CHAPTER 5

Web Typography

When you complete this chapter, you will be able to:

- Understand type design principles
- Understand Cascading Style Sheets (CSS) measurement units
- Use the CSS font properties
- Use the CSS text-spacing properties
- Build a font and text properties style sheet
- Customize bulleted and numbered lists

The type choices you make for your site provide the foundation for the clear communication of your content. The consistent use of type to express the hierarchy of your content provides valuable information cues to the reader, and the choices you make to enhance text legibility affect the usability of your Web site. Recent innovations provide powerful tools for increased typographic choices and control. In the past, Web typography meant having to use too many <font> tags and lots of text as graphics. Today, Cascading Style Sheets offers a potent style language, allowing you to manipulate a variety of text properties to achieve professional, effective results, all without resorting to graphics that add download time. The addition of new type properties in CSS3 will allow Web designers even more typographic choices as commercial fonts become available for use.

Understanding Type Design Principles

Type can flexibly express emotion, tone, and structure. Most of the type principles that apply to paper-based design apply to the Web as well. For example, it is possible to go overboard by using too many typefaces and sizes. Just because you have many typefaces at your disposal does not mean you should use them all. In addition, designing for the Web actually restricts your font choices to those your users have installed on their computers.

In strict typography terms, a **typeface** is the name of the type, such as Times New Roman or Futura Condensed. A **font** is the typeface in a particular size, such as Times Roman 24 point. For the most part, on the Web the two terms are interchangeable.

As you work with type, consider the following principles for creating an effective design:

- Choose fewer fonts and sizes.
- Use available fonts.
- Design for legibility.
- Avoid using text as graphics.

Choose Fewer Fonts and Sizes

Your pages will look cleaner when you choose fewer fonts and sizes of type. Decide on a font for each level of topic importance, such as page headings, section headings, and body text. Communicate the hierarchy of information with changes in the size, weight, or color of the typeface. For example, a page heading should have a larger, bolder type, while a section heading would appear in the same typeface, only lighter or smaller.

Pick a few sizes and weights in a type family. For example, you might choose three sizes: a large one for headings, a smaller size

for subheadings, and your body text size. You can vary these styles by changing the weight; for example, bold type can be used for topic headings within text. Avoid making random changes in your use of type conventions. Consistently apply the same fonts and the same combination of styles throughout your Web site; consistency develops a strong visual identity. The Web Style Guide Web site (*www.webstyleguide.com*) shown in Figure 5-1 is a good example of effective type usage. The site has a strong typographic identity, yet uses only two typefaces. The designers of this site built a visually interesting page simply by varying the weight, size, white space, and color of the text.

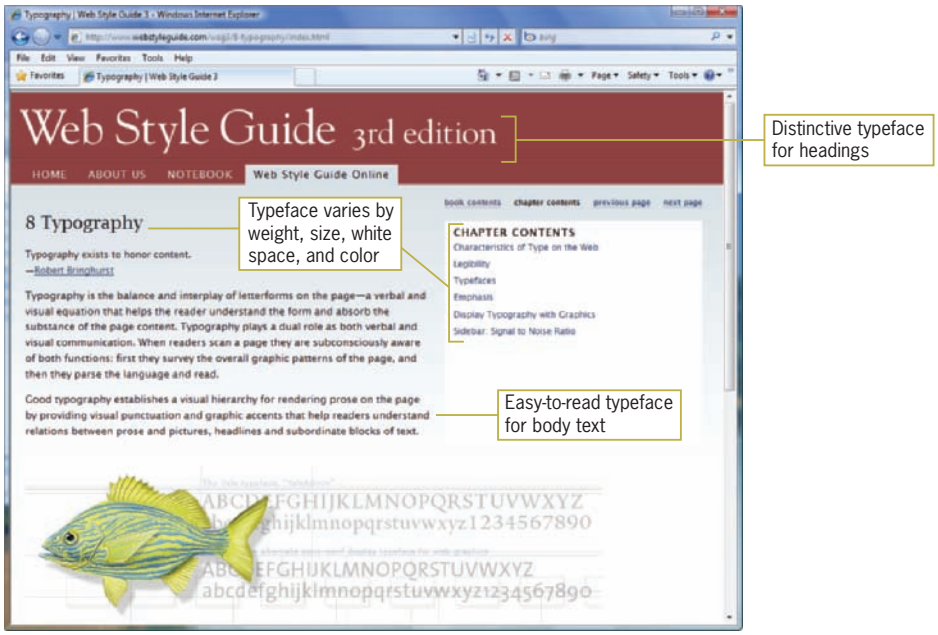

Figure 5-1 Effective typographic design

Use Common Web Fonts

Fonts often are a problem in HTML because font information is client based. The user's browser and operating system determine how a font is displayed, or if it is displayed at all. If you design your pages using a font that your user does not have installed, the browser defaults to Times on a Macintosh or Times New Roman on a PC. To make matters worse, even the most widely available fonts appear in different sizes on different operating systems. Unfortunately, the best you can do about this is to test on multiple platforms to judge the effect on your pages.

To control more effectively how text appears on your pages, think in terms of font families, such as serif and sans-serif typefaces (see Figure 5-2), rather than specific styles. Notice that serif fonts have strokes (or serifs) that finish the top and bottom of the letter. Sans-serif fonts consist of block letters without serifs.

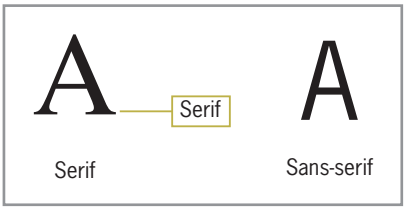

Figure 5-2 Serif and sans-serif type

Because of the variable nature of fonts installed on different computers, you never can be sure the user will see the exact font you have specified. You can, however, specify font fallback values (described later in this chapter), which let you specify a variety of fonts within a font family, such as the common sans-serif fonts, Arial and Helvetica.

Table 5-1 lists the most common installed fonts on the PC, Macintosh, and Linux operating systems according to the Code Style Web site (*www.codestyle.org*).

Common PC Fonts	Common Macintosh Fonts	Common Linux Fonts
Arial	Helvetica	Helvetica
Courier New	Courier	Times
Times New Roman	Times	URW Chancery L
Trebuchet MS	Trebuchet MS	Century Schoolbook
Verdana	Verdana	URW Gothic L
Tahoma	Arial	URW Bookman L
Comic Sans MS	Geneva	Nimbus Mono L
Lucida Console	Lucida Grande	URW Palladio L
Georgia	Monaco	DejaVu Sans Mono

Table 5-1 Common Installed Fonts

The fonts that become the most common for the Web are the result of the Core Fonts for the Web initiative, started by Microsoft in 1996. Their font pack, which was freely distributed, contained Arial, Georgia, Verdana, and Times New Roman. As Table 5-1 shows, Times (or Times New Roman) is available on all three operating systems; it is the default browser font. Courier is the default monospace font, and Arial or Helvetica is usually the default sans-serif font. Arial, Trebuchet, and Verdana come with Internet Explorer, so many Macintosh and PC users have these fonts installed. Some Macintosh users only have Helvetica, so it is a good idea to specify this font as an alternate choice when you are using sans-serif fonts.

Specifying Proprietary Web Fonts

Web designers have traditionally been limited to choosing fonts that are installed on users' systems. The CSS3 font-face property lets you link to a font, download it, and it use it in style rules as if it were installed on the user's computer. The common browsers support the font-face property, though they each implement it differently. Internet Explorer uses its proprietary Embedded Open Type format, while the other major browsers use TrueType or Open Type formats. This inconsistency means that if you use the font-face property, you need to specify multiple fonts to satisfy the needs of different browsers. To solve this problem, Microsoft, Mozilla, and Opera have collaborated on a new type format named the Web Open Font Format (WOFF). WOFF was developed during 2009 and is on its way to becoming a Web standard in 2010. WOFF is designed to offer a single interoperable format for Web fonts that all browsers support.

The font-face property opens a new range of fonts to make Web pages more attractive and legible, but in many instances Web designers or the clients they work with must be prepared to pay licensing fees for the fonts they want to use. Commercial Web sites that license fonts include Typekit (*www.typekit.com*) and Typotheque (*www.typotheque.com*). Other sources for fonts include the Open Font Library (*openfontlibrary.org/media*) and Font Squirrel (*www.fontsquirrel.com/fontface*). You will learn how to use the font-face property later in this chapter.

Design for Legibility

Figure 5-3 shows the same paragraph in Times, Trebuchet, Arial, Verdana, and Georgia at the default browser size in Firefox and Internet Explorer. Although these two examples look almost identical (Opera and Safari offer the same results), remember that browser version, operating system, and video capabilities can produce variations in the weight, spacing, and rendering of the font families to individual users.

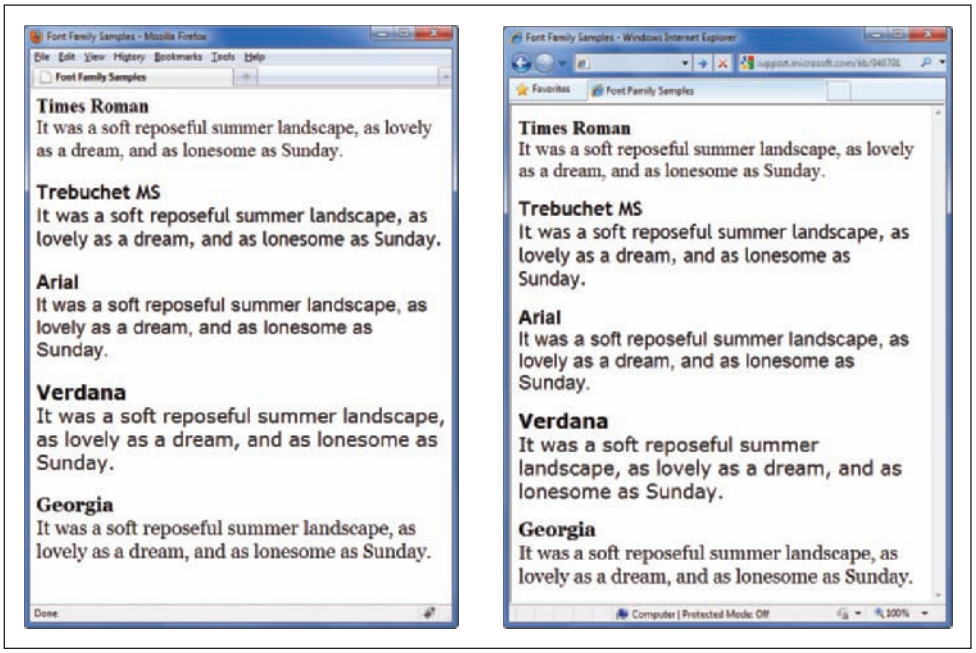

Figure 5-3 Common Web font families in Firefox and Internet Explorer

These examples show that where the text wraps at the end of each line depends on the font and the display characteristics of the browser. Because its **x-height** (the height of the letter *x* in the font) is smaller than that of other fonts, Times can be hard to read, even though it is a serif typeface. This makes it a poor choice for a default font. Trebuchet is a sans-serif face that has a large x-height and rounded letter forms for easy screen legibility. Arial is widely available and is the most commonly used sans-serif font. Verdana is an expanded font—each letter takes up more horizontal space than letters in the other font families. This makes the text easier to read online, but takes much more space on the page.

The size and face of the type you use on your pages determine the legibility of your text. The computer screen has a lower resolution than the printed page, making fonts that are legible on paper more difficult to read on screen. Keep fonts big enough to be legible, and avoid specialty fonts that degrade when viewed online. To aid the reader, consider adding more white space to the page around your blocks of text and between lines as well. Test your content with both serif and sans-serif body text. Finally, make sure that you provide enough contrast between your text color and the background color; in general, darker text on a light background is easiest to read.

Avoid Creating Text as Graphics

The increased number of common fonts and the availability of specialty fonts means that fewer Web designers must resort to creating graphics simply to present text. This technique used to be common in the earlier days of Web design when font choices were more restricted. Still, most Web sites use text graphics in one form or another whether for a main logo, banner, or advertisement. Some sites still use images for navigation graphics, which can be easily substituted with CSS. Because you add download overhead with every additional graphic, save text graphics for important purposes, as described here. Remember that including text as graphics means users cannot search for that text, and that your content will not be accessible to users with screen readers and other adaptive devices. Whenever possible, use HTML-styled text on your pages including creating HTML and CSS-based navigation, which you will learn about in Chapter 9.

You can control how your Web pages look when they are printed with a print style sheet, described in Appendix D.

Understanding CSS Measurement Units

CSS offers a variety of measurement units, including absolute units, such as points, and relative units, such as ems. The measurement values you choose depend on the destination medium for your content. For example, if you are designing a style sheet for printed media, you can use absolute units of measurement, such as points or centimeters. (See Appendix D, Print Style Sheets, for more information.) When you are designing a style sheet for a Web page, you can use relative measurement values that adapt to the user's display type, such as ems or pixels. In this section, you will learn about the CSS measurement units. These units are detailed in Table 5-2.

