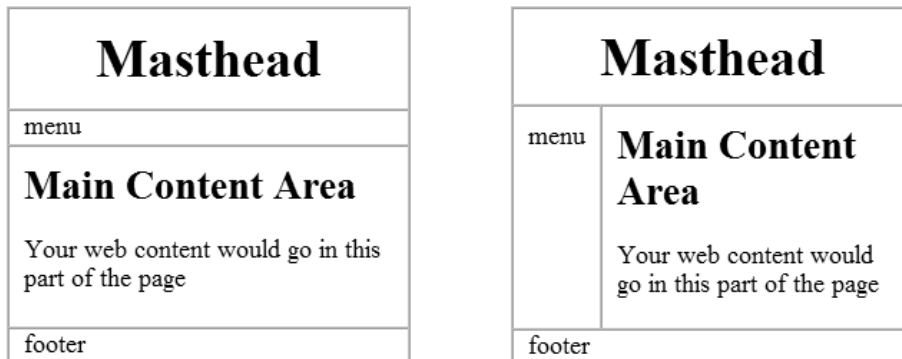




# CSS Positioning Using Expression Web Style Tools

**Y**ou can create a web page layout in two ways. The old style is to use a table and nest other tables inside. In the 1990s, that was the most effective way to create a page layout that would work in Netscape 4.0 and Internet Explorer 4. As with `<font>` tags for presentation, tables for layout are out of fashion and create bloated web pages that do not work well in alternative web devices ranging from cell phones to PDAs to television sets used as monitors for game consoles to WebTV-type browsers. Browsers have matured, and all current or even semicurrent web browsers support CSS 2.1 for page layout, which degrades gracefully to such devices; as a result, CSS is the preferred method to use for your web page layout.

The majority of websites on the Internet use one of two basic page layouts. The first type is a single column with or without a masthead and footer. This is the type of page layout that is created by default. One common modification to a single-column page layout is to constrain the width to fit inside of an 800-pixel-wide browser. The second type has a masthead and two or more columns, usually with the menu to the left of the main content section. Figure 7-1 shows the two most common layouts.



**Figure 7-1.** *Single-column and two-column layout*

In this chapter, you will be creating a multicolumn page layout using different methods of placing your content on the page; you'll do this by using the last four categories in the

New/Modify Style dialog box. Although you can create simple single-column page designs using just HTML elements to provide structure for your pages, you will need to use one or more <div> elements as containers to divide the page into sections.

## Creating the Initial Page

You will continue with the structural.html page you were using in Chapter 6 with the picture frame borders. Compare your border2.html page after the last Chapter 6 exercise to the following code, and then save it as chapter7.html:

```
<!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" />
  <meta http-equiv="Content-Language" content="en-us" />
  <title>Chapter 7</title>
  <style type="text/css">
h1 {
  border-style: outset inset inset outset;
  border-width: thick;
  border-color: #22292D #999999 #999999 #22292D;
  font-size: 1.8em;
  text-align: center;
  background-image: url('images/h1-bg.png');
  padding: 1em;
}
body {
  font-family: Tahoma, Helvetica, Arial, sans-serif;
  background-image: url('../chapter06/images/crystal-bg.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: center center;
  background-color: #C6C7C3;
}
</style>
</head>
<body>
  <h1><a name="top"></a>Structuring Your Page Exercise</h1>
  <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>
  <h2>Begin with Headers</h2>
  <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>
```

```
<h2>Lists as Menus</h2>
```

```
<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>
```

```
<ul>
  <li>page 1</li>
  <li>page 2</li>
  <li>page 3</li>
  <li>page 4</li>
  <li>page 5</li>
  <li>page 6</li>
```

```
</ul>
```

```
<p>Tables</p>
```

```
<p>Should be limited to tabular data such as event information or a price list.</p>
```

```
<table style="width: 100%">
  <tr>
    <th>item no.</th>
    <th>description</th>
    <th>price each</th>
    <th>per dozen</th>
  </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>
</tr>
```

```

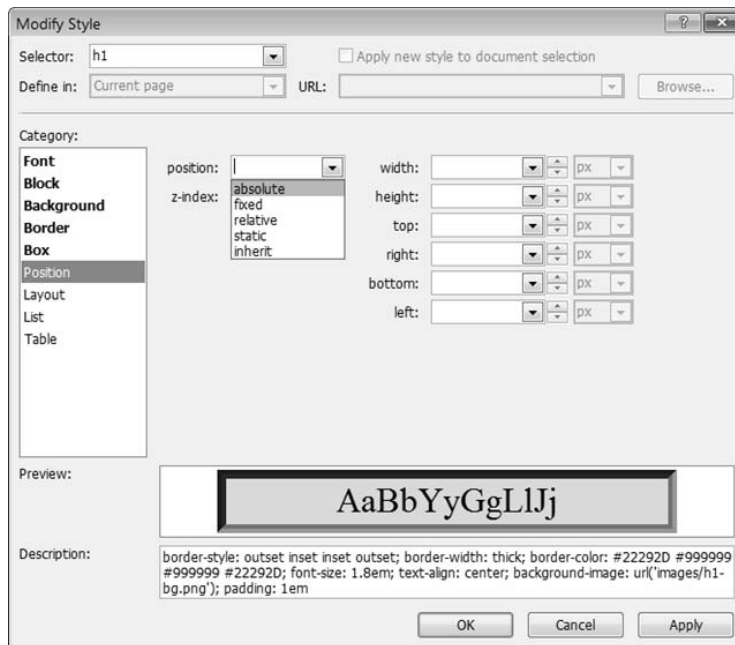
        <td>5</td>
        <td>gold mylar balloon</td>
        <td>$1.75</td>
        <td>$19.00</td>
    </tr>
</table>
<p><a href="#top">back to top</a></p>
</body>
</html>

```

You can also download this page from [http://foundationsofexpressionweb.com/exercises/chapter\\_7](http://foundationsofexpressionweb.com/exercises/chapter_7) along with the completed exercise files for this chapter for reference.

## Working with the Positioning Category

You will begin with the Position category's options of absolute, relative, and fixed, as shown in Figure 7-2.



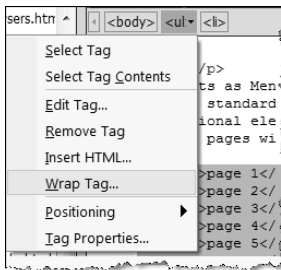
**Figure 7-2.** Only two of the position elements are in common use.

The best way to learn how you can use positioning is to see it in action. In Exercise 7-1, you will create a two-column layout below your `<h1>` using absolute positioning.

## Exercise 7-1. Positioning for Layout

With the `chapter7.html` page open, you will begin by creating a container for the unordered list that will become your site menu. Select the unordered list of pages.

1. From the Quick Tag bar, right-click the `<ul>` element, and from the drop-down list, select `Wrap Tag`, as shown in Figure 7-3.



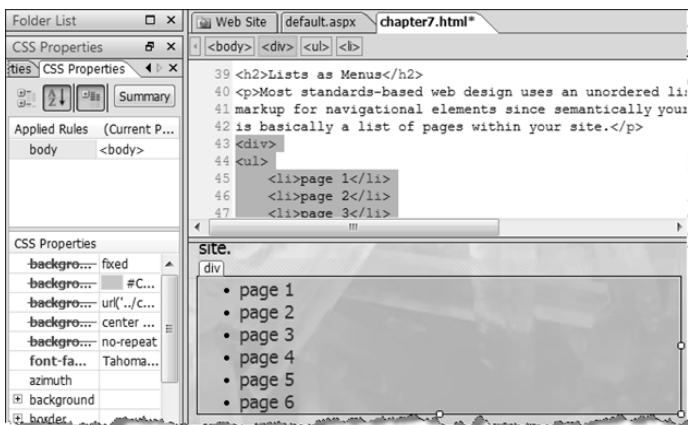
**Figure 7-3.** Using the Quick Tag bar is the fastest way to wrap a `<div>` around existing content.

2. In the Quick Tag bar, type `<div>` between the opening and closing angle brackets, as shown in Figure 7-4. Use the green check mark to save your changes.



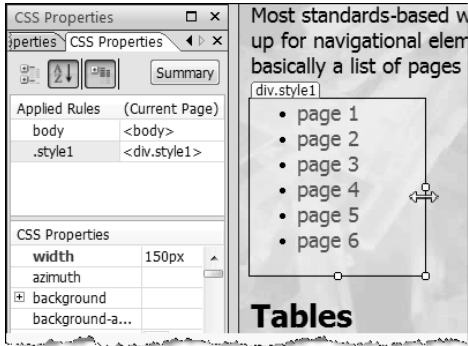
**Figure 7-4.** Using `Wrap Tag` will put your opening and closing `<div>` elements around the selection without having to go to Code view.

3. Make sure you can see the CSS Properties task pane. Before you make any changes, you will see the body properties as the only style definitions, as shown in Figure 7-5.



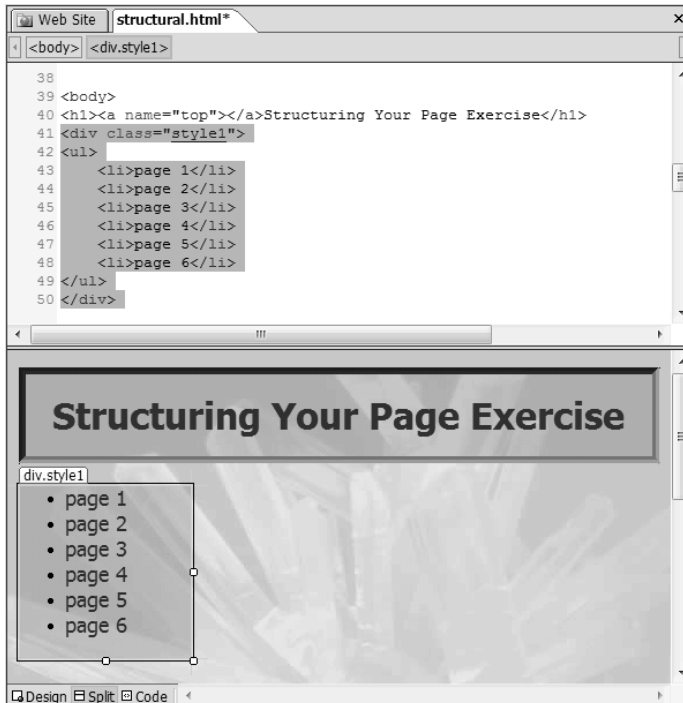
**Figure 7-5.** CSS properties applied to the new `<div>`

Grab the circle on the right side of the `<div>` bounding box, and drag to the left until the width is 150px. You will see the CSS properties change from the body element to style1 (or style + the next incremental number if you already have styles on your page). When you see 150px in the CSS Properties task pane (as shown in Figure 7-6), stop.



