To make text bold:

```
font-weight: bold;
```

And to make text un-bold:

```
font-weight: normal;
```

**NOTE** Since headlines are already displayed as bold type, you may want to find another way of highlighting a word or words that are strongly emphasized or bolded inside a headline. Here's one way:

```
h1 strong { color: #3399FF; }
```

This descendent selector changes the color of any `<strong>` tags (usually displayed as bold) that appear inside a `<h1>` tag.

## Capitalizing

Capitalizing text is pretty easy—just hit the caps lock key and start typing, right? But what if you want to capitalize every heading on a page, and the text you've copied and pasted from a Word document is lowercase? Rather than retyping the headline, turn to the CSS `text-transform` property. With it, you can make text all uppercase, all lowercase, or even capitalize the first letter of each word (for titles and headlines). Here's an example:

```
text-transform: uppercase;
```

For the other two options, just use `lowercase` or `capitalize`.

Because this property is inherited, a tag that's nested inside a tag with `text-transform` applied to it gets the same uppercase, lowercase, or capitalized value. To tell CSS *not* to change the case of text, use the `none` value:

```
text-transform: none;
```

### SMALL CAPS

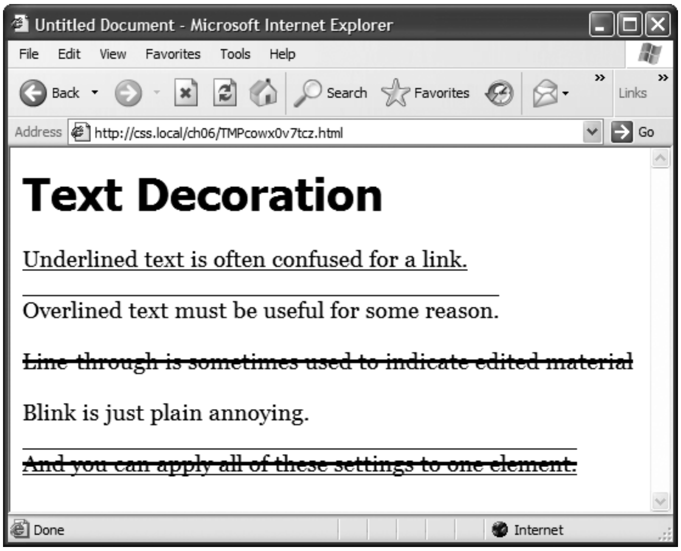
For more typographic sophistication, you can also turn to the `font-variant` property, which lets you set type as `small-caps`. In small cap style, lowercase letters appear as slightly downsized capital letters, like so: POMP AND CIRCUMSTANCE. While difficult to read for long stretches of text, small caps lend your page an old-world, bookish gravitas when used on headlines and captions. To create small-cap text:

```
font-variant: small-caps;
```

## Decorating

CSS also provides the `text-decoration` property to add various enhancements to text. With it, you can add lines over, under, or through the text (see Figure 6-12), or for real giggles, you can make the text blink like a No Vacancy sign. Use the `text-decoration` property by adding one or more of the following keywords: `underline`, `overline`, `line-through`, or `blink`. For example, to underline text:

```
text-decoration: underline;
```


Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Links
Address http://css.local/ch06/TMPcowx0v7tcz.html Go
Done Internet

Text Decoration

Underlined text is often confused for a link.

Overlined text must be useful for some reason.

~~Line-through is sometimes used to indicate edited material~~

Blink is just plain annoying.

~~And you can apply all of these settings to one element.~~

**FIGURE 6-12**

*The* `text-decoration` *property in action. If this is what the people at CSS headquarters call "decorations," you'd best not ask for their design help on your next home remodel.*

You can also combine multiple keywords for multiple effects. Here's how to add a line over and under some text:

```
text-decoration: underline overline;
```

But just because you *can* add these not-so-decorative decorations to text, doesn't mean you should. For one thing, anyone who's used the Web for any length of time instinctively associates any underlined text with a link and tries to click it. So it's not a good idea to underline words that aren't part of a link. And `blink` is like a neon sign flashing "Amateur! Amateur! Amateur!" (That's probably why most browsers don't make text blink even if you ask for it.)

**NOTE** You can get a similar effect to underlining and overlining by adding a border to the bottom or top of an element (see page 298). The big advantage of borders is that you can control their placement, size, and color to create a more attractive design that doesn't look like a link.

The `overline` option simply draws a line above text, while `line-through` draws a line right through the center of text. Some designers use this strike-through effect to indicate an edit on a page where text has been removed from the original manuscript.

Finally, you can turn off all decorations by using the none keyword like this:

```
text-decoration: none;
```

Why do you need a text-decoration property that removes decorations? The most common example is removing the line that appears under a link. (See page 297.)

