FAQ

How do I create a custom-color scroll bar?

It can be fun to color-coordinate the scroll bar with your Web site! Keep in mind that not all your Web visitors will see your handiwork. While this effect is supported by Internet Explorer, it is not supported by all browsers. To configure a scroll bar with colors that you choose, add the following styles to the body tag: `scrollbar-face-color`, `scrollbar-arrow-color`, `scrollbar-track-color`. For example:

```
body {  scrollbar-face-color: #cc66cc;
        scrollbar-arrow-color: #006600;
        scrollbar-track-color: #cccccc;
}
```

Note: Your CSS will not pass W3C validation tests if you use these Internet Explorer only properties.

6.6 CSS Debugging Tips

Using CSS for page layout requires some patience. It takes a while to get used to it. One of the biggest issues is that even modern browsers implement CSS in slightly different ways. Testing is crucial. Don't make it your goal that the pages must look exactly the same on every browser. Expect your pages to look slightly different on various browsers. Design so they look best on the most commonly used browser (currently Internet Explorer) and display acceptably well on other browsers. There are Web pages devoted to CSS bugs and browser support of CSS. The following are a few that you will find helpful:

- http://web.archive.org//20040202153928/http://devedge.netscape.com/library/xref/2003/css-support/css1/mastergrid.html

  The original "Master List" created by Eric Meyer
- http://www.westciv.com/style_master/academy/browser_support/index.html

  A comprehensive browser compatibility list
- http://www.positioniseverything.net

  John and Holly Bergevin's site focuses on CSS bugs in modern browsers—it contains some great sample CSS page layouts
- http://www.quirksmode.org

  Peter-Paul Koch's site is dedicated to studying and defeating browser incompatibilities related to CSS and JavaScript

CSS Debugging Techniques

Debugging CSS can be frustrating. The following are helpful techniques to use:

- **Manually Check Syntax Errors.** Sometimes a CSS style does not apply due to a syntax error. Carefully check your code. Many times the error is in the line above the style that is not correctly applied.
- **Programmatically Check Syntax Errors.** As mentioned earlier, you can use the W3C's CSS Validator at http://jigsaw.w3.org/css-validator to verify your syntax.

- **Configure Temporary Background Colors.** Sometimes your code is valid but the browser window is not configured the way you would expect. If you temporarily assign distinctive background colors such as red or yellow and test again, it should be easier to see where the "boxes" are ending up.
- **Configure Temporary Borders.** Similar to the temporary background colors, you could temporarily configure an element with a 3 pixel red solid border—this will really jump out at you and help you recognize the issue sooner.
- **Use Comments to Find the Unexpected Cascade.** Styles and XHTML attributes configured farther down the page can override earlier styles. If your styles are misbehaving, try commenting out (see below) some styles and test with a smaller group of statements. Then add the styles back in one by one to see where or when the breakdown occurs. Work patiently and test the entire style sheet in this manner.

Note that Comment Areas Are Ignored by Browsers. A style sheet comment begins with `/*` and ends with `*/`. Comments can span multiple lines. A code snippet with CSS comments follows:

```
/* Set Page Margins to Zero */
body { margin: 0
}
/* temporarily commented out during testing
.nav { text-decoration: none;
}
*/
```

The first comment is used to document the style sheet and describe the style applied to the body tag. The second comment spans multiple lines. It begins on the line above the `nav` class and ends on the line below the `nav` class. This causes the browser to skip the `nav` class when applying the style sheet. This technique can be useful in testing when you are experimenting with a number of properties.

6.7 CSS Page Layout Resources

This chapter introduces you to using CSS for page layout configuration and should get you started in your exploration of this technology. It may help you to know that you are not alone in your quest to learn CSS. There are many resources with documentation, tutorials, and support for this technology. The page layout techniques discussed in this textbook are just an introduction to using this technology. There are many Web sites that offer additional insight and techniques for configuring page layout with CSS. The following are a few that you may find useful:

- http://glish.com/css

  Large collection of CSS page layouts and links to tutorials
- http://www.websitetips.com/css/index.shtml

  Comprehensive list of tutorials and CSS-related sites
- http://www.meyerweb.com/eric/css

  The site of Eric Meyer, a leading-edge Web developer

- http://www.w3.org/Style/CSS/learning
  W3C's list of CSS resources
- http://www.bluerobot.com/web/layouts
  A "reservoir" of CSS page layouts
- http://www.blooberry.com/indexdot/css
  CSS syntax reference list
- http://www.w3.org/TR/1998/REC-CSS2-19980512
  W3C CSS Level 2 Recommendation
- http://www.w3.org/TR/REC-CSS1-961217.html
  W3C CSS Level 1 Recommendation

CHECKPOINT 6.2

1. The two-column layouts you created in the previous Hands-On Practice did not use absolute positioning. Open the twocolumn.html page in a browser. Resize the browser window. Describe what happens. What type of page design layout (ice, jello, or liquid) is being used?
2. Describe one CSS debugging tip that you have found helpful.
3. Describe how to choose whether to configure an XHTML tag, create a class, or create an id when working with CSS.

CHAPTER SUMMARY

This chapter introduced Cascading Style Sheet rules associated with page layout. Techniques for positioning and floating elements and configuring two-column page layouts were demonstrated. This topic is very deep and you have much to explore. Visit the resources in the chapter to continue learning about this technology.

Visit the textbook Web site at http://www.webdevfoundations.net for examples, the links listed in this chapter, and updated information.

