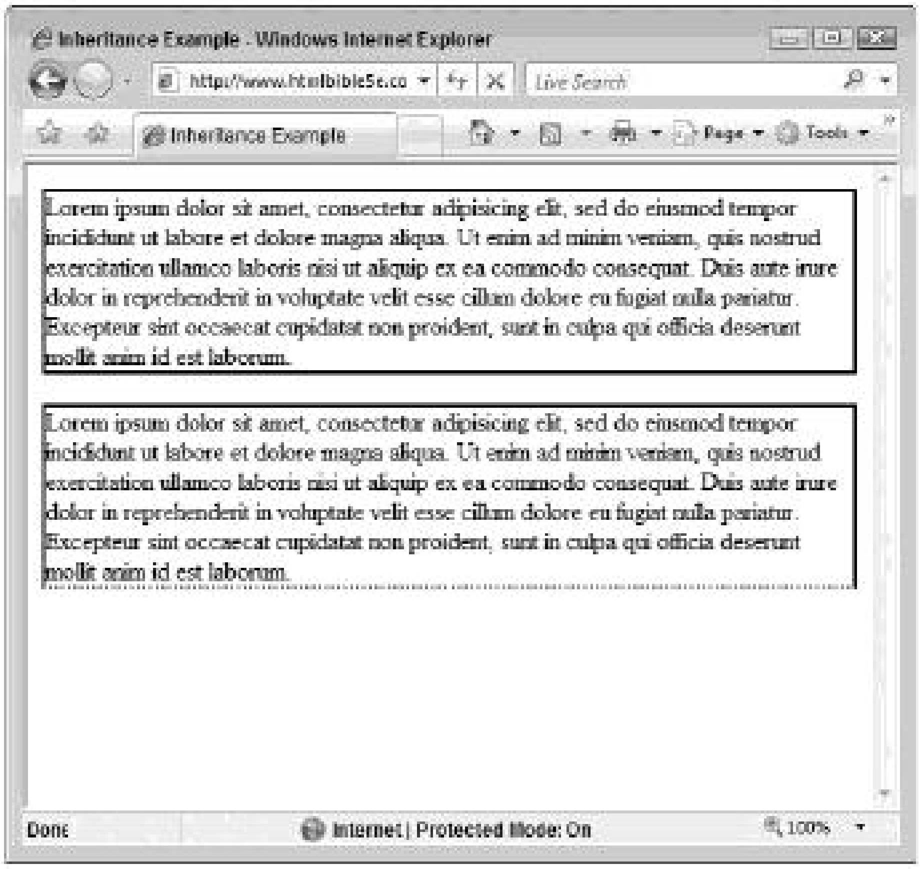
The first definition will cause all paragraph elements (`p`) in the document to have solid black borders on all sides. The second definition will cause paragraph elements with a class of `red-bottom` to have solid black borders on three sides (top, right, and left) and a dotted red border on the bottom, as shown in Figure 28-2.

FIGURE 28-2


Inheritance Example - Windows Internet Explorer
http://www.htmlbible5e.co
Live Search
Inheritance Example
Page
Tools
Done
Internet | Protected Mode: On
100%

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

The second paragraph has its bottom border specified by a specific class-based style but inherits its other borders from a generic paragraph style.

However, the second definition doesn't include property values for borders on any side other than the bottom. Where then do these other property values come from? They are inherited from the generic paragraph element definition.

Cascade

The term "cascade" has an entirely different meaning than "inheritance." The means by which styles come together to relate to a document is the cascade. Styles can be applied to a document from many different sources. These sources include the following:

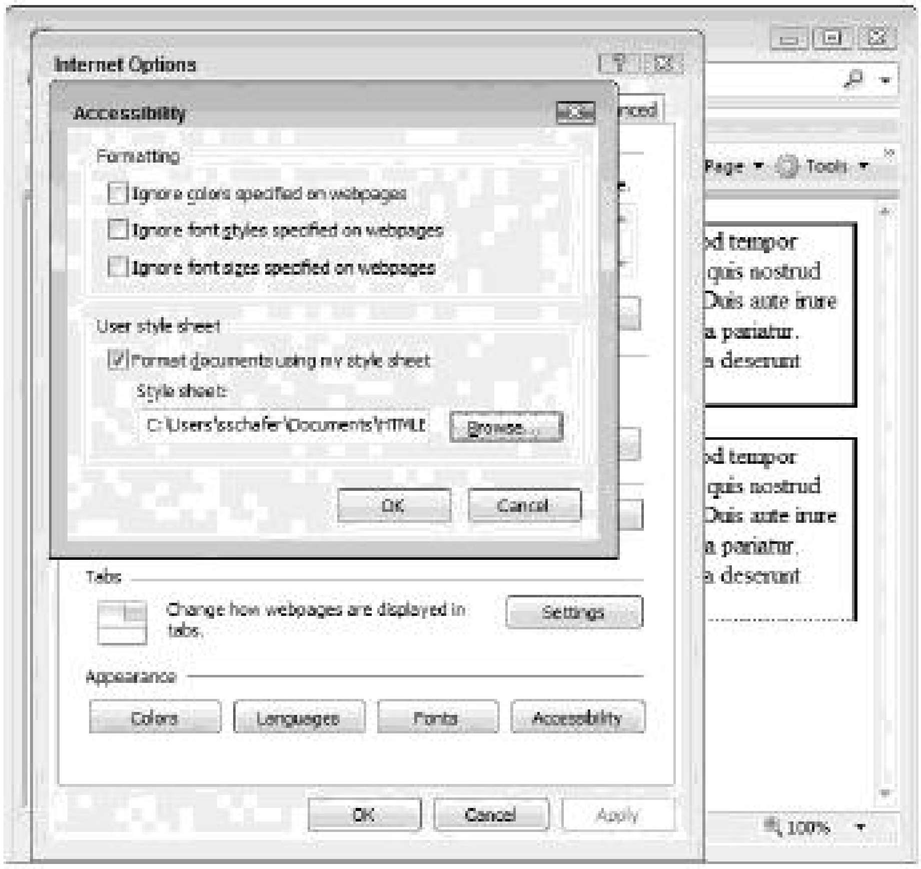
- **Author styles —** Styles that the document author includes, whether embedded directly in the head of the document, linked in as a separate style sheet (using the `link` tag, or the

CSS `@import` rule), or inline in individual elements. These styles represent the way the author intends the document to look.

- **User styles** — Styles that the end user specifies should be used for the document. These styles are selected by the end user from local style sheets and can be used to modify a document's default look. The styles can be changed in Microsoft Internet Explorer, for example, within the Accessibility section of Internet Options, as shown in Figure 28-3.
- **User agent styles** — Styles that a user agent uses by default when no other styles are specified for a particular element or document. These styles are usually very simple in nature — black text on a white background, slightly larger fonts for headings, and so on.

FIGURE 28-3

Internet Explorer users can assign default styles that override the user agent's default styles.



Accessibility

Formatting

[ ] Ignore colors specified on webpages
[ ] Ignore font styles specified on webpages
[ ] Ignore font sizes specified on webpages

User style sheet

[✓] Format documents using my style sheet

Style sheet: C:\Users\sschafer\Documents\HTMLE

Each style from each source is assigned a weight. When styles conflict between the three sources, their assigned weight is used to determine which style should apply. By default, author styles have priority over user styles, which have priority over user agent styles.

An exception can be forced by use of the `!important` rule. The author or user can tag a style with this rule, which adds extra weight to that style's priority. To mark a style as important, place the `!important` keyword after the declaration, as shown in the following example:

```
p { font-weight: bold !important; }
```

In general, important styles trump non-important styles. However, in the case of two conflicting `important` styles, the order of precedence (author, user) is reversed — user important styles trump author important styles. This gives both the user and the user agent the ability to specify their preferences (or requirements) for display — the user agent might have hardware-specific requirements and the user might have accessibility needs.

Note

When considering style precedence, you must also consider the order in which style sheets are loaded if you attach more than one sheet to a document or import one sheet inside another. In the following example, styles in the `hr-dept.css` style sheet will trump styles in the `corporate.css` style sheet because the `hr-dept.css` sheet is loaded last:

```
<link rel=stylesheet type="text/css"
      href="corporate.css" title="corporate styles">
<link rel=stylesheet type="text/css"
      href="hr-dept.css" title="hr dept styles">
```

Care should be taken to observe inheritance between the sheets. Any individual element's properties specified in the `corporate.css` sheet that are not specified in the `hr-dept.css` sheet will be inherited and applied to the document. ■

The actual means to determine the cascade sorting order is specified by the World Wide Web Consortium (W3C) as follows:

1. Take all definitions that apply to the element and property in question.
2. Sort the definitions by weight and origin: Author styles override user styles, which override user agent styles. For `!important` definitions, user styles override author styles. All `!important` definitions override normal definitions. Any imported definitions are considered to have the same origin as the style sheet that imported them.
3. Sort the definitions by specificity of selector: The more specific selectors override the more general ones.
4. Sort by the order in which the definitions were specified: If two definitions have the same weight, origin, and specificity, the last one specified prevails. Rules in imported style sheets are considered before any rules in the style sheet itself.

The styles are applied according to the order resolved by this process.

Specificity

There is one other aspect to CSS conflict resolution: specificity. To understand how specificity is used, consider the following style definitions:

```
div p  { color:  red; }
p      { color:  blue; }
```

Given the discussion thus far in the chapter, you would expect that the font in every paragraph element, including those in `div` elements, would be rendered in blue. However, that's not the case. Paragraph elements contained within div elements would have their text rendered in red. The first definition is more specific — it specifies paragraph elements that are children of div elements. Therefore, it carries more weight than the more generic definition.

As with cascade precedence, the W3C has a specification for calculating a definition's specificity value based on the selector:

1. Count the number of ID attributes in the selector and assign that number to A.
2. Count the number of other attributes and pseudo-classes in the selector and assign that number to B.
3. Count the number of element names and pseudo-elements in the selector and assign that number to C.
4. Concatenate the values to make one number, ABC. That number is the definition's specificity.

Note that pseudo-elements are not given specificity and are ignored in the preceding calculation.

For example, consider the following selectors and their resultant specificity values:

```
*               A=0 B=0 C=0 -> specificity =   0
p               A=0 B=0 C=1 -> specificity =   1
div p           A=0 B=0 C=2 -> specificity =   2
ul ol+li        A=0 B=0 C=3 -> specificity =   3
h1 + *[REL=up]  A=0 B=1 C=1 -> specificity =  11
ul ol li.red    A=0 B=1 C=3 -> specificity =  13
li.red.level    A=0 B=2 C=1 -> specificity =  21
#columnhead     A=1 B=0 C=0 -> specificity = 100
```

Note that the selector #columnhead, which refers to an element with a specific ID attribute, is given a high specificity, whereas the wildcard selector (*) is given a low specificity, as you would expect.

Summary

CSS can be a complex beast if documents have several style sheets or otherwise create competing and conflicting styles. However, definite rules are in place to handle competing sheets and individual competing styles — rules that effectively address the needs of a document's author, the user of a document, and the user agent displaying the document. Understanding the style sheet cascade, style inheritance, and specificity eliminates any doubt as to how your documents will be displayed.

CHAPTER 29

Font Properties

As previously mentioned throughout this book, the Web began as a vehicle for displaying very plain documents. The documents in question were of the research variety, needing only basic font handling, tables for data display, and the inclusion of graphics.

However, the Web has come a long way from that simple beginning. As more entities embraced the medium, the technology became more robust and able to handle more desktop publishing–like capabilities. Today's Web technologies can produce documents almost as rich in content and presentation as those produced by modern, dedicated publishing programs.

The most important characteristics are typography and layout. This chapter covers typography — namely, fonts — and how CSS handles them.

IN THIS CHAPTER

Understanding Fonts

Font Types

Font Sizing

Font Styling

Line Spacing

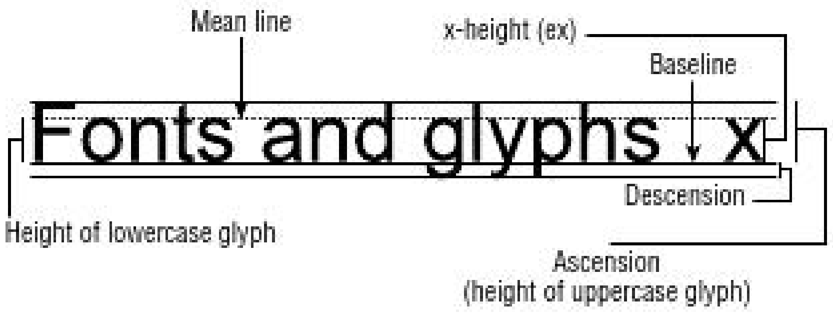
Embedding Fonts in a Document

Understanding Fonts

Fonts are stylized collections of letters and symbols, known as *glyphs*. Fonts can be used to convey information — for example, specialized fonts can provide special characters or symbols. Although fonts can be quite different from one another, they share the same basic characteristics, as shown in Figure 29-1.

FIGURE 29-1

Font characteristics



These elements are defined as follows:

- **Baseline** — The line on which glyphs of the font sit
- **Ascension** — The highest point reached by most capital glyphs in the font. Note that technically the ascension is the point at which the highest glyph reaches, as some fonts have special, ornate characters that reach higher than other, normal characters.
- **Descension** — The lowest position that some glyphs, such as *p*, *g*, or *q*, reach
- **Mean line** — The highest point that lowercase glyphs reach
- **x-height** — The height of the letter *x* in the font. Note that this is usually the same as the mean line, but occasionally the two heights are different. In addition, this value exists for all fonts, whether they contain an *x* or not.

Fonts are spaced vertically according to a system similar to ruled paper. Vertical font measurements, such as line spacing or leading, are typically measured between the baselines of text, at least as far as CSS is concerned. The spacing between individual glyphs is letter-spacing, and can vary between fonts and be adjusted within a font.

CSS offers many properties to control the fonts in your documents.

Font Types

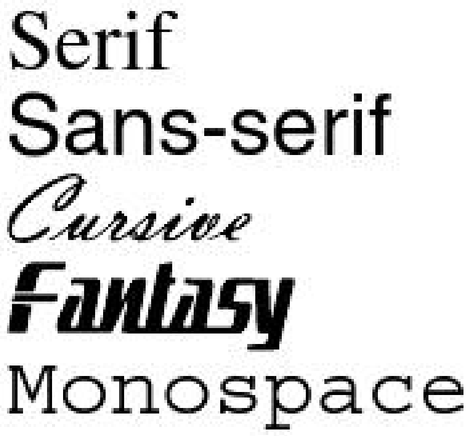
CSS supports five different font family types (see Figure 29-2). These general types can be used to apprise a user agent of the type of font face it should use. The five families are as follows:

- **Serif** — Serif fonts have ornamentation on each glyph. Typically, serif fonts are used in body text. The finishing strokes, flared or tapering ends, or serifed endings, make the lines of characters flow and tend to be easier on the eyes.
- **Sans-serif** — These fonts are fairly plain, having little or no ornamentation on their glyphs. Sans-serif fonts are typically used for headings or other large areas of emphasis.
- **Cursive** — Cursive fonts are quite ornate, approximating cursive writing. Such fonts should be used only in extreme cases where the emphasis is on ornamentation, rather than legibility.

- **Fantasy** — Fantasy fonts, much like cursive fonts, emphasize ornamentation over legibility. Fantasy fonts come in many styles but still retain the basic shape of letters. Like cursive fonts, fantasy fonts are generally used for logos and other ornamentation purposes where legibility is secondary to a particular look or design.
- **Monospace** — Monospace fonts come in serif and sans-serif varieties but all share the same attribute: All characters in the font have the same width. The effect is much like characters on a text-based terminal or typewriter. Such fonts are generally used in code listings and other listings approximating terminal output.

FIGURE 29-2

The five CSS-supported font types



The `font-family` property defines the font or fonts that should be used in the document or specific element to which the property is attached. The property has the following format:

```
font-family: [[ <family-name> | <generic-family> ][,
  <family-name> | <generic-family>]*] ;
```

Essentially, the property defines one or more font families that should be used, via an actual font name or a generic name. For example, to select a sans-serif font, you might use a definition similar to the following:

```
font-family: Verdana, Arial, Helvetica, Sans-Serif;
```

Tip

If the font family names contain any spaces they should be enclosed in quotes. ■

Note that this definition uses three specific family names (Verdana, Arial, Helvetica) and a generic family name (Sans-Serif) for versatility. The definition instructs the user agent that the sans-serif font Verdana should be used. If it is unavailable, the Arial font (popular on Windows-based platforms) should be used. If neither of those fonts is available, the Helvetica font should be used (popular on Macintosh-based platforms and other PostScript-based systems).

If none of the previously specified fonts are available, the user agent should use its default sans-serif font.

Tip

The preceding font-family definition is a good, universal sans-serif font specification that can be used for any platform. Likewise, the following definition can be used for a universal serif font specification:

```
font-family:  Palatino, "Times New Roman", "Times Roman", Serif; ■
```

The `font-family` definition doesn't control the font variant (bold, italic, and so on), size, letter spacing, and so forth. It does specify the font that should be used as the basis for fonts in the element where the `font-family` definition is placed. Individual font variant tags and elements (`<b>`, `<i>`, and so on) determine the variant of the font used when those variant elements are encountered by the browser. If the base font cannot be used for the variant, then the browser substitutes another font in the current font's stead.

Style definitions to set up a document in traditional serif font body text and bold sans-serif font headings would resemble the following:

```
body { font-family:  Palatino, "Times New Roman", "Times Roman", Serif; }
h1, h2, h3, h4, h5, h6 {
  font-family: Verdana, Arial, Helvetica, Sans-Serif;
  font-weight: bold; }
```

Font Sizing

Two properties can be used to control font sizing: `font-size` and `font-size-adjust`. Both properties can adjust a font absolutely or relative to the current font size. Possible value metrics are shown in Table 29-1.

The `font-size` property is used to set the actual size of the current font. For example, you could set an absolute font size of 12pt with the following property:

```
font-size: 12pt;
```

Likewise, you can adjust the font size relative to the current font size. For example, to set the font size to double its current size, you could use a property similar to this:

```
font-size: 200%;
```

TABLE 29-1

Font Size Value Metrics

Metric	Description
Absolute size keywords	Keywords corresponding to user agent absolute font sizes. These keywords include `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, and `xx-large`.
Relative size keywords	Keywords corresponding to user agent relative font sizes. These keywords include `larger` and `smaller`.
Absolute size	An absolute value corresponding to a font size. Negative values are not supported, but supported values include point sizes (e.g., `12pt`) and, optionally (although not as exact), other size values such as pixels (e.g., `10px`).
Percentage size	A percentage corresponding to a percentage of the current font. These values can be expressed in actual percentages (e.g., `150%`) or other relative metrics such as ems (e.g., `1.5em`).

The `font-size-adjust` property adjusts the aspect of the current font. The aspect of a font is the ratio between its size and x-height values. Tweaking this aspect can improve the legibility of some fonts at smaller sizes, but usually the aspect should not be changed.

Font Styling

Four properties can be used to affect font styling: `font-style`, `font-variant`, `font-weight`, and `font-stretch`. The syntax of each is shown in the following listing:

```
font-style: normal | italic | oblique;
font-variant: normal | small-caps;
font-weight: normal | bold | bolder | lighter | 100 | 200 |
  300 | 400 | 500 | 600 | 700 | 800 | 900;
font-stretch: normal | wider | narrower | ultra-condensed |
  extra-condensed | condensed | semi-condensed | semi-expanded |
  expanded | extra-expanded | ultra-expanded;
```

The `font-style` property controls the italic style of the text, whereas the `font-weight` property controls the bold style of the text. The other two properties control other display attributes of the font; `font-variant` controls whether the font is displayed in small caps, and `font-stretch` does exactly what its name suggests — stretches the font by adjusting its letter spacing.

The various values for the `font-weight` property can be broken down as follows:

- `100-900` — The font's darkness, where `100` is the lightest and `900` the darkest. Various numbers correspond to other values, as described in the following bulleted points.
- `lighter` — Specifies the next lightest setting for a font unless the font weight is already near the weight value corresponding to `100`, in which case it stays at `100`.

- `normal` — The normal darkness for the current font; it corresponds to weight `400`.
- `bold` — The darkness corresponding to the font's bold variety; it corresponds to weight `700`.
- `bolder` — Specifies the next darkest setting for a font unless the font weight is already near the weight value corresponding to `900`, in which case it stays at `900`.

The `font-style` and `font-weight` properties can be used to control a font's bold and italic properties without coding document text directly with italic (`i`) and bold (`b`) elements. For example, you might define a bold variety of a style using definitions similar to the following:

```
p { font-family:  Palatino, "Times New Roman", "Times Roman", Serif; }
p.bold { font-weight: bold; }
```

The `bold` class of the paragraph element inherits the base font from its parent, the paragraph element. The `font-weight` property in the `bold` class of the paragraph element simply makes such styled elements render as a bold variety of the base font.

