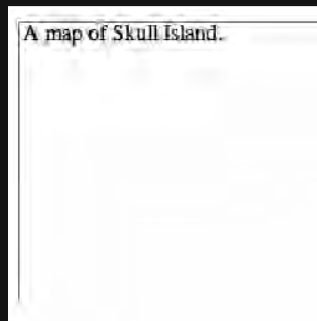


# CHAPTER 5 INCLUDING IMAGES



Ever since Mosaic integrated the display of images within web pages in 1993, imagery has been a cornerstone of the Web. They say a picture is worth a thousand words. Good news then: this is the chapter where we introduce a few thousand words through the inclusion of just a few well-chosen images.

This chapter is designed to give you a working knowledge of using images, introducing you to some fundamental concepts. We focus in particular on a variety of means of *compressing* your images for faster download online, an aspect that is once again emerging as an important consideration as we witness the rapid growth of the mobile Web, accessed via increasingly powerful mobile devices.

We'll introduce you to the three main image formats used online—JPG, GIF, and PNG—and explain why you'd select different image formats for different types of imagery. We'll also show you the HTML you'll need to use to include your images, introducing you to some additional image attributes that, though not strictly necessary for your images to display, are de rigeur for any aspiring Web Standardista.

Before we get started, a word of warning. *Contrary to popular opinion, the imagery you find online isn't free for you to copy at will.* It's a common misconception that everything found on the web is free to use because it's freely available. *It isn't.* The chances are that the image you've fallen in love with that you found online belongs to *someone* and, unless stated otherwise, is *copyright protected*. If you're not absolutely certain an image is free to use, don't use it. Simple.

If creating compelling imagery isn't your strong point, don't worry. There are a number of free or low-cost image resources on the Web at hand. We'll introduce you to some of these at the end of the chapter.

So, now that we've got the copyright issue out of the way, let's get started including images in our well-structured web pages.

## Introducing the <img /> tag

Including images in your web pages isn't difficult; all it requires is the introduction of one additional tag: the <img /> tag. In this section, we introduce you to this tag and its attributes, starting you on your visual journey on the Web.

The best way to show you how the <img /> tag is used, particularly to underline the aspect of *linking to images*, is to get straight into a practical example and look at how the markup is constructed.

### An <img /> tag in action

The minimum markup we need to display an image on a web page is as follows:

```

```

Before we look at this element in a little more detail, it's worth noting that the `<img />` tag, like the `<hr />` tag we introduced in Chapter 4, is a *self-closing tag*. As we've covered before, the closing “`/>`” (space, trailing slash, and angle bracket) are important for our XHTML pages to validate.

*As with all inline-level elements, the `img` element must be nested in a block-level element in order for your pages to validate.*

*Until your markup options increase we suggest enclosing images within an opening `<p>` and a closing `</p>` tag as you'll see in our reference examples for this chapter.*

So, what does the preceding markup do? In a nutshell, it provides a reference to where the image is stored in relation to the HTML file itself. This is a subtle, but important, point to note. Unlike, for example, the text in a `p` element, which is a part of our HTML file itself, the `img` element is used to add *references to images*, not the *actual images themselves* (i.e., the images are stored separately from the HTML file and referenced by it).

When a browser encounters a web page with links to images, it builds the page by following the links supplied within the `<img />` tag, retrieving the images from the server, and rendering the page in its entirety. So, your web page and the images that are included on it are separate files.

In short, the markup points the browser to the source of the image we'd like it to display. This is referenced in the `src` attribute which, in the preceding example, instructs the browser to look for a file called `king_kong.jpg` and display it. Without the information in the `src` attribute, the browser has no way of knowing where the image file you want to display is.

The example assumes that your image file (`king_kong.jpg`) is in the same folder (or directory) as your web page.

*We will cover how to link to files and images in different folders when we introduce you to linking and organizing files in the next chapter.*

Although it's possible to include an image using just the `src` attribute as in the short example earlier, we also need to include an `alt` attribute for our page to validate as in the following example:

```

```

Figure 5-1 shows the image of King Kong included on a web page using the preceding markup.

