CHAPTER

7

Adding Graphics

It's safe to say that the creators of the Internet never imagined it would look the way it does today—thick with pictures, ads, and animated graphics. They designed a meeting place for leading academic minds; we ended up with something closer to a Sri Lankan bazaar. But no one's complaining, because the Web would be an awfully drab place without graphics.

In this chapter, you'll master the art of Web images. You'll learn how to add graphics to a Web page and to position them perfectly. You'll also consider what it takes to prepare pictures for the Web—or to find good alternatives online.

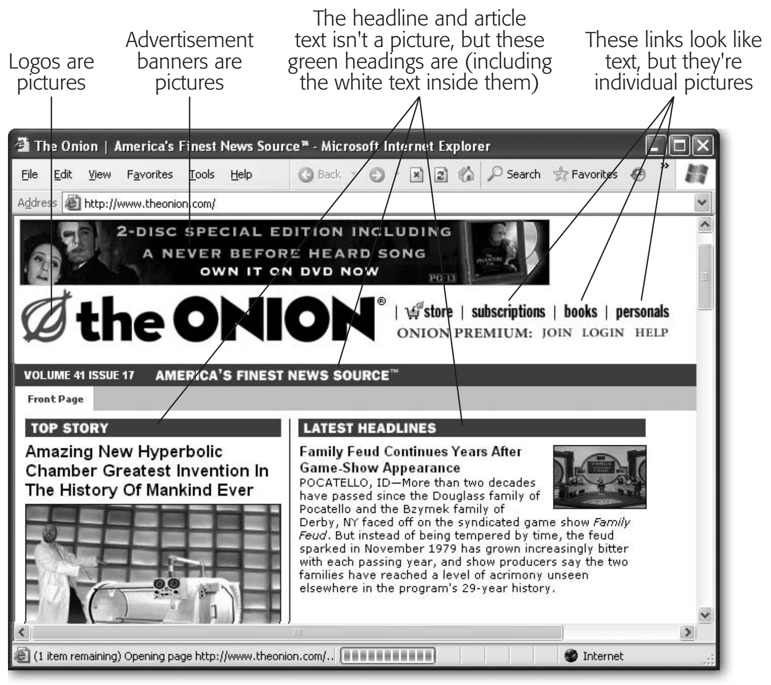
Understanding Images

To understand how images work on the Web, you need to know two things:

- They don't reside in your XHTML files. Instead, you store each one as a separate file.
- To display pictures on a page, you use the <img> element in your XHTML document.

You'll use images throughout your site, even in spots where you might think ordinary text would work just fine (see Figure 7-1).

Tip: If you can't tell whether a piece of content on a page is a graphic, try right-clicking it. If it's an image, browsers like Internet Explorer and Firefox give you a Save Picture As option in a pop-up menu.


Logos are pictures
Advertisement banners are pictures
The headline and article text isn't a picture, but these green headings are (including the white text inside them)
These links look like text, but they're individual pictures
The Onion | America's Finest News Source™ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
http://www.theonion.com/
2-DISC SPECIAL EDITION INCLUDING A NEVER BEFORE HEARD SONG OWN IT ON DVD NOW
the ONION
store | subscriptions | books | personals
ONION PREMIUM: JOIN LOGIN HELP
VOLUME 41 ISSUE 17 AMERICA'S FINEST NEWS SOURCE™
Front Page
TOP STORY
LATEST HEADLINES

Amazing New Hyperbolic Chamber Greatest Invention In The History Of Mankind Ever

Family Feud Continues Years After Game-Show Appearance
POCATELLO, ID—More than two decades have passed since the Douglass family of Pocatello and the Bzymek family of Derby, NY faced off on the syndicated game show *Family Feud*. But instead of being tempered by time, the feud sparked in November 1979 has grown increasingly bitter with each passing year, and show producers say the two families have reached a level of acrimony unseen elsewhere in the program's 29-year history.

(1 item remaining) Opening page http://www.theonion.com/... Internet

Figure 7-1: *It's easy to underestimate how many graphics sit on an average page. Besides just ordinary pictures, adornments like bullets, logos, text headings, and colorful borders are most likely graphics.*

The <img> Element

Pictures appear on your Web pages courtesy of the <img> element, which tells a browser where to find them. For example, here's an <img> element that displays the file named *photo01.jpg*:

```
<img src="photo01.jpg" />
```

Pictures are standalone elements (page 33), which means you don't need to include separate start and end tags in the element. Instead, you include the slash (/) character at the end of the tag, just before the closing angle bracket.

Pictures are also inline elements (page 46), which means you put them *inside* other block elements, like paragraphs:

```
<p><img src="photo01.jpg" /></p>
```

When a browser reads this <img> element, it sends out a request for the *photo01.jpg* file. Once it retrieves it, the browser inserts the file into the Web page where you put the <img> element. If the image file is large or the Internet connection is slow, you might actually see this two-stage process take place, because smaller page components, like text, will appear before the image does.

Tip: You'll usually want to organize your site's many files by putting images in a subfolder of the folder that holds your Web pages. You'll learn how to do this in Chapter 8.

Although it may seem surprising, the <img> element is the only piece of XHTML you need to display a picture. But to get the results you want, you need to understand a few more issues, including how to use alternate text, modify the size of your images, choose a file format, and align your images with other content on a page.

Alternate Text

Although a browser can display an <img> element as long as it has a *src* attribute, the rules of XHTML demand a little more. Technically, you also need to provide an *alt* attribute, which represents the alternate text a browser displays if it can't display the image itself. Here's an example:

```
<img src="photo01.jpg"
 alt="There's no picture, so all you get is this alternate text." />
```

Alternate text proves useful not only in the above circumstance, but in several other cases as well, like when:

- A browser doesn't support images (this is understandably rare these days).
- A Web visitor switches off his browser's ability to display pictures to save time (this isn't terribly common today, either).
- A browser requests a picture, but can't find it. (Perhaps you forgot to copy it to your Web server?)
- The visitor is viewing-impaired and uses a *screen-reading* program (a program that "speaks" text, including the words in an alt tag).
- A search engine (like Google) analyzes a page and all its content so it can index the content in a search catalog.

The last two reasons are the most important. Web experts always use meaningful text when they write alt descriptions to ensure that screen readers and search engines interpret the corresponding pictures correctly.

These days, many sites use alternate text for a completely different purpose—as a pop-up message that appears when you move your mouse over a picture (see Figure 7-2). The message might caption the picture rather than describe it, or it might be a humorous remark.

This behavior is a little controversial, because it defeats the true purpose of alternate text. If you want non-descriptive pop-up text like that shown in Figure 7-2, there's a better solution: the *title* attribute. After all, XHTML's creators designed the title attribute exclusively for this purpose. Here's an example:

```
<img src="bullhero.jpg" alt="A flying bull-headed superhero."
 title="A flying bull-headed superhero." />
```

If you specify a title attribute, browsers use it as the pop-up text. But if you *don't* specify one, different browsers react differently. Internet Explorer uses the alt text instead. Firefox uses the correct approach, and doesn't show any pop-up text at all.

