Lesson 5: Printer Friendly

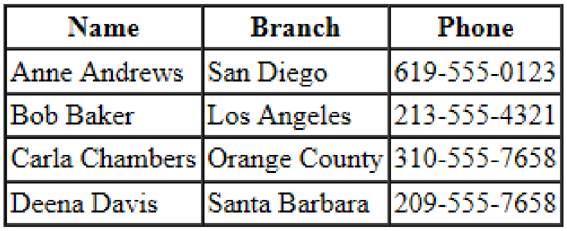
Creating Tables

Lessons > Lesson 5 > Printer Friendly

Chapter 1

Introduction

Welcome back! So far in this course, you've learned some core HTML for creating the kinds of elements you often see in websites and other electronic documents. That includes headings, paragraphs, different kinds of lists, links, and images (pictures), to name a few. Today you'll learn to create tables. Tables are also quite common design elements, and we use them to organize information into clear, easy-to-read, rows and columns. Here's an example:



Name	Branch	Phone
Anne Andrews	San Diego	619-555-0123
Bob Baker	Los Angeles	213-555-4321
Carla Chambers	Orange County	310-555-7658
Deena Davis	Santa Barbara	209-555-7658

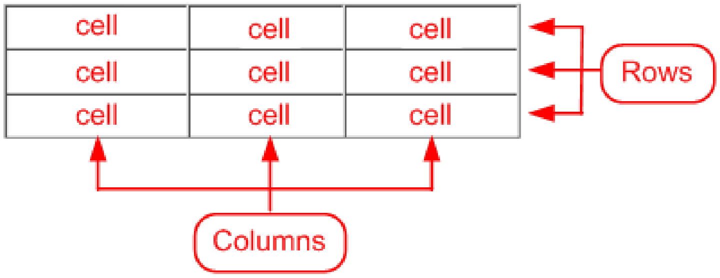
A simple table

Like other element types, you'll use tags to create tables. Ready to get started? Come on over to Chapter 2, and we'll begin by looking at the basics!

Chapter 2

Table Basics

Tables are a common design element for organizing information into rows and columns. Rows stretch horizontally across the table, and columns are vertical. At the junction of each row and column is a cell.



Components of a table

Tip

To remember the definitions of *row* and *column*, think of a boat you *row* across a lake, and *columns* standing vertically in front of a building.

Here are the tags we use to create tables:

<table>...</table>: Mark the beginning and end of a table.

<tr>...</tr>: Mark the beginning and end of a table row.

<th>...</th>: Marks the beginning and end of a table header cell.

<td>...</td>: Marks the beginning and end of a table data cell.

A *header cell* is typically the topmost cell in a column, or the leftmost cell in a row. Let's get started with a little practice working with tables. We'll create a table and put it in the recipe.htm page.

1. Open your MyWebsite folder.
2. Right-click (or CTRL + Click) **recipe.htm**, and open it with your editor (Notepad or TextEdit).
3. Put the cursor at the end of the line that reads <p>Eat and enjoy!</p>, and press ENTER to start a new blank line.
4. To define a table, start by typing the following:

   ```
   <table>
   ```

5. Press ENTER a couple of times, and then type this:

   ```
   </table>
   ```

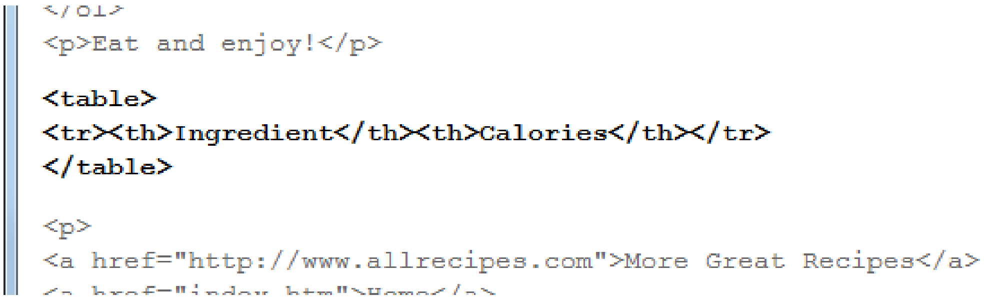
6. As always, you don't have to type the closing tag right after the opening tag. But it's a good habit to get into so you won't forget to type the closing tag later.
7. Put the cursor between the tags and that you just typed.
8. Now type the tags to define a table row:

   ```
   <tr></tr>
   ```

9. Next, we'll add a couple of header cells inside that row. Put the cursor between the <tr> and </tr> tags you just typed. Then, type the following code and content:

   ```
   <th>Ingredient</th><th>Calories</th>
   ```

At this point you have the beginnings of a table. In that section of the recipe.htm page, you should see the new table tags and content. In the image below, I'm showing some of the code that was already in recipe.htm just so you can see where you should place the new tags.



```
</ol>
<p>Eat and enjoy!</p>

<table>
<tr><th>Ingredient</th><th>Calories</th></tr>
</table>

<p>
<a href="http://www.allrecipes.com">More Great Recipes</a>
<a href="index.htm">Home</a>
```

Start of a table in recipe.htm

If you were to take a look at the page in a Web browser right now, you wouldn't see anything that looks like a table. But don't worry about that. We have more to do. Let's add some more rows (with tr tags) and some data cells (with td tags).