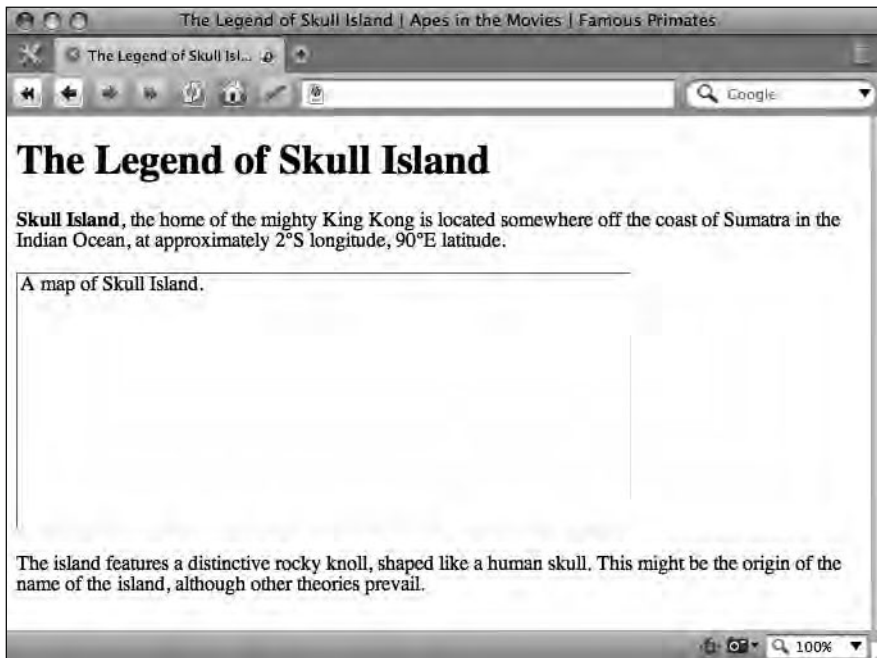


**Figure 5-1.** The image of King Kong included on our page as displayed in a browser

Although the `alt` attribute is not required to *display* our image, it is helpful for visually impaired users, and as such you should get into the practice of using it. Supplying an `alt` attribute is required when creating valid pages and has the added bonus of making your pages more accessible for assistive devices like screen readers. Lastly, the `alt` attribute provides useful text for search engines to index your page.

The text of your `alt` attribute is also displayed in a number of browsers when a user browses your web page with images switched off, for example, by someone browsing the Web on a nonbroadband, dial-up connection or over a mobile phone with a slow connection. Your `alt` attribute will also display in Firefox and some other browsers as the image is being downloaded, useful for images of a large file size that might take longer to download.

It's important to stress that your `alt` text is intended to be used *instead* of your image, and is not intended to provide additional information about the image. The example in Figure 5-2 shows an `alt` attribute in action when images have been switched off.



**Figure 5-2.** Opera, a standards-compliant browser, displaying “A map of Skull Island.”—the contents of our alt attribute as an alternative when images are switched off

When writing alt attributes, try to use meaningful language, for example, alt="The mighty King Kong, a fearsome giant ape." is better than alt="Image", and alt="ACME Widgets - Return to Home Page" is better than alt="Logo". This is particularly important in the latter of these two examples, where an image of a company logo is also serving a navigational purpose. Visually impaired users using assistive software like screen readers rely on this information, so ensure you give your alt text some thought.

Put yourself in the shoes of a visually impaired user using a screen reader and try to employ descriptive language that illustrates the images you’re using. The bottom line is to ask yourself whether the image is a type of content (in which case you should use a descriptive alt attribute) or only serves a decorative or presentational purpose (in which case an empty alt attribute is better).

There may be occasions where an image serves purely decorative purposes, for example, as a decorative banner, where an alt attribute doesn’t contribute much meaning to a visually impaired user. However, in most cases where images serve a purely decorative or presentational purpose, they should be handled with CSS.