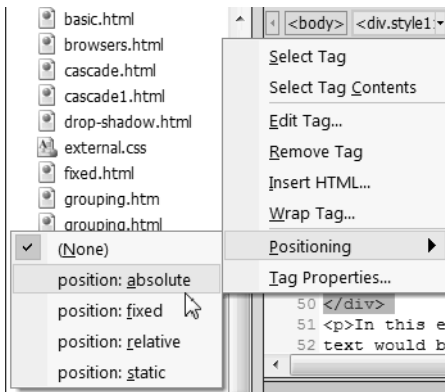
**Figure 7-6.** Pay attention to the CSS Properties box when you need precise measurements.

- Grab the top-left corner of your `div.style1`, and drag it to the beginning of the first paragraph under your `<h1>`, as shown in Figure 7-7.



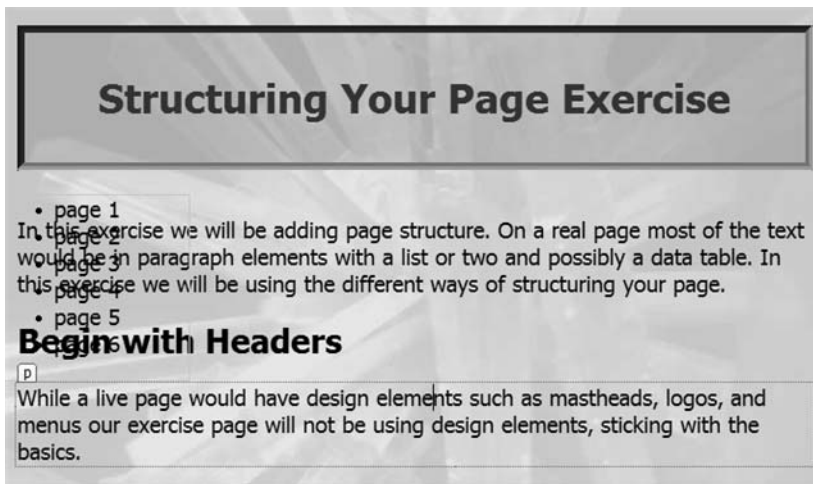
**Figure 7-7.** Remove the menu from the middle of what will become your content section.

5. To turn the `div.style1` into an absolutely positioned element, from the Quick Tag bar choose Positioning ► position: absolute from the fly-out menu, as shown in Figure 7-8.



**Figure 7-8.** Converting an element to `position: absolute`

6. When you convert part of your page to `position: absolute` but leave other elements in the flow, you will have content overlapping, as in Figure 7-9.



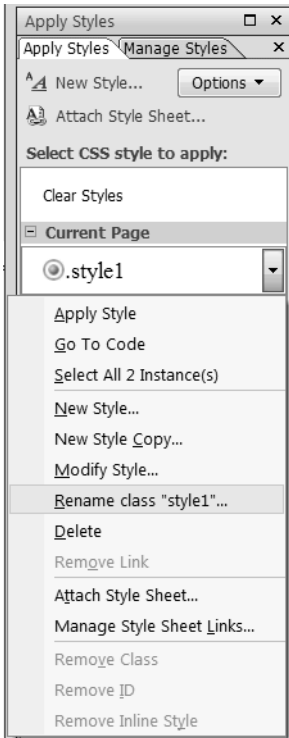
**Figure 7-9.** `div.style1` overlaps the content in the document flow.

---

**Note** Make sure you save your page and preview it in at least one browser before you go on to the next part of this exercise.

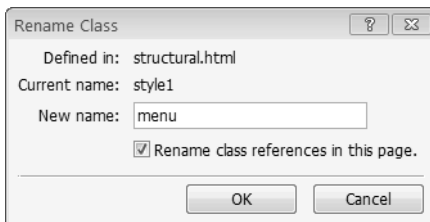
---

7. `style1` does not provide any information that will help you identify what the style does. Although it is easy to see when the style is defined in the page, when you move it to an external stylesheet, it is better practice to have a more descriptive style name. Before you move on to positioning the other content on your page, rename `style1` to `menu`. In the Apply Styles task pane, right-click `.style1`, and select the rename-related option, as shown in Figure 7-10.



**Figure 7-10.** Use the right-click menu on the Apply Styles or Manage Styles task pane to rename styles.

8. When the Rename Class dialog box shown in Figure 7-11 appears, type **menu** for the new name, and make sure the “Rename class references in this page” box is checked.



**Figure 7-11.** You can choose to update all of the elements using this style at the time you rename it.



9. The next step is to position the remaining content below the `<head>` section of the page in a second `<div>`. Click anywhere in the first paragraph away from the overlapping content of `div.menu`, and then select the `<p>` element. Scroll to the bottom of the page, and Shift-click at the end of the last line of text. This will highlight every element between the two points, as shown in Figure 7-12.

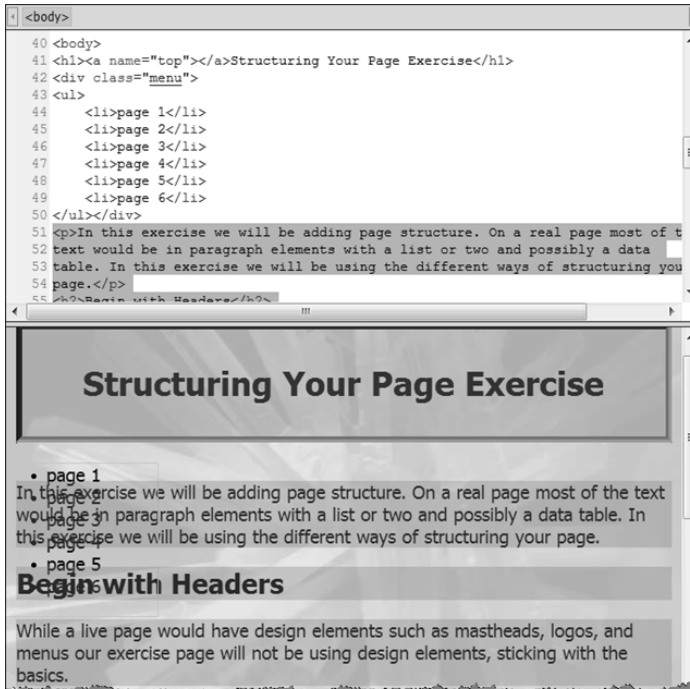


Figure 7-12. Select the content you want to wrap in a `<div>`.

10. Next, from the Toolbox ► HTML section, right-click the `<div>` tag, and choose Wrap, as shown in Figure 7-13.

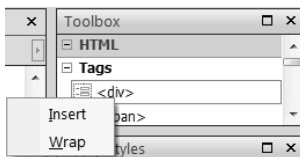
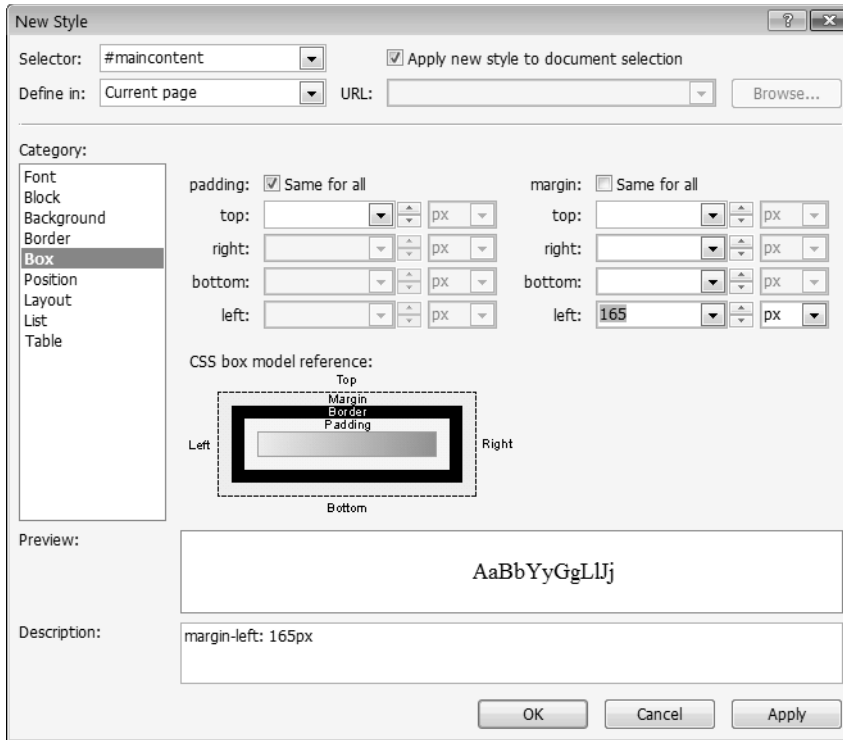


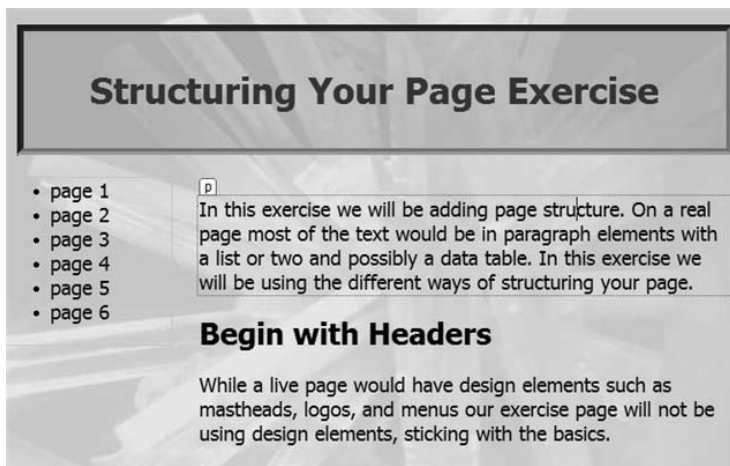
Figure 7-13. To create a `<div>` that contains multiple elements, use Wrap after you highlight the content you want inside.