1. Put the cursor past the </tr> tag, and press ENTER to start a new row.
2. Type the following tags and content so that the new row is under the first row:

```
<tr><td>Hot Dog</td><td>150</td></tr>
```

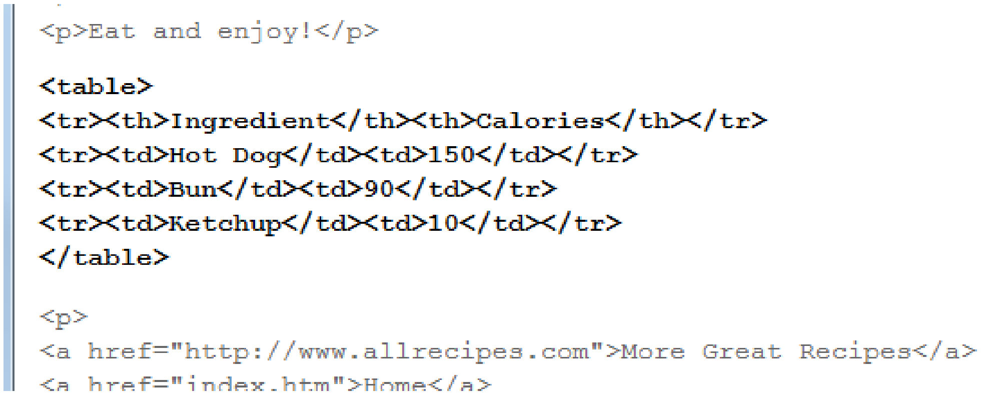
3. Press ENTER, and type the next row as follows:

```
<tr><td>Bun</td><td>90</td></tr>
```

4. Press ENTER, and type one more row of tags and text as follows:

```
<tr><td>Ketchup</td><td>10</td></tr>
```

The image below shows the entire table typed into the page. Take a look, and verify that what you typed matches what you see here:



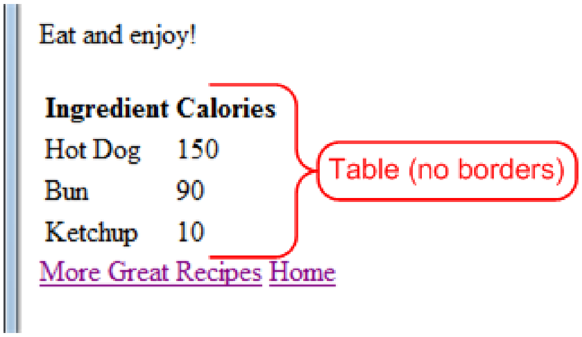
```
<p>Eat and enjoy!</p>

<table>
<tr><th>Ingredient</th><th>Calories</th></tr>
<tr><td>Hot Dog</td><td>150</td></tr>
<tr><td>Bun</td><td>90</td></tr>
<tr><td>Ketchup</td><td>10</td></tr>
</table>

<p>
<a href="http://www.allrecipes.com">More Great Recipes</a>
<a href="index.htm">Home</a>
```

Table rows and data cells added to the table

Would you like to view your table in a Web browser now? Here's what you should do:

1. Choose **File > Save** from your editor's menu.
2. Double-click **recipe.htm** in your MyWebsite folder to open the page in a browser.

It won't look quite like a table, because we're not done yet. But you'll see the words and numbers organized into rows and columns between the list and links you put into that page in earlier lessons. But it looks weird, because there are no borders (lines) between the table cells. We'll remedy that in a moment.



Eat and enjoy!

Ingredient	Calories
Hot Dog	150
Bun	90
Ketchup	10

More Great Recipes Home

Start of a table in recipe.htm

Adding Borders

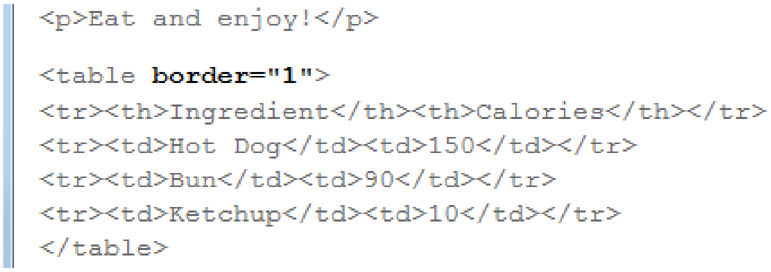
To add borders, you can add a border= attribute to the <table> tag. This is the syntax:

```
border="x"
```

Replace *x* with the width of the border in pixels. The smallest possible size is 1 for a 1-pixel wide border. So to put a thin border around the cells, use border=1 (see below). As always, there's a space before the word *border*, and there are quotation marks around the number *1*.

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

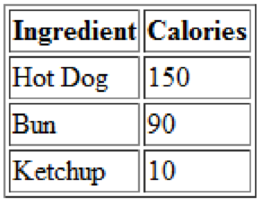
You can go ahead and make that change in Notepad or TextEdit. The image below shows this new attribute typed into the table tag.



```
<p>Eat and enjoy!</p>

<table border="1">
<tr><th>Ingredient</th><th>Calories</th></tr>
<tr><td>Hot Dog</td><td>150</td></tr>
<tr><td>Bun</td><td>90</td></tr>
<tr><td>Ketchup</td><td>10</td></tr>
</table>
```

Border attribute added to the table tag

Save the change in the editor, and open the page in a browser (or refresh it in the browser if it's already open there). Now you'll see a border around the table cells. In some browsers, you'll see double lines as below. Some may show beveled gray lines, some may show single lines. It all depends on what browser you use to view the page But don't worry about that. For now it's sufficient to know you can use the border= attribute to put border lines around table cells.