Key Terms

absolute positioning
border
`clear` property
content
CSS Box Model
CSS-P
`display` property
`float` property
`left` property
margin
normal flow
padding
relative positioning
`right` property
`top` property
visible width
`z-index` property

Review Questions

Multiple Choice

1. When using absolute positioning, which of the following properties may be used to determine the placement of the element?
   a. `top` and `left`
   b. `z-index` and `display`
   c. `float` and `clear`
   d. none of the above

2. Which of the following, from outermost to innermost, are components of the box model?
   a. margin, border, padding, content
   b. content, padding, border, margin
   c. content, margin, padding, border
   d. margin, padding, border, content

3. Which of the following is the default value of the `border` and `padding` properties for an element?
   a. 1 pixel
   b. 0 pixels
   c. 3 pixels
   d. 10 pixels

4. Which of the following configures a class called `sidebar` to float to the right?
   a. `.sidebar { right: float; }`
   b. `.sidebar { float: right; }`
   c. `.sidebar { float-right: 200px; }`
   d. none of the above

5. Which of the following is the rendering flow used by a browser by default?
   a. XHTML flow
   b. normal display
   c. browser flow
   d. normal flow

6. Which of the following is used to change the location of an element slightly in relation to where it would otherwise appear on the page?
   a. the `float` property
   b. absolute positioning
   c. relative positioning
   d. this cannot be done with CSS

7. Which of the following will configure padding that is 10 pixels on the top, 0 pixels on the left and right, and 5 pixels on the bottom?

   a. `margin: 0px 5px 0px 10px;`
   b. `margin:top-10, left-0, right-0, bottom-5;`
   c. `padding: 10px 0 5px 0;`
   d. none of the above

8. Which of the following is used along with the `left`, `right` and/or `top` property to configure the position of an element precisely?

   a. `position: relative`
   b. `position: absolute`
   c. `position: float`
   d. none of the above

9. Which of the following configures a margin for an element with the following values: top margin 20 pixels, left margin 300 pixels, right margin 0 pixels, and bottom margin 0 pixels?

   a. `margin: 300px 20px 0px 300px;`
   b. `margin: top-20, left-300, right-0, bottom-0;`
   c. `margin: 20px 0 0 300px;`
   d. none of the above

10. Which of the following will configure a class called news to stack on top of other elements that have a `z-index` of 5?

    a. `.news { z-index: high} ;`
    b. `.news { z-index:6} ;`
    c. `.news { z-index:4} ;`
    d. `none of the above`

Fill in the Blank

11. Configure a style with a(n) _____ if the style will only apply to one element on a page.

12. If an element is configured with `float:right`, the other content on the page will appear to its _____.

13. The _____ is always transparent.

14. Use the `position:relative` property along with the _____, _____, and/or _____ property to configure the position of an element in relation to the normal flow.

15. Configure a style with a_____ if the style could apply to more than one element on a page.

Apply Your Knowledge

1. **Predict the Result.** Draw and write a brief description of the Web page that will be created with the following XHTML code:

```
<html>
<head>
  <title>CircleSoft Web Design</title>
  <style type="text/css">
  h1 { border-bottom: 1px groove #333333;
       color: #006600;
       background-color: #cccccc
  }
  #content { position: absolute;
             left: 200px;
             top: 75px;
             font-family: Arial,sans-serif;
             width: 300px;
  }
  .nav { font-weight: bold;
  }
  </style>
</head>
```

```
<body>
  <h1>CircleSoft Web Design</h1>
  <div id="content">
    <p>Our professional staff takes pride in its working
relationship with our clients by offering personalized services
that listen to their needs, develop their target areas, and
incorporate these items into a well-presented Web site that
works.</p>
  </div>
  <ul>
    <li class="nav">Home</li>
    <li class="nav"><a href="about.html">About</a></li>
    <li class="nav"><a href="services.html">Services</a></li>
  </ul>
  </div>
</body>
</html>
```

2. Fill in the Missing Code. This Web page should be configured as a two-column page layout with a right column 150 pixels wide. The right column should have a 1 pixel border. The padding in the left column content area needs to allow for the room that will be used by the right column. Some CSS properties and values, indicated by "_", are missing. Fill in the missing code.

```
<?xml version="1.0" encoding="UTF-8"?>
<!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>
<title>Trillium Media Design</title>
<style type="text/css">
body { margin: 0;
       font-family: Verdana, Arial, sans-serif;
}
#rightcolumn { "_":"_";
               width: "_";
               background-color: #cccccc;
               height: 400px;
               border: "_";
}
#leftcolumn {
}
#logo { background-color: #cccccc;
        color: #663333;
        font-size: x-large;
        border-bottom: 1px solid #333333;
}
.content { padding: "_";
}
```

```
.footer { font-size: xx-small;
          text-align: center;
          clear: "_";
}
.navBar { color: #000066;
          text-decoration: none;
          padding: 3px;
          margin: 15px;
          display: "_";
}
</style>
</head>
<body>
<div id="rightcolumn">
  <a class="navBar" href="index.html">Home</a>
  <a class="navBar" href="products.html">Products</a>
  <a class="navBar" href="services.html">Services</a>
  <a class="navBar" href="about.html">About</a>
</div>
<div id="leftcolumn">
  <div id="logo">
    <h1>Trillium Media Design</h1>
  </div>
  <div class="content">
    <p>Our professional staff takes pride in its working
relationship with our clients by offering personalized services
that listen to their needs, develop their target  areas, and
incorporate these items into a well-presented Web site that
works.</p>
  </div>
  <div class="footer">
    Copyright &copy; 2008 Trillium Media Design<br />
    Last Updated on 01/15/08
  </div>
</div>
</body>
</html>
```

3. **Find the Error.** When this page is displayed using Internet Explorer 7, the heading information obscures the paragraph text. Correct the errors and describe the process you followed.

```
<?xml version="1.0" encoding="UTF-8"?>
<!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>
<title>CSS Float</title>
<style type="text/css">
```

```
h1 { background-color: #eeeeee;
     padding: 5px;
     color: #666633;
     position: absolute;
     left: 200px;
     top: 20px;
}
p { font-family: Arial,sans-serif;
    position;
    absolute;
    left: 100px;
    top: 100px;
}
#yls { float:right;
       margin: 0 0 5px 5px;
       border: solid;
}
</style>
</head>
<body>
<img id="yls" src="yls.jpg" alt="Yellow Lady Slipper" height="100"
  width="100" />
<h1>Floating an Image</h1>
<p> The Yellow Lady Slipper pictured on the right is a wildflower.
It grows in wooded areas and blooms in June each year. The Yellow
Lady Slipper is a member of the orchid family.</p>
</body>
</html>
```

Hands-On Exercises

1. Write the CSS for a class with the following attributes: a light blue background color, Arial or sans-serif font, dark blue text color, 10 pixels of padding, and a narrow dashed border in a dark blue color.
2. Write the CSS for an id with the following attributes: float to the left of the page, light beige background, Verdana or sans-serif large font, and 20 pixels of padding.
3. Write the CSS to configure a class that will produce a headline with a dotted line underneath it. Choose a color that you like for the text and dotted line.
4. Write the CSS for an id that will be absolutely positioned on a page 20 pixels from the top and 40 pixels from the right. This area should have a light gray background and a solid border.
5. Write the CSS for a class that is relatively positioned. This class should appear 15 pixels in from the left. Configure the class to have a light green background.
6. **Extending Hands-On Practice 6.6.** Design a two-column page layout with the navigation on the right side. Use the twocolumn.html file from Hands-On Practice 6.6 as a starting point. This file is in the Chapter6/wildflowers2folder in the student files.

