CHAPTER 7

More on Links, Lists, and Layout

Chapter Objectives

In this chapter, you will learn how to ...

- Code relative hyperlinks to Web pages in folders within a Web site
- Configure a hyperlink to a named anchor internal to a Web page
- Add interactivity to Web pages with CSS pseudo-classes
- Configure a navigation layout list with CSS
- Configure three-column page layouts using CSS
- Configure CSS for both screen and print display
- Utilize the "cascade" in CSS

Now that you've had some experience coding XHTML and CSS, you're ready to explore a variety of techniques in this chapter including XHTML relative hyperlinks and internal hyperlinks, CSS pseudo-classes, navigation list layout, three-column page layout, styling for print, and an overview of the "cascade" in CSS.

7.1 Another Look at XHTML Hyperlinks

Hyperlinks make the Web a "web" of interconnected information. In this section you'll revisit the topic of hyperlinks and explore coding relative links, using the target attribute to open Web pages in a new browser window, and coding hyperlinks that are internal to a Web page.

More on Relative Linking

As indicated earlier in Chapter 2, a relative link is used to link to Web pages within your site. You've been coding relative links to display Web pages within the same folder. There are times when you need to link to files in other folders on your Web site. Let's consider the example of a Web site for a dog groomer that highlights services and products. The Web developer for this site created separate folders called services and products in order to organize the site. See the folder and file listing shown in Figure 7.1.

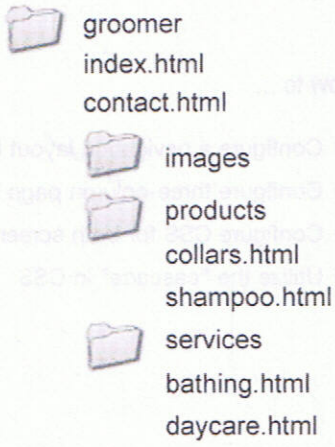


Figure 7.1
The dog groomer site contains the images, products, and services folders

Relative Link Examples

- To review, when linking to a file in the same folder or directory, the value of the `href` is the name of the file. For example, to link to the contact.html page from the home page (index.html), code the anchor element as follows:

```
<a href="contact.html">Contact</a>
```

- When linking to a folder located within the current directory, use both the folder name and the file name in the relative link. For example, to link to the collars.html page in the products folder from the home page (index.html), code the anchor element as follows:

```
<a href="products/collars.html">Collars</a>
```

- In Figure 7.1 the collars.html page is located in a subfolder of the groomer folder. The home page for the site, index.html is located in the groomer folder. When linking to a file that is up one directory level from the current page use `"../"` notation. To link to the home page for the site from the collars.html page, code the anchor element as follows:

```
<a href="../index.html">Home</a>
```

- When linking to a file that is in a folder on the same level as the current folder, the `href` value will use the `"../"` notation to indicate moving up one level and then down to the chosen folder. For example, to link to the bathing.html page in the services folder from the collars.html page in the products folder, code the anchor element as follows:

```
<a href="../services/bathing.html">Dog Bathing</a>
```

Don't worry if the use of `"../"` notation and linking to files in different folders seems new and different. In most of the exercises in this book you will code either absolute links to other Web sites or relative links to files in the same folder.

Opening a Link in a New Browser Window

The **`target` attribute** can be used on the anchor to open a link in a new browser window. For example,

```
<a href="http://yahoo.com" target="_blank">Yahoo!</a>
```

will open Yahoo!'s home page in a new window. Why not create a test page and try it? The `target` attribute with the value `"_blank"` configures the Web page to open in a new browser window.

By now you should be comfortable with hyperlinks. You may have noticed that these links display the top of the Web page. Sometimes it is helpful to link to an exact position on a Web page instead of to the top of the page. Internal links are used for this function.

Internal Links

Internal links are sometimes called bookmarks, **named anchors**, or named fragments. They can be very useful when you need to provide the capability to link to a specific portion of a Web page. Lists of frequently asked questions (FAQs) often use this technique.

When using internal links remember that there are two components:

1. The anchor tag that identifies a bookmark or **named fragment** of a Web page. This requires two attributes: the `id` attribute (supported by modern browsers) and the `name` attribute (used for compatibility with old browsers such as Netscape 4).
2. The anchor tag that links to the bookmark or named fragment of a Web page. This uses the `href` attribute.

To see how these two components are used, consider that Web pages sometimes have links to the top of a page (see Figure 7.2). This is accomplished in two steps as follows:

1. **Establish Target.** Type an anchor element that configures the `id` and `name` attributes on a blank line under the `<body>` tag. The value of the `id` and `name` attributes should describe the bookmark. It's a good idea to use lowercase letters and avoid punctuation, symbols, and spaces. The value given to the `id` attribute should be unique within the document. Place the following code near the top of a Web page document:

```
<a id="top" name="top"></a>
```

Figure 7.2
Notice how the anchor tags are used

```
<?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>Favorite Sites</title>
</head>
<body bgcolor="#FFFFFF">
<a id="top" name="top"></a><h1 align="center">Favorite Sites</h1>
<dl>
  <dt>Running</dt>
     <dd><a href="http://www.running.com">running.com</a></dd>
  <dt>Cooking</dt>
     <dd><a href="http://www.cooking.com">cooking.com</a></dd>
  The page
  <dt>Internet</dt>
     <dd><a href="http://www.internet.com">internet.com</a></dd>
</dl>
<p><a href="#top">Back to Top</a></p>
</body>
</html>
```

This anchor tag creates the named fragment for the top of the page.

This anchor tag indicates the link to the top of the page.

2. **Reference Target.** At the point of the page where you want to place a link to the top, type another anchor element. Use the `href` attribute and place a `#` (sometimes called a hash mark) before the name of the bookmark. The XHTML for a hyperlink to the named anchor `"top"` is

```
<a href="#top">Top of Page</a>
```

The hash mark indicates that the browser should search for an anchor tag on the same page. If you forget to type the hash mark, the browser will not look on the same Web page; it will look for an external file. A bookmark or named anchor does not have to be at the top of a page; it can be just about anywhere.

If you are coding only for an XHTML-compliant browser such as Internet Explorer 5 (or later), Mozilla Firefox, or Netscape 6 (or later), you can use the `id` attribute with any container tag, such as a `<p>` or a `<h1>`, to create a named fragment or bookmark. The top of page example uses the anchor element to provide for backward compatibility with Netscape 4.

HANDS-ON PRACTICE 7.1

You will work with internal links in this Hands-On Practice. Locate the Chapter7/starter1.html file in the student files. Figure 7.3 shows a partial screenshot of this Web page.

Launch Notepad and open the starter1.html file. Save the file as favorites.html. Examine the source code and notice that the top portion of the page contains an unordered list with categories of interest (such as Hobbies, XHTML, CSS, and Professional Organizations) that correspond to the text displayed in the `<h2>` elements below. After each `<h2>` element is a definition list of topics and URLs related to that category. It might be helpful to Web page visitors if they can click a category item and

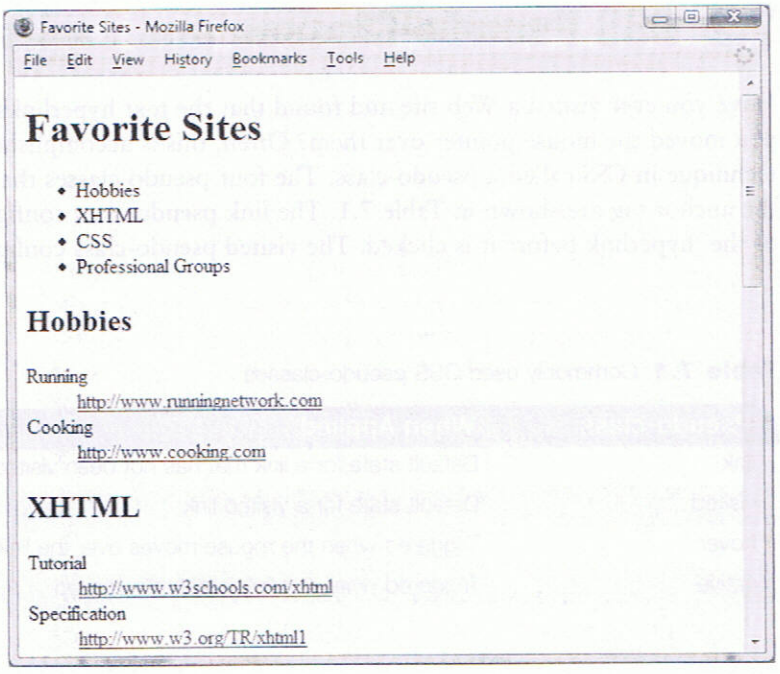


Figure 7.3
You will add internal links to this Web page

immediately jump to the page area that has information related to that item. This could be a useful application of internal links!