11. Your content will now have a `<div>` element surrounding all the content you previously selected. Next, launch the New Style dialog box using one of the methods described earlier. In the Selector box, type `#maincontent` to create an ID. Make sure you have a check in the “Apply new style to the document selection” box, and then go to the Box category. Remove the check box from the “margin: Same for all” box, and type `185` in the left box; finally, leave px as the unit of measurement, as shown in Figure 7-14.



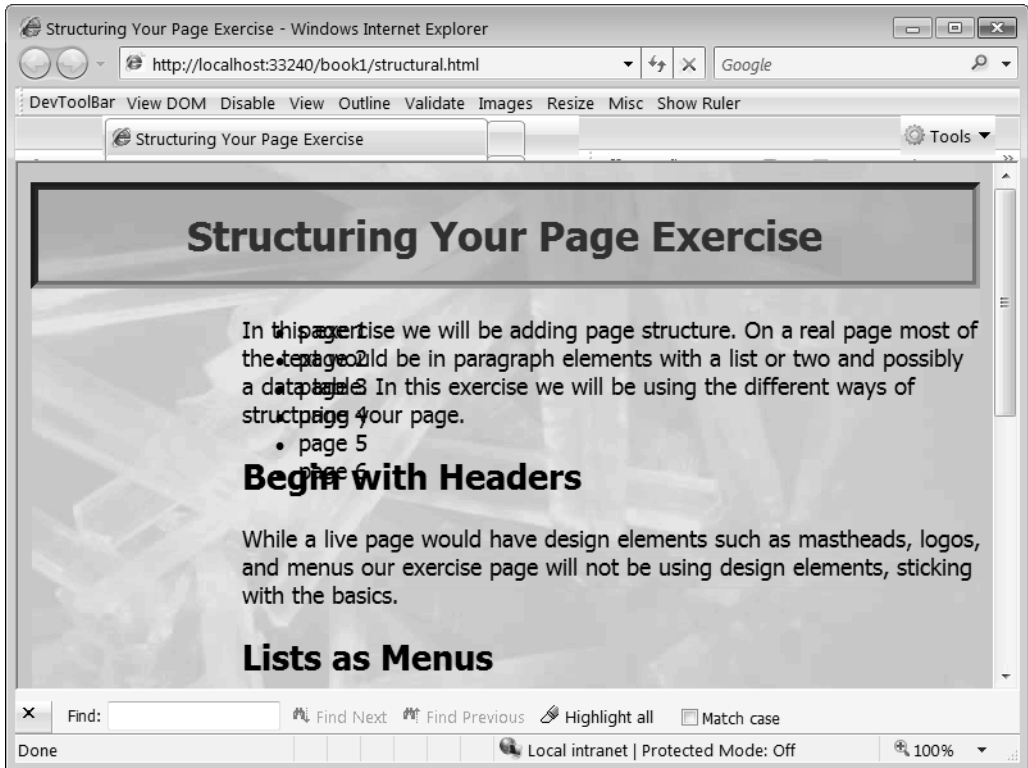
**Figure 7-14.** Define your main content area using a margin to leave space for the side menu.

Save this as `absolute.html` before you preview your page in a browser, as shown in Figure 7-15, in Design view.



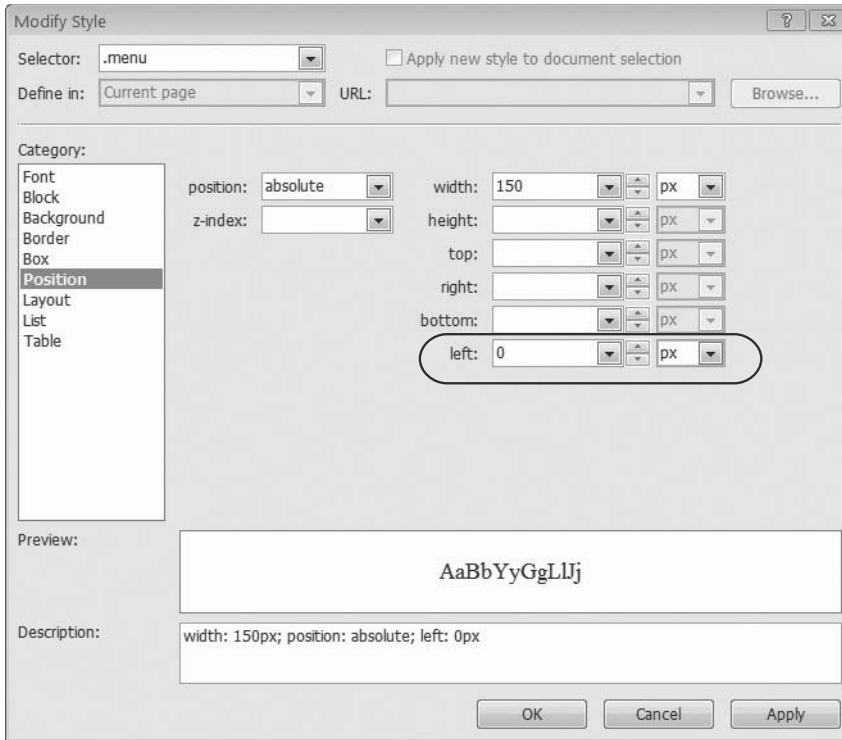
**Figure 7-15.** Under the W3C standards, this is how your web page should appear in a browser.

12. This is how your page should display in a web browser, but you may remember that I said in Chapter 5 that absolute positioning has browser issues. Preview your page in Internet Explorer and Firefox or Opera. Your page will display as expected in your Design view preview in Firefox and Opera, but in Internet Explorer 7 it will look like Figure 7-16.



**Figure 7-16.** Internet Explorer has a display bug and positions the menu incorrectly.

13. Fortunately, this display bug is easy to correct by modifying your menu style to add `left: 0;` to the CSS style definition, as shown in Figure 7-17.



**Figure 7-17.** Modify the `.menu` style to add 0 to the `left` property.

14. Save and preview your page again. Now when you preview your web page in Internet Explorer 7, it will look the same as it does in Firefox, Opera, and Safari. In Internet Explorer 6 and below, the layout will be the same, but the background image on the `<h1>` will be solid instead of translucent.

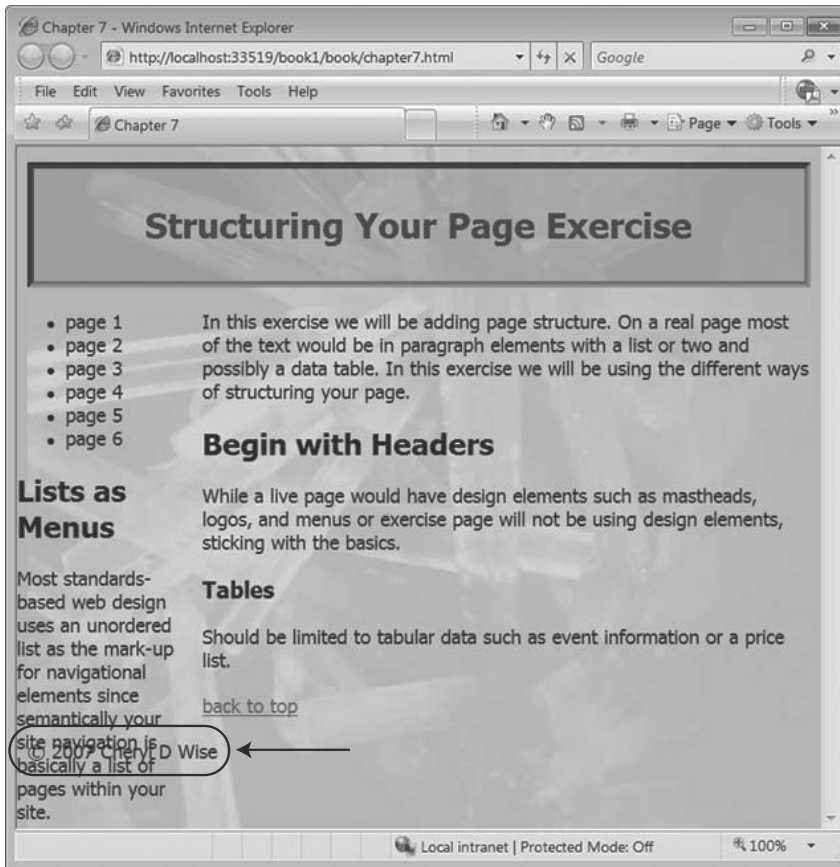
You'll learn more about why the background image displays differently in older browsers in the following chapter.

Although your `absolute.html` page now looks like a nice two-column page, this look is deceptive since it works properly only if the content in the `maincontent <div>` is longer than that in the `menu <div>` and there is not a footer or side column. If you were to add the footer `<p id="footer">© 2007 Cheryl D Wise</p>` to the bottom of the `structural.html` web page and add more content to the `menu <div>` to make it longer than the `maincontent` section without making any style definition changes, your copyright notice would overlap the menu section text, as shown in Figure 7-18.

---

**Note** For the purposes of demonstrating the effect, when the `menu <div>` is longer than the main content, I moved the “Lists as Menus” section over to the menu section and used `display: none` in the table definition to ensure the column was longer than the `maincontent` section.

---



**Figure 7-18.** Notice how the copyright symbol and year overlap the text?

Unlike the Internet Explorer bug that caused the absolutely positioned menu to overlap the main content section, this type of overlap is not easily cured by adding a bottom setting to move the menu <div> text up from the bottom of the page. This is why using absolute positioning for layout causes problems.

