8

Tables

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

- Describe the recommended use of a table on a web page
- Create a basic table with the table, table row, table header, and table cell elements
- Configure table sections with the thead, tbody, and tfoot elements

- Increase the accessibility of a table
- Style an HTML table with CSS
- Describe the purpose of CSS structural pseudo-classes

While back in the day tables were often used to format the layout of a web page, CSS is the page layout tool of choice for modern web developers. In this chapter, you'll become familiar with coding HTML tables to organize information on a web page.

8.1 Table Overview

VideoNote
Configure a Table

The purpose of a table is to organize information. In the past, before CSS was well supported by browsers, tables were also used to format web page layouts. An HTML table is composed of rows and columns, like a spreadsheet. Each individual table `cell` is at the intersection of a specific row and column.

Figure 8.1 Table with three rows and three columns

- Each table begins with a `<table>` tag and ends with a `</table>` tag.
- Each table row begins with a `<tr>` tag and ends with a `</tr>` tag.
- Each cell (table data) begins with a `<td>` tag and ends with a `</td>` tag.
- Table cells can contain text, graphics, and other HTML elements.

Figure 8.1 shows a sample table with three rows and three columns. The sample HTML for this table is

```
<table>
  <tr>
    <td>Name</td>
    <td>Birthday</td>
    <td>Phone</td>
  </tr>
  <tr>
    <td>Jack</td>
    <td>5/13</td>
    <td>857-555-5555</td>
  </tr>
  <tr>
    <td>Sparky</td>
    <td>11/28</td>
    <td>303-555-5555</td>
  </tr>
</table>
```

Notice how the table is coded row by row. Also, each row is coded cell by cell. This attention to detail is crucial for the successful use of tables. An example can be found in the student files (chapter8/table1.html).

Table Element

A **table element** is a block display element that contains tabular information. The table begins with a **`<table>`** tag and ends with a **`</table>`** tag. See Table 8.1 for common attributes of the table element. Notice that most of the attributes listed in Table 8.1 are obsolete in HTML5 and should be avoided. Although these attributes are obsolete in HTML5, as you work with the Web you'll see many web pages coded with earlier versions of HTML and it is useful to be aware of even obsolete attributes. Modern web developers prefer to configure the style of tables with CSS properties instead of with HTML attributes. A prominent attribute that remains in use is the border attribute.

Table 8.1 Table element attributes

Attribute	Value	Purpose
align	left (default), right, center	Horizontal alignment of the table (obsolete in HTML5)
bgcolor	Valid color value	Background color of the table (obsolete in HTML5)
border	0 1–100	Default; there is no visible border Visible border with pixel width specified
cellpadding	Numeric value	Specifies the number of pixels of padding between the content of a table cell and its border (obsolete in HTML5)
cellspacing	Numeric value	Specifies the number of pixels of space between the borders of each cell in a table (obsolete in HTML5)
summary	Text description	Provides accessibility with a text description that gives an overview of and the context for the information in the table (obsolete in HTML5)
title	Text description	A brief text description that provides an overview of the table; may be displayed by some browsers as a tooltip
width	Numeric value or percentage	Specifies the width of the table (obsolete in HTML5)

The border Attribute

In earlier versions of HTML (such as HTML4 and XHTML), the purpose of the border attribute was to indicate the presence and the width of a visible table border. The border attribute is used differently in HTML5. When following HTML5 syntax, code border="1" to cause the browser to render default borders around the table and table cells, as shown in Figure 8.1. CSS is used to style the border of a table. You'll get practice styling a table with CSS later in the chapter.

Table Captions

The **caption element** is often used with a table to describe its contents. The caption begins with a **<caption>** tag and ends with a </caption> tag. The text contained within the caption element displays on the web page above the table, although you'll see later in the chapter that you can configure the placement with CSS. The table shown in Figure 8.2 uses the caption element to set the table caption to "Bird Sightings". Notice that the caption element is coded on the line immediately after the opening <table> tag. An example can be found in the student files (chapter8/table2.html). The HTML for the table is

Figure 8.2 The caption for this table is "Bird Sightings"

```
<table border="1">
<caption>Bird Sightings</caption>
    <tr>
    <td>Name</td>
    <td>Date</td>
  </tr>
  <tr>
    <td>Bobolink</td>
    <td>5/25/10</td>
  </tr>
  <tr>
    <td>Upland Sandpiper</td>
    <td>6/03/10</td>
  </tr>
</table>
```

8.2 Table Rows, Cells, and Headers

Table Row Element

The **table row element** configures a row within a table on a web page. The table row begins with a **`<tr>`** tag and ends with a `</tr>` tag. Table 8.2 shows obsolete attributes of the table row element. Web pages coded with older versions of HTML may use these obsolete attributes. Modern web developers configure alignment and background color with CSS instead of with HTML.

Table 8.2 Obsolete attributes of the table row element

Attribute	Value	Purpose
`align`	`left` (default), `right`, `center`	Horizontal alignment of the table (obsolete in HTML5)
`bgcolor`	Valid color value	Background color of the table (obsolete in HTML5)

Table Data Element

The **table data element** configures a cell within a row in a table on a web page. The table cell begins with a **`<td>`** tag and ends with a `</td>` tag. See Table 8.3 for attributes of the table data cell element. Some attributes are obsolete and should be avoided. You'll explore configuring table styles with CSS later in the chapter.

Table 8.3 Attributes of the table data and table header cell elements

Attribute	Value	Purpose
`align`	`left` (default), `right`, `center`	Horizontal alignment of the table (obsolete in HTML5)
`bgcolor`	Valid color value	Background color of the table (obsolete in HTML5)
`colspan`	Numeric value	The number of columns spanned by a cell
`headers`	The id value(s) of a column or row header cell	Associates the table data cells with table header cells; may be accessed by screen readers
`height`	Numeric value or percentage	Height of the cell (obsolete in HTML5)
`rowspan`	Numeric value	The number of rows spanned by a cell
`scope`	`row`, `col`	The scope of the table header cell contents (row or column); may be accessed by screen readers
`valign`	`top`, `middle` (default), `bottom`	The vertical alignment of the contents of the cell (obsolete in HTML5)
`width`	Numeric value or percentage	Width of the cell (obsolete in HTML5)

Table Header Element

Figure 8.3 Using `<th>` tags to indicate column headings

The **table header element** is similar to a table data element and configures a cell within a row in a table on a web page. Its purpose is to configure column and row headings. Text displayed within a table header element is centered and bold. The table header element begins with a **`<th>`** tag and ends with a `</th>` tag. See Table 8.3 for common attributes of the table header element. Figure 8.3 shows a table with column headings configured by `<th>` tags. The HTML for

the table shown in Figure 8.3 is as follows (also see chapter8/table3.html in the student files). Notice that the first row uses `<th>` instead of `<td>` tags.

```
<table border="1">
  <tr>
    <th>Name</th>
    <th>Birthday</th>
    <th>Phone</th>
  </tr>
  <tr>
    <td>Jack</td>
    <td>5/13</td>
    <td>857-555-5555</td>
  </tr>
  <tr>
    <td>Sparky</td>
    <td>11/28</td>
    <td>303-555-5555</td>
  </tr>
</table>
```

Hands-On Practice 8.1

In this Hands-On Practice, you will create a web page similar to Figure 8.4 that describes two schools you have attended. Use the caption "School History Table." The table has three rows and three columns. The first row will have table header elements with the headings School Attended, Years, and Degree Awarded. You will complete the second and third rows with your own information within table data elements.