Code an external style sheet file called rightcolumn.css and a Web page called rightcolumn.html. The Web page should have two columns. The right column will be the navigation column and the left column will be the content column. Hand in printouts of rightcolumn.css, the rightcolumn.html source code (print in Notepad), and the browser display of your rightcolumn.html to your instructor.

7. **Extending Hands-On Practice 6.6.** Design a two-column page layout with a logo area across the top. Review the Hands-On Practice 6.6 twocolumn.html file for some examples. These files are in the Chapter6/wildflowers2 folder in the student files. Code an external style sheet file called mydesign.css and a Web page called mydesign.html. The Web page should have two columns and a logo area across the top. Hand in printouts of mydesign.css, the mydesign.html source code (print in Notepad), and the browser display of your mydesign.html to your instructor.

8. **Extending Hands-On Practice 6.6.** In Hands-On Practice 6.6 you created two files for a version of the Door County Wildflowers Web site. The files are available in the Chapter6/wildflowers2 folder in the student files. You will create two additional content pages for the Door County Wildflowers site, called spring.html and summer.html, in this exercise. Be sure that all CSS is placed in an external style sheet, called mywildflower.css. (Modify pre-existing pages to use this style sheet.) Here is some content to include on the new pages:

   Spring Page (spring.html):

   - Use the trillium.jpg image (see the Chapter6 folder in the student files).
   - Trillium facts: 8–18 inches tall, perennial, native plant, grows in rich moist deciduous woodlands, white flowers turn pink with age, fruit is a single red berry, protected flower species.

   Summer Page (summer.html):

   - Use the yls.jpg image (see the Chapter6 folder in the student files).
   - Yellow Lady's Slipper facts: 4–24 inches tall, perennial, native plant, grows in wet shaded deciduous woods, swamps, and bogs, an orchid, official flower of Door County.

   Hand in printouts of mywildflower.css, spring.html source code (print in Notepad), summer.html source code, the browser display of spring.html, and the browser display of summer.html to your instructor.

9. **Extending Hands-On Practice 6.2.** Modify the twocolumn.html page you created in Hands-On Practice 6.2. This file is in the Chapter6/wildflowers2 folder in the student files. Recall from Chapter 5 that a Web page using jello design has content in the center of the Web page with blank margins on either side. You can code this using CSS by configuring the margin property of the body tag to use percentages for the left and right. For example:

   ```
   body { margin: 0 10% 0 10%; }
   ```

   Hand in printouts of the source code (print in Notepad) and browser display for the Web page to your instructor.

10. Design a splash page called moviesplash.html about your favorite movie. Use absolute positioning and `z-index` to create an interesting display. First sketch the areas for images, text, and link to the first page on the site. Search the Web for

photos of the movie. Next, locate images from the movie. When you code your page use embedded CSS unless your instructor directs you otherwise. Hand in printouts of the moviesplash.html source code (print in Notepad), and the browser display of moviesplash.html to your instructor.


Focus on Ethics

(*Note*: It is unethical to steal an image from another Web site. Some Web sites have a link to their copyright policy. Most Web sites will give permission for you to use an image in a school assignment. If there is no available policy, e-mail the site's contact person and request permission to use the photo. If you are unable to obtain permission, you may substitute clip art or an image from a free site.)

Web Research

This chapter introduced using CSS to configure Web page layout. Use the resources listed in the text as a starting point. You can also use a search engine to search for CSS resources.

Create a Web page that provides a list of at least five CSS resources on the Web. For each CSS resource provide the URL, Web site name, and a brief description. Your Web page should use absolute positioning. Print both the source code (from Notepad) and the browser view of your Web page.

Focus on Web Design

There is still much for you to learn about CSS. A great place to learn about Web technology is on the Web itself. Use a search engine to search for CSS page layout tutorials. Choose a tutorial that is easy to read. Select a section that discusses a CSS technique that was not covered in this chapter. Create a Web page that uses this new technique. Consider how the suggested page layout follows (or does not follow) principles of design such as contrast, repetition, alignment, and proximity (see Chapter 5). The Web page should provide the URL of your tutorial, the name of the Web site, a description of the new technique you discovered, and a discussion of how the technique follows (or does not follow) principles of design. Print the external style sheet (if you used one), the Web page source code (from Notepad), and the browser view of your Web page.

WEB SITE CASE STUDY

Implementing CSS Two-Column Page Layout

Each of the following case studies continues throughout most of the text. This chapter implements CSS two-column page layout in the Web sites.

JavaJam Coffee House

See Chapter 2 for an introduction to the JavaJam Coffee House case. Figure 2.26 shows a site map for the JavaJam Web site. The pages were created in earlier chapters. In this

case study you will implement a new two-column CSS page layout for JavaJam. You will modify the external style sheet and the Home, Menu, Music, and Jobs pages. Unless your instructor directs you otherwise, use the Chapter 4 JavaJam Web site as a starting point for this case study.

Figure 6.22 displays a wireframe for the two-column page layout with a page container, logo, left column, navigation, right column, floating, and footer areas.