If your image *is* purely decorative, it is advisable to use an empty alt attribute, with no space between the quotes, as in the following example:

```

```

*Although an alt attribute can contain up to 1,024 characters (including spaces), it's best to keep your description short and to the point. Again, consider the impact of your alt text on visually impaired users: listening to a lengthy and over-detailed description might prove frustrating.*

So now you know you need to include an alt attribute whenever you add an image to a page. However, there are a number of additional image attributes the true Web Standardista should use. We introduce these in the next section.

## Adding width, height, and title attributes to images

In the last section, we introduced you to the minimum markup required to display an image on a web page. As you'll see here, we're now adding to our earlier example to include some additional information about the image:

```

```

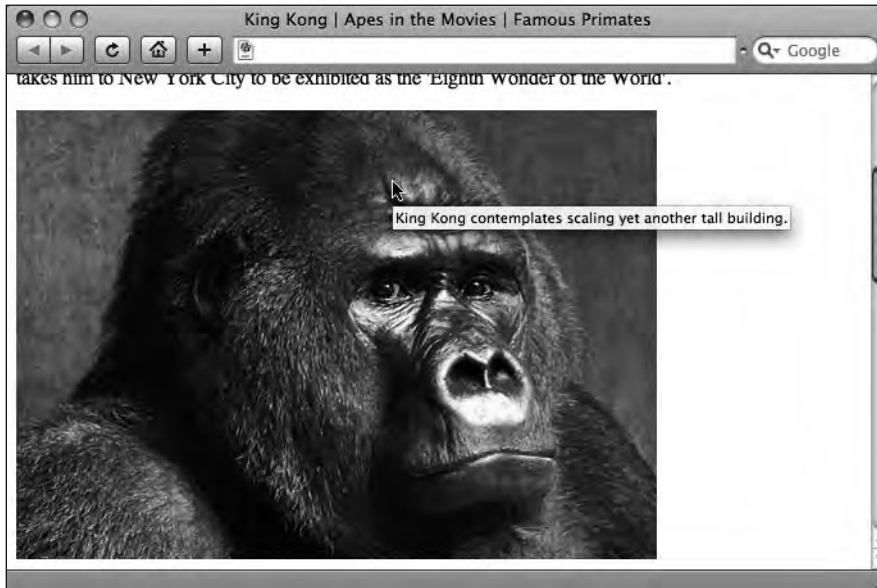
The width and height attributes tell the browser what width and height our image is in pixels. Although these aren't strictly required to display our image, it's good practice to use them. By supplying the browser with the image's width and height, it can allocate space for the image as the page renders, often before the image has fully downloaded; this stops the page from jumping about when loading.

*The word pixel is a 1960s' abbreviation of the term picture element—a useful fact worth storing mentally for your next pub quiz.*

The title attribute (not to be confused with the <title> tag we introduced you to in Chapter 2) is intended for the supply of additional information about our image. In a visual browser, it is usually displayed as a tooltip as in Figure 5-3.

The title attribute is not strictly required, and, if you need to provide additional information about an image, we suggest the use of a caption in a <p> tag beneath the image itself where users can read it without relying on a tooltip. After all, if you're supplying additional information that's important to the understanding of the image, you probably want to make this more accessible within the content of the page itself rather than hidden in a tooltip, which can be read only if a user mouses over an image long enough.

It's important to note that alt and title attributes serve different purposes as outlined previously. The former is intended for the supply of alternative text and is especially useful for accessibility purposes; the latter is intended for the supply of additional information. Don't confuse the two, a mistake beginners often make.



**Figure 5-3.** A browser displaying the contents of a title attribute as a tooltip

Perhaps one reason beginners get confused lies with Internet Explorer, which incorrectly displays the content of the `alt` attribute as a tooltip unless a `title` attribute is specified. This leads many to believe the `alt` attribute is for displaying tooltips. It isn't. It's for providing a text alternative to an image. Think "improving accessibility," not "nice little tooltip."

## Working with images

In this day and age of low-cost digital cameras and easy-to-use scanners, it's easier than ever to find images for your web site. Plug in your digital camera or fire up your scanner, import your images, and go.