Modify the page as follows:

1. Code a named anchor for each `<h2>` element in the definition list. For example:

```
<h2><a id="hobbies" name="hobbies"></a>Hobbies</h2>
```

2. Add hyperlinks to the items in the unordered list so that each entry will link to its corresponding `<h2>`.
3. Add a named fragment near the top of the page.
4. Near the bottom of the favorites.html page add a link to the top of the page.

Save the file and test it. Compare your work with the sample found in the student files (Chapter7/favorites.html).

There may be times when you need to link to a named fragment on another Web page. To accomplish this, place the internal link after the file name in the anchor tag. So, to link to the "Professional Groups" (given that it is a named fragment called "groups") from any other page on the mywebsite Web, you could use the following XHTML:

```
<a href="favorites.html#groups">Professional Organizations</a>
```

FAQ

Why don't some of my internal links work?

A Web browser cannot display less than the height of the browser window. If there is not enough space left on the bottom of the page below the named reference, it cannot be displayed at the top of the page. Try adding some blank lines (use the `<br />` tag) to the lower portion of the Web page. Save your work and test your internal links again.

7.2 CSS Pseudo-Classes and Links

Have you ever visited a Web site and found that the text hyperlinks changed color when you moved the mouse pointer over them? Often, this is accomplished using a special technique in CSS called a pseudo-class. The four pseudo-classes that can be applied to the anchor tag are shown in Table 7.1. The **link pseudo-class** configures the appearance of the hyperlink before it is clicked. The **visited pseudo-class** configures the appearance

Table 7.1 Commonly used CSS pseudo-classes

Pseudo-class	When Applied
link	Default state for a link that has not been visited
visited	Default state for a visited link
hover	Triggered when the mouse moves over the link
active	Triggered when the link is actually clicked

of the hyperlink after it is clicked. The **hover pseudo-class** configures the hyperlink as the mouse is held or "hovered" over it. The **active pseudo-class** configures the appearance of the hyperlink while it is being clicked. Notice the order in which the pseudo-classes are listed in Table 7.1. Anchor element pseudo-classes must be coded in this order (although it's okay to omit one or more of those listed). If you code the pseudo-classes in a different order, the styles will not be reliably applied. Some students find the order easier to remember if they think of the mnemonic device "lovehate" – link, visited, hover, active.

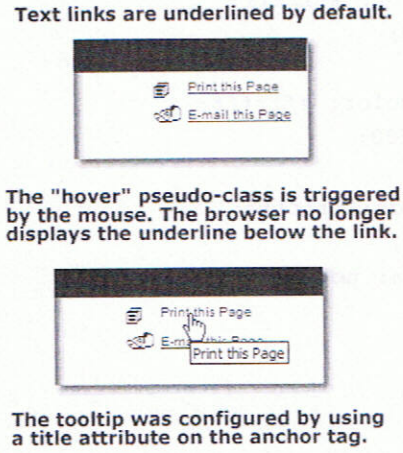
The syntax of pseudo-classes uses a colon (:) to apply the **pseudo-class** to the anchor tag. The following code sample will configure text hyperlinks to be red initially. The sample also uses the hover pseudo-class, `a:hover`, to configure the links to change their appearance when the visitor places the mouse pointer over them so that the underline disappears and the color changes.

```
<style type="text/css">
a:link {   color: #ff0000;
}
a:hover {  text-decoration: none;
           color: #000066;
}
</style>
```

Figure 7.4 shows part of a Web page that uses this technique. Note the position of the mouse pointer over the Print this Page link—the link color has changed and has no underline.

While some Web design experts, such as Jakob Nielsen, recommend that Web developers not change the default look of text links, this technique is often used. Most modern browsers (since Internet Explorer 4+ and Netscape 6+) support CSS pseudo-classes. Netscape 4.x does not support the hover pseudo-class, but the technique degrades gracefully and the hyperlink is still usable.

Figure 7.4
Using the hover pseudo-class



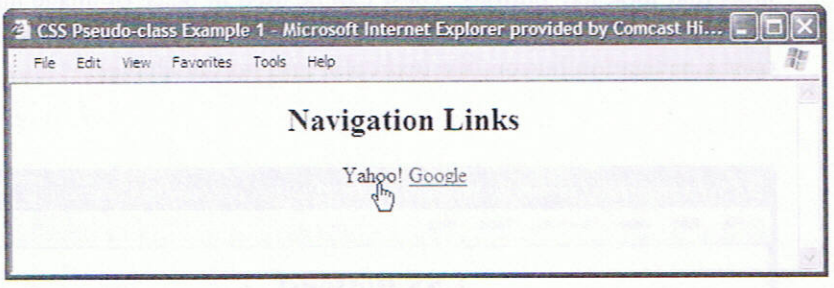
HANDS-ON PRACTICE 7.2

You will use pseudo-classes to create interactive hyperlinks in this Hands-On Practice as you create a series of pages that contain hyperlinks styled in different ways.

Part 1

The first page contains text links that you will configure to use CSS pseudo-classes. A sample is shown in Figure 7.5. When the mouse hovers over a link, it will change color and the underline will disappear. You will code embedded CSS to configure the link, visited, and hover pseudo-classes for the anchor selector.

Figure 7.5
The hyperlink's underline disappears when the mouse hovers



Launch Notepad and type the following XHTML:

```
<?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 Pseudo-class Example 1</title>
<style type="text/css">
body {   margin: 0 auto;
         width: 400px;
         text-align: center;
}
```

```
a:link {   background-color: #ffffff;
           color: #ff0000;
}
a:visited {  background-color: #ffffff;
             color: #00ff00;
}
a:hover {  background-color: #ffffff;
           color: #000066;
           text-decoration: none;
}
</style>
</head>
<body>
<div align="center">
  <h2>Navigation Links</h2>
  <p><a href="http://yahoo.com">Yahoo!</a>
  <a href="http://google.com">Google</a></p>
</div>
</body>
</html>
```

Save your file as link1.html. Test your page in a browser and compare it with Figure 7.5. The student files contain a sample solution at Chapter7/link1.html. The browser applies the CSS pseudo-class rules to every link on the page. In this example, the CSS was coded using embedded styles, but an external style sheet also could have been used.

Part 2

Now you will create a page that uses CSS and pseudo-classes to configure navigation links that look like buttons. These can be used in place of image links to save on the bandwidth used by graphics. See the sample in Figure 7.6. When the mouse hovers over a navigation button, the text color and border change.

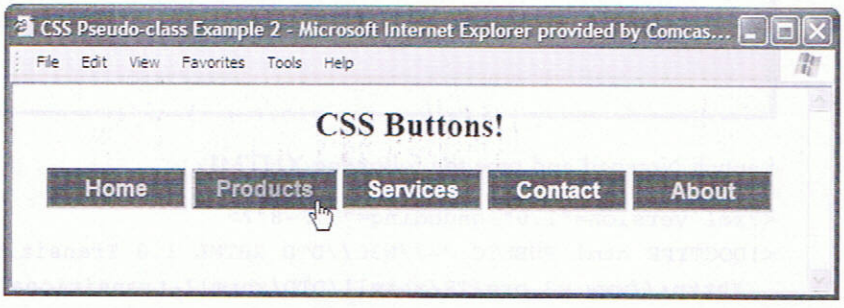


Figure 7.6 The hyperlink's appearance changes when the mouse hovers

You will use the following CSS properties to configure the buttons: `width`, `border`, and `padding`. Let's review these properties. The `width` property configures the amount of horizontal space used by the element in the browser window. The `border` property configures the width (`border-width`), style (`border-style`), and color (`border-color`) of the border around an element. The `padding` property configures the amount of padding—the blank space between the element and its border.

Launch Notepad and type the following XHTML:

```
<?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 Pseudo-class Example 2</title>
<style type="text/css">
body {   margin: 0 auto;
         width: 550px;
         text-align: center;
}
.button {  border: 2px inset #CCCCCC;
           width: 100px;
           padding: 3px 15px;
           color: #FFFFFF;
           background-color: #006600;
           font-family: Arial,Helvetica,sans-serif;
           font-size: 16px;
           font-weight: bold;
           text-align: center;
           text-decoration: none;
}
a.button: link {  color : #FFFFFF; }
a.button: visited { color : #CCCCCC; }
a.button: hover { color : #66CC33;
                  border: 2px outset #CCCCCC;
}
</style>
</head>
<body>
<div align="center">
  <h2>CSS Buttons!</h2>
  <a href="index.htm" class="button">Home</a>
  <a href="products.htm" class="button">Products</a>
  <a href="services.htm" class="button">Services</a>
  <a href="contact.htm" class="button">Contact</a>
  <a href="about.htm" class="button">About</a>
</div>
</body>
</html>
```

Save your file as link2.html. Test your page in a browser and compare it with Figure 7.6. The student files contain a sample solution at Chapter7/link2.html. The new technique here is to configure a class called `button`, which has all the initial properties of the navigation link. Because this example produced a button look, the `width`, `border`, and `padding` attributes were used. Then, CSS rules using an `anchor` selector with the `button` class and `link`, `visited`, and `hover` pseudo-classes were configured. The W3C prefers that the normal class names precede pseudo-classes in the selector. That is why the selector for the pseudo-classes use the `a.button:hover` notation. Finally, the `class`

attribute in the XHTML anchor tags connects the link on the page with the CSS style rules in the header.