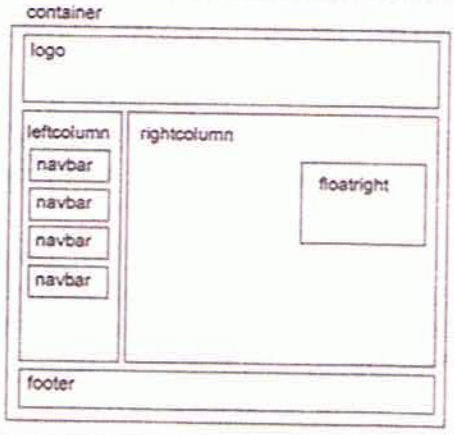


Figure 6.22
JavaJam two-column page layout

Hands-On Practice Case

1. **Create a Folder.** Create a folder called javajamcss. Copy all the files from your Chapter 4 javajam folder into the javajamcss folder. You will modify the javajam.css file and each Web page file (index.html, menu.html, and music.html) to implement the two-column page layout shown in Figure 6.22. See the new JavaJam Home page, as shown in Figure 6.23 (shown also in the color insert section).
2. **Configure the CSS.** Open javajam.css in Notepad. Edit the style rules as follows:
   - Modify the `container` id to have a minimum width of 700 pixels (use `min-width: 700px`), background (`#e8d882`) and text (`#000000`) color, 80% width, and a 2 pixel black double border (`border: 2px double #000000`).
   - Configure the logo area. Remove the `h1` selector and style rules. Create a new id named `logo` with a background (`#ccaa66`) and text (`#000000`) color, center alignment (`text-align: center`) and a bottom border that is 2 pixels, double, and black (`border-bottom: 2px double #000000`).
   - Configure the left column area. Add a new style rule for the `leftcolumn` id to configure an area that floats to the left, is 100 pixels wide, and has 10 pixels of padding on the top side.

```
#leftcolumn { float: left;
              width: 100px;
              padding-top: 10px;
}
```

   - Configure the navigation area. Remove the `nav` id. Add a new style rule for the `navbar` class to configure an area with no underlines on hyperlinks, a 15

Figure 6.23
The new JavaJam two-column index.html

See the center color insert

pixel margin, and is displayed as a block element (with line breaks above and below) by the browser.

```
.navBar{ text-decoration: none;
         margin: 15px;
         display: block;
}
```

- Configure the right column area. Add a new style rule for the `rightcolumn` id to configure an area with a 150 pixel left margin, background (`#f1e8b0`) and text (`#000000`) color, and 10 pixels of padding.

```
#rightcolumn { margin-left: 150px;
               background-color: #f1e8b0;
               color: #000000;
               padding: 10px;
}
```

- Configure an area that floats to the right. Notice how the winding road graphic shown in Figure 6.23 floats on the right side—this is configured with the `floatright` class. Images are more compelling when separated from other elements (such as text) by empty space. Add 40 pixels of padding to the left side of this area.

```
.floatright { padding-left: 40px;
              float: right;
}
```

- Modify the `footer` id to display a 2 pixel double black top border (`border-top: 2px double #000000`).

Save the javajam.css file.

3. **Modify the index.html File.** Add `<div>` elements and modify the code as follows:
   - Configure the logo area. Remove the `<h1>` opening and closing tags surrounding the JavaJam logo image. Code a `<div>` that surrounds the JavaJam logo image element. Assign the `<div>` to the id `logo`.
   - Configure the left column. The navigation links are the only content in the left column. Change `<div id="nav">` to `<div id="leftcolumn">`. Assign each anchor element in this area to the `navbar` class (`class="navbar"`).
   - Configure the right column. This area contains the remaining page content, including the footer area. Add a `<div>` that surrounds the text, winding road image, and footer area. Assign the `<div>` to the id `rightcolumn`.
   - Configure the area that floats to the right. Modify the winding road image element. Remove the `align="right"` attribute and add `class="floatright"` to the winding road image element.

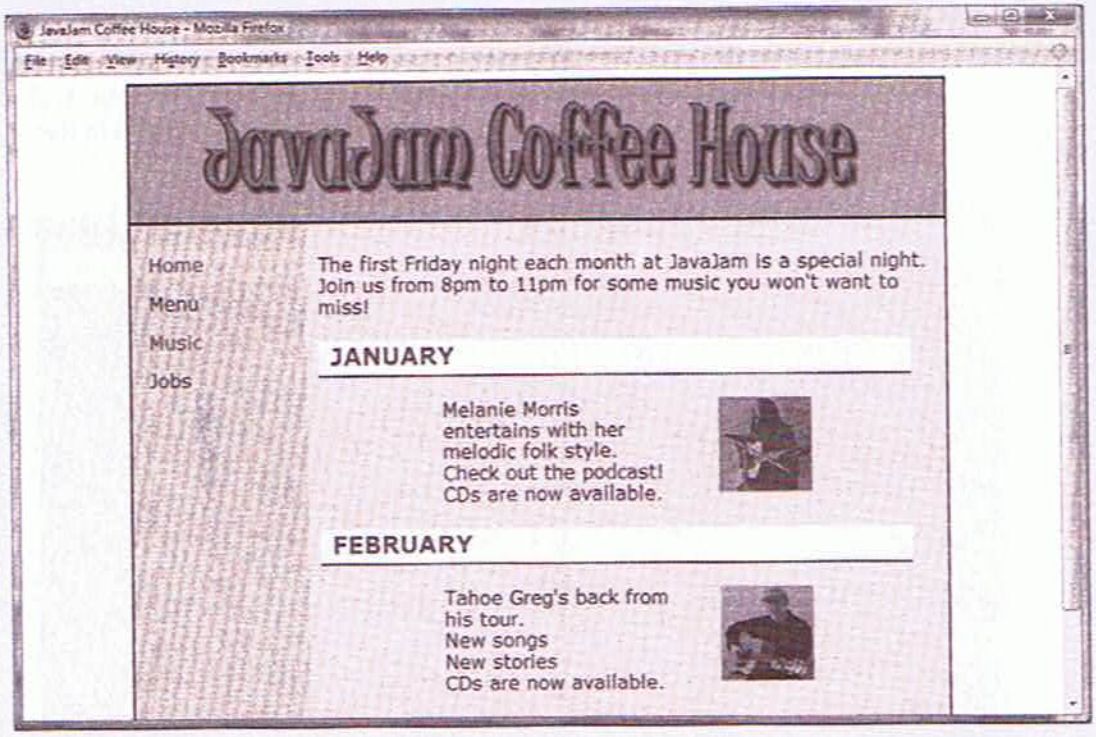
   Save the index.html file. It should look similar to the Web page shown in Figure 6.23. Remember that validating your XHTML and CSS can help you find syntax errors. Test and correct this page before you continue.
4. **Modify the menu.html and music.html Files.** Modify the code in these Web page files in a similar manner as you did in Step 3. Save and test your pages in a browser. As you test your pages, use the CSS and XHTML validators to help you find syntax errors.
5. **Bonus Style: `text-transform`.** Figure 6.24 shows an alternate design for the music.html page. Notice how the `<h3>` elements are styled differently—with all uppercase text (using a new property, `text-transform`) different background and text colors, font size, bottom border, and margin. Open javajam.css in a text editor and replace the `h3` selector style rules with the following:

```
h3 { text-transform: uppercase;
     background-color: #ffffcc;
     color: #663300;
     font-size: 20px;
     border-bottom: 1px solid #000000;
     margin-right: 20px;
}
```