Ingredient	Calories
Hot Dog	150
Bun	90
Ketchup	10

Borders in the table in the browser

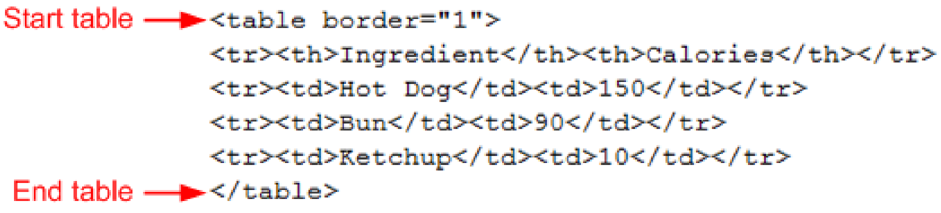
If your table looks a lot different in your browser, it's probably a typographical error in your code. Here's the entire table, once again, so you can see all of the tags in place. You can copy and paste the code into your own page, if you'd like, rather than hunt for every little typographical error.

```
<table border="1">

<tr><th>Ingredient</th><th>Calories</th></tr>

<tr><td>Hot Dog</td><td>150</td></tr>
```

```
<tr><td>Bun</td><td>90</td></tr>

<tr><td>Ketchup</td><td>10</td></tr>

</table>
```

Let's take a moment now to review what the tags are all about. There are a lot of tags. But every table you ever create will consist of the same tags, organized in the same way. And if you pick it apart piece by piece, it's not quite as intimidating.

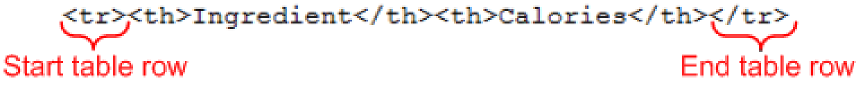
The <table> tag marks the start of the table. The </table> tag marks the end of the table, as illustrated here:



```
<table border="1">
<tr><th>Ingredient</th><th>Calories</th></tr>
<tr><td>Hot Dog</td><td>150</td></tr>
<tr><td>Bun</td><td>90</td></tr>
<tr><td>Ketchup</td><td>10</td></tr>
</table>
```

Tags marking start and end of a table

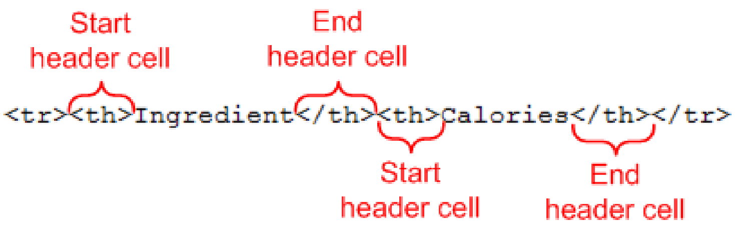
Inside the table, there are four table rows. Each row starts with a <tr> tag and ends with a </tr> tag.



```
<tr><th>Ingredient</th><th>Calories</th></tr>
```

The <tr>...</tr> tags mark one table row

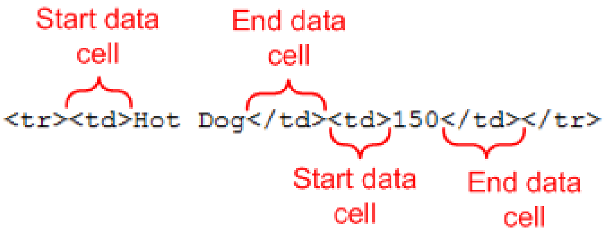
Inside each row are two table cells. In the top row, we use <th>...</th> tags to mark each cell. That marks the cells as a table header. The text inside each cell is boldface. But as you'll learn later, you can style any element to look any way you want. For now, we want these to be header cells, because they define what's in each column.



```
<tr><th>Ingredient</th><th>Calories</th></tr>
```

Two header cells in a table row

The second, third, and fourth rows each contain two data cells. Each data cell starts with a <td> tag and ends with a </td> tag.



```
<tr><td>Hot Dog</td><td>150</td></tr>
```

Two data cells in a table row

It's important to always remember that the HTML language exists so that you can use tags to define what every element in an electronic document *is*. In this chapter, we used <table>...</table> tags to define a table. Within that table, we used <tr>...</tr> tags to define each row. We used <th>...</th> to define each header cell in the top row. And we used <td>...</td> tags to define each table data cell in subsequent rows.

There's no rule that says you must arrange the tags themselves as I did above. So long as you don't arbitrarily breaks lines inside the < and > angle brackets of a single tag, you can break lines and use indents to your heart's content. Those line breaks and white space won't carry over to the Web browser at all. For example, you could type the table we just typed like this:

```
<table border="1">

        <tr>

            <th>Ingredient</th>

            <th>Calories</th>

        </tr>

        <tr>

            <td>Hot Dog</td>

            <td>150</td>

        </tr>

        <tr>

           <td>Bun</td>

           <td>90</td>

        </tr>

        <tr>

           <td>Ketchup</td>

           <td>10</td>

        </tr>

</table>
```

The table would look exactly the same in the Web browser, despite the radically different formatting of tags and content within the source code. This is because the browser does all of its formatting based on what the tags say, not what the line breaks and indents in the source code say.