## Letter and Word Spacing

Another way to make text stand out from the crowd is to adjust the space that appears between letters or words (see Figure 6-13). Reducing the space between letters using the CSS `letter-spacing` property can tighten up headlines, making them seem even bolder and heavier while fitting more letters on a single line. Conversely, increasing the space can give headlines a calmer, more majestic quality. To reduce the space between letters, you use a negative value like this:

```
letter-spacing: -1px;
```

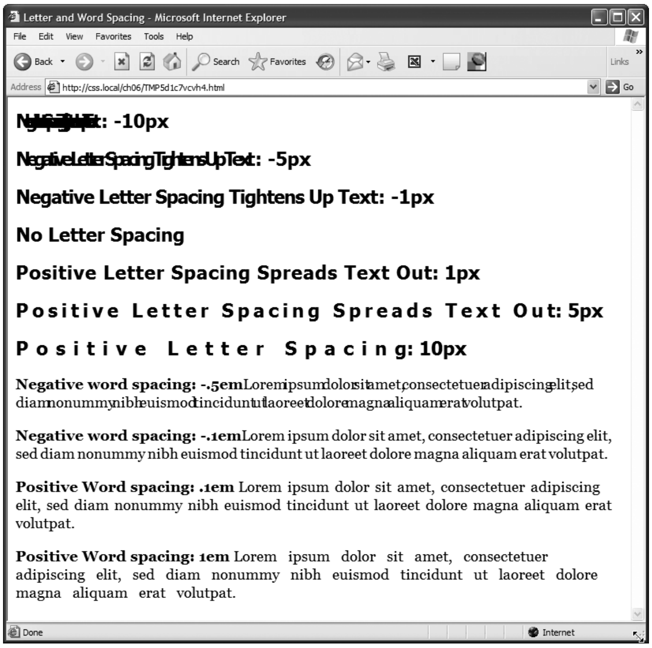
A positive value adds space between letters:

```
letter-spacing: .7em;
```

Likewise, you can open up space (or remove space) between words using the `word-spacing` property. This property makes the space between words wider (or narrower) without actually affecting the spacing between the letters inside a word:

```
word-spacing: 2px;
```

With either of these properties, you can use any type of measurement you'd use for text sizing—pixels, ems, percentages—with either positive or negative values.


Letter and Word Spacing - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Links
Address http://css.local/ch06/TMP5d1c7vcvh4.html Go
Done Internet

**Ngt: -10px**

**NegativeLetterSpacingTightensUpText: -5px**

**Negative Letter Spacing Tightens Up Text: -1px**

**No Letter Spacing**

**Positive Letter Spacing Spreads Text Out: 1px**

**P o s i t i v e   L e t t e r   S p a c i n g   S p r e a d s   T e x t   O u t: 5px**

**P o s i t i v e   L e t t e r   S p a c i n g: 10px**

**Negative word spacing: -.5em**Loremipsumdolorsitamet,consectetueradipiscingelit,sed diamnonummynibheuismodtinciduntutlaoreetdoloremagnaaliquameratvolutpat.

**Negative word spacing: -.1em**Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

**Positive Word spacing: .1em** Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

**Positive Word spacing: 1em** Lorem  ipsum  dolor  sit  amet,  consectetuer adipiscing  elit,  sed  diam  nonummy  nibh  euismod  tincidunt  ut  laoreet  dolore magna  aliquam  erat  volutpat.

**FIGURE 6-13**

*Use word and letter spacing judiciously. Too much or too little of either can make text difficult, if not impossible, to read.*

Unless you're going for some really far-out design effect—in other words, totally unreadable text—keep your values small. Too high a negative value, and letters and words overlap. To keep the message of your site clear and legible, use both letter and word spacing with care.

## Adding Text Shadow

CSS3 includes one property that lets you add drop shadows to text to add depth and interest to headlines, lists, and paragraphs (see Figure 6-14).

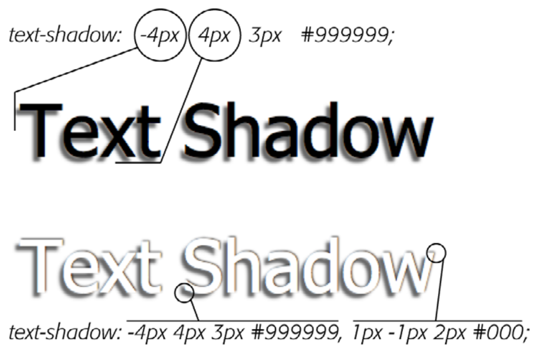

text-shadow: -4px 4px 3px #999999;
Text Shadow
Text Shadow
text-shadow: -4px 4px 3px #999999, 1px -1px 2px #000;

**FIGURE 6-14**

*Text shadows are a great way to add subtle (or, if you insist, not so subtle) depth to headlines and other text. However, the* `text-shadow` *property doesn't work in Internet Explorer 9 or earlier.*

The `text-shadow` property requires four pieces of information: the horizontal offset (how far to the left or right of the text the shadow should appear), the vertical offset (how far above or below the text the shadow should appear), the blurriness of the shadow, and the color of the drop shadow. For example, here's the `text-shadow` property that creates the effect at the top of Figure 6-14:

```
text-shadow: -4px 4px 3px #999999;
```

The first value—`-4px`—means "place the shadow 4 pixels to the left of the text." (A positive value here would place the shadow to the right of the text.) The second value—`4px`—places the shadow 4 pixels below the text. (A negative value would place the shadow above the text.) The `3px` value defines how blurry the shadow should be. A `0px` value (no blur) results in a sharp drop shadow; the larger the value, the more blurry and indistinct the shadow. Finally, the last value is the drop shadow's color.