In fact, it's a little more complicated than that. As you'll see in the upcoming text, there are a few stages you'll need to go through to create web-friendly images—images that download quickly and, equally importantly, look good. While these stages are important, nothing's too complicated, and we'll walk you through the process of optimizing your images step by step as we progress through the chapter.

We looked previously at how you add images to your web pages using the `<img />` tag and its various attributes. Now we'll explore some practical aspects of working with images, in particular editing them so that they're in the right size and optimized for delivery on the Web. To do this, we'll need to look at some image editors.

## Which image editor? (Or how long is a piece of string?)

Which image editor you choose depends upon a number of factors, primarily driven by what you need to achieve. In this section, we suggest some alternatives that range from paid-for software that allows for a more comprehensive approach to image editing to free software with a few limitations. Which option you choose depends upon both what you need to achieve and what you're willing to spend on software.

In the upcoming text, we outline the benefits of three options: Photoshop, Fireworks, and Photoshop Express, before concluding the section with an introduction to GIMP (GNU Image Manipulation Program) and a list of alternative image editors you might also like to try that range in price from free to low cost.

If you're simply importing photographs from a digital camera, scaling them, and making minor adjustments, Adobe's free web-based Photoshop Express is more than ample for your needs and has a very shallow learning curve. It also provides you with an ample set of tools to make basic image adjustments and, equally importantly, to resize and crop photographs and save them at specific sizes.

If, however, you need a more extensive range of tools, in particular the ability to work with type or nonphotographic images—logos or illustrations, for example—Photoshop and Fireworks are better suited to your needs, as they allow for extensive image creation and manipulation.

We've used Photoshop to illustrate some of our examples; however, if you're approaching the book from a position where access to expensive software is an issue, you can still experiment with one of the lower-cost alternatives we recommend at the end of this section.

The bottom line? The basic principles we cover in this chapter are applicable across the range of image editors available. We're focusing on the fundamentals in this chapter; if you're not using the image editor we're using, don't worry, the principles will remain the same.

### Photoshop

As long as the Web has been in existence, Photoshop has been one of the leading applications for creating and manipulating imagery for web-based delivery. This is one reason why so many alternative image-editing applications follow the majority of its interface metaphors.

Despite its name, Photoshop is useful for much more than extensive manipulation of photographic imagery. It also features a number of tools for the creation of images from scratch—both photographic and *nonphotographic*.

This becomes important as you start to consider including nonphotographic imagery on your web site. For example, when we needed to create the typographic brand for our Famous Primates web site, we turned to Photoshop.

One drawback of Photoshop is price, but as the old saying goes, you get what you pay for; while it's true that Photoshop is expensive, it's also extremely versatile and powerful.



Throughout this chapter, we've used Photoshop to illustrate our examples; however, as we've stated earlier, the basic principles we're covering apply regardless of preferred software.

## Fireworks

A less expensive alternative to Photoshop that offers a slightly different approach, but is certainly worth exploring, is Adobe's Fireworks. Formerly developed by Macromedia, but now part of Adobe's stable, Fireworks is a credible alternative for creating graphics for use on the Web and has a loyal following among web developers.

Fireworks takes a hybrid approach, integrating both bitmap and vector tools in one package, allowing you to create and edit both bitmap images and vector objects with tools that are familiar to both image formats in an integrated environment.

Although the focus of this chapter is on raster images (images made up of pixels), in the section "Vector graphics," we give you a brief introduction to this other image type.

## Photoshop Express

Throughout this book, we've emphasized an approach that embraces the use of free tools, ensuring a low barrier to entry. With this in mind, we felt it important to include an image editor with a price point set at zero. Luckily for you, you can now tap into Adobe's web-based Photoshop Express, 100% free. That's right, an exorbitant \$0.00 (£0.00, €0.00, ¥0.00). Good news indeed. You can access Photoshop Express at

[www.photoshop.com](http://www.photoshop.com)

About this tool, Adobe states "You shot it—now do something to it. Make it pop. Make it impossible to ignore. Upload, sort, polish . . . all for free. Resize, tint, distort, and more—add your mark to all your images."