Part 3

It is often the case that the design of the Web page requires the main navigation links to look different from the links within the content of the pages. You have already created a page, link2.html, with specially configured navigation links. You used the class called `button` to configure these links. In this part of the Hands-On Practice, you will add a line of text containing a hyperlink to the page to verify that the hyperlink retains the default browser appearance and behavior. Figure 7.7 shows a sample page.

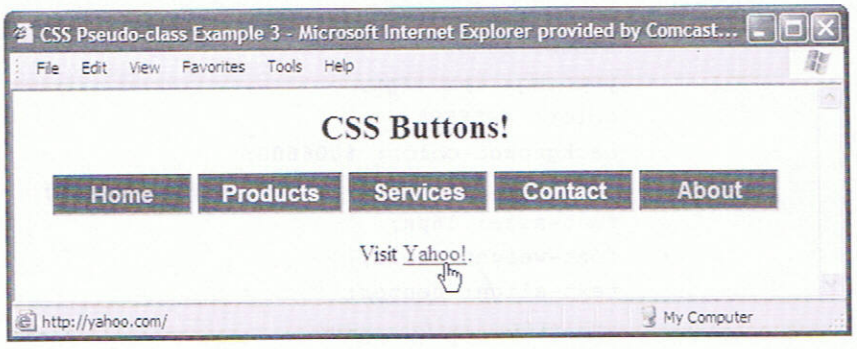


Figure 7.7
The link to Yahoo! retains the default hyperlink properties

Launch Notepad and open your link2.html file. Save the file as link3.html. Modify the title to be "CSS Pseudo-class Example 3" and add the following paragraph under the navigation links:

```
<p>Visit <a href="http://yahoo.com">Yahoo!</a>.</p>
```

Save your file, test your page in a browser, and compare it with the one shown in Figure 7.7. The student files contain a sample solution at Chapter7/link3.html. Because the new link is not part of the defined class `button`, it retains the default hyperlink characteristics. If you needed yet another set of characteristics for links in another section of the page such as the footer, you could define a new class with a unique name and configure pseudo-classes, as was done in Part 2 of this Hands-On Practice.

As you can see, pseudo-classes—along with careful configuration of classes, can be a powerful tool for a Web developer.

7.3 CSS Navigation Layout Using Lists

One of the advantages of using CSS for page layout involves the use of semantically correct code. Writing semantically correct code means using the markup tag that most accurately reflects the purpose of the content. Using the various levels of heading tags for content headings and subheadings, or placing paragraphs of text within paragraph tags (rather than using line breaks) are examples of writing semantically correct code. This type of coding is a step in the direction to support the Semantic Web. Leading Web

developers such as Eric Meyer, Mark Newhouse, Jeffrey Zeldman, and others have promoted the idea of using unordered lists to configure navigation menus. After all—a navigation menu is a list of links—semantically speaking it's a much better fit than coding links in separate paragraphs or using the `display:block` property on anchor tags.

Figure 7.8 shows the top portion of a revised twocolumn.html (the page you created in Hands-On Practice 6.6). In this version the CSS declaration for the `navBar` class was changed (`display:block` and `margin:15px` were removed), the left column was widened a bit, and the navigation links were coded in an unordered list. An XHTML code snippet follows:

```
<ul class="navBar">
  <li><a class="navBar" href="home.html">Home</a></li>
  <li><a class="navBar" href="spring.html">Spring</a></li>
  <li><a class="navBar" href="summer.html">Summer</a></li>
  <li><a class="navBar" href="fall.html">Fall</a></li>
  <li><a class="navBar" href="winter.html">Winter</a></li>
</ul>
```

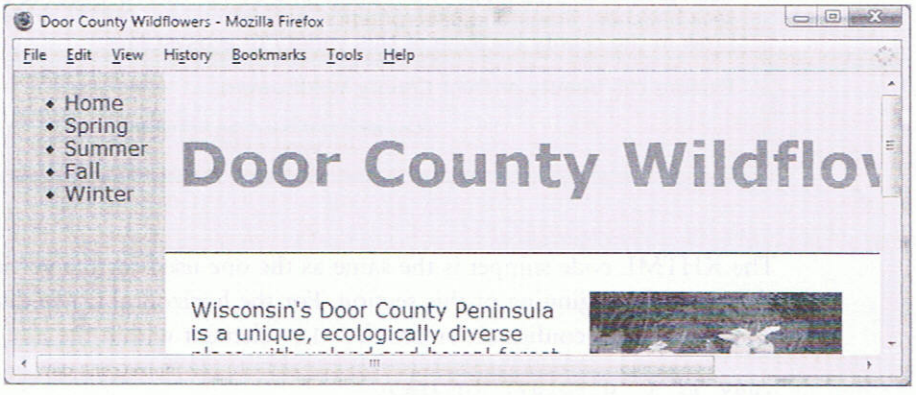


Figure 7.8
An unordered list to configure the navigation menu

Perhaps you would prefer that the bullets in the unordered list were not displayed. Use the **`list-style-type` property** to configure the list-item markers (bullets). The property `list-style-type:none` prevents the browser from displaying the bullets.

Figure 7.9 shows the effect of adding `list-style-type:none` to the `navBar` class.

If you would like a custom image to replace the bullet, use the **`list-style-image` property**. In Figure 7.10 an image named arrow.gif was configured to replace the bullets using: `list-style-image:url(arrow.gif)`. View the twocolumn1.html file in the Chapter7 folder in the student files to examine the code.

Figure 7.9
An unordered list with `list-style-type:none`

Figure 7.10
An unordered list with an image replacing the bullet

You may be wondering how to apply this technique to a horizontal navigation menu such as the one coded on the page1.html page used in Hands-On Practice 6.5. The answer is CSS! List items are block elements. They need to be configured as inline elements to display in a single line. The `display:inline` property is used to accomplish this. Figure 7.11 displays a new version of the page using this technique. The page looks about the same as the original (Figure 6.16) when displayed in a browser even though the XHTML and CSS are configured to use a list.

Figure 7.11 Horizontal navigation using an unordered list configured with CSS

The XHTML code snippet is the same as the one used for the vertical navigation menus shown at the beginning of this section. For the horizontal list to display properly, you must add a CSS configuration for the `<li>` element within the `nav` class as follows:

```
.nav li {  display: inline;
           list-style-type: none;
}
```

Focus on Accessibility

View the home0.html and wildflower0.css files in the Chapter7 folder in the student files to experiment with this technique. See Chapter7/skipnav.html for a version of this page that includes a transparent image configured as an internal link to the named fragment maincontent. This "skip navigation" method allows visitors using screen readers to easily skip repetitive navigation links.

7.4 Three-Column CSS Page Layout

Often a Web page layout will consist of a header across the top of the page with three columns below: navigation, content, and sidebar. If you are thinking about this layout as a series of boxes—you're thinking correctly for configuring pages using CSS! Figure 7.12 shows a wireframe sketch of this page layout design. Figure 7.13 (shown also in the color insert section) shows a Web page configured using this design. You will create this page in the next Hands-On Practice.

Figure 7.12
Sketch of three-column page layout

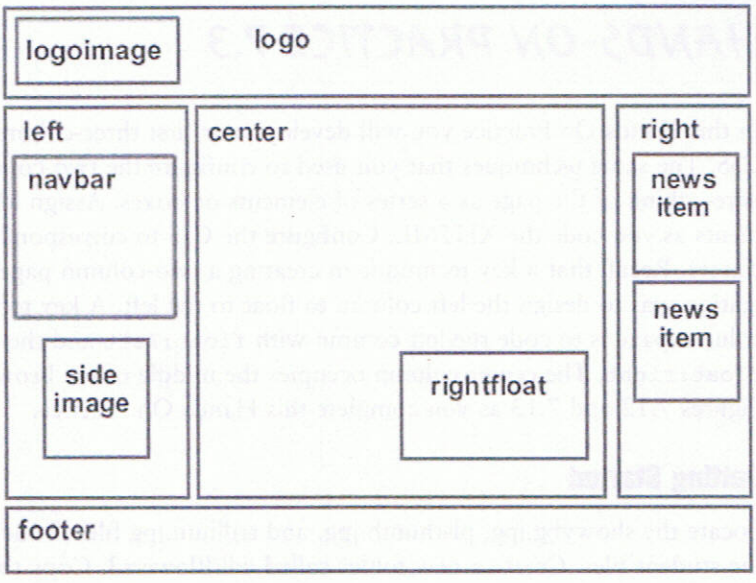


Figure 7.13
This three-column page layout is designed using CSS and no tables

See the center color insert

HANDS-ON PRACTICE 7.3

In this Hands-On Practice you will develop your first three-column Web page using CSS. The same techniques that you used to configure the two-column page will apply here—think of the page as a series of elements or boxes. Assign ids or classes to the elements as you code the XHTML. Configure the CSS to correspond to the ids and classes. Recall that a key technique in creating a two-column page with left column navigation was to design the left column to float to the left. A key technique in our three-column page is to code the left column with `float:left` and the right column with `float:right`. The center column occupies the middle of the browser window. Refer to Figures 7.12 and 7.13 as you complete this Hands-On Practice.