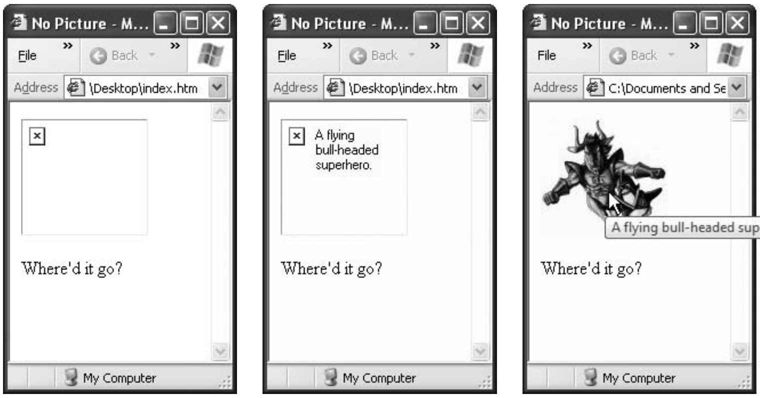


Figure 7-2:
Left: For this <img> element to work, you have to put the file it points to in the same folder as the Web page. Otherwise, you'll see the dreaded broken image link icon.

Middle: The alternate text helps a bit–use it to explain what your visitor should have seen.

Right: In many browsers, if you don't specify a title, the alternate text becomes pop-up text if the picture appears, which can be confusing.

Picture Size

When you start thinking about the size of your images, remember that the word *size* has two possible meanings: it can refer to the dimensions of the picture (how much screen space it takes up on a Web page), or it can signify the picture's file size (the number of bytes required to store it). To Web page creators, both measures are important.

Picture dimensions are noteworthy because they determine how much screen real estate an image occupies. Web graphics are measured in units called pixels. A pixel represents one tiny dot on a PC screen (see the discussion on page 232). Fixed units like inches and centimeters aren't useful in the Web world because you never know how large your visitor's monitor is, and therefore how many pixels it can cram in. (Page 230 has a detailed discussion of screen size and how to design your pages to satisfy the largest number of potential viewers.)

File size is also important because it determines how long it takes to send a picture over the Internet to a browser. Large pictures can slow down a Web site significantly, especially if you have multiple pictures on a page and your visitor is struggling with a slow Internet connection. If you're not careful, impatient people might give up and go somewhere else. (To understand file size and how you can control it, you need to understand the different image file formats Web browsers use, a topic discussed in the next section.)

Interestingly, the <img> element lets you resize a picture through its optional height and width attributes. Consider this element:

```
<img src="photo01.jpg" alt="An explicitly sized picture" width="100"
height="150" />
```

In this line of code, you give the picture a width of 100 pixels and a height of 150 pixels. If this doesn't match the real dimensions of your source picture, browsers stretch and otherwise mangle the image until it fits the size you set (see Figure 7-3).

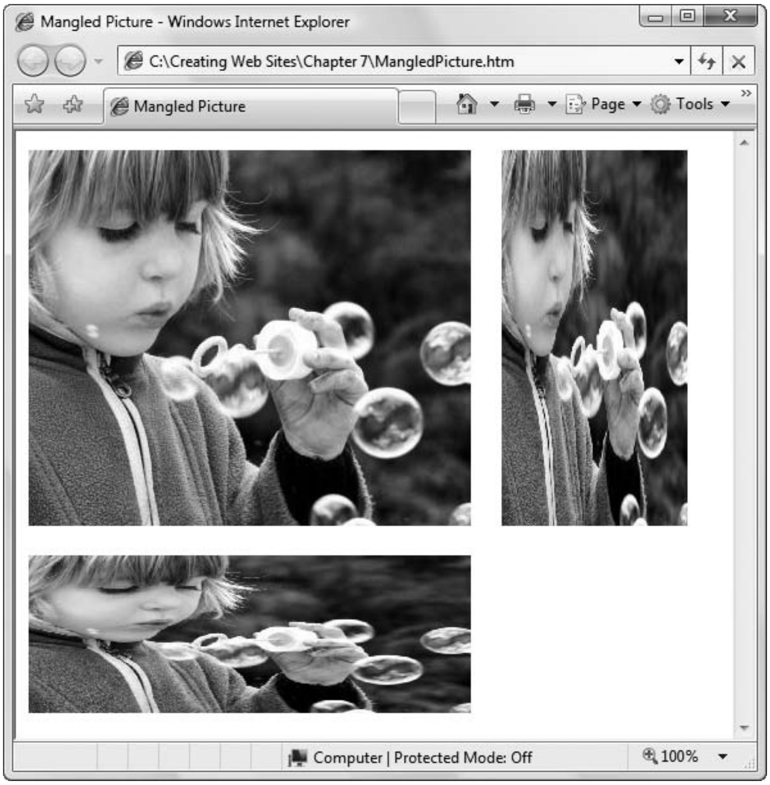

Mangled Picture - Windows Internet Explorer
C:\Creating Web Sites\Chapter 7\MangledPicture.htm
Mangled Picture
Page
Tools
Computer | Protected Mode: Off
100%

Figure 7-3:
Never use the height and width attributes to resize a picture, because the results are almost always unsatisfying. Enlarged pictures are jagged, shrunken pictures are blurry, and if you change the ratio of height to width (as with the top-right and bottom images shown here), browsers squash pictures out of their normal proportions.

Note: Approach height and width attributes with extreme caution. Sometimes, novice Web authors use them to make *thumbnails*, small versions of large pictures. But using the height and width attributes to scale down a large picture comes with a performance penalty—namely, the browser still needs to download the original, larger image, even though it displays it at a smaller size. On the other hand, if you create thumbnails in a graphics editor like Photoshop, you can save them with smaller file sizes, ensuring that your pages download much speedier.

Many Web page designers leave out image height and width attributes. However, experienced Web developers sometimes add them using the *same* dimensions as the actual picture. As odd as this sounds, there are a couple of good reasons to do so.

First, when you include image size attributes, browsers know how large a picture is and can start laying out a page even as the graphic downloads (see Figure 7-2, left). On the other hand, if you don't include the height and width attributes, the browser won't know the dimensions of the picture until it's fully downloaded, at which point it has to rearrange the content. This is potentially distracting if your visitors have slow connections and they've already started reading the page.

The second reason to use size attributes is because they control the size of the picture box. If a browser can't download an unsized image for some reason, it displays a picture box just big enough to show a tiny error icon and any alternate text. In a complex Web page, that might mess up the alignment of other parts of your page.

So should you use the height and width attributes? It's up to you, but they're probably more trouble than they're worth for the average Web site. If you use them, you need to make sure to update them if you change the size of your picture, which quickly gets tedious.

Note: Many XHTML editors, like Expression Web, automatically add the height and width attributes when you insert a picture.

File Formats for Graphics

Browsers can't display every type of image. In fact, they're limited to just a few image formats, including:

- **GIF** (pronounced "jif" or "gif") format is suitable for graphics with a very small number of colors (like simple logos or clip art). It gives terrible results if you use it to display photos.
- **JPEG** (pronounced "jay-peg") format is suitable for photos that can tolerate some loss of quality. (As you'll learn in a moment, the JPEG format shrinks down, or compresses, an image's file size so that it downloads more quickly.) JPEG doesn't work well if your picture contains text or line art.
- **PNG** (pronounced "ping") format is suitable for all kinds of images, but old browsers don't support it, and it doesn't always compress as well as JPEG.

All of these formats are known as bitmap or *raster* graphics, because they represent pictures as a grid of dots. Browsers don't support *vector* graphics, which represent pictures as mathematically rendered shapes.

Raster graphics generally have much larger file sizes than vector graphics. For that reason, Web designers spend a lot of time worrying about *compression*—reducing the amount of disk space an image takes up. Web page graphics use two types of compression: *lossy*, which compresses files to a greater degree than its alternative but also reduces image quality; and *lossless*, which preserves image quality but doesn't compress as much. For full details, see the box on page 187. Table 7-1 gives you a quick overview of the different image formats.