## LEGEND FOR STYLES TASK PANES

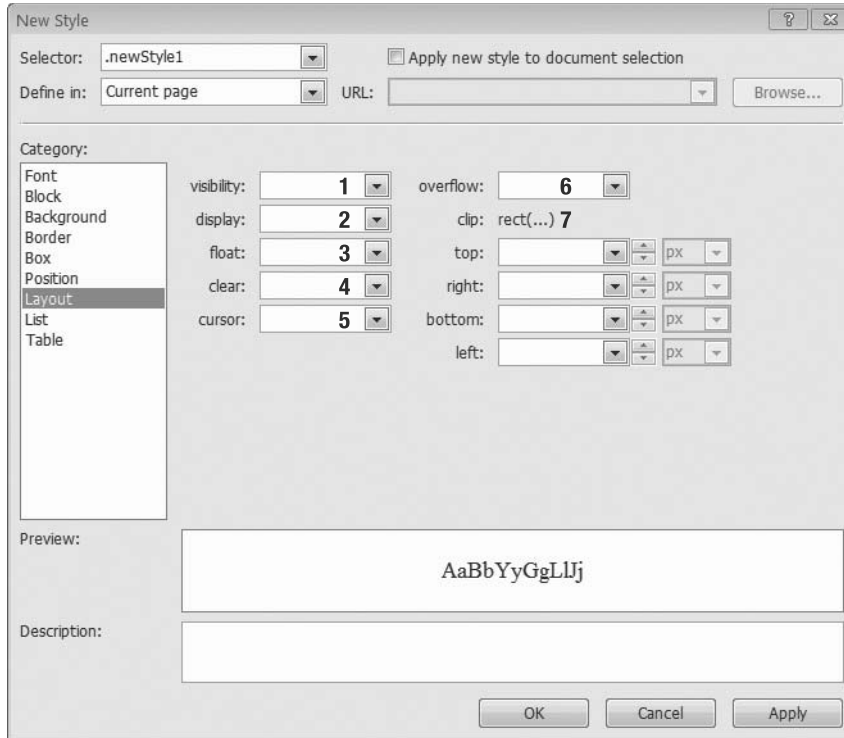
Both the Apply Styles and Manage Styles task panes use the same color-coded system of dots:

- blue dots = HTML element
- green dot = CSS class
- red dot = CSS ID

In addition to color coding, styles that are in use are circled.

## Working with the Layout Category

The next category in the New Style and Modify Style dialog boxes is Layout, as shown in Figure 7-19. This is where Microsoft chose to group `float` and other properties that affect page layout. Certain properties here can help you avoid the overlap issues you saw in Figure 7-18.



**Figure 7-19.** This group is used primarily for page display affects.

Here are the options (the numbers relate to Figure 7-19):

1. *visibility*: This is usually reserved for dynamic effects such as hiding or showing sections of your page based on the visitor's actions. This offers you the ability to hide or collapse the space occupied by the content to which this property is applied. Although on most elements hidden and collapse should have the same effect, collapse has a special meaning for table rows and columns. Unfortunately, Internet Explorer (version 7 and below) does not implement collapse correctly.
2. *display*: This changes the default display. Some settings of this property are only partially supported by most web browsers. Changing between block and inline is supported by all but the oldest of CSS-aware browsers. The other display properties all require browser testing to determine whether the browsers you are concerned about will implement the property in the manner you desire.

3. *float*: This is the most commonly used property in this category. The float property is the replacement for the HTML align attribute for images and other block-level content. You can also use float to make columns for page layout instead of absolute positioning because of better browser support. You will use this property for layout in Exercise 7-2.
4. *clear*: This is the companion property for float and is what makes it possible to use floats for page layout.
5. *cursor*: This is a setting that you should rarely use. The primary use for this setting is when you want to alert your visitor to a JavaScript action or link that they can click, since JavaScript actions do not change to indicate that there is a link or other action available for the text or image.
6. *overflow*: This provides a way to set a fixed size and provide scroll bars. This is useful when you want to include content in a section with a specific height and width but still want the visitor to be able to see all the content by adding scroll bars.
7. *clip*: This acts like cropping in your image editor and will display only the portion of the content you choose. Using top, right, bottom, and left tells the browser where to clip the portion to be displayed. clip is a rarely used property.

## Exercise 7-2. Using Float for Layout

Although you are using the same basic page you used at the beginning of this chapter's exercises, I have removed the table to make it easier to see the effect of using float instead of position: absolute to create columns.

1. Use the following HTML for this exercise, and save it as floats.html:

```
<!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" />
  <meta http-equiv="Content-Language" content="en-us" />
  <title>Float Layout Exercise</title>
  <style type="text/css">
h1 {
  border-style: outset inset inset outset;
  border-width: thick;
  border-color: #22292D #999999 #999999 #22292D;
  font-size: 1.8em;
  text-align: center;
  background-image: url('images/h1-bg.png');
  padding: 1em;
}
}
```

```
body {
  font-family: Tahoma, Helvetica, Arial, sans-serif;
  background-image: url('../chapter06/images/crystal-bg.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: center center;
  background-color: #C6C7C3;
}
.menu {
  position: absolute;
  width: 150px;
  left: 0;
}
#maincontent {
  margin-left: 165px;
}
</style>
</head>
<body>
  <h1><a name="top"></a>Structuring Your Page Exercise</h1>
  <div class="menu">
    <ul>
      <li>page 1</li>
      <li>page 2</li>
      <li>page 3</li>
      <li>page 4</li>
      <li>page 5</li>
      <li>page 6</li>
    </ul>
    <h2>Lists as Menus</h2>
    <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>
  </div>
  <div id="maincontent">
    <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>
    <h2>Begin with Headers</h2>
    <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>
    <h3>Tables</h3>
```

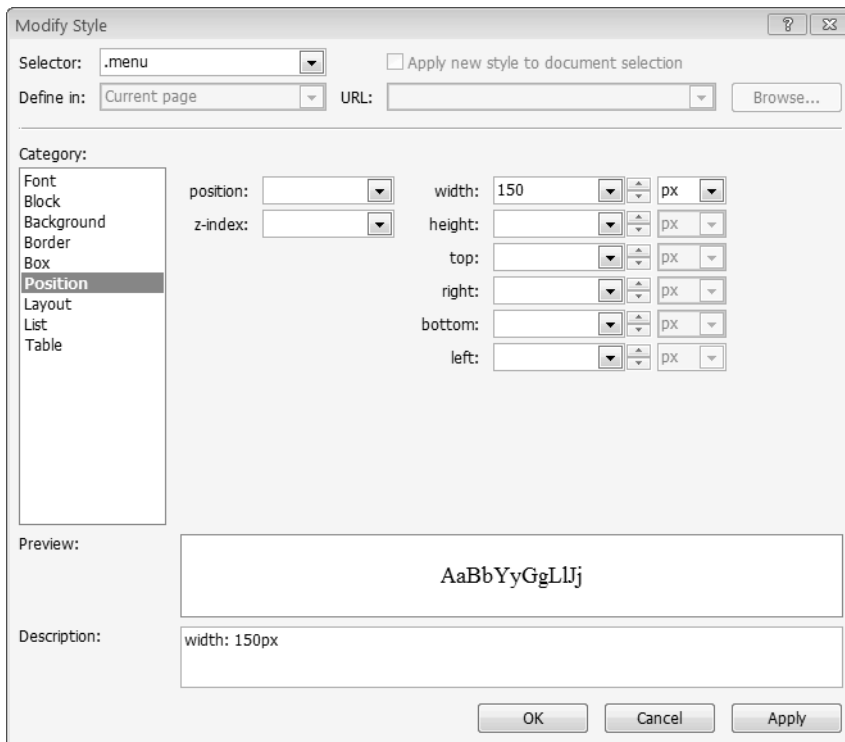


```

    <p>Should be limited to tabular data such as event information or
    a price list.</p>
    <p><a href="#top">back to top</a></p>
  </div>
  <p>&copy; 2007 Cheryl D Wise</p>
</body>
</html>

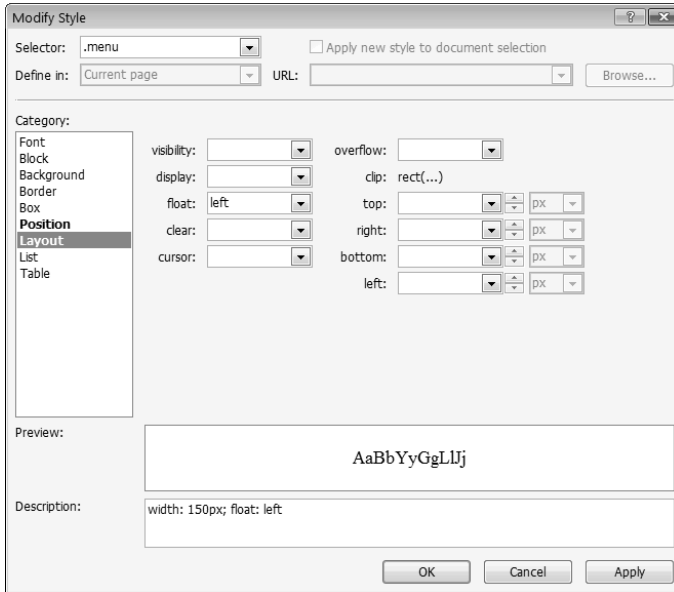
```

- From one of the styles task panes, right-click the `.menu` style, and select **Modify Style**. On the **Position** category page, highlight and delete `absolute` from the position box and `0` from the left box, as shown in Figure 7-20.



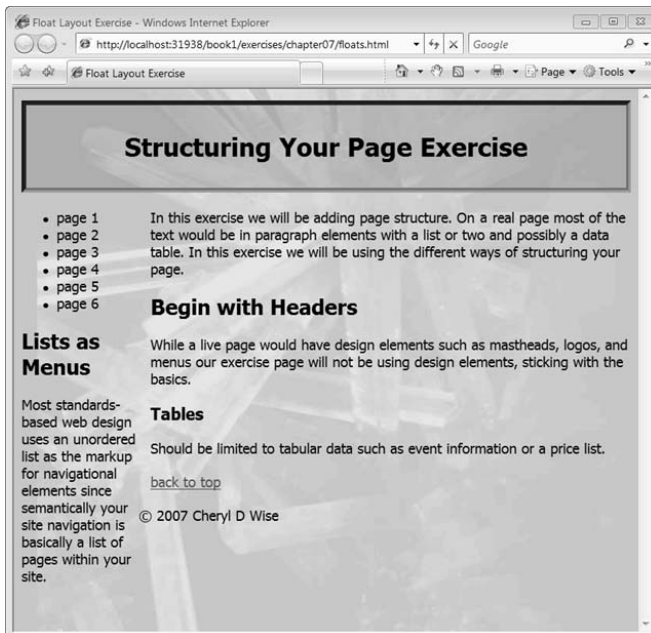
**Figure 7-20.** Leave only the width definition in the Position category.

- Next, select the **Layout** category, and choose `left` from the float box, as shown in Figure 7-21.



**Figure 7-21.** To maintain the menu in the left column, choose left for the float drop-down list.

4. Apply your changes, and then preview them in your web browser. Notice that the copyright text no longer overlaps the menu text but flows to the right of the menu, as shown in Figure 7-22.