To get started, launch a text editor and open chapter2/template.html in the student files. Save the file as mytable.html. Modify the title element. Use table, table row, table header, table data, and caption elements to configure a table similar to Figure 8.4.

Figure 8.4 School History Table

School Attended	Years	Degree Awarded
Schaumburg High School	2008—2012	High School Diploma
Harper College	2012—2014	Web Developer Certificate

Hints:
- The table has three rows and three columns.
- To configure a border, use `border="1"` on the `<table>` tag.
- Use the table header element for the cells in the first row.

Save your file and display it in a browser. A sample solution is found in the student files (chapter8/ 8.1/index.html).

8.3 Span Rows and Columns

You can alter the gridlike look of a table by applying the `colspan` and `rowspan` attributes to table data or table header elements. As you get into more complex table configurations like these, be sure to sketch the table on paper before you start typing the HTML.

The `colspan` Attribute

Figure 8.5 Table with a row that spans two columns

The **`colspan` attribute** specifies the number of columns that a cell will occupy. Figure 8.5 shows a table cell that spans two columns.

The HTML for the table is

```
<table border="1">
  <tr>
    <td colspan="2">This spans two columns</td>
  </tr>
  <tr>
    <td>Column 1</td>
    <td>Column 2</td>
  </tr>
</table>
```


Figure 8.6 Table with a column that spans two rows

The `rowspan` Attribute

The **`rowspan` attribute** specifies the number of rows that a cell will occupy. An example of a table cell that spans two rows is shown in Figure 8.6.

The HTML for the table is

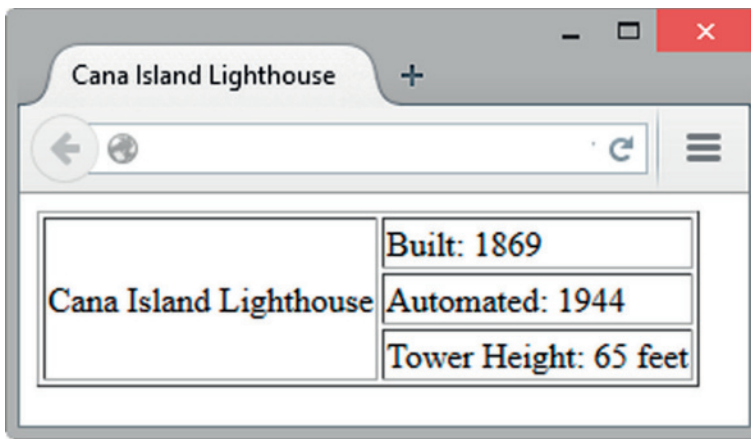```
<table border="1">
  <tr>
    <td rowspan="2">This spans two rows</td>
    <td>Row 1 Column 2</td>
  </tr>
  <tr>
    <td>Row 2 Column 2</td>
  </tr>
</table>
```

An example of the tables in Figures 8.5 and 8.6 can be found in the student files (chapter8/table4.html).

# Hands-On Practice 8.2

You will practice with the `rowspan` attribute in this Hands-On Practice. To create the web page shown in Figure 8.7, launch a text editor and open chapter2/template.html in the student files. Save the file as myrowspan.html. Modify the title element. Use table, table row, table head, and table data elements to configure the table.



**Figure 8.7**  Practice with the `rowspan` attribute

1. Code the opening `<table>` tag. Configure a border with `border="1"`.

2. Begin the first row with a `<tr>` tag.

3. The table data cell with "Cana Island Lighthouse" spans three rows. Code a table data element. Use the `rowspan="3"` attribute.

4. Code a table data element that contains the text "Built: 1869".

5. End the first row with a `</tr>` tag.

6. Begin the second row with a `<tr>` tag. This row will only have one table data element because the cell in the first column is already reserved for "Cana Island Lighthouse".

7. Code a table data element that contains the text "Automated: 1944".

8. End the second row with a `</tr>` tag.

9. Begin the third row with a `<tr>` tag. This row will only have one table data element because the cell in the first column is already reserved for "Cana Island Lighthouse".

10. Code a table data element that contains the text "Tower Height: 65 feet".

11. End the third row with a `</tr>` tag.

12. Code the closing `</table>` tag.

Save the file and view it in a browser. A sample solution is found in the student files (chapter8/8.2/index.html). Notice how the "Cana Island Lighthouse" text is vertically aligned in the middle of the cell, which is the default vertical alignment. You can modify the vertical alignment using CSS—see section "8.5 Style a Table with CSS" later in this chapter.

# 8.4  Configure an Accessible Table

**Focus on Accessibility**

Tables can be useful to organize information on a web page, but what if you couldn't see the table and were relying on assistive technology like a screen reader to read the table to you? You'd hear the contents of the table just the way it is coded—row by row, cell by cell. This might be difficult to understand. This section discusses coding techniques to improve the accessibility of tables.

For a simple informational data table like the one shown in Figure 8.8, the W3C Web Accessibility Initiative (WAI) Web Content Accessibility Guidelines 2.0 (WCAG 2.0) recommend the following:

- Use table header elements (`<th>` tags) to indicate column or row headings.

- Use the caption element to provide a text title or caption for the table.

**Figure 8.8**  This simple data table uses `<th>` tags and the caption element to provide accessibility

An example web page is in the student files (chapter8/table5.html). The HTML is

```
<table border="1">
<caption>Bird Sightings</caption>
  <tr>
    <th>Name</th>
    <th>Date</th>
  </tr>
  <tr>
    <td>Bobolink</td>
    <td>5/25/10</td>
  </tr>
  <tr>
    <td>Upland Sandpiper</td>
    <td>6/03/10</td>
  </tr>
</table>
```

However, for more complex tables, the W3C recommends specifically associating the table data cell values with their corresponding headers. The technique that is recommended uses the id attribute (usually in a `<th>` tag) to identify a specific header cell and the **headers attribute** in a `<td>` tag. The code to configure the table in Figure 8.8 using headers and ids is as follows (also see chapter8/table6.html in the student files):