Table 7-1. *Image file formats for the Web*

Format	Type of Compression	Maximum Colors	Best Suited For:
GIF	Lossless	8-bit color (256 colors)	Logos, graphical text, and diagrams with line art
JPEG	Lossy	24-bit (16.7 million colors)	Photos
PNG-8	Lossless	8-bit color (256 colors)	Rarely used, since it's similar to GIF but with less browser support
PNG-24	Lossless	24-bit (16.7 million colors)	Images that would normally be GIF files, but need more colors

Tip: Some browsers give you a few more options, but you're better off steering away from them to ensure widest browser compatibility. For example, Internet Explorer supports bitmaps (image files that end with the .bmp file name extension). Don't ever use them–not only will they confuse other browsers, they're also ridiculously large because they don't support compression.

You'll probably end up using different file formats throughout your site, depending on what kind of image you want to display (photo, illustration, graphical text, and so on). Each format occupies its own niche (see Figure 7-4). The following sections help you decide when to use each format.

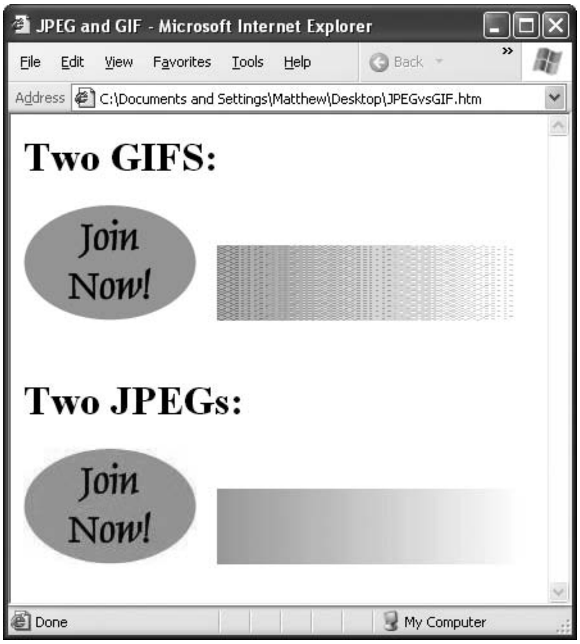

JPEG and GIF - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address C:\Documents and Settings\Matthew\Desktop\JPEGvsGIF.htm
Two GIFS:
Join Now!
Two JPEGs:
Join Now!
Done
My Computer

Figure 7-4:
JPEGs and GIFs are the two most commonly used image file formats on the Web. You'll notice that GIFs produce clearer text, while JPEGs do a much better job of handling continuous bands of color. GIFs simulate extra colors through dithering, a process that mixes different colored dots to simulate a solid color. The results are unmistakably unprofessional. (You may not be able to see the reduced text quality in this black-and-white screen capture, but if you take a look at the downloadable samples for this chapter, you'll see the difference up close.)

FREQUENTLY ASKED QUESTION

Typical File Sizes for Images

How much disk space does a typical picture occupy?

There's no single answer to this question, because it depends on several factors, including the dimensions of the picture, the file format you use, the amount of compression you apply, and how well the picture responds to compression techniques. However, here are a few basic things to keep in mind.

The file size of a typical Web site logo is vanishingly small. Amazon's small logo (about 150×50 pixels) has a file size of a paltry 2 kilobytes (KB), less than the size of most other site logos. Google's signature logo banner clocks in nearly as tiny, at 10 KB. Both are GIF files.

A picture can take up much more disk space. A small news picture in an article on the *New York Times* Web site rarely uses more than 20 KB. A typical eBayer includes a picture of her product that's 30 KB to 150 KB. At this size, the picture usually takes up a larger portion of your browser window. However, that's nothing compared to the size the picture would be if you weren't using compression. For example, even a low-end 1-megapixel camera can take a raw, uncompressed picture of about 3,000 KB. In a Web page, you might compress this to 300 KB or less by using the JPEG file format with a lower quality level.

Of course, the important number is how long it takes a Web visitor to download a page that has a picture in it. Obviously, this depends on the speed of the visitor's Internet connection–a broadband connection won't blink while grabbing a huge graphic, while someone with a dial-up modem can only get about 5 KB each second, meaning it takes about 20 seconds to see all of a 100 KB eBay photo. In Internet time, 20 seconds is a lifetime.

The best advice for keeping your pictures small is to crop them to the right dimensions, use the right image format, and try lowering the quality level of JPEGs to get better compression.

Compression

In Web graphics, space is a key concern. You may have tons of storage space on your Web server, but large files take more time to send across the Internet, which means your visitors will experience some frustrating, toe-tapping seconds before your page appears. To make a graphics-heavy Web site run smoothly—and these days, what Web site *doesn't* have lots of graphics?—you need to pare down the size of your pictures.

Of course, it's not quite that simple. JPEGs give you the best compression, but they throw out some image detail in the process (see the box on page 187). As you compress a JPEG image, you introduce various problems collectively known as *compression artifacts*. The most common artifacts are blocky regions of an image, halos around edges, and a general blurriness. Some pictures exhibit these flaws more than others, depending on the image's amount of detail.

Tip: Most graphics programs let you choose how much you compress a picture, and many even let you preview the result before you save anything.

Figure 7-5 shows the effect of compression settings on a small section of a picture of a church.

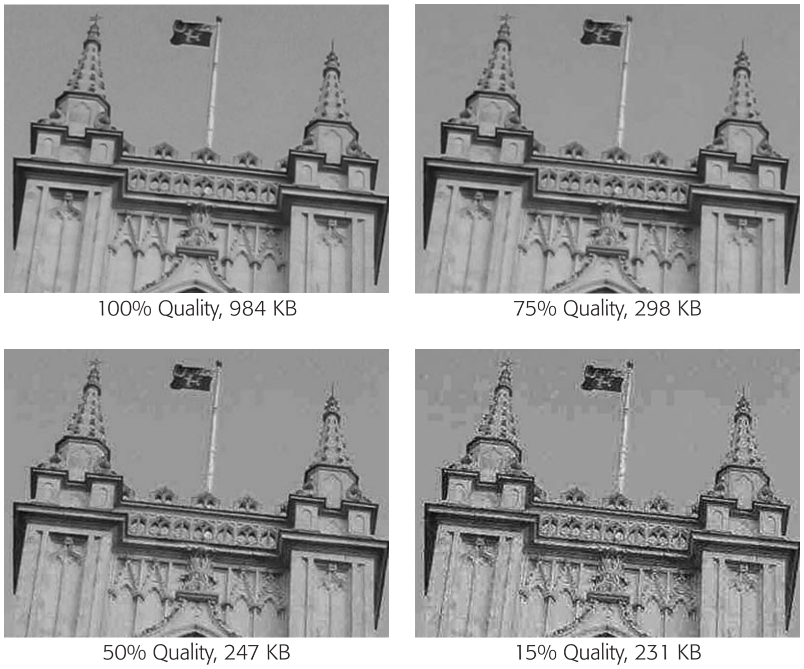


Figure 7-5: *Compression can work–up to a point. In this example, cutting the quality factor from 100 percent to 75 percent shaves the file size of the picture to one-third without compromising its appearance. Reducing the quality further doesn't save much more disk space, and introduces a raft of compression artifacts. Note that the file sizes listed are for the whole picture, which is much bigger than the small portion shown here.*

UP TO SPEED

How Compression Works in JPEG, GIF, and PNG Files

All three of the common Web image formats use *compression* to shrink picture information. However, the type of compression you get with each format differs significantly.