   Save the javajam.css file. Test your pages in a browser. Your music.html page should look similar to the one shown in Figure 6.23. The other pages do not use `<h3>` elements and should appear as they did at the end of Step 4.

In this case study you changed the page layout of the JavaJam Web site pages. Notice that with just a few changes in the CSS and XHTML code, you configured a two-column page layout.

Figure 6.24
New style rules for the h3 selectors

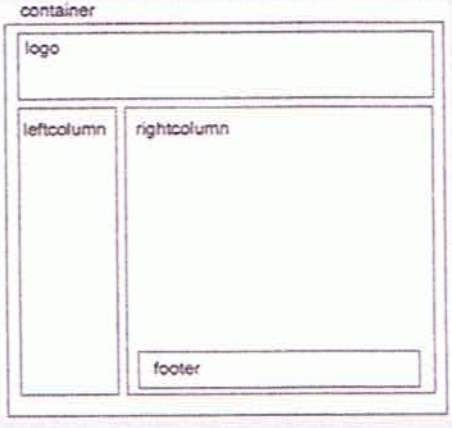


Fish Creek Animal Hospital

See Chapter 2 for an introduction to the Fish Creek Animal Hospital Case Study. Figure 2.30 shows a site map for the Fish Creek Web site. The pages were created in earlier chapters. In this case study you will implement a new two-column CSS page layout. You will modify the external style sheet and the Home, Services, and Ask the Vet pages. Unless your instructor directs you otherwise, use the Chapter 4 Fish Creek Web site as a starting point for this case study.

Figure 6.25 displays a wireframe for the two-column page layout with a page container, logo, left column, navigation, right column, and footer areas.

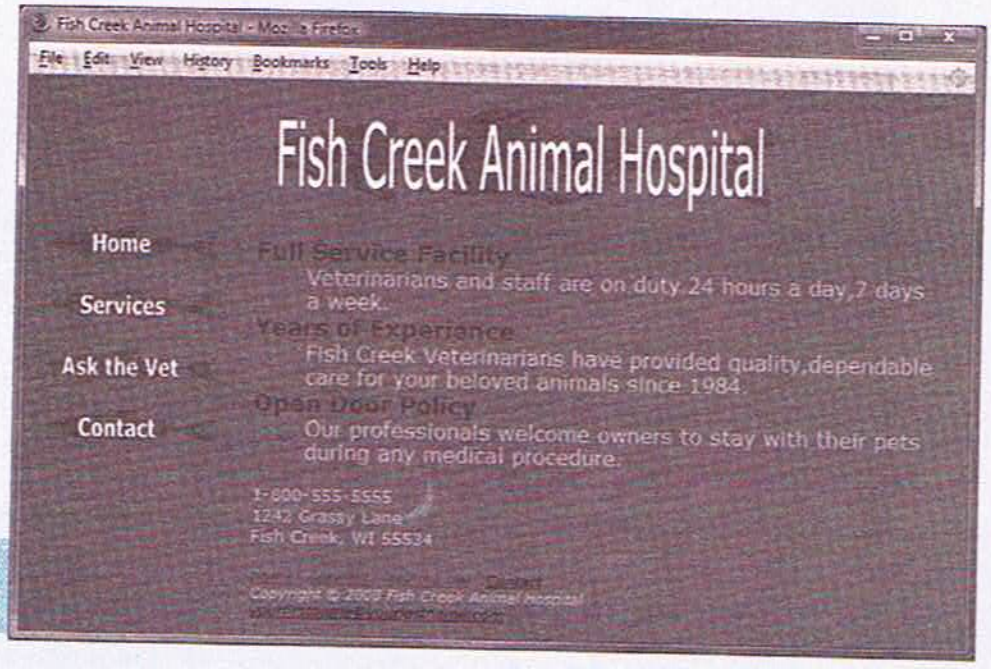
Figure 6.25
Fish Creek two-column page layout



Hands-On Practice Case

1. **Create a Folder.** Create a folder called fishcreekcss. Copy all the files from your Chapter 4 fishcreek folder into the fishcreekcss folder. You will modify the fishcreek.css file and each Web page file (index.html, services.html, and askvet.html) to implement the two-column page layout, as shown in Figure 6.25. See the new Fish Creek home page, as shown in Figure 6.26 (shown also in the color insert section).

Figure 6.26
The new Fish Creek two-column index.html



See the center color insert

2. **Configure the CSS.** Open fishcreek.css in Notepad. Edit the style rules as follows:
   - Modify the container id style rules. Notice that the new page layout aligns with the left margin. Remove the properties previously used to center the content (`margin-left` and `margin-right`).
   - Configure the logo area. Remove the `h1` selector and style rules. Create a new id named `logo` with 70 pixels of padding on the left side (`padding-left: 70px`).
   - Configure the left column area. Add a new style rule for the `leftcolumn` id to configure an area that floats to the left and is 140 pixels wide.

```
#leftcolumn { float: left;
              width: 140px;
}
```

   - Configure the navigation area. Remove the `imgnav` id. Add a property to the style rules for the `img` selector—configure 10 pixels of padding (`padding: 10px`).
   - Configure the right column area. Add a new style rule for the `rightcolumn` id to configure an area with a 170 pixel left margin.
   - Configure the footer area. Remove the `nav` id.