Not too difficult, right? Great work!

In the next chapter, you'll learn about some more attributes that you can use to fine-tune the appearance of your tables.

Chapter 3

Basic Table Styles

As you saw in the previous chapter, when you use only tags to create a table, the content is displayed in neat rows and columns, but without border lines between the cells. We talked about the *border attribute*, which lets you define the width of the line used to separate border cells. The *default* (meaning, what you get if you don't use a border attribute at all) is zero, so that's why you don't see any border lines between cells if you don't put in a border. If you add border="1" to the <table> tag, then you get a nice thin 1-pixel border around the cells. But as you saw, the lines are doubled. That's because there's a little *spacing* between cells. You can use another attribute, cellspacing, to control that spacing between the cells. Let's take a look!

The Cellspacing Attribute

The cellspacing attribute allows you to control the empty space between cells. The syntax is much like other attributes. It's all one word (no spaces), all lowercase, followed by an equal sign, and then by a value enclosed in quotation marks:

```
cellspacing="x"
```

In practice, you replace the *x* with a number that represents how much space, in pixels, you want between cells. The default is 2, which means that if you don't use cellspacing at all, or if you put in cellspacing="2", as below, you get two pixels of space between cells.

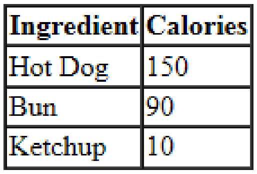
If you want to get rid of that space between cells, you can remove the cellspacing by setting the cellspacing attribute to zero as (see below).

```
<table border="1" cellspacing="0">

<tr><th>Ingredient</th><th>Calories</th></tr>

<tr><td>Hot Dog</td><td>150</td></tr>

<tr><td>Bun</td><td>90</td></tr>

<tr><td>Ketchup</td><td>10</td></tr>

</table>
```

Note

When you write code, if you mean to write *zero*, make sure you always type the number *0* on the keyboard, not the letter *O*. Unlike with typewriters, when you write code, you can't substitute the letter *O* for the number *0*, and you can't substitute a lowercase letter *l* for the number *1*.

Want to try it out for yourself? Go ahead and add cellspacing="0" to the <table> tag in the recipe.htm page. Save the page, and refresh the browser. If you did everything right, the gap between the cells disappears, and the table looks like this:



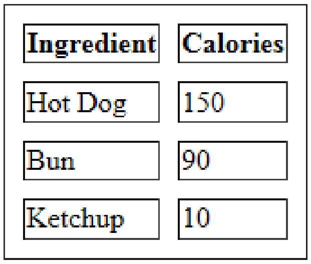
Ingredient	Calories
Hot Dog	150
Bun	90
Ketchup	10

Table with cellspacing set to zero

Say you want to increase the amount of space between cells. You just have to set the cellspacing to some number greater than 2. Now I'm not recommending you do this in real life, unless you happen to like how the table looks. But just to help you get a better sense of what the cellspacing attribute does, this is how you would increase the cellspacing to 10, for instance:

```
<table border="1" cellspacing="10">

<tr><th>Ingredient</th><th>Calories</th></tr>

<tr><td>Hot Dog</td><td>150</td></tr>

<tr><td>Bun</td><td>90</td></tr>

<tr><td>Ketchup</td><td>10</td></tr>

</table>
```

And here's how the table would look in a Web browser:



Ingredient	Calories
Hot Dog	150
Bun	90
Ketchup	10

Result on increasing cellspacing to 10

Remember, the cellspacing attribute affects space *between* table cells. Now let's look at another style attribute.

The Cellpadding Attribute

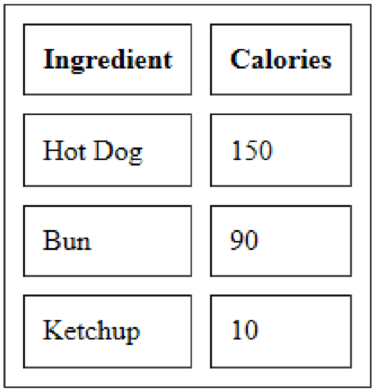
The *cellpadding* attribute controls the amount of space *inside* cells. The syntax is similar to that of other attributes:

```
cellpadding="x"
```

Replace *x* with a number indicating how much padding, in pixels, you want inside each cell. The default is 1. So there isn't a whole lot of space between words in a cell and the borders at the cell's edges. Let's take a look at what happens if you increase the cellpadding to 10:

```
<table border="1" cellspacing="10" cellpadding="10">

<tr><th>Ingredient</th><th>Calories</th></tr>

<tr><td>Hot Dog</td><td>150</td></tr>

<tr><td>Bun</td><td>90</td></tr>

<tr><td>Ketchup</td><td>10</td></tr>

</table>
```