The GIF and PNG formats support *lossless compression*, which means there's no *loss* of any information from your picture. Lossless compression uses a variety of techniques to perform its space-shrinking magic–for example, it might find a repeating pattern in the file, and replace each occurrence of it with a short abbreviation. When the browser decompresses your file, it gets all the original image data back.

The JPEG format uses *lossy compression*, which means that some information about your picture is discarded, or *lost*. As a result, your picture's quality diminishes, and there's no way to get it back to its original tip-top shape. However, the JPEG format is crafty, and it tries to trick your eye by discarding information that doesn't harm the picture that much. For example, it might convert slightly different colors to the same color, or replace fine details with smoothed-out blobs, because the human eye isn't that sensitive to small changes in color and shape. Usually, the overall result is a picture that looks softer and (depending how much compression you use) more blurry. On the other hand, the size-shrinking results you get with lossy compression are more dramatic than those offered by lossless compression.

Choosing the right image format

It's important to learn which format to use for a given task. To help you decide, walk through the following series of questions.

Is your picture a hefty photo or does it have fine gradations of color?

YES: JPEG is the best choice for cutting large, finely detailed pictures down to size. Depending on the graphics program you use, you may be able to choose how much compression you want to apply.

Does your picture have sharp edges, text, or does it contain clip art images? Does it use 256 colors or less?

YES: GIF is your format—it compresses pictures without creating blurred edges around text and shapes (the way JPEG files often do). However, keep a watch on your file size, because GIFs don't compress as well as JPEGs.

Does your picture have sharp edges and need more than 256 colors?

YES: PNG is the best answer here. It supports full color, gives you lossless compression, and you don't lose any detail. However, there are two caveats. First, old browsers don't support some PNG features (like semi-transparency). Second, PNG files can sometimes end up being bigger than they should be. The problem is that even though PNG offers good compression, not all graphics programs take advantage of it. So check your file sizes to make sure you aren't getting a raw deal.

Does your picture include a transparent area?

YES: Use GIF. Although PNG supports transparency (and even goes further, with support for partially transparent areas), support for this feature is sketchy in many browsers. But think twice before you use transparency—the next section explains the problems you'll face.

Putting Pictures on Colored Backgrounds

Graphics editing programs always store image files as rectangles, even when the image itself isn't rectangular. For example, if you create a smiley face graphic, your editing program saves that round illustration on a white, rectangular background.

If your page background is white as well, this doesn't pose a problem because the image background blends in with the rest of your page. But if your page has a different background color (page 150), you'll run into the graphical clunkiness shown in Figure 7-6.

Web designers came up with two solutions to this problem. One nifty idea is to use transparency, a feature that GIF graphics support (as do PNG graphics, but not all browsers support PNG transparency). The basic idea is that your graphic contains *transparent pixels*—pixels that don't have any color at all. When a browser comes across these, it doesn't paint anything. Instead, it lets the background of the page show through. To make part of an image see-through, you define a transparent color using your graphics program. In the example above, for instance, you'd set the white background of your smiley face graphic as the transparent color.

UP TO SPEED

Graphics Programs

It's up to you to choose the format for your image files. Most good graphics programs (like Macromedia Fireworks and Adobe Photoshop) save your documents in a specialized file format that lets you perform advanced editing procedures. Photoshop, for example, saves files in the .psd format. When you're ready to put your picture on a Web page, you save a copy of the .psd file in a *different* format, one specially designed for the Web, like JPEG or GIF. Usually, you do so by choosing File → Save As from the program's menu (although sometimes it's something a little different, like File → Export or File → Save For Web).

As a rule of thumb, you always need at least two versions of every picture you create—a copy in the original format your graphics program uses, and a copy in the GIF, JPEG, or PNG format you use on your Web site. You need to keep the original file so you can make changes whenever necessary, and to make sure the image quality for future versions of the picture are as high as possible.

Once you choose your Web format, your graphics program gives you a number of other options that let you customize details like the compression level. At higher compression levels, your image file is smaller but of lower quality. Some really simple image editors (like the Paint program that ships with Windows) don't let you tweak these settings, so you're stuck with the program's built-in settings.

Graphics programs usually come in two basic flavors—*image editors*, which let you retouch pictures and apply funky effects to graphics, and *drawing programs*, which let you create your own illustrations by assembling shapes and text. Adobe Photoshop (and its lower-priced, less powerful sibling, Photoshop Elements), Corel PHOTO-PAINT, and Corel Paint Shop Pro are well-known image editors. Adobe Illustrator, CorelDRAW, and Macromedia FreeHand are popular drawing programs. Which type of tool you use depends on what you're trying to do. If you're editing pictures of the office party to cut out an embarrassing moment, an image editor makes sense. If you're creating a logo for your newly launched cookie company, you need a drawing program.

If you don't have the luxury of getting a professional graphics program, you can hunt for one on a shareware site like *www.download.com*. Two popular free image editors are GIMP (*www.gimp.org*), which supports all the major operating systems, and Paint.NET (*www.getpaint.net*), which is Windows only.

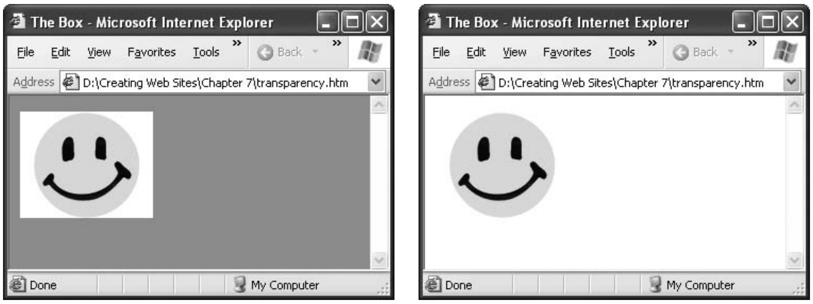
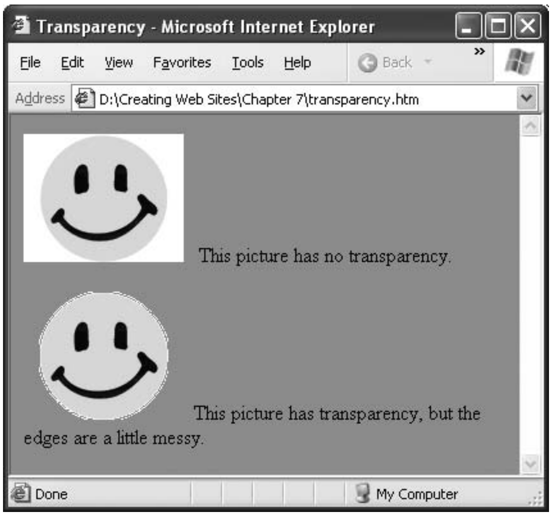

The Box - Microsoft Internet Explorer
File Edit View Favorites Tools
Back
Address D:\Creating Web Sites\Chapter 7\transparency.htm
Done
My Computer
The Box - Microsoft Internet Explorer
File Edit View Favorites Tools
Back
Address D:\Creating Web Sites\Chapter 7\transparency.htm
Done
My Computer

Figure 7-6:
Left: With a non-white background, the white box around your picture is glaringly obvious.

Right: But when you place the picture on a page with a white background, the smiley face blends right in.

Although transparency seems like a handy way to make sure your image always has the correct background, in practice, it rarely looks good. The problem you usually see is a jagged edge where the colored pixels of your picture end and the Web page background begins (see Figure 7-7).