Line Spacing

The `line-height` property controls the line height of text. The line height is the distance between the baseline of two vertically stacked lines of text. This value is also known as *leading*.

Note
Refer to Figure 29-1 for an illustration of the baseline of a font. ■

The `line-height` property has the following syntax:

```
line-height: normal | <number> | <length> | <percentage>
```

This property sets the size of the surrounding box of the element for which it is applied, affecting the vertical distance between text lines. The normal value sets the line height to the default size for the current font. Specifying a number (for example, 2) causes the current line height to be multiplied by the number specified. Absolute lengths (for example, `1.2em`) cause the line height to be set to that absolute value. A percentage value is handled like a number value; the percentage is applied to the current font's value. Note that this property does not change the size of the font, only the distance between the lines of text.

For example, the following two definitions both set a class up to double-space text:

```
p.doublespace { line-height: 2; }
p.doublespace { line-height: 200%; }
```

Embedding Fonts in a Document

Two technologies exist to enable you to embed fonts in your documents, though support for either is almost non-existent. Embedding fonts enables your readers to download the specific font to their local machine so your documents use the *exact* font you designate.

Unfortunately, as with most progressive Web technologies, the market is split into distinct factions:

- OpenType is a standard developed by Microsoft and Adobe Systems. OpenType fonts, thanks to the creators of the standard, share similar traits with PostScript and TrueType fonts used in other publishing applications. Currently, only Internet Explorer supports OpenType.
- TrueDoc is a standard developed by BitStream, a popular font manufacturer. Currently, only Netscape-based browsers natively support TrueDoc fonts, but BitStream does make an ActiveX control for support on Internet Explorer.

Note

Even when a font is available for low cost or without cost, that doesn't mean you can reuse it, especially in a commercial application. When acquiring fonts for use on the Web, make sure that you have the appropriate rights for the use you intend. ■

To embed OpenType fonts in your document, you use an `@font-face` definition in the style section of your document. The `@font-face` definition has the following syntax:

```
@font-face { font-definition }
```

The `font-definition` contains information about the font, including stylistic information and the path to the font file. This information is contained in typical `property: value` form, similar to the following:

```
@font-face {
  font-family: Dax;
  font-weight: bold;
  src: url('http://www.example.com/fontdir/Dax.pfr');
}
```

To embed TrueDoc fonts in your document, you use the link tag (`<link>`) in a format similar to the following:

```
<link rel="fontdef" src="http://www.example.com/fontdir/Amelia.pfr" />
```

To use TrueDoc fonts in Internet Explorer you also have to include the TrueDoc ActiveX control using code like the following:

```
<script language="JavaScript" src="http://www.truedoc.com/activex/tdserver.js">
</script>
```

Tip

Several fonts are available for use from the TrueDoc website: `www.truedoc.com`. ■

Embedding fonts is not recommended for several reasons:

- The two standards make supporting embedded fonts difficult.
- Embedded fonts increase the download time of your document and increase the overall load on the user agent.

- Embedded fonts decrease the flexibility of your documents, limiting how user agents can adjust the display of text.

Instead of using embedded fonts, I recommend that you stick to CSS definitions for specifying font attributes. If you know your audience and their platform and you need your document to look *exactly* as you intend, investigate embedded fonts.

Summary

This chapter is the first of the topical coverage chapters in this CSS part. In this chapter, you learned about fonts — how to present them, control them, and even embed them, if you so choose. The following several chapters continue to present CSS subject matter in concrete, related chunks.

CHAPTER 30

Text Formatting

IN THIS CHAPTER

Aligning Text

Indenting Text

Controlling White Space Within Text

Controlling Letter and Word Spacing

Specifying Capitalization

Using Text Decorations

Autogenerated Text

Using CSS Table Properties

Controlling Table Attributes

Table Layout

Aligning and Positioning Captions

The Web was initially text-based, and text is still a major part of online content today. CSS offers many styles for text formatting, from simple justification to autogenerated text. Although CSS includes options for page layout without tables, it also includes styles for formatting HTML tables. This chapter covers the basics of text and table formatting with CSS.

Aligning Text

Multiple properties in CSS control the formatting of text. Several properties enable you to align text horizontally and vertically — aligning with other pieces of text or other elements around them.

Controlling horizontal alignment

You can use the `text-align` property to align blocks of text in four basic ways: left, right, center, or full. The following code and the output displayed in Figure 30-1 show the effect of the justification settings:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Text Justification</title>
  <style type="text/css">
    p.left { text-align: left;}
    p.right { text-align: right;}
    p.center { text-align: center;}
    p.full { text-align: justify;}
  </style>
</head>
<body>
```

```
<div style="margin: 50px">
<h3>Left Justified (default)</h3>
<p class="left">The Oasis boasts three saunas, two whirlpools, and
a full-size swimming pool for the use of our clients. Each of these
facilities has a small usage fee, but many services include the use
of one or more of these facilities--be sure to ask your service
consultant about our many combination packages.</p>
<h3>Right Justified</h3>
<p class="right">The Oasis boasts three saunas, two whirlpools, and
a full-size swimming pool for the use of our clients. Each of these
facilities has a small usage fee, but many services include the use
of one or more of these facilities--be sure to ask your service
consultant about our many combination packages.</p>
<h3>Center Justified</h3>
<p class="center">The Oasis boasts three saunas, two whirlpools, and
a full-size swimming pool for the use of our clients. Each of these
facilities has a small usage fee, but many services include the use
of one or more of these facilities--be sure to ask your service
consultant about our many combination packages.</p>
<h3>Fully Justified</h3>
<p class="full">The Oasis boasts three saunas, two whirlpools, and
a full-size swimming pool for the use of our clients. Each of these
facilities has a small usage fee, but many services include the use
of one or more of these facilities--be sure to ask your service
consultant about our many combination packages.</p>
</div>
</body>
</html>
```

Note that the default justification is left; that is, the lines in the block of text are aligned against the left margin, and the lines wrap where convenient on the right, leaving a jagged right margin.

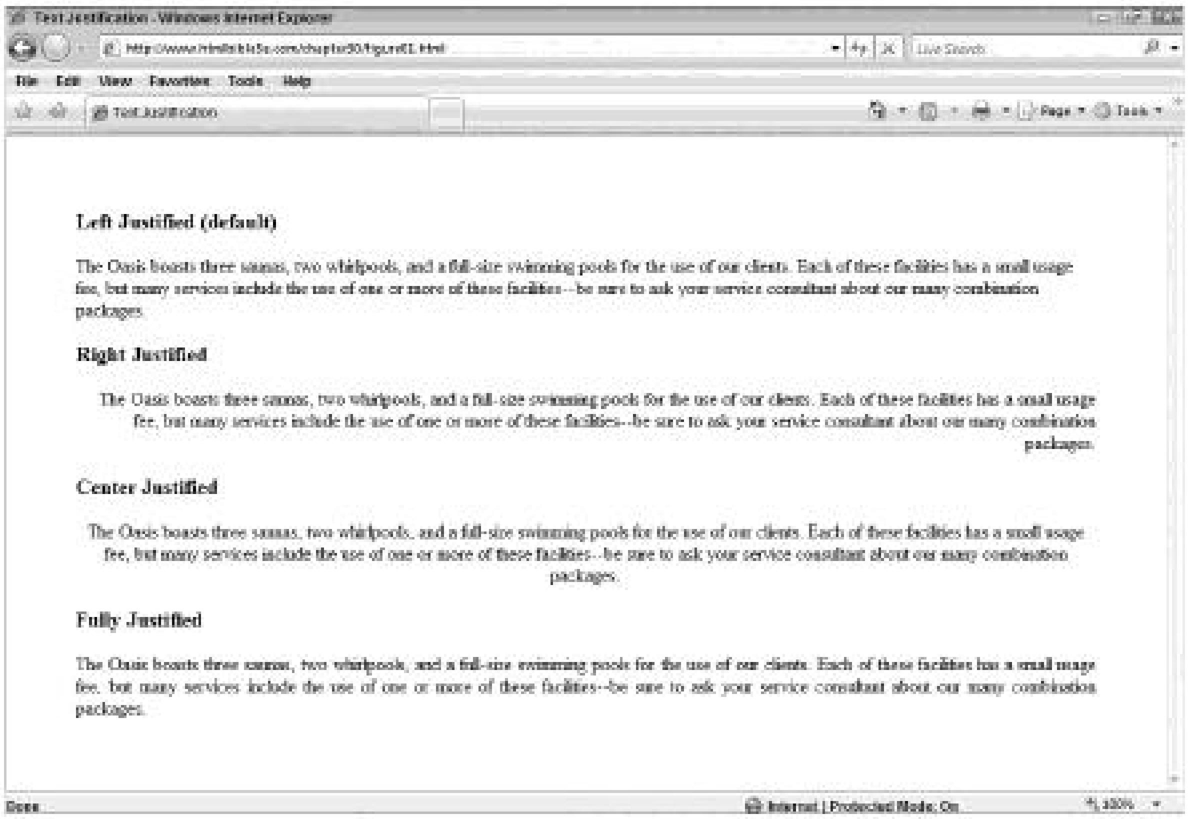
In addition to the four standard alignment options, you can also use `text-align` to align columnar data in tables to a specific character. For example, the following code results in the data in the Balance column being aligned on the decimal place:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Table Column Justification</title>
  <style type="text/css">
    td.dec { text-align: ".";}
  </style>
</head>
<body>
  <table border="1">
  <tr>
    <th>Customer</th>
    <th>Balance</th>
  </tr>
```

```
    <tr>
      <td>Wendy Weatherbee</td>
      <td class="dec">$50.95</td>
    </tr>
    <tr>
      <td>Katy Keene</td>
      <td class="dec">$284.99</td>
    </tr>
    <tr>
      <td>Elizabeth Cooper</td>
      <td class="dec">$90.99</td>
    </tr>
    <tr>
      <td>Ronnie Lodge</td>
      <td class="dec">$525.99</td>
    </tr>
    <tr>
      <td>Nancy Woods</td>
      <td class="dec">$410.99</td>
    </tr>
    </table>
  </body>
  </html>
```

FIGURE 30-1

The four types of text justification


Text Justification - Windows Internet Explorer
http://www.htmlbible5e.com/chapter30/figure01.html
Live Search
File Edit View Favorites Tools Help
Text Justification
Page Tools
Done
Internet | Protected Mode: On
100%

Left Justified (default)

The Oasis boasts three saunas, two whirlpools, and a full-size swimming pools for the use of our clients. Each of these facilities has a small usage fee, but many services include the use of one or more of these facilities--be sure to ask your service consultant about our many combination packages.

Right Justified

The Oasis boasts three saunas, two whirlpools, and a full-size swimming pools for the use of our clients. Each of these facilities has a small usage fee, but many services include the use of one or more of these facilities--be sure to ask your service consultant about our many combination packages.

Center Justified

The Oasis boasts three saunas, two whirlpools, and a full-size swimming pools for the use of our clients. Each of these facilities has a small usage fee, but many services include the use of one or more of these facilities--be sure to ask your service consultant about our many combination packages.

Fully Justified

The Oasis boasts three saunas, two whirlpools, and a full-size swimming pools for the use of our clients. Each of these facilities has a small usage fee, but many services include the use of one or more of these facilities--be sure to ask your service consultant about our many combination packages.

Note
Columnar alignment using the `text-align` property is not well supported in current user agents. You should test your target agents to ensure compliance before using `text-align` this way. ■

Controlling vertical alignment

In addition to aligning text horizontally, CSS also enables you to align text vertically via the `vertical-align` property. The `vertical-align` property supports the following values:

- `baseline` — This is the default vertical alignment; text uses its baseline to align to other objects around it.
- `sub` — This value causes the text to descend to the level appropriate for subscripted text based on its parent's font size and line height. (This value has no effect on the size of the text, only its position.)
- `super` — This value causes the text to ascend to the level appropriate for superscripted text based on its parent's font size and line height. (This value has no effect on the size of the text, only its position.)
- `top` — This value causes the top of the element's bounding box to be aligned with the top of the element's parent bounding box.
- `text-top` — This value causes the top of the element's bounding box to be aligned with the top of the element's parent text.
- `middle` — This value causes the text to be aligned using the middle of the text and the midline of objects around it.
- `bottom` — This value causes the bottom of the element's bounding box to be aligned with the bottom of the element's parent bounding box.
- `text-bottom` — This value causes the bottom of the element's bounding box to be aligned with the bottom of the element's parent text.
- `length` — This value causes the element to ascend (positive value) or descend (negative value) by the value specified.
- `percentage` — This value causes the element to ascend (positive value) or descend (negative value) by the percentage specified. The percentage is applied to the element's line height.

The following code and the output displayed in Figure 30-2 show the effect of each value:

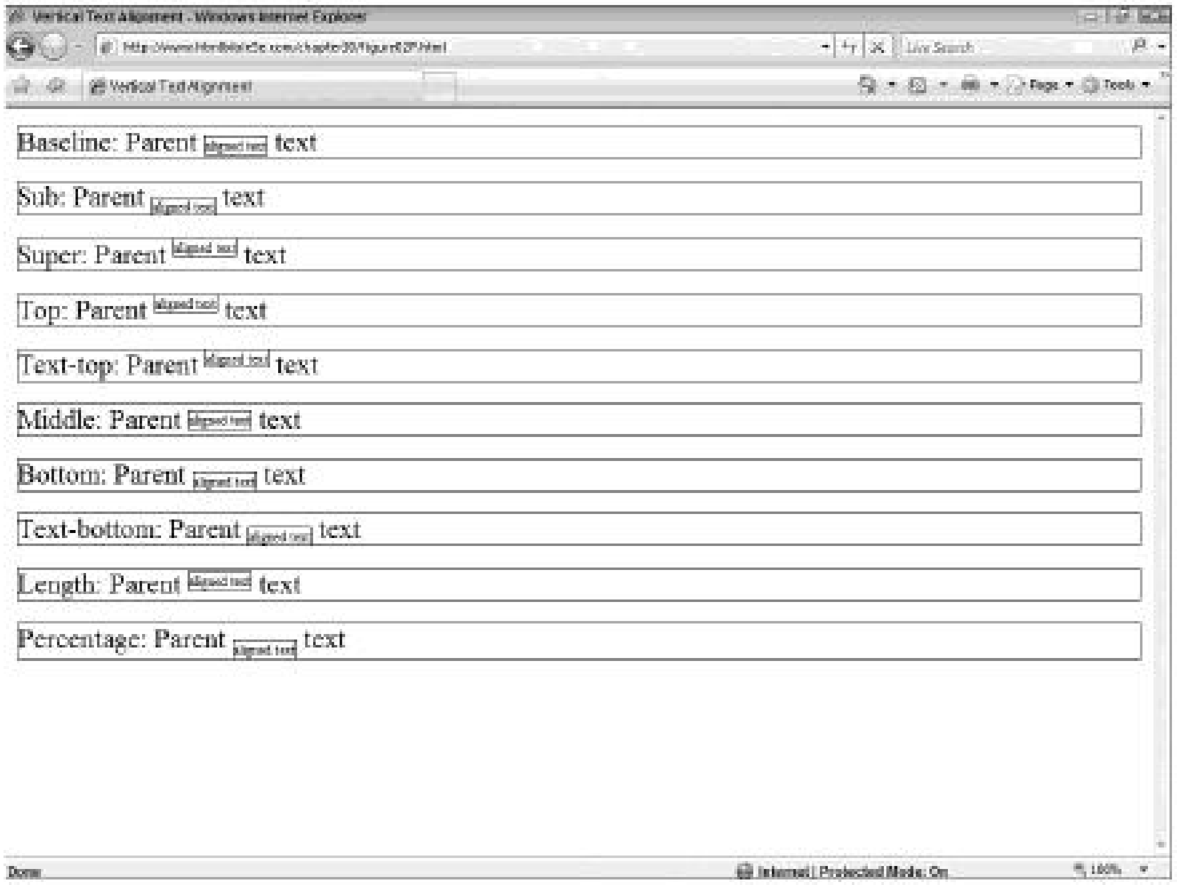
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Vertical Text Alignment</title>
  <style type="text/css">
    .baseline { vertical-align: baseline;}
```

```
      .sub { vertical-align: sub;}
      .super { vertical-align: super;}
      .top { vertical-align: top;}
      .text-top { vertical-align: text-top;}
      .middle { vertical-align: middle;}
      .bottom { vertical-align: bottom;}
      .text-bottom { vertical-align: text-bottom;}
      .length { vertical-align: .5em;}
      .percentage { vertical-align: -50%;}
       /* All elements get a border */
       body * { border: 1px solid black;}
       /* Parent (paragraph) font larger for visibility
       p { font-size: 150%;}
       /* Reduce the spans' font by 50% */
       p * { font-size: 50%;}
    </style>
  </head>
  <body>
    <p>Baseline: Parent
      <span class="baseline">aligned text</span> text</p>
    <p>Sub: Parent
      <span class="sub">aligned text</span> text</p>
    <p>Super: Parent
      <span class="super">aligned text</span> text</p>
    <p>Top: Parent
      <span class="top">aligned text</span> text</p>
    <p>Text-top: Parent
      <span class="text-top">aligned text</span> text</p>
    <p>Middle: Parent
      <span class="middle">aligned text</span> text</p>
    <p>Bottom: Parent
      <span class="bottom">aligned text</span> text</p>
    <p>Text-bottom: Parent
      <span class="text-bottom">aligned text</span> text</p>
    <p>Length: Parent
      <span class="length">aligned text</span> text</p>
    <p>Percentage: Parent
      <span class="percentage">aligned text</span> text</p>
  </body>
  </html>
```

Of course, text isn't the only element that can be affected by the `vertical-align` property. Figure 30-3 shows an image next to text. The image has the `vertical-align` property set to middle. Note how the midpoint of the image is aligned to the text beside it.

FIGURE 30-2

The effect of various vertical-align settings. (Borders were added to the text to help contrast the alignment.)


Vertical Text Alignment - Windows Internet Explorer
Vertical Text Alignment
Done
Internet | Protected Mode: On
100%

Baseline: Parent aligned text text

Sub: Parent aligned text text

Super: Parent aligned text text

Top: Parent aligned text text

Text-top: Parent aligned text text

Middle: Parent aligned text text

Bottom: Parent aligned text text

Text-bottom: Parent aligned text text

Length: Parent aligned text text

Percentage: Parent aligned text text

Indenting Text

You can use the `text-indent` property to indent the first line of an element. For example, to indent the first line of a paragraph of text by 25 pixels, you could use code similar to the following:

```
 <p style="text-indent: 25px;">The Oasis boasts three saunas, two
whirlpools, and a full-size swimming pool for the use of our clients.
Each of these facilities has a small usage fee, but many services
include the use of one or more of these facilities--be sure to ask
your service consultant about our many combination packages.</p>
```

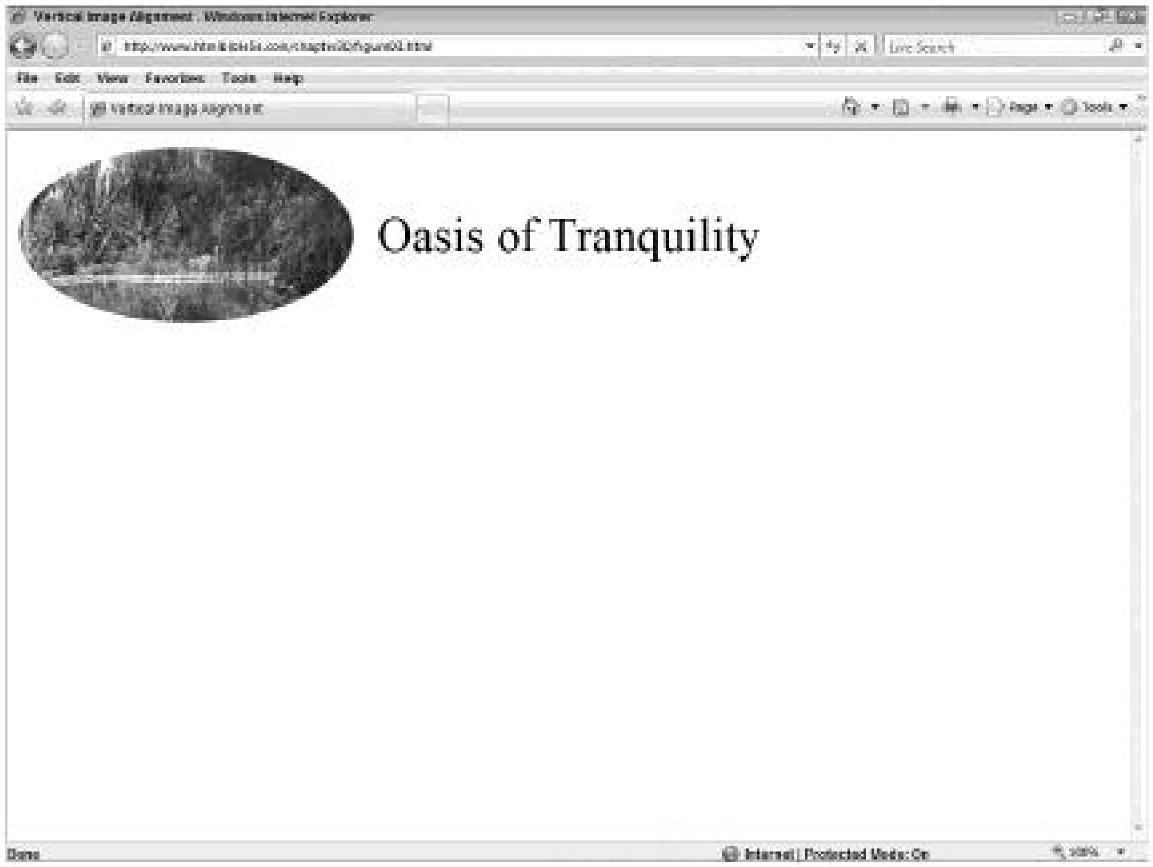
Note that the `text-indent` property indents only the first line of the element. If you want to indent the entire element, use the appropriate `margin` properties instead.

Cross-Ref

See Chapter 32 for more information about the `margin` properties. ■

FIGURE 30-3

The vertical-align property can be used to vertically align most elements.


Vertical Image Alignment - Windows Internet Explorer
File Edit View Favorites Tools Help
Vertical Image Alignment
Oasis of Tranquility
Done
Internet | Protected Mode: On

You can specify the indent as a specific value (`1in`, `25px`, and so on), or as a percentage of the containing element's width. When specifying the indent as a percentage, the width of the containing element(s) will play a prominent role in the actual size of the indentation. Therefore, when you want a uniform indent, use a specific value.

Controlling White Space Within Text

White space is typically not a concern in HTML documents. However, at times you'll want better control over how white space is interpreted and how certain elements line up to their siblings.

Clearing floating objects

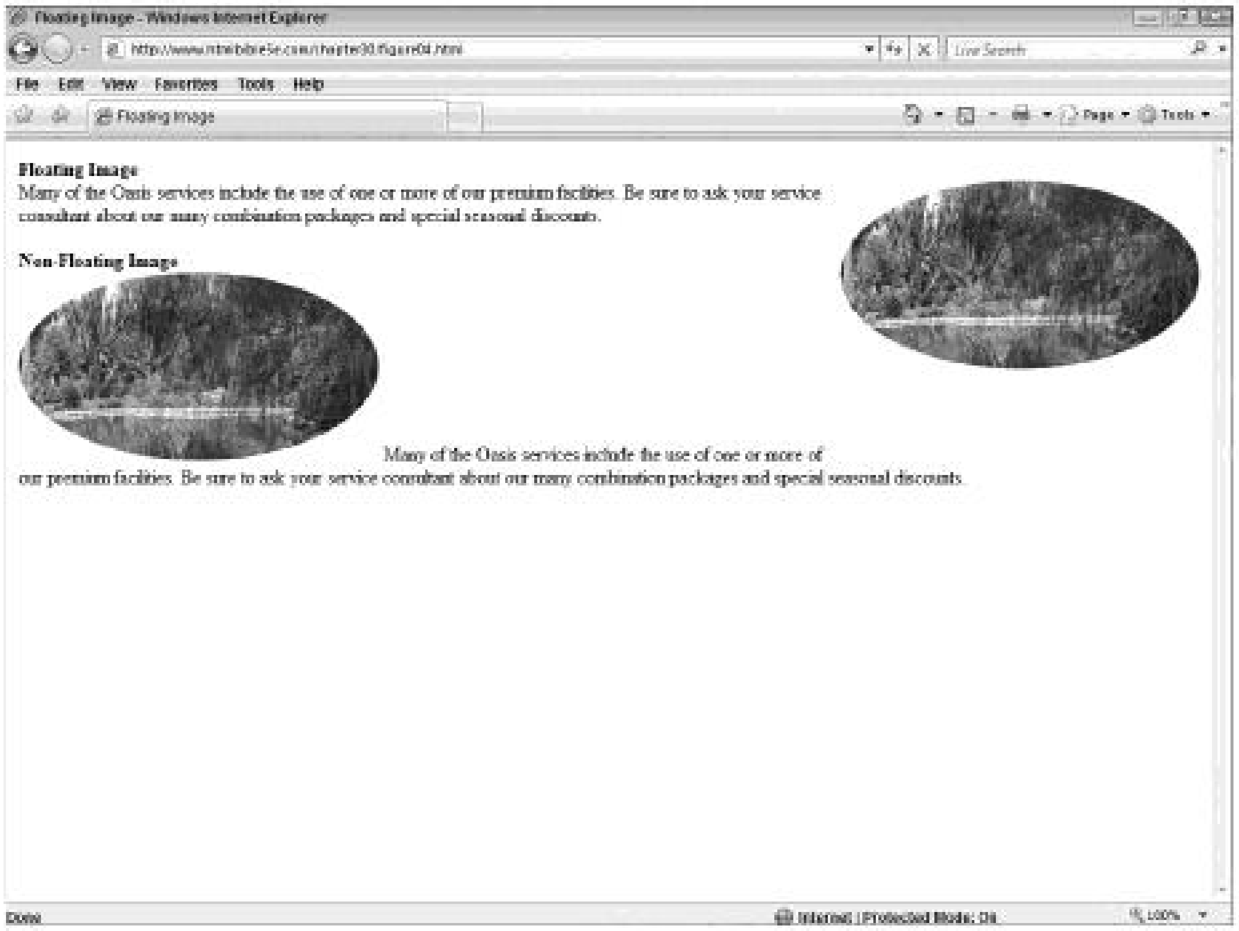
The `float` property can cause elements to ignore the normal flow of the document and "float" against a particular margin. For example, consider the following code, whose resulting output is shown in Figure 30-4:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
```

```
 <head>
   <title>Floating Image</title>
 </head>
 <body>
   <p><b>Floating Image</b><br>
   <img src="sphere.png" style="float: right;" />
   The Oasis boasts three saunas, two whirlpools, and a full-size
swimming pool for the use of our clients. Each of these facilities
has a small usage fee, but many services include the use of one or
more of these facilities--be sure to ask your service consultant
about our many combination packages.</p>
   <p><b>Non-Floating Image</b><br>
   <img src="sphere.png" />
   The Oasis boasts three saunas, two whirlpools, and a full-size
swimming pool for the use of our clients. Each of these facilities
has a small usage fee, but many services include the use of one or
more of these facilities--be sure to ask your service consultant
about our many combination packages.</p>
 </body>
 </html>
```

FIGURE 30-4

Floating images can add a dynamic feel to your document.


Floating Image - Windows Internet Explorer
http://www.htmlbible5e.com/chapter30/figure04.html
Live Search
File Edit View Favorites Tools Help
Floating Image
Page Tools
Done
Internet | Protected Mode: On
100%

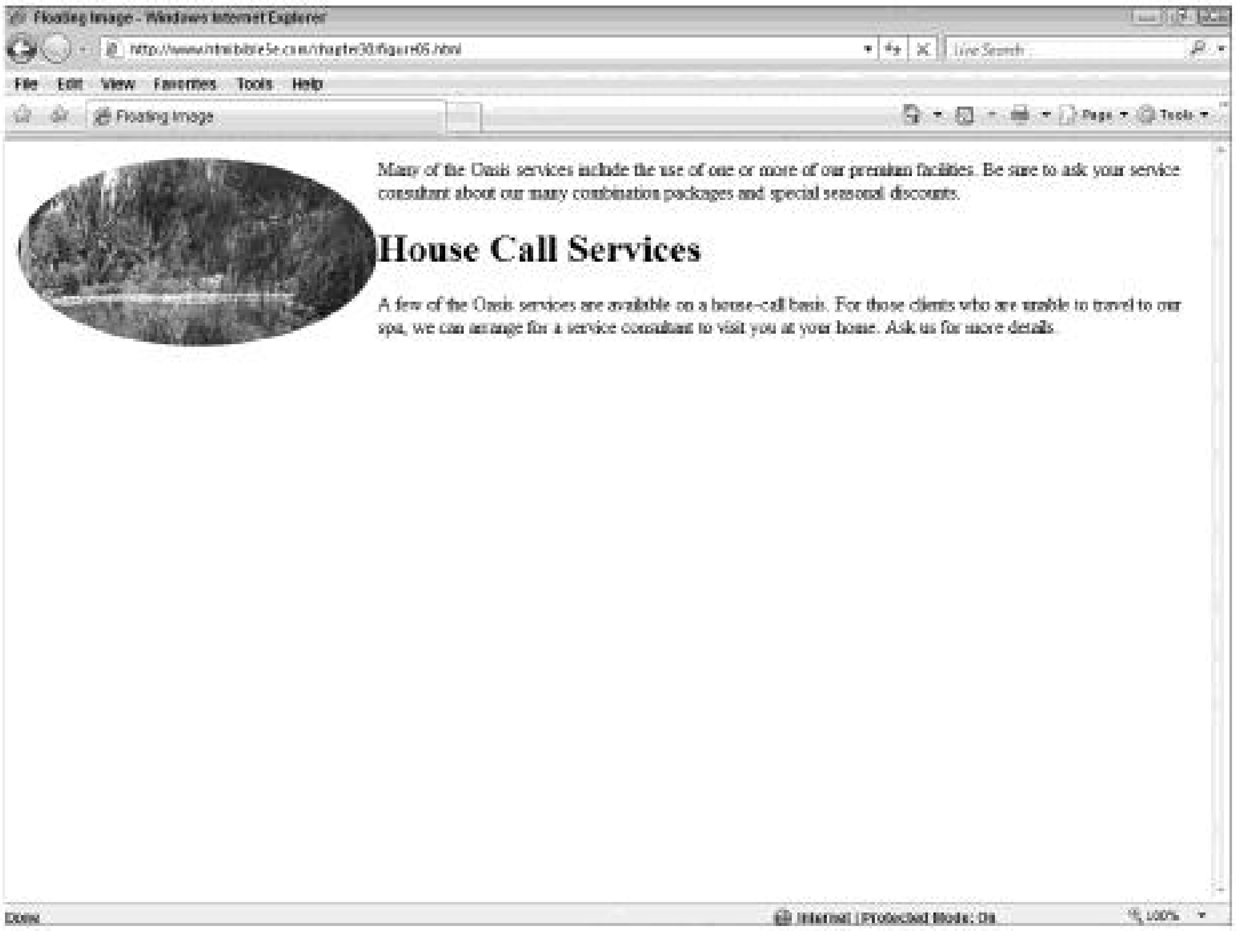
Floating Image
Many of the Oasis services include the use of one or more of our premium facilities. Be sure to ask your service consultant about our many combination packages and special seasonal discounts.

Non-Floating Image
Many of the Oasis services include the use of one or more of our premium facilities. Be sure to ask your service consultant about our many combination packages and special seasonal discounts.

Although floating images can add an attractive, dynamic air to your documents, their placement is not always predictable. As such, it's helpful to be able to indicate that specific elements should not allow floating elements next to them. One good example of when you would want to disallow floating elements is next to headings. Consider the document shown in Figure 30-5.

FIGURE 30-5

Floating images can sometimes get in the way of positioning other elements, such as headings.


Floating Image - Windows Internet Explorer
http://www.htmlbible5e.com/chapter30/figure06.html
Live Search
File Edit View Favorites Tools Help
Floating Image
Page Tools
Done
Internet | Protected Mode: On
100%

Many of the Oasis services include the use of one or more of our premium facilities. Be sure to ask your service consultant about our many combination packages and special seasonal discounts.

House Call Services

A few of the Oasis services are available on a house-call basis. For those clients who are unable to travel to our spa, we can arrange for a service consultant to visit you at your home. Ask us for more details.

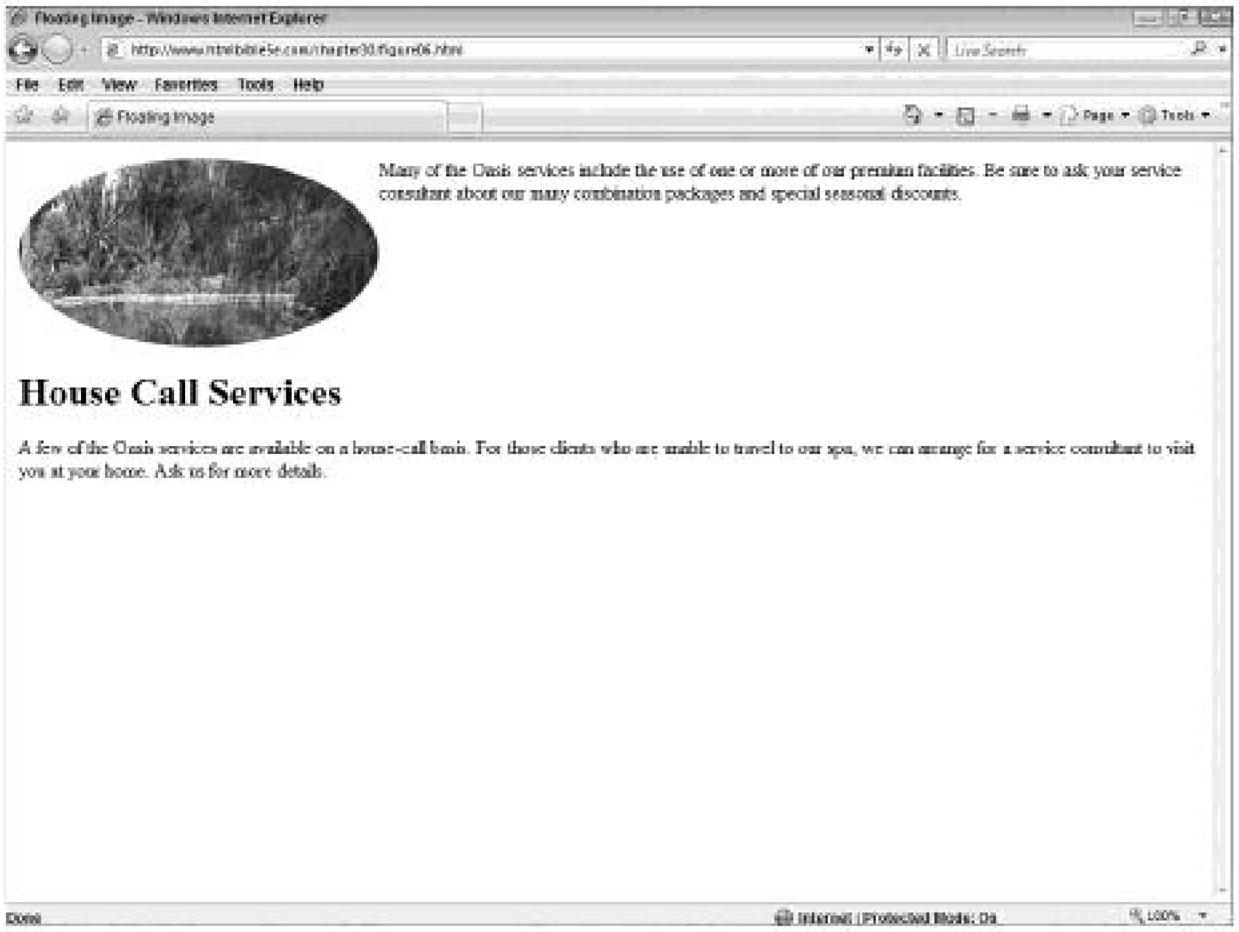
Using the `clear` property, you can ensure that one side or both sides of an element remain free of floating elements. You can specify `left`, `right`, `both`, or `none` (the default) for values of the `clear` property. Note that the `clear` property doesn't affect the floating element. Instead, it forces the element containing the `clear` property to avoid the floating element(s) by placing itself after the floating element(s).

For example, adding the following style to the document shown in Figure 30-5 ensures that both sides of all heading levels are clear of floating elements. This results in the display shown in Figure 30-6, with the heading being placed after the floating figure.

```
h1,h2,h3,h4,h5,h6 { clear: both;}
```

FIGURE 30-6

Use the clear property to force an element to start past the floating element's bounding box (and before any additional floating elements begin).


Floating Image - Windows Internet Explorer
http://www.htmlbible5e.com/chapter30/figure06.html
Live Search
File Edit View Favorites Tools Help
Floating Image
Page Tools
Done
Internet | Protected Mode: On
100%

Many of the Oasis services include the use of one or more of our premium facilities. Be sure to ask your service consultant about our many combination packages and special seasonal discounts.

House Call Services

A few of the Oasis services are available on a house-call basis. For those clients who are unable to travel to our spa, we can arrange for a service consultant to visit you at your home. Ask us for more details.

The white-space property

User agents typically ignore extraneous white space in documents. However, at times you want the white space to be interpreted literally, without having to use a `<pre>` tag to do so. Enter the `white-space` property.

The `white-space` property can be set to the following values:

- `normal`
- `pre`
- `nowrap`

The default setting is `normal` — that is, ignore extraneous white space.

If the property is set to `pre`, text will be rendered as though it were enclosed in a `<pre>` tag. Using `pre` does not affect the font or other formatting of the element; it just causes white space to be rendered verbatim. For example, the following text will be spaced exactly as shown in the following code:

```
<p style="white-space:  pre;">This         paragraph's   words
 are irregularly          spaced,       but will be rendered     as
     such
by      the               user              agent.</p>
```

Setting the `white-space` property to `nowrap` causes the element not to wrap at the right margin of the user agent. Instead, it continues to the right until the next explicit line break. User agents should add horizontal scroll bars to enable users to fully view the content.