   Save the fishcreek.css file.

3. **Modify the index.html File.** Add `<div>` elements and modify the code as follows:
   - Configure the logo area. Remove the `<h1>` opening and closing tags surrounding the Fish Creek image. Code a `<div>` that surrounds the Fish Creek logo image element. Assign the `<div>` to the id `logo`.
   - Configure the left column. The navigation image links are the only content in the left column. Change `<div id="imgnav">` to `<div id="leftcolumn">`.
   - Configure the right column. This area contains the definition list and the paragraph with the contact information. Code a `<div>` that surrounds this area. Assign the `<div>` to the id `rightcolumn`.
   - Configure the page footer area. You need to adjust the starting location of the `footer` id. Locate `<div id="footer">` in the code and remove the assignment to the id from the . Next, change `<div id="nav">` to `<div id="footer">`. The area assigned to the `footer` id now includes the text navigation, copyright information, and e-mail link.

   Save the index.html file. It should look similar to the Web page shown in Figure 6.26. Remember that validating your XHTML and CSS can help you find syntax errors. Test and correct this page before you continue.
4. **Modify the services.html and askvet.html Files.** Modify these Web page files in a similar manner as you did in Step 3. Save and test your pages in a browser. As you test your pages, use the CSS and XHTML validators to help you find syntax errors.

In this case study you changed the page layout of the Fish Creek Web site pages. Notice that with just a few changes in the CSS and XHTML code, you configured a two-column page layout.

Pete the Painter

See Chapter 2 for an introduction to the Pete the Painter Case Study. Figure 2.34 shows a site map for the Pete the Painter Web site. The pages were created in earlier chapters. In this case study you will implement a new two column CSS page layout for Pete the Painter. You will modify the external style sheet and of the Home, Services, and Testimonials, pages. Unless your instructor directs you otherwise, use the Chapter 4 Pete the Painter Web site as a starting point for this case study.

Figure 6.27 displays a wireframe for the two-column page layout with a page container, logo, left column, navigation, right column, and footer areas.

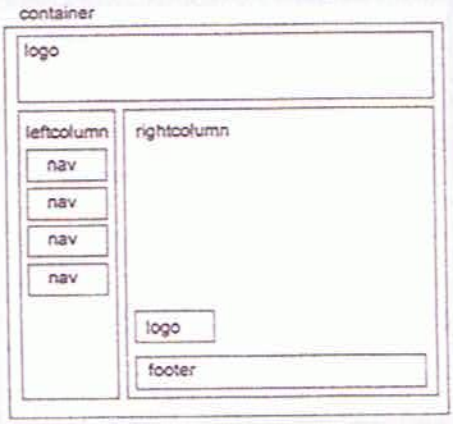


Figure 6.27 Pete the Painter two-column page layout

Hands-On Practice Case

1. **Create a Folder.** Create a folder called paintercss. Copy all the files from your Chapter 4 folder into the paintercss folder. You will modify the painter.css file and each Web page file (index.html, services.html, and testimonials.html) to implement the two-column page layout shown in Figure 6.27. See the new Pete the Painter home page, as shown in Figure 6.28 (shown also in the color insert section).

Figure 6.28
The new Pete the Painter two-column index.html

See the center color insert

2. **Configure the CSS.** Open painter.css in Notepad. Edit the style rules as follows:
   - Configure the left column area. Add a new style rule for the `leftcolumn` id to configure an area that floats to the left and is 150 pixels wide.

```
#leftcolumn { float: left;
              width: 150px;
}
```

   - Configure the navigation area. Remove the `nav` id. Add a new style rule for the `nav` class to configure an area with bold font, a 15 pixel margin on the right, bottom, and left sides, and is displayed as a block element (with line breaks above and below) by the browser.

```
.nav { font-weight: bold;
       margin: 0 15px 15px 15px;
       display: block;
}
```

   - Configure the right column area. Add a new style rule for the `rightcolumn` id to configure an area with a 150 pixel left margin and a 10 pixel top margin (`margin: 10px 0 0 150px`).
   - Configure an area that floats to the right. Add a new style rule for the `floatright` class.

```
.floatright { float: right;
}
```

   Save the painter.css file.

3. **Modify the index.html File.** Add `<div>` elements and modify the code as follows:
   - Configure the logo area. Remove the `<h1>` opening and closing tags surrounding the Pete the Painter image. Code a `<div>` that surrounds the Painter logo image element. Assign the `<div>` to the id `logo`.
   - Configure the left column. The navigation links are the only content in the left column. Change `<div id="nav">` to `<div id="leftcolumn">`. Assign each anchor element in this area to the `nav` class (`class="nav"`).
   - Configure the right column. This area contains the content (paragraph, unordered list, and heading 3 elements) and the footer section. Code a `<div>` that surrounds this area. Assign the `<div>` to the `rightcolumn` id.
   - Assign the `<h3>` element to the `logo` class.

   Save the index.html file. It should look similar to the Web page shown in Figure 6.28. Remember that validating your XHTML and CSS can help you find syntax errors. Test and correct this page before you continue.

4. **Modify the services.html and testimonials.html Files.** Modify these Web page files in a similar manner as you did in Step 3. Configure the room images on the testimonials.html page—on the opening image tag for each room photo, remove the `align="right"` attribute and add `class="floatright"`.

5. **Save and Test Your Pages in a Browser.** As you test your pages, use the CSS and XHTML validators to help you find syntax errors.

6. **Bonus Style.** Figure 6.29 shows an alternate design for the testimonials.html page. Notice how the `<h4>` elements are styled differently—it is set to float and is configured with a *negative* top margin. This allows the dark green box to stand out better on the page. The paragraph elements in this area are each assigned to a class that configures a top border of the same color green and extra padding.