Transparency - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address D:\Creating Web Sites\Chapter 7\transparency.htm
Done
My Computer

This picture has no transparency.

This picture has transparency, but the edges are a little messy.

Figure 7-7:
The picture at the bottom of this page uses transparency, but the result–a jagged edge around the smiley face–is less than stellar. To smooth this edge, graphics programs use a sophisticated technique called anti-aliasing, which blends the picture color with the background color. Web browsers can't perform this feat, so the edges they make aren't nearly as smooth.

The best solution is to use the correct background color when you create your Web graphic. In other words, when you draw your smiley-face image, give it the same background color as your Web page. Your graphics program can then perform anti-aliasing, a technology that smoothes an image's jagged edges to make them look nice. That way, the image edges blend in well with the background, and when you display the image on your Web page, it fits right in.

The only limitation with this approach is its lack of flexibility. If you change your Web page color, you need to edit all your graphics. Sadly, this is the price of creating polished Web graphics.

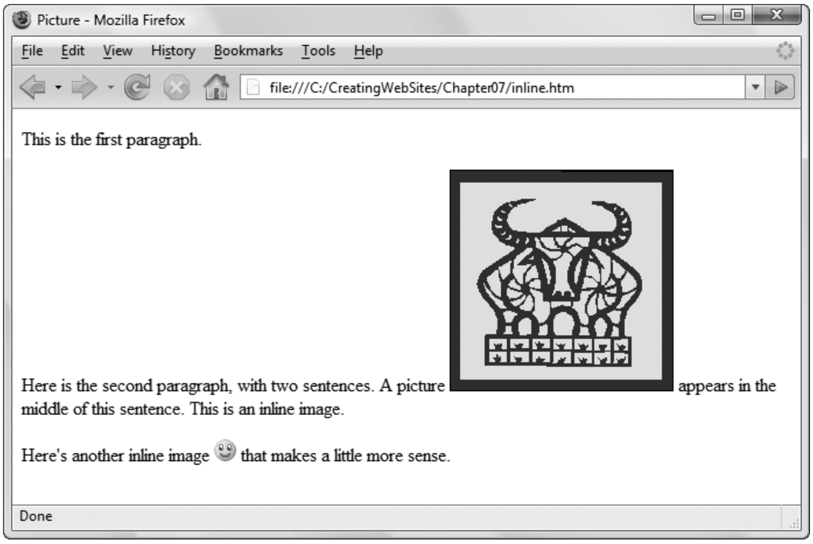
Images and Styles

The <img> element supports a few optional attributes you can use to control an image's alignment and borders. But in the modern world, these attributes are obsolete, and you won't use them in this book. Instead, you'll learn the best way to position images—with style sheet rules.

The following sections describe your image-alignment options, and help you practice some of the style sheet smarts you picked up last chapter.

Inline Images in Text

If you don't take any extra steps, a browser inserts every image right into the flow of XHTML text. It lines up the bottom of a graphic with the baseline of the text that surrounds it, as shown in Figure 7-8. (The baseline is the imaginary line on which a line of text sits.)


Picture - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/CreatingWebSites/Chapter07/inline.htm
Done

This is the first paragraph.

Here is the second paragraph, with two sentences. A picture appears in the middle of this sentence. This is an inline image.

Here's another inline image that makes a little more sense.

Figure 7-8: *Usually, you don't want a picture inside an ordinary line of text (unless it's a very small emoticon, like the kind of symbols used in instant message programs). You can use paragraphs, line breaks, or tables to do a better job of separating images from your text.*

You can change the vertical alignment of text using the vertical-align property. Specify a value of top, middle, or bottom, depending on whether you want to line the picture up with the top, middle, or bottom of the line of text.

Here's an example of an inline style that uses the vertical-align property to line a picture up with the top of the line of text.

```
<img src="happy.gif" alt="Happy Face" style="vertical-align: top" />
```

This technique is worthwhile if you're trying to line up a very small picture, like a fancy bullet. But it doesn't work very well with large images. That's because no matter which vertical-align option you choose, only one line of text can appear alongside the picture (as you can see in Figure 7-8). If you want to create floating pictures with wrapped text, see page 192.

Borders

In Chapter 6, you considered style properties that let you add and modify borders around boxes of text. It should come as no surprise that you can use these borders just as easily around images.

For example, here's a style that applies a thin, grooved border to all sides of an image:

```
img.BorderedImage {
  border-style: groove;
  border-width: 3px;
}
```

As with all style sheet rules, you need to place the rule in an internal style sheet in the current Web page or in an external style sheet that your page uses (see page 138 for a discussion of the difference).

Notice that you give the style in this example a class name (BorderedImage). That's because you don't want your browser to apply the style to every picture. Instead, you want to choose when to apply it using the class attribute:

```
<img src="food.jpg" alt="A Meal" class="BorderedImage" />
```

Figure 7-9 shows the basic border styles. Remember, you can change the thickness of any border to get a very different look.

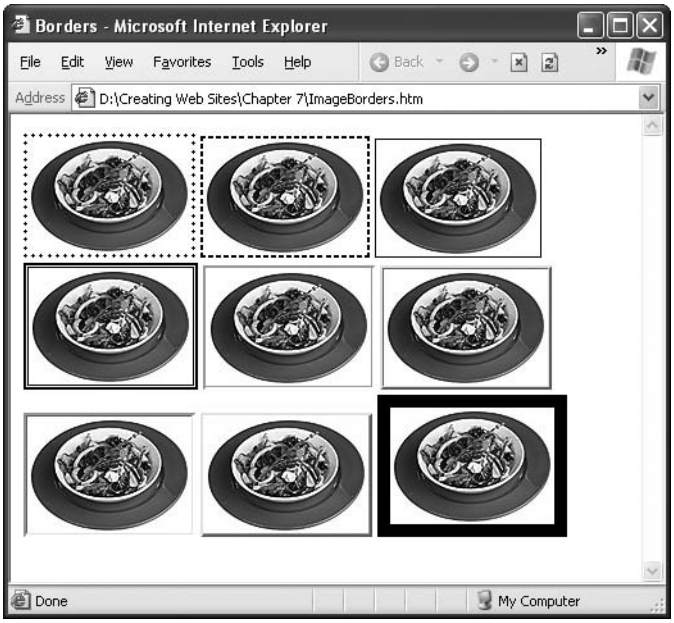

Borders - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address D:\Creating Web Sites\Chapter 7\ImageBorders.htm
Done
My Computer

Figure 7-9:
This example shows several inline images in a row, separated from one another with a single space. Each image in this example is the same, but sports a different border. The browser fits all the pictures it can on the same line, but when it reaches the right edge of the browser window, it wraps to the next line. If you resize the window, you'll see the arrangement of pictures change.

Wrapping Text Around an Image

Using inline images is the simplest way to add pictures to your pages, but they have a downside: they pop up in the middle of text. To prevent this from happening, you can separate your pictures and text using paragraph elements (<p>), line breaks (<br />), horizontal rules (<hr>), and other divisions. You might decide, for example, to put a picture between two paragraphs of text, like this:

```
<p>This paragraph is before the picture.</p>
<p><img src="food.jpg" alt="A Meal" /></p>
<p>This paragraph is after the picture.</p>
```

Inline images are locked into place. They never move anywhere you don't expect.

Sometimes, however, you want a different effect. Instead of separating images and text, you want to put them *alongside* each other. For example, you may want your text to wrap *around* one side of a picture.

