Lesson 5:
HTML Tables

Objectives

By the end of this lesson, you will be able to:

- 2.1.5: Design and format HTML tables to present information in an organized way.

Pre-Assessment Questions

1. Which table element encloses table cell contents?

   a. <tr>
   b. <th>
   c. <td>
   d. <table>

2. Which table element allows you to add a descriptive title that appears above the table by default?

   a. <border>
   b. <caption>
   c. <title>
   d. <table>

3. What CSS property manipulates the amount of space between cell borders in a table?

   _____

© 2014 Certification Partners, LLC — All Rights Reserved.

OBJECTIVE
2.1.5: HTML tables

Introduction to HTML Tables

NOTE:
Tables were first introduced in HTML 2.0 and have been expanded upon ever since.

When you want to display data that is best suited to a tabular format, you can use the set of table tags provided in HTML. Tables can provide useful and attractive grids for Web page content. As you learned earlier in this course, you can also use the <pre> tag to structure tabular data. However, the <pre> tag does not allow you to format data exactly as you want it. The <table> tag gives you many formatting options.

Figure 5-1 describes the individual elements that compose a table. Each of these elements will be explored in this lesson.

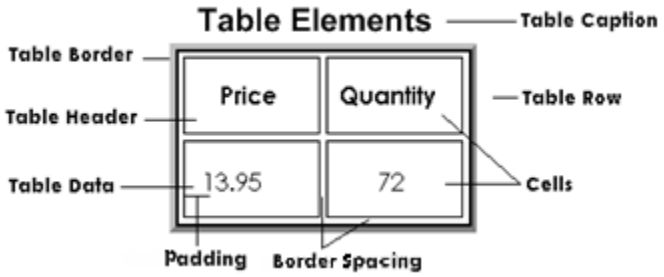

Figure 5-1: Table elements and CSS properties for HTML pages

The code for the table in the preceding figure could be written as follows:

NOTE:
The <pre> tag is extremely limited in function. HTML tables allow images, alignment, color and many customization attributes that far exceed the capabilities of the <pre> tag.

```
<table border="1">
<caption>Table Elements</caption>
<tr>
<th>Price</th>
<th>Quantity</th>
</tr>
<tr>
<td>13.95</td>
<td class="center">72</td>
</tr>
</table>
```

The table style could be defined in CSS as the following:

```
table, td, th
{
border:1px solid gray;
}
td, th
{
padding:15px;
}
td.center
{
text-align:center;
}
```

Table element attributes are not supported in HTML. CSS must be used. The only <table> element attribute still used is *border*, which simply states if the table has a border or not. The border attribute has a value of "0" for no border, and "1" for a border. The style of the border is defined in CSS.

Table 5-1 describes the table element tags that correspond to each element in the preceding figure.

© 2014 Certification Partners, LLC — All Rights Reserved.

Table 5-1: Table element tags

Element	Tag	Required?	Description
Table	<table>...</table>	Required	Creates a table. Contains all other table elements.
Table caption	<caption>...</caption>	Optional	Adds a caption or title, which appears above the table by default.
Table row	<tr>...</tr>	Required	Contains all data for a table row.
Table header	<th>...</th>	Optional	Typically designates cells in the top row or left column. By default, text in a header cell will appear bold and centered.
Table data	<td>...</td>	Required (unless <th> is being used)	Designates table cell contents. By default, the data is left-justified.

Tables can be very complex or very simple, as is the one you will create in the next lab. Straightforward tables are easy to create if you understand where to place each element and CSS property.

CIW Online Resources – Movie Clips

Visit CIW Online at http://education.Certification-Partners.com/CIW to watch a movie clip about this topic.

Lesson 5: HTML Tables

CIW Online Resources – Online Exercise

Visit CIW Online at *http://education.Certification-Partners.com/CIW* to complete an interactive exercise that will reinforce what you have learned about this topic.

Exercise 5-1: Defining table structure

CSS Properties for All Table Elements

The CSS properties for tables can be used for any element used in the table. For example, the <table> element defines the overall table structure, but the CSS properties used to provide its style are also used for the <th>, <tr> and <td> elements.

Recall that the *border* attribute determines whether a border will appear around the table. By default, borders do not appear. If the value is "1", a border appears. If the value is "0" or the attribute does not exist, a border does not appear.

NOTE: Deprecated style attributes will not pass validation for HTML5. CSS is required for table styles.

Table 5-2 describes CSS properties used to create tables.

Do not confuse the border property with the border attribute. In CSS, the border property determines the style of the border, but only after it is activated in the HTML document with the <table border="1"> tag.

© 2014 Certification Partners, LLC — All Rights Reserved.

Table 5-2: CSS properties used for <table> element

Property	Description	Values
border	Determines the style of the border.	- border width (e.g., *1px*) - border line style (e.g., *solid*) - border color (e.g., *gray*)
border-collapse	Collapses the borders of adjacent cells into a single border instead of separating them.	- *collapse*
border-spacing	Determines the amount of space between the borders of adjacent cells.	- Number of pixels
padding	Determines the amount of space between cell data and the cell border.	- Number of pixels
width	Determines how far the table or cell will extend horizontally across the page.	- Number of pixels - Percentage of the browser window — number followed by percent symbol (%)
background-color	Determines the background color for table elements.	- Color name (e.g., *green*) - Hexadecimal code (e.g., *#00FF00*)
text-align	Aligns text horizontally.	-*left* -*center* -*right* -*justify*
vertical-align	Aligns content vertically.	- *top* - *bottom* - *middle*

CIW Online Resources – Online Exercise

Visit CIW Online at *http://education.Certification-Partners.com/CIW* to complete an interactive exercise that will reinforce what you have learned about this topic.

Exercise 5-2: CSS properties for all table elements

In the following lab, you will create tables using HTML5 and CSS. Suppose you notice that the content on the syb_day.html page would work well in a tabular format, which improves readability and adds visual interest. You can organize the existing content into an HTML table by adding the appropriate element structure. Then you can supply a couple sentences to introduce the table, as well as a table caption with the title for the table: "A Day in the Life of a Habitat Worker."

© 2014 Certification Partners, LLC — All Rights Reserved.