Getting Started

Locate the showybg.jpg, plsthumb.jpg, and trillium.jpg files in the Chapter 7 folder in the student files. Create a new folder called wildflowers3. Copy the files to the folder.

Part 1—Code the XHTML

Review Figures 7.12 and 7.13. Notice the page elements: a logo area with both a logo and a background image that repeats; a left column with a navigation area and an image; a center column with paragraphs of text, a heading, and an image that floats to the right; a right column with two news items; and a footer. These will all be coded to use ids and classes corresponding to CSS, which configures a number of properties including the `float`, `margin`, `border`, `font-family`, and so on. The navigation menu links will be configured using an unordered list. As you code the XHTML document, you will place the elements on the page and assign `id` and `class` values that correspond to the areas in the sketch in Figure 7.12. Launch Notepad and type in the following XHTML:

```
<<?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>Door County Wildflowers</title>
</head>
<body>
<div id="container">
  <div id="logo">
    Door County Wildflowers
  </div>
  <div id="left">
    <ul class="navBar">
      <li><a class="nav" href="home.html">Home</a></li>
      <li><a class="nav" href="spring.html">Spring</a></li>
      <li><a class="nav" href="summer.html">Summer</a></li>
      <li><a class="nav" href="fall.html">Fall</a></li>
      <li><a class="nav" href="winter.html">Winter</a></li>
    </ul>
```

```
    <img class="sideimages" src="plsthumb.jpg" width="100"
    height="100" alt="Showy Pink Lady Slipper" />
  </div>
  <div id="right">
    <h4>The Ridges</h4>
    <p class="newsitem">The Ridges Nature Sanctuary offers wild
    orchid hikes during the summer months. For more info, visit
    <a href="http://www.ridgesanctuary.org">The Ridges</a>.</p>
    <h4>Newport State Park</h4>
    <p class="newsitem">The Newport Wilderness Society sponsors
    free meadow hikes at 9am every Saturday. Stop by the park
    office to register.</p>
  </div>
  <div id="center">
    <p>Wisconsin's Door County Peninsula is a unique,
    ecologically diverse place with upland and boreal forest,
    bogs, swamps, sand and rock beaches, limestone escarpments, and
    farmlands.</p>
    <p>A wide array of wildflowers grow in the county because
    of this variety of ecosystems.</p>
    <img src="trillium.jpg" width="200" height="150" alt="Trillium"
    id="floatright" />
    <h3>Explore the beauty <br />of Door County Wildflowers....</h3>
    <p>With five state parks, tons of county parks, and private
    nature sanctuaries, Door County is teeming with natural areas
    for you to stalk your favorite wildflowers.</p>
  </div>
  <div class="footer">
    Copyright &copy; 2007 Door County Wild Flowers<br />
    Last Updated on 07/07/07
  </div>
</div>
</body>
</html>
```

Save your page as threecolumn.html in your wildflowers3 folder. Test the page in a browser. Your display will not look like Figure 7.13 since you have not yet configured the CSS. The top of your page should look similar to the page shown in Figure 7.14.

Part 2—Code the Basic CSS

For ease of editing, in this Hands-On Practice you will code the CSS as embedded styles in the header section of the Web page. However, if you were creating an entire Web site you would most likely use an external style sheet as you did in Hands-On Practice 6.5.

Launch Notepad and open threecolumn.html. Let's take a moment to consider the main elements used on the page shown in Figure 7.13: logo, left column, right column, center column, and footer. The left column will contain a navigation area and a small image. The center column will contain paragraphs, a heading, and a right-floating image. The right column will contain a series of headings and news items. Locate these areas on the sketch in Figure 7.12. Notice also that the same font is used throughout

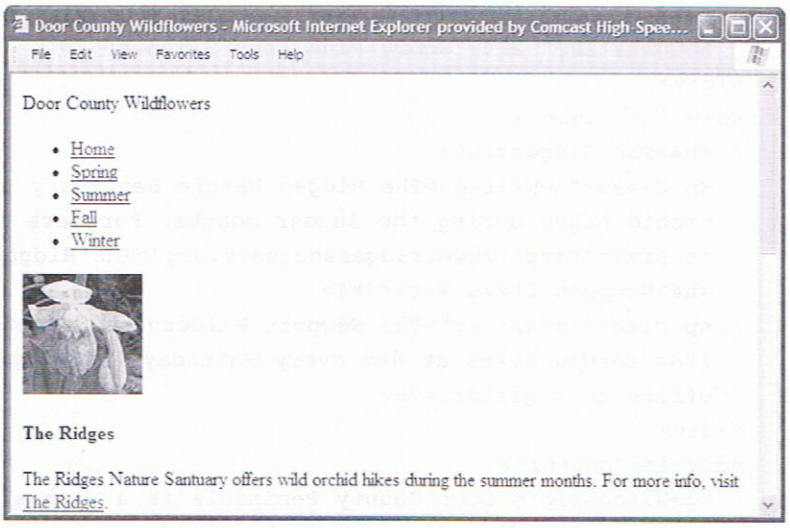


Figure 7.14
The three-column page before CSS is applied

the page and the page begins right at the browser margin. Launch Notepad and open your threecolumn.html file. In the header section of your Web page document, add a tag to begin the embedded styles:

```
<style type="text/css">
```

Now let's consider the CSS configuration. Type the CSS in your document as it is discussed as follows:

1. **Body Selector.** Set the margin to 0 pixels. Configure the background color to `#ffffff`.

```
body {  margin:0;
        background-color: #ffffff;
}
```

2. **Container.** Configure this area with background (`#eeeeee`) and text (`#006600`) colors, a minimum width of 700 pixels, and with font family of Verdana, Arial or sans-serif.

```
#container {   background-color: #eeeeee;
               color: #006600;
               min-width: 700px;
               font-family: Verdana, Arial, sans-serif;
}
```

3. **Logo.** Code this area so that the image showybg.jpg will repeat using `background-image:url(showybg.jpg)`. The text should be set to 2.5em font size and bold. The height of the logo area is 100 pixels—this corresponds to the height of the background image. Although it will most likely never display, configure the background color to `#eeeeee`. The text color should be `#cc66cc`. Set the left padding to 20 pixels. Configure a 2 pixel solid black border across the bottom of this area as follows:

```
#logo {  color: #cc66cc;
         background-color: #eeeeee;
         font-size: 2.5em;
```

```
        font-weight: bold;
        border-bottom: 2px solid #000000;
        height: 100px;
        background-image: url(showybg.jpg);
        padding-left: 20px;
}
```

4. **Left Column.** One of the keys to this three-column page layout is that the left column is designed to float to the left of the browser window. Configure a width of 150.

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

5. **Right Column.** One of the keys to this three-column page layout is that the right column is designed to float to the right of the browser window. Configure a width of 200 pixels.

```
#right { float: right;
         width: 200px;
}
```

6. **Center.** The center column will take up all the room that is available after the left and right columns float. The content area has a special need for margins since the left and right columns are floating on either side. Set the left margin to 150 pixels, the right margin to 200 pixels, and the remaining side margins to 0. Configure padding for this area, also. Set the background (`#ffffff`) and text (`#006600`) colors for this area.

```
#center { margin: 0 200px 0 150px;
          padding: 1px 10px 20px 10px;
          background-color: #ffffff;
          color: #006600;
}
```

7. **Footer.** Configure the page footer with very small text that is centered. Configure the background (`#ffffff`) and text (`#006600`) colors for this area. Set the top padding to 10 pixels. A `clear:both` is needed to clear the float of the right and left columns as follows:

```
.footer { font-size: .70em
          text-align: center;
          color: #006600;
          background-color: #ffffff;
          padding-top: 10px;
          clear:both;
}
```

At this point you have configured the main elements of the three-column page layout. It's a good idea to save and do a quick test to make sure you are on the right track. Code the closing XHTML style tag: `</style>`.

Save the threecolumn.html file in the wildflowers3 folder. Test your page in a browser. It should look similar to the one shown in Figure 7.15. Note that there is still some detail work to do but you are well on your way!