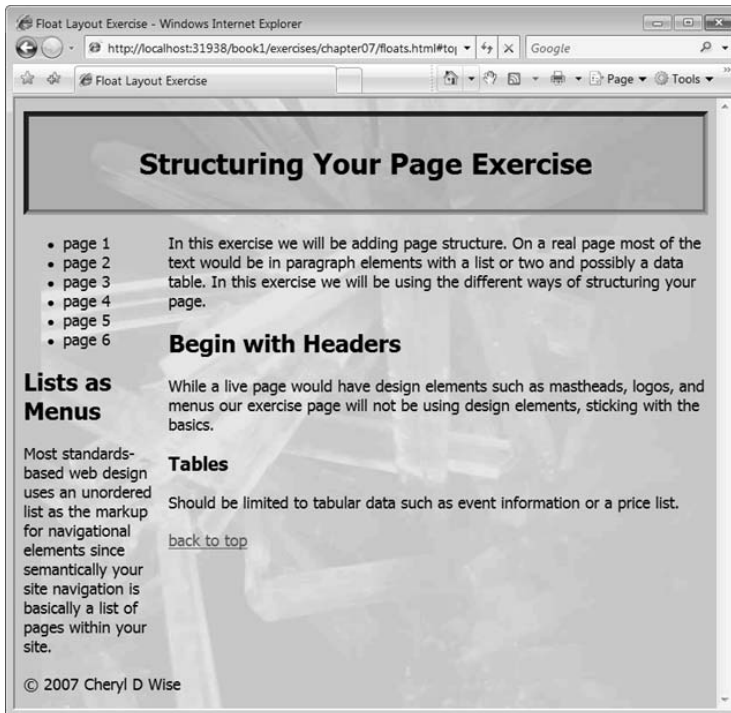


**Figure 7-22.** There's no more overlap, but it's still not where you want the copyright information to display.

Because the copyright text is not inside the `maincontent <div>`, it is slightly closer to the text in the menu section. Although this layout is not bad, if your `maincontent` was shorter or the menu was longer than the page, it would look strange to your visitors.

### Moving the Copyright Information

In most cases, you will want your copyright information to be at the very bottom of the page, as in Figure 7-23, which is usually considered to be a more desirable page layout.



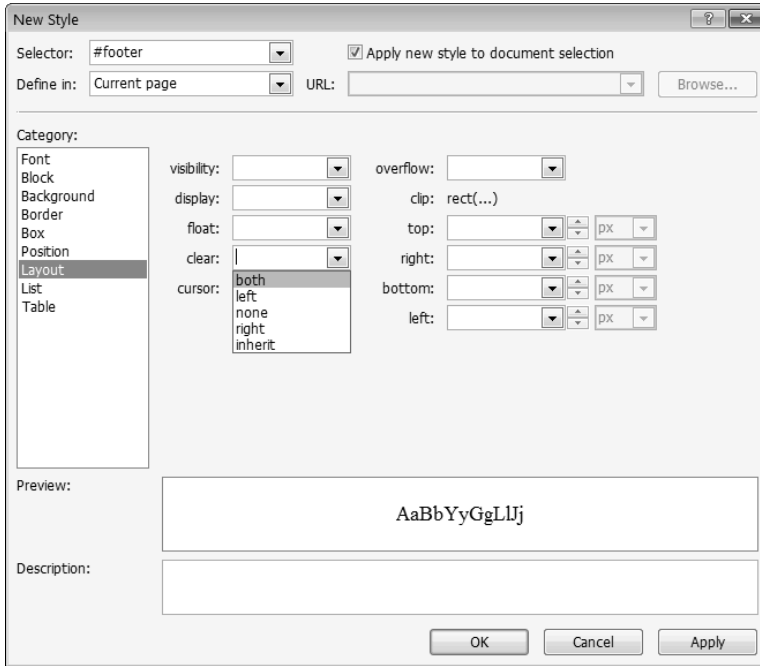
**Figure 7-23.** *If the main content is longer than the menu content, then your copyright moves to the left and flows across the page as far as necessary to display the content.*

To achieve the goal of having the copyright information in the expected place, I made the browser window narrow enough so that the `maincontent` section is longer than the menu. You do not know what the visitor's browser settings are, so how do you get the copyright below both sections of your page layout?

You use the `clear` property on the first element you want to appear below your columns. In this case, you have a paragraph of content:

1. Select the paragraph you created with your copyright or other information in it, and launch the New Style dialog box.
2. Change the selector name to `#footer`, and make sure you check the "Apply new style to document selection" option.

- Then from the Layout category, in the clear box, select both, as shown in Figure 7-24.



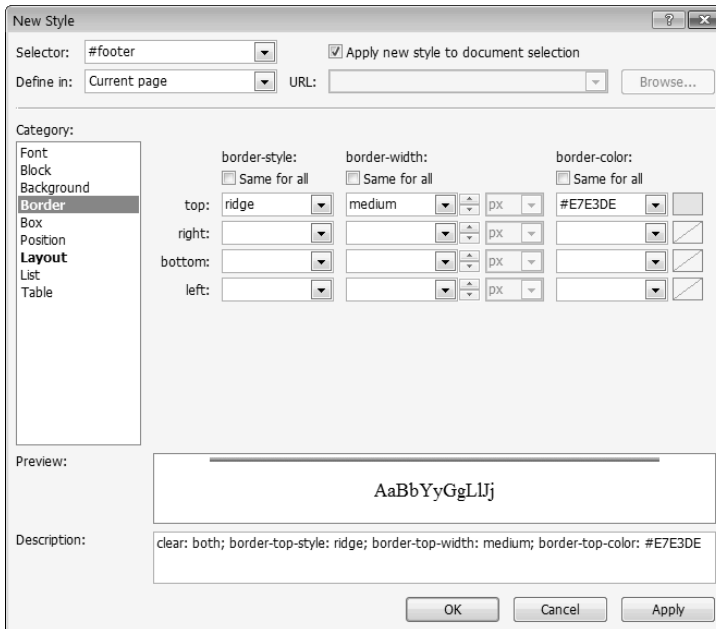
**Figure 7-24.** *The clear property will start your content after the floated elements.*

In this case, you could have chosen left since there is no floated content on the right, but since you do not always know what content will be inside the `maincontent` section, using both is the safe choice. That way, if you do float an image or some other content inside the `maincontent` section, your footer will always clear it without having to change your style in the future.

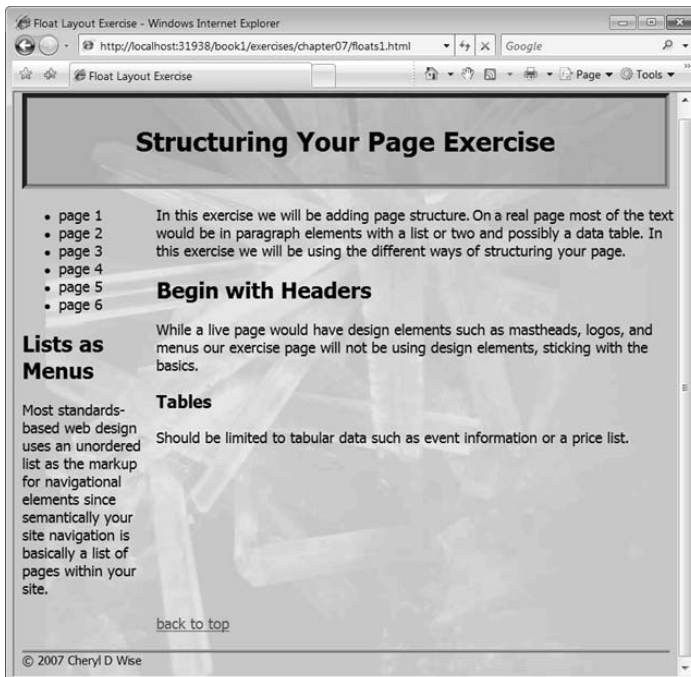
### Learning More About Floats

To more clearly illustrate the effect of clear on floats, you'll now add a top border to the footer and a bit of padding between the border and the text:

- Before you use the OK button, go to the Border category, and set the top border as shown in Figure 7-25, or choose your own top border settings. I urge you to experiment.
- Optional: change the font size for your `#footer` to `.8em` to follow the tradition of making your legal notices and other footer text smaller than the rest of your web page. Now when you preview your page in your web browsers, you will see a finished web page. When you preview in your web browser, your page should look like Figure 7-26.



**Figure 7-25.** Remember to remove the check mark for each of the “Same for all” boxes.



**Figure 7-26.** Your page now uses the most common page layout on the Web: a masthead, two columns, and a footer.

## Lists

Lists are one of the most useful HTML elements. You can use a list to group related items, create steps, rank options, define terms, and create an outline.

In the `structural.html` page, you are following the recommendation for web professionals to use lists for site navigation. Although it is clear when you look at the page, as shown in Figure 5-27, that the page list is the menu, it is not the most attractive menu. Using the List category in the style dialog boxes, you can change that to make the menu more attractive. Since you may not want all of your lists to look like menus, you will be using a contextual selector to style the list inside the menu in Exercise 7-3.