Unit	Unit Abbreviation	Description
Absolute Units		
Centimeter	cm	Standard metric centimeter
Inch	in	Standard U.S. inch
Millimeter	mm	Standard metric millimeter
Pica	pc	Standard publishing unit equal to 12 points
Point	pt	Standard publishing unit, with 72 points in an inch
Relative Units		
Em	em	The width of the capital *M* in the current font, usually the same as the font size; 1 em is the default font size, 1.5 em is one-and-one-half times the default font size, and so on
Ex	ex	The height of the letter *x* in the current font
Pixel	px	The size of a pixel on the current monitor
Percentage	Example: 150%	Works exactly like em: 100% is the default font size, 150% em is one-and-one-half times the default font size, and so on

Table 5-2 CSS Measurement Units

Absolute Units

CSS lets you use absolute measurement values that specify a fixed value. The measurement values require a number followed by one of the unit abbreviations listed in Table 5-2. By convention, do not include a space between the value and the measurement unit. The numeric value can be a positive value, negative value, or fractional value. For example, the following rule sets margins to 1.25 inches:

```
p {margin: 1.25in;}
```

You generally want to avoid using absolute units for Web pages because they cannot be scaled to an individual user's display type. Absolute units are appropriate when you know the exact measurements of the destination medium. For example, if you know a document will be printed on 8.5 × 11-inch paper, you can plan your style rules accordingly because you know the physical dimensions of the finished document. For this reason, absolute units are better suited to print destinations than Web destinations. Although the point is the standard unit of measurement for type sizes, it is not the best measurement value for the Web. Because computer displays vary widely in size, they lend themselves better to relative units of measurement that can adapt to different monitor sizes and screen resolutions.

Relative Units

The relative units are designed to let you build scalable Web pages that adapt to different display types and sizes. This practice ensures that your type sizes are properly displayed relative to each other or to the default font size set for the browser.

Relative units are always relative to the inherited size of their containing element. For example, the following rule sets the font size for the <body> element to 1.5 times (150%) the size of the browser default:

```
body {font-size: 150%;}
```

Child elements inherit the percentage values of their parents. For example, a <p> element within this body element inherits the 150% sizing.

The em Unit

The em is a printing measurement, traditionally equal to the horizontal length of the capital letter *M* in any given font size. In CSS, the **em unit** is equal to the font size of an element. It can be used for both horizontal and vertical measurement. In addition to stating font sizes, em is useful for padding and margins. You can read more about this in Chapter 6.

The size of the em is equivalent to the font size of the element. For example, if the default paragraph font size is 12-point text, the em equals 12 points. Stating a text size of 2em creates 24-point text—two times the default size. This is useful because it means that measurements stated in em are always relative to their environment. For example, assume that you want a large heading on your page. If you set the <h1> element to 24 points, it always remains that size. If a user sets his or her default font size to 24 points, the headings are the same size as the text. However, if you use the relative em unit, the size of the heading is always based on the size of the default text. The following rule sets heading divisions to twice the size of the default text:

```
div.heading {font-size: 2em;}
```

Percentage

Percentage values for fonts work exactly the same as ems described above. For example, if the default paragraph font size is 12-point text, a 100% font size equals 12 points. A font size set to 125% based on a 12-point default would be 15 points.

The ex Unit

The **ex unit** is equal to the height of the lowercase letter *x* in any given font. As shown in Figure 5-4, the height of the lowercase letter *x* varies widely from one typeface to another.

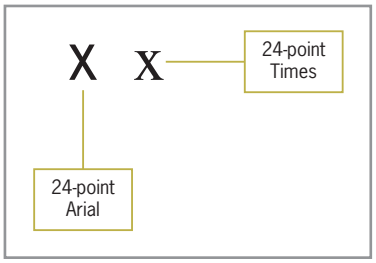

Figure 5-4 Differences in height in the ex unit

Ex is a less reliable unit of measurement than em because the size of the letter *x* changes in height from one font family to another, and the browser cannot always calculate the difference correctly. Most browsers simply set the ex value to one-half the value of the font's em size. Ex is not commonly used to set font sizes.

The px Unit

Pixels are the basic picture element of a computer display. The size of the pixel is determined by the display resolution. Resolution is the measure of how many pixels fit on a screen. As the resolutions grow in value, the individual pixel size gets smaller, making the pixel relative to the individual display settings. Pixel measurements work well for computer displays, but they are not so well suited to other media, such as printing, because some printers cannot accurately determine the size of the pixel. Pixels are not a good choice for font sizes because they do not adapt well when the user changes the size of their display fonts. Ems or percentages are always the best choice.

CSS Property Descriptions

The property descriptions on the following pages and in other chapters provide key information about each CSS property. A property description looks like the following:

border-width property description

Value:	thin | medium | thick | <length>
Initial:	medium
Applies to:	all elements
Inherited:	no
Percentages:	N/A

Table 5-3 lists the five property description categories.

Category	Definition
Value	The valid keyword or variable values for the property; variable values are set between angle brackets; for example, <length> means enter a length value; (Table 5-4 lists the value notation symbols)
Initial	The initial value of the property
Applies to	The elements to which the property applies
Inherited	Indicates if the property is inherited from its parent element
Percentages	Indicates if percentage values are allowed

Table 5-3 Property Description Categories

Table 5-4 lists the value category notation.

Notation	Definition
< >	Words between angle brackets specify a variable value; for example, <length>
|	A single vertical bar separates two or more alternatives, one of which must occur; for example, thin | medium | thick
||	Two vertical bars separate options; one or more of the values can occur in any order; for example, underline || overline || line-through
[ ]	Square brackets group parts of the property value together; for example, none | [underline || overline || line-through] means that the value is either none or one of the values within the square brackets
?	A question mark indicates that the preceding value or group of values is optional

Table 5-4 Value Category Notation

Using the CSS Font Properties

The CSS font properties allow you to control the appearance of your text. These properties describe how the form of each letter looks. The CSS text properties, covered later in this chapter, describe the spacing around the text rather than affecting the actual text itself. In this chapter, you will learn about the following properties:

- font-family
- font-face
- font-size
- font-style
- font-variant
- font-weight
- font-stretch
- font-size-adjust
- font (shorthand property)

Specifying Font Family

font-family property description	
Value:	<family-name> | <generic-family>
Initial:	depends on user agent
Applies to:	all elements
Inherited:	yes
Percentages:	N/A

The font-family property lets you state a generic font-family name, such as sans-serif, or a specific font-family name, such as Helvetica. You can also string together a list of font families, separated by commas, supplying a selection of fonts that the browser can attempt to match. Font names containing more than one word must be quoted.

When considering fonts for your Web designs, start by thinking in terms of font families, such as serif and sans-serif typefaces, rather than specific styles. If you are not using licensed fonts, be aware of the variable nature of fonts installed on different computers. You

can never be sure that the user will see the exact font you have specified. You can, however, use font fallback values to specify a variety of fonts within a font family, such as Arial or Helvetica, which are both common sans-serif fonts.

Generic Font Families

You can use the following generic names for font families:

- **Serif** fonts are the traditional letter form, with strokes (or serifs) that finish off the top and bottom of the letter. The most common serif font is Times.
- **Sans-serif** fonts have no serifs. They are block letters. The most common sans-serif fonts are Helvetica and Arial.
- **Monospace** fonts are fixed-width fonts. Every letter has the same horizontal width. Monospace is commonly used to mimic typewritten text or for programming code. The style rules and HTML code in this book are printed in Courier, a monospace font.
- **Cursive** fonts are designed to resemble handwriting. Although often displayed as Comic Sans, this choice can provide inconsistent results.
- **Fantasy** fonts are primarily decorative. Fantasy is not a widely used choice.

The practice of using generic names ensures greater portability across browsers and operating systems because it does not rely on a specific font being installed on the user's computer. The following rule sets <p> elements to the default sans-serif font:

```
p {font-family: sans-serif;}
```

Of course, if you don't specify any font family, the browser displays the default font, usually some version of Times. Figure 5-5 shows the generic font families in Internet Explorer, Firefox, Opera, and Safari. Notice the difference in the display size of the monospace font. Also notice that the cursive font is actually Comic Sans, except in Internet Explorer 8, where it does not resemble handwriting. If a certain font is not available, a different font will be substituted, based on the user's operating system.

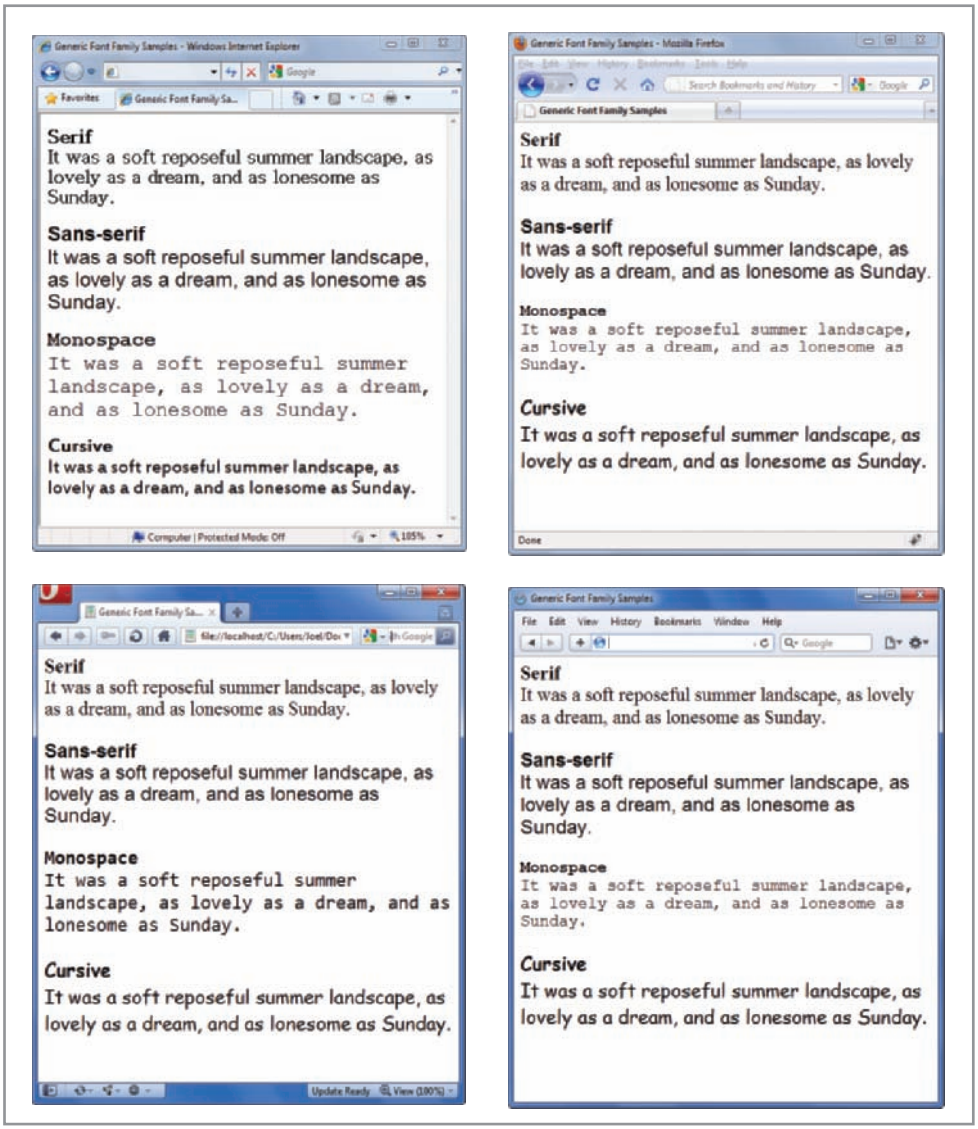

Figure 5-5 Generic font families in Internet Explorer, Firefox, Opera, and Safari

Specific Font Families

In addition to generic font families, the font-family property lets you declare a specific font family, such as Futura or Garamond. The user must have the font installed on his or her computer; otherwise, the browser uses the default font. If the font family name contains white space, such as "lucida console," the font name must be contained within quotes.

The following rule specifies Lucida Console as the font family for the <p> element:

```
p {font-family: "lucida console";}
```

Font Fallbacks

You can specify a list of alternate fonts using a comma as a separator. The browser attempts to load each successive font in the list. If no fonts match, the browser falls back to the default font. The following code tells the browser to use Arial; if Arial is not present, use Helvetica.

```
p {font-family: arial, helvetica;}
```

This font substitution string produces a sans-serif font on PCs that have Arial installed and Macintosh computers that have Helvetica installed. To further ensure the portability of this rule, add a generic font family name to the list, as shown in the following rule:

```
p {font-family: arial, helvetica, sans-serif;}
```

This rule ensures that the <p> element is displayed in some type of sans-serif font, even if it is not Arial or Helvetica.

You should always include a default generic font, such as sans-serif, as the last choice in your font fallback choices. Doing so makes sure that the browser does not use its default font.

Specifying Font-Face

The @font-face property lets you specify a font to be downloaded and displayed in the browser, overcoming the limitations of only using fonts that reside on a user's computer. The font-face property lets you define the name and location of the desired font. Fonts are usually specified in the TrueType Format (TTF). You then specify the font using the font-family property. The following code shows an example of the font-face property.

```
@font-face {font-family: generica;
   src: url(http://www.generic.com/fonts/generica.ttf)}

h1 {font-family: generica, serif;}
```

Remember to always include fallback values in case the browser has a problem downloading the primary font.