You can even add multiple drop shadows for more complex effects (see the bottom image in Figure 6-14): just add a comma followed by additional drop shadow values, like this:

```
text-shadow: -4px 4px 3px #666, 1px -1px 2px #000;
```

There's no limit (except good taste) to the number of shadows you can add this way. Sadly, this effect doesn't work in Internet Explorer 9 or earlier. It does, however, work in all other current browsers (even IE 10). In other words, *don't* rely on this effect to make text readable. The bottom image in Figure 6-14 shows you what not to do: The text color is white and it's readable only because the drop shadows define the outline of the text. In Internet Explorer 9 and earlier, the text would be invisible—white text on a white background.

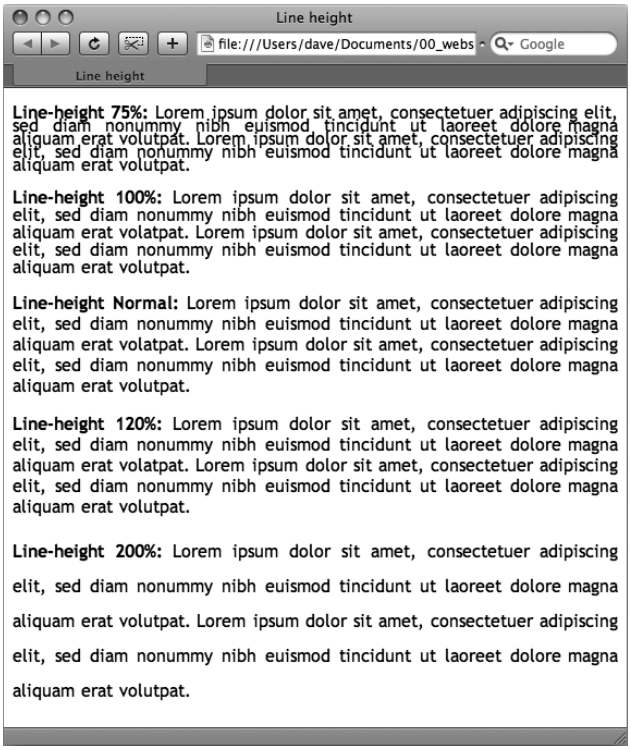
**NOTE** For some examples of beautiful ways to use text shadows, visit *http://webexpedition18.com/articles/css3-text-shadow-property/*.

## Formatting Entire Paragraphs

Some CSS properties apply to chunks of text rather than individual words. You can use the properties in this section on complete paragraphs, headlines, and so on.

### Adjusting the Space Between Lines

In addition to changing the space between words and letters, CSS lets you adjust the space between lines of text using the `line-height` property. The bigger the line height, the more space that appears between each line of text (see Figure 6-15).



**Line-height 75%:** Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

**Line-height 100%:** Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volatpat. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

**Line-height Normal:** Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volatpat. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

**Line-height 120%:** Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volatpat. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

**Line-height 200%:** Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

**FIGURE 6-15**

*The* `line-height` *property lets you spread a paragraph's lines apart or bring them closer together. The normal setting is equivalent to 120 percent, so a smaller percentage tightens up the lines (top), while a larger percentage pushes them apart (bottom).*

### LINE SPACING BY PIXEL, EM, OR PERCENTAGE

Just as with the `font-size` property, you can use pixels, ems, or percentages to set the size of line height:

```
line-height: 150%;
```

In general, percentages or ems are better than pixels, because they change according to, and along with, the text's `font-size` property. If you set the line height to 10 pixels and then later adjust the font size to something much larger (like 36 pixels), because the line height remains at 10 pixels, your lines then overlap. However, using a percentage (150% percent, say) means the `line-height` spacing adjusts proportionally whenever you change the `font-size` value.

The normal `line-height` setting for a browser is 120%. So, when you want to tighten up the line spacing, use a value less than 120%; and to spread lines apart, use a value greater than that.

**NOTE** To determine the amount of space that appears between lines of text, a web browser subtracts the font size from the line height. The result—called *leading*—is the amount of space between lines in a paragraph. Say the font size is 12 pixels, and the line height (set to 150%) works out to 18 pixels. 18 - 12 = 6 pixels, so the browser adds 6 pixels of space between each line.