### Exercise 7-3. Creating a List As a Menu Using Contextual Selectors

In this exercise, you will make the menu list look like buttons using the `absolute.html` file saved as `menu.html`.

```
<!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" />
  <meta http-equiv="Content-Language" content="en-us" />
  <title>Chapter 7</title>
  <style type="text/css">
  body {
    font-family: Tahoma, Helvetica, Arial, sans-serif;
    color: #000000;
    background-image: url('images/crystal-bg.jpg');
    background-color: #C6C7C3;
    background-repeat: no-repeat;
    background-position: center center;
    background-attachment: fixed;
  }
  h1 {
    border-left: thick outset #22292D;
    border-right: thick inset #999999;
    border-top: thick outset #22292D;
    border-bottom: thick inset #999999;
    padding: 1em;
    font-size: 1.8em;
    color: #333333;
    background-image: url('images/h1-bg.png');
    text-align: center;
  }
</style>
</head>
</html>
```

```
.menu {
  position: absolute;
  width: 150px;
  left: 0;
}
#maincontent {
  margin-left: 165px;
}
</style>
</head>
<body>
  <h1><a name="top"></a>Structuring Your Page Exercise</h1>
  <div class="menu">
    <ul>
      <li>page 1</li>
      <li>page 2</li>
      <li>page 3</li>
      <li>page 4</li>
      <li>page 5</li>
      <li>page 6</li>
    </ul>
  </div>
  <div id="maincontent">
    <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>
    <h2>Begin with Headers</h2>
    <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>
    <h2>Lists as Menus</h2>
    <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>Tables</p>
    <p>Should be limited to tabular data such as event information
    or a price list.</p>
    <table style="width: 100%">
      <tr>
        <th>item no.</th>
```

```

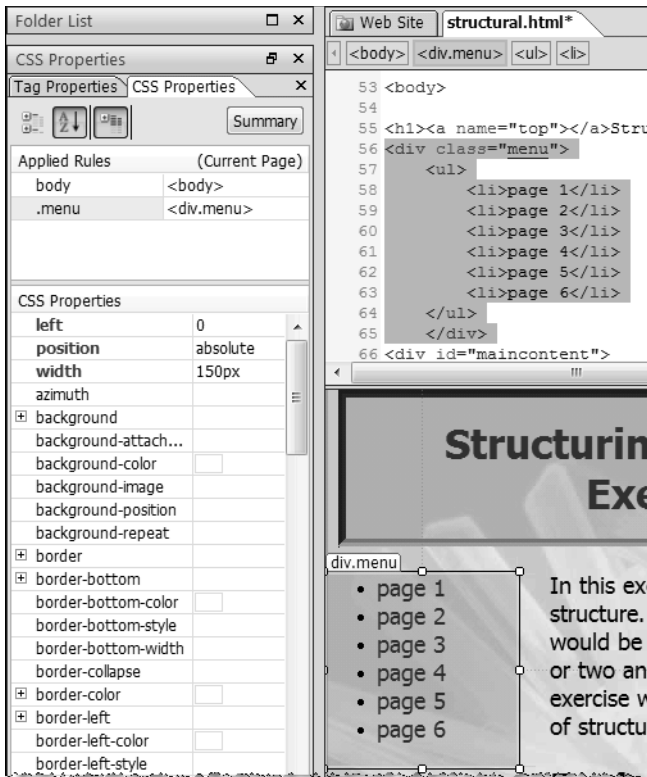
        <th>description</th>
        <th>price each</th>
        <th>per dozen</th>
    </tr>
    <tr>
        <td>1</td>
        <td>red balloon</td>
        <td>$0.25</td>
        <td>$2.25</td>
    </tr>
    <tr>
        <td>2</td>
        <td>blue balloon</td>
        <td>$0.25</td>
        <td>$2.25</td>
    </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>
    <tr>
        <td>5</td>
        <td>gold mylar balloon</td>
        <td>$1.75</td>
        <td>$19.00</td>
    </tr>
</table>
<p><a href="#top">back to top</a></p>
</div>
</body>
</html>

```

Before you start creating the appearance of buttons, you will change the menu <div> in the structural.html page to float for the reasons stated in Exercise 7-2. This time you will use the CSS Properties task pane instead of the Modify Style dialog box since you know exactly what you want to change.

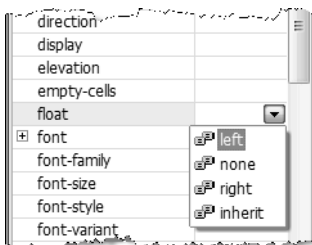
1. With the `div.menu` selected in Design view or from the Quick Tag bar, you will see the properties that apply to the <div> highlighted in the Applied Rules section of the CSS Properties task pane, as shown in Figure 7-27. Delete the entries for position and left.





**Figure 7-27.** The fastest way to remove or modify a property is to use the CSS Properties task pane.

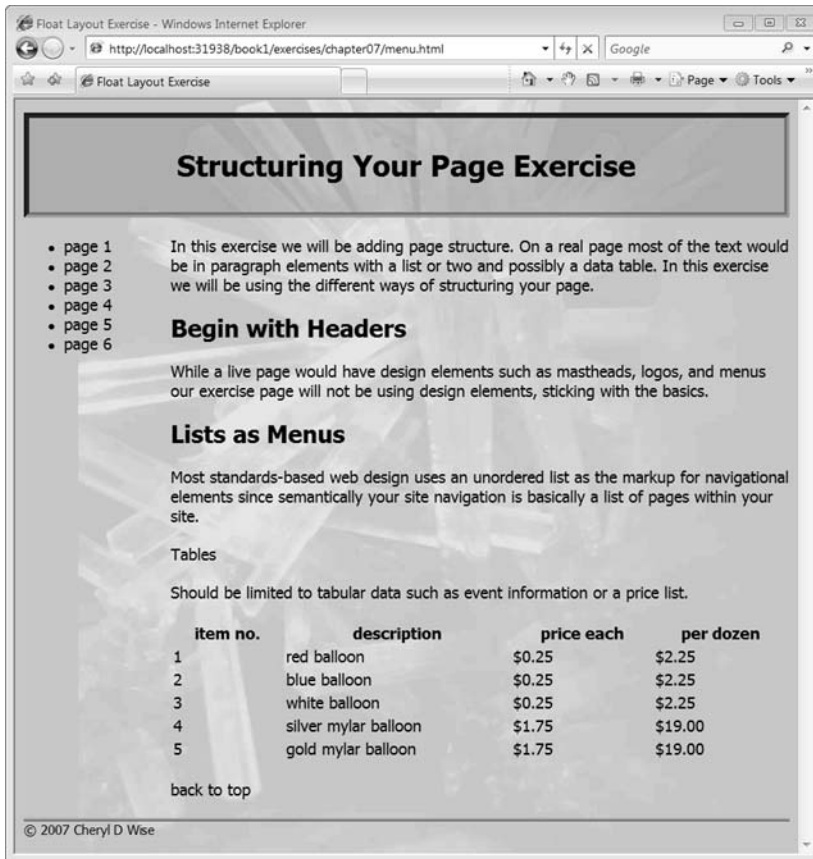
- Next scroll down until you see float in the properties list, and select left from the value drop-down list, as shown in Figure 7-28.



**Figure 7-28.** All of the options available in the style dialog boxes are listed in the CSS Properties task pane, sorted alphabetically in this view.

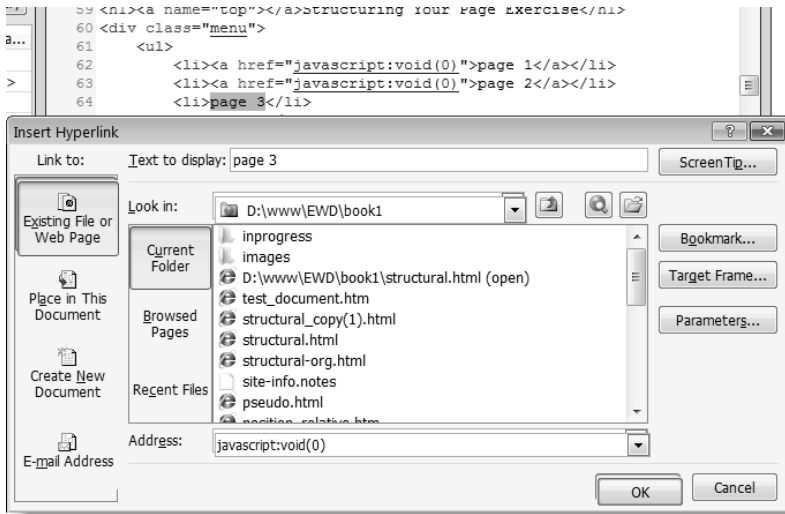
- Optional: re-create or copy and paste your #footer rule from the floats.html page.

4. Preview the page in your browser, as shown in Figure 7-29, before you go to the next step.

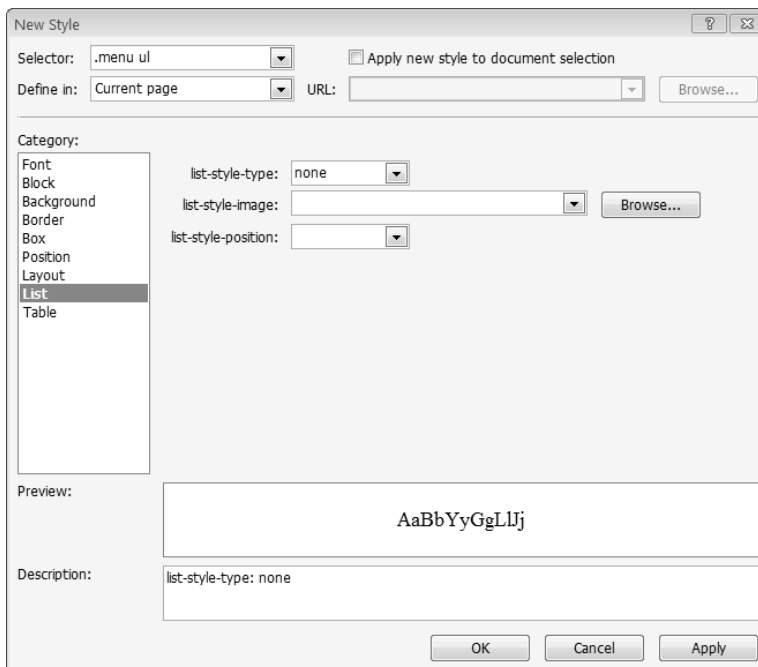


**Figure 7-29.** A very simple two-column page ready to have the menu styled

5. Each of the list items must be a hyperlink before you can style your menu. Individually select each of the page items, right-click, and choose Hyperlink to launch the Insert Hyperlink dialog box.
6. Instead of selecting a real file to link to, type `javascript:void(0)`, which will allow you to have an active link that doesn't go anywhere until you have real pages to link to in your website. Repeat for each of the page items in your menu, as shown in Figure 7-30.
7. Now launch the New Style dialog box.
8. For your selector, type `.menu ul`. This is your first use of a contextual selector using Expression Web's CSS tools. This selector will target this rule to any unordered list inside the menu `<div>`. Because this definition relies on the placement of the HTML `<ul>` element, you do not need to check the "Apply new style to document selection" box.
9. Open the List category. The next step will be to remove the default bullet by selecting none from the list-style-type drop-down list, as shown in Figure 7-31.