The following code shows an example of specifying type in the two formats:

```
@font-face {
   font-family: generic;

/* EOT Format */
   src: url(http://www.generic.com/fonts/generica.eot)}
```

Most modern browsers support the @font-face property, but as described earlier, Internet Explorer 8 and earlier use a proprietary Microsoft font format. This means you must specify two versions of the same font, one in the Microsoft EOT format and one in TTF format. You can convert fonts to Microsoft's format with one of the following online converters. Remember to specify fallback values that fit your design needs.

www.microsoft.com/typography/weft.mspx

www.kirsle.net/wizards/ttf2eot.cgi

Most Web fonts are proprietary and must be purchased and licensed for use. Using licensed fonts protects you from legal liability and also protects the rights of the font designers.

```
/* TTF Format */
   src:
url(http://www.generic.com/fonts/generica.eot)format("truetype");
}
```

Specifying Font Size

font-size property description	
Value:	<absolute-size> | <relative-size> | <length> | <percentage>
Initial:	medium
Applies to:	all elements
Inherited:	computed value is inherited
Percentages:	refer to parent element's font size

The font-size property gives you control over the specific sizing of your type. You can choose from length units, such as ems or pixels, or a percentage value that is based on the parent element's font size.

The following rule sets the <blockquote> element to 1.5 em Arial:

```
blockquote {font-family: arial, sans-serif; font-size: 1.5em;}
```

To specify a default size for a document, use *body* as the selector. This rule sets the text to .85 em Arial:

```
body {font-family: arial, sans-serif; font-size: .85em;}
```

You can also choose from a list of absolute size and relative size keywords, described in the following sections.

Absolute Font Size Keywords

These keywords refer to a table of sizes that is determined by the browser. The keywords are:

- xx-small
- x-small
- small
- medium
- large
- x-large
- xx-large

The CSS specification recommends a scaling factor of 1.2 between sizes for the computer display. Therefore, if the medium font is 10 points, the large font would be 12 points (10 × 1.2 = 12).

Solving the Font Size Dilemma

Figure 5-6 shows a variety of font size samples in the Google Chrome browser. With so many methods of font sizing available, which should you use? The designers of CSS2, Hakon Lie, and Bert Bos, recommend always using relative sizes (specifically, the em value) to set font sizes on your Web pages. Here are some reminders about each relative measurement value:

- *Ems or percentage*—The best choice for fonts because ems and percentages are scalable based on the user's default font size. For the same reason, padding and margins specified in ems are another way to make Web pages more adaptable.
- *Pixels*—Pixel values entirely depend on the user's screen resolution, making it very difficult to ensure consistent presentation. Pixels are a good choice for borders and other design elements, but not a good choice for fonts.

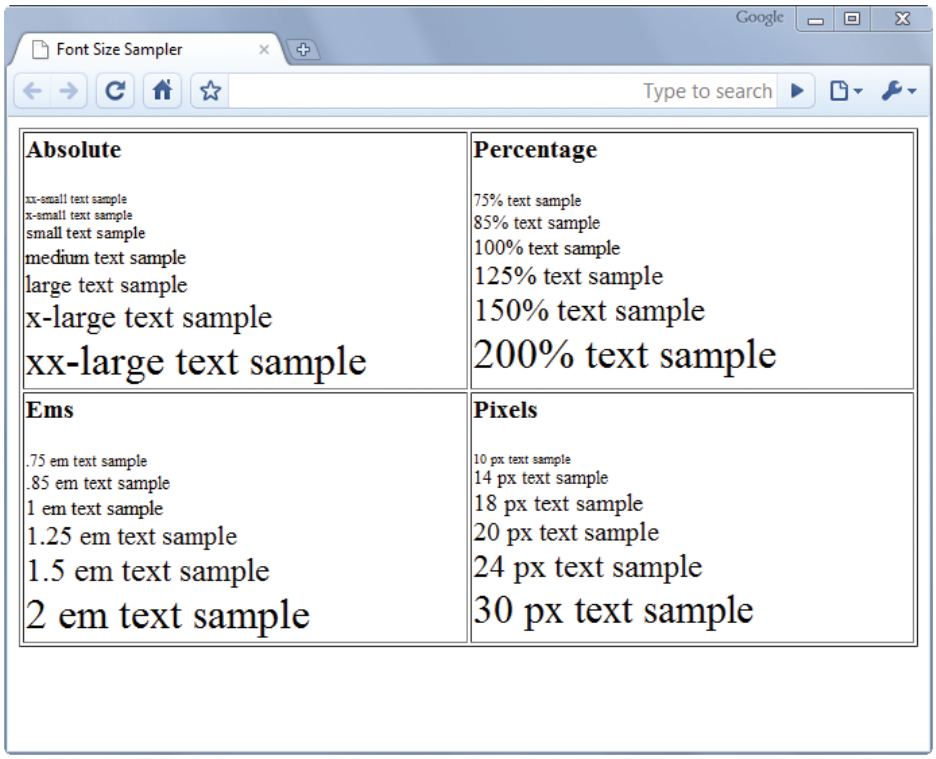

Figure 5-6 Various font sizes

Specifying Font Style

font-style property description	
Value:	normal | italic | oblique
Initial:	normal
Applies to:	all elements
Inherited:	yes
Percentages:	N/A

The font-style property lets you specify italic or oblique text. The difference between italic and oblique text is subtle. The italic form of a typeface is designed with different letter forms to create the slanted font, while the oblique form is simply normal text slanted to the right. In print-based typography, oblique text is considered inferior to italic. On the Web, however, current browsers cannot make the distinction between the two—either value creates slanted text. The following example sets italicized text for the note class attribute.

```
.note {font-style: italic;}
```

Here is the note class applied to a element:

```
<div class="note">A note to the reader:</div>
```

The text contained in the appears italicized in the browser. Remember that italic text is hard to read on a computer display. Use italics for special emphasis rather than for large blocks of text.

Specifying Font Variant

font-variant property description	
Value:	normal | small-caps
Initial:	normal
Applies to:	all elements
Inherited:	yes
Percentages:	N/A

The font-variant property lets you define small capitals, which are often used for chapter openings, acronyms, and other special purposes. Small capitals are intended to be a different type style from regular capital letters, but this is not supported in all browsers. In fact, some simply downsize the regular capital letters. Figure 5-7 shows an example of small capitals.

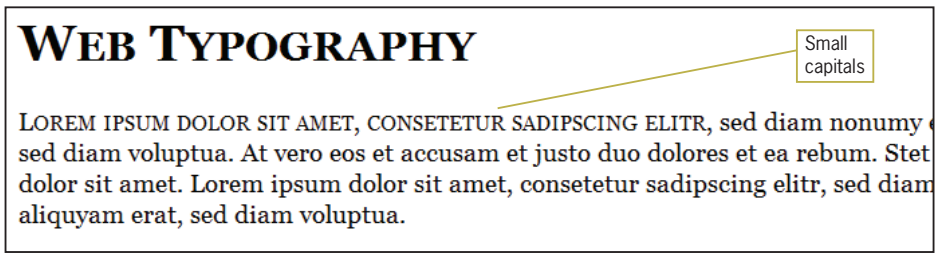

Figure 5-7 Small capitals add a distinctive look

In this example, a style rule specifies a class named *small* to be used with a element. The element in the HTML code contains the text that is converted to small capitals.

```
span.small {font-variant: small-caps}
```

```
<p><span class="small">I first heard of Antonia</span> on what
seemed to be...</p>
```

Specifying Font Weight

font-weight property description	
Value:	normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900
Initial:	normal
Applies to:	all elements
Inherited:	yes
Percentages:	N/A

The font-weight property lets you set the weight of the typeface. The numeric values express nine levels of weight from 100 to 900, although most browsers and fonts do not support such a wide range of weights. The default type weight is equal to 400, with bold text equal to 700. Bolder and lighter are relative weights based on the weight of the parent element. Using the bold value produces the same weight of text as the element. The following style rule sets the *warning* class to bold:

```
.warning {font-weight: bold;}
```

Specifying Font Stretch

font-stretch property description	
Value:	normal | wider | narrower | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded | inherit
Initial:	normal
Applies to:	all elements
Inherited:	yes
Percentages:	N/A

The font-stretch property is a CSS3 property that currently has limited support in browsers.

The font-stretch property lets you expand or compress the font face. Not all forms of the font may be available, so the next closest choice is substituted: the closest condensed face substitutes for any condensed choice, and the closest expanded face for any expanded choice. The following rule expands the h1 font face if available.

```
h1 {font-family: sans-serif; font-stretch: expanded}
```

Using the Font Shortcut Property

font property description	
Value:	[[<'font-style'> || <'font-variant'> || <'font-weight'>]? <'font-size'> [/ <'line-height'>]? <'font-family'>]
Initial:	see individual properties
Applies to:	all elements
Inherited:	yes
Percentages:	allowed on 'font-size' and 'line-height'

The **font property** is a shortcut that lets you specify the most common font properties in a single statement. The syntax of this property is based on a traditional typographical shorthand notation to set multiple properties related to fonts.

As shown in the value listing for the font shortcut property, the font property lets you state the font-style, font-variant, font-weight, font-size, line-height, and font-family in one statement. The only two values that are required are font-size and font-family, which must be in the correct order for the style rule to work. The following rules are examples of basic use of the font property:

```
p {font: 12pt arial, sans-serif;}
h1 {font: 2em sans-serif;}
```

The font properties other than font-size and font-family are optional and do not have to be included unless you want to change

their default. If you want to include line-height, note that it must always follow a slash after the font-size. The following rule sets .85em Arial text on 1em line height:

```
p {font: .85em/1em arial;}
```

The font shortcut property lets you abbreviate the more verbose individual property listings. For example, both of the following rules produce the same result:

```
p {font-weight: bold;
   font-size: .85em;
   line-height: 1em;
   font-family: arial;
}

p {font: bold .85em/1em arial;} /* Same rule as above */
```

Although the font shortcut property is a convenience, you may prefer to state explicitly the font properties as shown in the more verbose rule, because they are easier to understand. It is also a good idea to choose a convention of using either the individual property names or the shortcut notation, and then use the convention consistently throughout your Web site.

Using the CSS Text Spacing Properties

The CSS text properties let you adjust the spacing around and within your text and add text decorations. The properties in this section let you create distinctive text effects. In this section, you will learn about the following properties:

- text-indent
- text-align
- line-height
- vertical-align
- letter-spacing
- word-spacing
- text-decoration
- text-transform
- text-shadow

Specifying Text Indents

text-indent property description	
Value:	<length> | <percentage>
Initial:	0
Applies to:	block-level element
Inherited:	yes
Percentages:	refer to width of containing block

Use the text-indent property to set the amount of indentation for the first line of text in an element, such as a paragraph. You can specify a length or percentage value. The percentage is relative to the width of the containing element. If you specify a value of 15%, the indent will be 15% of the width of the element. Negative values let you create a hanging indent. Figure 5-8 shows two text-indent effects.

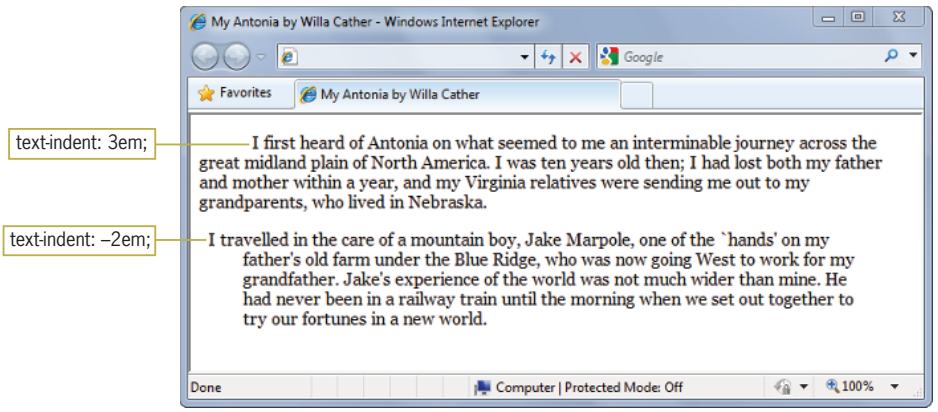

Figure 5-8 Text indents

The following rules set an indent of 2em for the <p> element and -2em for the <blockquote> element:

```
p {text-indent: 2em;}
blockquote {text-indent: –2em;}
```

Indents are sensitive to the language specification for the document. You can specify the document language with the *lang* attribute in the opening <html> tag, such as <html lang="en">. In left-to-right reading languages (such as English), the indent is added to the left of the first line; in right-to-left reading languages (such as Hebrew), the indent is added to the right of the first line.

Indents are inherited from parent to child elements. For example, the following rule sets a 2em text indent to a element:

```
div {text-indent: 2em;}
```

Any block-level elements, such as <p>, that are contained within this division have the same 2em text indent specified in the rule for the parent .