Lab 5-1: Creating HTML tables to organize content

OBJECTIVE
2.1.5: HTML tables

In this lab, you will convert existing content into tabular format by creating an HTML5 table.

1. From **C:\CIW\Site_Dev\Lab Files\Lesson05\Lab_5-1**, copy the file **syb_day.html** and the **syb_day** folder to the **Habitat\CCYP** directory. The syb_day.html file should reside on the same level as syb.html, default.html and other Web page files.

2. Load the file **syb_day.html** into your browser. It should resemble Figure 5-2. Notice that the content is not presented in tabular format.

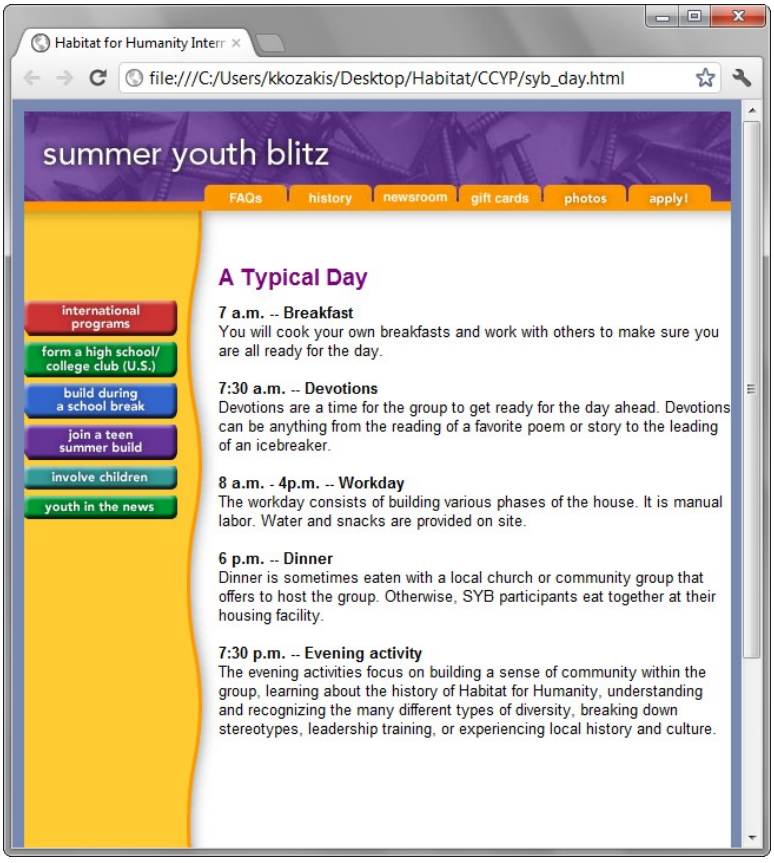

Figure 5-2: Page syb_day.html presented without table

3. Open **syb_day.html** in a text editor.

4. Search for the following line of code:

```
<h2>A Typical Day</h2>
```

NOTE:
In Step 5, ensure that you use the Edit | Cut command to cut the text. You should not press delete.

5. Use the **Edit | Cut** command (or **CTRL+X**) to cut all content that refers to the "***typical day***," including all text and markup tags. Start by cutting the ** ** tag located below the line specified in the previous step, continue through the text that ends with "***experiencing local history and culture***," and also include the **
** tags and the ** ** tag just above the closing </article> tag. (Do not cut the </article> tag.)

© 2014 Certification Partners, LLC — All Rights Reserved.

6. Open a second instance of Notepad, and paste the text you just cut into a new file.

7. Save this new file as ***old.txt***. In later steps in this lab, you will copy the content (but not the markup) from this text file into the new table you create in syb_day.html. For the purpose of learning tables, it is best to begin by creating the table elements first, then adding content to the table later.

8. Minimize **old.txt**.

9. Maximize **syb_day.html**. Add opening and closing tags to create a table where the text once existed:

```
<table>
</table>
```

10. In **syb_day.html**, create some white space by entering several returns between the opening and closing <table> tags you just entered.

11. By default, tables do not render with a visible border. Give the table a border by adding the ***border="1"*** attribute and value to the opening **<table>** tag. You now will be able to see your work better.

12. Create a table caption that reads ***A Day in the Life of a Habitat Worker***.

13. Consider the content on the original page. As you format the content into a table, you will need to label each column heading with the type of information that column will contain. What categories of information does the existing content provide? Answer the following question in the space provided: How many columns will your table need, and what will each column be titled?

    _____

14. You are now ready to create a header row for the table. Between the opening and closing <table> tags, insert a ***<tr>*** tag. Remember to close the table row with a ***</tr>*** tag, and use good coding practice by placing each new element on its own line.

15. Now, insert the necessary number of ***<th>*** tags between the <tr> </tr> tags to create the table header cells. When you are done, you should have three <th> tags, all properly closed with </th>.

16. The table header row will require the most tag information. When you are finished, your code should appear as follows:

```
<table border="1">
<caption>
A Day in the Life of a Habitat Worker
</caption>
<tr>
<th>Time</th>
<th>Activity</th>
<th>Description</th>
</tr>
</table>
```

17. Add the next row to your table. Use the ***<tr> </tr>*** tags again, but instead of using three sets of <th> tags within the row, use three sets of ***<td>*** tags to create the cells because this row is not a table header.

© 2014 Certification Partners, LLC — All Rights Reserved.

18. Within the <td> tags, enter the following content from old.txt:

7 a.m.
Breakfast
You will cook your own breakfasts and work with others to make sure you are all ready for the day.

19. The next task involves centering text in table data. Because not all table data will be centered, a class must be created in CSS. The class will then be specified in the HTML document. To create the class, open **syb_day/syb_day.css**.

20. In the **syb_day.css** file, add the following rule:

```
td.center
{
text-align:center;
}
```

21. Add the ***center*** class to center the content that appears in the first two cells you specify with <td> tags. When you are finished, your code should appear as follows:

```
<tr>
<td class="center">7 a.m.</td>
<td class="center">Breakfast</td>
<td>You will cook your own breakfasts and work with others to make sure you
are all ready for the day.</td>
</tr>
```

22. Insert the following text, with   tags, just above the table and just below the phrase *A Typical Day*:

```
<p>Why is our program so popular? Because it gives you the opportunity to help
others. Below are some examples of activities.</p>
```

23. Save the file, then view your page in a browser. You should see a table caption introducing a table with three columns, a header row and one row of information, as shown in Figure 5-3.