```
<table border="1">
<caption>Bird Sightings</caption>
  <tr>
    <th id="name">Name</th>
    <th id="date">Date</th>
  </tr>
  <tr>
    <td headers="name">Bobolink</td>
    <td headers="date">5/25/10</td>
  </tr>
```

```
   <tr>
     <td headers="name">Upland Sandpiper</td>
     <td headers="date">6/03/10</td>
   </tr>
</table>
```

FAQ   What about the `scope` attribute?

The **`scope` attribute** specifies the association of table cells and table row or column headers.

It is used to indicate whether a table cell is a header for a column (`scope="col"`) or row (`scope="row"`). An example of the code for the table in Figure 8.8 that uses this attribute is as follows (also see chapter8/table7.html in the student files):

```
<table border="1">
<caption>Bird Sightings</caption>
   <tr>
      <th scope="col">Name</th>
      <th scope="col">Date</th>
   </tr>
   <tr>
      <td>Bobolink</td>
      <td>5/25/10</td>
   </tr>
   <tr>
      <td>Upland Sandpiper</td>
      <td>6/03/10</td>
   </tr>
</table>
```

As you reviewed the code sample in the previous page, you may have noticed that using the `scope` attribute to provide accessibility requires less coding than implementing the `headers` and `id` attributes. However, because of inconsistent screen reader support of the `scope` attribute, the WCAG 2.0 recommendations for coding techniques encourage the use of `headers` and `id` attributes rather than the `scope` attribute.

Checkpoint 8.1

1. What is the purpose of using a table on a web page?

2. How is the text contained within a th element displayed by the browser?

3. Describe one coding technique that increases the accessibility of an HTML table.

8.5 Style a Table with CSS

Before CSS was well supported by browsers, it was common practice to configure the visual aesthetic of a table with HTML attributes. The modern approach is to use CSS to style a table. In this section, you'll explore using CSS to style the border, padding, alignment, width, height, vertical alignment, and background of table elements. Table 8.4 lists corresponding CSS properties with the HTML attributes used to style tables.

Table 8.4 CSS properties used to style tables

HTML Attribute	CSS Property

align	To align a table, configure the width and margin properties for the table selector. For example, to center a table with a width of 75% of the container element, use
	`table { width: 75%;`
	`        margin: auto; }`
	To align items within table cells, use `text-align`

width	width
height	height
cellpadding	padding
cellspacing	**border-spacing**; a numeric value (px or em) or percentage. If you set a value to 0, omit the unit. One numeric value with unit (px or em) configures both horizontal and vertical spacing. Two numeric values with unit (px or em): The first value configures the horizontal spacing and the second value configures the vertical spacing. **border-collapse** configures the border area. The values are `separate` (default) and `collapse`. Use `border-collapse: collapse;` to remove extra space between table and table cell borders.
bgcolor	background-color
valign	**vertical-align** specifies the vertical placement of content. The values are numeric pixel or percentage, `baseline` (default), `sub` (subscript), `super` (superscript), `top`, `text-top`, `middle`, `bottom`, and `text-bottom`
border	border, border-style, border-spacing
none	background-image
none	**caption-side** specifies the placement of the caption. Values are `top` (default) and `bottom`

Hands-On Practice 8.3

In this Hands-On Practice, you will code CSS style rules to configure an informational table on a web page. Create a folder named ch8table. Copy the starter.html file from the chapter8 folder to your ch8table folder. We'll use embedded styles for ease of editing and testing your page. Open the starter.html file in a browser. The display should look similar to the one shown in Figure 8.9.

Lighthouse Island Bistro Specialty Coffee Menu		
Specialty Coffee	**Description**	**Price**
Lite Latte	Indulge in a shot of espresso with steamed, skim milk.	$3.50
Mocha Latte	Choose dark or milk chocolate with steamed milk.	$4.00
MCP Latte	A luscious mocha latte with caramel and pecan syrup.	$4.50

Figure 8.9 This table is configured with HTML

Launch a text editor and open starter.html from your ch8table folder. Locate the style tags in the head section. You will code embedded CSS in this Hands-On Practice. Place your cursor on the blank line between the style tags.

1. Configure the table element selector to be centered, have a dark blue, 5 pixel border, and have a width of 600px.

```
table { margin: auto;
        border: 5px solid #000066;
        width: 600px; }
```

Save the file as menu.html. Open your page in a browser. Notice that there is a border surrounding the entire table but not surrounding each table cell.

2. Configure the td and th element selectors with a border, padding, and Arial or the default sans-serif font typeface.

```
td, th { border: 1px solid #000066;
         padding: 0.5em;
         font-family: Arial, sans-serif; }
```

Save the file and open your page in a browser. Each table cell should now be outlined with a border and should display text in a sans-serif font.

3. Notice the empty space between the borders of the table cells. This empty space can be eliminated with the **border-spacing property**. Add a `border-spacing: 0;` declaration to the table element selector. Save the file and open your page in a browser.

4. Configure the caption to be displayed with Verdana or the default sans-serif font typeface, bold font weight, 1.2em font size, and 0.5em of bottom padding. Configure a style rule as follows:

```
caption { font-family: Verdana, sans-serif;
          font-weight: bold;
          font-size: 1.2em;
          padding-bottom: 0.5em; }
```

5. Let's experiment and configure background colors for the rows instead of cell borders. Modify the style rule for the td and th element selectors, remove the border declaration, and set `border-style` to none. The new style rule for the cells is

