HTML since the `` element is deprecated and is not valid under the doctype being used.

Make headers relevant to content that follows, and visitors will want to stay and read the page to learn more. Search engines also appreciate a structured site with appropriate headers and other semantic markup. Each of your pages should have at least two levels of headers. Most web pages have at least three. Fortunately, Expression Web makes creating the structural markup easy.

The best way to understand how to use structural markup is to create a structured document, as you will do in Exercise 4-1.

Exercise 4-1. Creating Headers

In this exercise, you'll create some headers for your website.

1. Begin by creating a new blank web page with the following code. Expression Web creates the `<head>` section in the following code by default if you are using the recommended settings, which means you can type the content directly into the design area if you choose and the paragraphs should be created for you automatically. Make sure you include the commas in the price list.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  <title>Structuring Your Page Exercise</title>
</head>
<body>
  <p>Structuring Your Page Exercise</p>
  <p>In this exercise we will be adding page structure. On a real page
    most of the text would be in paragraph elements with a list or two
    and possibly a data table. In this exercise we will be using the
    different ways of structuring your page.</p>
  <p>Begin with Headers</p>
  <p>While a live page would have design elements such as mastheads,
    logos, and menus, our exercise page will not be using design elements,
    sticking with the basics.</p>
  <p>Lists as Menus</p>
  <p>Most standards-based web design uses an unordered list as the
    markup for navigational elements since semantically your site
    navigation is basically a list of pages within your site.</p>
  <p>page 1</p>
```

```

<p>page 2</p>
<p>page 3</p>
<p>page 4</p>
<p>page 5</p>
<p>Tables</p>
<p>Should be limited to tabular data such as event information or
  a price list.</p>
<p>item no., description, price each, per dozen</p>
<p>1,red balloon, $0.25,$2.25</p>
<p>2,blue balloon,$0.25,$2.25</p>
<p>3,white balloon,$0.25,$2.25</p>
<p>4,silver mylar balloon,$1.75,$19.00</p>
<p>5,gold mylar balloon,$1.75, $19.00</p>
</body>
</html>

```

2. Before you go any further, save your file under the name `structural.html`.
3. Next, click anywhere in the “Structuring Your Page Exercise” paragraph. Then select the `<p>` element from the Quick Tag bar, as shown in Figure 4-3.



Figure 4-3. Select the element you plan to change.

4. Then expand the Styles drop-down list on the Format toolbar. Select Heading 1 `<h1>`, as shown in Figure 4-4.

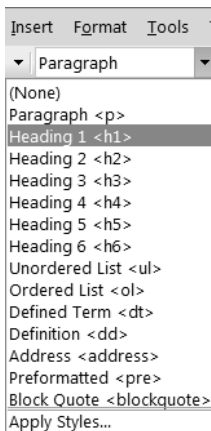


Figure 4-4. Choose the desired HTML element.

Your page will now have an `<h1>` element. As you can see in Figure 4-5, the `<h1>` is larger, is bold, and has additional margin between it and the elements that follow.

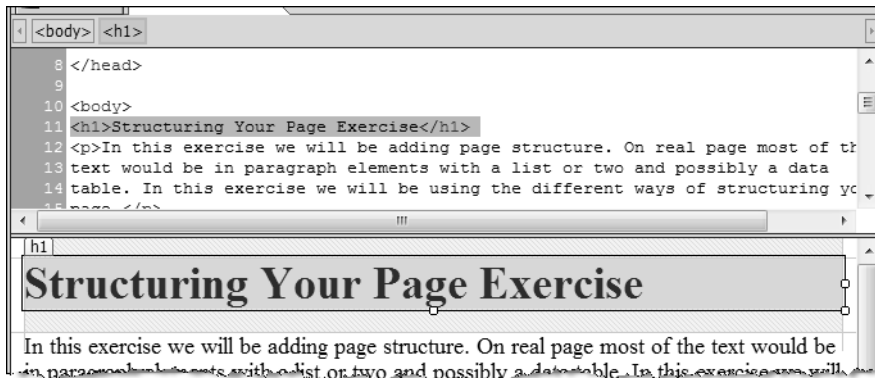


Figure 4-5. You can easily see the default presentation differences between `<h1>` and `<p>` elements.

5. Select the “Begin with Headers” text, and change the paragraph to an `<h2>`.
 6. Repeat with the “Lists as Menus” paragraph.
-

Lists

Lists are one of the most commonly used elements on the Web after paragraphs and headers. Most visitors skim web pages for highlights, bullets or other list elements, and the whitespace around them, which draws your visitor’s attention to lists on your page. Once the visitor finds an indication that the page contains information they want, then the visitor will actually read the page or at least the section that caught their attention.

Three types of lists are available in HTML: unordered, ordered, and definition.

Ordered Lists

By default ordered lists start with 1 and increment with each item in the list. Typically, ordered lists provide a step-by-step set of instructions, but you can use them for any content that has a hierarchical order. You can change the number format via CSS.

Unordered Lists

Unordered lists are one of the most useful HTML elements, allowing you to group sections of content together in a way that lets your visitor know there is a relationship between the items even when there is no style applied. By default, browsers will use a bullet for each item in a list, but you can change this using CSS. For this reason, unordered lists are the structure of choice for menus, which brings you to the second structural exercise (see Exercise 4-2).

Exercise 4-2. Creating a List

In the `structural.html` page created in the preceding exercise, you have a series of paragraphs starting with page 1 and ending with page 5.

1. In the design area, select all the paragraphs with “page x” on them, as shown in Figure 4-6.
2. Then from the Styles drop-down list, select Unordered List .

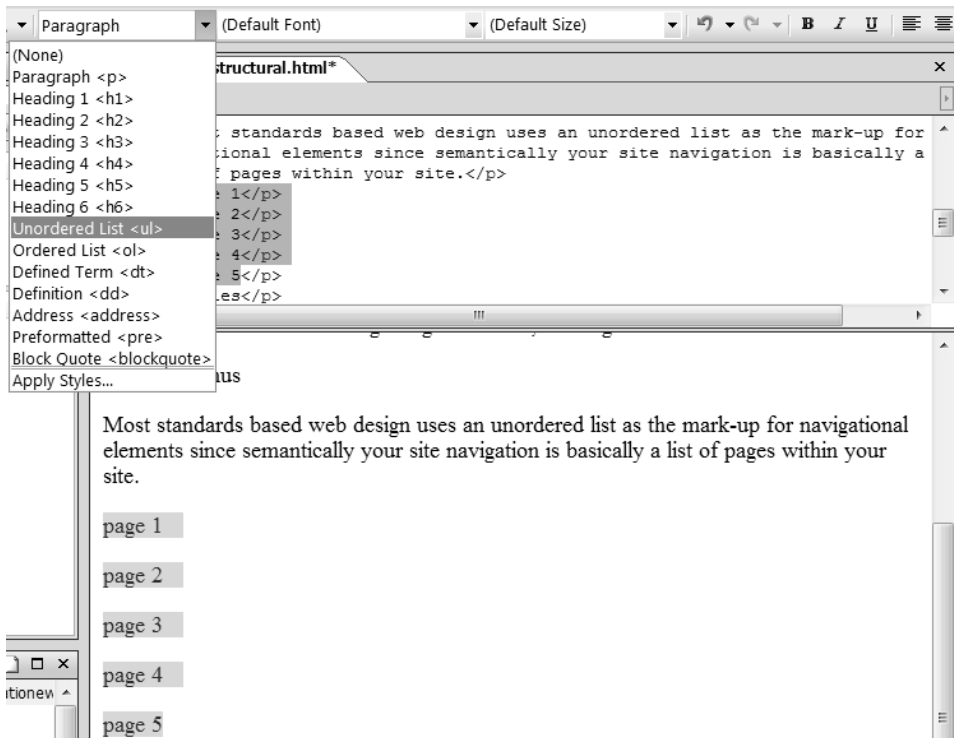


Figure 4-6. Highlight all the paragraphs you want to convert to list items before you convert to an unordered list.

3. Now when you check the code area, you will see the following code:

```
<ul>
  <li>page 1</li>
  <li>page 2</li>
  <li>page 3</li>
  <li>page 4</li>
  <li>page 5</li>
</ul>
```

Each of the paragraphs that you highlighted have turned into list items surrounded by a to tell the browser this is an unordered list.

Note Both ordered lists and unordered lists use list items, ``, to identify each of the elements in the list. Your list will display with bullets or numbers, as determined by the type of list element surrounding the list item group.

So far in this exercise you have converted existing text to a list. Next, you will add another item to the list.

4. Begin by placing your cursor immediately after the last character in the list, “page 5,” as shown in Figure 4-7.

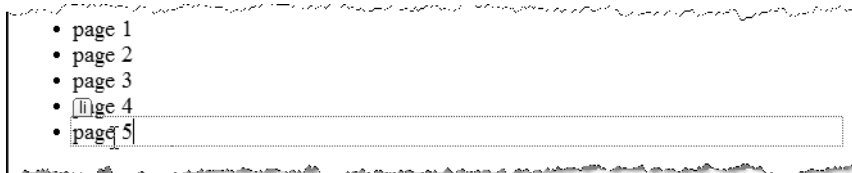


Figure 4-7. To add a new list entry, put your cursor at the end of the list item to which you want to add a new list item.

5. To finish adding your new list item, simply press the Enter key and start typing **page 6**. You will see that `` `` has been added with your cursor between them, as shown in Figure 4-8.

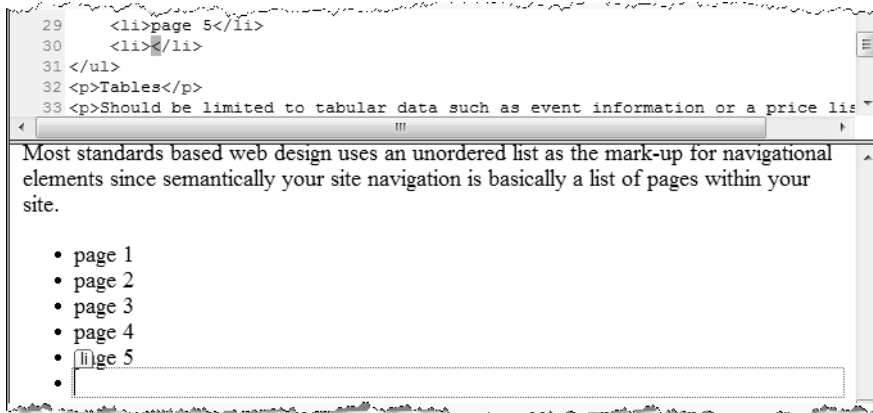


Figure 4-8. When a new list item is inserted by using the Enter key, you are ready to enter your text immediately.

The procedure is the same whether you are using an unordered or ordered list. The only difference is which you choose from the Styles drop-down list.

Definition Lists

Definition lists are an underused HTML element. Unlike the other two types of list, each entry in a definition list has two parts. The two parts are the term and the definition, but they are not limited to just dictionary-type definitions because they may also be used for any terms that have a direct relationship to each other. For example, you can represent session information at an event with a definition list:

```
<dl>
  <dt>Session 1 Topic</dt>
  <dd>Speaker 1</dd>
  <dd>Speaker 2</dd>
  <dd>Speaker 3</dd>
  <dt>Panel Discussion</dt>
  <dd>Panelist</dd>
  <dd>Panelist</dd>
  <dd>Panelist</dd>
</dl>
```

You can create definition lists the same way that you create ordered and unordered lists.

Caution On occasion, Expression Web will not replace paragraphs with a `<dt>` or `<dd>` but instead will wrap the `<p>` with the definition item. If this occurs, press Ctrl+Z to undo your changes, and then apply the correct element to each paragraph.

Tables

When HTML was created, the primary purpose was to share information between educational institutions, governments, and the military. Data tables present information in a grid to show the relationships between the data. Each data table should have a column and/or row headers. Captions are optional but recommended.

In the early days of the World Wide Web, most web designers came from print and were frustrated by the lack of presentational elements in HTML. These designers were used to working in a grid- or column-based layout for paper and saw data table markup as a way to re-create that working environment for web pages. This led to very complex code or “slice-and-dice” page design where the page would be created in a graphics program such as Adobe Photoshop (the industry standard for print designers) and then sliced into bits and pieces that were assembled back together with tables using merged and split data cells and nested tables. As mentioned in the introduction, this table-based method became unworkable, and CSS was introduced to separate content from presentation. As a result, the abuse of tables has gone out of fashion, so you should use tables for data. In the `structural.html` page, you have a section of comma-separated pricing information that you will convert to a table in Exercise 4-3.

Exercise 4-3. Creating Tables

In this exercise, you'll create a table from comma-separated values.

1. Open your `structural.html` web page, and select the section starting with “item no.” and ending with the last “\$19.00,” as shown in Figure 4-9.

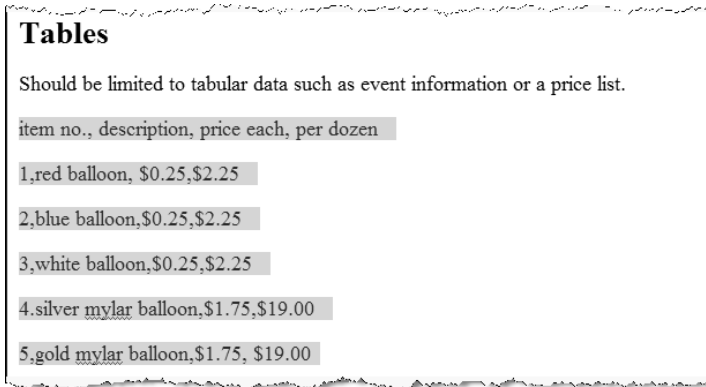


Figure 4-9. Select all the comma-separated items.

2. To create a table from these paragraphs, select **Tables** ► **Convert** ► **Text to Table**, as shown in Figure 4-10.

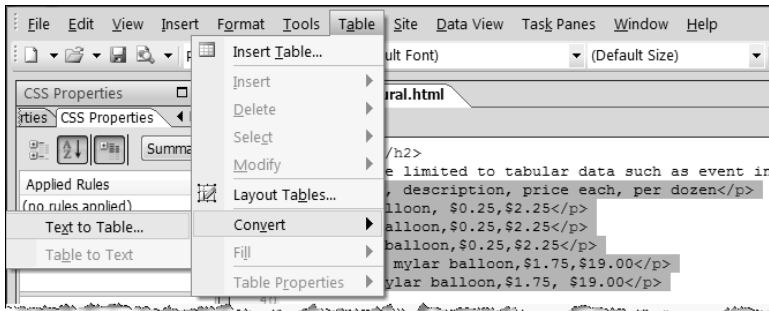


Figure 4-10. Converting comma-separated text into a table gives you clean table code.

3. This will launch the Convert Text to Table dialog box. Make sure you select Commas as the separator type, as shown in Figure 4-11.

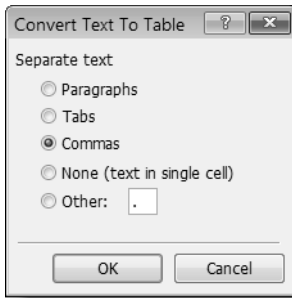


Figure 4-11. *The default setting is Paragraphs, but tables are more commonly exported from Microsoft Excel or a database in a comma-separated format.*

This will turn each comma-separated section into a data cell. The code created looks like this:

```
<table style="width: 100%">
  <tr>
    <td>item no.</td>
    <td>description</td>
    <td>price each</td>
    <td>per dozen</td>
  </tr>
  <tr>
    <td>1</td>
    <td>red balloon</td>
    <td>$0.25</td>
    <td>$2.25</td>
  </tr>
  <tr>
    <td>2</td>
    <td>blue balloon</td>
    <td>$0.25</td>
    <td>$2.25</td>
  </tr>
  <tr>
    <td>3</td>
    <td>white balloon</td>
    <td>$0.25</td>
    <td>$2.25</td>
  </tr>
  <tr>
    <td>4</td>
    <td>silver mylar balloon</td>
    <td>$1.75</td>
    <td>$19.00</td>
  </tr>
</table>
```

```

    <td>5</td>
    <td>gold mylar balloon</td>
    <td>$1.75</td>
    <td>$19.00</td>
  </tr>
</table>

```

However, this does not create table headers. In the example, the first row contains the column headers.

4. To turn a data cell into a header cell, right-click the cell, and choose Cell Properties from the pop-up menu, as shown in Figure 4-12. Start with the `<td>item no.</td>` cell.

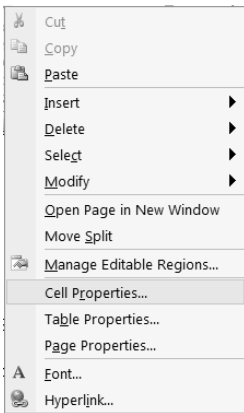


Figure 4-12. *Cell Properties* allows you to change a cell type from data to a header.

5. This will launch the Cell Properties dialog box where you can change the cell from a data to a header cell by selecting the Header cell box, as shown in Figure 4-13. The other options are for creating CSS styles.

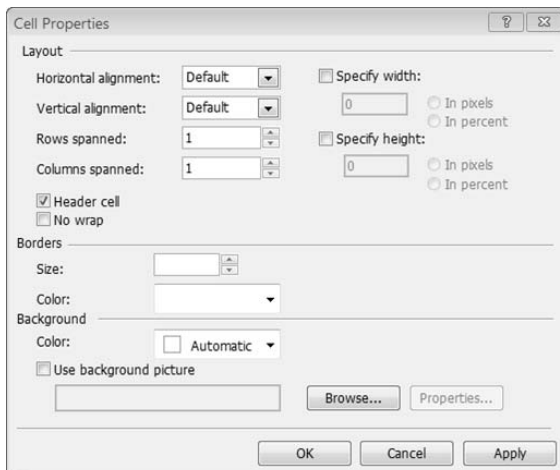


Figure 4-13. *The options here will be applied using CSS with the exception of header cells.*

6. Repeat for each of the other data cells that need to be converted to header cells.

By default web browsers will make table headers bold and centered. When you have finished your table, the page should look like Figure 4-14 when viewed in a web browser:

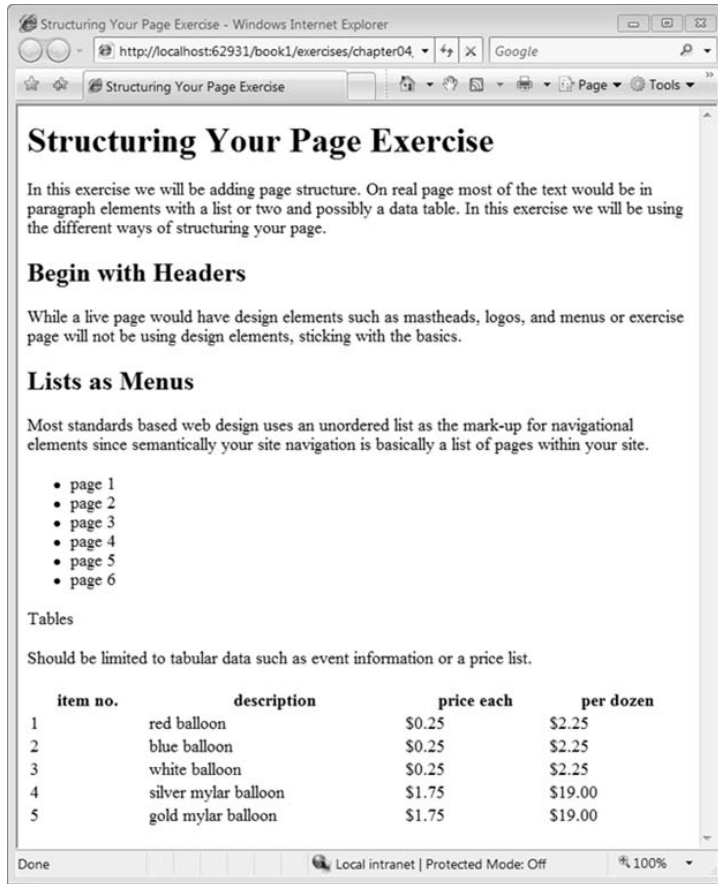


Figure 4-14. Structured web page viewed in Windows Internet Explorer 7

When you view any properly structured web page without styles applied, you should see headings, lists, and tables that look like this.

Other HTML Elements from the Drop-Down List

Figure 4-4 shows three other HTML element options:

- `<pre>`: The `<pre>` element, short for *preserve format*, displays the text exactly as it is written in Code view, including all spaces and line breaks. Although this is frequently used for code, better practice is to use the `<code>` element with a style definition set to preserve space.

Note To use the `<code>` element, you must add it in Code view and define the element using the New Styles methods discussed in Chapter 6.