Believe it or not, this isn't marketing hype; it's true. As broadband has taken off and the Web has evolved, web-based applications like Photoshop Express have emerged as potential competitors to traditional desktop applications.

Despite its nonexistent price tag, Photoshop Express offers a great deal of potential for anyone wishing to experiment with basic image manipulation. Like Apple's iPhoto, the emphasis is on ease of use with one-click options to reduce red eye, adjust exposures, and touch up images. The best way to get a feel for it is to create an account and get started.

It gets better: Adobe has even provided a full set of video tutorials to get you started. You can find these tutorials here:

[www.photoshoponline.com](http://www.photoshoponline.com)

One key limitation of Photoshop Express is that it is an image *manipulation* program, not an image *creation* program. If you need to create images for your web site from scratch, for example, section headers using custom type, you'll need to use Photoshop or Fireworks, or try one of the other programs listed next.

## Bring out the GIMP (and its friends)

Reading the preceding text, you might be mistaken for thinking that Adobe is the only company that offers image editors suitable for web development. Not so. There are a number of alternatives available that are worth mentioning and are also considerably less expensive.

So why have we have used Photoshop to illustrate our examples? The answer is that all of the applications we suggest in this section are built using similar user interface metaphors to Photoshop. All follow very similar approaches to image editing; indeed at first glance their tools and working environments are all very similar.

GIMP, primarily developed for Linux, a platform for which Photoshop was never available, is an open source alternative to proprietary image-editing software. While it has the complexity and many of the features of Photoshop, a number of other image-editing applications offer similar features that are more accessible and beginner friendly.

We’ve listed a number of these alternatives—for a variety of platforms and potential uses—here. These vary in complexity, features, and price; all are detailed at the respective applications’ web sites.

Regardless of the application you choose—free, low cost, or fully featured—we’ve focused on the basic principles of creating imagery for use on the Web throughout the chapter. At the end of the day, the fundamental principles—image size, image resolution, and compression formats, to name but a few—are the important aspects to focus on.

As free or low-cost alternatives to Photoshop and Fireworks, we recommend Pixelmator or Acorn if you’re using a Macintosh, or GIMP or Photoshop Elements if you’re running Windows. If your platform is a flavor of Linux, GIMP will probably not intimidate you; however, Pixel is also worth considering.

You can find links to all of these applications at the Web Standardistas web site:

[www.webstandardistas.com/tools](http://www.webstandardistas.com/tools)

## Image optimization

While you’ve seen that it’s not too difficult to insert images into your web pages, you first need to convert them to a web-friendly format. If you haven’t yet met, welcome to the wonderful world of pixels.

A number of different image formats optimized for web delivery are available to the aspiring Web Standardista, all having different strengths and weaknesses. Picking the right image type for the job is important when designing your web pages. In this section, we guide you through the maze of image types available, highlighting when and where to use different formats (and when and where not to use them).

JPG or, in full, JPEG (Joint Photographic Experts Group), GIF (Graphic Interchange Format), and PNG (Portable Network Graphic)—trust us, you don’t need to remember any of these

full terms—all have different uses. All, however, share one common feature: they allow for varying levels of image compression, resulting in faster image downloads.

## Why compressing your files is important

When delivering images online, compression is important. You've no doubt encountered web pages that take an eternity to download; waiting for the page to load feels a little like watching paint dry. You can help to alleviate this problem, especially when creating a web page that features a lot of images, by using image compression to reduce the amount of information that needs to be transferred from the server to the user's browser. The goal is to reduce the size of your image files as much as possible while retaining as high a quality as possible.

All of the image types we briefly introduced previously allow for varying levels of compression, resulting in faster downloads, and each uses a different compression algorithm suited to a different type of imagery, photographs or illustrations, for example.

The compression of images for delivery online might not seem important now as we enter a phase of broadband delivery, where everything appears to download instantly. However, the web pages we're building in this book are *also* optimized for delivery on nondesktop media, for example, mobile phones and PDAs, where download speeds aren't quite so fast (see Figure 5-4).