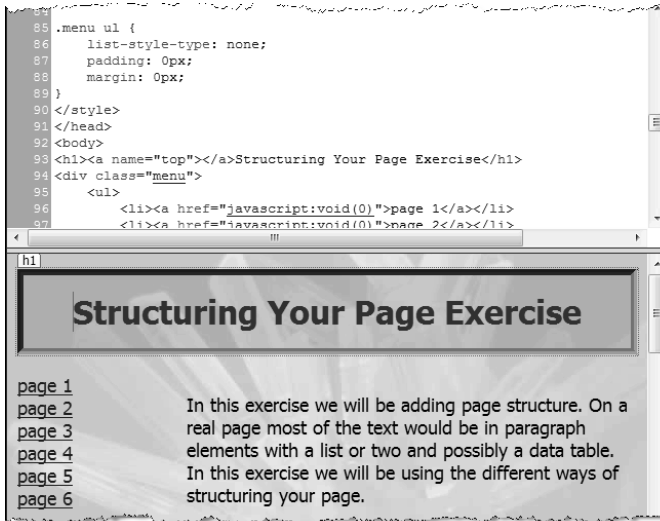


**Figure 7-30.** You need to create your hyperlinks to see all the effects you will be creating.



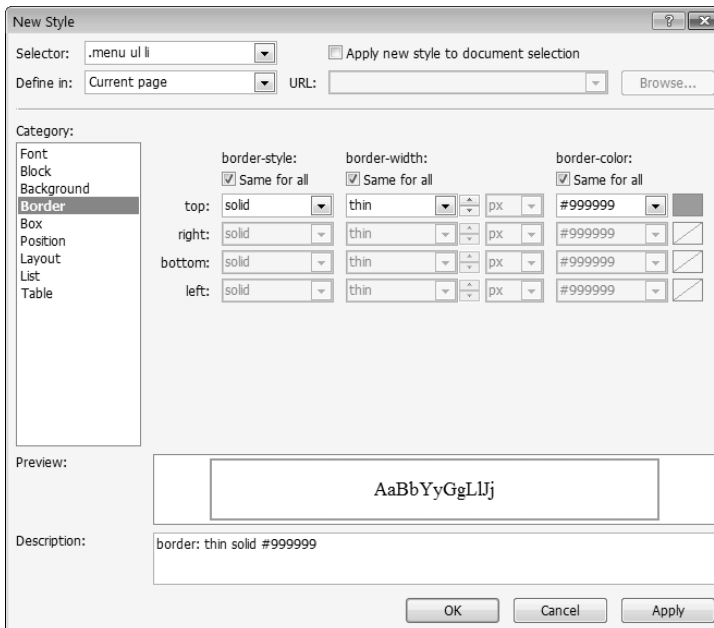
**Figure 7-31.** Begin by removing the bullets from your list.

10. By default, list items are indented even when the bullet is not being displayed. To remove this space, move to the Box category, and set the margin and padding to 0. Your page should look like Figure 7-32.



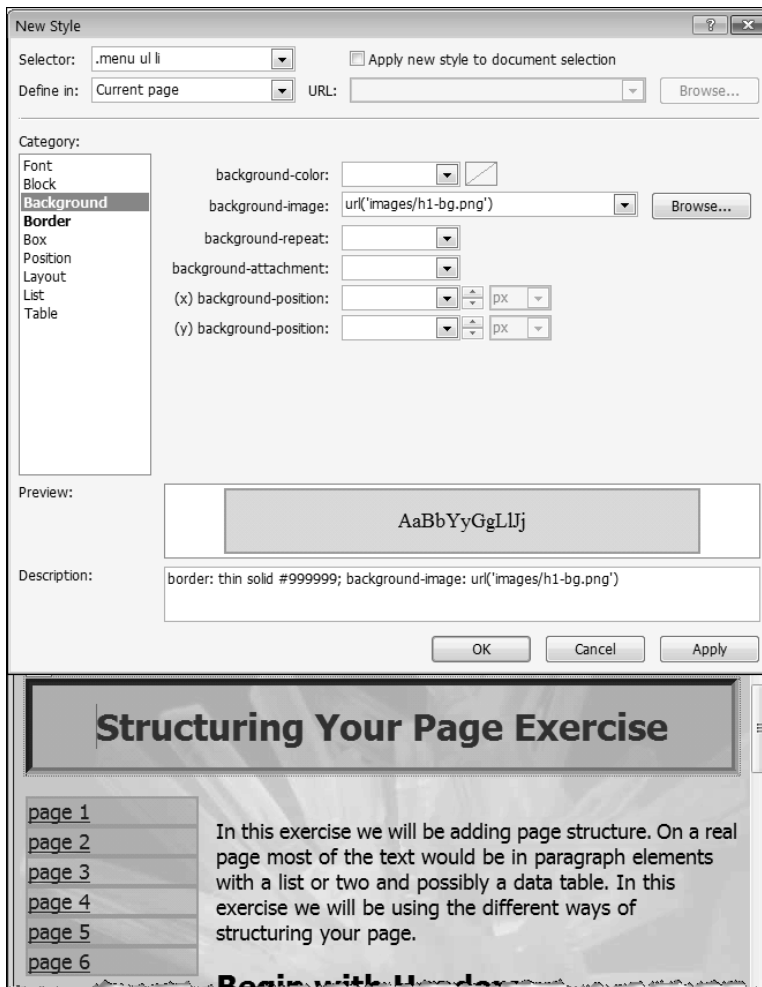
**Figure 7-32.** Notice the bullets have disappeared and your list has moved to the left.

11. Next, you will make the links look more like a button by styling the list items. Open a New Style dialog box, and type `.menu ul li` as the selector. You could use the less-specific `.menu li`, but it would also style list items inside an ordered list as well as those inside an unordered list. For now you will stay with the more specific definition.
12. Move to the Box category, and apply a border to all four sides. I have made those changes in Figure 7-33.



**Figure 7-33.** The choice of border style can make a difference to how your buttons look.

13. To make the list item look more like a button, you'll use a background image. In this case, let's reuse `h1-bg.png`, as shown in Figure 7-34.



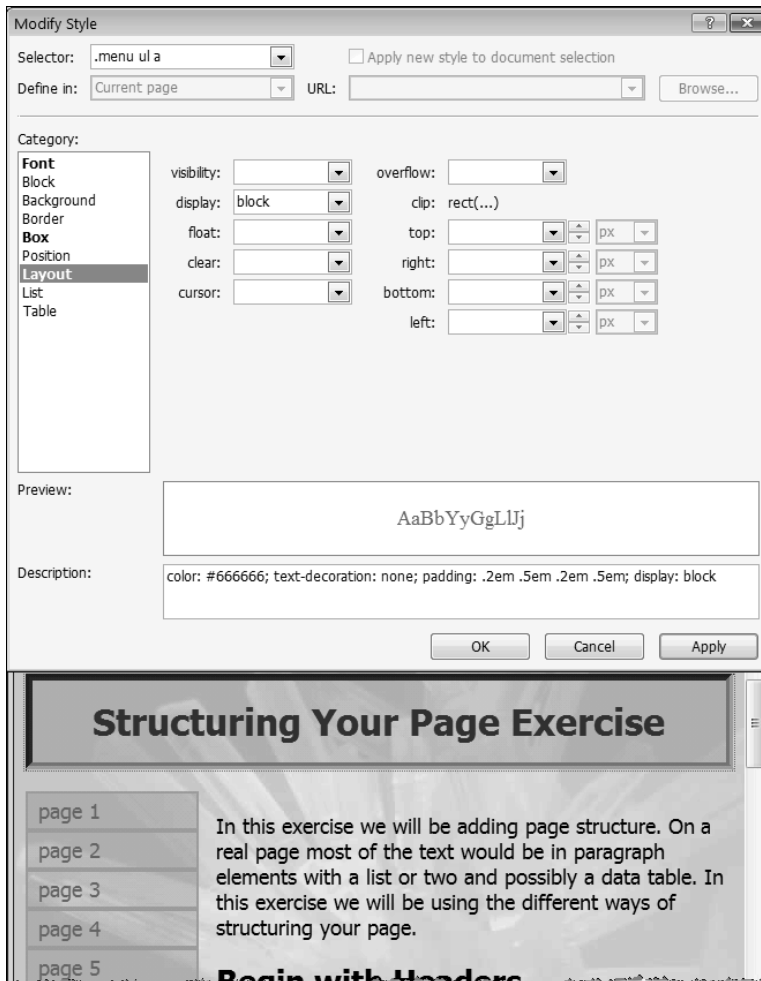
**Figure 7-34.** The unordered list is beginning to look more like buttons.

### Changing the Link to Block

Since you removed the default padding and margin, you need to add space between the border and the text. Although you could have added padding to the list items for the links to act more like buttons, you want the entire box to be a clickable link, not just the text. You can accomplish this by changing the hyperlink from inline to block and applying padding to the link. On a real website, if the links were close to the same width, you could center the text, but when the link text varies, it generally looks better to use padding.

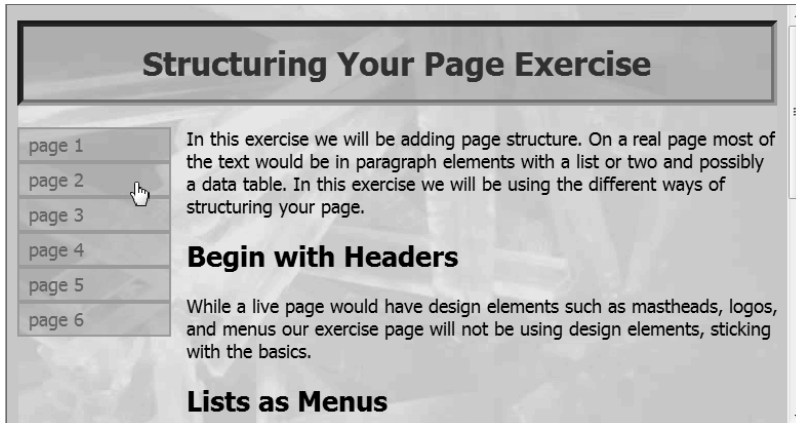
In addition, you will want to change the link default blue color and underline.

1. Open the Modify Style dialog box again. This time your selector will be `.menu ul a`, which will target all anchor elements inside an unordered list. There is no need to specify the `li`. Any other links in the menu `<div>` will still have the default link style for the page. The settings you will use are as follows:
  - Font category, color: #666666; text-decoration: none;
  - Box category, padding: .2em .5em .2em .5em;
  - Layout category, display: block;
2. When you use the Apply button, the menu should look like the one in Figure 7-35.