### LINE SPACING BY NUMBER

CSS offers one other measurement method specific to line height, which is simply a number. You write it like this:

```
line-height: 1.5;
```

There's no unit (like em or px) after this value. The browser multiplies this number by the font size to determine the line height. So if the text is 1em and the `line-height` value is 1.5, then the calculated line height is 1.5em. In most cases, the effect is no different from specifying a value of 1.5em or 150%. But sometimes this multiplication factor comes in handy, especially since nested tags inherit the `line-height` value of their parents.

For example, say you set the `line-height` property of the `<body>` tag to 150%. All tags inside the page would inherit that value. However, it's not the percentage that's inherited; it's the *calculated* line height. So, say the font size for the page is set to 10 pixels; 150 percent of 10 is 15 pixels. Every tag would inherit a line height of 15 pixels, not 150 percent. So if you happened to have a paragraph with large, 36 pixel text, then its line height—15 pixels—would be much smaller than the text, making the lines squish together in a hard-to-read mess.

In this example, instead of using a `line-height` of 150% applied to the `<body>` tag, you could have all tags share the same basic proportional line height by setting the `line-height` to `1.5`. Every tag, instead of inheriting a precise pixel value for line height from the body style, simply multiplies its font size by 1.5. So in the above example of a paragraph with 36-pixel text, the line height would be 1.5 x 36 or 54 pixels.

## Aligning Text

One of the quickest ways to change the look of a web page is with paragraph alignment. Using the `text-align` property, you can center a paragraph on a page, align the text along its left or right edge, or justify both left and right edges (like the paragraphs in this book). Normally, text on a page is left aligned, but you may want to center headlines to give them a formal look. Languages that read from right to left, like Hebrew and Arabic, require right-alignment. To change the alignment of text, use any of the following keywords—`left`, `right`, `justify`, `center`:

```
text-align: center;
```

Justified text looks great on a printed page—mainly because the fine resolution possible with printing allows for small adjustments in spacing, and because most programs used to lay out printed material can hyphenate long words (thus attempting to equally distribute the number of characters per line). This prevents large, unsightly gaps or rivers of white space flowing through the paragraphs. Web pages are limited to much coarser spacing because of the generally low resolution of monitors, and because web browsers don't know how to hyphenate long words. So when you use the `justify` option, the space between words can vary significantly from line to line, making the text harder to read. When you want to use the `justify` option on your web pages, test it thoroughly to make sure the text is attractive and readable.

## Indenting the First Line and Removing Margins

In many books, the first line of each paragraph is indented. This first-line indent marks the beginning of a paragraph when there are no spaces separating paragraphs. On the Web, however, paragraphs don't have indents but are instead separated by a bit of space—like the paragraphs in this book.

If you have a hankering to make your web pages look less like other web pages and more like a handsomely printed book, take advantage of the CSS `text-indent` and `margin` properties. With them, you can add a first-line indent and remove (or increase) the margins that appear at the beginnings and ends of paragraphs.

### FIRST-LINE INDENTS

You can use pixel and em values to set the first-line indent like this:

```
text-indent: 25px;
```

or

```
text-indent: 5em;
```

A pixel value is an absolute measurement—a precise number of pixels—while an em value specifies the number of letters (based on the current font size) you want to indent.

**TIP** You can use negative `text-indent` values to create what's called a *hanging indent*, where the first line starts further to the left than the other lines in the paragraph. (Think of it as "hanging" off the left edge.) You'll usually use a negative text indent along with a position margin value, so that the negative text indent doesn't stick outside the left side of a page, column, or layout box.

You can also use a percentage value, but with the `text-indent` property, percentages take on a different meaning than you've seen before. In this case, percentages aren't related to the font size; they're related to the width of the element containing the paragraph. For example, if the `text-indent` is set to 50%, and a paragraph spans the entire width of the web browser window, then the first line of the paragraph starts half the way across the screen. If you resize the window, both the width of the paragraph and its indent change. (You'll learn more about percentages and how they work with the width of elements in the next section.)

**POWER USERS' CLINIC**

## A Shorthand Method for Text Formatting

Writing one text property after another gets tiring, especially when you want to use several different text properties at once. Fortunately, CSS offers a shorthand property called `font`, which lets you combine the following properties into a single line: `font-style` (page 162), `font-variant` (page 163), `font-weight` (page 162), `font-size` (page 37), `line-height` (page 167), and `font-family` (page 127). For example, consider the following declaration:

```
 font: italic bold small-caps 18px/150%
Arial, Helvetica, sans-serif;
```

It creates bold, italicized type in small caps, using 18px Arial (or Helvetica or sans-serif) with a line height of 150 percent. Keep these rules in mind:

- You don't have to include every one of these properties, but you *must* include the font size and font family:

  `font: 1.5em Georgia, Times, serif;.`

- Use a single space between each property value. You use a comma only to separate fonts in a list like this: Arial, Helvetica, sans-serif.