```
td, th { padding: 0.5em;
         border-style: none;
         font-family: Arial, sans-serif; }
```

6. Create a new class called `altrow` that sets a background color.
`.altrow { background-color: #eaeaea; }`

7. Modify the `<tr>` tags in the HTML. Assign the second and fourth `<tr>` tags to the `altrow` class. Save the file and open your page in a browser. The table area should look similar to the one shown in Figure 8.10.

Lighthouse Island Bistro Specialty Coffee Menu

Specialty Coffee	Description	Price
Lite Latte	Indulge in a shot of espresso with steamed, skim milk.	$3.50
Mocha Latte	Choose dark or milk chocolate with steamed milk.	$4.00
MCP Latte	A luscious mocha latte with caramel and pecan syrup.	$4.50

Figure 8.10 Rows are configured with alternating background colors

Notice how the background color of the alternate rows adds subtle interest to the web page. In this Hands-On Practice, you configured the display of an HTML table using CSS. Compare your work with the sample in the student files (chapter8/8.3/index.html).

FAQ   **Is there a way to create a table-like page layout with CSS?**

Not yet, but there will be more options for CSS page layout in the future! The W3C is developing several new CSS page layout modules which allow web developers to generate table-like grid layouts, including:

- **Flexible Box Layout** (referred to as Flexbox) features an enhanced box model that allows for flow of boxes with flexible sizes. http://www.w3.org/TR/css3-flexbox/

- **Multi-column Layout** provides for the flow of content into multiple columns. http://www.w3.org/TR/css3-multicol/

- **Grid Layout** structures a web page into rows and columns within a grid. http://www.w3.org/TR/css3-grid-layout/

Be aware that it will be some time before these innovative technologies are ready for use in commercial web pages. Visit http://caniuse.com/ for information about current browser support of these and other CSS coding techniques.

8.6 CSS3 Structural Pseudo-Classes

In the previous section, you configured CSS and applied a class to every other table row to configure alternating background colors, often referred to as "zebra striping." You may have found this to be a bit inconvenient and wondered if there was a more efficient method. Well, there is! CSS3 **structural pseudo-class selectors** allow you to select and apply classes to elements based on their position in the structure of the document, such as

every other row. CSS3 pseudo-classes are supported by current versions of Firefox, Opera, Chrome, Safari, and Internet Explorer. Table 8.5 lists common CSS3 structural pseudo-class selectors and their purpose.

Table 8.5 Common CSS3 structural pseudo-classes

Pseudo-class	Purpose
`:first-of-type`	Applies to the first element of the specified type
`:first-child`	Applies to the first child of an element (CSS2 selector)
`:last-of-type`	Applies to the last element of the specified type
`:last-child`	Applies to the last child of an element
`:nth-of-type(n)`	Applies to the "nth" element of the specified type Values: a number, odd, or even

To apply a pseudo-class, write it after the selector. The following code sample will configure the first item in an unordered list to display with red text.

```
li:first-of-type { color: #FF0000; }
```

Hands-On Practice 8.4

In this Hands-On Practice, you will rework the table you configured in Hands-On Practice 8.3 to use CSS3 structural pseudo-class selectors to configure color.

1. Launch a text editor and open menu.html in your ch8table folder (it can also be found as chapter8/8.3/index.html in the student files). Save the file as menu2.html.

2. View the source code and notice that the second and fourth tr elements are assigned to the `altrow` class. You won't need this class assignment when using CSS3 structural pseudo-class selectors. Delete `class="altrow"` from the tr elements.

3. Examine the embedded CSS and locate the `altrow` class. Change the selector to use a structural pseudo-class that will apply the style to the even-numbered table rows. Replace `.altrow` with `tr:nth-of-type(even)` as shown in the following CSS declaration:

   ```
   tr:nth-of-type(even) { background-color: #eaeaea; }
   ```

4. Save the file and open your page in a browser. The table area should look similar to the one shown in Figure 8.10 if you are using a modern browser that supports CSS3 structural pseudo-classes.

5. Let's configure the first row to have a dark blue background (#006) and light gray text (#eaeaea) with the `:first-of-type` structural pseudo-class. Add the following to the embedded CSS:

   ```
   tr:first-of-type { background-color: #006;
                      color: #eaeaea; }
   ```

6. Save the file and open your page in a browser. The table area should look similar to the one shown in Figure 8.11 if you are using a modern browser that supports CSS3 structural pseudo-classes. A sample solution is available in the student files (chapter8/8.4/index.html).

Lighthouse Island Bistro Specialty Coffee Menu

Specialty Coffee	Description	Price
Lite Latte	Indulge in a shot of espresso with steamed, skim milk.	$3.50
Mocha Latte	Choose dark or milk chocolate with steamed milk.	$4.00
MCP Latte	A luscious mocha latte with caramel and pecan syrup.	$4.50

Figure 8.11 CSS3 pseudo-class selectors style the table rows

CSS structural pseudo-classes are convenient to use, but be aware that Internet Explorer 8 and earlier versions do not support this technology. Although browser support will increase in the future, today it's best to apply these pseudo-classes with progressive enhancement in mind.

8.7 Configure Table Sections

There are a lot of configuration options for coding tables. Table rows can be put together into three types of groups: table head with `<thead>`, table body with `<tbody>`, and table footer with `<tfoot>`.

These groups can be useful when you need to configure the areas in the table in different ways, using either attributes or CSS. The `<tbody>` tag is required if you configure a `<thead>` or `<tfoot>` area, although you can omit either the table head or table footer if you like. When you use table row groups, the `<thead>` and `<tfoot>` sections must be coded *before* the `<tbody>` section to pass W3C XHTML validation. The code sample in this section uses HTML5 syntax with the `<tfoot>` coded after the `<tbody>`, which is more intuitive.

The following code sample (see chapter8/tfoot.html in the student files) configures the table shown in Figure 8.12 and demonstrates the use of CSS to configure a table head, table body, and table footer with different styles.

The CSS styles a centered 200-pixel-wide table with a caption that is rendered in large, bold font; a table head section with a light-gray (#eaeaea) background color; a table body section styled with slightly smaller text (.90em) using Arial or sans-serif font; table body td element selectors set to display with some left padding and a dashed bottom border; and a table footer section that has centered, bold text and a light-gray background color (#eaeaea). The CSS code is

Time Sheet

Day	Hours
Monday	4
Tuesday	3
Wednesday	5
Thursday	3
Friday	3
Total	**18**

Figure 8.12 CSS configures the thead, tbody, and tfoot element selectors

```css
table { width: 200px;
        margin: auto; }
caption { font-size: 2em;
          font-weight: bold; }
thead { background-color: #eaeaea; }
tbody { font-family: Arial, sans-serif;
        font-size: .90em; }
tbody td { border-bottom: 1px #000033 dashed;
           padding-left: 25px; }
tfoot { background-color: #eaeaea;
        font-weight: bold;
        text-align: center; }
```

The HTML for the table is