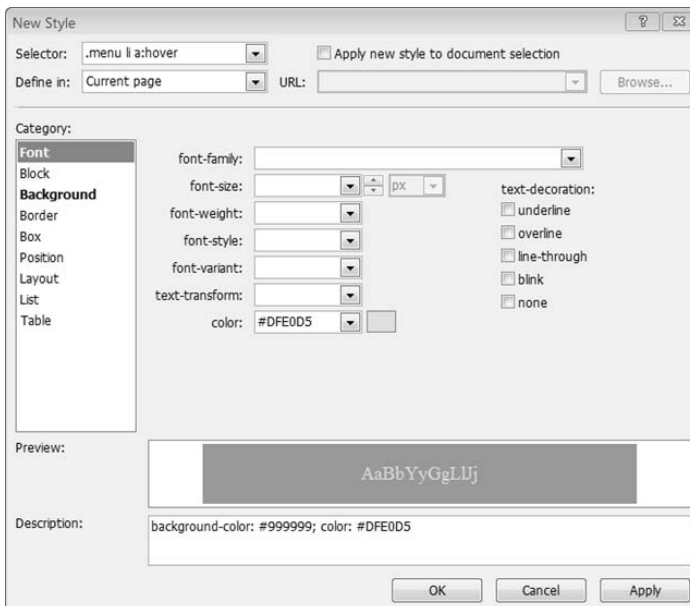
**Figure 7-35.** The menu now looks more like buttons, and the entire area is clickable.

3. Before you add a hover effect, save your file, and preview it in your browser. Notice, as in Figure 7-36, that anywhere inside the list item, even away from the menu text, your cursor will change to a hand to indicate a link.



**Figure 7-36.** *Hyperlinks created using the `<a>` element will always show a hand cursor.*

- The final piece of style for this web page will be to add a hover or rollover to the menu. Launch the New Style dialog box. This time the selector will be `.menu li a: hover`. For the mouseover action, you will reverse the color so that the text is light (hex color `#DFE0D5`) on a dark background (hex color `#999999`) using the settings shown in Figure 7-37.



**Figure 7-37.** *Now when you mouse over the background, color will replace the translucent PNG image.*

Save your file before you continue. You have now styled a list to look like a set of buttons, but the exact look will differ between browsers.

## CONDITIONAL COMMENTS FOR BROWSER BUGS

When you were doing Exercise 7-3, you saw significant differences between web browsers. These differences in Internet Explorer 7, Firefox, Opera, and Safari primarily show up in the border styles and are relatively trivial. On the other hand, Internet Explorer 6 and below will look quite different because those browsers do not support translucent PNG images. This means that the `<h1>` and menu have a solid background that might not be the color you want.

### Telling Internet Explorer to Use a Different Background

To tell Internet Explorer to use a different background from other web browsers, use conditional comments. Ever since Internet Explorer version 5 came out, there has been the ability to pass information to Internet Explorer. This means you can easily give Internet Explorer styles and even content that is not displayed in other browsers to compensate for some of the CSS rendering errors or lack of support for features such as PNG alpha transparency. It gets better because you can target specific versions of Internet Explorer. Internet Explorer 7 has fixed many CSS rendering bugs and offers full PNG support. As a result, you want Internet Explorer 7 to continue to receive `h1-bg.png`, but you do not want Internet Explorer 6 or earlier to apply the background image. Instead, you might choose to display either a different image or a solid color background. You can achieve this using a simple comment in the `<head>` section of your page:

```
<!--[if lte IE 6]>
<style type="text/css">
.menu ul a, h1 {
    background-image: none;
    background-color: #C0C0C0;}
</style>
```

To break this piece of code down, the first thing you see is an opening HTML comment tag, `<!--`. This means the code will not render in your browser and will be ignored by every browser and the W3C validator if the browser the visitor is using is not Internet Explorer.

The conditional comment itself starts with a square bracket and contains the browser target information. Here you are using `lte`, which means earlier than or equal to Internet Explorer 6. To target a different browser version than the one listed, in each following snippet replace the 5 or 6 with the appropriate version such as 5.5, 6, or 7:

```
<!--[if IE]>
This conditional comment is for all versions of Internet Explorer
<![endif]-->
```

```
<!--[if IE 5]>
This conditional comment is for Internet Explorer 5 only.
<![endif]-->
```

```
<!--[if gt IE 5]>
According to the conditional comment this is Internet Explorer
greater than but not equal to 5.
<![endif]-->
```



```
<!--[if gte IE 5]>
According to the conditional comment this is Internet Explorer
5 and above.
<![endif]-->

<!--[if lt IE 6]>
This conditional comment is for Internet Explorer lower than
but not including 6.
<![endif]-->

<!--[if lte IE 6]>
This conditional comment is for Internet Explorer lower or equal to 6.
<![endif]-->
```

By using conditional comments targeted toward Internet Explorer 6 and below, you can remove the translucent background and replace it with a solid color that harmonizes with the page background image. The following code would go after the closing `<style>` element and before the opening `<head>` element on your page:

```
<!--[if lte IE 6]>
<style type="text/css">
.menu ul a, h1 {
    background-image: none;
    background-color: #COCOCO
}
</style>
<![endif]-->
```

Remember that one of the primary benefits of using CSS is that you can provide an enhanced view to those with modern browsers and still provide a usable and attractive site to those who do not have the latest and greatest browser.

## Building a Page Layout

In this chapter you've been building a page layout as you worked through the CSS options in Expression Web. There are two types of page layouts:

- A fluid layout will shrink and grow depending on the width of the visitor's browser. Typically a fluid layout will have two or more columns. Frequently at least one of the columns will have a fixed width, while the other will expand or contract with the width of the browser.
- A fixed-width layout has a specified width and is usually centered in the web browser.

The structural.html web page is a fluid page that, because of the fixed background, would look better as a fixed-width page. If you have been previewing your web pages in a browser that is 1,024 pixels or wider, you may have noticed that the line length on the text is longer than most people prefer to read even with a second column. The content overflows the

background image in a way that can be distracting, especially in those browsers that fully support the translucent background. There is more than one way to restrict the width of our page. The simplest method is to put a width on the `<body>` element and center it in the page over the background image; you'll do that in Exercise 7-4.

### Exercise 7-4. Creating a Centered, Fixed-Width Layout

With your `menu.html` web page open, right-click the body style in the Modify Styles task pane, and select Modify Style.

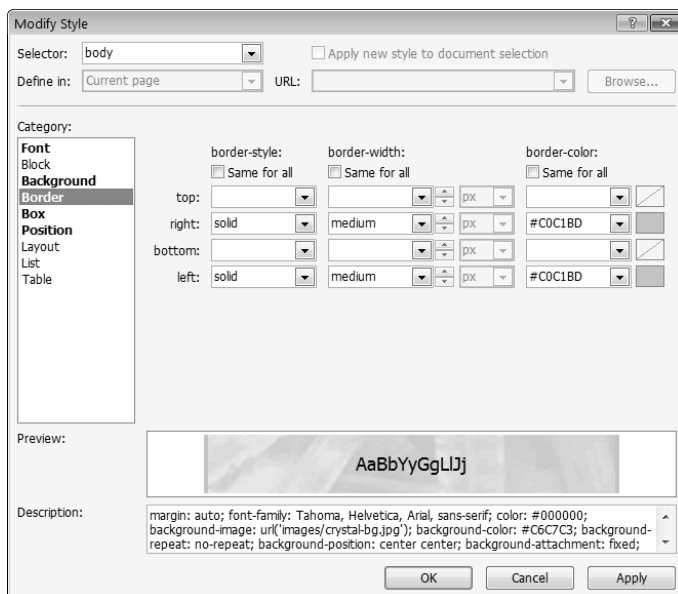
1. From the Position category, set the width to 740px, which is the width of the image. You're using 740px because it will not cause a horizontal scroll in any web browser viewed at 800×600.

---

**Note** 800×600 is the outside dimension of the web browser when viewed full screen on a computer monitor that has a screen resolution of 800×600. To get the amount of space that is available to display your web page, you must subtract a minimum of 20px for the horizontal scroll bar (Internet Explorer always reserves space for the horizontal scroll bar whether it is needed or not) and the side chrome of the browser. Although in most cases 760px will not cause a horizontal scroll, you might not want the layout right against the sides of the browser.

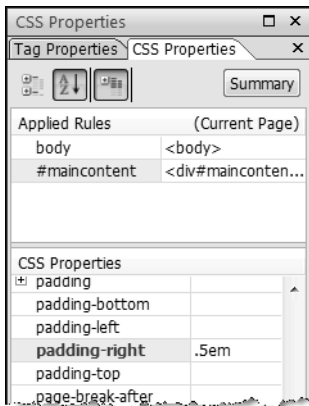
---

2. In the Box category, use 0 for the top and bottom margins and auto for the left and right margins. You do not want to add padding to the body because that would cause your `<h1>` and menu to have space between the items. Set the padding to 0 to make sure that all browsers do not apply any default padding. You will be applying a border to just the left and right sides of the `<body>` element, as shown in Figure 7-38.



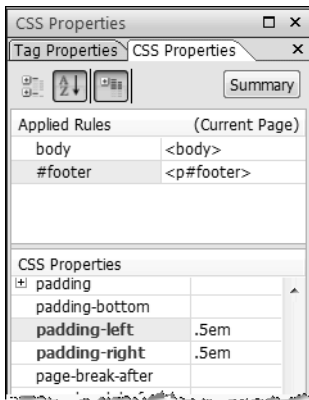
**Figure 7-38.** You can see the faint border that defines not only the edge of the text area but also the edge of the background image.

3. At this point you should also add a small amount of padding to the right side of the #maincontent, as shown in Figure 7-39.



**Figure 7-39.** You built padding into the left side when you created the main content left margin. Adding padding here moves the text away from the right border.

4. Add padding to the right and left side of #footer, as shown in Figure 7-40.



**Figure 7-40.** Text always looks better when there is some space between the text and the border.

Don't forget to preview your finished web page in a variety of web browsers. You might notice that there are differences in how the masthead appears. Some of these differences are because of the different way that border inset and border outset are rendered. There is nothing you can do to change how that style border displays, but you can remove the extra space at the top of Firefox by adding `margin-top: 0` to the `h1` definition.

---

## Summary

Over the course of your CSS exercises you have taken a plain XHTML outline and turned it into an attractive web page using modern CSS presentation. Using older-style table-based layouts with `<font>` elements would have increased the size of your page.

In the following chapters, as you finalize your web page layout, going from one web page to a full website, you will be taking the CSS you created and moving it from the `<head>` section of your page to an external stylesheet to keep your web pages consistent within your site and take advantage of browser caching to speed up page load.