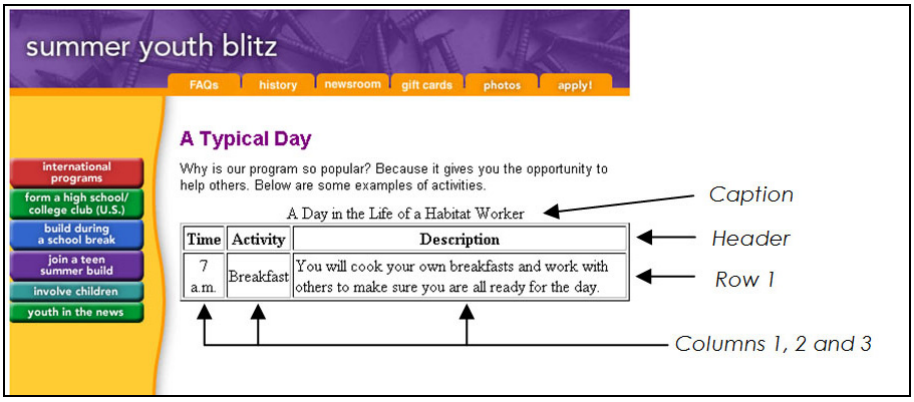

Figure 5-3: Page syb_day.html with content in tabular format

24. If your table does not render correctly, review your code and make any necessary changes. When the page renders correctly in the browser, validate your code at ***http://validator.w3.org***. The file is short at this point so errors will be easier to locate and correct.

25. Add four more rows to your table, so that you have five rows containing the content you cut from the page. (This content is now in the old.txt file you created, so copy it

© 2014 Certification Partners, LLC — All Rights Reserved.

from there.) Center the content that appears in the first two cells you specify with <td> tags in each row.

26. Notice that the <tr> tags add rows, but that the <td> tags add cells to contain content. The content in each cell is categorized by the column headers, which are defined by the <th> tags.

27. To ensure the table font matches the document font, add the following code to the **syb_day.css** file. Tables often look better when the table text is slightly smaller than the surrounding text, so the font size will be adjusted from 80% to 70%. The width of the table will be set to 90% of the section width so that a margin appears:

```
table
{
font-family: Arial, Helvetica, Geneva, Swiss, sans-serif;
font-size: 70%;
width:90%;
}
```

NOTE:
Table data alignment can drastically affect the look of a table. For instance, if each row uses a different horizontal alignment, the table may not appeal to viewers. For the best visual result, align content consistently within columns. Table data containing numbers or a product's price, for example, is often right-aligned in columns to give an appearance similar to spreadsheets. All spreadsheets use a tabular format for information.

28. To center the table in the Web page section, a *class* must be created in **syb_day.css**, similar to centering an image. The *margin-left* and *margin-right* properties are set to *auto*. This automatically creates equal spacing on both sides of the object, whether it is a table or an image.

```
table.center
{
margin-left: auto;
margin-right: auto;
}
```

29. Add the center *class* to the **syb_day.html** file by adding the following code in bold:

```
<table class="center" border="1">
```

30. To improve readability, add five pixels of padding to all table data. This will create white space around the data. Enter the following code in **syb_day.css**:

```
td
{
padding:5px;
}
```

31. The last adjustment will be the width of the "Time" column. The table would look better if the text "a.m." and "p.m." were on the same line as the time (e.g., 7 a.m.). The width can be adjusted by adding a *class* that specifies the header's width in pixels. Add the following code to **syb_day.css**:

```
th.width
{
width:50px;
}
```

32. Add the *class* to the **syb_day.html** file:

```
<th class="width">Time</th>
```

33. When you are finished, save **syb_day.html** and **syb_day.css** and load them into a browser. If your table does not render as expected, review your code. Edit the code as necessary until it renders as shown in Figure 5-4.

© 2014 Certification Partners, LLC — All Rights Reserved.

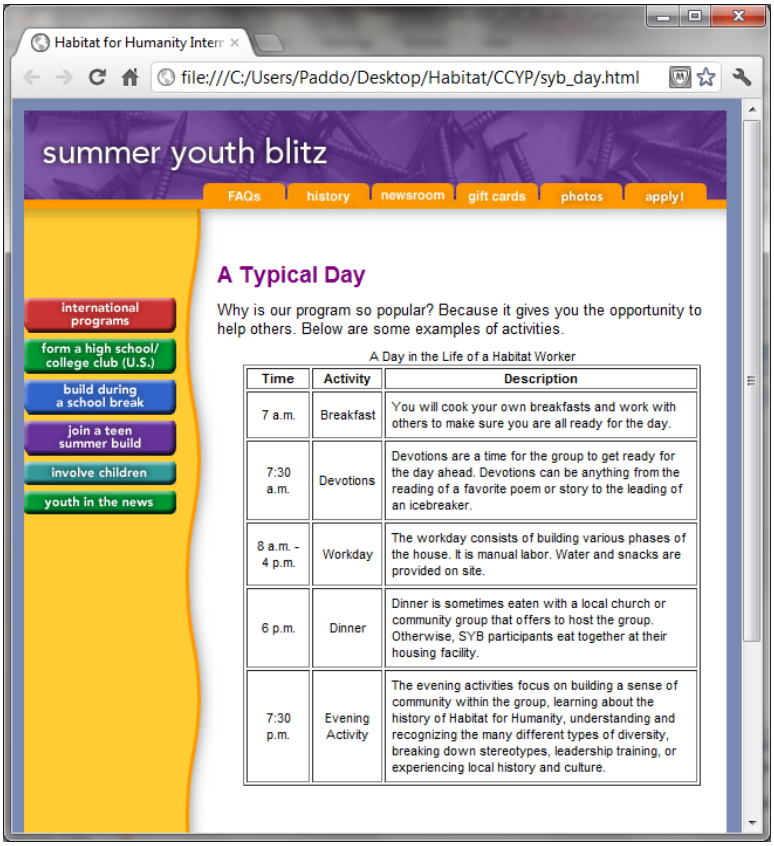

Figure 5-4: Page syb_day.html with table

34. Validate your code and resolve any issues. You will edit this table further in a later lab.

In this lab, you converted existing content into tabular format by creating an HTML table.