Note
Text contained in a `pre` element is displayed using a monospace font. If you preserve white space by using the `white-space` property with a value of `pre`, your document will generally be rendered in a proportional font. ■

Controlling Letter and Word Spacing

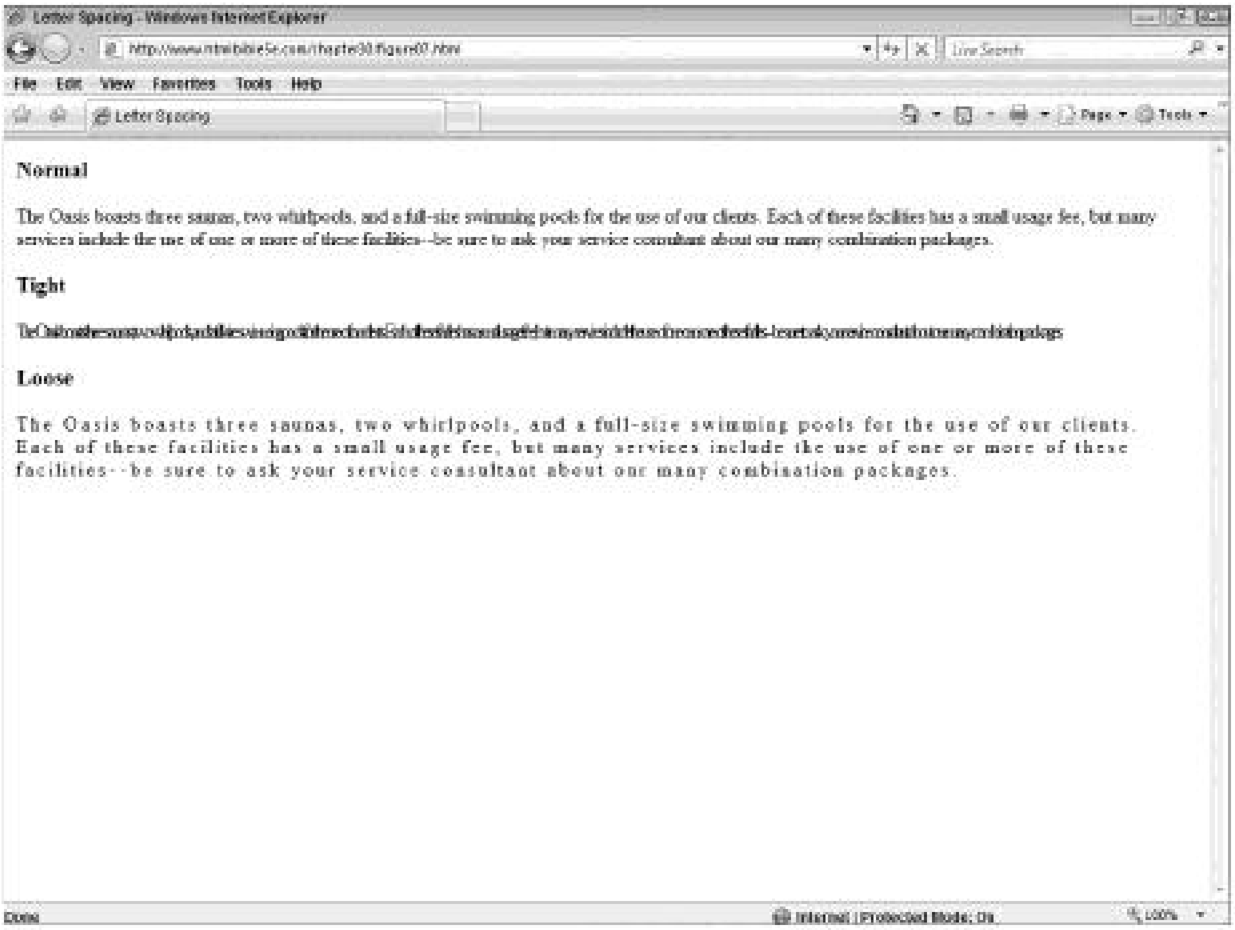
The `letter-spacing` and `word-spacing` properties can be used to control the letter and word spacing in an element, respectively. Both elements take an explicit or relative value to adjust the spacing — positive values add more space, negative values remove space. For example, consider the following code, whose output is shown in Figure 30-7:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Letter Spacing</title>
  <style type="text/css">
    .normal { letter-spacing: normal; }
    .tight  { letter-spacing: -.2em; }
    .loose  { letter-spacing: .2em; }
  </style>
</head>
<body>
  <h3>Normal</h3>
  <p class="normal"> The Oasis boasts three saunas, two whirlpools,
and a full-size swimming pool for the use of our clients. Each of
these facilities has a small usage fee, but many services include
the use of one or more of these facilities--be sure to ask your
service consultant about our many combination packages.</p>
  <h3>Tight</h3>
  <p class="tight"> The Oasis boasts three saunas, two whirlpools,
and a full-size swimming pool for the use of our clients. Each of
these facilities has a small usage fee, but many services include
```

```
the use of one or more of these facilities--be sure to ask your
service consultant about our many combination packages.</p>
   <h3>Loose</h3>
   <p class="loose"> The Oasis boasts three saunas, two whirlpools,
and a full-size swimming pool for the use of our clients. Each of
these facilities has a small usage fee, but many services include
the use of one or more of these facilities--be sure to ask your
service consultant about our many combination packages.</p>
 </body>
 </html>
```

FIGURE 30-7

The letter-spacing property does exactly what its name indicates; it adjusts the spacing between letters.



Letter Spacing - Windows Internet Explorer

http://www.htmlbible5e.com/chapter30/figure07.html

Normal

The Oasis boasts three saunas, two whirlpools, and a full-size swimming pools for the use of our clients. Each of these facilities has a small usage fee, but many services include the use of one or more of these facilities--be sure to ask your service consultant about our many combination packages.

Tight

[illegible]

Loose

The Oasis boasts three saunas, two whirlpools, and a full-size swimming pools for the use of our clients. Each of these facilities has a small usage fee, but many services include the use of one or more of these facilities--be sure to ask your service consultant about our many combination packages.

Note that the user agent can govern the minimum amount of letter spacing allowed. Setting the letter spacing to too small a value can have unpredictable results.

The `word-spacing` property behaves exactly like the `letter-spacing` property, except that it controls the spacing between words instead of letters. Like `letter-spacing`, using a positive value with `word-spacing` results in more space between words, and using a negative value results in less space.

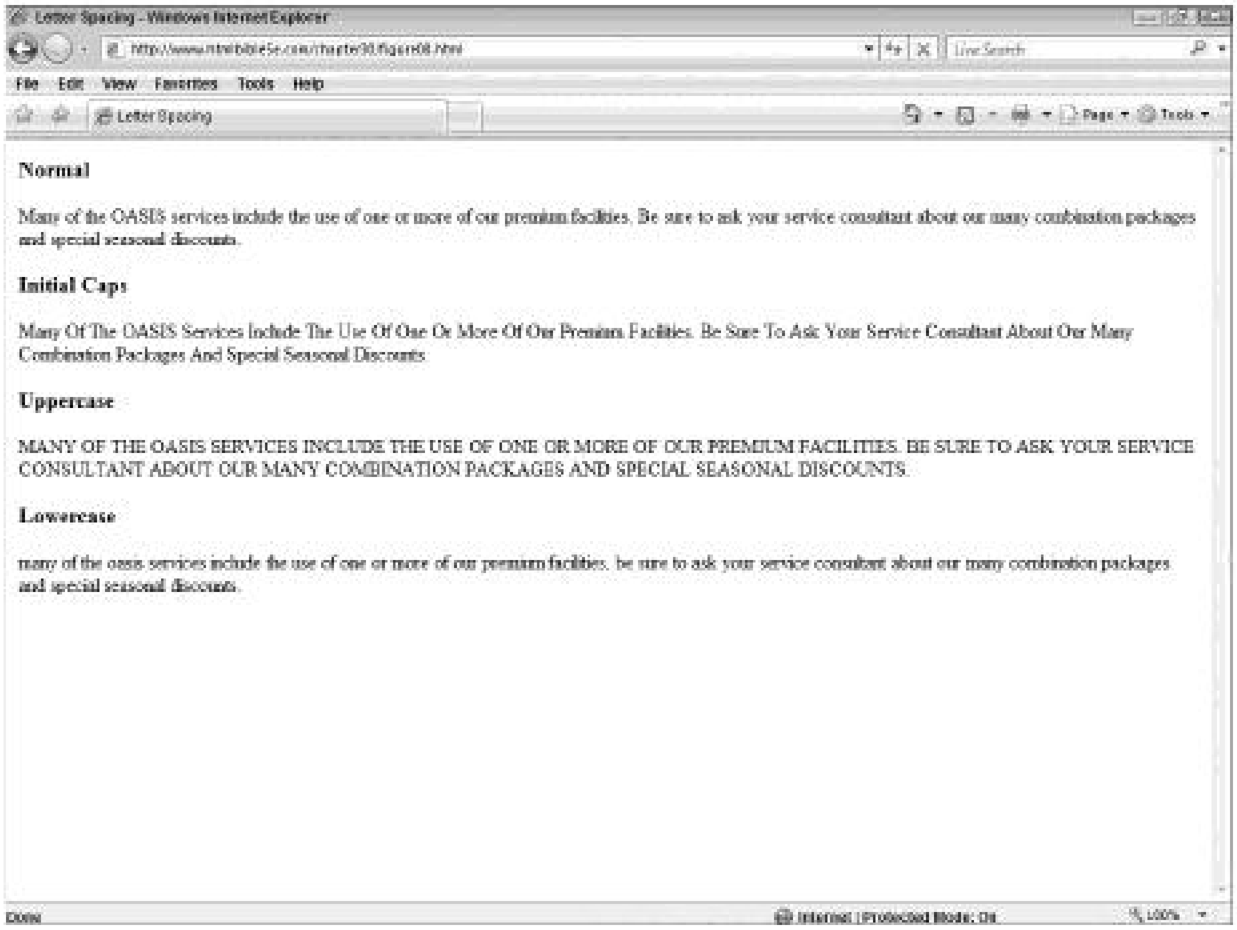
Specifying Capitalization

You can also use styles to control the capitalization, or case, of text. The `text-transform` property can be set to four different values, as shown in the following code and Figure 30-8:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Letter Spacing</title>
  <style type="text/css">
    .normal { text-transform: none;}
    .initcaps { text-transform: capitalize;}
    .upper { text-transform: uppercase;}
    .lower { text-transform: lowercase;}
  </style>
</head>
<body>
  <h3>Normal</h3>
  <p class="normal"> Many of the OASIS services include the use of
one or more of our premium facilities. Be sure to ask your service
consultant about our many combination packages and special seasonal
discounts.</p>
  <h3>Initial Caps</h3>
  <p class="initcaps"> Many of the OASIS services include the use of
one or more of our premium facilities. Be sure to ask your service
consultant about our many combination packages and special seasonal
discounts.</p>
  <h3>Uppercase</h3>
  <p class="upper"> Many of the OASIS services include the use of
one or more of our premium facilities. Be sure to ask your service
consultant about our many combination packages and special seasonal
discounts.</p>
  <h3>Lowercase</h3>
  <p class="lower"> Many of the OASIS services include the use of
one or more of our premium facilities. Be sure to ask your service
consultant about our many combination packages and special seasonal
discounts.</p>
</body>
</html>
```

FIGURE 30-8

The text-transform property enables you to influence the capitalization of elements.


Letter Spacing - Windows Internet Explorer
http://www.htmlbible5e.com/chapter30/figure08.html
Live Search
File Edit View Favorites Tools Help
Letter Spacing
Page Tools
Done
Internet | Protected Mode: On
100%

Normal

Many of the OASIS services include the use of one or more of our premium facilities. Be sure to ask your service consultant about our many combination packages and special seasonal discounts.

Initial Caps

Many Of The OASIS Services Include The Use Of One Or More Of Our Premium Facilities. Be Sure To Ask Your Service Consultant About Our Many Combination Packages And Special Seasonal Discounts.

Uppercase

MANY OF THE OASIS SERVICES INCLUDE THE USE OF ONE OR MORE OF OUR PREMIUM FACILITIES. BE SURE TO ASK YOUR SERVICE CONSULTANT ABOUT OUR MANY COMBINATION PACKAGES AND SPECIAL SEASONAL DISCOUNTS.

Lowercase

many of the oasis services include the use of one or more of our premium facilities. be sure to ask your service consultant about our many combination packages and special seasonal discounts.

There are some rules as to what `text-transform` will and won't affect. For example, the `capitalize` value ensures that each word starts with a capital letter, but it doesn't change the case of the rest of the word. Likewise, setting the property to `normal` will not change the case of the element (for example, "OASIS" remains in all caps).

Using Text Decorations

You can add several different effects to text through CSS. Most are accomplished via the `text-decoration` and `text-shadow` properties.

The `text-decoration` property enables you to add the following attributes to text:

- `underline`
- `overline` (line above text)
- `line-through`
- `blink`

As with most properties, the values are straightforward:

```
<p style="text-decoration: none;">No Decoration</p>
<p style="text-decoration: underline;">Underlined</p>
<p style="text-decoration: overline;">Overlined</p>
<p style="text-decoration: line-through;">Line Through</p>
<p style="text-decoration: blink;">Blink</p>
```

The `text-shadow` property is a bit more complex but can add stunning drop shadow effects to text. The `text-shadow` property has the following format:

```
text-shadow: "[color] horizontal-distance
vertical-distance [blur]"
```

The property takes two values to offset the shadow: one horizontal, the other vertical. Positive values set the shadow down and to the right. Negative values set the shadow up and to the left. Using combinations of negative and positive settings, you can move the shadow to any location relative to the text it affects.

The optional `color` value sets the color of the shadow. The blur value specifies the blur radius — or the width of the effect — for the shadow. The exact algorithm for computing the blur radius is not specified by the CSS specification, so your experience may vary with this value.

The `text-shadow` property enables multiple shadow definitions for multiple shadows. Simply separate the definitions with commas.

The following code creates a drop shadow on all `h1` headings. The shadow is set to display above and to the right of the text in a gray color:

```
h1 { text-shadow: #666666 2em -2em;}
```

The following definition provides the same shadow as the previous example but adds another, lighter gray shadow directly below the text:

```
h1 { text-shadow: #666666 2em -2em, #AAAAAA 0em 2em;}
```

Unfortunately, not many user agents support `text-shadow`. If you want such an effect, you might be better off creating it with a graphic instead of text.

Autogenerated Text

CSS has a few mechanisms for autogenerating text. Although this doesn't fit in well with the presentation-only function of CSS, it can be useful to have some constructs to automatically generate text for your documents. There are properties and elements to automatically supply quotation marks, provide arbitrary text before or after an element, or autogenerate a counter.

Cross-Ref
Although these properties bear mention here, they are covered in depth with the other pseudo-elements and generated content in Chapter 35. ■

Using CSS Table Properties

Because the `<table>` tag attributes, such as `border`, `rules`, `cellpadding`, and `cellspacing`, have not been deprecated, you might be tempted to use them instead of CSS properties when defining your tables. You should resist that temptation.

Using styles for tables provides the same advantages as using styles for any other element — consistency, flexibility, and the ability to easily change the format later.

For example, consider the following table tag:

```
<table border="1" width="200px" cellpadding="3px"
    cellspacing="5px">
```

Now suppose you had four tables using this same beginning tag in your document, and you had four other documents just like it. What if you decided to decrease the width of the table and increase the padding within the tables? You would have to edit each table manually, potentially 16 individual tables between four documents.

If the table formatting were contained in styles at the top of the documents, you would have to make only four changes, one change in each document. Better yet, if the formatting were contained in a separate, external style sheet, you would have to make only one change.

Note
The CSS `border` properties are used to control table borders, and the `padding` and `margin` properties are used to affect the spacing of cells and the padding of their contents. ■

Controlling Table Attributes

You can use CSS properties to control the formatting of tables, but note that some of the property names do not match up with the tag attributes. For example, there are no `cellspacing` or `cellpadding` CSS properties. The `border-spacing` and `padding` CSS properties fill those roles, respectively.

Table 30-1 shows how CSS properties match table tag attributes.

Each of the various properties is covered in the following sections.

TABLE 30-1

CSS Properties for Table Attributes

Purpose	Table Attribute	CSS Property(ies)
Borders	`border`	`border` properties
Spacing inside cell	`cellpadding`	`padding` properties
Spacing between cells	`cellspacing`	`border-spacing` property
Width of table	`width`	`width` and `table-layout` properties
Table framing	`frame`	`border` properties
Alignment	`align`, `valign`	`text-align`, `vertical-align` properties

Table borders

You can use the `border` properties to control the border of a table and its sub-elements, just like any other element. For example, the following definition causes all tables and their elements to have single, solid, 1-point borders around them (as shown in Figure 30-9):

```
table, table * { border: 1pt solid black;}
```

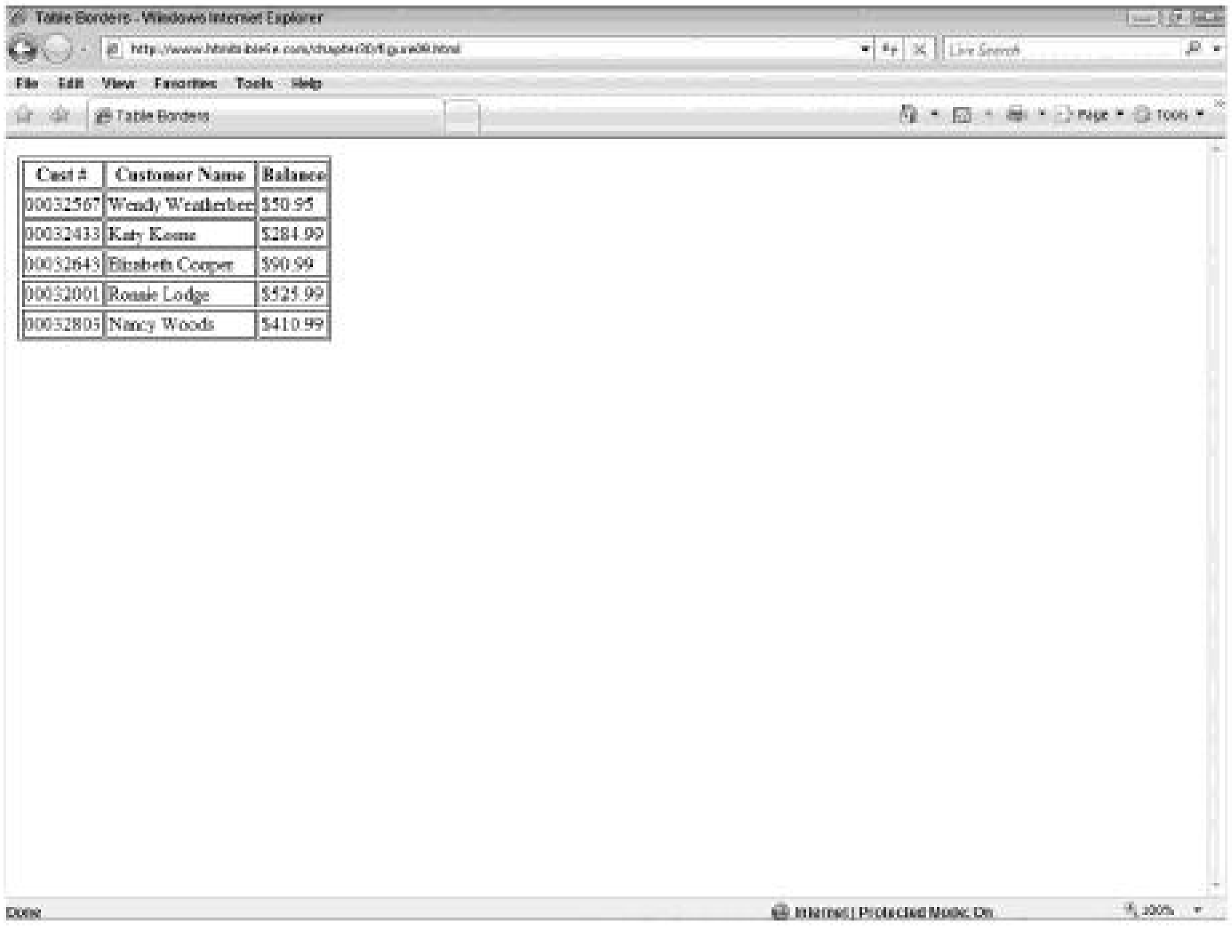
The preceding example specifies all tables and all table descendants (`table, table *`) to ensure that each cell, as well as the entire table, has a border. If you wanted only the cells or only the table to have borders, you could use the following definitions:

```
/* Only table cells have borders */
table * { border: 1pt solid black;}
 or
/* Only table body has borders */
table { border: 1pt solid black;}
```

The results of these two definitions are shown in Figure 30-10.

FIGURE 30-9

A table using CSS properties to define its borders



Cust #	Customer Name	Balance
00032567	Wendy Weatherbee	$50.95
00032433	Katy Keene	$284.99
00032643	Elizabeth Cooper	$90.99
00032001	Ronnie Lodge	$525.99
00032805	Nancy Woods	$410.99

You can also combine border styles. For example, the following definitions create a table with borders similar to using the `table` element's `border` attribute. The result of this definition is shown in Figure 30-11.

```
table { border: outset 5pt;}
td, th { border: inset 5pt;}
```

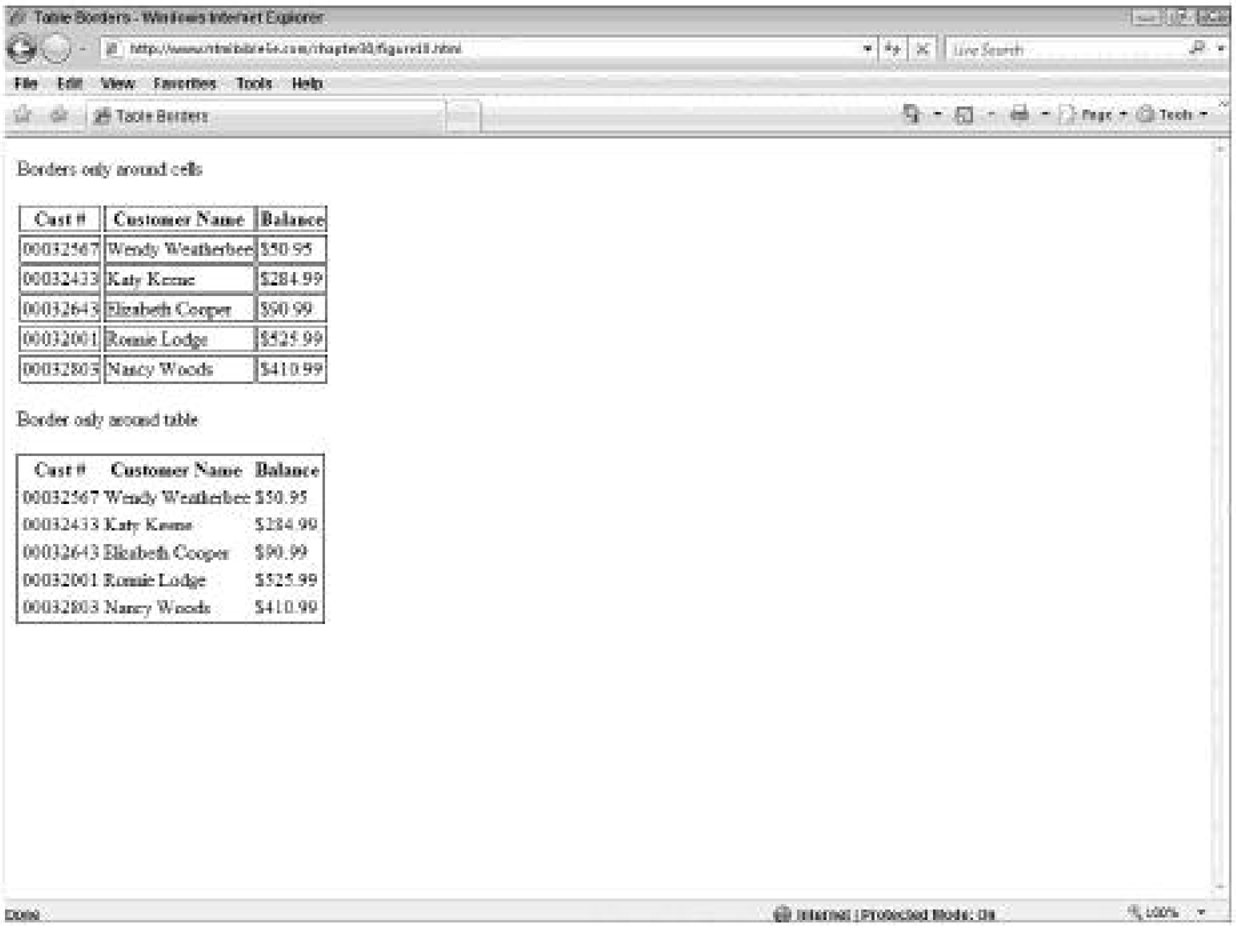
Table border spacing

To increase the space around table borders, use the `border-spacing` and `padding` CSS properties. The `border-spacing` property adjusts the space between table cells much like the

`<table>` tag's `cellspacing` attribute. The `padding` property adjusts the space between a table cell's contents and the cell's border.