You can find a complete list of the standard language codes here: *www.loc.gov/standards/iso639-2/php/code_list.php*

Specifying Text Alignment

text-align property description	
Value:	left | right | center | justify
Initial:	depends on user agent and language
Applies to:	block-level elements
Inherited:	yes
Percentages:	N/A

Use the text-align property to set horizontal alignment for the lines of text in an element. You can specify four alignment values: left, center, right, and justify. The justify value lines up the text on both horizontal margins, adding white space between the words on the line, like a column of text in a newspaper. The following style rule sets the <p> element to justified alignment:

```
p {text-align: justify;}
```

Figure 5-9 shows a sample of all four alignment values.

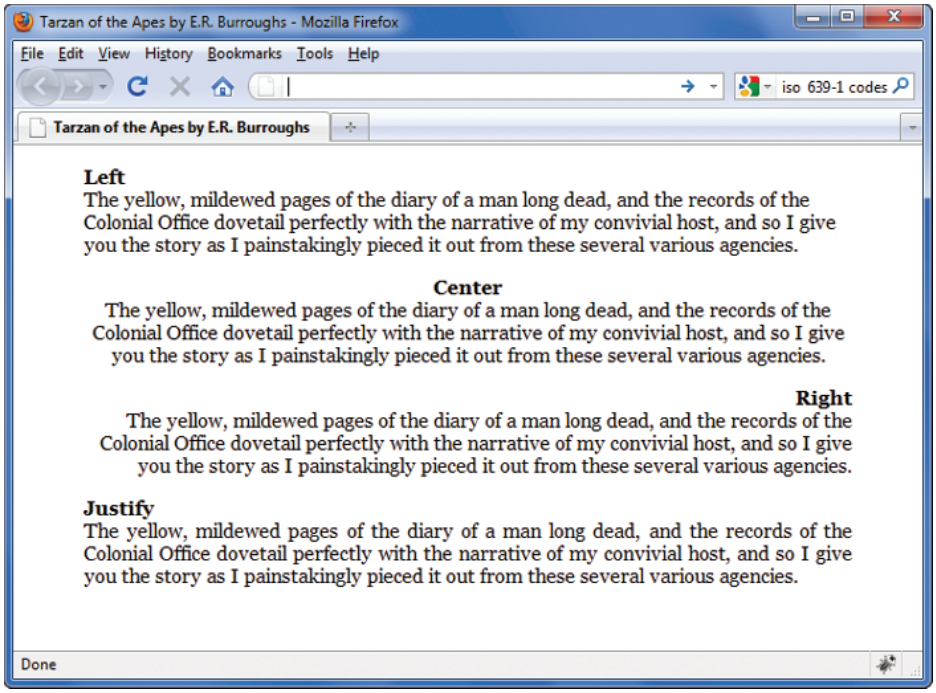

Figure 5-9 Text alignments

When choosing an alignment value, keep the default settings for the language and the user's preferences in mind. For example, most Western languages read from left to right, and the default alignment is left. Unless you are trying to emphasize a particular section of text, use the alignment with which most readers are comfortable. Both right and center alignment are fine for short sections of text, but they make reading difficult for lengthier passages.

Justified text lets you create newspaper-like alignment where the lines of text all have the same length. The browser inserts white space between the words of the text to adjust the alignment so both margins of the text align, as shown in Figure 5-10. Justify is not supported by all browsers, and different browsers might justify the text differently.

Specifying Line Height

line-height property description	
Value:	normal | <number> | <length> | <percentage>
Initial:	normal
Applies to:	all elements
Inherited:	yes
Percentages:	refer to the font size of the element itself

CSS allows you to specify either a length or percentage value for the line height, which is also known as **leading**, the white space between lines of text. The percentage is based on the font size. Setting the value to 150% with a 1em font size results in a line height of 1.5em. The following rule sets the line height to 150%:

```
p {line-height: 150%;}
```

Figure 5-10 shows the default line height and various adjustments in line height. Notice that the line height is evenly divided between the top and bottom of the element.

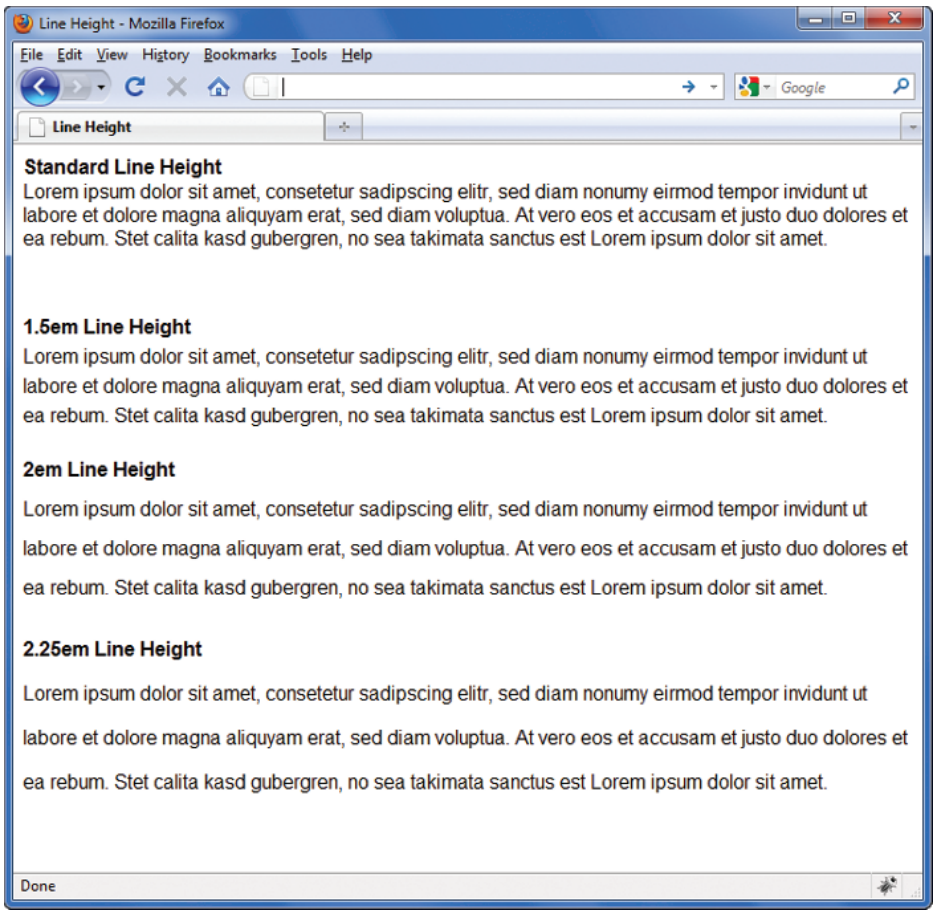

Figure 5-10 Adjusting line height increases legibility

The line-height property can increase the legibility of your text. Adding to the default line height inserts additional white space between the lines of text. On a display device, increasing the white space helps guide the user's eyes along the line of text and provides rest for the eye. As Figure 5-10 shows, increasing the line height adds to the legibility of the text.

Specifying Vertical Alignment

vertical-align property description	
Value:	baseline | sub | super | top | text-top | middle | bottom | text-bottom | <percentage> | <length>
Initial:	baseline
Applies to:	inline-level and 'table-cell' elements
Inherited:	no
Percentages:	refer to the 'line-height' of the element itself

The vertical-align property lets you adjust the vertical alignment of text within the line box. Vertical-align works only on inline elements. You can use this property to superscript or subscript characters above or below the line of text and to align images with text. Table 5-5 defines the different vertical-align values. The baseline, sub, and super values are the most evenly supported by the different browsers.

Value	Definition
baseline	Align the baseline of the text with the baseline of the parent element
sub	Lower the baseline of the box to the proper position for subscripts of the parent's box; this value does not automatically create a smaller font size for the subscripted text
middle	The CSS2 specification defines "middle" as "the vertical midpoint of the box with the baseline of the parent box plus half the x-height of the parent"; realistically, this means the middle-aligned text is aligned to half the height of the lowercase letters
super	Raise the baseline of the box to the proper position for superscripts of the parent's box; this value does not automatically create a smaller font size for the superscripted text
text-top	Align the top of the box with the top of the parent element's font
text-bottom	Align the bottom of the box with the bottom of the parent element's font
top	Align the top of the box with the top of the line box
bottom	Align the bottom of the box with the bottom of the line box

Table 5-5 Vertical-align Property Values

The following rule sets superscripting for the superscript class:

```
.superscript {vertical-align: super;}
```

Figure 5-11 shows different types of vertical alignments.

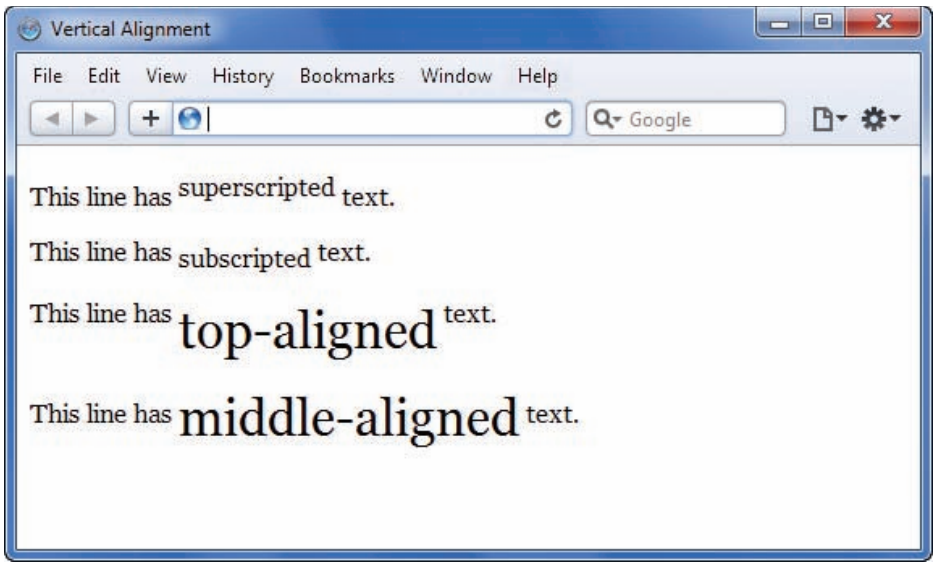

Figure 5-11 Vertical alignments

You can also use vertical alignment to align text with graphics. The following rule, added to the <img> element with the style attribute, sets the vertical alignment to top:

```
<img src="image.gif" style="vertical-align: text-top;">
```

Figure 5-12 shows various alignments of images and text. Note that the vertical alignment affects only the one line of text that contains the graphic, because the graphic is an inline element. If you want to wrap a paragraph of text around an image, use the float property, described in Chapter 6.

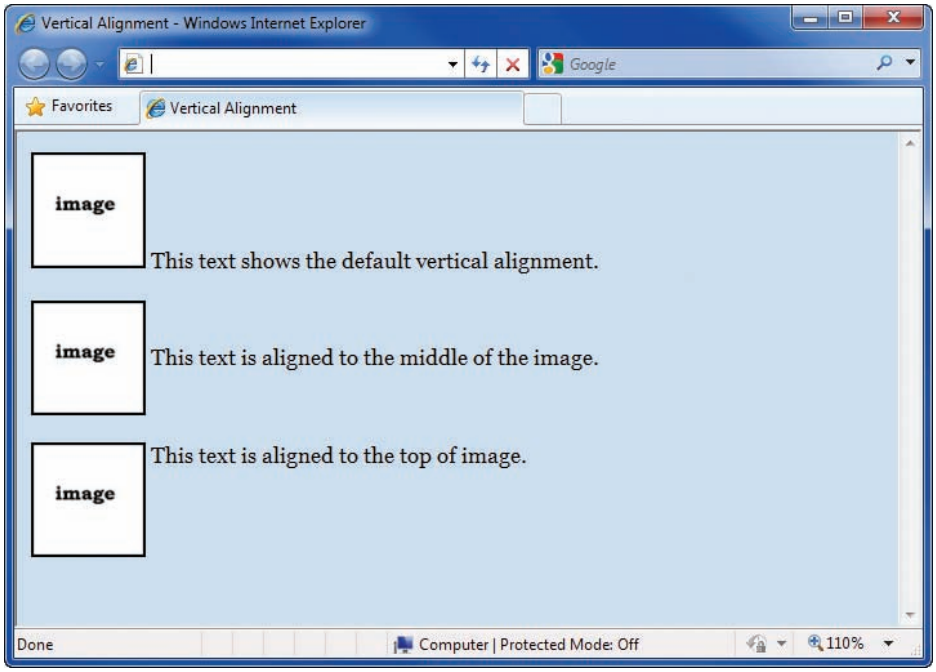

Figure 5-12 Vertically aligning text and graphics

Specifying Letter Spacing

letter-spacing property description	
Value:	normal | <length>
Initial:	normal
Applies to:	all elements
Inherited:	yes
Percentages:	N/A

The letter-spacing property lets you adjust the white space between letters. In publishing terminology, this adjustment is called kerning. The length you specify in the style rule is added to the default letter spacing. The following code sets the letter spacing to 4 pixels:

```
h1 {letter-spacing: 4px;}
```

Figure 5-13 shows samples of different letter-spacing values. The letter-spacing property is an excellent method of differentiating headings from the rest of your text.