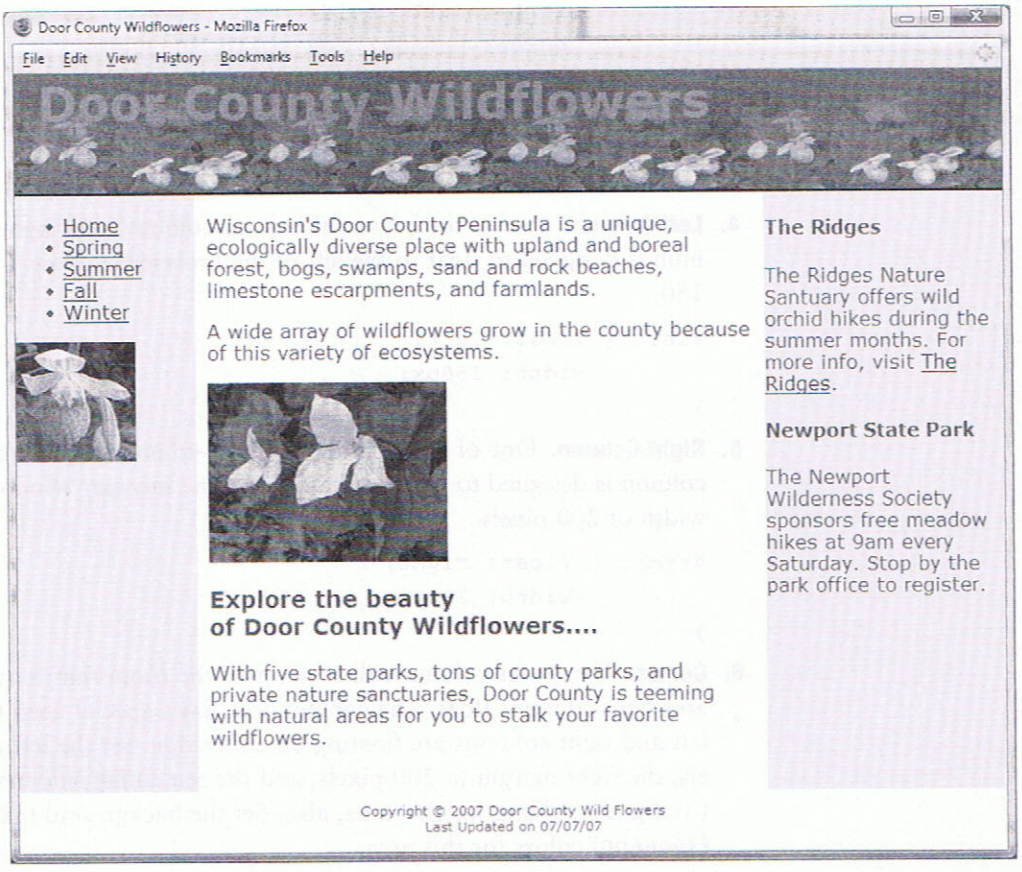


Figure 7.15
The CSS for the basic elements of the three-column layout is complete

Part 3—Continue Coding CSS

Now you are ready to continue with your styles. Open the threecolumn.html page in Notepad and position your cursor on a blank line above the closing style tag. First we will configure the components in the left column as follows:

1. **Navigation Menu.** Configure the unordered list to provide for a 20 pixel top margin and not to display any bullets.

```
.navBar { margin-top: 20px;
          list-display-type: none;
}
```

Configure the navigation links to have no underline (`text-decoration:none`). Configure the font size to 1.2em. Pseudo-classes should be configured for link, visited, and hover with different text colors as follows:

```
.nav {  text-decoration: none;
        font-size: 1.2em;
}
a.nav:link {  color:#006600;}
              background-color: #eeeeee; }
a.nav:visited {  color: #003300;}
                 background-color: #eeeeee; }
a.nav:hover {  color: #cc66cc;}
               background-color: #eeeeee; }
```

2. **Left Column Image (`sideimages`).** Configure this class with a margin of 30 pixels as follows:

```
.sideimages { margin: 30px;}
```

Next, we'll configure the contents of the center column. Styles for paragraphs, `heading elements`, and an image that floats to the right need to be constructed.

3. **`p`:** Configure the paragraph selector to use a margin of 20 pixels as follows:

```
p { margin: 20px; }
```

4. **`h3`:** Configure the `h3` selector with the same text color as the logo and the same background color as the main body of the page as follows:

```
h3 {  color: #cc66cc;
      background-color: #FFFFFF;
}
```

5. **Image Floating at the Right.** Set the `floatright` id to use a 10 pixel margin and `float:right` as follows:

```
#floatright {  margin: 10px;
               float: right;
}
```

Now, we'll configure the styles for the contents of the right column. The announcements consist of a heading (contained within `<h4>` tags) and a paragraph (assigned to a class called `newsitem`).

6. **`h4`:** Configure the heading to have a 1 pixel black solid bottom border, 2 pixels of padding at the bottom, a 10 pixel margin, the same text color as the logo, and the same background color as the right column:

```
h4 {  padding-bottom: 2px;
      border-bottom: 1px solid #000000;
      margin: 10px;
      color: #cc66cc;
      background-color: #eeeeee;
}
```

7. **News Items.** Configure a class called `newsitem` that uses a small font and has a 10 pixel margin as follows:

```
.newsitem {  font-size:.9em;
             margin: 10px;
}
```

Save the threecolumn.html file in the wildflowers3 folder.

Part 4—Test the Page

Now that your styles are coded, test the threecolumn.html page again. Your page should look similar to the screenshot shown in Figure 7.13. If there are differences, verify the `id` and `class` values in your XHTML. Also check the syntax of your CSS. You may find the W3C CSS validator at http://jigsaw.w3.org/css-validator helpful when verifying CSS syntax. The student files contain a copy of threecolumn.html in the Chapter7 folder.

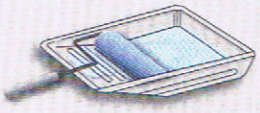
FAQ

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`, and `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.



CHECKPOINT 7.1

1. Describe a reason to organize the files in a Web site using folders and subfolders.
2. State a reason to use an unordered list to configure navigation links.
3. You are using CSS pseudo-classes on a Web page to configure the navigation links to look like buttons. You want the "regular" links in the Web page content to be configured as they normally would (not look like a button). Describe how you could configure the styles and XHTML to accomplish this.

7.5 CSS Styling for Print

Even though the advent of the "paperless society" has been talked about for decades, the fact is that many people still love paper and you can expect your Web pages to be printed. CSS offers you some control over what gets printed and how the printouts are configured.

This is easy to do using external style sheets. Create one external style sheet with the configurations for browser display and a second external style sheet with the special printing configurations. Associate both of the external style sheets to the Web page using two `<link>` elements. The `<link>` elements will utilize a new attribute, called `media`. Configure the link element for your browser display with `media="screen"`. Configure the link element for your printout with `media="print"`. Modern browsers will use the correct style sheet depending on whether they are rendering a screen display or preparing to print a document. An example of the XHTML follows:

```
<link rel="stylesheet" href="wildflower.css" type="text/css"
  media="screen" />
<link rel="stylesheet" href="wildflowerprint.css" type="text/css"
  media="print" />
```

Often `display:none` is used in the print style sheet to prevent banner ads, navigation, or other extraneous areas from appearing on the printout. Another common practice is to configure the font sizes on the print style sheet to use `pt` sizes—this will better con-

trol the text on the printout. You can also use styles to configure areas in the document, such as detailed contact info, that are only printed out and do not appear in the browser window. Figure 7.16 shows the print preview of the content page you created in Hands-On Practice 6.5 (see Figure 6.16). Notice that the print preview includes the navigation area. Figure 7.17 displays an alternate version of the page that uses CSS to prevent the navigation area from printing. You will explore this technique in the next Hands-On Practice.