FIGURE 30-10

Tables using selective bordering


Borders only around cells

Cust #	Customer Name	Balance
00032567	Wendy Weatherbee	$50.95
00032433	Katy Keene	$284.99
00032643	Elizabeth Cooper	$90.99
00032001	Ronnie Lodge	$525.99
00032803	Nancy Woods	$410.99

Border only around table

Cust #	Customer Name	Balance
00032567	Wendy Weatherbee	$50.95
00032433	Katy Keene	$284.99
00032643	Elizabeth Cooper	$90.99
00032001	Ronnie Lodge	$525.99
00032803	Nancy Woods	$410.99

The `border-spacing` property has the following format:

```
border-spacing: horizontal_spacing  vertical_spacing;
```

Note that you can choose to include only one value, in which case the spacing is applied to both the horizontal and vertical border spacing.

For example, Figure 30-12 shows the same table shown in Figure 30-11, but with the following `border-spacing` definition:

```
Table { border-spacing: 5px 15px;}
```

Note

The `border-spacing` property works only with tables that have their `border-collapse` property set to `separate`. Also, some user agents, such as Internet Explorer, disregard the `border-spacing` property. ■

FIGURE 30-11

You can combine border styles to create custom table formats.

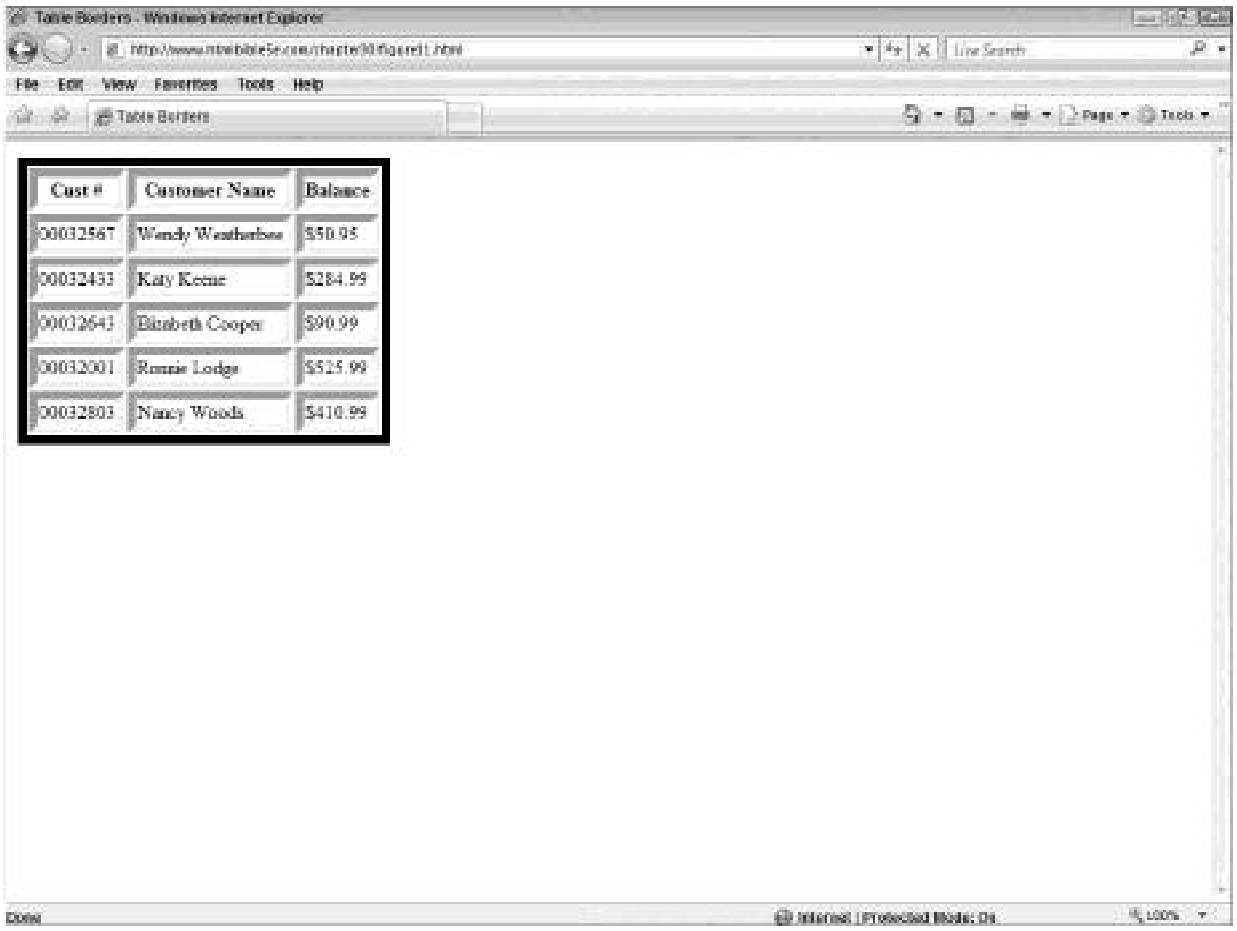


Table Borders - Windows Internet Explorer
File Edit View Favorites Tools Help
Table Borders
Done

Cust #	Customer Name	Balance
00032567	Wendy Weatherbee	$50.95
00032433	Katy Keene	$284.99
00032643	Elizabeth Cooper	$90.99
00032001	Ronnie Lodge	$525.99
00032803	Nancy Woods	$410.99

Collapsing borders

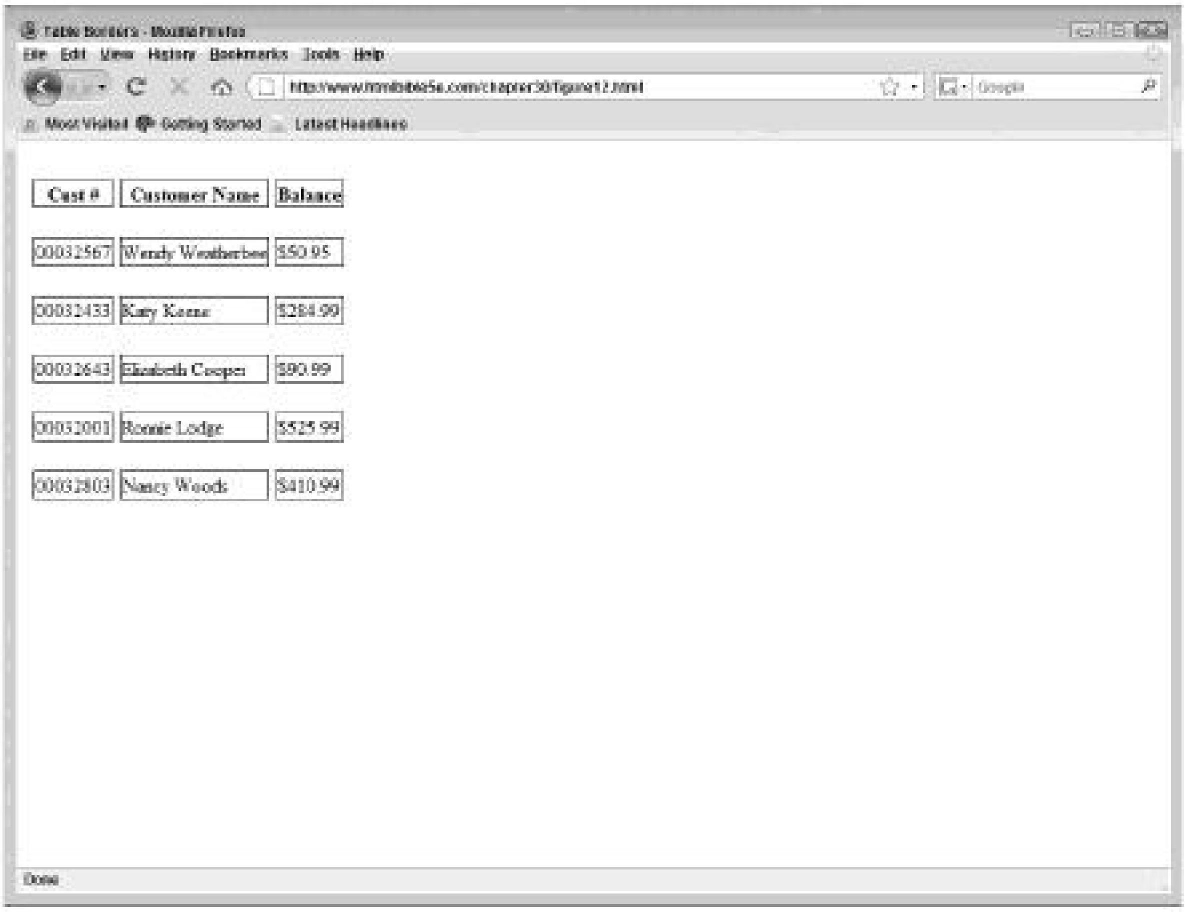
Sometimes you will want to remove the spacing between borders in a table, creating gridlines instead of distinct individual borders. To do so, use the `border-collapse` property. This property takes either the value of `separate` (default) or `collapse`. If you specify `collapse`, the cells merge their borders with neighboring cells (or the table) into one line. Whichever cell has the most visually distinctive border determines the collapsed border's look.

For example, consider the two tables in Figure 30-13, shown with their table definitions directly above them.

Notice how the borders between the table headers (`th`) and normal cells inherited the inset border, while the rest of the borders remained solid. This is because the border around the table headers was more visually distinctive and won the conflict between the borders styles being collapsed.

FIGURE 30-12

Different horizontal and vertical border-spacing can help distinguish data in columns or rows.



Cust #	Customer Name	Balance
00032567	Wendy Weatherbee	$50.95
00032433	Katy Keene	$284.99
00032643	Elizabeth Cooper	$90.99
00032001	Ronnie Lodge	$525.99
00032803	Nancy Woods	$410.99

Borders on empty cells

Typically, the user agent does not render empty cells, but you can use the `empty-cells` CSS property to control whether the agent should or should not show empty cells. The `empty-cells` property takes one of two values: `show` or `hide` (default).

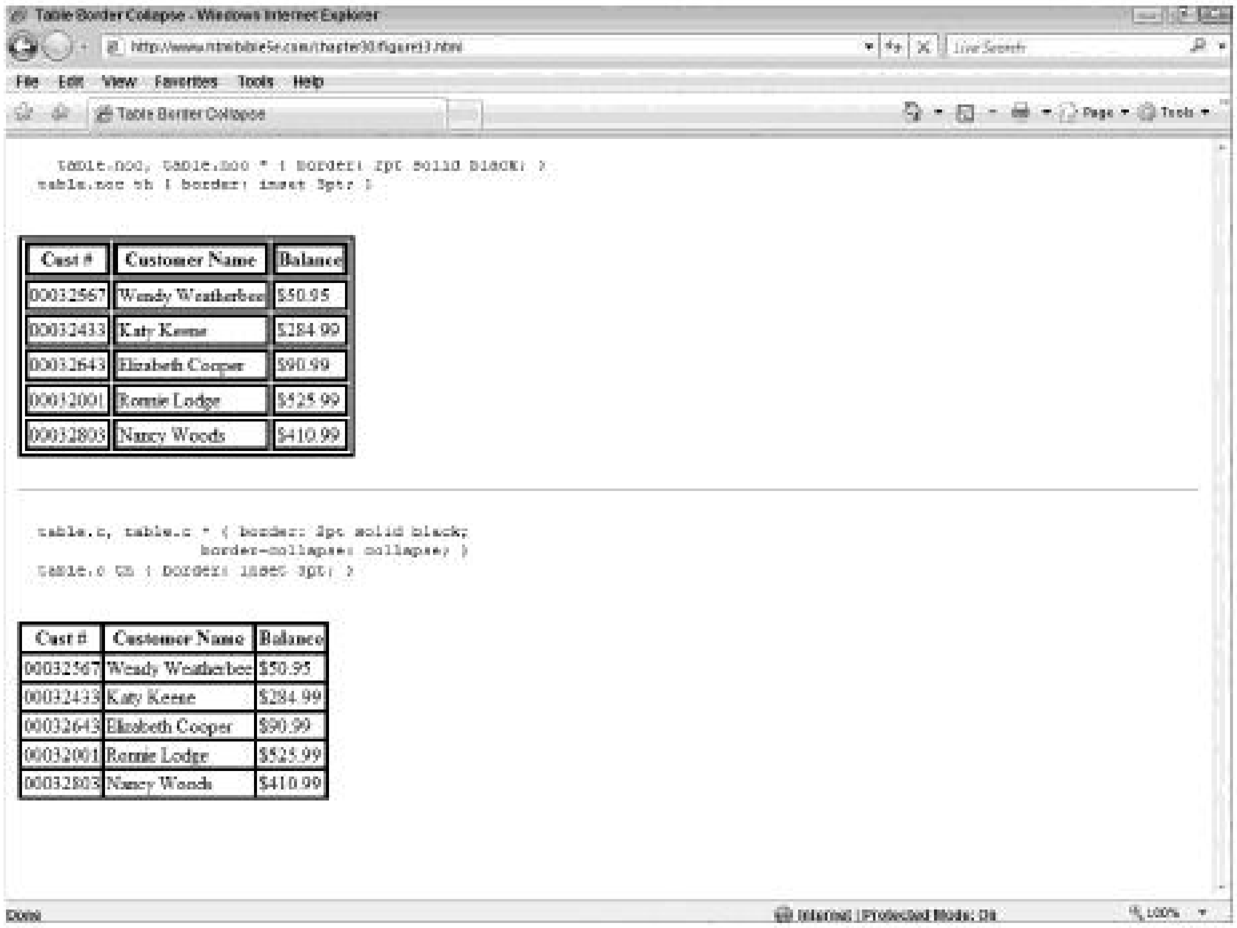
Figure 30-14 shows the following table with various settings of the `empty-cells` property:

```
<table>
  <tr><th>Heading</th><th>Heading</th><th>Heading</th></tr>
```

```
    <tr><td>X</td><td></td><td>X</td></tr>
    <tr><td></td><td>X</td><td></td></tr>
    <tr><td>X</td><td>X</td><td>X</td></tr>
  </table>
```

FIGURE 30-13

Collapsing table borders turns individual borders into gridlines between cells.



```
  table.noc, table.noc * { border: 2pt solid black; }
 table.noc th { border: inset 3pt; }
```

Cust #	Customer Name	Balance
00032567	Wendy Weatherbee	$50.95
00032433	Katy Keene	$284.99
00032643	Elizabeth Cooper	$90.99
00032001	Ronnie Lodge	$525.99
00032803	Nancy Woods	$410.99

```
 table.c, table.c * { border: 2pt solid black;
                      border-collapse: collapse; }
 table.c th { border: inset 3pt; }
```

Cust #	Customer Name	Balance
00032567	Wendy Weatherbee	$50.95
00032433	Katy Keene	$284.99
00032643	Elizabeth Cooper	$90.99
00032001	Ronnie Lodge	$525.99
00032803	Nancy Woods	$410.99

Note

Some user agents, such as Internet Explorer, disregard the `empty-cells` property. In such cases, the only recourse is to place a nonbreaking space (`&nbsp;`) or other non-printable character in each empty cell, making the cell not empty but containing no visible contents. ■

FIGURE 30-14

The empty-cells property controls whether the user agent displays empty cells or not.

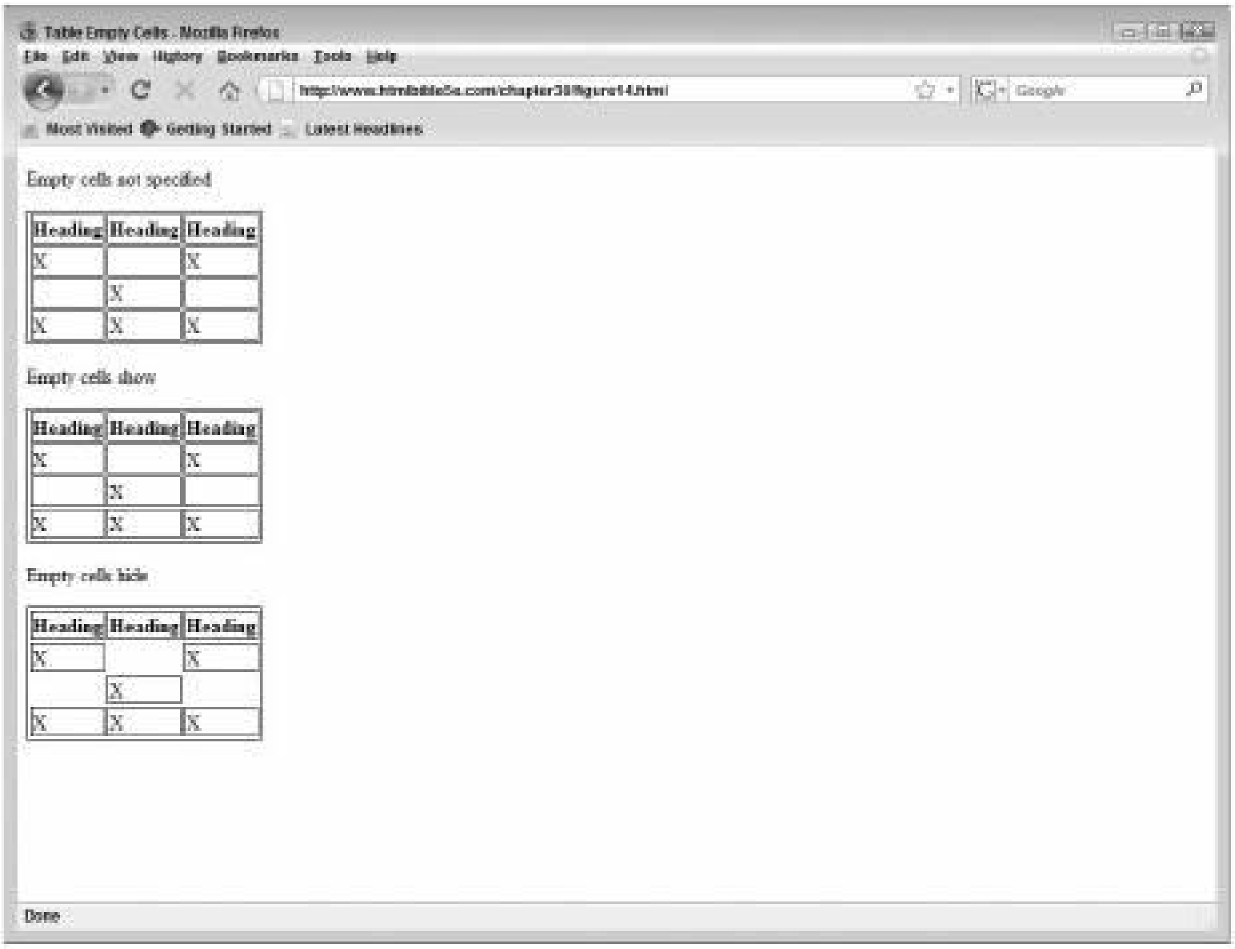


Table Empty Cells - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.htmlbible5e.com/chapter30/figure14.html
Google
Most Visited Getting Started Latest Headlines
Done

Empty cells not specified

Heading	Heading	Heading
X		X
	X	
X	X	X

Empty cells show

Heading	Heading	Heading
X		X
	X	
X	X	X

Empty cells hide

Heading	Heading	Heading
X		X
	X	
X	X	X

Table Layout

The `table-layout` property determines how a user agent sizes a table. This property takes one of two values: `auto` or `fixed`. If this property is set to `auto`, the user agent automatically determines the table's width primarily from the contents of the table's cells. If this property is set to `fixed`, the user agent determines the table's width primarily from the width values defined in the other properties and attributes affecting the table.