- When specifying the line height, add a slash after the font size followed by the line-height value, like this: `1.5em/150% or 24px/37px.`

The last two properties must be `font-size` (or `font-size/line-height`) followed by `font-family`, in that order. All the other properties may be written in any order. For example, these two declarations are the same:

```
font: italic bold small-caps 1.5em Arial;
font: bold small-caps italic 1.5em Arial;
```

Finally, omitting a value from the list is the same as setting that value to normal. Say you create a `<p>` tag style that formats all paragraphs in bold, italics, and small caps with a line height of 2000 percent (not that you'd actually *do* that). You can then create a class style named, say, `.special-Paragraph` with the following font declaration:

```
font: 1.5em Arial;
```

When you apply this style to one paragraph on the page, then that paragraph would *not* inherit the italic, bold, small caps, or line height. Omitting those four values in the `.special-Paragraph` style is the same as writing this:

```
font: normal normal normal 1.5em/normal
Arial/;
```

### CONTROLLING MARGINS BETWEEN PARAGRAPHS

Many designers hate the space that every browser throws in between paragraphs. Before CSS, there was nothing you could do about it. Fortunately, you can now tap into the `margin-top` and `margin-bottom` properties to remove (or, if you wish, expand) that gap. To totally eliminate a top and bottom margin, write this:

```
margin-top: 0;
margin-bottom: 0;
```

To eliminate the gaps between *all* paragraphs on a page, create a style like this:

```
p {
  margin-top: 0;
  margin-bottom: 0;
}
```

As with `text-indent`, you can use pixel or em values to set the value of the margins. You can also use percentages, but as with `text-indent`, the percentage is related to the *width* of the paragraph's containing element. Because it's confusing to calculate the space above and below a paragraph based on its width, it's easier to stick with either em or pixel values.

**NOTE** Because not all browsers treat the top and bottom margin of headlines and paragraphs consistently, it's often a good idea to simply *zero out* (that is, eliminate) all margins at the beginning of a style sheet. To see how this works, turn to page 548.

For a special effect, you can assign a *negative* value to a top or bottom margin. For example a –10px top margin moves the paragraph up 10 pixels, perhaps even visually overlapping the page element above it. (See step 4 on page 190 for an example.)

## Formatting the First Letter or First Line of a Paragraph

CSS also provides a way of formatting just a part of a paragraph by using the `:first-letter` and `:first-line` pseudo-elements (see Figure 6-16). Technically, these aren't CSS properties, but types of selectors that determine what part of a paragraph CSS properties should apply to. With the `:first-letter` pseudo-element, you can create an initial capital letter to simulate the look of a hand-lettered manuscript. To make the first letter of each paragraph bold and red you could write this style:

```
p:first-letter {
  font-weight: bold;
  color: red;
}
```

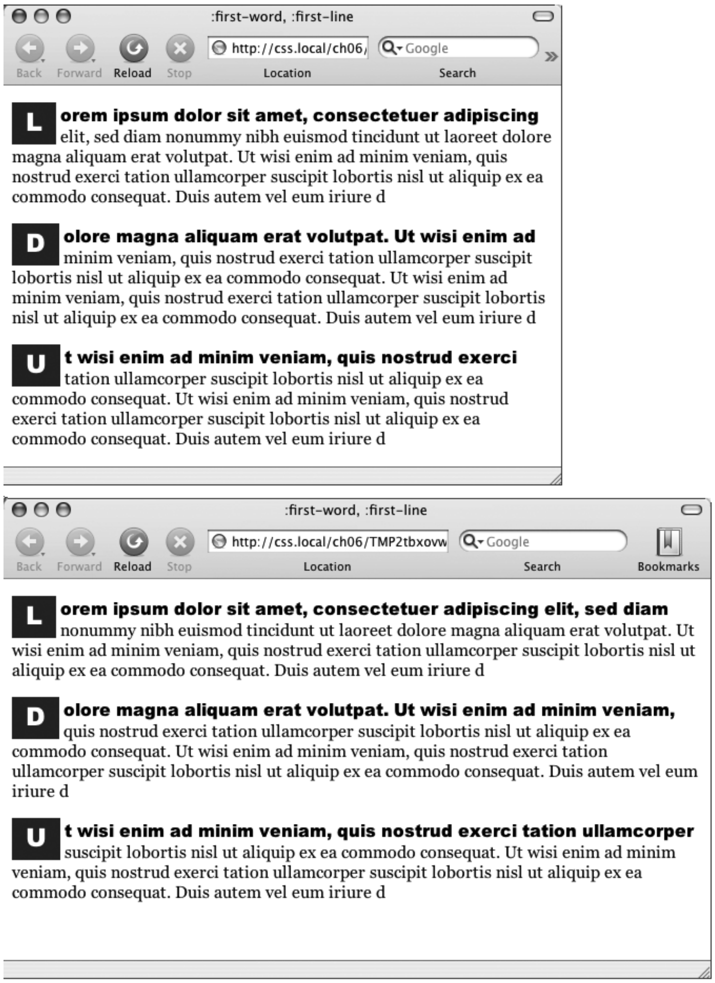
To be more selective and format just the first letter of a particular paragraph, you can apply a class style to the paragraph—`.intro`, for example:

```
<p class="intro">Text for the introductory paragraph goes here...</p>
```

Then you could create a style with a name like this: `.intro:first-letter`.