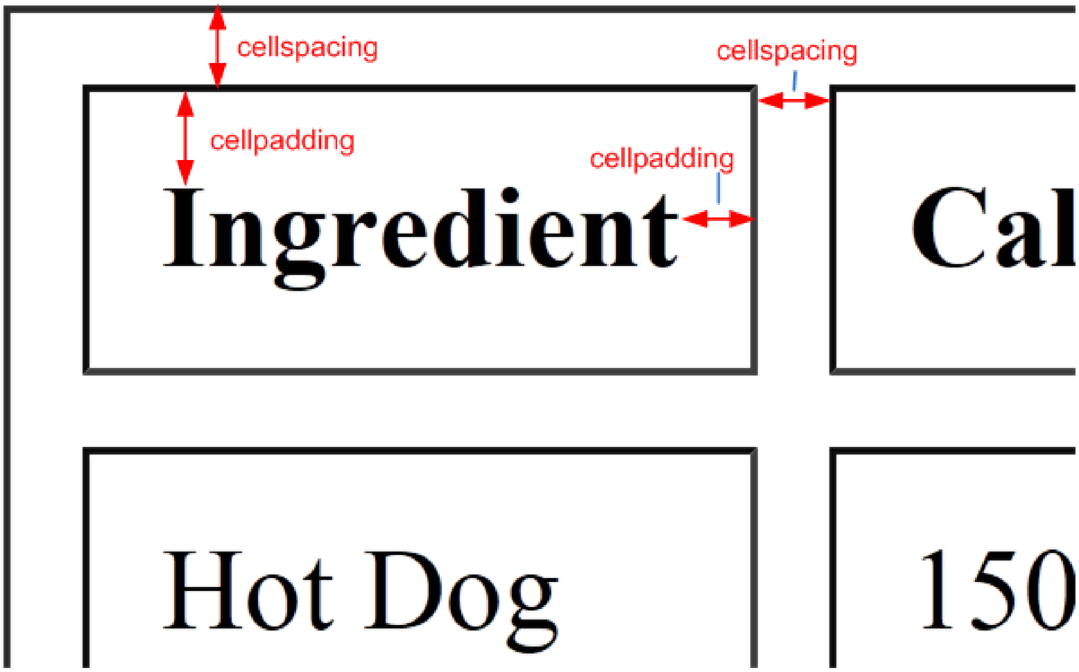
Here's how the table would look after increasing the cellpadding to 10.



Ingredient	Calories
Hot Dog	150
Bun	90
Ketchup	10

Result on increasing cellpadding to 10

Let me zoom in on a little portion of that table and point out exactly what's what.



Cell padding (inside cells) and cell spacing (between cells)

It isn't always easy to remember the difference between spacing and padding. But here's a trick you can use. Notice how each cell is like a *box* that contains some information. When you ship a cardboard box that contains something valuable, you put padding *inside* the box to protect the contents of the box:

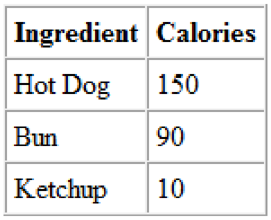
Padding goes inside a box

In HTML, padding is the same idea. It's space between the contents of the box and the edge of the box. It's always *inside* the box, not outside the box.

There's no limit to the ways you can combine the border, cellspacing, and cellpadding attributes. So feel free to experiment. You can't break or harm anything. And experimenting with different values and seeing how they play out in a browser is a great way to learn. You're free to design things as you see fit. But if you're looking to achieve the somewhat standard look you see in books and magazines, a border of 1 with no cellspacing and maybe 3 or 4 pixels of padding will likely do. The code looks like this:

```
<table border="1" cellspacing="0" cellpadding="4">

<tr><th>Ingredient</th><th>Calories</th></tr>

<tr><td>Hot Dog</td><td>150</td></tr>

<tr><td>Bun</td><td>90</td></tr>

<tr><td>Ketchup</td><td>10</td></tr>

</table>
```

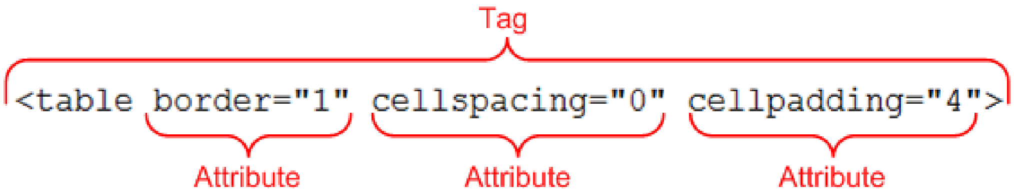
And the resulting table looks something like the example below—a single-line border around each cell, and a little extra padding inside each cell so the text inside each cell isn't touching the border line. The appearance of the border lines might vary slightly, depending on what Web browser you're using to view your page.



Ingredient	Calories
Hot Dog	150
Bun	90
Ketchup	10

Table with 1 pixel border, no spacing, and 4 pixels of padding

Feel free to set up the table in the recipe.htm page like that. Pay close attention when typing your table tag and attributes. Remember, a tag always starts with an opening angle bracket (<) and a word or abbreviation. The tag always ends with >. You can put one or more attributes inside the tag. When you do use attributes, you need a space in front of each attribute name. But the attribute name itself never contains spaces. For example, it's *cellspacing* not *cell spacing*, and its *cellpadding* not *cell padding*. Like this:



```
<table border="1" cellspacing="0" cellpadding="4">
```

Table tag with three attributes

The order of the attributes doesn't matter. For example, any of the tags below is equivalent to the sample shown above.

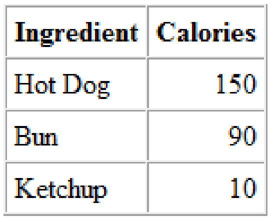
```
<table cellpadding="4" border="1" cellspacing="0">

<table cellspacing="0" cellpadding="4" border="1">

<table cellpadding="4" cellspacing="0" border="1">
```

Got it? Great! Now, let's head over to Chapter 4 and talk about some more ways to make your tables look their best.