Aligning and Positioning Captions

CSS can also help control the positioning of table caption elements. The positioning of the caption is controlled by the `caption-side` property. This property has the following format:

```
caption-side: top | bottom | left | right;
```

The property's value determines where the caption is positioned in relationship to the table. To align the caption in its position, you can use typical text alignment properties such as `text-align` and `vertical-align`.

For example, the following code places the table's caption to the right of the table, centered vertically and horizontally, as shown in Figure 30-15:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Table Caption Positioning</title>
  <style type="text/css">
    table { margin-right: 200px;}
    table, table * { border: 1pt solid black;
                     caption-side: right;}
    caption { margin-left: 10px;
              vertical-align: middle;
              text-align: center;}
  </style>
</head>
<body>
  <table>
  <tr>
    <th>Cust #</th>
    <th>Customer Name</th>
    <th>Balance</th>
  </tr>
  <tr>
    <td>00032567</td>
    <td>Wendy Weatherbee</td>
    <td>$50.95</td>
  </tr>
  <tr>
    <td>00032433</td>
    <td>Katy Keene</td>
    <td>$284.99</td>
  </tr>
  <tr>
    <td>00032643</td>
    <td>Elizabeth Cooper</td>
    <td>$90.99</td>
    </tr>
    <tr>
      <td>00032001</td>
      <td>Ronnie Lodge</td>
      <td>$525.99</td>
    </tr>
    <tr>
      <td>00032803</td>
      <td>Nancy Woods</td>
      <td>$410.99</td>
    </tr>
    <caption>Daily Balance for<br/>07/20/07</caption>
    </table>
  </body>
  </html>
```

FIGURE 30-15

Using CSS you can position the caption of a table, in this case to the bottom of the table.

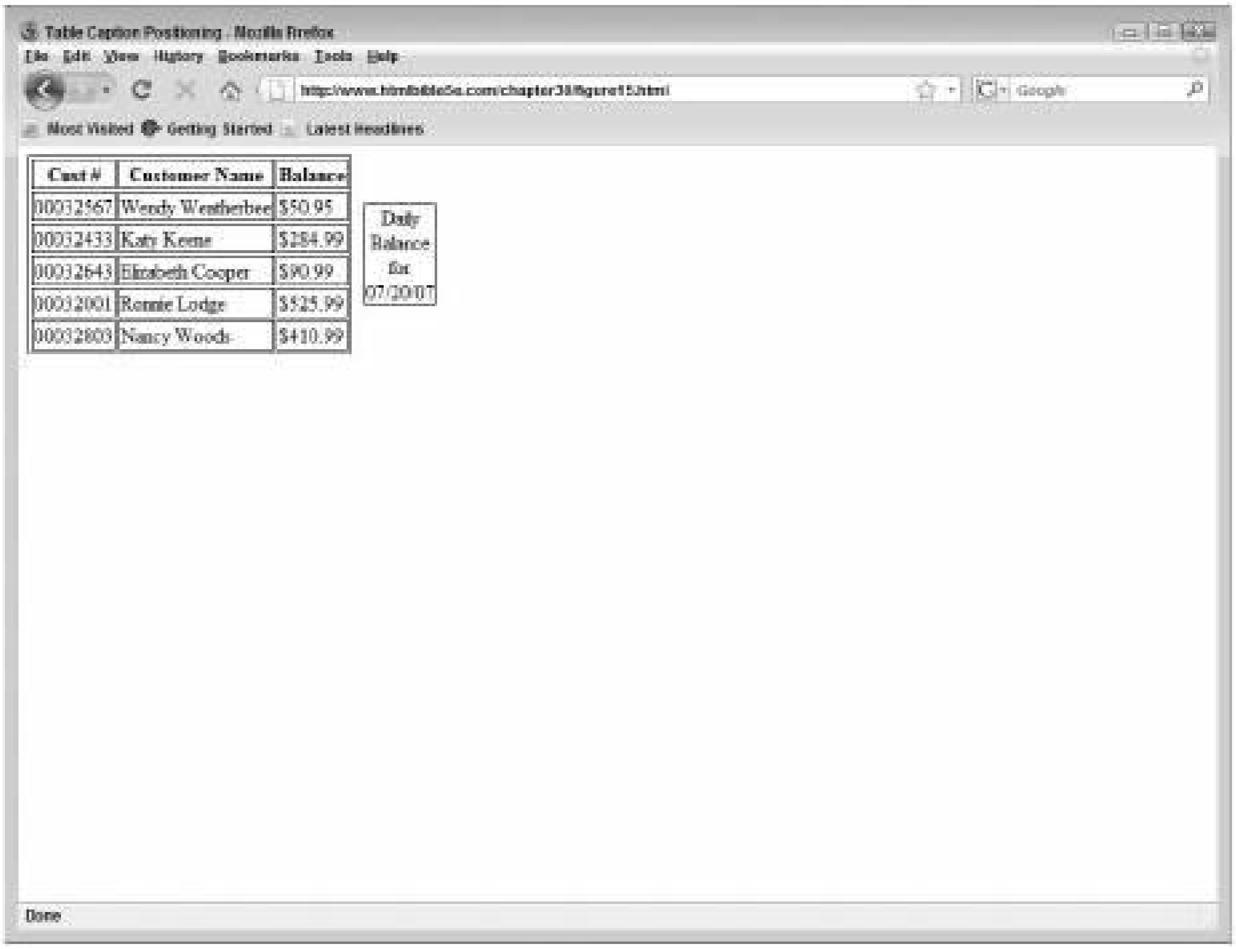


Table Caption Positioning - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.htmlbible5e.com/chapter30/figure15.html
Google
Most Visited Getting Started Latest Headlines
Done

Cust #	Customer Name	Balance
00032567	Wendy Weatherbee	$50.95
00032433	Katy Keene	$284.99
00032643	Elizabeth Cooper	$90.99
00032001	Ronnie Lodge	$525.99
00032803	Nancy Woods	$410.99

Daily Balance for 07/20/07

Note
This property does not currently work in Internet Explorer. ■

Note that the table's caption is positioned inside the table's margin. By increasing the table's margin, you allow more text per line of the caption. You can also explicitly set the width of the caption using the `width` property, which increases the table's margins accordingly.

Summary

This chapter covered CSS properties used to format tables and text. You learned how to affect basic text formatting such as aligning and indenting, as well as how to control spacing of various textual entities. You also learned how to use CSS to effectively format HTML tables, bringing even more versatility to an already powerful formatting tool.

The rest of the chapters in this part of the book continue to cover CSS in a topical manner: lists (Chapter 31), box elements (Chapter 32), and so on. The latter sections of this part cover topics such as page layout (Chapter 34) and using CSS to define a document for printing (Chapter 37).

CHAPTER

31

CSS Lists

IN THIS CHAPTER

An Overview of Lists

CSS Lists — Any Element Will Do

List Style Type

Positioning of Markers

Using Images as List Markers

Lists are one of the most versatile textual constructs in HTML. Many HTML authors rely on them to render text in a variety of ways — not just text in list form. Several CSS properties modify lists and you can take full advantage of those properties. You can change the list type or the position of the elements, and specify images to use instead of bullets. This chapter covers the CSS list-related properties.

Tip

HTML lists have been pressed into service for a variety of formatting and use functions online. The Max Design site (`http://css.maxdesign.com.au/index.htm`) contains many examples of lists serving other purposes. ■

An Overview of Lists

There are two types of lists in standard HTML: ordered and unordered. Ordered lists have each of their elements numbered and are generally used for steps that must follow a specific order. Unordered lists are typically a list of related items that do not need to be in a particular order (commonly formatted as bulleted lists).

Cross-Ref

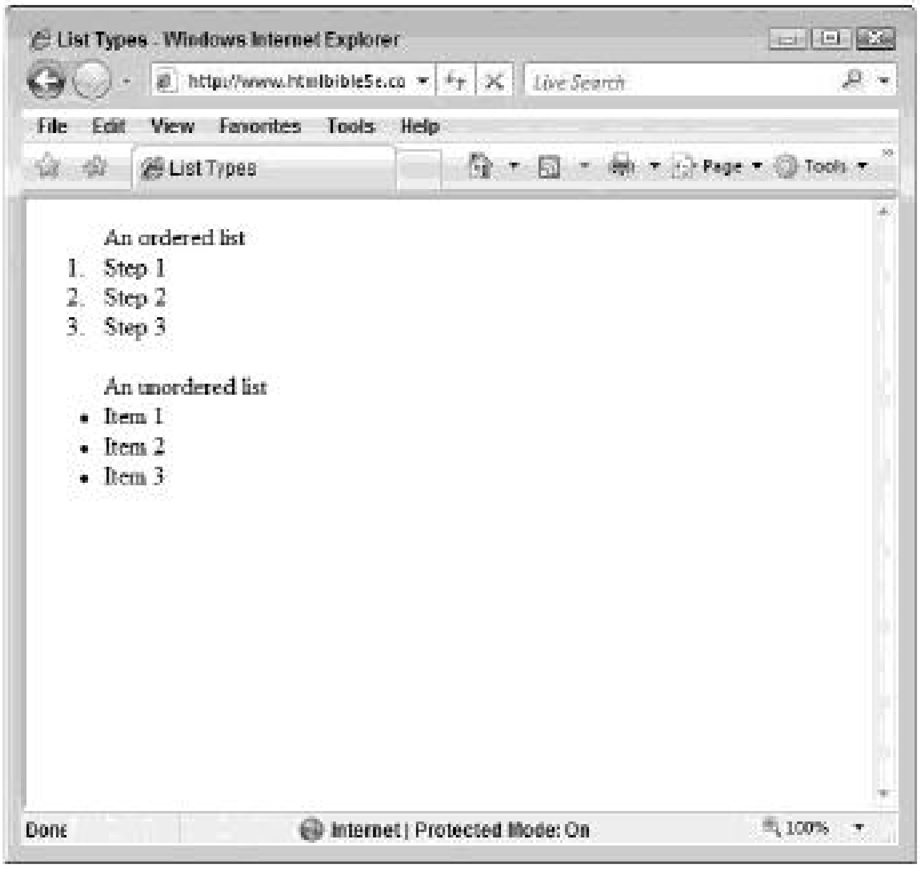
HTML formatting of lists is covered in Chapter 7. ■

Ordered lists are enclosed in the ordered list tag, `<ol>`. Unordered lists are enclosed in the unordered list tag, `<ul>`. A list item tag (`<li>`) encapsulates each item in either list. The following code shows short examples of each type of list, and Figure 31-1 shows the output of this code:

```
<ol>An ordered list
  <li>Step 1</li>
  <li>Step 2</li>
  <li>Step 3</li>
</ol>
<ul>An unordered list
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

FIGURE 31-1

The two types of lists, ordered (numbered) and unordered (bulleted)


List Types - Windows Internet Explorer
http://www.htmlbible5e.co
Live Search
File Edit View Favorites Tools Help
List Types
Page Tools
Done
Internet | Protected Mode: On
100%

An ordered list
1. Step 1
2. Step 2
3. Step 3

An unordered list
- Item 1
- Item 2
- Item 3

CSS Lists — Any Element Will Do

An important distinction of CSS lists is that you don't need to use the standard list element tag (`<li>`) for list items. CSS uses the `list-item` value of the `display` property, which, in effect, makes any inline element a list item. The `li` element is a list item by default.

Tip

There is a list style shortcut property that you can use to set list properties with a single property assignment. Use the `list-style` property to define the other list properties — style type, style position, and style image — as follows:

```
list-style: <list-style-type> <list-style-position>
<list-style-image> ■
```

For example, to create a new class that you can use to create a list item from almost any element, you can use the following definition:

```
.item { display: list-item;}
```

Thereafter, you can use that class to define elements as list items:

```
<p class="item">This is now a list item</p>
```

As you read through the rest of this section, keep in mind that list properties can apply to any element defined as a `list-item`.

Note

Any elements preceding list items, such as bullets or numbers, are known as *markers*. ■

List Style Type

The `list-style-type` property sets the type of the list and, therefore, what marker is used by default with each item within the list — bullet, number, Roman numeral, and so on.

The `list-style-type` property can have the following values:

- `disc`
- `circle`
- `square`
- `decimal`
- `decimal-leading-zero`
- `lower-roman`
- `upper-roman`
- `lower-greek`
- `lower-alpha`
- `lower-latin`
- `upper-alpha`
- `upper-latin`

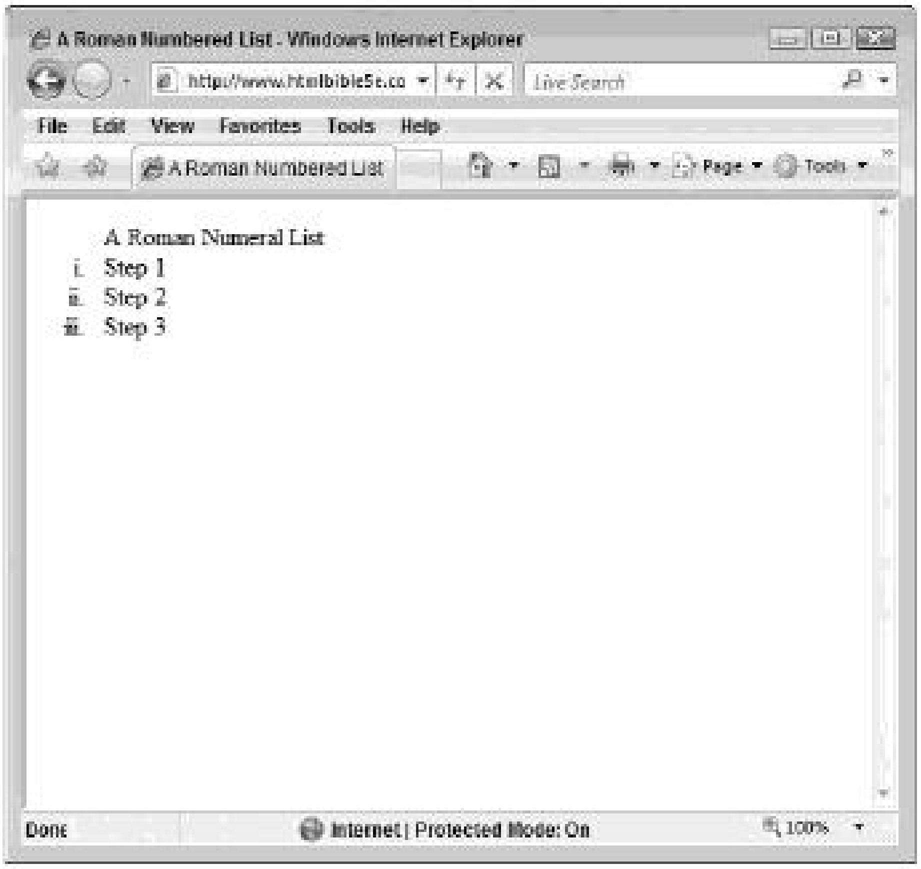
- `hebrew`
- `armenian`
- `georgian`
- `cjk-ideographic`
- `hiragana`
- `katakana`
- `hiragana-iroha`
- `katakana-iroha`
- `none`

The values are all fairly mnemonic to the markers they generate; setting the style provides a list with the appropriate item marker. For example, consider this code and the output shown in Figure 31-2:

```
<ol style="list-style-type:lower-roman;">
  A Roman Numeral List
  <li>Step 1</li>
  <li>Step 2</li>
  <li>Step 3</li>
</ol>
```

FIGURE 31-2

Roman numeral list markers


A Roman Numbered List - Windows Internet Explorer
http://www.htmlbible5e.co
Live Search
File Edit View Favorites Tools Help
A Roman Numbered List
Page Tools
Done
Internet | Protected Mode: On
100%

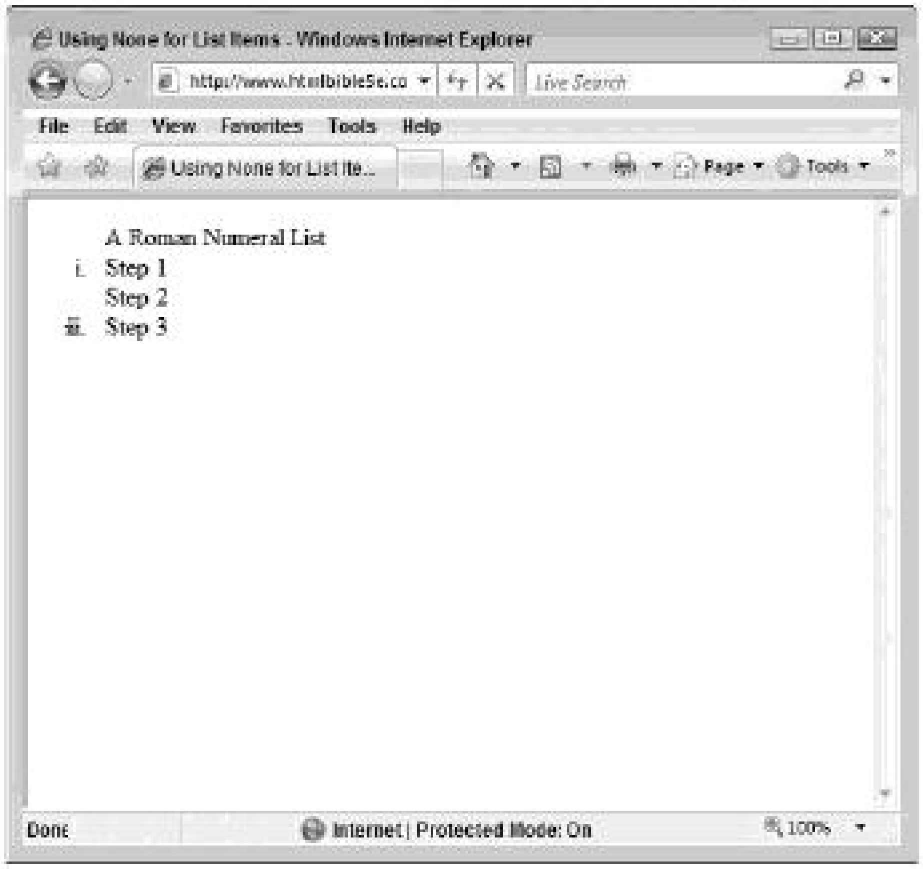
A Roman Numeral List
i. Step 1
ii. Step 2
iii. Step 3

You can use the none value to suppress markers for individual items. However, this does not change the sequential generation of markers; the markers for that item in the sequence are just not displayed. For example, consider the following revised code and the output shown in Figure 31-3:

```
<ol style="list-style-type:lower-roman;">
  A Roman Numeral List
  <li>Step 1</li>
  <li style="list-style-type:none;">Step 2</li>
  <li>Step 3</li>
</ol>
```

FIGURE 31-3

Using the none value for list markers


Using None for List Items - Windows Internet Explorer
http://www.htmlbible5e.co
Live Search
File Edit View Favorites Tools Help
Using None for List Ite...
Page Tools
Done
Internet | Protected Mode: On
100%

A Roman Numeral List
i. Step 1
Step 2
iii. Step 3

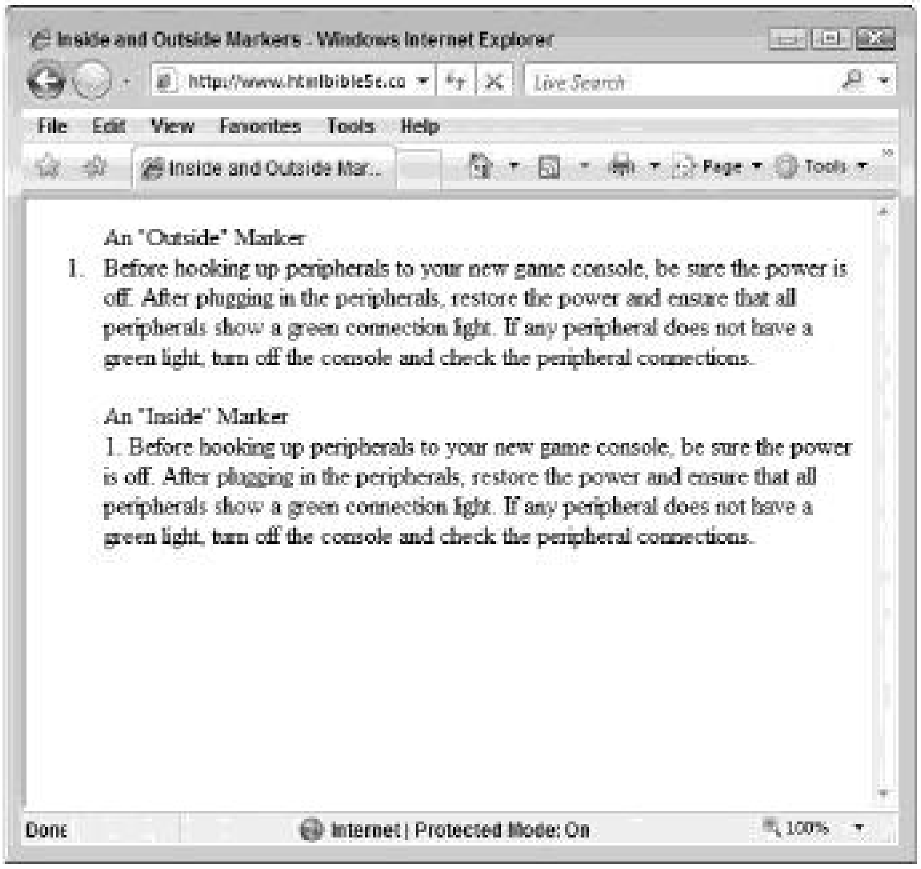
Note that the third item still has a marker of iii, despite suppressing the marker on the second item. Changing marker types in the middle of the list has a similar effect — the marker type may change, but its output will still reflect its proper sequence within the list.