```
<pre>
This would
be displayed just
as you see
it
here. Which can make it ideal for poetry.
</pre>
```

- `<address>`: The `<address>` element is underutilized. You should use this W3C element to supply contact information about the web page and/or website.

```
<address>
Apress, Inc.<br />
2560 Ninth St., Ste. 219<br />
Berkeley, CA 94710<br />
Phone: (510) 549-5930<br />
Fax: (510) 549-5939
</address>
```

- `<blockquote>`: Like tables, the `<blockquote>` element is frequently abused for presentation since by default most web browsers will indent the contents of this element. If all you want to do is create an indented block of text, use padding or margin in a style class.

```
<blockquote>
<p>Founded in 1998, The Web Standards Project (WaSP)
fights for standards that reduce the cost and complexity of
development while increasing the accessibility and long-term viability
of any site published on the Web. We work with browser companies,
authoring tool makers, and our peers to deliver the true power of
standards to this medium.<br />
<a href=http://www.webstandards.org/about/>
http://www.webstandards.org/about/</a></p>
</blockquote>
```

Note In HTML your text will wrap automatically when it reaches the end of the space available in the browser. Sometimes you may want to start a new line without starting a new paragraph or other element as in the previous address example. The HTML element for a line break is a `
`. To insert a line break in your code in Design view, press Ctrl+Enter.

Hyperlinks

The Internet would not work without hyperlinks. Hyperlinks enable you to move from one page to another inside a website or to link to external pages on the Web. Hyperlinks also are used to link to non-HTML documents such as Adobe PDF files. Other types include links to CSS to format web-page display and to scripts, which provide added functionality.

Hyperlinks written in HTML consist of an anchor element and an href attribute. If you type a URL in Design view, Expression Web will convert the URL to a hyperlink as soon as you add a space after the URL. Although this works well, you will frequently want to create links with text other than the URL of the resource. In Exercise 4-4, you will take the first item in the unordered list and turn it into a hyperlink.

Exercise 4-4. Creating a Hyperlink

In this exercise, you'll use the Expression Web to create hyperlinks.

1. With your `structural.html` page open, highlight the first list item, and right-click. From the resulting menu, select **Hyperlink**, as shown in Figure 4-15.

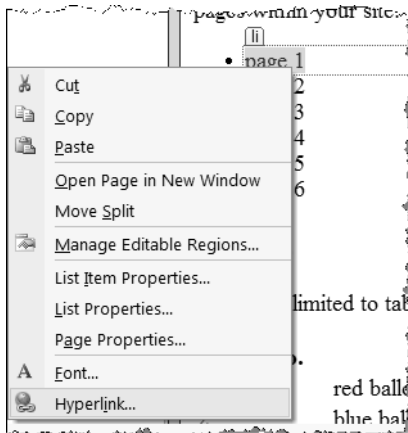


Figure 4-15. The fastest way to create a hyperlink is to use the right-click menu.

The Insert Hyperlink dialog box opens, as shown in Figure 4-16. The text you selected in Design view will be in the “Text to display” field; if you want to change it, you can simply type over it.

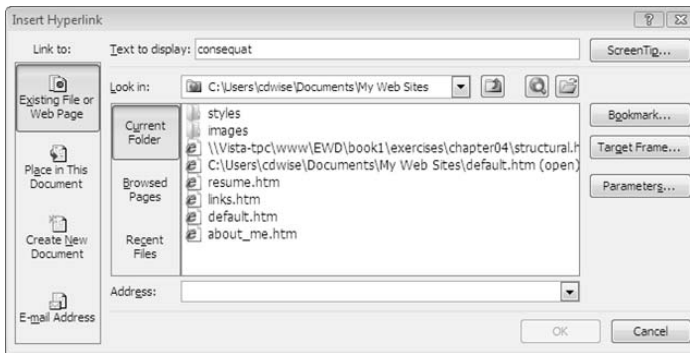


Figure 4-16. You can find many options for creating hyperlinks in the Insert Hyperlink dialog box.