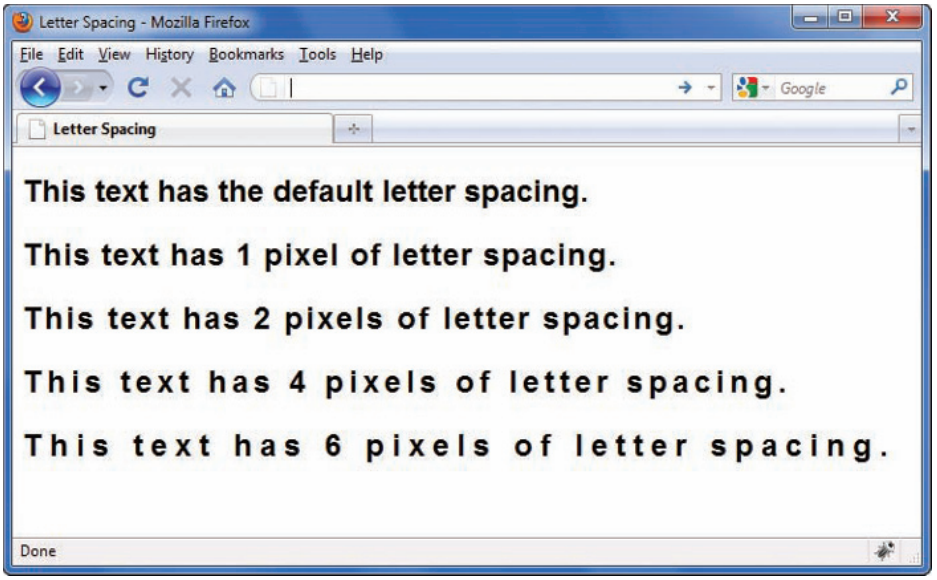

Figure 5-13 Adjusting letter spacing

Specifying Word Spacing

word-spacing property description	
Value:	normal | <length>
Initial:	normal
Applies to:	all elements
Inherited:	yes
Percentages:	N/A

The word-spacing property lets you adjust the white space between words in the text. The length you specify in the style rule is added to the default word spacing. The following code sets the word spacing to 2em:

```
h1 {word-spacing: 2em;}
```

Figure 5-14 shows the result of the word-spacing property. Like the letter-spacing property, word-spacing is an effective way to make your headings stand out.

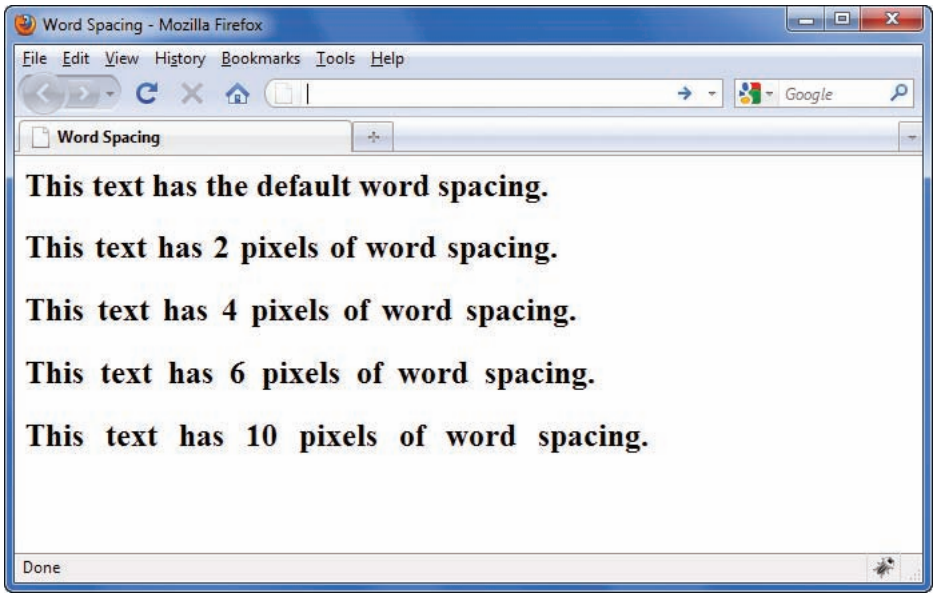

Figure 5-14 Adjusting word spacing

Specifying Text Decoration

text-decoration property description	
Value:	none | [underline || overline || line-through || blink]
Initial:	none
Applies to:	all elements
Inherited:	no
Percentages:	N/A

Text decoration lets you underline text, an effect that has particular meaning in a hypertext environment. See Figure 5-15. Your users know to look for underlined words as the indicators for hypertext links. Any text you underline appears to be a hypertext link. Except for text links, underlining is an inappropriate text style for a Web page.

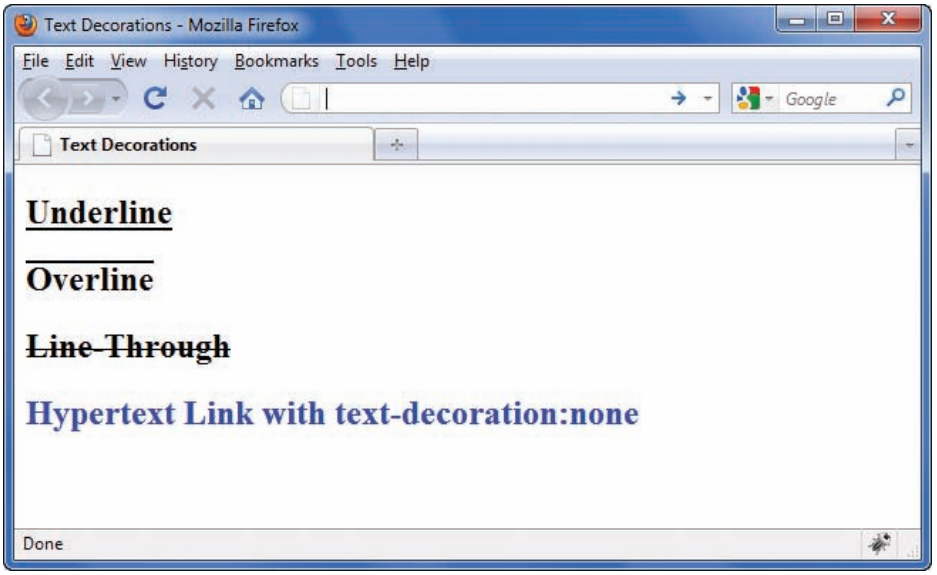

Figure 5-15 Text decorations

As Figure 5-15 shows, the text-decoration property lets you remove the underlining from the <a> element. As you read earlier, the user commonly relies on the underlining text to indicate a hypertext link. However, some Web sites choose to remove link underlining, indicating links with a color different from the standard text color. You can remove the underlining from your anchor elements with the following rule:

```
a {text-decoration: none;}
```

Users with sight disabilities can have trouble finding the links in your content if you choose to remove the underlining. Alternately, a user can override the author's style rules by setting preferences in his or her browser or applying his or her own style sheet.

Specifying Capitalization

The text-transform property lets you change the capitalization of text. This property is very useful for headings anywhere you want to change the capitalization of text from its original capitalization format to a different format without actually editing the text.

text-transform property description	
Value:	capitalize | uppercase | lowercase | none
Initial:	none
Applies to:	all elements
Inherited:	no
Percentages:	N/A

The *capitalize* value capitalizes the first letter of every word. *Uppercase* and *lowercase* transform the case of an entire word. The following code transforms the case of an <h1> element to uppercase.

```
h1 {text-transform: uppercase;}
```

Specifying Text Shadow

text-shadow property description	
Value:	none | [<shadow>,] * <shadow>
Initial:	none
Applies to:	all elements
Inherited:	yes
Percentages:	N/A

The text-shadow property lets you define a shadow that is displayed behind text. You can specify the vertical and horizontal offset as well as the blur of the shadow. This property is ideal for adding depth and character to headings and other important typographic elements on your Web page, although it is best used sparingly. The following code shows the text-shadow property syntax.

```
h1 {text-shadow: 2px 2px 2px #666;}
```

The first two length values indicate the horizontal and vertical offset from direct alignment with the text. Positive values move to the right and down, negative values move to the left and up.

The third length value specifies the blur amount, which determines how soft the edges of the shadow will display. The final value sets the color of the shadow. Figure 5-16 shows examples of different shadow values both with and without blur.

Remember to test the text-shadow property carefully; although if text-shadow is not supported, the result is not too bad, as the text is simply displayed without a shadow.

The text-shadow property is not supported by Internet Explorer 8.

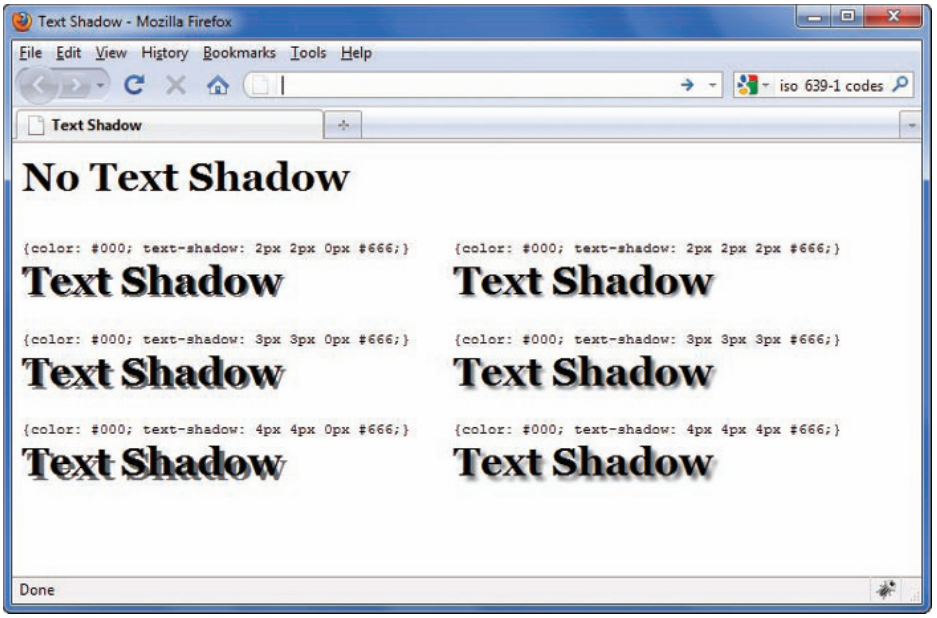

Figure 5-16 Text shadow

Currently Unsupported CSS3 Properties

The CSS3 recommendation contains some properties that are not supported evenly or at all by different browsers and devices. These properties may change, so check the latest CSS3 documentation at *www.w3.org* for the latest updates. As new versions of the browsers are released, these properties should gradually become supported. Table 5-6 contains descriptions of these properties.

Property	Characteristics
white-space Controls how paragraph text wraps, and whether to preserve white space Normally the browser ignores extra spaces between words; you can preserve this space using the *pre* value	Value: normal | pre | nowrap | pre-wrap | pre-line Initial: not defined for shorthand properties Applies to: all elements Inherited: yes Percentages: N/A
text-wrap Controls text wrapping	Value: normal | unrestricted | none | suppress Initial: normal Applies to: all elements Inherited: yes Percentages: N/A

Table 5-6 Currently Unsupported CSS3 Properties *(continues)*

(continued)

Property	Characteristics
word-wrap Controls whether words can be broken to wrap a sentence	Value: normal | break-word Initial: normal Applies to: all elements Inherited: yes Percentages: N/A
text-align-last When text is set to justify, the last line in a paragraph may align unevenly; this property controls the alignment of the last line	Value: start | end | left | right | center | justify Initial: start Applies to: all elements Inherited: yes Percentages: N/A
text-emphasis Intended for Asian languages; adds accent marks above characters for emphasis	Value: none | [[accent | dot | circle | disc] [before | after]?] Initial: none Applies to: all elements Inherited: yes Percentages: N/A
text-outline Specifies a text outline's thickness and blur length	Value: none | [<color> <length> <length>? | <length> <length>? <color>] Initial: none Applies to: all elements Inherited: yes Percentages: N/A
font-stretch The font-stretch property lets you expand or compress the font face; not all forms of the font may be available, so the next closest choice is substituted: the closest condensed face substitutes for any condensed choice, and the closest expanded face for any expanded choice	Value: normal | wider | narrower | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded | inherit Initial: none Applies to: all elements Inherited: yes Percentages: N/A
font-size-adjust For font sizes, the size of the actual characters varies by font—if you are using font substitution, one of the substituted fonts may be smaller and less legible than the primary font you specified; the font-size-adjust property lets you maintain the legibility of your text when substitution occurs. It affects the x-height of the font	Value: <number> | none | inherit Initial: see individual properties Applies to: all elements Inherited: yes Percentages: N/A

Table 5-6 Currently Unsupported CSS3 Properties

Activity: Building a Font and Text Properties Style Sheet

In the following set of steps, you will build a style sheet that uses the typographic techniques you learned about in this chapter. Save your file, and test your work in the browser as you complete each step.