Figure 7.16
Print preview of the page displayed in Figure 6.16

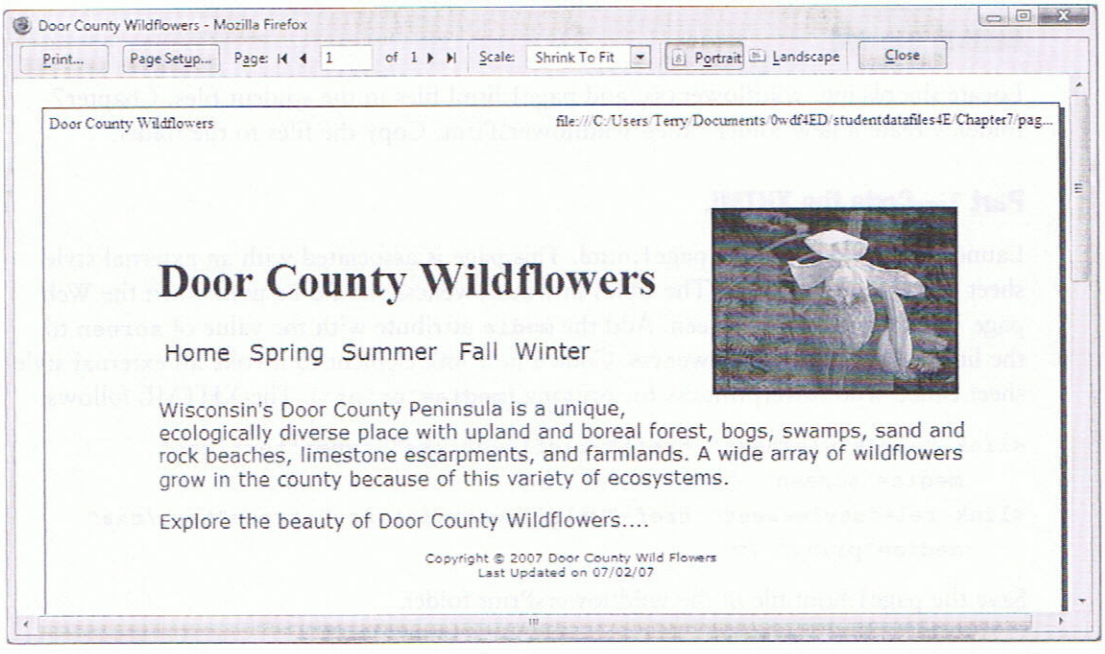
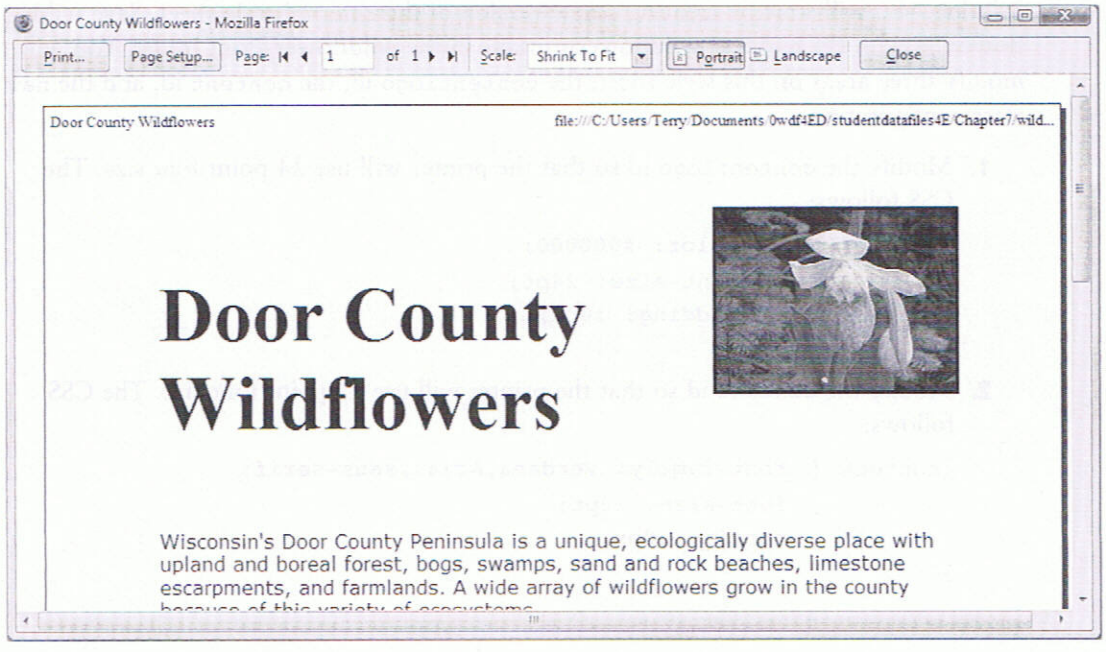
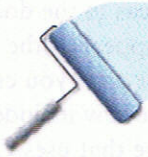


Figure 7.17
Print preview using CSS to remove the navigation from the printout





HANDS-ON PRACTICE 7.4

In this Hands-On Practice you will code special styles to use when printing a Web page. We will use the page1.html and wildflower.css files that you created in Hands-On Practice 6.5 as a starting point. Figure 6.16 shows the browser display of the page1.html file. You will create a new version of the page1.html file and a new style sheet configured for printing. When printed, the logo will be configured using a 24 `pt` size and the navigation will not display.

Getting Started

Locate the pls.jpg, wildflower.css, and page1.html files in the student files, Chapter7 folder. Create a new folder called wildflowersPrint. Copy the files to the folder.

Part 1—Code the XHTML

Launch Notepad and open page1.html. This page is associated with an external style sheet called wildflower.css. The styles in wildflower.css should be used when the Web page is displayed on the screen. Add the **`media` attribute** with the value of `screen` to the link element for wildflower.css. Code a new link element to invoke an external style sheet called wildflowerprint.css for printing (`media="print"`). The XHTML follows:

```
<link rel="stylesheet" href="wildflower.css" type="text/css"
    media="screen" />
<link rel="stylesheet" href="wildflowerprint.css" type="text/css"
    media="print" />
```

Save the page1.html file in the wildflowersPrint folder.

Part 2—Code the New CSS

Launch Notepad and open wildflower.css. Since you want to keep most of the styles for printing, you will start by creating a new version of the external style sheet. Save wildflower.css with the name of wildflowerprint.css in the wildflowersPrint folder. You will modify three areas on this style sheet: the `contentlogo` id, the `content` id, and the `nav` class configuration.

1. Modify the `contentlogo` id so that the printer will use 24 point font size. The CSS follows:

```
#contentlogo { color: #000000;
               font-size: 24pt;
               padding: 10px;
}
```

2. Modify the `content` id so that the printer will use 12 point font size. The CSS follows:

```
.content {  font-family: Verdana,Arial,sans-serif;
            font-size: 12pt;
            margin: 10px;
}
```

3. Configure the nav class to not be printed with the page. Delete all styles associated with the nav class and replace them with the following CSS:

```
.nav { display: none;
}
```

Save your file in the wildflowersPrint folder.

Part 3—Test Your Work

Test your page1.html file in a browser. Select Print, Preview. Your display should look similar to the page shown in Figure 7.17. The logo and content font sizes have been configured. The navigation does not display. The student files contain a copy of page1.html and wildflowerprint.css in the Chapter7/wildflowersPrint folder.

7.6 The "Cascade"

Figure 7.18 shows the "cascade" (**rules of precedence**) that applies the styles in order from outermost (external styles) to innermost (actual XHTML coded on the page). This way site-wide styles can be configured but overridden when needed by more granular (or page-specific) styles.

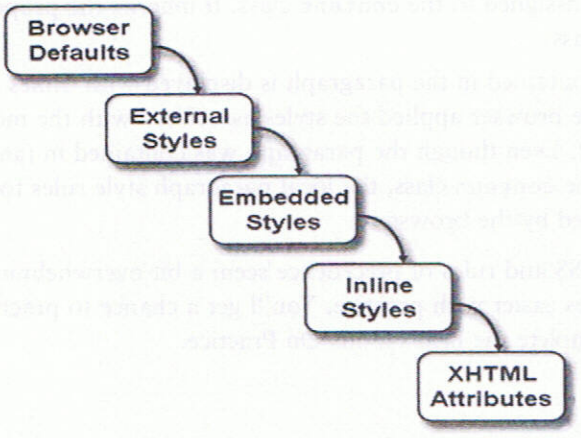


Figure 7.18
The "cascade" of Cascading Style Sheets

External styles can apply to multiple pages. If a Web page contains both a link to an external style sheet and embedded styles, the external styles will be applied first, and then the embedded styles will be applied. This allows a Web developer to override global external styles on selected pages.

If a Web page contains both embedded styles and inline styles, the embedded styles are applied first, and then the inline styles are applied. This allows a Web developer to override page-wide styles for particular XHTML tags or classes.

Any XHTML tag or attribute will override styles. For example, a `<strong>` tag will override corresponding font-related styles configured for the page. If no attribute or style is applied to an element, the browser default is applied. The appearance of the browser default may vary by browser and you might be disappointed with the result. Specify the properties of your text and Web page elements using CSS. Avoid depending on the browser default.

The overall CSS cascade was described above. In addition to this general cascade of CSS types, the style rules themselves follow rules of precedence. Style rules applied to more local elements (such as a paragraph) take precedence over those applied to more global elements (such as a `<div>` which contains the paragraph).

Let's look at an example of the cascade. The CSS and XHTML code is shown below. The CSS has two style rules: a rule creating a class named `content` which configures text using the Arial (or generic sans-serif) font family, and a rule configuring all paragraphs to use the Times New Roman (or generic serif) font family. The CSS follows:

```
.content { font-family:Arial, sans-serif; }
p { font-family: "Times New Roman", serif;}
```

The XHTML on the page contains a `<div>` with multiple elements, such as headings and paragraphs. Partial code follows:

```
<div class="content">
  <h1>Main Heading</h1>
  <p>This is a paragraph. Notice how the paragraph is contained in
  the div.</p>
</div>
```

Here's how the browser would render the code:

1. The text contained in the heading is displayed with Arial font because it is part of the `<div>` assigned to the `content` class. It inherits the properties from its parent (`<div>`) class.
2. The text contained in the paragraph is displayed with Times New Roman font because the browser applied the styles associated with the most local element (the paragraph). Even though the paragraph was contained in (and is considered a child of) the `content` class, the local paragraph style rules took precedence and were applied by the browser.

Don't worry if CSS and rules of precedence seem a bit overwhelming at this point. CSS definitely becomes easier with practice. You'll get a chance to practice with the "cascade" as you complete the next Hands-On Practice.