2. For this exercise, select the existing file or web page from the “Link to” column and the current folder in the “Look in” column.
3. With these two options selected, your open website should be in the location block. Select the `index.html` page you created earlier.
4. Then click the OK button. Your code should now read `page 1`, and `page 1` should be blue and underlined.

Note As shown in Figure 4-16, you have many options for creating hyperlinks. You can launch the Insert Hyperlink dialog box at any time by right-clicking in Design view, selecting **Insert** ► **Hyperlink**, and then using the **Create New Document** option to add new pages to your site, as shown in Figure 4-17.

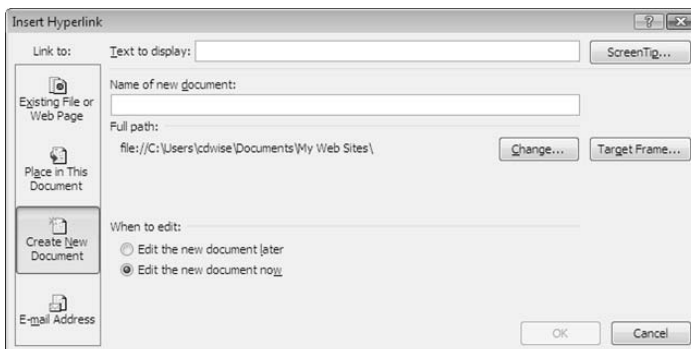


Figure 4-17. You can use Insert Hyperlink to create new pages in your site as you create your links.

One feature that is particularly useful is the ability to browse recently viewed web pages, as you see in Figure 4-18. When you select a page you have previously visited in your web browser, the page's HTML `<title>` element will appear in the “Text to display” box and the URL appears in the Address box.

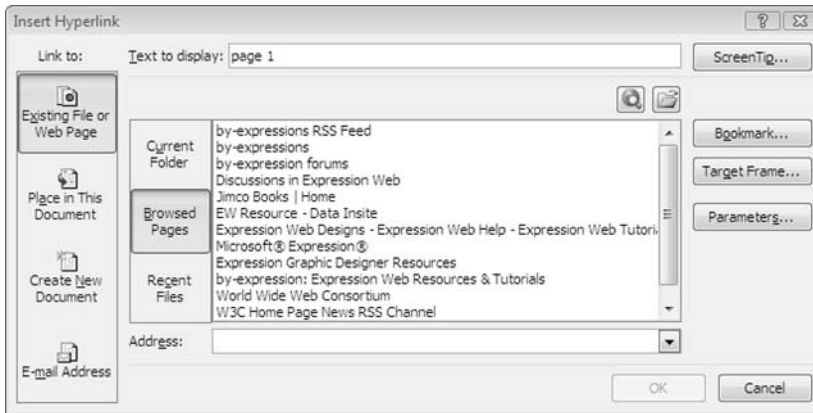


Figure 4-18. Visit the sites you want to link to, and you can avoid typing errors by selecting from the *Browsed Pages* list.

You can link to a place in this document, called a *bookmark* by Expression Web or an *anchor* by the W3C and most web designers. The most common type of bookmark is a back-to-top link. You'll create one in Exercise 4-5.

Exercise 4-5. Creating a Back-to-Top Bookmark

In this exercise, you'll create a back-to-top bookmark.

1. To add a back-to-top bookmark, place the cursor next to the first word on the page, and from the Insert menu, select **Bookmark** (or press **Ctrl+G**), as shown in Figure 4-19.

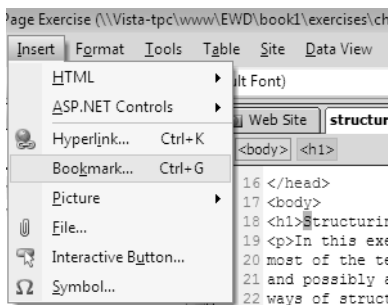


Figure 4-19. Launch the *Bookmark* dialog box.

2. Type **top** in the “Bookmark name” box in the Bookmark dialog box, as shown in Figure 4-20, and then click OK.

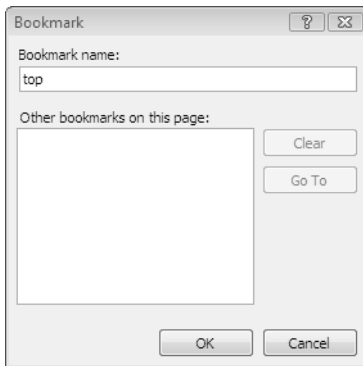


Figure 4-20. *Keep your bookmark names lowercase with no spaces.*

3. Next, go to the bottom of your `structural.html` page, and right-click anywhere in Design view after the table.
4. Select Hyperlink.
5. Now when you use the Place in This Document selection from the “Link to” column, you will see a list of any bookmarks on the current page.
6. When you select the `top` bookmark, it will put the same name in the “Text to display” box.

In Figure 4-21 this has been changed to back to `top` since your link should be more descriptive than the short name used for your anchor.

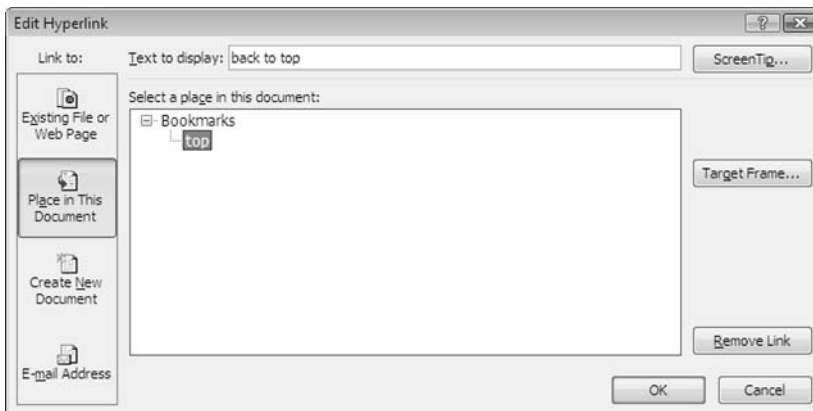


Figure 4-21. *Use the Screen Tips button to add a tooltip using the title attribute.*

The code you created in this exercise for the bookmark is as follows:

```
<h1><a name="top"></a>Structuring Your Page Exercise</h1>
```

And the code for the link created is as follows:

```
<p><a href="#top">back to top</a></p>
```

7. Finally, make sure you that you save and preview your web page in your browser. Make sure you try the links you created, including the back-to-top bookmark. Make sure your browser window is short enough that you have a vertical scroll bar, or you may not be able to see the effect of using your back-to-top link.

You can also create a hyperlink that will launch the visitor's default e-mail program with an e-mail address already filled in by using the E-mail Address option in the "Link to" column.

Note Using an e-mail hyperlink will dramatically increase the amount of spam you receive. It is better to use a contact form or to put your e-mail address in without a link using a format that does not include the @ symbol.

Other HTML elements perform specific functions such as the form elements. You'll use them in later chapters.

The Importance of Hidden Elements

Now you'll turn to those elements in the <head> section of the page that the visitor will not see and learn how these elements affect your web page.

The <title> element will tell the browser what to display at the top of the visitor's browser. This is important since many people are using tabbed browsers or have more than one browser instance open at a time. Without the page title, it can be difficult to determine which tab or window contains your web page. In addition, when you are adding a hyperlink to a document you have viewed, Expression Web will use the page title as the default link. This makes inserting hyperlinks to resources and referencing other pages much easier.

Meta-elements are elements that let you specify information about your document for search engines and other indexing information as well as specify page information for display. Browser instructions called http-equiv include the following:

- *Character set*: This tells the browser what type of characters to expect. In most cases, you'll be using Latin characters, but there are other character sets that you could use, including Unicode or language-specific sets such as Greek, Chinese, or other languages that are not based on Latin.

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

- *Language*: This tells the browser, search engines, and screen readers what language the page is written in.

```
<meta http-equiv="Content-Language" content="en-us" />
```

Common search engine–related meta-tags are as follows:

- *Description*: In some search engines, the text you put in the description meta-element is displayed when your page is returned in search results. Other search engines such as Google will display either the first lines of content or the section relevant to the search results.

```
<meta name="description" content="content to be displayed"/>
```

- *Keywords*: This is another meta-element used by some search engines to categorize or use in matching search terms. This is of much less importance now than it was in the early days when search engines paid more attention to meta-tags than they do now.

```
<meta name="keywords" content="a list of keywords" />
```

All the major search engines have a section for webmasters that provides information on how to improve your website's ranking. If you are interested in search engine optimization (SEO), I recommend starting with the following:

- *Google*: <http://www.google.com/webmasters/>
- *MSN Live Search*: <http://search.msn.com> (use the “for site owners” link at the bottom of the page)
- *Yahoo!*: <http://help.yahoo.com/help/us/ysearch/webmaster/webmaster-01.html>
- *General SEO information at Search Engine Watch*: <http://searchenginewatch.com>

You can easily set each of the previous meta-elements in Expression Web using the General tab of the Page Properties dialog box. You can set less commonly used meta-elements by using the Custom tab. The ones listed here are all content-type information for search engines:

- *Robot*: Tells search engines which pages to index or which links to follow and which pages and links to ignore
- *Author*: Gives the name of the person or company that created the web page
- *Copyright*: Provides the name of the copyright holder and the date of creation

In Exercise 4-6, you'll add meta-elements to your page.

Exercise 4-6. Adding Meta-elements Using the Page Properties Dialog Box

In this exercise, you'll add meta-elements to your web page to provide additional information for search engines and browsers.

1. Right-click anywhere in Design view, and the right-click menu with Page Properties will appear, as you saw in Figure 4-15. Select this option to launch the Page Properties dialog box, as shown in Figure 4-22.

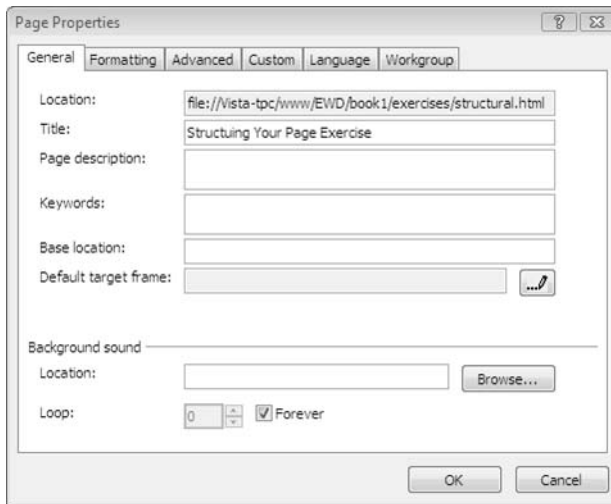


Figure 4-22. The General tab is for entering your page title, keywords, and description.

Since your page already has a title, Expression Web has placed it in the Title box ready for you to edit if you choose.

2. Add text to describe your page in the “Page description” text box. I will be using “An example of a well-structured semantic HTML page.” If you choose to use a description on your page, make sure it is short, is accurate, and makes sense when read. You should keep it to less than 200 characters and realize that some search engines may display as few as 50 characters.

You should separate keywords with commas and keep them to ten words or less. They must also appear in the content of your web pages if you want the search engines to pay attention to your search keyword suggestions. Search engines are constantly revising their search algorithms, and the best source for how to optimize your web pages for search engines is the search engines themselves.

3. Next, move to the Language tab, and select the language your page is written in from the “Mark current document as” drop-down list, as shown in Figure 4-23.

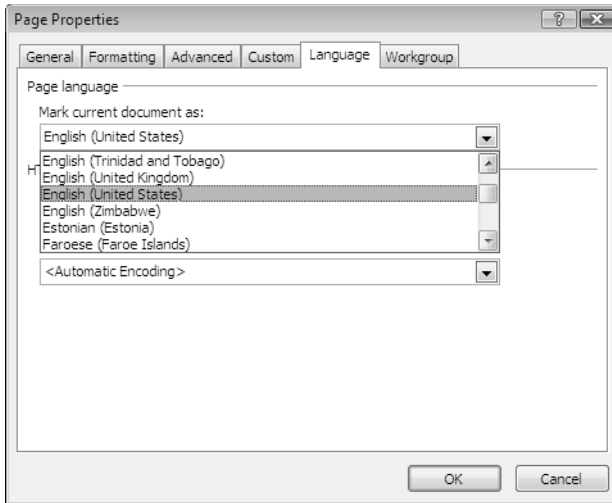


Figure 4-23. Your HTML encoding should show your site default.

- Moving to the Custom tab, shown in Figure 4-24, click the Add button in the “User variables” half of this tab. Expression Web calls the optional meta-attributes that provide additional information to search engines and other web crawler user variables.

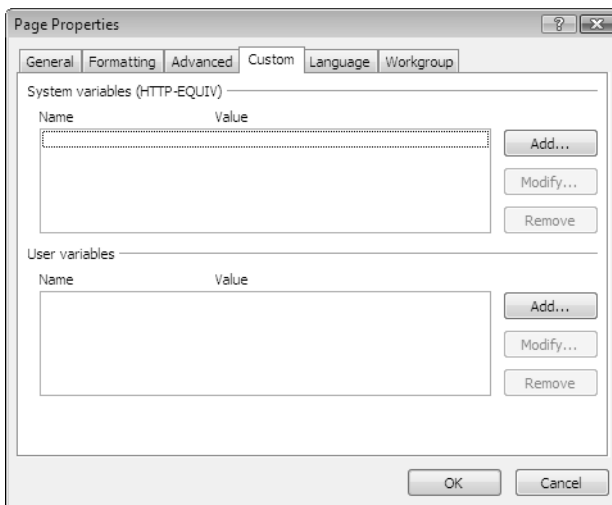


Figure 4-24. Launch the box to assign values to the meta-attributes by clicking the Add button.

- Type **copyright** in the Name text box and the year followed by your name in the Value text box, as shown in Figure 4-25.

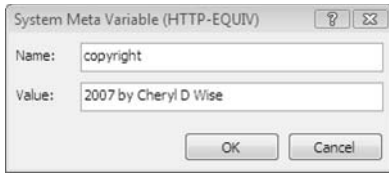


Figure 4-25. You can add almost any name and value here to provide additional information.

The <head> section of your page should now look like this:

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta http-equiv="Content-Language" content="en-us" />
<title>Structuring Your Page Exercise</title>
<meta name="description" content="An example of a well structured
semantic HTML page" />
<meta name="keywords" content="structured html" />
<meta name="copyright" content="2007 by Cheryl D Wise" />
</head>
```