```html
<table>
<caption>Time Sheet</caption>
   <thead>
      <tr>
         <th id="day">Day</th>
         <th id="hours">Hours</th>
      </tr>
   </thead>
   <tbody>
      <tr>
         <td headers="day">Monday</td>
         <td headers="hours">4</td>
      </tr>
      <tr>
         <td headers="day">Tuesday</td>
         <td headers="hours">3</td>
      </tr>
      <tr>
         <td headers="day">Wednesday</td>
         <td headers="hours">5</td>
      </tr>
      <tr>
         <td headers="day">Thursday</td>
         <td headers="hours">3</td>
      </tr>
      <tr>
         <td headers="day">Friday</td>
         <td headers="hours">3</td>
      </tr>
   </tbody>
   <tfoot>
      <tr>
         <td headers="day">Total</td>
         <td headers="hours">18</td>
      </tr>
   </tfoot>
</table>
```

This example demonstrates the power of CSS in styling documents. The `<td>` tags within each table row group element selector (thead, tbody, and tfoot) inherited the font styles configured for their parent group element selector. Notice how a descendant selector configures the padding and border only for `<td>` tags that are contained within (actually, "children of") the `<tbody>` element. Sample code is located in the student files (chapter8/tfoot.html). Take a few moments to explore the web page code and open the page in a browser.

Checkpoint 8.2

1. Describe a reason to configure a table with CSS properties instead of HTML attributes.

2. List three elements that are used to group table rows.

Chapter Summary

This chapter introduces both the HTML techniques used to code tables to organize information and the CSS properties that configure the display of tables on web pages. Visit the textbook website at http://www.webdevfoundations.net for examples, the links listed in this chapter, and updated information.

Key Terms

<caption>
<table>
<tbody>
<td>
<tfoot>
<th>
<thead>
<tr>
:first-child
:first-of-type
:last-child
:last-of-type

:nth of type
align attribute
border attribute
border-collapse property
border-spacing property
caption element
caption-side property
cell
cellpadding attribute
cellspacing attribute
colspan attribute
headers attribute

rowspan attribute
scope attribute
structural pseudo-class selectors
summary attribute
table element
table data element
table header element
table row element
title attribute
vertical-align property
valign attribute

Review Questions

Multiple Choice

1. Which HTML tag pair is used to specify table headings?
   a. `<td> </td>`
   b. `<th> </th>`
   c. `<head> </head>`
   d. `<tr> </tr>`

2. Which CSS property specifies the background color of a table?
   a. `background`
   b. `bgcolor`
   c. `background-color`
   d. `table-color`

3. Which HTML tag pair is used to group rows in the footer of a table?
   a. `<footer> </footer>`
   b. `<tr> </tr>`
   c. `<tfoot> </tfoot>`
   d. `<td> </td>`

4. Which HTML element uses a `border` attribute to display a table with a border?
   a. `<td>`
   b. `<tr>`
   c. `<table>`
   d. `<tableborder>`

5. Which CSS property eliminates the space between the borders on table cells?
   a. `border-style`
   b. `padding`
   c. `border-spacing`
   d. `cellspacing`

6. Which HTML attribute associates a table data cell with a table header cell?
   a. `head`
   b. `headers`
   c. `align`
   d. `th`

7. Which HTML tag pair is used to begin and end a table row?
   a. `<td> </td>`
   b. `<tbody> </tbody>`
   c. `<table> </table>`
   d. `<tr> </tr>`

8. Which of the following is the intended use of tables on web pages?
   a. configuring the layout of an entire page
   b. organizing information
   c. forming hyperlinks

387

d. configuring a resume

9. Which CSS property specifies the distance between the cell text and the cell border?

   a. border-style
   b. padding
   c. border-spacing
   d. cellpadding

10. Which CSS pseudo-class applies to the first element of a specified type?

   a. :first-of-type
   b. :first-type
   c. :first-child
   d. :first

Fill in the Blank

11. The CSS _____ property can be used to configure the color and width of a table border.

12. The _____ CSS property specifies the vertical alignment of the contents of a cell in a table.

13. Use the _____ attribute to configure a table cell to occupy more than one row in the table.

14. _____ is an attribute of the td element that associates the table data cell with a table header cell.

15. Use the _____ element to provide a brief description of a table that displays on the web 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 HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Predict the Result</title>
<meta charset="utf-8">
</head>
<body>
<table>
   <tr>
      <th>Year</th>
      <th>School</th>
      <th>Major</th>
   </tr>
   <tr>
      <td>2008-2012</td>
      <td>Schaumburg High School</td>
      <td>College Prep</td>
   </tr>
   <tr>
      <td>2012-2016</td>
      <td>Harper College</td>
      <td>Web Development Associates Degree</td>
   </tr>
</table>
</body>
</html>
```

2. **Fill in the Missing Code.** This web page should have a table with a background color of #cccccc and a border. Some CSS properties and values, indicated by "_", are missing. Fill in the missing code.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CircleSoft Web Design</title>
<meta charset="utf-8">
<style>
table {  "_":"_";
         "_":"_"; }
</style>
</head>
<body>
<h1>CircleSoft Web Design</h1>
<table>
<caption>Contact Information</caption>
    <tr>
        <th>Name</th>
        <th>Phone</th>
    </tr>
    <tr>
        <td>Mike Circle</td>
        <td>920-555-5555</td>
    </tr>
</table>
</body>
</html>
```

3. **Find the Error.** Why doesn't the table information display in the order it was coded?

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CircleSoft Web Design</title>
<meta charset="utf-8">
</head>
<body>
<h1>CircleSoft Web Design</h1>
<table>
<caption>Contact Information</caption>
<tr>
    <th>Name</th>
    <th>Phone</th>
</tr>
<tr>
    <tr>Mike Circle</td>
    <td>920-555-5555</td>
</tr>
</table>
</body>
</html>
```

Hands-On Exercises

1. Write the HTML for a two-column table that contains the names of your friends and their birthdays. The first row of the table should span two columns and contain the following heading: Birthday List. Include at least two people in your table.

2. Write the HTML for a three-column table to describe the courses you are taking this semester. The columns should contain the course number, course name, and instructor name. The first row of the table should use `th` tags and contain descriptive headings for the columns. Use the table row grouping tags `<thead>` and `<tbody>` in your table.

3. Write the HTML for a table with three rows, two columns, and no border. The cell in the first column of each row will contain the name of one of your favorite movies. The corresponding cell in the second column of each row will contain a description of the movie. Configure alternating rows to use the background color #CCCCCC.

4. Use CSS to configure a table that has a border around both the entire table and the table cells. Write the HTML to create a table with three rows and two columns. The cell in the first column of each row will contain the name of one of your favorite movies. The corresponding cell in the second column of each row will contain a description of the movie.

5. Modify the table you created in Hands-On Exercise 8.1 to be centered on the page, use a background color of #CCCC99, and display text in Arial or the browser default sans-serif font. Configure this table using CSS instead of obsolete HTML attributes. Place an e-mail link to yourself on the web page. Save the file as mytable.html.