OBJECTIVE
2.1.5: HTML tables

Table and Data Alignment Options

Frequently you will want to align cell content in a manner that differs from the default alignment. Following are the defaults for table data:

- Content in table header cells is aligned both horizontally and vertically to the center of the cell.
- Content in table data cells is aligned horizontally to the left and vertically to the center.

Table data alignment can drastically affect the look of a table. For instance, if each row uses a different horizontal alignment, the table may not appeal to viewers. For the best visual result, align content consistently within columns.

So far, you have examined only tables with single lines of cell content. Now consider the example in Figure 5-5.

© 2014 Certification Partners, LLC — All Rights Reserved.

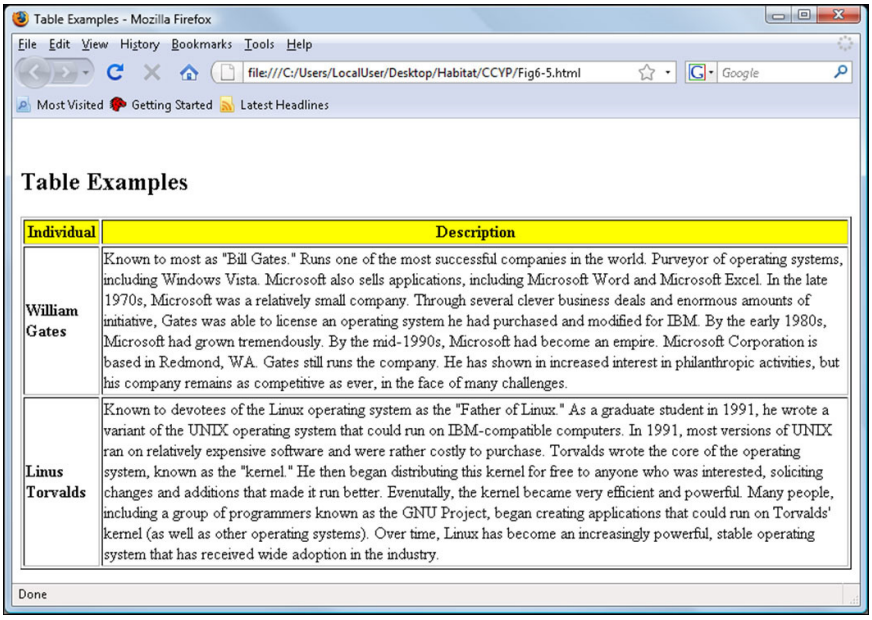

Figure 5-5: Alignment of cell content

You can change the alignment for an individual cell or for an entire row. The CSS property you use to specify horizontal alignment is *text-align*. For vertical alignment, use the *vertical-align* property.

Either of the following code samples in Figure 5-6 could be used to produce the alignment shown in Figure 5-7.

HTML document code:

```
<tr>
<td>
William Gates
</td>
```

Corresponding CSS document code:

```
tr
{
vertical-align:top;
}
```

HTML document code:

```
<tr>
<td class="valign">
William Gates
</td>
```

Corresponding CSS document code:

```
td.valign
{
vertical-align:top;
}
```

Figure 5-6: Two ways to change vertical alignment

The code example on the left in the preceding figure sets the alignment of the entire row to "top", meaning that all cell content in that row will start at the top of the cell. The code example on the right creates a *class* named *valign* that specifies a vertical alignment only for the cell containing "William Gates". This *class* can be used in any <td> tag.

The cell to the right of "William Gates" is already aligned to the top because it is completely full, so the appearance generated by either choice will match that shown in Figure 5-7. (The second row was left unchanged for contrast.)

© 2014 Certification Partners, LLC — All Rights Reserved.

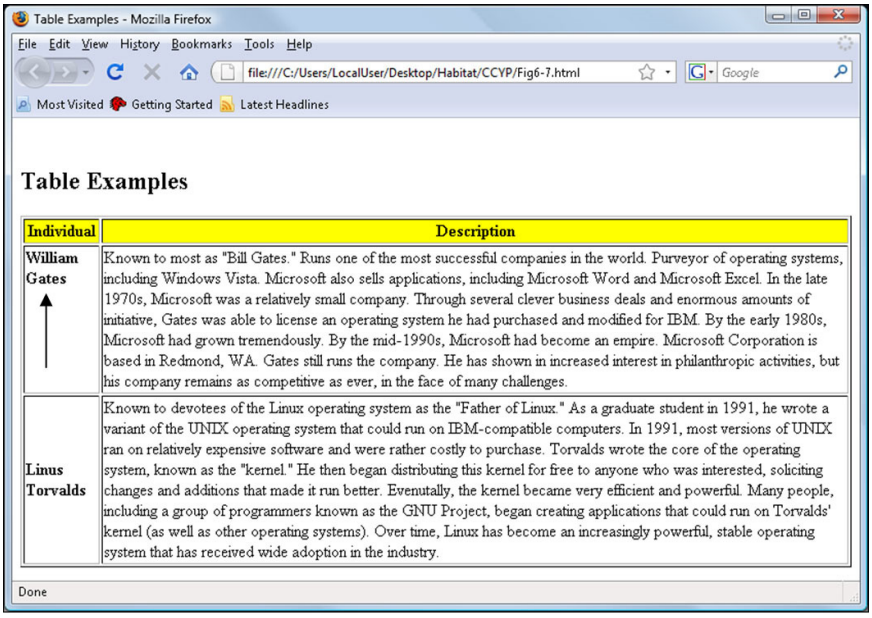

Figure 5-7: After specifying vertical alignment affecting cell or row containing "William Gates"

You can also align the contents of any table cell horizontally to the left, right or center.

OBJECTIVE
2.1.5: HTML tables

Height and Width of Table Elements

NOTE:
Adding height and width to table header and data cells increased this table's readability. However, adding unnecessary height or width to your table can waste space on your Web page.

You can change the height or width of the table and individual cells by specifying pixel or percentage values. Examine Figure 5-8.