Chapter 4

Aligning Text in Columns

The table we've created could use one minor formatting improvement. The numbers in the second column are left-aligned. Usually, numbers in a table are a little more readable if you make them right-aligned, as below.



Ingredient	Calories
Hot Dog	150
Bun	90
Ketchup	10

Numbers in right column right-aligned

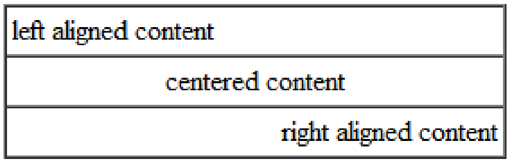
To align the contents of a cell, add an align= attribute to the <td> or <th> cell whose contents you want to align. The syntax is as follows:

```
align="value"
```

Replace the word *value* with one of the following words:

```
left

center

right
```

As their names imply, these control how text (or numbers) in the cell are aligned if there's any extra space within the cell.



left aligned content
centered content
right aligned content

Examples of alignment within a table cell

To right-align every cell in our sample table that contains a number, we need to add align="right" to the <td> tag of every cell that contains a number. And, as always, we'll need a space before the word *align*, just like every other time you've added an attribute to a tag in this course. (And just like every time you'll add an attribute to a tag in the future—no exceptions!)

As always, to make a change to the code, you'll need to open the page in your editor (Notepad, TextEdit, or whatever you've been using so far in this course). Then, add align="right" to the <td> tag for each cell that contains a number. In the image below, the new content is shown darker than what was already in the table, just to make it stand out.

```
<table border="1" cellspacing="0" cellpadding="4">

<tr><th>Ingredient</th><th>Calories</th></tr>

<tr><td>Hot Dog</td><td align="right">150</td></tr>

<tr><td>Bun</td><td align="right">90</td></tr>

<tr><td>Ketchup</td><td align="right">10</td></tr>

</table>
```

If you save that change, and refresh the page in your Web browser, the numbers in the second column will be right-aligned, as in the table shown at the start of this chapter. Take a look and see how your table looks. Easier to read, right?

Well, that's enough hands-on practice with tables for the moment. You can close and save recipe.htm if you'd like.

There are a couple more attributes I'd like to mention before ending this lesson. They only apply to very large tables, so they won't be relevant to the small table in the recipe page. And they may or may not be useful to you someday. But I'll just tell you the facts and let you decide for yourself whether you'll use them in your own work.

Merging Cells

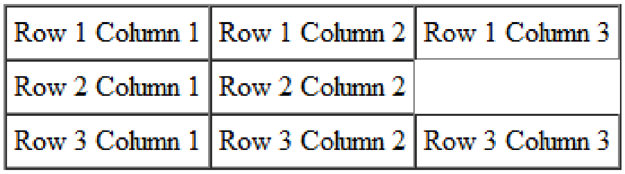
When creating tables, you'll typically want to ensure that each row contains exactly the same number of cells, as that's what gives your table an organized appearance. There may be times, however, when creating exceptionally large tables, where you might want to combine cells. I'll present a generic example to illustrate the concepts.

The table below contains three rows. The first and last rows contain three columns, and the middle row contains two columns.

```
<table cellspacing="0" cellpadding="4" border="1">

<tr>

<td>Row 1 Column 1</td><td>Row 1 Column 2</td><td>Row 1 Column 3</td>
```

```
</tr>

<tr>

<td>Row 2 Column 1</td><td>Row 2 Column 2</td>

</tr>

<tr>

<td>Row 3 Column 1</td><td>Row 3 Column 2</td><td>Row 3 Column 3</td>

</tr>

</table>
```

If you were to look at that table in a browser, it basically works. But things look a little weird where the one table cell is missing (Row 2 Column 3).



Row 1 Column 1	Row 1 Column 2	Row 1 Column 3
Row 2 Column 1	Row 2 Column 2	
Row 3 Column 1	Row 3 Column 2	Row 3 Column 3

Second row missing a cell

Of course, you could just put in the <td>...</td> tags for the missing cell. But if for whatever reason you really only need two cells in that row, you could make either of the existing cells span the width of two columns. To do that, you use a *colspan attribute* with the following syntax:

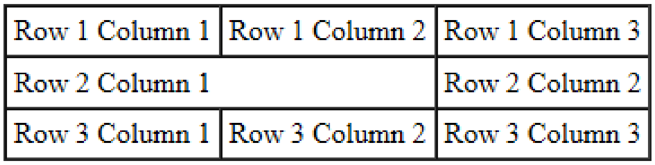
```
colspan="x"
```

Replace *x* with the number of columns to span. But the *x* shouldn't be just be any number you pick out of a hat. It should be a number that equals the number of missing columns plus one. For example, in Row 2 above, there's one column missing. So one column should span the width of two to make up for the missing column.