Images that have text wrapped on one side or the other are called *floating* images, because they float next to an expanse of text (see Figure 7-10). You create floating images using a CSS property named *float*. You set the value of the float property to either left or right, which lines up the image on either the left or right edge of the text.

```
img.FloatLeft {
  float: left;
}
```

Notice that this example uses a class name. You probably don't want every image on your Web page to float, so it's always a good idea to use a class name. Here's an <img> using the above class, followed by some text:

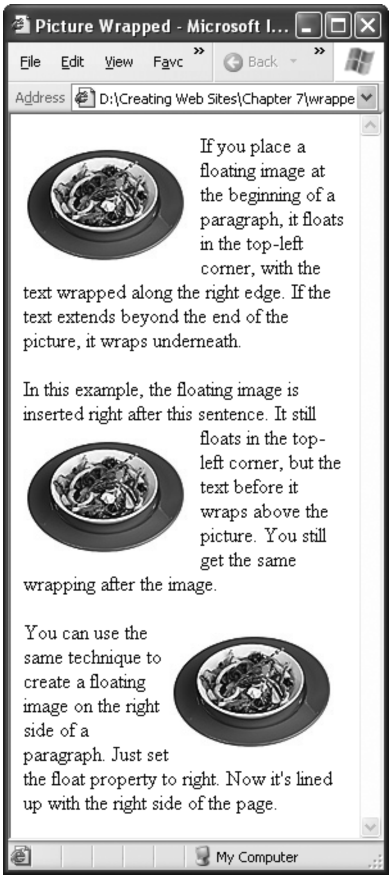
```
<p>
  <img src="food.jpg" alt="A Meal" class="FloatLeft" />
  If you place a floating image at the beginning of a paragraph,
  it floats in the top-left corner, with the text wrapped along
  the right edge.
</p>
```

When you set the float attribute, it makes sense to adjust the image's margin settings at the same time, so you have a little breathing room between your image and the surrounding text:

```
img.FloatLeft {
  float: left;
  margin: 10px;
}
```

Figure 7-10 shows several floating images.

Tip: To get floating text to work the way you want, always put the <img> element just *before* the text that should wrap around the image.


Picture Wrapped - Microsoft I...
File Edit View Favc
Back
Address D:\Creating Web Sites\Chapter 7\wrappe
My Computer

If you place a floating image at the beginning of a paragraph, it floats in the top-left corner, with the text wrapped along the right edge. If the text extends beyond the end of the picture, it wraps underneath.

In this example, the floating image is inserted right after this sentence. It still floats in the top-left corner, but the text before it wraps above the picture. You still get the same wrapping after the image.

You can use the same technique to create a floating image on the right side of a paragraph. Just set the float property to right. Now it's lined up with the right side of the page.

Figure 7-10:
Remember, all image files are really rectangles that include the surrounding white space (see Figure 7-6). As a result, the browser wraps text around the borders of these invisible squares.

Wrapping text can get a little tricky, because the results you get depend on the width of the browser window. For example, you might think your text is long enough to wrap around a graphic, but in a wide window that might take up just a few short lines, letting the rest of the page's content bump into your floating graphic, which isn't what you want (see Figure 7-11). To prevent this from happening, you can put your images in different containers, which is like having different cells in a table. Alternatively, you can manually stop your browser from wrapping text at any point using the *clear* property in a line break (<br />) element:

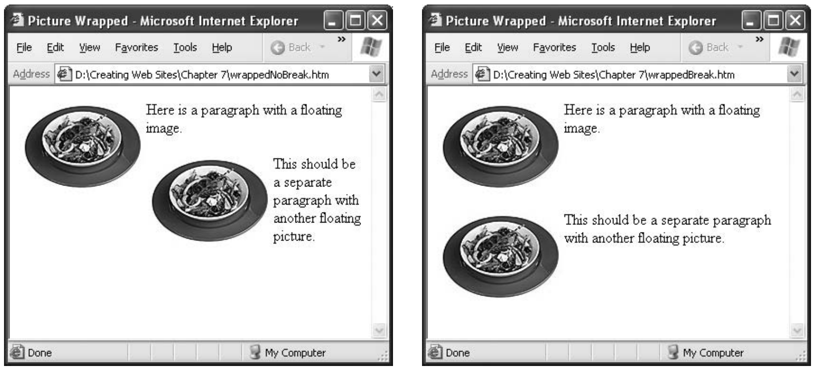
```
<br style="clear: both;" />
```

Place this line at the end of the wrapped paragraph, like so:

```
<p>
  <img src="food.jpg" alt="A Meal" class="FloatLeft" />
  Here is a paragraph with a floating image.
  <br style="clear: both;" />
</p>
```

```
<p>
  <img src="food.jpg" alt="A Meal" class="FloatLeft" />
  This should be a separate paragraph with another
  floating picture.
</p>
```

The clear property in a <br> element tells your browser to stop wrapping text, ensuring that the next paragraph starts after the floating picture (see Figure 7-11).


Picture Wrapped - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address D:\Creating Web Sites\Chapter 7\wrappedNoBreak.htm
Done
My Computer
Picture Wrapped - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address D:\Creating Web Sites\Chapter 7\wrappedBreak.htm
Done
My Computer

Here is a paragraph with a floating image.

This should be a separate paragraph with another floating picture.

Here is a paragraph with a floating image.

This should be a separate paragraph with another floating picture.

Figure 7-11:
Left: Without the clear property, you're in danger of having your floating images run into each other if a browser window is wider than you expect.

Right: The clear property lets you turn off wrapping at a specific point in your document. However, you'll still end up with some extra empty space.

Based on these examples, you might think that the *float* property sends a picture to the left or right side of a page, but that's not exactly what happens. Remember, in CSS, XHTML treats each element on a page as a container. When you create a floating image, the image actually goes to the left or right side of its container. In the previous examples, this means that the image goes to the left or right side of a paragraph, because the containing element is a paragraph.

In the example above, the paragraph took up the full width of the page. But that doesn't have to be the case. You can use style rules to put a paragraph into a padded note box to get a completely different effect.

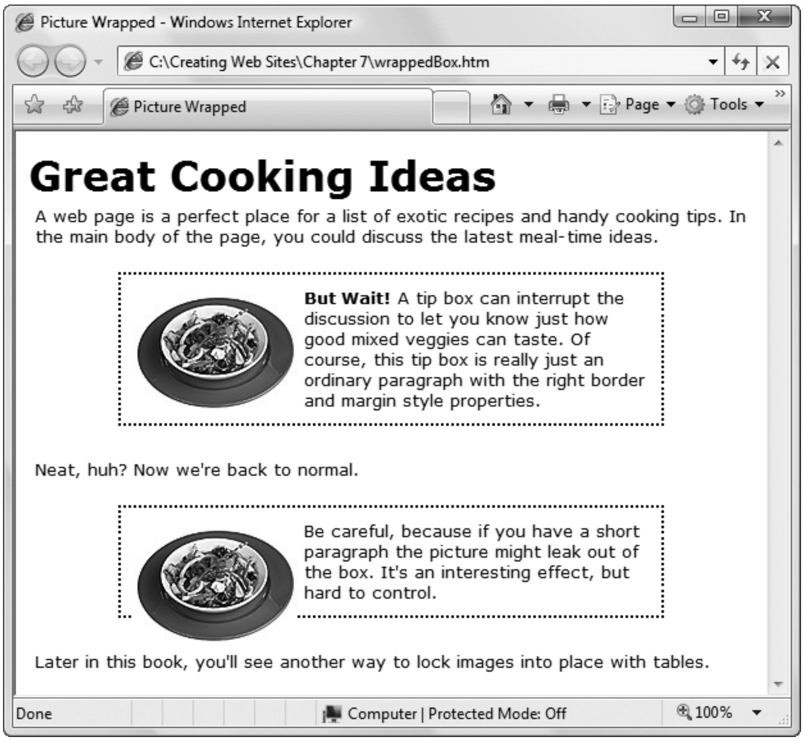
To try this out, you need to wrap the image and the paragraph in a <div> element, like this:

```
<div class="Box">
  <p>
    <img src="food.jpg" alt="A Meal" class="FloatLeft" />
    <b>But Wait!</b> A tip box can interrupt the discussion
    to let you know just how good mixed veggies can taste.
    Of course, this tip box is really just an ordinary paragraph with
    the right border and margin style properties.
  </p>
</div>
```

You can then apply a fancy border to the <div> element through a style rule:

```
div.Box {
  margin-top: 20px;
  margin-bottom: 10px;
  margin-left: 70px;
  margin-right: 70px;
  padding: 5px;
  border-style: dotted;
  border-width: 2px
}
```

Figure 7-12 shows the result.