Positioning of Markers

You can use the list-style-position property to change the position of the marker in relation to the list item(s). The valid values for this property are inside or outside. The outside value provides the more typical list style, whereby the marker is offset from the list item and the item's text is indented. The `inside` value sets the list to a more compact style, whereby the marker is indented with the first line of the item. Figure 31-4 shows an example of both positioning types.

FIGURE 31-4

Markers can be positioned either outside or inside the list element's margin.


Inside and Outside Markers - Windows Internet Explorer
http://www.htmlbible5e.co
Live Search
File Edit View Favorites Tools Help
Inside and Outside Mar...
Page Tools
Done
Internet | Protected Mode: On
100%

An "Outside" Marker

1. Before hooking up peripherals to your new game console, be sure the power is off. After plugging in the peripherals, restore the power and ensure that all peripherals show a green connection light. If any peripheral does not have a green light, turn off the console and check the peripheral connections.

An "Inside" Marker

1. Before hooking up peripherals to your new game console, be sure the power is off. After plugging in the peripherals, restore the power and ensure that all peripherals show a green connection light. If any peripheral does not have a green light, turn off the console and check the peripheral connections.

Of course, additional positioning and formatting styles can further change the position of the list item, but the position of the marker relative to the paragraph will be governed by the `list-style-position` property setting.

Using Images as List Markers

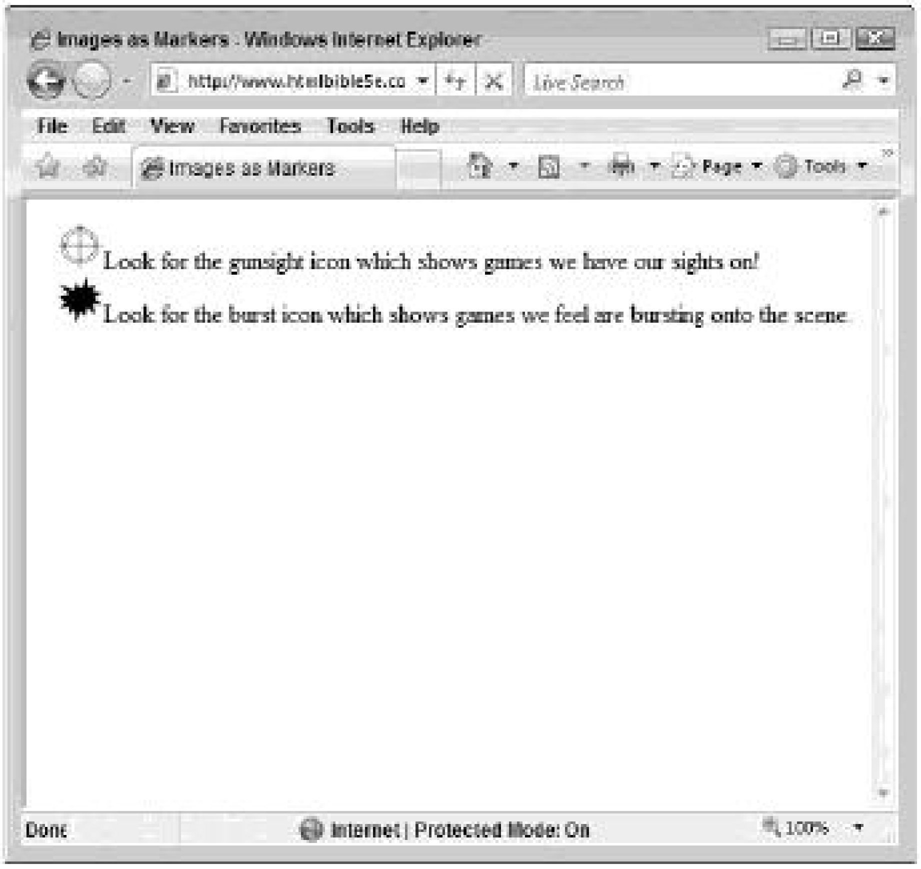
Using the `list-style-image` property, you can also specify an image to use as a marker. When this property's value is set, an image is used instead of the marker specified by the `list-style-type` property, even if the `list-style-property` value is set subsequently to the `list-style-image` value. The image to be used is specified via the `url` function. For example, the following code references sight_bullet.jpg and burst_bullet.jpg as images to use in the list, where both images reside in the images directory on the server:

```
<ol>
  <li style="list-style-image: url('images/sight_bullet.jpg')">
  Look for the gunsight icon which shows games we have our sights
  on!</li>
  <li style="list-style-image: url('images/burst_bullet.jpg')">
  Look for the burst icon which shows games we feel are bursting
  onto the scene.</li>
</ol>
```

The output is shown in Figure 31-5.

FIGURE 31-5

You can use images as list markers, such as the sight and burst shown here.


Images as Markers - Windows Internet Explorer
http://www.htmlbible5e.co
Live Search
File Edit View Favorites Tools Help
Images as Markers
Page Tools
Done
Internet | Protected Mode: On
100%

Look for the gunsight icon which shows games we have our sights on!

Look for the burst icon which shows games we feel are bursting onto the scene.

Note that you can use any URL-accessible image with the list-style-image. However, it is important to use images sized appropriately for your list.

Summary

This chapter demonstrated how you can use CSS to format HTML lists — formatting list elements and choosing or providing markers. Using the information in this chapter, you should be able to format a list to suit your particular needs. In the next few chapters, you'll learn how to use CSS to manipulate an element's box model, use colors for various design purposes, and use some of the more flexible, but esoteric CSS elements.

CHAPTER

32

Padding, Margins, and Borders

IN THIS CHAPTER

The CSS Box Formatting Model

Element Padding

Element Borders

Element Margins

Dynamic Outlines

All elements in an HTML document can be formatted in a variety of ways using CSS. Previous chapters in this part of the book covered the CSS basics — how to write a style definition and how to apply it to various elements within your documents. This chapter begins coverage of the concentric areas that surround elements — also known as the *box model*.

The next chapter continues this discussion, covering colors and background images.

The CSS Box Formatting Model

Although not overtly obvious, all elements in an HTML document are contained within a box. That box has several properties — margins, padding, and borders — that can be configured to help distinguish the enclosed element from nearby elements.

Take a look at Figure 32-1, which shows a document that isn't overtly boxy.

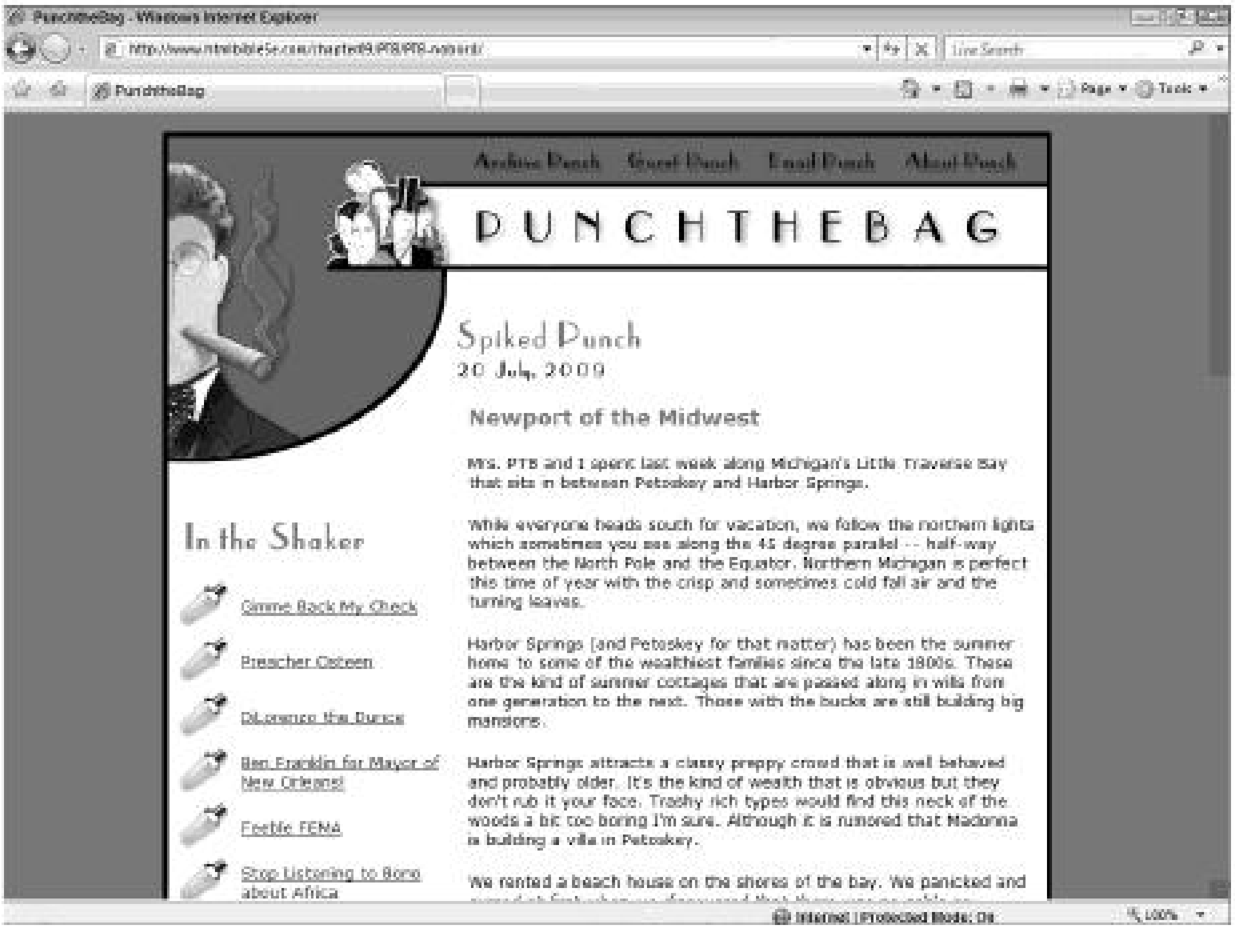
The same document is shown in Figure 32-2, but a thin border has been added to every element courtesy of the following style:

```
    * { border: thin solid black; }
```

Note how all the elements in the document pick up the border in a rectangular box shape. The border becomes much thicker at the intersection of two or more elements.

FIGURE 32-1

This Web document does not appear overly boxy in appearance.



PunchtheBag - Windows Internet Explorer

Archive Punch Guest Punch Email Punch About Punch

PUNCHTHEBAG

Spiked Punch
20 July, 2009

Newport of the Midwest

Mrs. PTB and I spent last week along Michigan's Little Traverse Bay that sits in between Petoskey and Harbor Springs.

While everyone heads south for vacation, we follow the northern lights which sometimes you see along the 45 degree parallel -- half-way between the North Pole and the Equator. Northern Michigan is perfect this time of year with the crisp and sometimes cold fall air and the turning leaves.

Harbor Springs (and Petoskey for that matter) has been the summer home to some of the wealthiest families since the late 1800s. These are the kind of summer cottages that are passed along in wills from one generation to the next. Those with the bucks are still building big mansions.

Harbor Springs attracts a classy preppy crowd that is well behaved and probably older. It's the kind of wealth that is obvious but they don't rub it your face. Trashy rich types would find this neck of the woods a bit too boring I'm sure. Although it is rumored that Madonna is building a villa in Petoskey.

We rented a beach house on the shores of the bay. We panicked and

In the Shaker

- Gimme Back My Check
- Preacher Osteen
- DiLorenzo the Dunce
- Ben Franklin for Mayor of New Orleans!
- Feeble FEMA
- Stop Listening to Bono about Africa

All elements have a margin, padding, and border property. These properties control the space around the element's contents and other elements around it. These properties stack around an element in concentric box containers, as shown in Figure 32-3.

The element's content (text, image, and so on) are immediately surrounded by padding. The padding defines the distance between the element's contents and its border.

The element's border (if any) is typically drawn right inside the edge of the element's padding.

Note

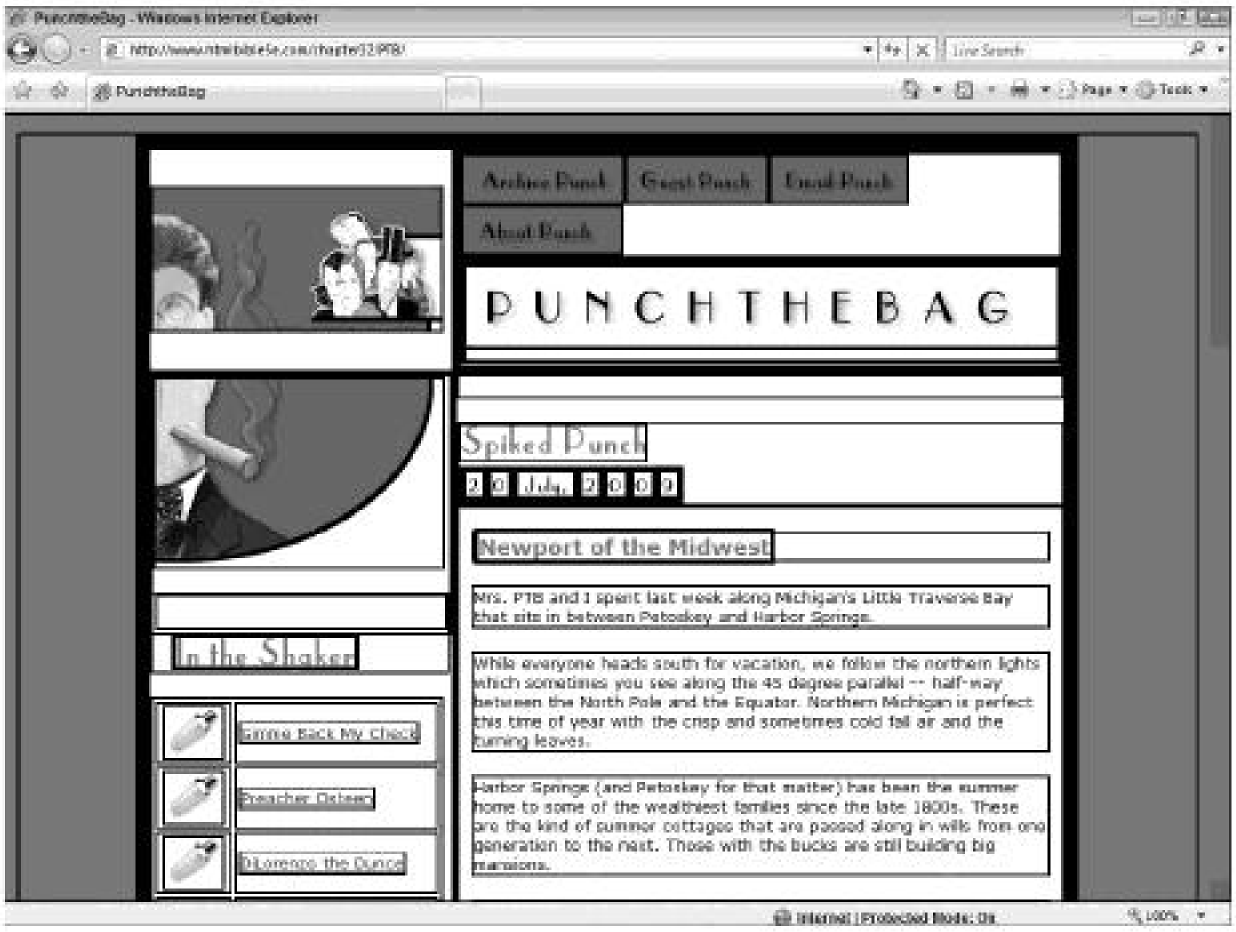
Some user agents place the border on the outside of the edge of an element's padding. ■

The element's margin surrounds the element's border, or the space the border would occupy if no border is defined. The margin defines the distance between the element's padding and neighboring elements.

The next few sections cover each of these properties in more detail.

FIGURE 32-2

Every element in the document is in a box container, as shown when every element's border is enabled.



Spiked Punch

20 July, 2009

Newport of the Midwest

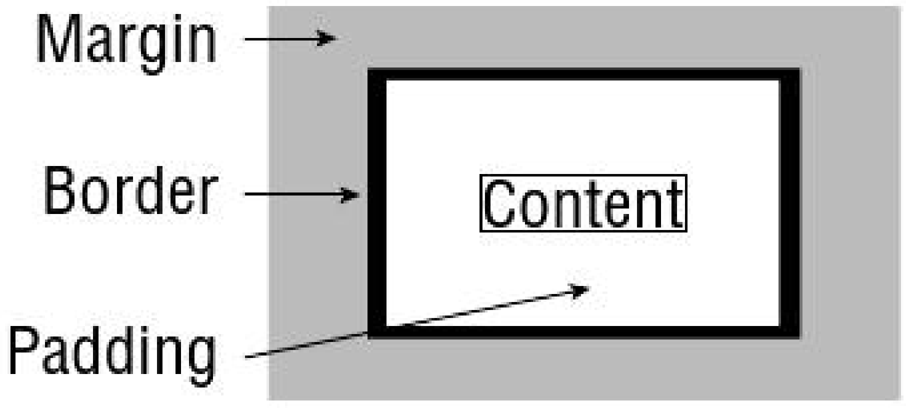
Mrs. PTB and I spent last week along Michigan's Little Traverse Bay that sits in between Petoskey and Harbor Springs.

While everyone heads south for vacation, we follow the northern lights which sometimes you see along the 45 degree parallel -- half-way between the North Pole and the Equator. Northern Michigan is perfect this time of year with the crisp and sometimes cold fall air and the turning leaves.

Harbor Springs (and Petoskey for that matter) has been the summer home to some of the wealthiest families since the late 1800s. These are the kind of summer cottages that are passed along in wills from one generation to the next. Those with the bucks are still building big mansions.

FIGURE 32-3

The box model comprises padding, a border, and a margin.



Element Padding

An element's padding defines the space between the element's contents and the space its border would occupy. This padding space can be increased, decreased, or set to an absolute value using the following CSS properties:

- `padding-top`
- `padding-right`
- `padding-left`
- `padding-bottom`
- `padding`

The first four properties are predictable in their behavior. For example, `padding-top` will change the padding on the top of the element, `padding-right` will change the padding on the right side of the element, and so forth. The fifth property, `padding`, is a shortcut for all sides; its effect is determined by the number of values provided, as explained in Table 32-1.

TABLE 32-1

Padding Values

Number of Values Provided	Effect of the Values
One	All sides are set to the value provided.
Two	The top and bottom are set to the first value provided; the left and right are set to the second value provided.
Three	The top is set to the first value provided, the left and right are set to the second value provided, and the bottom is set to the third value provided.
Four	The top is set to the first value provided, the right is set to the second value provided, the bottom is set to the third value provided, and the left is set to the fourth value provided. (In this case, the values are applied in a clockwise order around the element, starting with the top.)

For example, the following style will set the top and bottom padding value to 5 pixels and the right and left padding to 10 pixels:

```
padding:  5px 10px;
```

Note

Although changing an element's padding value will change its distance from neighboring elements, you should use an object's margin property to control that distance.