HANDS-ON PRACTICE 7.5

You will experiment with the "cascade" in this Hands-On Practice as you work with a Web page that uses external, embedded, and inline styles. Begin by creating an external style sheet called site.css that sets the `background-color` of the Web page to a shade of yellow (`#FFFF66`) and the `font-size` to `24px`. The code follows:

```
body {  background-color: #FFFF66;
        font-size: 24px;
}
```

Next, create a Web page called mypage1.html that is associated with the file site.css and has an embedded style that sets the text color to blue. The file mypage1.html will contain two paragraphs of text. The XHTML used to code the first paragraph will not use

any styles. The XHTML used to code the second paragraph will use inline styles to set the text color to red and the `font-size` to `14px`. The code for mypage1.html follows:

```
<?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>External Styles</title>
  <link rel="stylesheet" href="site.css" type="text/css" />
  <style type="text/css">
  body { color: #0000FF;
  }
  </style>
</head>
<body>
<p>This paragraph does not contain inline styles.</p>
<p style="color:#FF0000;font-size:14px">This paragraph contains inline
styles and should be red with 14 px font</p>
</body>
</html>
```

Save both site.css and mypage1.html in the same folder. Display mypage1.html in a browser. Your page should look similar to the sample shown in Figure 7.19. The student files contain a sample solution at Chapter7/mypage1.html.

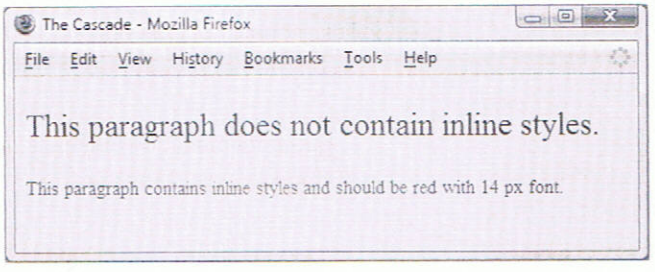


Figure 7.19 Mixing external, embedded, and inline styles

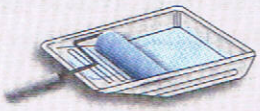
Take a moment to examine the mypage1.html Web page and compare it with its source code. The Web page picked up the yellow background and the `24px` font size from the external style. The embedded style configured the text to be the color blue. The first paragraph in the Web page does not contain any inline styles, so it inherits the style rules in the external and embedded style sheets. The second paragraph contains an inline style of red text color and `14px` font size—these settings override the corresponding external and embedded styles.

FAQ

Is there a quick way to apply the same styles to more than one XHTML tag or more than one class?

Yes, you can apply the same style rules to multiple selectors (such as XHTML elements, classes, or ids) by listing the selectors in front of the rule. The code sample below shows the `font-size` of `1em` being applied to both the paragraph and line item elements.

```
p, li { font-size: 1em; }
```



CHECKPOINT 7.2

1. State an advantage of using CSS to style for print.
2. Describe how to choose whether to configure an XHTML tag, create a class, or create an id when working with CSS.
3. List the following terms in the order that the properties and attributes are applied when using CSS.

   Inline styles
   External styles
   XHTML attributes
   Embedded styles

CHAPTER SUMMARY

This chapter explored a variety of Web development topics including XHTML relative hyperlinks and internal hyperlinks, CSS pseudo-classes, navigation list layout, three-column page layout, styling for print, and an overview of the "cascade" in Cascading Style Sheets.

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

Key Terms

active pseudo-class
the "cascade"
hover pseudo-class
internal links
link pseudo-class
`list-style-type` property
`list-style-image` property
`media` attribute
named anchor
named fragment
pseudo-class
rules of precedence
`target` attribute
visited pseudo-class

Review Questions

Multiple Choice

1. Which of the following anchor tag attributes define an internal bookmark or named reference in a page?

a. `id` and `name`
b. `name` and `bookmark`
c. `internal` and `id`
d. `id` and `bookmark`

2. How would you link to the bookmark `#jobs` on the page employ.html from the home page of the site?

a. `<a href="employ.html#jobs">Employment Opportunities</a>`
b. `<a name="employ.html#jobs">Employment Opportunities</a>`
c. `<a link="employ.html#jobs">Employment Opportunities</a>`
d. none of the above

3. Which of the following causes an object not to display either in the browser window or on a printed page?

a. `display:block;`
b. `display: 0px;`
c. `display:none;`
d. this cannot be done with CSS

4. Which property and value is used to configure an unordered list item so that the bullet does not display?

a. `list-bullet:none;`
b. `list-style-type:none;`
c. `list-style-type:off;`
d. `list-marker:none;`

5. How would you define an internal bookmark or named fragment at the top of a page, called `"top"`?

a. `<a name="top" id="top"></a>`
b. `<a bookmark="top"></a>`
c. `<a name="#top" id="#top"></a>`
d. id and bookmark

6. Which of the following is true if a Web page contains both a link to an external style sheet and embedded styles?

a. embedded styles will be applied first, then the external styles will be applied
b. the inline styles will be used
c. external styles will be applied first, and then the embedded styles will be applied
d. Web page will not display

7. Which of the following is the file extension for an external style sheet?

a. ess
b. css
c. html
d. no file extension is necessary

8. Which of the following elements is used to associate a Web page with an external style sheet?

a. `<target>`
b. `<a>`
c. `<include>`
d. `<link />`

9. Which of the following properties configures an image to use as a bullet point in an unordered list?

a. `bullet-image`
b. `image-style`
c. `list-style-image`
d. `bullet-style-image`

10. Which style rule below causes other page content to appear at the left of the element?

a. `position:left;`
b. `position:relative;`
c. `float:left;`
d. `float:right;`

Fill in the Blank

11. To indicate that an external style sheet is used to configure printing, code _____ on the `<link>` element.

12. The _____ is always transparent.

13. The _____ pseudo-class can be used to modify the display of a hyperlink when a mouse passes over it.

14. _____ is an attribute of the anchor element that can cause the new Web page to open in its own browser window.

15. The rules of _____ describe how Cascading Style Sheet rules, XHTML attributes, and browser defaults are applied.

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:

```
<?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>Predict the Result</title>
<style type="text/css">
body { background-color: #000066;
       color: #CCCCCC;
       font-family: Arial,sans-serif;
}
h1 { background-color: #FFFFFF;
     color: #000066;
     padding: 20px;
}
.navBar { list-style-type: none;
          display: inline;
          padding: 20px;
}
```

```
.nav { text-decoration: none;
       font-size: 1.2em;
}
a.nav:link {color: #eeeeee; }
a.nav:visited {color: #778899; }
a.nav:hover {color: #3399CC; }
</style>
</head>
<body>
<h1>Trillium Media Design</h1>
<ul>
  <li class="navBar"><a class="nav" href="index.html">
  Home</a></li>
  <li class="navBar"><a class="nav" href="about.html">
  About</a></li>
  <li class="navBar"><a class="nav" href="services.htm">
  Services</a></li>
</ul>
<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>Contact <a
href="mailto:web@trilliumtechnologies.com">Trillium</a><br />
Copyright &copy; 2008 Trillium Media Design
</div>
</body>
</html>
```

2. Fill in the Missing Code. This Web page should be configured so that the left navigation column has a light pastel background color and floats on the left side of the browser window. Instead, the navigation displays with a white background color. CSS properties and values, indicated by "_" (underscore), are missing. Fill in the missing code to correct the error.

```
<?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>Fill in the Missing</title>
<style type="text/css">
body { background-color: #d5edb3;
       color: #000066;
       font-family: Verdana, Arial, sans-serif;
}
#leftcolumn { float: left;
              width: 120px;
}
```

```
#rightcolumn { "_": "_";
               background-color: #ffffff;
               color: #000000;
               padding: 20px;
}
</style>
</head>
<body>
<div id="leftcolumn">
  <ul>
    <li><a href="home.html">Home</a></li>
    <li><a href="spring.html">Spring</a></li>
    <li><a href="summer.html">Summer</a></li>
    <li><a href="fall.html">Fall</a></li>
    <li><a href="winter.html">Winter</a></li>
  </ul>
</div>
<div id="rightcolumn">
  <h1>Trillium Media Design</h1>
  <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>
  Contact <a href="mailto:web@trilliumtechnologies.com">
  web@trilliumtechnologies.com</a>
  <br />Copyright &copy; 2008 Trillium Media Design
  </div>
</div>
</body>
</html>
```

3. Find the Error. The page below is intended for the navigation area to display on the right side of the browser window. What needs to be changed to make this happen?

```
<?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>Find the Error</title>
<style type="text/css">
body { background-color: #d5edb3;
       color: #000066;
       font-family: Verdana, Arial, sans-serif;
}
#rightcolumn { float: left;
               width: 120px;
}
```

```
#maincontent {  padding: 20px 150px 20px 20px;
                background-color: #ffffff;
                color: #000000;
}
</style>
</head>
<body>
<div id="rightcolumn">
  <ul>
    <li><a href="home.html">Home</a></li>
    <li><a href="spring.html">Spring</a></li>
    <li><a href="summer.html">Summer</a></li>
    <li><a href="fall.html">Fall</a></li>
    <li><a href="winter.html">Winter</a></li>
  </ul>
</div>
<div id="maincontent">
  <h1>Trillium Media Design</h1>
  <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>Contact <a href="mailto:web@trilliumtechnologies.com">
  web@trilliumtechnologies.com</a>
  <br />Copyright &copy; 2008 Trillium Media Design
  </div>
</div>
</body>
</html>
```