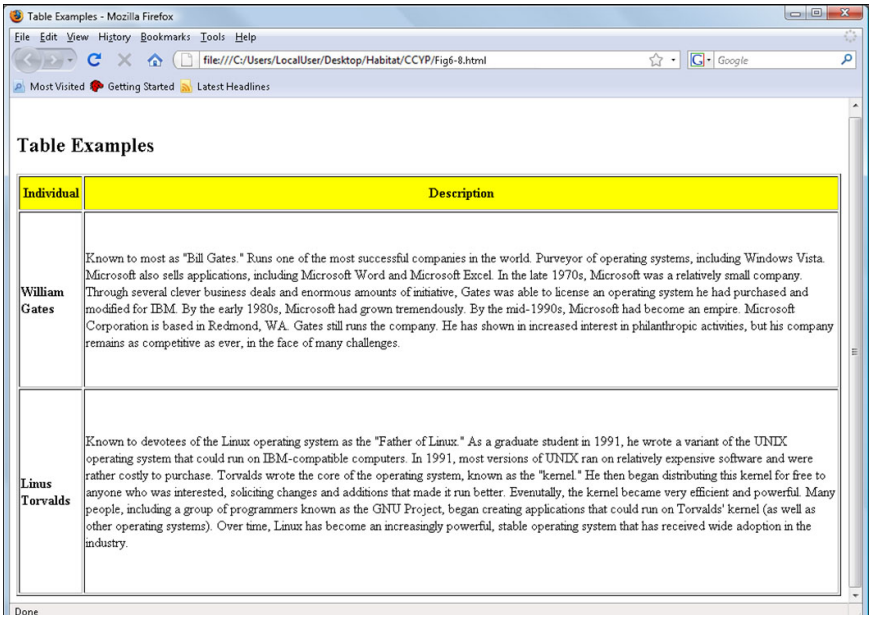

Figure 5-8: After increasing table height

In the preceding figure, the table rows increased in size because the *height* property of the CSS *table* rule was set to 500 pixels. You could also specify a percentage of the browser window. Note that the additional height was equally distributed among the various rows.

© 2014 Certification Partners, LLC — All Rights Reserved.

In Figure 5-9, the *height* property has been deleted from the CSS *table* rule, and is set only for the cell containing the word *Description*. The cell height has been set to 75 pixels. If you mistakenly set multiple values in multiple cells, the highest value will take precedence; the order of entries does not matter.

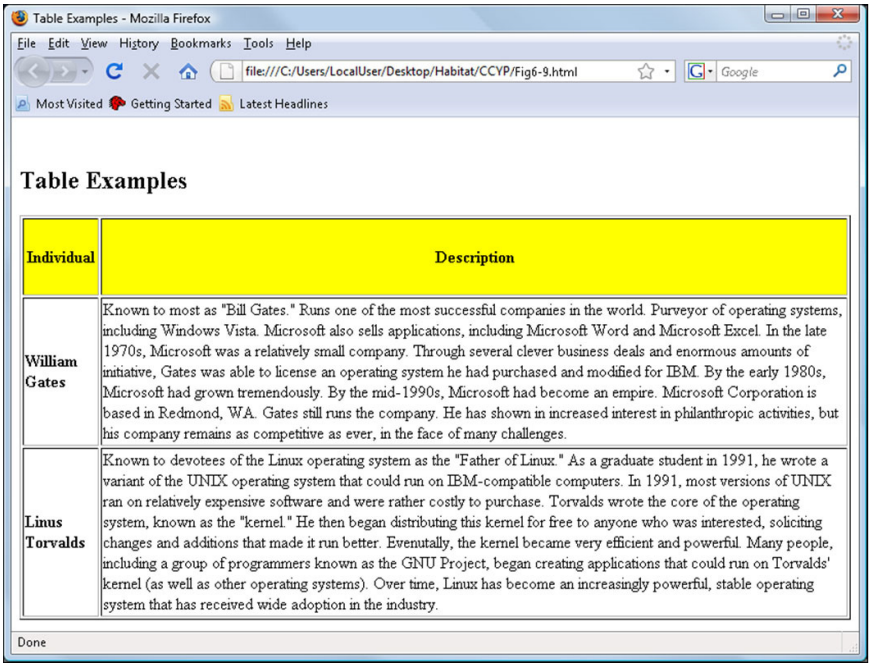

Figure 5-9: After specifying and adding CSS class td.height {height: 75px;} *in table cell*

In Figure 5-10, all *height* properties have been deleted, except that the width of the first column (with the title *Individual*) has been set to 50 percent of the browser window.

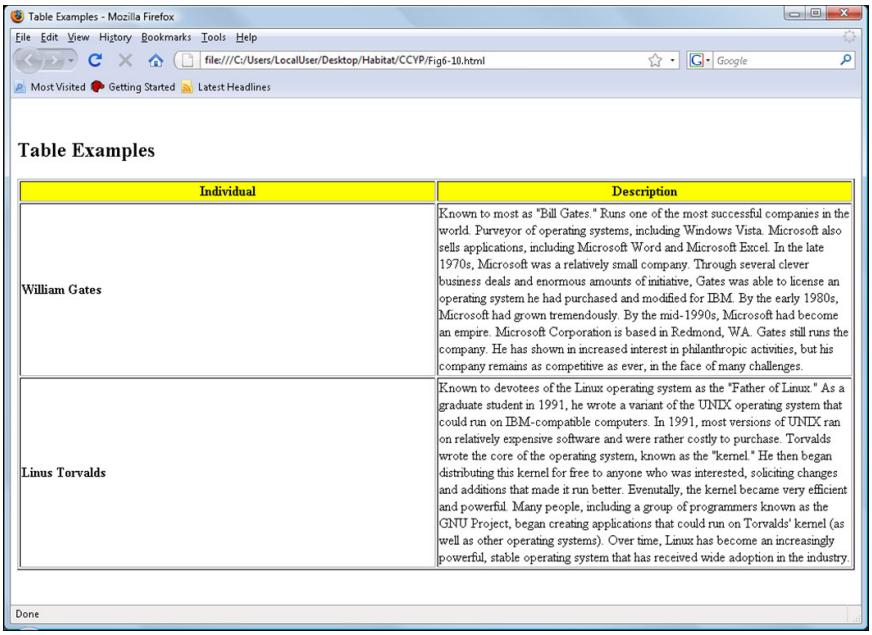

Figure 5-10: After specifying and adding CSS class th.width {width:50%;} *to first <th> tag*

© 2014 Certification Partners, LLC — All Rights Reserved.