6. Create a web page about your favorite sports team with a two-column table that lists the positions and starting players. Use embedded CSS to style the table border, background color, and center the table on the web page. Place an e-mail link to yourself on the web page. Save the file as sport8.html.

7. Create a web page about your favorite movie that uses a two-column table containing details about the movie. Use CSS to style the table border and background color. Include the following in the table:

   - Title of the movie
   - Director or producer
   - Leading actor
   - Leading actress
   - Rating (G, PG, PG-13, R, NC-17, NR)
   - A brief description of the movie
   - An absolute link to a review about the movie

   Place an e-mail link to yourself on the web page. Save the page as movie8.html.

8. Create a web page about your favorite music CD that uses a four-column table. The column headings should be as follows:

   - **Group:** Place the name of the group and the names of its principal members in this column.
   - **Tracks:** List the title of each music track or song.
   - **Year:** List the year the CD was recorded.
   - **Links:** Place at least two absolute links to sites about the group in this column.

   Include an e-mail link to yourself on the web page. Save the page as band8.html.

9. Create a web page about your favorite recipe. Organize the ingredients and directions in a single table. Use two columns for the ingredients. Use a row that spans two columns to contain the instructions for creating your culinary delight. Place an e-mail link to yourself on the web page. Save the page as recipe8.html.

Web Research

Search the Web and find a web page configured with one or more HTML tables. Print the browser view of the page. Print out the source code of the web page. On the printout, highlight or circle the tags related to tables. On a separate sheet of paper, create some HTML notes by listing the tags and attributes related to tables found on your sample page, along with a brief description of their purpose. Hand in the browser view of the page, source code printout, and your HTML notes page to your instructor.

Focus on Web Design

Good artists view and analyze many paintings. Good writers read and evaluate many books. Similarly, good web designers view and scrutinize many web pages. Search the Web and find two web pages, one that is appealing to you and one that is unappealing to you. Print out each page. Create a web page that answers the following questions for each of your examples:

a. What is the URL of the website?

b. Does this page use tables? If so, for what purpose (page layout, organization of information, or another reason)?

c. Does this page use CSS? If so, for what purpose (page layout, text and color configuration, or another reason)?

d. Is this page appealing or unappealing? List three reasons for your answer.

e. If this page is unappealing, what would you do to improve it?

WEBSITE CASE STUDY

Using Tables

Each of the following case studies continues throughout most of the textbook. This chapter incorporates an HTML table in the case study websites.

JavaJam Coffee House

See Chapter 2 for an introduction to the JavaJam Coffee House case study. Figure 2.30 shows a site map for JavaJam. Use either the Chapter 7 or Chapter 6 JavaJam website as a starting point for this case study. In this case study, you will modify the Menu page

(menu.html) to display information in an HTML table. You will use CSS to style the table. You have three tasks in this case study:

1. Create a new folder for this JavaJam case study.

2. Modify the style sheet (javajam.css) to configure style rules for the new table.

3. Modify the Menu page to use a table to display information as shown in Figure 8.13.

Figure 8.13 Menu page with a table

Hands-On Practice Case Study

Task 1: Create a Folder. Create a folder called javajam8. If you have skipped Chapter 7, copy all the files from your Chapter 6 javajam6 folder into the javajam8 folder. If you have completed Chapter 7, copy all the files from your Chapter 7 javajam7 folder into the javajam8 folder.

Task 2: Configure the CSS. Modify the external style sheet (javajam.css). Open javajam.css in a text editor. Review Figure 8.13 and note the menu descriptions, which are coded in an HTML table. Add style rules to the javajam.css external style sheet to configure a table that is centered, takes up 90% the width of its container, is configured with 0 `border-spacing`, has a #ffffcc background color, has td and th selectors with 10 pixels of padding, and displays a background color of #d2b48c in alternate rows (use a class or the `:nth-of-type` pseudo-class to configure odd table rows). Save the javajam.css file.

Task 3: Modify the Menu Page. Open menu.html in a text editor. The menu descriptions are configured with a description list. Replace the description list with a table that has three rows and two columns. Use th and td elements where appropriate. Configure line breaks as needed for the pricing information. Save your page and test it in a browser. If the page does not display as you intended, review your work, validate the CSS, validate the HTML, modify as needed, and test again.

Fish Creek Animal Hospital

See Chapter 2 for an introduction to the Fish Creek Animal Hospital case study. Figure 2.34 shows a site map for Fish Creek. Use either the Chapter 7 or Chapter 6 Fish Creek website as a starting point for this case study. You will modify the Services

page (services.html) to display information in an HTML table. You will use CSS to style the table. You have three tasks in this case study:

1. Create a new folder for this Fish Creek case study.
2. Modify the style sheet (fishcreek.css) to configure style rules for the new table.
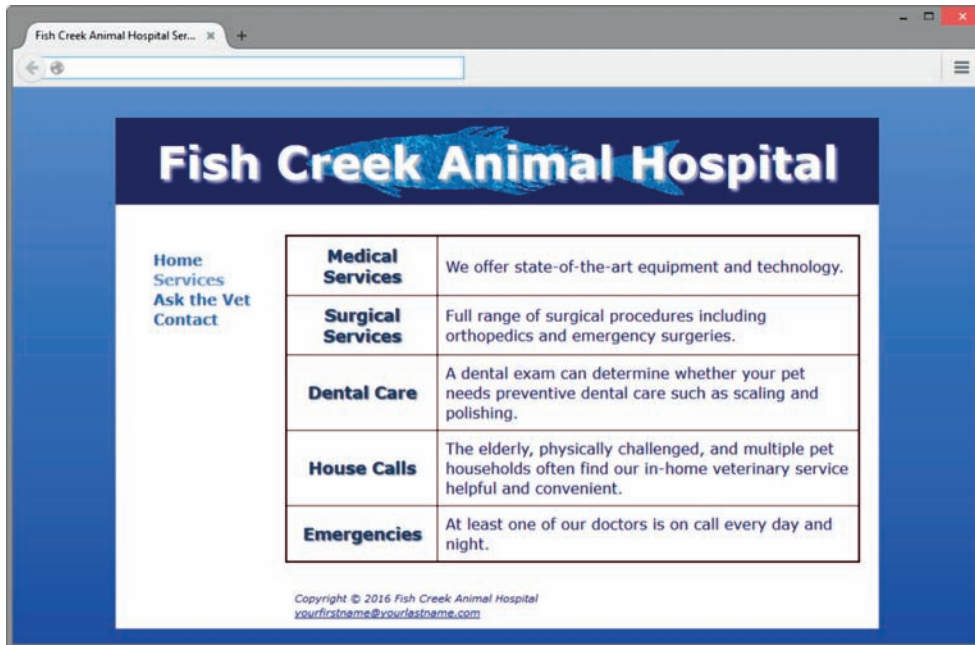3. Modify the Services page to use a table to display information as shown in Figure 8.14.