To build the style sheet:

1. Copy the **font activity.html** file from the Chapter05 folder provided with your Data Files to the Chapter05 folder in your work folder. (Create the Chapter05 folder, if necessary.)
2. Open the file **font activity.html** in your HTML editor, and save it in your Chapter05 folder as **font activity1.html**.
3. In your browser, open the file **font activity1.html**. When you open the file, it looks like Figure 5-17.

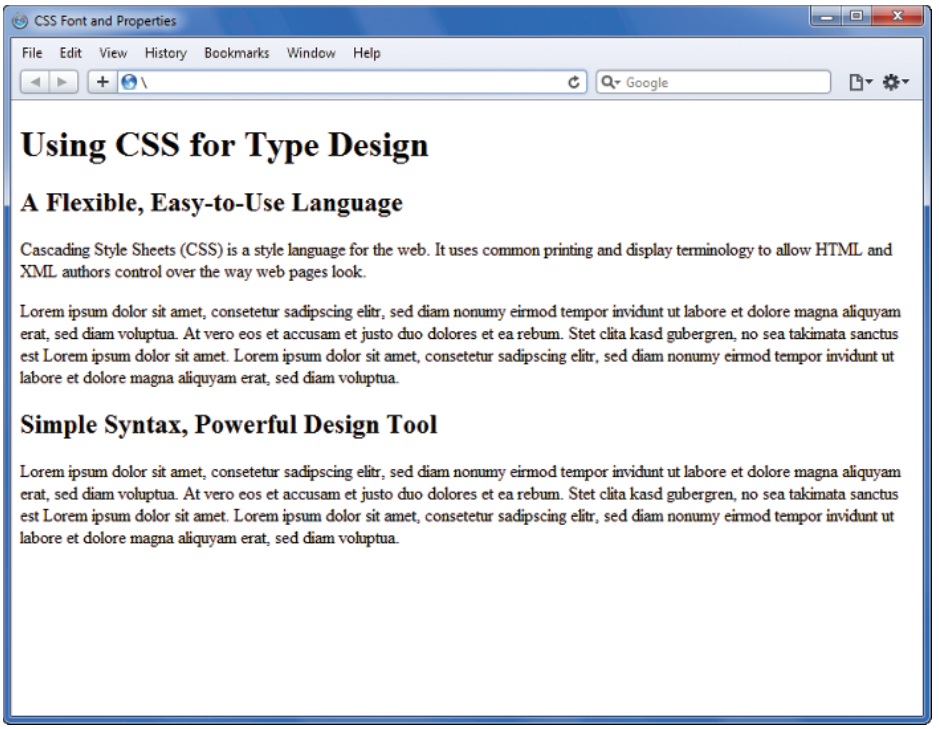

Figure 5-17 Basic HTML document

Adding the <style> Section

Because you are working on a single document, you can use a <style> element in the <head> section to contain your style rules. In the steps throughout the chapter, enter the code shown in bold and blue.

To add the <style> section:

1. Using your text editor, add a <style> element in the <head> section to contain your style rules as shown in the following code. Leave a few lines of white space between the <style> tags to contain the style rules.

```
<head>
<title>CSS Font Activity</title>
<style type="text/css">



</style>
</head>
```

2. Save the file.

Styling the Headings

To style the headings:

1. Write a style rule that selects the <h1> element. Set the font-size to 3em and the font-family to Georgia, with a fallback generic serif font, as shown in the following code:

```
h1 {
   font-size: 3em;
   font-family: georgia, serif;
}
```

2. Write a style rule that selects the <h2>. Set the font-size to 1.5em and the font-family to arial. Add fallback values of helvetica and sans-serif to the font-family declaration.

```
h2 {
   font-size: 1.5em;
   font-family: arial, helvetica, sans-serif;
}
```

3. Save your file and check your work in the browser. Figure 5-18 shows the changes from the style rules you added.

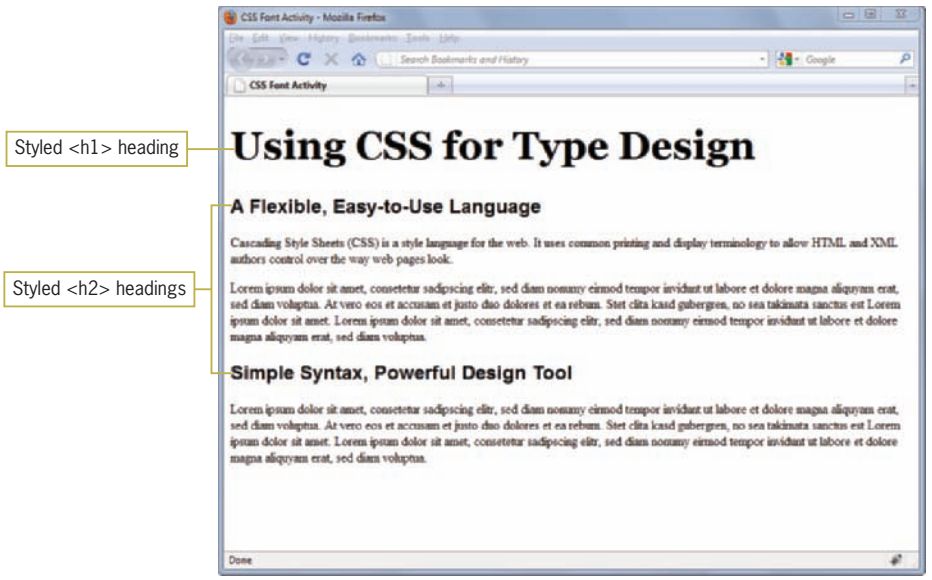

Figure 5-18 Styled headings

Styling the Paragraphs

Write a rule that makes the paragraph text easier to read by changing the font, increasing the white space between lines, and adding a left margin to move the paragraphs away from the edge of the browser window. This rule will also use a class name to apply the style rule to the paragraphs.

To style the paragraphs:

1. Write a style rule that uses a class selector and a class named *copy*. Set the font-family property to *georgia, serif* as you did for the <h1> element. Set the line-height property to 1.5em, and add a margin-left of 20 pixels (20px). You will learn more about margins in Chapter 6.

```
.copy {
   font-family: georgia, serif;
   line-height: 1.5em;
   margin-left: 20px;
}
```

2. Locate the first <p> element within the file and add class="copy" to the opening <p> tag as shown:

```
<p class="copy">Cascading Style Sheets (CSS) is a
style language for the web. It uses common printing
and display terminology to allow HTML and XML authors
control over the way web pages look.</p>
```

3. Repeat Step 2 and add the *class* attribute to the remaining <p> elements. There are three in all.
4. Save your file and check your work in the browser. Figure 5-19 shows the changes from the style rules you added.

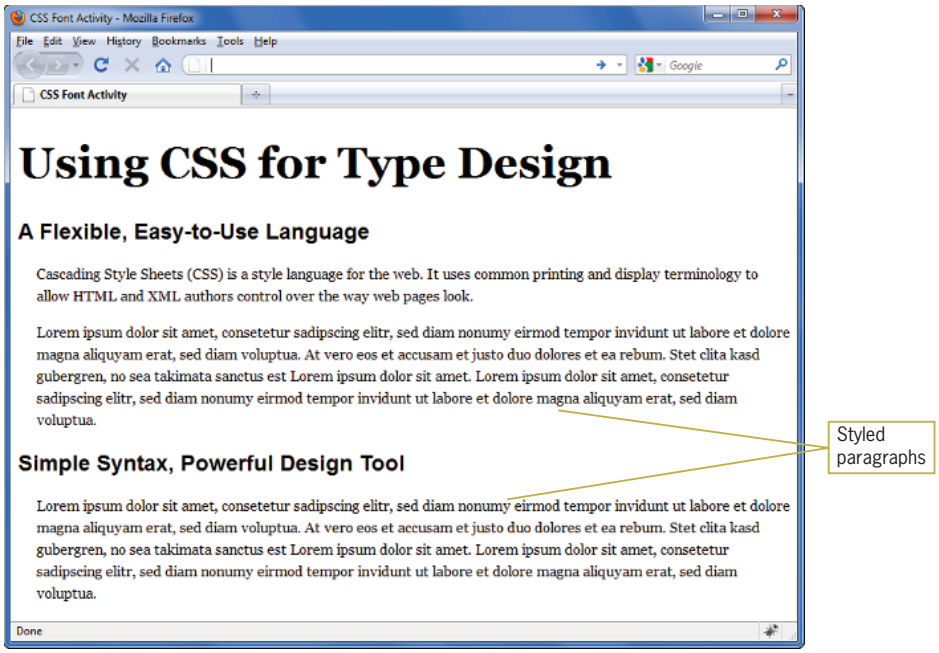

Figure 5-19 Styled paragraphs

Making the Second-Level Headings More Distinctive

The page is looking better with more legible text, but the second-level headings could stand out more. To do this, you will add a left margin and letter-spacing to the existing style rule.

To add styles to the <h2> elements:

1. Include a style rule in the existing h2 selector to add a left margin of 20 pixels to align the headings with the paragraphs.

```
h2 {
   font-size: 1.5em;
   font-family: arial, helvetica, sans-serif;
   margin-left: 20px;
}
```

2. Add one more style rule to set the letter-spacing to 4px.

```
h2 {
   font-size: 1.5em;
   font-family: arial, helvetica, sans-serif;
   margin-left: 20px;
   letter-spacing: 4px;
}
```

3. Save the file and view your changes in the browser. It should look similar to Figure 5-20.

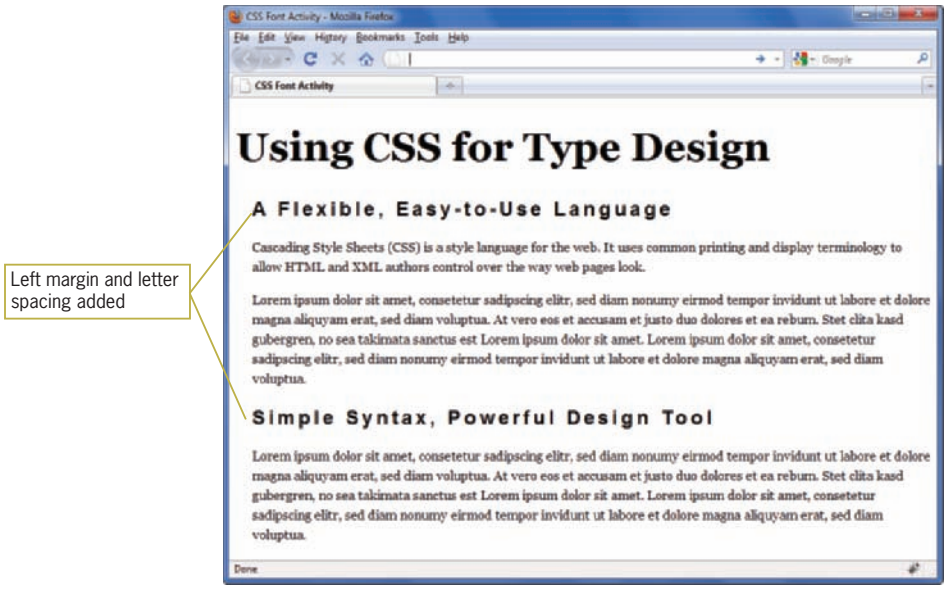

Figure 5-20 New rules added to the <h2> elements

Capitalizing Key Words

You can use the text-transform property along with a element and class selector to select the first few words in the first paragraph and capitalize them, a common printing technique that adds a professional look to the page.