In Figure 5-11, the table width has been changed to 70 percent of the screen and the table has been centered.

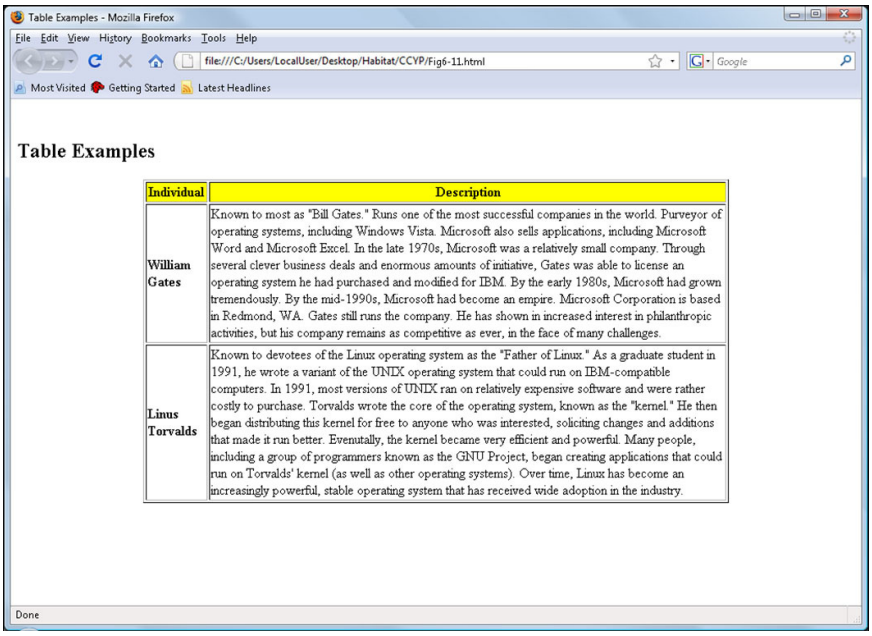

Figure 5-11: After centering table

OBJECTIVE
2.1.5: HTML tables

Column and Row Spanning

Sometimes you will want a column to span across two or more cells. In other cases, you may need a cell to span more than one row. Figure 5-12 demonstrates a cell that horizontally spans two columns.

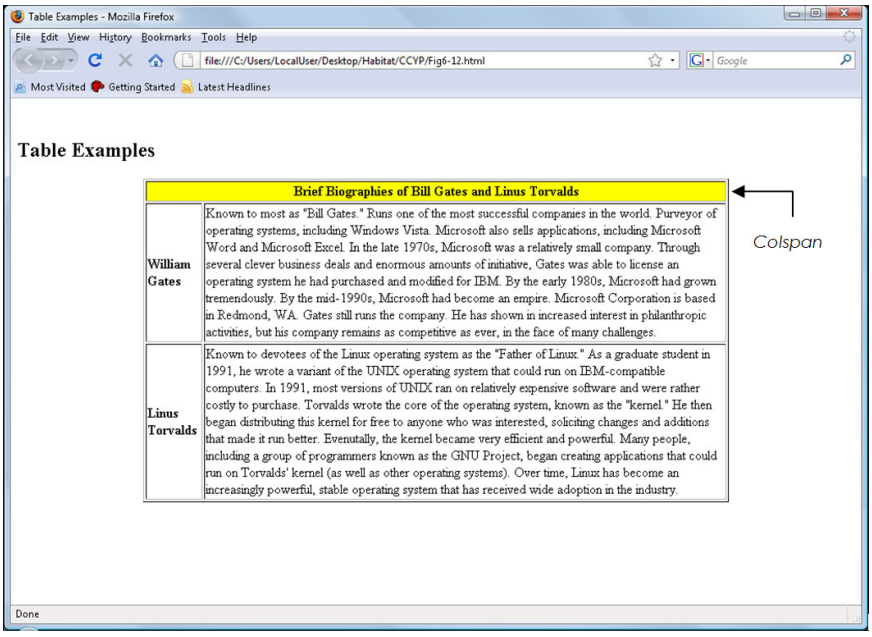

Figure 5-12: Using colspan attribute

NOTE:
To review the elements and attributes introduced in this lesson, see **Activity 5-1: Identifying HTML table elements and attributes**.

The following code created the spanned column in the preceding figure:

```
<tr>
<th colspan="2">
Brief Biographies of Bill Gates and Linus Torvalds
</th>
</tr>
```

Notice that one of the <th> headings has also been removed. This change is logical because now one <th> tag is occupying both spaces. Figure 5-13 shows a cell (i.e., column) that vertically spans two rows.

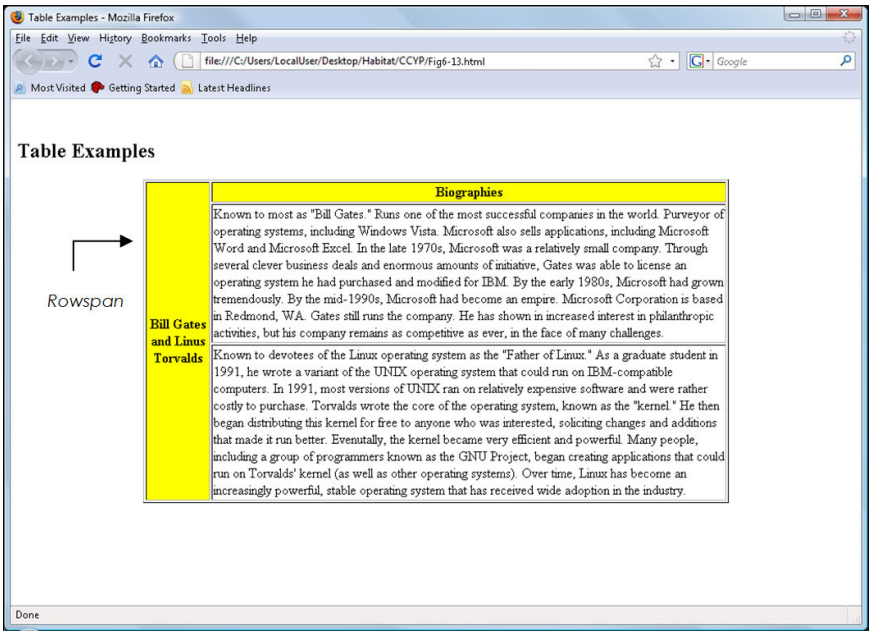

Figure 5-13: Using rowspan attribute

The following code created the spanned row in the preceding figure:

```
<tr>
<th rowspan="3">
Bill Gates and Linus Torvalds
</th>
<th>
Biographies
</th>
</tr>
```

Note that the *colspan* and *rowspan* attributes can also be used with the <td> tag.