Figure 8.14 Services page with a table

Hands-On Practice Case Study

Task 1: Create a Folder. Create a folder called fishcreek8. If you have skipped Chapter 7, copy all of the files from your Chapter 6 fishcreek6 folder into the fishcreek8 folder. If you have completed Chapter 7, copy all the files from your Chapter 7 fishcreek7 folder into the fishcreek8 folder.

Task 2: Configure the CSS. Modify the external style sheet (fishcreek.css). Open fishcreek.css in a text editor. Review Figure 8.14 and note the services descriptions, which are coded in an HTML table. Add style rules to the fishcreek.css external style sheet as indicated:

1. Configure a table that has a dark blue, 2 pixel border.
2. Configure the borders in the table to collapse (use `border-collapse: collapse;`).
3. Configure td and th element selectors with 0.5em of padding and a dark blue 1 pixel border.

Save the fishcreek.css file.

Task 3: Modify the Services Page. Open services.html in a text editor. The services descriptions are configured with an unordered list. Replace the unordered list with a table that has five rows and two columns. Use th and td elements where appropriate. *Hint*: Assign the th element to the `category` class. Save your page and test it in a browser. If the page does not display as you intended, review your work, validate the CSS, validate the HTML, modify as needed, and test again.

Pacific Trails Resort

See Chapter 2 for an introduction to the Pacific Trails Resort case study. Figure 2.38 shows a site map for Pacific Trails. Use either the Chapter 7 or Chapter 6 Pacific Trails website as a starting point for this case study. You will modify the Yurts page (yurts.html) to display additional information in an HTML table. You will use CSS to style the table. You have three tasks in this case study:

1. Create a new folder for this Pacific Trails case study.

2. Modify the style sheet (pacific.css) to configure style rules for the new table.

3. Modify the Yurts page to use a table to display information as shown in Figure 8.15.
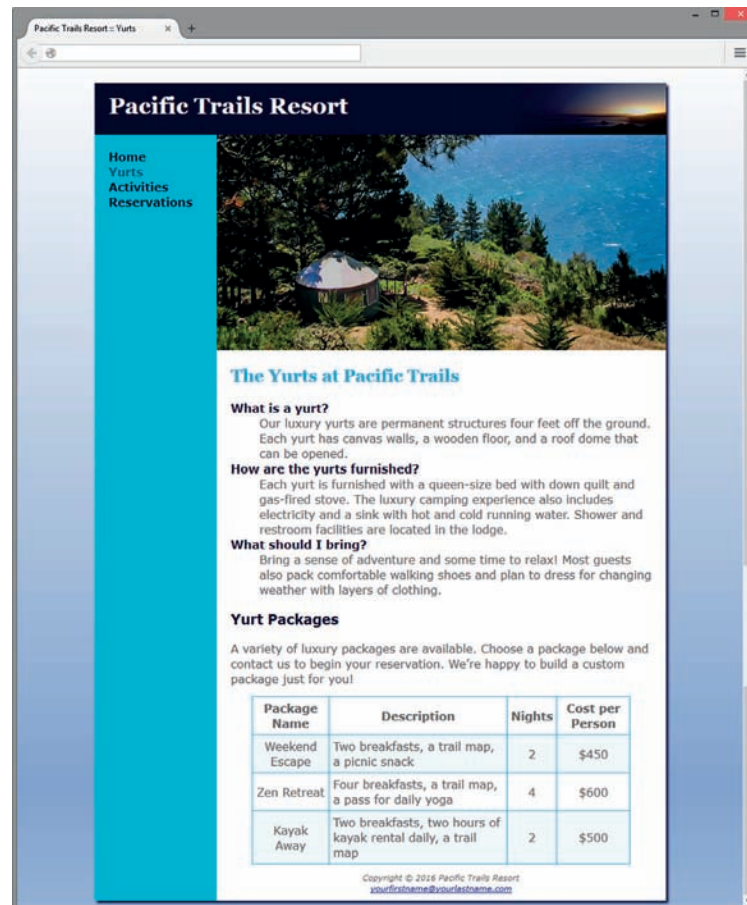

Figure 8.15 Yurts page with a table

Hands-On Practice Case Study

Task 1: Create a Folder. Create a folder called pacific8. If you have skipped Chapter 7, copy all the files from your Chapter 6 pacific6 folder into the pacific8 folder. If you have completed Chapter 7, copy all the files from your Chapter 7 pacific7 folder into the pacific8 folder.

Task 2: Configure the CSS. Modify the external style sheet (pacific.css). Open pacific.css in a text editor. Add styles to configure the table on the Yurts page as shown in Figure 8.15.

1. Configure the table. Code a new style rule for the table element selector that configures a centered table with a 1 pixel blue border (#3399cc) and 90% width.

Also configure the borders in the table to collapse (use `border-collapse: collapse;`).

2. Configure the table cells. Code a new style rule for the td and th element selectors that configure 5 pixels of padding and a 1 pixel blue border (#3399cc).

3. Center the td content. Code a new style rule for the td element selector that centers text (use `text-align: center;`).

4. Configure the `text` class. Notice that the content in the table data cells that contain the text description is not centered. Code a new style rule for a class named `text` that will override the td style rule and left-align the text (use `text-align: left;`).

5. Configure alternate-row background color. The table looks more appealing if the rows have alternate background colors, but it is still readable without them. Apply the `:nth-of-type` CSS3 pseudo-class to configure the even table rows with a light-blue background color (#F5FAFC).

6. Save the pacific.css file.

Task 3: Modify the Yurts Page. Open yurts.html in a text editor.

1. Configure an h3 element with the following text: "Yurt Packages" after the description list in the main element.

2. Below the new h3 element, configure a paragraph with the following text:

    A variety of luxury yurt packages are available. Choose a package below and contact us to begin your reservation. We're happy to build a custom package just for you!

3. You are ready to configure the table. Code a table below the paragraph with four rows and four columns. Use the table, th, and td elements. Assign the td elements that contain the detailed descriptions to the class named `text`. The content for the table is as follows.