The `:first-line` pseudo-element formats the initial line of a paragraph. You can apply this to any block of text like a heading (`h2:first-line`) or paragraph (`p:first-line`). As with `:first-letter`, you can apply a class to just one paragraph and format only the first line of that paragraph. Say you want to capitalize every letter in the first line of the first paragraph of a page. Apply a class to the HTML of the first paragraph—`<p class="intro">`—and then create a style like this:

```
.intro:first-line { text-transform: uppercase; }
```



**L** **orem ipsum dolor sit amet, consectetuer adipiscing** elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure d

**D** **olore magna aliquam erat volutpat. Ut wisi enim ad** minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure d

**U** **t wisi enim ad minim veniam, quis nostrud exerci** tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure d

**L** **orem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam** nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure d

**D** **olore magna aliquam erat volutpat. Ut wisi enim ad minim veniam,** quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure d

**U** **t wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper** suscipit lobortis nisl ut aliquip ex ea commodo consequat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure d

**FIGURE 6-16**

*The `:first-letter` pseudo-element formats the first letter of the styled element, like the initial caps to the left. The `:first-line` selector, on the other hand, styles the first line of a paragraph. Even if your guest resizes the window (bottom), the browser still styles every word that appears on a paragraph's first line.*

**NOTE** For some strange reason, neither Chrome nor the Safari web browser understands the `text-transform` property (page 163) when it's used with the `:first-line` pseudo-element. In other words, you can't use CSS to capitalize the letters of a paragraph's first line in Chrome or Safari.

## Styling Lists

The `<ul>` and `<ol>` tags create bulleted and numbered lists, like lists of related items or numbered steps. But you don't always want to settle for the way web browsers automatically format those lists. You may want to swap in a more attractive bullet, use letters instead of numbers, or even completely eliminate the bullets or numbers.

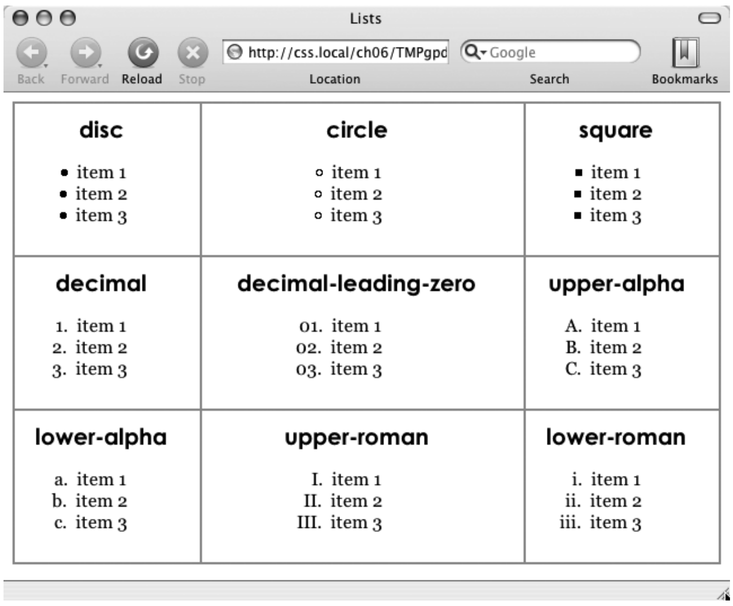
### Types of Lists

Most web browsers display unordered lists (`<ul>` tags) using round bullets, and numbered lists (`<ol>` tags) using...well...numbers. With CSS, you can choose from among three types of bullets—`disc` (a solid round bullet), `circle` (a hollow round bullet), or `square` (a solid square). There are also six different numbering schemes—`decimal`, `decimal-leading-zero`, `upper-alpha`, `lower-alpha`, `upper-roman`, or `lower-roman` (see Figure 6-17). You select all these options using the *list-style-type* property, like so:

```
    list-style-type: square;
```

or

```
    list-style-type: upper-alpha;
```



| | | |
|---|---|---|
| **disc**<br>● item 1<br>● item 2<br>● item 3 | **circle**<br>○ item 1<br>○ item 2<br>○ item 3 | **square**<br>■ item 1<br>■ item 2<br>■ item 3 |
| **decimal**<br>1. item 1<br>2. item 2<br>3. item 3 | **decimal-leading-zero**<br>01. item 1<br>02. item 2<br>03. item 3 | **upper-alpha**<br>A. item 1<br>B. item 2<br>C. item 3 |
| **lower-alpha**<br>a. item 1<br>b. item 2<br>c. item 3 | **upper-roman**<br>I. item 1<br>II. item 2<br>III. item 3 | **lower-roman**<br>i. item 1<br>ii. item 2<br>iii. item 3 |