In the following lab, you will customize an HTML table. Suppose you want to add some formatting to your Web page table and the content organized within it. You can use CSS font style, color, alignment and spacing options to modify the table's appearance. Your formatting choices will improve the table's readability, as well as make it more visually exciting.

© 2014 Certification Partners, LLC — All Rights Reserved.

Lab 5-2: Customizing HTML tables with CSS formatting

OBJECTIVE
2.1.5: HTML tables

In this lab, you will modify table data with formatting options to make it more aesthetically pleasing.

1. Open **syb_day.html** and **syb_day/syb_day.css** in an editor. If necessary, you can also copy them from your student Lab 5-2 folder and paste them in your Habitat/CCYP folder, similar to step 1 of the previous lab.

2. Find the table caption, which reads *A Day in the Life of a Habitat Worker*. Notice that it is in a standard font. Add the ****** tag to make this caption bold.

3. Use CSS to change the first **<tr>** element so the background color of the table headers is light blue (***"#6699FF"***). Load the page in a browser to see how it renders, then validate your code at ***http://validator.w3.org***.

4. Find the first **<th>** heading tag. Change the ***width*** property used in its CSS *class* to a value of ***"93"***. Load the page in a browser to see how it renders, then validate your code.

5. Change the ***padding*** attribute in the CSS table data (*td*) rule to a value of ***"3"***.

6. Experiment with other modifications and formatting additions to the table. For instance, add background colors to table cells and rows, change the font colors, or collapse the border so only one line appears around each cell.

7. In the HTML document, insert a blank line between the end of the table and the footer section by adding a ***
*** tag immediately after the closing table tag </table>. Load the page in a browser to see how it renders, then validate your code.

8. You have now placed all of the page's content into a table. However, the changes you made may have caused the yellow navigation (*nav*) sidebar graphic (navbg_04.gif) to be too short or too long for the page. Open the style sheet and find the ***nav*** and ***article*** entries.

9. Adjust the ***height:*** value of the *nav* and *article* entries so that the yellow sidebar graphic is the same length as the table. The height values should be the same to ensure the footer is evenly spaced beneath them. Experiment with the setting until the *nav* and *article* sections look even when rendered in the browser.

10. When you are finished, your page may resemble Figure 5-14, depending on how you experimented and customized your table.

© 2014 Certification Partners, LLC — All Rights Reserved.

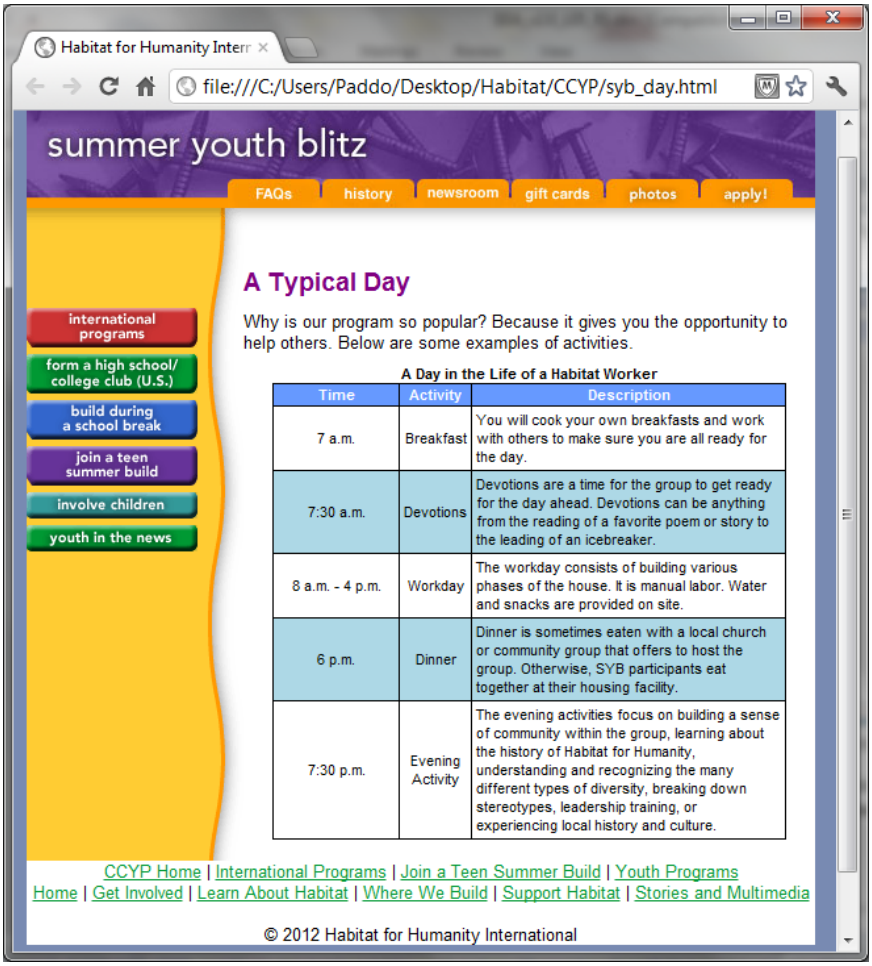

Figure 5-14: Page syb_day.html with customized formatting

11. **If time permits:** Remove the border around the table and refresh your browser. Use the ***border="1"*** attribute and value in the **<table>** tag to reapply the border.