You could add colspan="2" to the first column in Row 2, like this:

```
<tr>

<td colspan="2">Row 2 Column 1</td><td>Row 2 Column 2</td>

</tr>
```

And then the first two columns will combine into one in the browser:



Row 1 Column 1	Row 1 Column 2	Row 1 Column 3
Row 2 Column 1		Row 2 Column 2
Row 3 Column 1	Row 3 Column 2	Row 3 Column 3

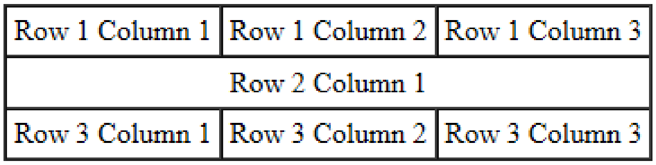
First column in Row 2 spans two columns

If you just wanted one wide column rather than two, you could remove the second pair of <td>...</td> tags in that row so there's only one cell in the row. Then, use colspan="3" to make that one cell extend across all three columns. The code below shows an example. (I used align=center in that special column to show how you can combine the attributes.)

```
<table cellspacing="0" cellpadding="4" border="1">
```

```
<tr>

<td>Row 1 Column 1</td><td>Row 1 Column 2</td><td>Row 1 Column 3</td>

</tr>

<tr>

<td colspan="3" align="center">Row 2 Column 1</td>

</tr>

<tr>

<td>Row 3 Column 1</td><td>Row 3 Column 2</td><td>Row 3 Column 3</td>

</tr>

</table>
```

Viewing that table in a browser would reveal a single cell spanning three columns in the middle row. The align="center" in that cell's td tag centers the text within the wide cell.



Row 1 Column 1	Row 1 Column 2	Row 1 Column 3
Row 2 Column 1		
Row 3 Column 1	Row 3 Column 2	Row 3 Column 3

Middle row has one cell spanning three columns

You can also make a cell expand multiple rows if you'd like. To do this, you use the *rowspan attribute* rather than *colspan attribute*. But the syntax is like other attributes you've learned about:

```
rowspan="x"
```

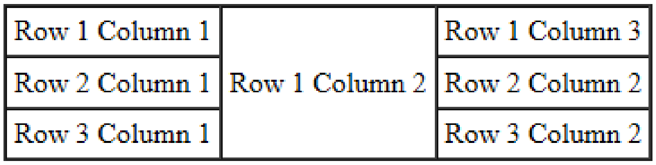
Replace *x* with the number of rows you want the column to span.

Below is a sample table where the first row (only) contains three cells (three <td>...</td> tag pairs). The middle cell contains rowspan="3"—meaning that one cell will be three rows tall. The next two rows in the table each contain only two cells (two <td>...</td> tag pairs).

```
<table cellspacing="0" cellpadding="4" border="1">

<tr>

<td>Row 1 Column 1</td><td rowspan="3">Row 1 Column 2</td><td>Row 1 Column
 3</td>

</tr>

<tr>

<td>Row 2 Column 1</td><td>Row 2 Column 2</td>

</tr>

<tr>

<td>Row 3 Column 1</td><td>Row 3 Column 2</td>
```

```
</tr>

</table>
```

And here's how that table would look in a browser:



Row 1 Column 1	Row 1 Column 2	Row 1 Column 3
Row 2 Column 1		Row 2 Column 2
Row 3 Column 1		Row 3 Column 2

Middle cell spans three rows with rowspan="3"

Notice how the middle column is three rows tall, and the text in the column is vertically centered. That's because that's the default for vertical alignment in table cells. So there was no need to use an attribute in the <td> tag to make the text vertically centered.

You might be wondering if that confusing colspan and rowspan stuff is something you'll ever use in real life. The answer to that is a resounding "maybe." It's very possible that you won't, so I wouldn't fret over it if I were you. But keep in mind that part of being a good Web developer is knowing the possibilities. Even if you don't need and can't use something right this minute, that doesn't necessarily mean you never will. So try to at least keep in the back of your mind that it's possible to make a single cell span multiple cells or rows.

Using Programmer Comments

There's one more thing I'd like to run by you before we close this lesson. This isn't directly related to tables. It's more about HTML in general. And it has to do with *programmer comments*, often called *comments* for short. Comments are like notes to yourself (or team members, if you work in a group) that you can type right into your code. The comments don't show up in the Web browser, so people visiting your site won't see them. They're not required either. Comments have no effect on how the code functions, so there'll never be a time when you'll be required to put a comment in your code.

But comments can still be useful. Most programmers don't use them enough. The idea behind comments is to simply jot down notes in your code that might serve as a reminder to yourself six months, a year, or several years in the future when you're editing or updating a page. Or if you work in a team, the comments help other developers understand the purpose of your code. It can even be helpful to quickly scan through the code looking at the plain-English comments rather than the more complex code.

The syntax for using comments in HTML is the following:

```
<!-- any text here -->
```

That's a "less than" character, an exclamation point, and exactly two hyphens (no spaces in between). The *any text here* can be anything that you want to write. These are just plain-English notes, so there are no rules on length or syntax or anything. When you've finished typing your comment, type two hyphens and a "greater than" character (>) to end the comment.

When a Web browser (or any other user agent) encounters the opening comment characters (<!--), it simply ignores everything else it sees until the --> characters. In other words, only you and other programmers who look directly at the source code will ever see the comments.

Some people like to add lots of extra characters to their comments, to make them stand out from the code and content. That's perfectly okay. Just don't use a lot of extra hyphens. Firefox and some other browsers mess up with that and end up showing the comment right in the browser. Better to use equal signs or some other characters, like this:

```
<!-- ================ This is a large comment ================== -->
```

And always, no matter what you put inside the comment, make sure your comment starts with <!-- and ends with --> as above.

In the assignment for this course, I'll suggest some places where you may want to add comments to recipe.htm. That will give you some hands-on practice, and also make it more useful as a study tool.

Let's hop over to Chapter 5 to wrap up this lesson.