Picture Wrapped - Windows Internet Explorer
C:\Creating Web Sites\Chapter 7\wrappedBox.htm
Picture Wrapped
Page
Tools
Done
Computer | Protected Mode: Off
100%

Great Cooking Ideas

A web page is a perfect place for a list of exotic recipes and handy cooking tips. In the main body of the page, you could discuss the latest meal-time ideas.

But Wait! A tip box can interrupt the discussion to let you know just how good mixed veggies can taste. Of course, this tip box is really just an ordinary paragraph with the right border and margin style properties.

Neat, huh? Now we're back to normal.

Be careful, because if you have a short paragraph the picture might leak out of the box. It's an interesting effect, but hard to control.

Later in this book, you'll see another way to lock images into place with tables.

Figure 7-12:
With crafty use of styles, you can lay out your pictures with the same flexibility you get when using styles to manipulate text.

Adding Captions

Another nice touch is to caption your pictures above or below an image. You can do this easily with inline images—just put a line of text immediately above or after the picture, separated by a line break. It's not so easy with a floating image, however. In this case, you need to have your image *and* the caption float in the same way.

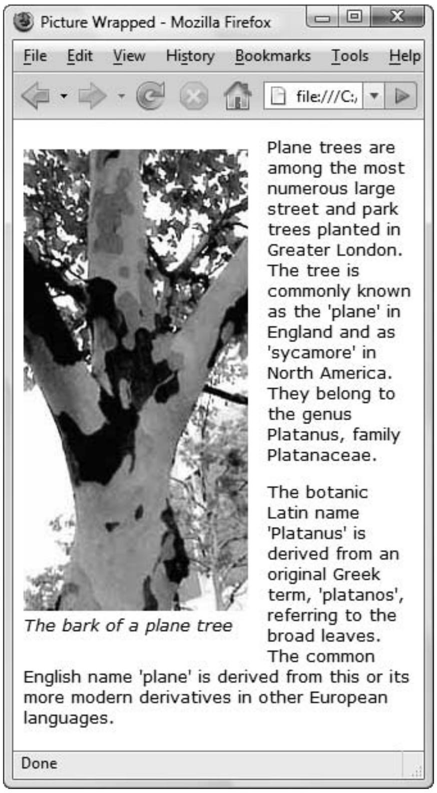
As it happens, the solution is quite easy. You simply take the *FloatLeft* style rule shown earlier and change the name from *img.FloatLeft* to *.FloatLeft* so you can use the rule with *any* element:

```
.FloatLeft {
  float: left;
  margin: 10px;
}
```

Next, you wrap the <img> element and your text into a <span> element. You can then make the entire <span> element float, by using the *FloatLeft* style rule:

```
<span class="FloatLeft">
   <img src="planetree.jpg" alt="Plane Tree" />
   <br />
   <i>The bark of a plane tree</i>
</span>
```

Figure 7-13 shows the result.


Picture Wrapped - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/
Done

Plane trees are among the most numerous large street and park trees planted in Greater London. The tree is commonly known as the 'plane' in England and as 'sycamore' in North America. They belong to the genus Platanus, family Platanaceae.

The bark of a plane tree

The botanic Latin name 'Platanus' is derived from an original Greek term, 'platanos', referring to the broad leaves. The common English name 'plane' is derived from this or its more modern derivatives in other European languages.

Figure 7-13:
Use styles to create captions for floating pictures.

Note: The reason you use a <span> element in this example instead of a <div> element is because you can put a <span> element *inside* other block elements, like a paragraph. In other words, by using a <span> element, you can easily put your floating picture-and-caption container inside one of your paragraphs.

Background Images

CSS makes it possible to use an image as a page background, which is a particularly handy way to create "themed" Web sites. For example, you could use light parchment paper as a background if you're developing a literary site. A *Buffy* fan site might put a dark cemetery image to good use. Some people find the effect a little distracting, but it's worth considering if you want to add a really dramatic touch and you can restrain yourself from going overboard.

Tip: Background images can make your Web site seem tacky. Be wary of using them for a résumé page or a professional business site. On the other hand, if you want to go a little kitschy, have fun!

Web designers almost always choose to *tile* background images, which means the browser copies a small picture over and over again until the image fills the window (see Figure 7-14). You can't use a single image to fill a browser window because you have no way of knowing how wide and tall to make it, given people's variable browser settings. Even if you did have visitors' exact screen measurements, you'd need to create an image so ridiculously large that it would take an impractically long time to download.

To create a tiled background, use the *background-image* style property. Your first step is to apply this property to the <body> element, so that you tile the whole page. Next, you need to provide the file name of the image using the form *url('filename')*, as shown here:

```
body {
  background-image: url('stones.jpg');
}
```

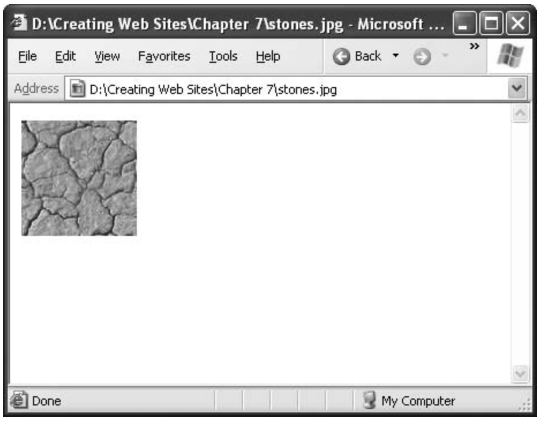
This takes the image *stones.jpg* and tiles it across a page to create your background.

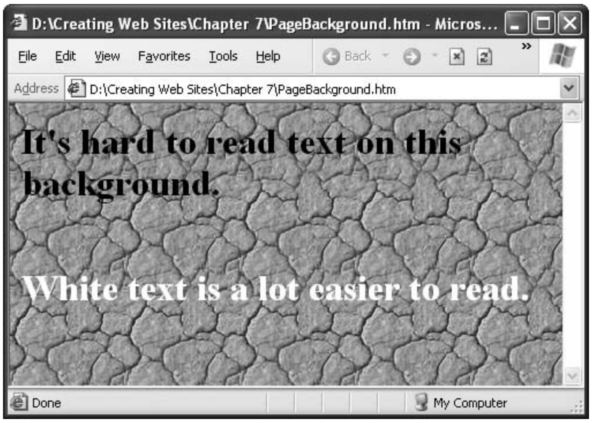
Keep these points in mind when you create a tiled background:

• Make your background light, so the text displayed on top of it remains legible. (If you really have to go dark, you can use white, bold text so that it stands out. But don't do this unless you're creating a Web site for a trendy new band or you're opening a gothic clothing store.)