The Advanced and Formatting tabs create CSS and will be used in Chapters 5, 6, and 7 when I cover CSS.

The <link> element defines a relationship between the document and another page. This element is commonly used to link external stylesheet(s) for document presentation. Other link relationships that can be used are as follows:

- *Next*:
- *Back*:
- *Start*:
- *Alternate language page*: <link rel="alternate" lang="fr" xml:lang="fr" title="Bases de L'Expression Web" type="text/html" hreflang="fr" href="http://foundationsewd.com/fr/index.html">

Tip Start is most useful at the beginning of a slide show or other multipage presentation since it tells search engines which page in a series is the first page.

The <style> element defines the area containing page-level styles. You'll add styles to your pages in Chapters 5, 6, and 7, so I won't cover this element now.

<script> <noscript> are usually used together. The <script> element tells the browser that the next section of code is a client-side script for the browser to execute and specifies the

language of the script. Any scripts that you put inside a `<script>` element for execution on the browser should use JavaScript. The code would look like this:

```
<script type="text/javascript">
...some JavaScript...
</script>
```

A `<noscript>` element displays alternative information in the visitor's browser at the location the script output would usually show. Please do not use this:

```
<noscript>
This page requires you to use JavaScript.
Please enable JavaScript support and reload the page.
</noscript>
```

Most people who do not have JavaScript turned on are either on devices where JavaScript is not supported or are under corporate lockdown policies where they cannot enable JavaScript. Instead, use something meaningful such as this:

```
<noscript>
Our site navigation requires JavaScript. Your browser
is not able to display JavaScript, please use our
<a href="sitemap.html">site map</a> to navigate our site.
</noscript>
```

The added benefit of offering alternative navigation is that search engines can follow links in the `<noscript>` element that they may not be able to follow when JavaScript is required.

Summary

In this chapter, you learned how a structured HTML web page will give you a good foundation on which to hang your site's presentation elements to build a site's look, identity, or brand and help you with search engine placement, file size reduction, and other things. Over the next two chapters, you will turn your attention to something you've already looked at briefly, CSS, and use it to create the presentation layer that specifies your site's look and feel.