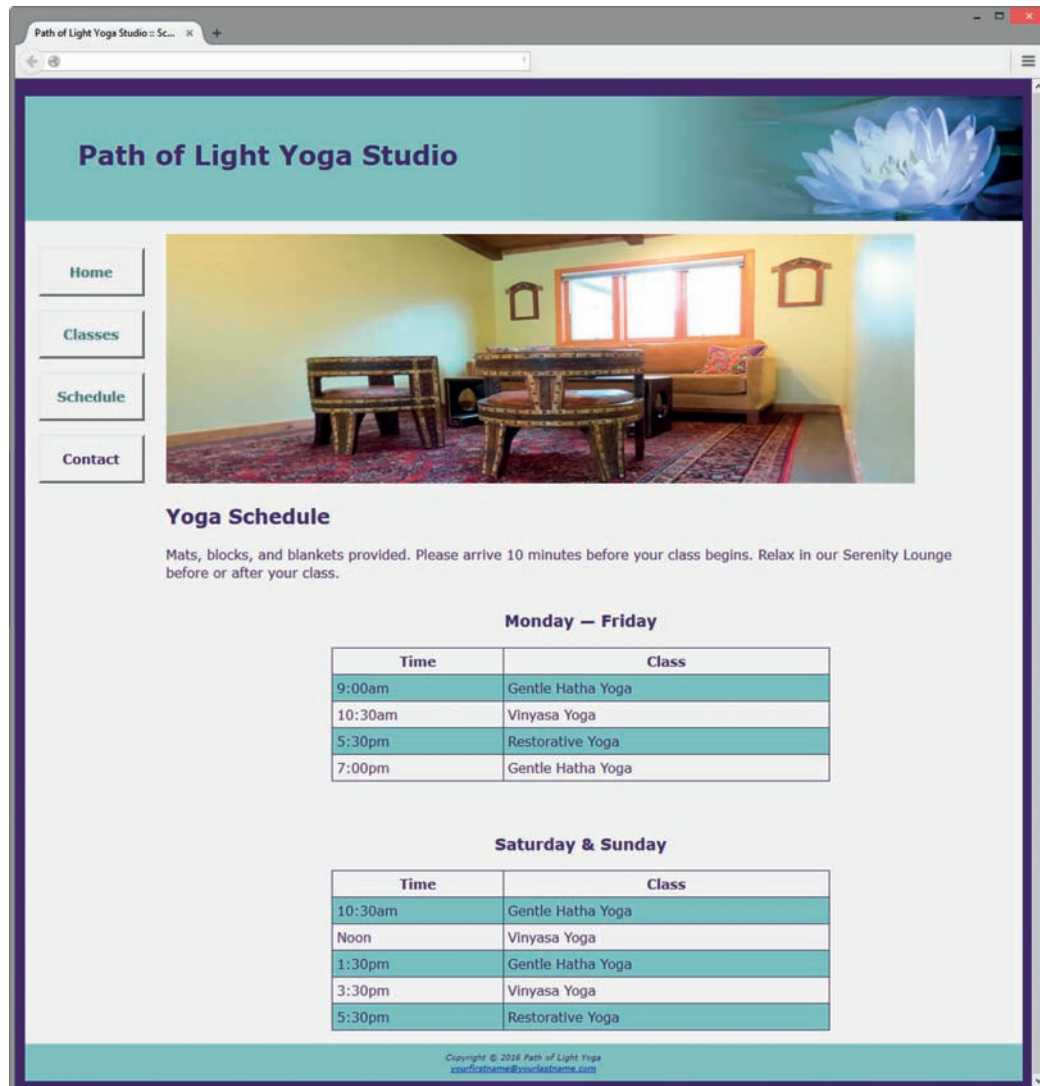| Package Name	Description	Nights	Cost per Person
Weekend Escape	Two breakfasts, a trail map, a picnic snack	2	$450
Zen Retreat	Four breakfasts, a trail map, a pass for daily yoga	4	$600
Kayak Away	Two breakfasts, two hours of kayak rental daily, a trail map	2	$500

Save your page and test it in a browser. If the page does not display as you intended, review your work, validate the CSS, validate the HTML, modify as needed, and test again.

Path of Light Yoga Studio

See Chapter 2 for an introduction to the Path of Light Yoga Studio case study. Figure 2.42 shows a site map for Path of Light Yoga Studio. Use either the Chapter 7 or Chapter 6 Path of Light Yoga Studio website as a starting point for this case study. You will modify the Schedule page (schedule.html) to display the class schedule in an HTML table. You will use CSS to style the table. You have three tasks in this case study:

1. Create a new folder for this Path of Light Yoga Studio case study.

2. Modify the style sheet (yoga.css) to configure style rules for the new table.

3. Modify the Schedule page to use a table to display information as shown in Figure 8.16.

Figure 8.16 Schedule page with tables

Hands-On Practice Case Study

Task 1: Create a Folder. Create a folder called yoga8. If you have skipped Chapter 7, copy all the files from your Chapter 6 yoga6 folder into the yoga8 folder. If you have completed Chapter 7, copy all the files from your Chapter 7 yoga7 folder into the yoga8 folder.

Task 2: Configure the CSS. Modify the external style sheet (yoga.css). Open yoga.css in a text editor. Review Figure 8.16 and note the class schedule information, which is coded in two HTML tables. Add style rules to the yoga.css external style sheet to configure the following:

1. A centered table with 60% width, a 1 pixel purple (#3F2860) border, collapsed borders (use `border-collapse: collapse;`), and a 1em bottom margin.

2. A style for the td and th element selectors that configures 5 pixels of padding and a 1 pixel purple border (#3F2860).

3. Configure alternate-row background color. The table looks more appealing if the rows have alternate background colors, but it is still readable without them. Apply

the :nth-of-type CSS3 pseudo-class to configure the even table rows with a #9BC1C2 background color.

   4. A caption element selector with a 1em margin, bold text, and 120% font size.

Save the yoga.css file.

Task 3: Modify the Schedule Page. Open schedule.html in a text editor. The schedule information currently uses the `<h3>`, `<ul>`, and `<li>` elements. You will rework the page to use two tables to display the schedule information instead of the h3 and unordered list tags. Use a caption element within each table. Note that the table rows have two columns. Configure "Time" and "Class" table headings within each table. Refer to Figure 8.16.

Save your page and test it in a browser. If the page does not display as you intended, review your work, validate the CSS, validate the HTML, modify as needed, and test again.

Web Project

See Chapters 5 and 6 for an introduction to the Web Project case study. You will modify the design of one of the pages to display information in an HTML table. Use CSS to style the table.

Hands-On Practice Case Study

   1. Choose one of your project web pages to modify. Sketch a design of the table you plan to create. Decide on borders, background color, padding, alignment, and so on.

   2. Modify your project's external CSS file (project.css) to configure the table (and table cells) as needed.

   3. Update your chosen web page and add the HTML code for a table.

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.