• Set the page's background color to match the image. For example, if you have a dark background picture with white text, make the background color black. That way, if a browser can't download the background image, visitors can still see the text.

• Use small tiles to reduce the amount of time your visitors need to wait before they can see the page.


D:\Creating Web Sites\Chapter 7\stones.jpg - Microsoft ...
File Edit View Favorites Tools Help
Back
Address D:\Creating Web Sites\Chapter 7\stones.jpg
Done
My Computer


D:\Creating Web Sites\Chapter 7\PageBackground.htm - Micros...
File Edit View Favorites Tools Help
Back
Address D:\Creating Web Sites\Chapter 7\PageBackground.htm
Done
My Computer

It's hard to read text on this background.

White text is a lot easier to read.

Figure 7-14:
Left: A small tile graphic with a stony pattern.

Right: Using style sheets, you can tile this graphic over the whole page. In a good tiled image, the edges line up to create the illusion of a seamless larger picture.

- If your tiled image has an irregular pattern, make sure the edges line up. The left edge should continue the right edge, and the top edge should continue the bottom edge. Otherwise, when the browser tiles your image, you'll see lines where it stitches the tiles together.

Tip: The Web is full of common background images, like stars, blue skies and clouds, fabric and stone textures, fires, dizzying geometric patterns, borders, and much more. You can find these by searching Google for "backgrounds," or head straight to top sites like *www.grsites.com/textures* (with over 5,000 backgrounds indexed by dominant color), *www.backgroundcity.com*, and *www.backgroundsarchive.com*.

Background "watermarks"

Most Web sites tile a picture to create a background image, but that's not your only option. You can also take a single image and place it at a specific position on your page. Think, for example, of a spy site whose background image faintly reads "Top Secret and Confidential."

An inconspicuous single-image background like this is called a *watermark*. (The name stems from the process used to place a translucent logo on paper saturated with water.) To make a good watermark, use a background picture that's pale and unobtrusive.

To add a watermark to your page, use the same background-image property you learned about above. But you need to add a few more style properties to the mix (see Table 7-2). First, you have to use the *background-repeat* property to turn off tiling. At the same time, it makes sense to use the *background-position* property to align your picture to a side of the page or to its center.

Table 7-2. *Background image properties*

Property	Description	Common Values	Can Be Inherited
background-image	The image file you want to use as your page background.	A URL pointing to the image file, as in *url('mypig. jpg').*	No[a]
background-repeat	Whether or not you tile the image to fill the page; you can turn off tiling altogether, or turn it off in one dimension (so that images tile vertically but not horizontally, for example).	repeat, repeat-x, repeat-y, no-repeat	No
background-position	Where you want to place the image. Use this only if you *aren't* tiling the image.	top left, top center, top right, center left, center, center right, bottom left, bottom center, bottom right	No
background-attachment	Whether you want to fix the image (or tiles) in place when the page is scrolled.	scroll, fixed	No

[a] Background pictures aren't inherited (see page 148). However, if you don't explicitly assign a background color to an element, it's given a transparent background, which means the background of the containing element will show through.

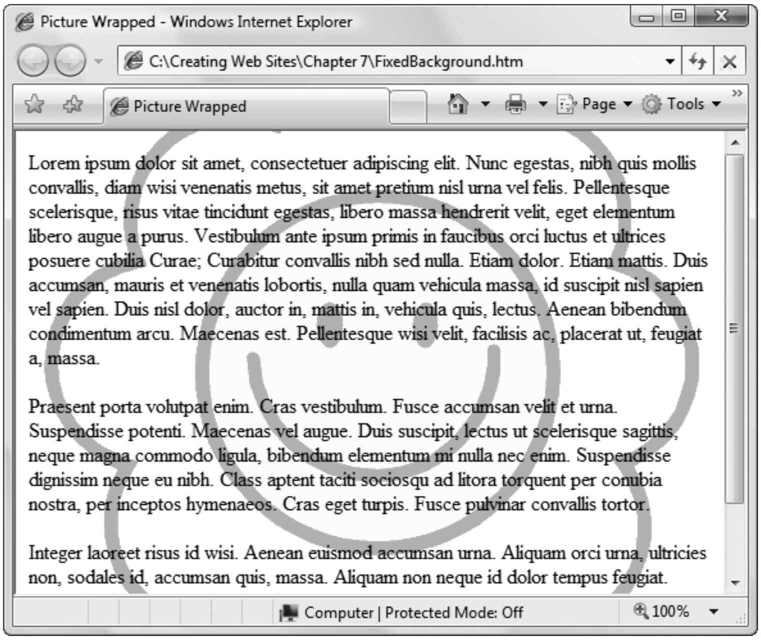
Here's an example that places a picture in the center of a Web page:

```
body {
  background-image: url('smiley.jpg');
  background-repeat: no-repeat;
  background-position: center;
}
```

Note: The center of your document isn't necessarily the center of your browser window. If you position your image in the center of a long Web page, you won't see it until you scroll down.

You can also turn off an image's ability to scroll along with the rest of a page to get the rather odd effect of an image that's fixed in place (see Figure 7-15). For example, use this style to create a background image that sits squarely in the center of a window:

```
body {
  background-image: url('smiley.gif');
  background-repeat: no-repeat;
  background-position: center;
  background-attachment: fixed;
}
```


Picture Wrapped - Windows Internet Explorer
C:\Creating Web Sites\Chapter 7\FixedBackground.htm
Picture Wrapped
Page
Tools
Computer | Protected Mode: Off
100%

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc egestas, nibh quis mollis convallis, diam wisi venenatis metus, sit amet pretium nisl urna vel felis. Pellentesque scelerisque, risus vitae tincidunt egestas, libero massa hendrerit velit, eget elementum libero augue a purus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur convallis nibh sed nulla. Etiam dolor. Etiam mattis. Duis accumsan, mauris et venenatis lobortis, nulla quam vehicula massa, id suscipit nisl sapien vel sapien. Duis nisl dolor, auctor in, mattis in, vehicula quis, lectus. Aenean bibendum condimentum arcu. Maecenas est. Pellentesque wisi velit, facilisis ac, placerat ut, feugiat a, massa.

Praesent porta volutpat enim. Cras vestibulum. Fusce accumsan velit et urna. Suspendisse potenti. Maecenas vel augue. Duis suscipit, lectus ut scelerisque sagittis, neque magna commodo ligula, bibendum elementum mi nulla nec enim. Suspendisse dignissim neque eu nibh. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Cras eget turpis. Fusce pulvinar convallis tortor.

Integer laoreet risus id wisi. Aenean euismod accumsan urna. Aliquam orci urna, ultricies non, sodales id, accumsan quis, massa. Aliquam non neque id dolor tempus feugiat.

Figure 7-15: *This staring smiley face remains perpetually in the center of the window, even when you scroll up or down. It's a little creepy.*

Techniques with Graphics

Now that you've mastered the <img> element, it's time to learn a few tricks of the trade. In the following sections, you'll tour three common techniques that Web developers everywhere use to create more polished pages.

Graphical Text

In Chapter 6, you learned that using exotic fonts on Web pages can be risky, since you don't know which typefaces your visitors has. Although there's no way to get around this problem when you've got large blocks of text, enterprising Web *artistes* commonly put text for headings, buttons, and logos into picture files. That way, they get *complete* control of what the text looks like.