**Figure 5-4.** The same web page delivered to a desktop browser and to an iPhone

Mobile phones—the iPhone in particular—have broadened the appeal of the Web beyond desktop computers, and when delivering to mobile platforms, image compression is important. Imagine this scenario: you love your breakthrough Internet communication device, and you use it all the time. You'd like every page you download to be lean and mean and load as fast as possible. Cue compression.

Take a look at the two images side by side in Figure 5-5. To the naked eye, there isn't much difference between the two; however, one of the images is almost six times smaller than the other.



**Figure 5-5.** Two images, barely distinguishable; however, the one to the right is a sixth of the size of the other

Both of the JPG images in Figure 5-5 are saved at different levels of compression. The first image is 160K, the second is 28K. Although the images look almost identical to the naked eye, the 28K image will download almost six times faster than the 160K image. If viewed on a mobile phone, with an expensive data contract, the 160K image will cost almost six times as much to download and view. Clearly, you'd rather be downloading the smaller image.

Efficient compression of images isn't just aimed at browsing the Web via mobile phones, however; it's of importance on any platform. Smaller file sizes result in faster downloads, and this makes for a better user experience, something you should strive for.

In the following section, we cover the benefits of the different image compression formats we've mentioned and identify when and where best to use them.

## Save for Web

One major advantage that Photoshop, Fireworks, and GIMP offer is the ease with which you can optimize images for delivery on the Web. This feature, known as *Save for Web and Devices* in Photoshop, is also available as a *Save for Web* plug-in for GIMP and is built into Fireworks' interface.

As its title suggests, *Save for Web* is designed specifically for optimizing images for delivery on the Web and makes the image optimization process a simple and relatively painless one. The ability to allow real-time previews of images as you alter image formats and compression settings sets Photoshop, Fireworks, and GIMP apart from most other image-editing applications when optimizing images for web delivery.

## Image formats for the Web

Welcome once again to the abbreviation minefield: JPG, GIF, and PNG. All are image formats. All are suited for specific uses. In this section, we cover each image type, introducing you to each format's specific uses.

### JPG: Photographs

As Joint Photographic Experts Group, the full term for JPG, indicates, this format is primarily intended for the display of photographic imagery.

JPG is a lossy format, which means that when you save an image as a JPG, the file doesn't store all the data included in the original image. Instead, the compression format tries to discard any information that isn't needed, removing information that is difficult to distinguish by the human eye.

This compression makes the file size much smaller, but it also means that each time you open, edit, and save a JPG file, the quality of your image reduces. This is one of the reasons it's advisable to work from an original file and save your resized or edited images as copies, leaving the original file untouched. If you later need to make some tweaks to your image, you can start again from your high-quality original.

When optimizing imagery for the Web, you always have to make a choice between image quality and file size. A high-quality image that looks great results in a larger file size, which takes longer to download. Saving your image at a lower quality setting, with more compression, results in a smaller file size, which will download more quickly. When saving a JPG through Photoshop's Save for Web and Devices feature, you can experiment with various levels of compression, either by choosing one of the presets (Low to Maximum) or by manually changing the Quality settings as in the example in Figure 5-6.

Photoshop's Save for Web and Devices feature also discards any additional information about the image that isn't absolutely necessary to display it. In particular, this includes any Exchangeable Image File (EXIF) format data—metadata about the photograph, saved when the photograph was taken—including the date and time a photograph was taken, the camera that was used to take the photograph, shutter speed, and other related information.

Using the 2-Up view as in Figure 5-7, you can compare the visual quality of your original image with your optimized version. As you change the Quality setting to a lower value, you will notice how the file size, displayed below the image, is reduced.



**Figure 5-6.** Photoshop's Save for Web and Devices feature (File ► Save for Web and Devices)



**Figure 5-7.** Photoshop's 2-Up view allows us to compare the quality and file size of our original image alongside the optimized version.