Chapter 5

Summary

In this lesson you've learned about HTML tables. We use tables in Web pages to neatly organize information into rows and columns, just as we use tables in books and other publications. If you've been in the business for a while, you may have heard rumors of tables being "deprecated" and removed from the language. That's a common misconception based on the fact that we don't use tables *for page layout* anymore. There's a big difference between not using tables for page layout, and not using tables ever! We still use tables for the kinds of tables discussed in this lesson. The tags and attributes you learned about in this lesson are valid in all versions of HTML and XHTML. Page layout is a separate issue, and we'll address that in upcoming lessons.

We also talked briefly about comments, notes that you can write to yourself or other developers in a team, right in your code. The comments don't show up in a Web browser, so people browsing to your site on the Internet won't see them. But they can come in handy nonetheless when you're changing or editing your code in the future, because they serve as useful reminders. You always start a comment with <!-- and end it with -->. Those characters are the same in all versions of HTML and XHTML.

Speaking of XHTML, just about everything you've learned so far is the core HTML that applies to virtually all documents and all versions of HTML and XHTML. The code we've created is pretty much the same today as it was in the early 1990s. But like everything else high tech, HTML has evolved over the past decade. In the next lesson, we'll talk about that evolution, and we'll start focusing more on the language as it exists today—and is likely to be for many years to come. See you there!

Supplementary Material

HTML Table Tutorial

http://www.tizag.com/htmlT/tables.php

Click this link for a nice table tutorial from the Tizag website.

HTML Tables

http://www.w3schools.com/html/html_tables.asp

This page offers another tutorial on tables (though there are quite a few ads to dodge on this page!).

HTML Comments

http://www.tutorialspoint.com/html/html_comments.htm

Here's a quick tutorial on HTML comments.

FAQs

Q: I find that if I leave a cell empty by putting <td></td> in my table (with nothing in between), the cell looks weird in the browser. Is there anything I can do to make it look better?

A: Good question. Putting a nonbreaking space (&nbsp;) inside the cell, as below, should help.

```
<td>&nbsp;</td>
```

Q: You used align= to align text in table cells. Can't we use that to center text in headings, paragraphs and such too?

A: It depends. In modern versions of HTML and XHTML, the align= attribute is deprecated for virtually all tags except <td> and <th>. They'll work in the current Web browsers, but they're considered antiquated. In modern languages, we use CSS for alignment and centering, as you'll see later in this course. So the answer is "yes" if you don't mind using old, outdated techniques. The answer is "no" if you want to keep up with current and future ways of doing things.

Q: You didn't mention align="justify" when discussing aligning content in table cells. How come?

A: It's not very reliable. It works in some, but not all, browsers. Using the CSS style="text-align:justify; in the <td> tag is more reliable. We'll discuss CSS a little later in this course.

Assignment

Comments in code are always optional, never required. That's because they don't really *do* anything in the code. They're just notes to yourself (or colleagues). You might be tempted to skip commenting in code. But six months or a year later, when you find yourself updating the page for whatever reason, you might wish you'd taken a moment to jot down some comments. Just as reminders about what the code is and why you put it there.

To get a little hands-on practice, in this assignment, I'll point out some places where you might want to insert some comments in recipe.htm. Granted, you may never actually use recipe.htm in an actual website. But having the comments in there can still serve as study notes for this course. You might even consider printing the code, from Notepad or TextEdit or whatever editor you're using, just to have as a kind of cheat sheet for reviewing the code. The more you look at and read the code, the less bizarre, alien, and meaningless it'll seem. That's a big step in the direction of becoming "fluent" in HTML, a level of familiarity where you can create virtually anything with minimal effort!

You're welcome to use your own words inside the <!-- and --> characters if you want. The comments, shown in darker text below, are just some suggestions.

```
<html>

<head>

<!-- Title for browser window -->


<title>My Favorite Recipe</title>


</head>

<body>

<!-- Main title -->

<h1>Hot Dogs</h1>

<!-- Hot dog picture -->

<p><img src="hotdog.png" alt="Hot dog" width="100" /></p>

<p>Here is my gourmet hot dog recipe.</p>

<!-- Subheading -->

<h2>Ingredients</h2>

<!-- Unordered (bulleted) list -->

<ul>

<li>Hot dogs</li>

<li>Hot dog buns</li>

<li>Mustard, relish, chopped onions</li>

</ul>
```

```
<!-- Subheading -->

<h2>Directions</h2>

<!-- Ordered (numbered) list -->

<ol>

<!-- Special character for degrees below -->

<li>Preheat grill to 350&deg;.</li>

<li>Place dogs on grill, roll occasionally for even cooking.</li
>

<li>Place cooked dogs in buns.</li>

<li>Apply mustard, relish, and onions to taste.</li>

</ol>

<p>Eat and Enjoy!</p>

<!-- A table with single-line borders and right-aligned numbers
-->

<table border="1" cellspacing="0" cellpadding="4">

<!-- Row 1, header cells -->

<tr><th>Ingredient</th><th>Calories</th></tr>

<tr><td>Hot Dog</td><td align="right">150</td></tr>

<tr><td>Bun</td><td align="right">90</td></tr>

<tr><td>Ketchup</td><td align="right">10</td></tr>

</table>

<!-- Links to other pages -->

<p><a href="http://www.allrecipes.com">More Great Recipes</a>


<a href="index.htm">Home</a></p>

</body>

</html>
```

Don't forget to save and close the recipe.htm page when you're done. To verify that you did it right, open the page in a browser. You should *not* see any of the comments you put in your code.

Back to top

web-1