To select and capitalize the words:

1. Write a style rule that uses a class selector for a class named *caps*. Add the text-transform property set to uppercase.

```
.caps {text-transform: uppercase;}
```

2. Add another declaration to the rule, specifying a bold font-weight.

```
.caps {
   text-transform: uppercase;
   font-weight: bold;
}
```

3. Locate the text "Cascading Style Sheets (CSS)" in the first paragraph, and place span tags before and after this text.

```
<p class="copy"><span>Cascading Style Sheets (CSS)
</span> is a style language for the web. It uses common
printing and display terminology to allow HTML and XML
authors control over the way web pages look.</p>
```

4. Add the class attribute to the opening tag and set the value to *caps*.

```
<p class="copy"><span class="caps">Cascading Style
Sheets (CSS)</span> is a style language for the web.
It uses common printing and display terminology to
allow HTML and XML authors control over the way web
pages look.</p>
```

5. Save the file and view your changes in the browser. Figure 5-21 shows the result of the style rule.

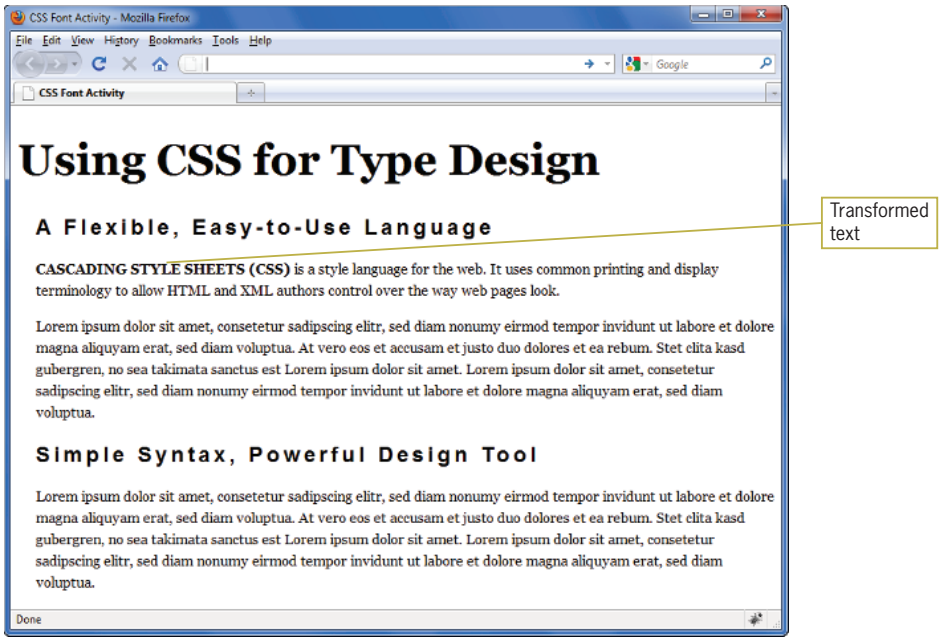

Figure 5-21 Transforming text to uppercase

Customizing Bulleted and Numbered Lists

The list-style properties let you control the visual characteristics of elements that have a display property value of *list-item*, which are the numbered and bulleted lists signified by the <ol> and <ul> elements. These properties let you set the appearance of the marker that indicates each item within the list. With CSS, the marker can be a symbol, number, or image. You can also determine the position of the marker next to the list content.

The two common elements that have a default display value of *list-item* are <ul> and <ol>, which generate a bulleted (unordered) and ordered list, respectively. The following code shows a sample of each type of list:

```
<!-- Bulleted List -->
<h3>Things to do...</h3>
  <ul>
   <li>Buy dog food</li>
   <li>Clean up the house</li>
   <li>Take a rest</li>
  </ul>
<!-- Ordered List -->
<h3>Places to go...</h3>
  <ol>
   <li>Paris</li>
   <li>Sydney</li>
   <li>Cairo</li>
  </ol>
```

Figure 5-22 shows the result of this code. Notice that the default markers are a solid bullet, called a "disc" for the <ul> list, and an Arabic numeral called "decimal" for the <ol> list. You can change these marker values with the CSS list-style properties.

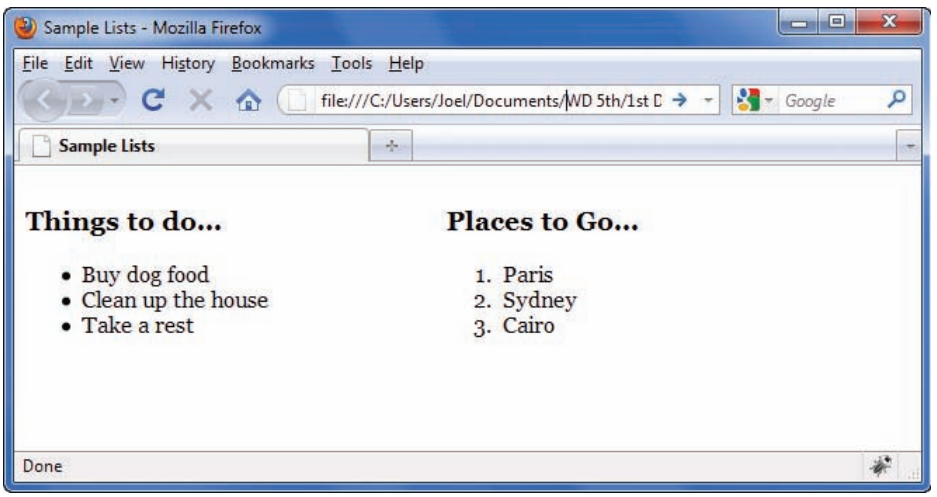

Figure 5-22 Unordered and ordered list elements

Specifying the list-style-type Property

The list-style-type property lets you customize the list marker to a variety of different values.

list-style-type

Value:	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
Initial:	disc
Applies to:	elements with 'display: list-item'
Inherited:	yes
Percentages:	N/A

The list-style-type property lets you specify one of three types of markers for a list. You can choose a symbol, a numbering system, or an alphabetical system. CSS allows a wide variety of marker values. Remember to test support for these markers in different browsers. Tables 5-7, 5-8, and 5-9 list the different values and their descriptions.

Value	Description
disc	Filled circle (see Figure 5-23)
circle	Hollow circle (see Figure 5-23)
square	Filled square (see Figure 5-23)

Table 5-7 Bulleted List Values

Value	Description
decimal	Decimal numbers, beginning with 1; this is the default numbering
decimal-leading-zero	Decimal numbers padded by initial zeros (01, 02, 03...)
lower-roman	Lowercase roman numerals (i, ii, iii...)
upper-roman	Uppercase roman numerals (I, II, III...)
hebrew	Traditional Hebrew numbering
georgian	Traditional Georgian numbering
armenian	Traditional Armenian numbering
cjk-ideographic	Plain ideographic numbers
hiragana	Japanese hiragana language characters
katakana	Japanese katakana language characters
hiragana-iroha	Japanese hiragana-iroha language characters
katakana-iroha	Japanese katakana-iroha language characters

Table 5-8 Numerical List Values

Value	Description
lower-alpha	Lowercase ASCII letters (a, b, c, ... z)
upper-alpha	Uppercase ASCII letters (A, B, C, ... Z)
lower-greek	Lowercase classical Greek

Table 5-9 Alphabetical list values

Figure 5-23 shows the list types that most browsers support.

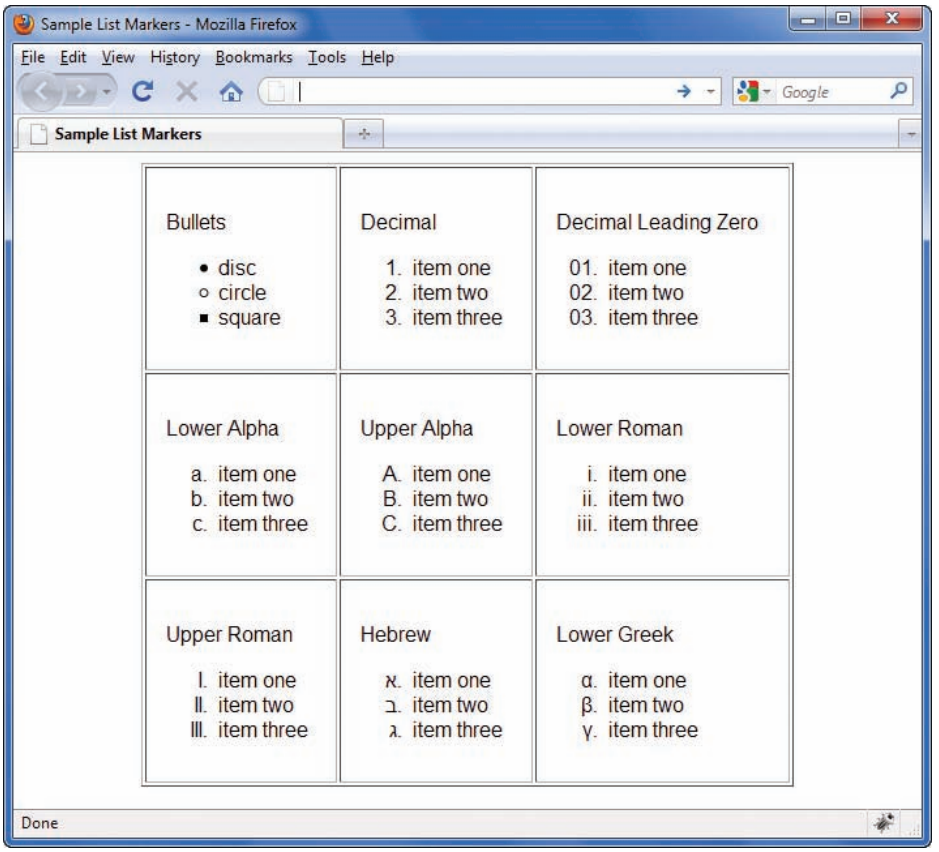

Figure 5-23 Different list marker types

To specify a list-style marker type, select the list container element, either ul or ol, and specify the value as shown in the following style rules:

```
ol {list-style-type: decimal-leading-zero;}
ul {list-style-type: circle;}
```

There may be times when you want to specify a list-style-type within an individual list. You can do this by using the *style* attribute within the <ul> or <ol> element as shown in the following:

```
<ol style="list-style-type: lower-alpha;">
<li>Item One</li>
<li>Item Two</li>
<li>Item Three</li>
</ol>
```

This style rule affects only this one instance of the list.

Specifying the list-style-image Property

The list-style-image property lets you easily attach an image to a list and have it repeated as the marker symbol.

list-style-image	
Value:	<url> | none | inherit
Initial:	none
Applies to:	elements with 'display: list-item'
Inherited:	yes
Percentages:	N/A
Media:	visual

The list-style-image property lets you replace the standard symbol with an image of your choice. The following code shows the style rule that attaches an image to a bulleted list:

```
ul {list-style-image: url(pawprint.gif);}
```

Figure 5-24 shows the result of the style rule. The image is repeated whenever a list element is used.

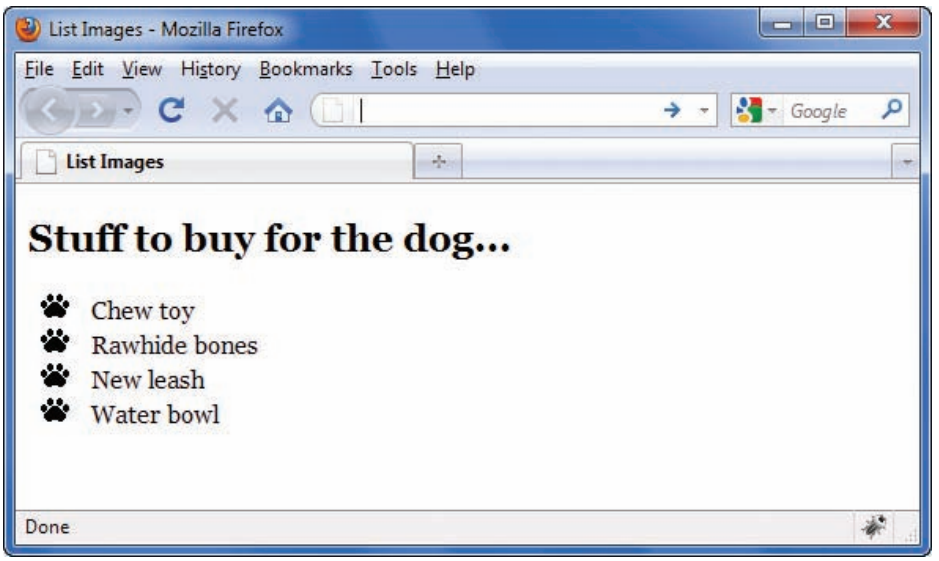

Figure 5-24 Attaching an image as a list marker

Specifying the list-style-position Property

The list-style-position property lets you determine the placement of the list marker, either inside or outside the list-item content box.

list-style-position	
Value:	inside | outside | inherit
Initial:	outside
Applies to:	elements with 'display: list-item'
Inherited:	yes
Percentages:	N/A
Media:	visual

The default value is outside. Figure 5-25 shows the two types of list position values.