However, an element's background color typically extends to the edge of the element's padding. Therefore, increasing an element's padding can extend the background away from an element. This is one reason to use padding instead of margins to increase space around an element. For more information on backgrounds, see Chapter 33. ■

As with all CSS properties, you can specify an absolute value (as in the preceding example) or a relative value. When specifying a relative value, the value is applied to the size of the element's content (such as font size, and so on), not the default value of the padding. For example, the following code would define padding as two times the element's font size:

```
padding:  200%;
```

Element Borders

Borders are among the most versatile CSS properties. As you saw in Figure 32-2, every element in an XHTML document can have a border. However, that figure showed only one type of border — a single, thin, black line displayed around the entire element. Each side of an element can have a different size and style of border, all controlled by CSS properties corresponding to width (thickness), style (solid, dashed, dotted, and so on), and color of the border. The following sections detail how each of the respective CSS properties can be used to affect borders.

Border width

The width of an element's border can be specified using the border width properties, which include the following:

- `border-top-width`
- `border-right-width`
- `border-bottom-width`
- `border-left-width`
- `border-width`

As with other properties that have an effect on multiple sides of an element, there are border width properties for each side, and the `border-width` shortcut property can be used for all sides.

Note

The `border-width` shortcut property accepts one to four values. The way the values are mapped to the individual sides depends on the number of values specified. The rules for this behavior are the same as those used for the `padding` property. Revisit the "Element Padding" section earlier in this chapter for the specific rules. ■

As with other properties, the width can be specified in absolutes or relative units. For example, the first style in the following code sets all of an element's borders to two pixels wide. The second style sets all of an element's borders to 50 percent of the element's content size (generally font size):

```
p.two-pixel  { border-width: 2px;}
p.fifty-percent { border-width: 50%;}
```

You can also use keywords such as `thin`, `medium`, or `thick` to roughly indicate a border's width. The actual width used and what keywords are supported is up to the user agent. If you want exact control over a border's width, you should specify it using absolute values.

Border style

There are ten different types of predefined border styles. These types are shown in Figure 32-4, generated by the following code:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Border Types</title>
  <style type="text/css">
    p { font-size: 12pt; border-width: 6pt;
        text-align: center; padding: 20px;
        margin: 10px; font-weight: bold;}
  </style>
</head>
<body>
<p>
<table width="100%" cellspacing="20px">
<tr><td width="50%">
  <p style="border-style:none ;">None & Hidden</p>
  <p style="border-style:dotted ;">Dotted</p>
  <p style="border-style:dashed ;">Dashed</p>
  <p style="border-style:solid ;">Solid</p>
  <p style="border-style:double ;">Double</p>
</td><td>
<p>&nbsp;</p>
<p style="border-style:groove;">Groove</p>
<p style="border-style:ridge ;">Ridge</p>
<p style="border-style:inset;">Inset</p>
<p style="border-style:outset;">Outset</p>
</td></tr>
</table>
</p>
</body>
</html>
```

Note

The border type `hidden` is identical to the border type `none`, except that the border type `hidden` is treated like a border for border conflict resolution. Border conflicts happen when adjacent elements share a common border (when there is no margin spacing between the elements). In most cases, the most eye-catching border is used. However, if either conflicting element has the conflicting border set to `hidden`, the border between the elements is unconditionally hidden. ■

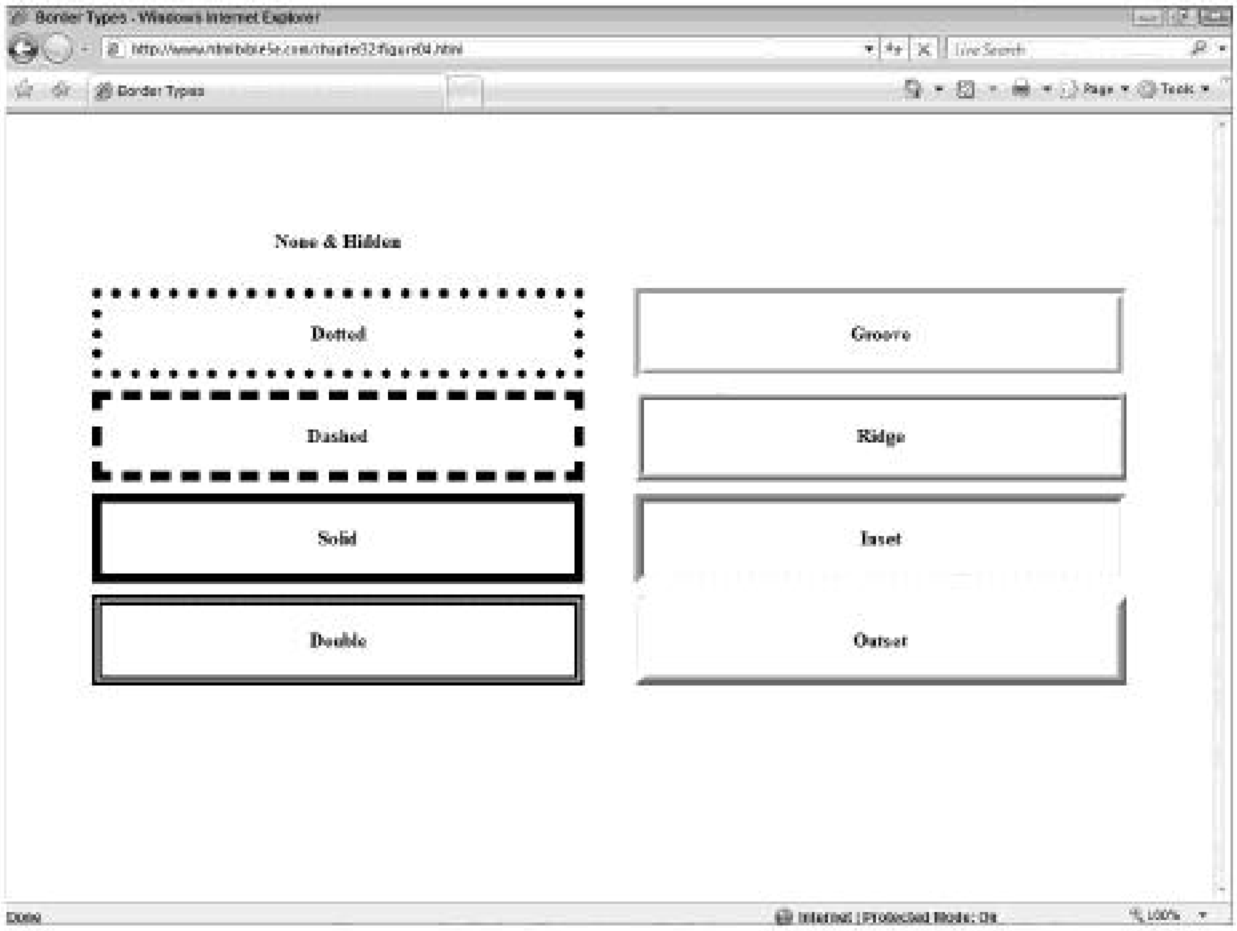
As with other properties of this type, there are several different border style properties:

- `border-top-style`
- `border-right-style`

- `border-bottom-style`
- `border-left-style`
- `border-style`

FIGURE 32-4

The various border styles available via CSS


Border Types - Windows Internet Explorer
http://www.htmlbible5e.com/chapter32/figure04.html
Border Types
Page
Tools
None & Hidden
Dotted
Dashed
Solid
Double
Groove
Ridge
Inset
Outset
Done
Internet | Protected Mode: On
100%

The first four properties affect the side for which they are named. The last, `border-style`, acts as a shortcut for all sides, following the same rules as other shortcuts covered in this chapter. Refer to the "Element Padding" section for more information.

Border color

The border color properties enable you to set the color of the element's visible border. As with the other properties in this chapter, there are border color properties for each side of an element (`border-top-color`, `border-right-color`, and so on) as well as a shortcut property (`border-color`) that can affect all sides.

You can choose from three different methods to specify colors in the border color properties:

- **Color keywords** — `Black`, `white`, `maroon`, and so on. Note that the exact color (mix of red, green, and blue) is left up to the browser and its default colors. (See Appendix A for a list of common color keywords.)
- **Color hexadecimal values** — Values specified in the form `#rrggbb`, where `rrggbb` is two digits (in hexadecimal notation) for each of the colors red, green, and blue. For example, `#FF0000` specifies red (255 red, 0 green, 0 blue) and `#550055` specifies purple (equal parts of red and blue, but no green).
- **Color decimal or percentage values** — Values specified using the `rgb()` function. This function takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255, or a percentage designated with an ending percent sign. For example, the following specifies the color purple (equal parts red and blue, but no green) in integer form and then again in percentages:

```
rgb(100, 0, 100)
rgb(50%, 0%, 50%)
```

Tip

Most graphic editing programs supply color values in multiple formats, including percentage RGB values and perhaps even HTML-style hexadecimal format. ■

Border property shortcuts

You can use the `border` property as a shortcut when specifying an element's border properties. The `border` property has the following syntax:

```
border: < border-width> <border-style> <border-color>;
```

For example, the following two styles set the same border for different paragraph styles:

```
p.one { border-width: thin;
        border-style: solid;
        border-color: black;}
p.two { border: thin solid black;}
```

Like the other properties, the border shortcut also has side-specific variants:

```
border-top
border-right
border-bottom
border-left
```

Each of these properties follows the same syntax as the `border` property, specifying the following properties within its definition:

```
<border-width> <border-style> <border-color>
```

Tip

Keep in mind that you can use CSS inheritance to your advantage when specifying an element's border. For example, suppose you want all but the top border to be thick and black, and you want the top border to be thin and red. Instead of specifying each side individually, you can use the `border` property to define all borders to be thick and black, and then use the `border-top` property to define the top border as an exception, similar to the following:

```
p.bordered  {  border:      thick solid black;
               border-top:  thin solid red;}
```
■

Border spacing

Two additional border properties are worth mentioning here, both of which are primarily used with tables:

- `border-spacing` — This property controls how the user agent renders the space between cells in tables.
- `border-collapse` — This property selects the collapsed method of table borders.

Cross-Ref

These properties are covered in more depth, along with other table properties, in Chapter 30. ■

Element Margins

Margins are the space between an element's border and neighboring elements. Margins are an important property to consider and adjust as necessary within your documents. Most elements have suitable default margins, but sometimes you will find it necessary to increase or decrease an element's margin(s) to suit your unique needs.

For example, consider the image and text shown in Figure 32-5, rendered using the following code:

```
<img src="square.png" style="float: left;"><p>Text next
 to an image using default margins</p>
```

Notice how the "T" in "Text" is almost touching the image next to it. In this case, additional margin space would be welcome.

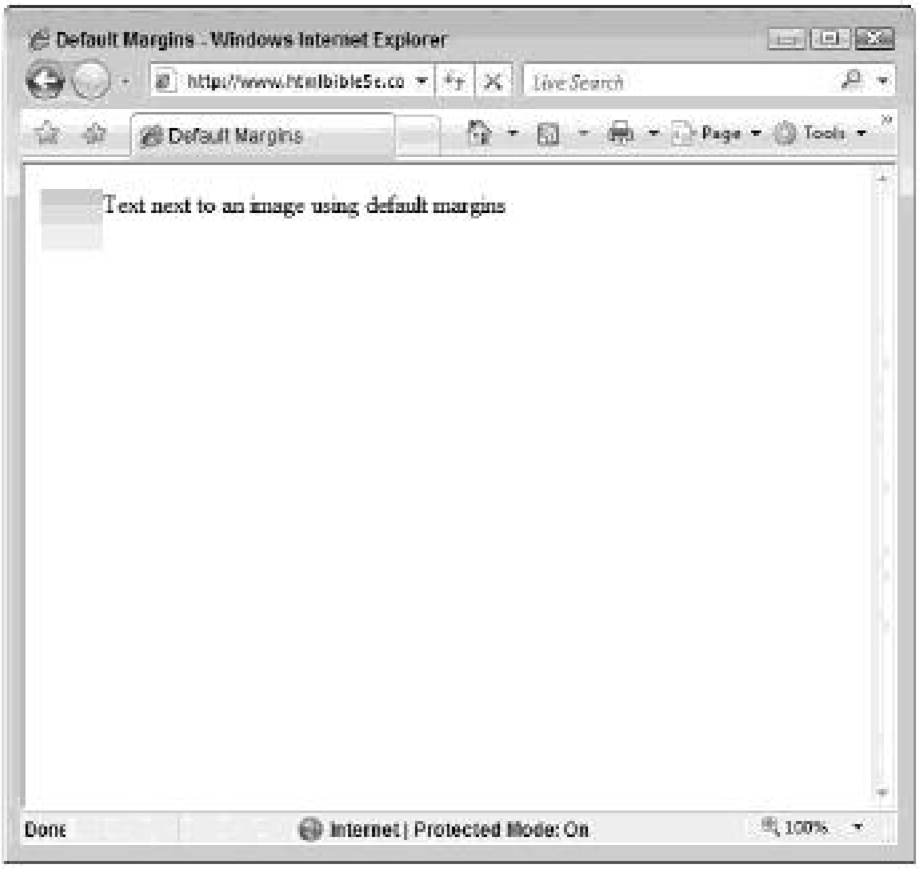
As with other properties in this chapter, margin properties exist for each individual side (`margin-top`, `margin-left`, and so on) as well as a shortcut property to set all sides at once (`margin`). As with the other shortcut properties described herein, the `margin` property accepts one to four values, and the number of values specified determines how the property is applied to an element. See the "Element Padding" section earlier in this chapter for more information.

For example, you can increase the margins of the image in Figure 32-5 using a style similar to the following:

```
margin-right:  5px;
```

FIGURE 32-5

The default borders can sometimes cause elements to render too close to each other.


Default Margins - Windows Internet Explorer
http://www.htmlbible5e.co
Live Search
Default Margins
Page
Tools
Done
Internet | Protected Mode: On
100%

Text next to an image using default margins

This would set the right border of the image (the edge next to the text) to five pixels. Likewise, you can change all four margins using a shortcut such as the following:

```
margin:  2px  4px  10px  4px;
```

Note

The `margin` property is used as a shortcut for all sides, following the same rules as other shortcuts covered in this chapter. See the "Element Padding" section for more information. ■

Just as with the other properties in this chapter, the margin properties have variants that affect each individual side of an element:

```
margin-top
margin-right
margin-bottom
margin-left
```

There are no guidelines for which margins you should adjust on what elements. However, it's usually best to modify the fewest margins possible or to be consistent with which margins you do change. This will help to ensure that your elements look the way you intend and are maintainable with the least amount of effort.

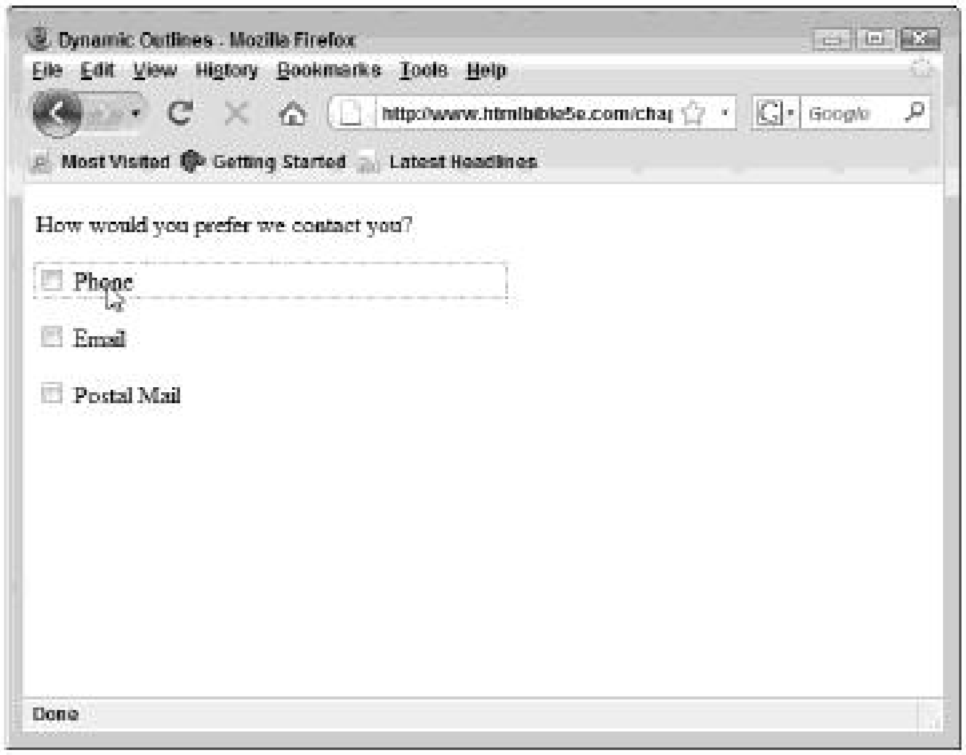
Dynamic Outlines

Outlines are an additional layer that exists around an element, enabling the user agent to highlight that element, if necessary. The highlight is generally used to indicate that a form element has focus. Note that outlines do not occupy any space whether active or not; the element occupies the same amount of space whether its outline is visible or not.

Figure 32-6 shows an example of a dynamic outline around the Phone label.

FIGURE 32-6

Dynamic outlines can be used to highlight objects that have focus or other importance.


Dynamic Outlines - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.htmlbible5e.com/chap
Google
Most Visited Getting Started Latest Headlines
Done

How would you prefer we contact you?

Phone

Email

Postal Mail

Note
Using CSS you can modify the look of outlines. However, unlike other properties of elements covered in this chapter, all sides of an outline must be the same. The CSS properties governing outlines include `outline-color`, `outline-style`, `outline-width`, and the shorthand property `outline`. These properties operate much like the other properties in this chapter, allowing the same values and having the same effects. The format of the `outline` shortcut property is as follows:

```
outline: <outline-color>  <outline-style>  <outline-width>;
```
■

To use the outline properties dynamically, use the `:focus` and `:active` pseudo-elements. These two pseudo-elements specify when an element's outline should be visible — when it has focus or when it is active. For example, the following definitions specify a thick, green border when form elements have focus, and a thin, blue border when they are active:

```
form *:focus { outline-width: thick; outline-color: green;}
form *:active { outline-width: thin; outline-color: blue;}
```

Be aware that, as of this writing, user agent support for outlines is very inconsistent, if support exists at all. If you intend to use outlines in your documents, you should test your code extensively on all platforms you expect your audience to use.

Summary

This chapter covered the box model concept of CSS, the pieces making up the model for each element, and the CSS properties that can be used to influence each. The box model is important because it is part of every element in your document. As you have seen in this chapter, using CSS to modify an element's box can have drastic effects on the element and the document's formatting. The next chapter covers colors and backgrounds.

CHAPTER

33

Colors and Backgrounds

IN THIS CHAPTER

Element Colors

Background Images

The previous chapter covered the box formatting model of CSS. You learned how you can manipulate an element's concentric boxes to better format your HTML documents. This chapter continues the discussion, covering element foreground and background colors, and the use of images for element backgrounds.

Element Colors

Most elements in an HTML document have two color properties, a foreground property and a background property. Both of these properties can be controlled using CSS styles. The following sections discuss both types of color properties.

Foreground colors

An element's foreground color is typically used on the visible portion of that element. In most cases, the visible foreground portion of an element is text, but there are instances where the foreground contains other, nontextual components. You can control the foreground color of an element using the CSS `color` property, which has the following format:

```
color:  <color_value>;
```

As with other properties that use color values, the value of the `color` property can be expressed using one of three methods:

- Predefined color keywords (such as `blue`, `red`, `black`, or `green`)
- Hexadecimal color values in `#rrggbb` form (`#000000` for black, `#FF0000` for red, `#FF00FF` for dark purple, and so on)
- An RGB value using the `rgb()`function (`rgb(100%,0%,0%)` or `rgb(255,0,0)` for red)

For example, the following style defines a class of the paragraph element, which will be rendered with a red font because the foreground color is set to red via the `color` property:

```
p.redtext { color: red; }
```

Expanding on this example, the following paragraph, when used with the preceding style, will be rendered with red text:

```
<p class="redtext">This paragraph is important, and as such, appears
in red text. Other paragraphs in this section that are less
important, appear in standard black text.</p>
```

As with all style properties, you are not limited to element-level definitions. As shown in the following style definition, you can define a generic class that can be used with elements, spans, divisions, and more:

```
.redtext { color: red; }
```

Note

When defining an element's foreground color, you should pay attention to what that element's background color will be, avoiding dark foregrounds on dark backgrounds and light foregrounds on light backgrounds. However, matching foreground and background colors can have its uses, as discussed near the end of the "Background colors" section. ■

Keep in mind that user agent settings can affect the color of elements, as can the user's default local style sheet. If you don't explicitly define an element's color using appropriate styles, it might be otherwise chosen for you.

Background colors

An element's background color can be thought of as the color of the virtual page on which the element is rendered. For example, consider Figure 33-1, which shows two paragraphs: The first is rendered against the user agent's default background (in this case, white) and the second against a light-gray background.

Note

Saying that a document has a default color of white is incorrect. The document will have the color specified in the user agent's settings or in the user's default style sheet, if not otherwise instructed to change it. ■