**FIGURE 6-17**

*CSS provides many different ways to mark bulleted and numbered lists, from a handful of geometric shapes to many different numbering systems. If you feel like rushing a fraternity or sorority, you can also replace numbers with Greek letters—$\alpha$, $\beta$, $\gamma$—using the* `lower-greek` *option. In fact, there are a number of other numbering schemes including Armenian, Georgian, Katakana, and other regional variations. You can find out about them at https://developer.mozilla.org/en-US/docs/CSS/list-style-type.*

Most of the time, you use this property on a style that's formatting an `<ol>` or `<ul>` tag. Typical examples include an `ol` or `ul` tag style—`ul { list-style-type: square; }`—or a class you're applying to one of those tags. However, you can also apply the property to an individual list item (`<li>` tag) as well. You can even apply different types of bullet styles to items within the same list. For example, you can create a style for a `<li>` tag that sets the bullets to `square`, but then create a class named `.circle` that changes the bullet type to `circle`, like this:

```
li {list-style-type: square; }
.circle { list-style-type: circle; }
```

You can then apply the class to every other item in the list to create an alternating pattern of square and circular bullets:

```
<ul>
<li>Item 1</li>
<li class="circle">Item 2</li>
<li>Item 3</li>
<li class="circle">Item 4</li>
</ul>
```

Or, using the nth-of-type selector (page 78) you could skip the class name entirely:

```
li {list-style-type: square; }
li:nth-of-type(odd) { list-style-type: circle; }
```

At times you'll want to completely hide bullets, like when you'd rather use your own graphic bullets (see page 176). Also, when a site's navigation bar is a list of links, you can use an `<ul>` list, but hide its bullets (see the example on page 286). To turn off the bullets, use the keyword `none`:

```
list-style-type: none;
```

## Positioning Bullets and Numbers

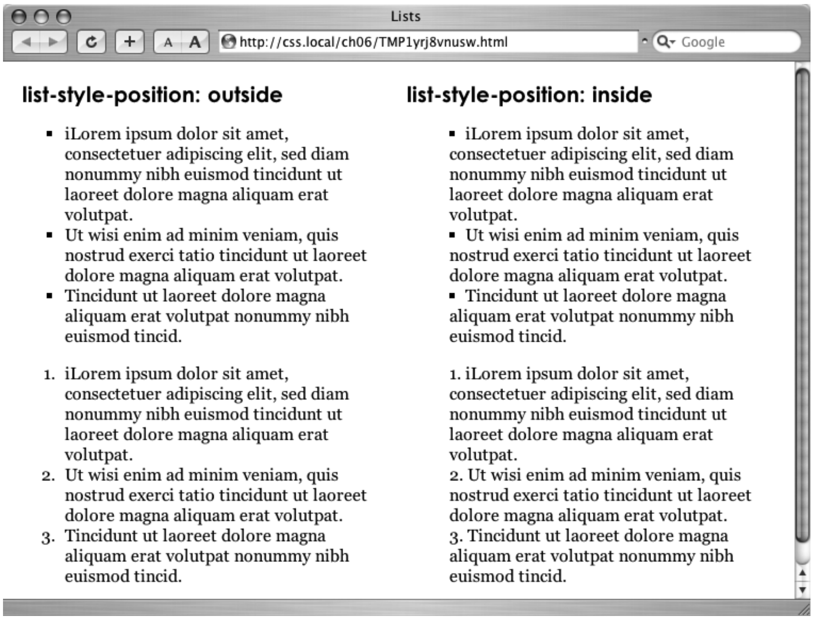
Web browsers usually display bullets or numbers hanging to the left of the list item's text (Figure 6-18, left). With CSS, you can control the position of the bullet (somewhat) using the `list-style-position` property. You can either have the bullet appear `outside` of the text (the way browsers normally display bullets) or `inside` the text block itself (Figure 6-18, right):

```
list-style-position: outside;
```

or

```
list-style-position: inside;
```

**TIP** You can adjust the space between the bullet and its text—increase or decrease that gap—by using the `padding-left` property. To use it, you create a style that applies to the `<li>` tags. This technique works only if you set the `list-style-position` property to the `outside` option (or don't use `list-style-position` at all).



**list-style-position: outside**

- iLorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
- Ut wisi enim ad minim veniam, quis nostrud exerci tatio tincidunt ut laoreet dolore magna aliquam erat volutpat.
- Tincidunt ut laoreet dolore magna aliquam erat volutpat nonummy nibh euismod tincid.

1. iLorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
2. Ut wisi enim ad minim veniam, quis nostrud exerci tatio tincidunt ut laoreet dolore magna aliquam erat volutpat.
3. Tincidunt ut laoreet dolore magna aliquam erat volutpat nonummy nibh euismod tincid.

**list-style-position: inside**

- iLorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
- Ut wisi enim ad minim veniam, quis nostrud exerci tatio tincidunt ut laoreet dolore magna aliquam erat volutpat.
- Tincidunt ut laoreet dolore magna aliquam erat volutpat nonummy nibh euismod tincid.

1. iLorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
2. Ut wisi enim ad minim veniam, quis nostrud exerci tatio tincidunt ut laoreet dolore magna aliquam erat volutpat.
3. Tincidunt ut laoreet dolore magna aliquam erat volutpat nonummy nibh euismod tincid.

**FIGURE 6-18**

*Using the `list-style-position` property, you can control the position of bullets and numbers in a list. The outside option (left) emphasizes the "listness" of your list. Use the `inside` option (right) if you need to maximize the width of your list.*

In addition, if you don't like how web browsers indent a list from the left edge, then you can remove that space by setting both the `margin-left` and `padding-left` properties to `0` for the list. To remove the indent from all lists, you can create this group selector:

```
ul, ol {
  padding-left: 0;
  margin-left: 0;
}
```

Or, you can create a class style with those properties and apply it to a particular `<ul>` or `<ol>` tag. The reason you need to set both the padding and margin properties is that some browsers use padding (Firefox, Mozilla, Safari) and some use margin (Internet Explorer) to control the indent. (You'll learn more about the margin and padding properties in the next chapter.)

Browsers normally display one bulleted item directly above another, but you can add space between list items using the `margin-top` or `margin-bottom` properties on the particular list items. These properties work for spacing list items exactly the same way they work for spacing paragraphs, as explained on page 171. You just need to make sure that the style applies to the `<li>` tags by creating a class style and applying it individually to each `<li>` tag. Or, better yet, create an `<li>` tag style or descendent selector. The style should *not* apply to the `<ul>` or `<ol>` tag. Adding margins to the top or bottom of those tags simply increases the space between the entire list and the paragraphs above or below it—not the space between each item in the list.

## Graphic Bullets

If you're not happy with squares and circles for your bullets, create your own. Using an image-editing program like Photoshop or Fireworks, you can quickly create colorful and interesting bullets. Clip art collections and most symbol fonts (like Webdings) provide great inspiration.

**TIP** For a listing of loads of sites with free icons and bullets, check out this page: *www.cssjuice.com/38-free-icon-checkpoints/*.

The CSS `list-style-image` property lets you specify a path to a graphic on your site, much as you specify a file when adding an image to a page by using the `src` attribute of the HTML `<img>` tag. You use the property like this:

```
    list-style-image: url(images/bullet.gif);
```

The term `url` and the parentheses are required. The part inside the parentheses—`images/bullet.gif` in this example—is the path to the graphic. Notice that, unlike HTML, you don't have to use quotation marks around the path (though you can, if you like).

**FREQUENTLY ASKED QUESTION**

## Customizing List Bullets and Numbers

*I'd like the numbers in my numbered lists to be bold and red instead of boring old black. How do I customize bullets and numbers?*

CSS gives you a few ways to customize the markers that appear before list items. For bullets, you can use your own graphics, as described previously. You have two other techniques available: one that's labor intensive, but works on most browsers, and one that's super geeky, cutting edge, and doesn't work in Internet Explorer 7 or earlier.

First, the labor-intensive way. Say you want the numbers in an ordered list to be red and bold, but the text to be plain, unbolded black. Create a style that applies to the list—like a class style you apply to the `<ol>` or `<ul>` tags—with a text color of red and the font weight set to bold. At this point, everything in the list—text included—is red and bold.

Next, create a class style—`.regularList`, for example—that sets the font color to black and font weight to normal (that is, not bold). Then (and this is the tedious part), wrap a `<span>` tag around the text in each list item and apply the class style to it. For example:

```
<li><span class="regularList">Item 1</
span></li>
```

Now the bullets are bold and red and the text is black and normal. Unfortunately, you have to add that `<span>` to *every* list item!

The cool, "I'm so CSS-savvy" way is to use what's called *generated content*. Basically, generated content is just stuff that isn't actually typed on the page but is added by the web browser when it displays the page. A good example is bullets themselves. You don't type bullet characters when you create a list; the browser adds them for you. With CSS, you can have a browser add content, and even style that content, before each `<li>` tag. You read about generated content on pages 70-71, but just so you have the code at hand if you want to make regular bullets next to the list items red, add this CSS to your style sheet:

```
ul li {
  list-style-type: none;
}
ul li:before {
  content: counter(item, disc) " ";
  color: red;
}
```

And, if you wanted to make the items in a numbered list red, you can add this CSS:

```
ol li {
  list-style-type: none;
  counter-increment: item;
}
ol li:before {
  content: counter(item) ". ";
  color: red;
}
```

For a more in-depth explanation of styling numbered lists visit *www.456bereastreet.com/archive/201105/styling_ordered_list_numbers/*.

**NOTE** When specifying a graphic in an *external* style sheet, the path to the image is relative to the style sheet file, not the web page. You'll learn more about how this works in the box on page 244, as you start to use images with CSS.