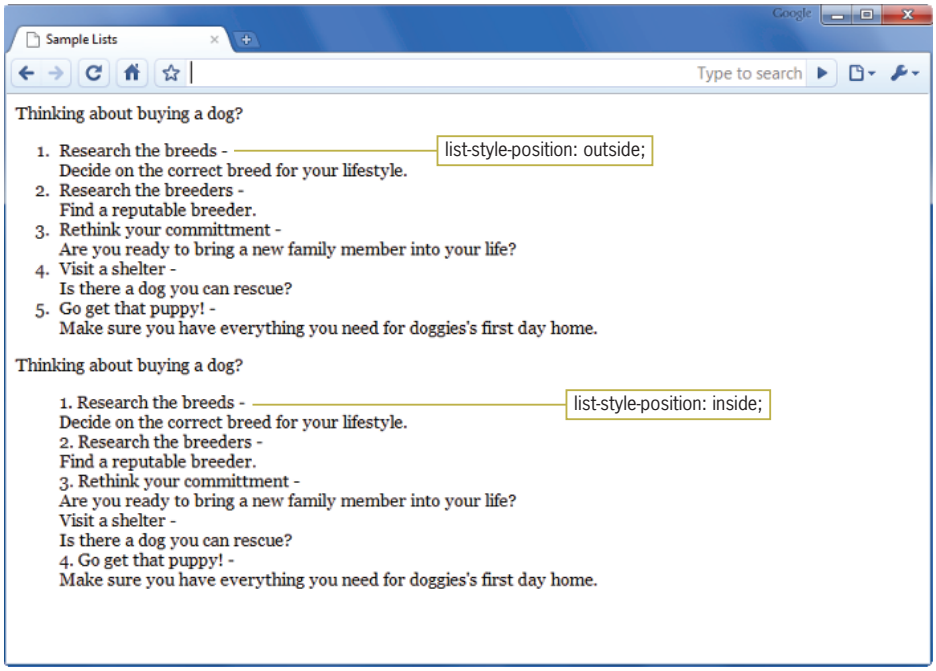

Figure 5-25 Positioning the list marker

This figure shows two <ol> lists. The style rules for this page use a class selector to differentiate between the two:

```
ol.outside {list-style-position: outside;}
ol.inside {list-style-position: inside;}
```

The class is then applied to each <ol> element, as shown in the following code fragment for the first list:

```
<ol class="outside">
```

The class is applied to the second list in the same way:

```
<ol class="inside">
```

Using the list-style Shorthand Property

Like the font shorthand property, the list-style property lets you write a single rule to specify all of the list-item properties.

list-style shorthand property description	
Value:	[<list-style-type> || <list-style-position> || <list-style-image>] | inherit
Initial:	not defined for shorthand properties
Applies to:	elements with 'display: list-item'
Inherited:	yes
Percentages:	N/A
Media:	visual

The list-style shortcut property lets you state the following list-style properties in one concise style rule:

- List-style-type
- List-style-image
- List-style-position

You can specify values in any order. In the following style rule, the list is set to lowercase alphabetical letters that are inside the list box:

```
ol {list-style: lower-alpha inside;}
```

Chapter Summary

You can use Cascading Style Sheets to manipulate a variety of text properties and achieve professional-quality typography on your Web site. Keep the following points in mind:

- Use type to communicate information structure. Be sparing with your type choices; use fonts consistently and design for legibility.
- Remember that HTML text downloads faster than graphics-based text. Use HTML text whenever possible.
- Use fonts that appear as consistently as possible across operating systems.
- Standardize your styles by building external style sheets and linking them to multiple documents.
- Test your work. Different browsers and computing platforms render text in different sizes.
- Use type effectively by choosing available fonts and sizes. Design for legibility and use text to communicate information about the structure of your material.

- Choose the correct measurement unit based on the destination medium. For the computer screen, ems or percentage measurements can scale the text to the user's preferences.
- Use font properties to control the look of your letter forms. Specify font-substitution values to ensure that your text is displayed properly across different platforms.
- Use the text spacing properties to create more visually interesting and legible text.

Key Terms

cursive—A generic value for the CSS font-family property. Cursive fonts are designed to resemble handwriting. Most browsers do not support this font family.

em unit—In CSS, a unit equal to the font size of an element.

ex unit—In CSS, a unit equal to the height of the lowercase letter *x* in any given font.

fantasy—A generic value for the CSS font-family property. Fantasy fonts are primarily decorative. Most browsers do not support this font family.

font—A typeface in a particular size, such as Times Roman 24 point.

font property—In CSS, a shortcut that lets you specify the most common font properties in a single statement.

leading—The white space between lines of text.

monospace—A generic value for the CSS font-family property. Monospace fonts are fixed-width fonts. Every letter has the same horizontal width.

sans-serif—A generic value for the CSS font-family property. Sans-serif fonts have no serifs. The most common sans-serif fonts are Helvetica and Arial.

serif—A generic value for the CSS font-family property. Serif is the traditional printing letter form, with strokes (or serifs) that finish off the top and bottom of the letter. The most common serif fonts on the Web are Times and Times Roman.

text property—A CSS property that lets you adjust the spacing around and within your text.

typeface—The name of a type family, such as Times Roman or Futura Condensed.

x-height—The height of the letter *x* in a particular font.

Review Questions And Exercises

1. What is the default browser font?
2. What does the browser do if you specify a font that is not stored on a user's computer?
3. What are two drawbacks to the use of graphics-based text?
4. What are the three types of CSS measurement units?
5. What is the best destination for absolute units of measurement?
6. Why would you use relative or percentage values for a Web page?
7. What is the size of the em?
8. What determines the size of a pixel?
9. What is the advantage of the generic font families?
10. Write a font-family substitution string that selects Arial, Helvetica, or any sans-serif font for a <p> element.
11. Write a style rule for an <h2> element that specifies bold text that is twice the size of the default font size.
12. Write a rule specifying that <p> elements appear as 1.5em text with 2em leading.
13. Write a rule specifying that elements are displayed in red only when they appear within <p> elements.
14. Write a rule defining a division named *note*. Specify 12-point bold Arial text on a yellow background.

15. What three typographic white-space areas can you affect with style rules?

16. Write a style rule for a <p> element with a 25-point hanging indent and a 30-pixel margin on the left and right sides.

17. Rewrite the following rule using the font shortcut property:

```
blockquote {font-style: italic; font-size: 1.2em;
line-height: 1.75em; font-family: times, serif;}
```

18. What is a benefit of increasing the standard text line height?

19. What is the size of the text indent and line height relative to the user's default font size in the following style rule?

```
p {text-indent: 3em; line-height: 150%;}
```

Hands-On Projects

1. In the following set of steps, you will learn how to style list-item elements with the list-style properties. As you work through the exercise, refer to Figure 5-27 to see the results you will achieve. Save your file and test your work in the browser as you complete each step.

   To apply the list-style properties:

   a. Open the file **lists.html** in your HTML editor, and save it in your work folder as **lists1.html**.

   b. Copy the image file **diamond.gif** into your work folder.

   c. In your browser, open the file **lists1.html**. When you open the file, it looks like Figure 5-26. Notice that the file contains three lists. You will apply a different list-style to each list.

   d. The first list on the page is a bulleted list that currently displays the default disc (bullet) style. Write a style rule that uses a class selector *circle* to uniquely select the list. Set the list-style-type property to change the bullet style to *circle*.

      ```
      ul.circle {list-style-type: circle;}
      ```

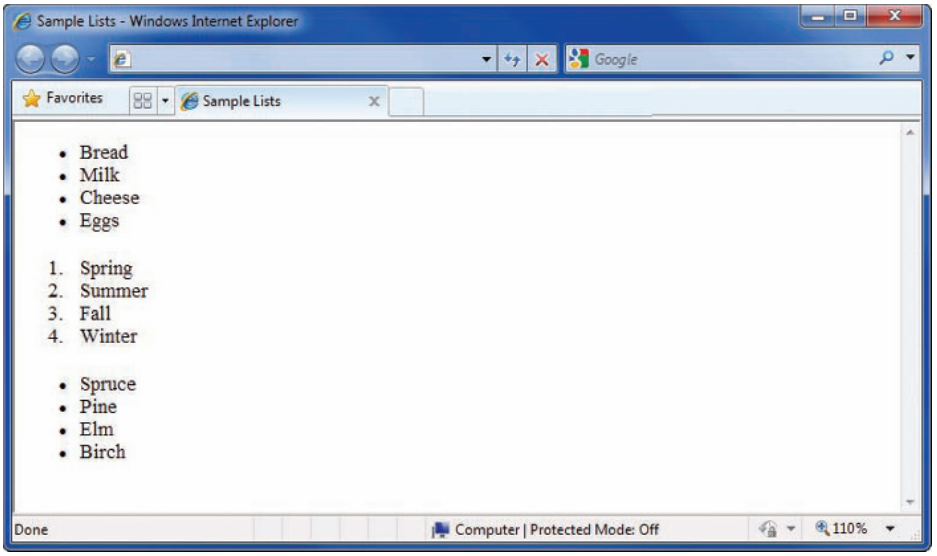

Figure 5-26 Beginning Web page

e. Now apply the style to the first list by adding the *class* attribute to the <ul> element.

```
<!-- Bulleted List -->
<ul class="circle">
<li>Bread</li>
<li>Milk</li>
<li>Cheese</li>
<li>Eggs</li>
</ul>
```

f. The second list on the page is an ordered list that currently displays the default decimal style. Write a style rule that uses a class selector *alpha* to uniquely select the list. Set the list-style-type property to change the style to upper-alpha.

```
ol.alpha {list-style-type: upper-alpha;}
```

g. Now apply the style to the first list by adding the *class* attribute to the <ol> element.

```
<!-- Alphabetical List -->
<ol class="alpha">
<li>Spring</li>
<li>Summer</li>
<li>Fall</li>
<li>Winter</li>
</ol>
```

h. The third list on the page is an unordered list that currently displays the default bullet style. Write a style rule that uses a class selector *image* to uniquely select the list. Set the list-style-image property to a URL value, using the image file diamond.gif.

```
ul.image {list-style-image: url(diamond.gif);}
```

i. Now apply the style to the first list by adding the *class* attribute to the <ol> element. Figure 5-27 shows the finished document.

```
<!-- List Image -->
<ul class="image">
<li>Spruce</li>
<li>Pine</li>
<li>Elm</li>
<li>Birch</li>
</ul>
```

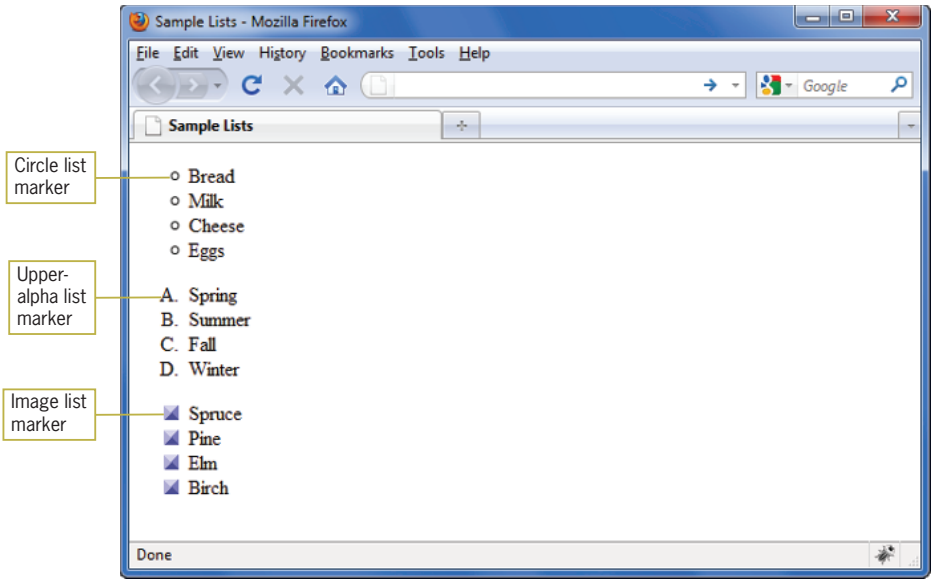

Figure 5-27 Completed Web page in Firefox

2. Modify an existing HTML document to use Cascading Style Sheets.

   a. Build styles using the existing standard HTML elements in the file.

   b. Test the work in multiple browsers to verify that all styles are portable.

c. Remove the files and place them in an external style sheet.

d. Link the HTML file to the style sheet. Test to make sure the file is displayed properly.

3. Browse the Web for examples of good typography. Write a short design critique of why the type works effectively on the Web sites you find. Save and print screen shots of the sample Web pages to accompany your critique.

4. Browse the Web for examples of poor typography. Write a short design critique of why the type is confusing or misleading to the user. Save and print screen shots of the sample Web pages to accompany your critique.

5. In this project, you have a chance to test the font and text properties on paragraphs of text. Save and view the file in your browser after completing each step.

   a. Using your HTML editor, create a simple HTML file (or open an existing file) that contains multiple <p> elements and so on. Save the file in your Chapter05 folder as **fonts1.html**.

   b. Add a <style> element to the <head> section, as shown in the following code:

```
<head>
<title>CSS Test Document</title>
<style type="text/css">

</style>
</head>
```

   c. Write a style rule that uses p as a selector and sets the font-family to a sans-serif font. You can use a generic font family, or choose one of the fonts available on your computer.

   d. Specify a list of alternate fonts to ensure that your font choice is displayed properly across a range of computers.

   e. Specify a text indent for the <p> elements. Use the em value as the measurement unit.

   f. Add the line-height property to the style rule. Experiment with different line heights until you find one that you feel enhances the legibility of the paragraph text.

Individual Case Project

Design the type hierarchy for your Case Project Web site. Create a type specification HTML page that shows examples of the different typefaces and sizes and how they will be used. This can be a mock-up page that uses generic content but demonstrates the overall typographic scheme. Consider the following questions:

- What will be the typefaces and styles for the body type and headings?
- How many levels of headings are necessary?
- What are the different weights and sizes of the headings?
- How will text be emphasized?
- Will hypertext links be standard or custom colors?
- How will you ensure the legibility and readability of your text?
- What will your line length be?

Team Case Project

Meet as a team and discuss the type hierarchy and options for your site. Each team member should bring a type specification HTML page with his or her ideas and examples of the different typefaces and sizes and how they can be used. This can be a mock-up page that uses generic content but demonstrates the overall typographic scheme. Consider the following questions:

- What will be the typefaces and styles for the body type and headings?
- How many levels of headings are necessary?
- What are the different weights and sizes of the headings?
- How will text be emphasized?
- Will hypertext links be standard or custom colors?
- How will you ensure the legibility and readability of your text?
- What will your line length be?