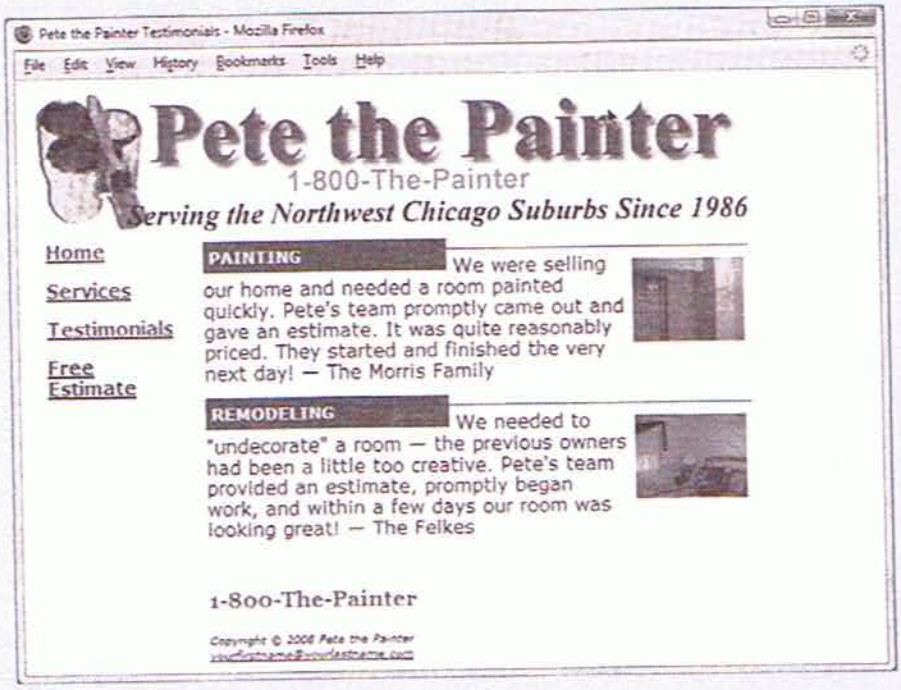


Figure 6.29
New style rules for the `h4` selector

Open painter.css in a text editor and update the style rules with the following:

```
h4 { margin: -10px 5px 5px 0px;
     padding: 5px;
     font-family: Verdana, sans-serif;
     font-size: 14px;
     float: left;
     width: 200px;
     background-color: #336600;
     color: #ffffff;
     text-transform: uppercase;
}
.desc { padding: 5px 0 20px 0;
        border-top: 1px solid #336600;
}
```

Save the painter.css file. Test your pages in a browser. Your testimonials.html page should be similar to the one shown in Figure 6.29. The other pages should display as they did at the end of Step 4. Consider modifying the unordered list on the Services page (services.html) to use `<h4>` and `<p>` elements (assigned to the `desc` class) instead—the result will be a more cohesive design for your Web site.

In this case study you changed the page layout of the Pete the Painter Web site pages. Notice that with just a few changes in the CSS and XHTML code, you configured a two-column page layout.

Prime Properties

See Chapter 2 for an introduction to the Prime Properties Case Study. Figure 2.38 shows a site map for the Prime Properties Web site. The pages were created in earlier chapters. In this case study you will implement a new two-column CSS page layout for Prime Properties. You will modify the external style sheet and the Home, Listings, and Financing pages. Unless your instructor directs you otherwise, use the Chapter 4 Prime Properties Web site as a starting point for this case study.

Figure 6.30 displays a wireframe for the two-column page layout with a page wrapper, logo, left column, navigation, right column, and footer areas.

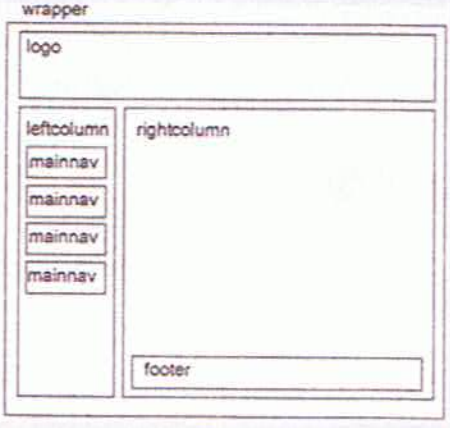


Figure 6.30 Prime Properties two-column page layout

Hands-On Practice Case

1. **Create a Folder.** Create a folder called primecss. Copy all the files from your Chapter 4 prime folder into the primecss folder. You will modify the prime.css file and each Web page file (index.html, listings.html, and financing.html) to implement the two-column page layout shown in Figure 6.30. See the new Prime Properties Home page, as shown in Figure 6.31 (shown also in the color insert section).