Hands-On Exercises

1. Write the XHTML to create a named anchor or bookmark at the beginning of a Web page designated by "top".
2. Write the XHTML to create an internal link to the named anchor designated by "top".
3. Write the XHTML to associate a Web page with an external style sheet named myprint.css to configure a printout.
4. Write the CSS to configure an image file named myimage.gif as the "bullet" in an unordered list.
5. Write the CSS to configure an unordered list not to display a "bullet".
6. **Extending Hands-On Practice 7.3.** In Hands-On Practice 7.3 you created files for a version of the Door County Wildflowers Web site. These files are also available in the Chapter7 folder in the student files. In this exercise, you will create two additional content pages for the Door County Wildflowers site, called spring.html and summer.html. Be sure that all CSS is placed in an external style sheet, called

mywildflower.css. (Modify pre-existing pages to use this style sheet). Rename threecolumn.html as appropriate. The following is some content to include on the new pages:

Spring Page (spring.html):

- Use the trillium.jpg image (see the Chapter7 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 Chapter7 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.

7. **Configure Printing for Hands-On Practice 7.3.** Configure special printing for the threecolumn.html file created in Hands-On Practice 7.3. Use the threecolumn.html file from Hands-On Practice 7.3 as a starting point. This file is in the Chapter7 folder in the student files. Save a copy of this file as threecolumnprint.html. Modify the file so that it links to an external style sheet called threecolumn.css instead of using embedded styles. Save and test your page. Create a new style sheet, called myprint.css, which will prevent the navigation from displaying when the page is printed. Modify the threecolumnprint.html page to link to this file. Review the use of the `media` attribute on the link element. Save all files and test your page. Select File, Print Preview to test the print styles. Hand in printouts of myprint.css, threecolumn.css, threecolumnprint.html source code (print in Notepad), and the browser display of threecolumprint.html to your instructor.

8. **Modify the Design of Hands-On Practice 7.3.** Locate the threecolumn.html page you created in Hands-On Practice 7.3. This file is in the Chapter7 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. Configure the style rules for threecolumn.html to display the page in this manner. Refer to Chapter 5 for CSS style rule suggestions. Hand in printouts of the source code (print in Notepad) and browser display for the Web page to your instructor.

9. **Practice Validating CSS.** Choose a CSS external style sheet file to validate—perhaps you have created one for your own Web site. Otherwise, use an external style sheet file that you worked with in this chapter. Use the W3C free CSS validator (http://jigsaw.w3.org/css-validator/). If your CSS does not immediately pass the validation test, modify it and test again. Repeat this process until the W3C validates your CSS code. Write a one or two paragraph summary about the validation process. Answer the following questions. Was it easy to use? Did anything surprise you? Did you encounter a number of errors or just a few? How easy was it to determine how to correct the CSS file? Would you recommend this to other students? Why or why not?

Web Research

You've been working a lot with navigation links in this chapter. There is one aspect that we did not discuss—using background images in navigation links. There are numerous tutorials on the Web that present this technique. Visit the following sites and choose a tutorial you find easy to read.

- http://www.cssplay.co.uk/menus/menu5teen.html
- http://superfluousbanter.org/archives/2004/05/navigation-matrix-reloaded
- http://www.alistapart.com/articles/sprites
- http://www.wpdfd.com/issues/73/film-strip_rollovers_a_simpler_way_to_do_rollovers
- http://css.maxdesign.com.au/listutorial/roll_introduction.htm
- http://www.shapeshed.com/journal/overlapping_tabbed_navigation_in_css/

Choose and follow one of the tutorials listed above. Create a Web page that uses this new technique. The Web page should provide the URL of your tutorial, the name of the Web site, and a description of the new technique you discovered. Place your name in the e-mail address at the bottom of the Web page. Print the external style sheet (if you used one), the Web page source code (from Notepad), and the browser view of your Web page.

Focus On Web Design

Take a few moments and visit the CSS Zen Garden at http://www.csszengarden.com. Explore the site and note the widely different designs. What thought processes and decisions are needed as a person creates a new design for this site? Visit http://www.stopdesign.com/articles/process, http://www.mikepick.com/news/archives/000086.html, or http://www.bobbyvandersluis.com/articles/gardenparty.php for a behind-the-scenes look at how Web developers have approached this task. Reflect on their stories and suggestions. Write a one page (double-spaced) essay that describes ideas about the design process you'll be able to use as you begin to design Web sites for personal or professional use. Be sure to include the URL of the resources you used.

WEB SITE CASE STUDY:
Navigation Links in a List

Each of the following case studies continues throughout most of the text. This chapter configures the main navigation in your Web sites to utilize an unordered list.

JavaJam Coffee House

See Chapter 2 for an introduction to the JavaJam Coffee House Case Study. Figure 2.26 shows a site map for the JavaJam Web site. The pages were created in earlier chapters. You will use the existing Web site in the javajamcss folder (unless your instructor specifies otherwise) as a start and create a new version that configures the main navigation using an unordered list.

Hands-On Practice Case

1. Review Section 7.2 CSS Pseudo-classes and Links and Section 7.3 CSS Navigation Layout using Lists.
2. Modify the javajam.css file as needed to configure the main navigation links in an unordered list without "bullets". Also configure the main navigation links to change color when a mouse hovers over them.
3. Modify the index.html, menu.html, and music.html Web pages to display the main navigation links in an unordered list.
4. Launch a browser and test the pages in the javajamcss folder. Modify your java.css file as needed to configure your pages. Be sure to test in more than one browser.

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. You will use the existing Web site in the fishcreekcss folder (unless your instructor specifies otherwise) as a start and create a new version that configures the main navigation using an unordered list.

Hands-On Practice Case

1. Review Section 7.3 CSS Navigation Layout using Lists.
2. Modify the fishcreek.css file as needed to configure the main navigation links in an unordered list without "bullets". *Hint*: To eliminate the extra space on the left side of the fish navigation links, use CSS to configure the unordered list to have 0 margin and padding on the left side.
3. Modify the index.html, services.html, and askvet.html Web pages to display the main navigation links in an unordered list.
4. Launch a browser and test the pages in the fishcreekcss folder. Modify your fishcreek.css file as needed to configure your pages. Be sure to test in more than one browser.

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. You will use the existing Web site in the paintercss folder (unless your instructor specifies otherwise) as a start and create a new version that configures the main navigation using an unordered list.

Hands-On Practice Case

1. Review Section 7.2 CSS Pseudo-classes and Links and Section 7.3 CSS Navigation Layout using Lists.
2. Modify the painter.css file as needed to configure the main navigation links in an unordered list without "bullets". Also configure the main navigation links to change color when a mouse hovers over them. *Hint*: To eliminate the extra space

on the left side of the main navigation links, use CSS to configure the unordered list to have 0 margin and padding on the left side.

3. Modify the index.html, services.html, and testimonials.html Web pages to display the main navigation links in an unordered list.
4. Launch a browser and test the pages in the paintercss folder. Modify your painter.css file as needed to configure your pages. Be sure to test in more than one browser.

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. You will use the existing Web site as in the primecss folder (unless your instructor specifies otherwise) as a start and create a new version that configures the main navigation using an unordered list.

Hands-On Practice Case

1. Review Section 7.2 CSS Pseudo-classes and Links and Section 7.3 CSS Navigation Layout using Lists.
2. Modify the prime.css file as needed to configure the main navigation links in an unordered list without "bullets". Also remove the image buttons and, instead, configure CSS buttons with text that changes color when the mouse hovers over them. *Hint*: To eliminate the extra space on the left side of the main navigation links, use CSS to configure the unordered list to have 0 margin and padding on the left side.
3. Modify the index.html, financing.html, and listings.html Web pages to display the main navigation links in an unordered list.
4. Launch a browser and test the pages in the primecss folder. Modify your prime.css file as needed to configure your pages. Be sure to test in more than one browser.

Web Project

See Chapters 5 and 6 for an introduction to the Web Project case. You will modify the main navigation to use an unordered list. If appropriate, also add interactivity to the main navigation area with CSS pseudo-classes.

Hands-On Practice Case

1. Review Section 7.2 CSS Pseudo-Classes and Links and Section 7.3 CSS Navigation Layout Using Lists.
2. Modify your project's external CSS file and Web page files as needed to configure the main navigation in an unordered list.
3. Optional: Modify your project's external style sheet to configure CSS link and hover pseudo-classes for your main navigation hyperlinks.
4. Launch a browser and test the Web pages. Modify your files as needed to configure your pages. Be sure to test in more than one browser.