In this lab, you customized the appearance of a table and its content.

HTML Tables vs. CSS Page Structure

NOTE:
As your tables become more complex, you will quickly understand why GUI HTML editors are so popular. However, it is still essential that you create HTML tables manually so you know how to manipulate and customize tables created with GUI HTML editors.

HTML tables have often been used to create content structure for entire Web pages. In HTML5, however, you should not use tables to structure a page. Instead, the W3C recommends that you use the new HTML5 document structure tags.

As you learned earlier in this course, HTML5 with CSS has introduced a more effective and simple way to structure Web pages. Instead of using the tag as a "jack of all trades" element with its array of attributes, the developers of HTML5 created specific elements to define the document structure. These elements include <header>, <footer>, <nav>, <article> and <aside>. Review the simple Web page structure shown in Figure 5-15.

© 2014 Certification Partners, LLC — All Rights Reserved.

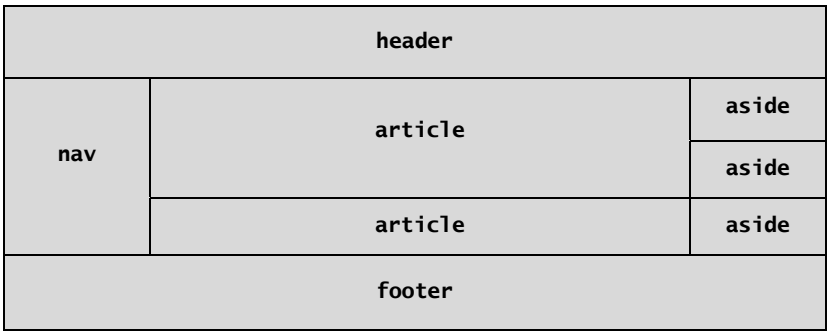

Figure 5-15: Sample HTML5 structural elements of Web page

As shown in the figure, a developer can structure the Web page with elements that are easily interpreted and native to any HTML5-compliant browser, regardless of whether the browser is on a mobile device, laptop or tablet. The structure elements can be defined as:

- **<header>** — Top of the Web page, similar to the header in a word-processing document.
- **<nav>** — Defines navigation links, such as hypertext menus to access various pages of the Web site.
- **<article>** — Web site content, such as company services, articles, blogs, images and videos.
- **<aside>** — Content that is "aside" from the article content, such as advertisements or news feeds.
- **<footer>** — Bottom of the Web page, similar to the footer in a word-processing document.

The header might contain the company logo; the footer could include copyright and contact info; the nav section may contain navigational links to other resources on the site; the articles could be blogs, tweets, video, social networking posts, news articles, or any other content; and the aside could be advertisements, highlighted resource links, and even more content.

A search engine will automatically rank a page lower if the page uses tables for structure. Use the HTML5 structure elements and CSS instead. Also, if you overuse a table on a page, you may also experience problems. Use tables tastefully and sparingly.

CIW Online Resources – Course Mastery

Visit CIW Online at *http://education.Certification-Partners.com/CIW* to take the Course Mastery review of this lesson or lesson segment.

SDA Lesson 5

© 2014 Certification Partners, LLC — All Rights Reserved.

Case Study

To Use Tables or Not To Use Tables

An-Mei supervised a Web development team responsible for converting the company Web site's HTML 4.0 code into HTML5. She began the project by reviewing the existing Web pages and code with her team.

An-Mei and her team found that most of the existing pages had used HTML tables to structure all content, rather than only the content that lent itself to a tabular format.

An-Mei chose to limit the use of tables whenever possible, reserving them for content that required a structured layout, such as numbers, dates and other items often viewed within spreadsheets or complex lists.

An-Mei directed the team to remove all HTML tables used for page structuring, and to replace the structure using the HTML5 structure elements instead. Although this change required a significant investment of time, the team found that the time was well spent. Without the table structuring, all pages could be quickly updated, and future modifications to the site content were easier. Upper management was pleased that Web site updates were implemented more readily than in the past.

* * *

As a class, discuss the ways that HTML tables can improve a Web page.

- What types of content benefit from a tabular structure? What types of content do not?
- Are there any advantages in using an invisible HTML table to structure an entire Web page? Why did Web developers use them for so many years? What disadvantages does the table structure pose?
- Is such a situation likely to occur in the workplace when you are dealing with a business or organization Web site? Why or why not?

© 2014 Certification Partners, LLC — All Rights Reserved.

Lesson Summary

Application project

Tables are useful for presenting information that naturally lends itself to tabular format. HTML allows you to do this. In the past, tables have been used to add structure to a Web page. However, the W3C recommends against this practice for HTML5, so you should use the HTML5 structure elements instead.

Visit *www.w3schools.com* and *www.w3.org*, and review the <table> element. Then browse some sites, visiting some that include tables. See how other Web developers use tables to organize content. Do you think tables provide an effective presentation? What types of content are best suited for tabular presentation?

Skills review

In this lesson, you learned to create and manipulate HTML tables. You used the basic structure tags to generate a table, rows and individual cells. You also aligned cell content, learned how to span cell content across rows and columns, and added background colors to both tables and cells. Finally, you learned that manipulating a border can dramatically affect the appearance of your table.

Now that you have completed this lesson, you should be able to:

- ✓ 2.1.5: Design and format HTML tables to present information in an organized way.

CIW Practice Exams

Visit CIW Online at *http://education.Certification-Partners.com/CIW* to take the Practice Exams assessment covering the objectives in this lesson.

SDA Objective 2.01 Review

Note that some objectives may be only partially covered in this lesson.

© 2014 Certification Partners, LLC — All Rights Reserved.

Lesson 5 Review

1. The size of an HTML table can be modified by specifying measurements in either of which two units of measure?

2. What element can be used instead of the <table> element to create a simple table by preserving spacing and line breaks in preformatted text?

3. What components of table appearance can be formatted using the CSS *text-align* property or by creating a CSS *class* that specifies either a *right*, *left* or *center* justification value?

4. What is the default alignment of the content in table data cells?

5. What attributes are used to span a single cell across multiple rows or columns, respectively?

© 2014 Certification Partners, LLC — All Rights Reserved.

© 2014 Certification Partners, LLC — All Rights Reserved.