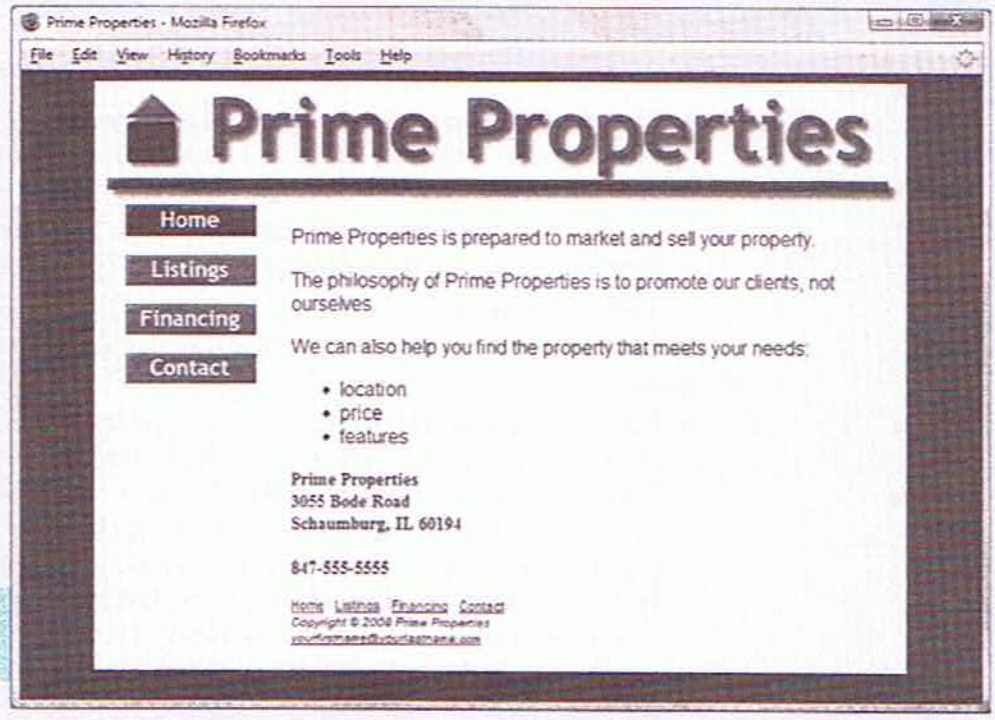


Figure 6.31 The new Prime Properties two-column index.html

See the center color insert

2. **Configure the CSS.** Open prime.css in Notepad. Edit the style rules as follows:
   - Configure the page background color. Modify the style rules for the `body` selector. Set the `background-color` property to `#003366`. Set the text color to `#ffffcc`.
   - Create a new `wrapper` id to contain the page content. Configure the area with a width of 680 pixels and centered (`margin:0 auto`). Set the minimum width to 680 pixels, also. Configure the background (`#ffffcc`) and text (`#003300`) colors. Set the left padding to 10 pixels.

```
#wrapper { width: 680px;
           min-width: 680px;
           margin: 0 auto;
           background-color: #ffffcc;
           color: #003300
           padding-left: 10px;
}
```

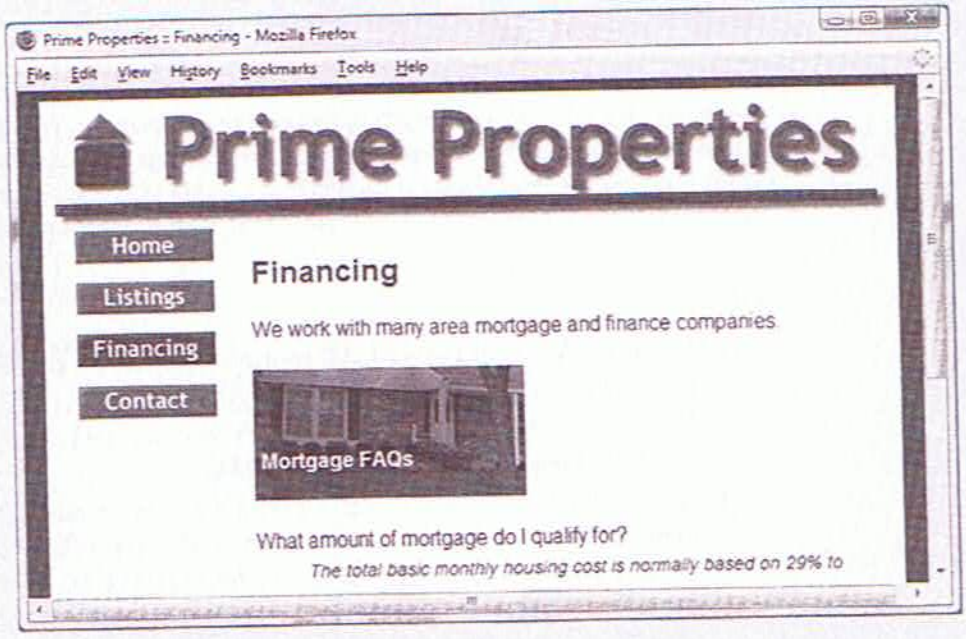


Figure 6.32
New style rules for the Mortgage FAQs heading

CSS creates the effect. Styles are declared for the `home` class with the following properties: a background image that does not repeat and is positioned very carefully, a width of 200 pixels, light text, and generous padding.

```
.home { background-image: url(schaumburg.jpg);
        background-position: -100px -260px;
        background-repeat: no-repeat;
        color: #ffffcc;
        padding: 60px 5px 20px 5px;
        width: 200px;
}
```

Notice that the `background-position` property is used with carefully chosen values. As indicated in Table 6.1, the `background-position` property can use two numeric pixel values—horizontal and vertical. The *negative numbers* cause the background image to shift 100 pixels to the left and 260 pixels down from the top. The effect is somewhat abstract and ends up displaying just part of the image. In this case, it is a part of a house to tie in with the home financing theme. Padding is set quite high to allow room for the image to display around the text. The text color is light in order to contrast well with the image.

Save the prime.css file. Modify the financing.html page as indicated above—add `class="home"` to the `<h4>` element. Save the financing.html file. Test your pages in a browser. Your financing.html page should look similar to the one shown in Figure 6.32. The other pages should display as they did at the end of Step 4.

In this case study you changed the page layout of the Prime Properties Web site pages. Notice that with just a few changes in the CSS and XHTML code, you configured a two-column page layout.

Web Project

See Chapter 5 for an introduction to the Web Project case. As you completed the Chapter 5 Web Project Case Study activities you completed a Web Project Topic Approval, Web Project Site Map, and Web Project Page Layout Design. In this case study you will use your design documents as a guide as you develop the pages for your Web Project using CSS (external style sheet) for both formatting and page layout.

Hands-On Practice Case

1. Create a folder called project. All your project files and graphics will be organized in this folder and subfolders as needed.
2. Refer to your Site Map to view the pages that you need to create. Jot down a list of the file names. Add these to the Site Map.
3. Refer to the Page Layout Design. Make a list of the common fonts and colors used on the pages. These may become the CSS you configure for the body element. Note where typical elements used for organization (such as headings, lists, paragraphs, and so on) may be used. You may want to configure CSS for these elements. Identify various page areas such as logo, navigation, footer, and so on—and list any special configurations needed for these areas. These will be configured as classes in your CSS. Create an external style sheet, called project.css, which contains these configurations.
4. Using your design documents as a guide, code a representative page for your site. Use CSS to format text, color, and layout. Be sure to apply classes and ids where appropriate. Associate the Web page to the external style sheet.

   Save and test the page. Modify both the Web page and the project.css file as needed. Test and modify until you have achieved the look you want.
5. Using the completed page as a template wherever possible, code the rest of the pages on your site. Test and modify them as needed.
6. Experiment with modifying the project.css file. Change the page background color, the font family, and so on. Test your pages in a browser. Notice how a change in a single file can affect multiple files when external style sheets are used.