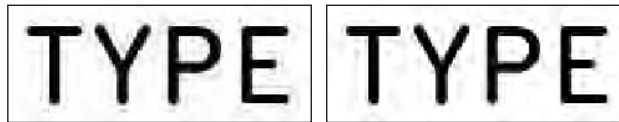


**Figure 5-8.**  
An example of JPG artifacts in a photograph

The trick is to find a good balance between file size and image quality. As each image is different, you will need to use your eye and your judgment to determine where this balance lies. The danger in compressing an image too much is that the compression can become evident through what is known as artifacts—groups of pixels where too much information has been discarded in the compression process—as shown in the example in Figure 5-8.

As we noted earlier, JPGs are perfect for the compression of photographic images. They are not, however, the perfect choice for graphics or type. Let's repeat that because it's important: JPGs are not the perfect choice for graphics or type. If you need to work with type or graphic imagery—a logo or a graphic banner, for example—a far better choice of image type is GIF or PNG. We introduce you to GIFs and PNGs in the upcoming text.

Using the JPG format to compress graphic images, especially type, results in unsightly artifacts—an effect like a halo of pixels—which you can see in Figure 5-9. Not only does this look extremely unsightly, but it's also a definite no-no for the Web Standardista.



**Figure 5-9.** The example on the left shows unsightly artifacts when using the JPG format to compress type. The example on the right, saved as a PNG instead, has no artifacts. The images are enlarged for clarity.

## GIF: Graphics and type

As the full term for GIF, Graphic Interchange Format, suggests, this format is intended for the display of primarily graphic imagery: type, logos, line drawings, and icons, for example.

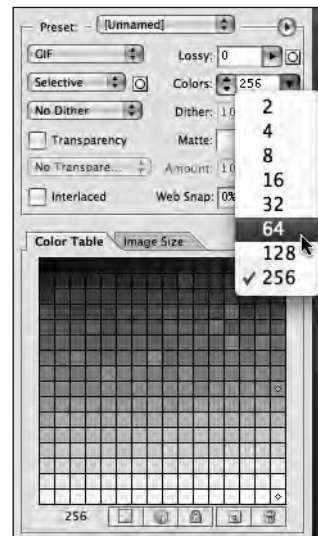
The GIF color palette has a maximum of 256 colors (or 8 bits), a much more limited color palette than the JPG color palette, which is capable of displaying up to 16 million colors. However, GIFs are better suited to displaying images containing large areas of solid color and are perfect for compressing and optimizing logos or type.

GIF is a lossless format, that is, no image data is lost when saving images containing 256 colors or fewer. This means it's possible to resave a GIF over and over again without further loss of data.

GIFs are well suited for images with a limited number of solid colors; essentially the algorithm looks for repeated patterns of pixels and compresses this information. One method of reducing the file size of a GIF is to reduce the number of colors it contains. In Photoshop, the Color Table shows how many colors your image contains. You can adjust this number in the Colors drop-down menu as shown in Figure 5-10.

Again, how many colors you choose to use is a trade-off between file size and quality. A full-color logo saved as a 256-color GIF might look great, but will result in a larger file size. The same logo saved as a 4-color version will probably look too pixilated and blocky. The best result will probably be to reduce the number of colors to somewhere in between, creating an image that looks great but is also a smaller file. As with JPG compression, the trick lies in getting the balance of compression and image quality right.

GIF also allows for dithering of colors, a method of adding patterned or random pixels to the image, which fools the eye by making it appear that the image contains more colors than are actually in the palette. Using dithering allows you to reduce the color palette further, bringing the file size down even more. However, it's worth noting



**Figure 5-10.** GIF Color Table for Photoshop's Save for Web and Devices feature

that if your image contains 256 colors and you need dithering to make the image look reasonable, you might be better off saving the image as a JPG or a PNG-24, which we cover after this discussion on GIFs.

One thing GIFs are not intended for is photographic imagery. With a maximum of 256 colors, photographs do not reproduce well as GIFs. Should you need to compress photographs, use the JPG format.

GIFs have 1-bit transparency, that is, we can pick one color within the image's color table and set it to be transparent, allowing the background on which the image is sitting to show through. However, 1-bit transparency has limitations, especially where we need an image to blend seamlessly with the background as in Figure 5-11.



**Figure 5-11.** The limitation of 1-bit transparency

If blending an image to the background on which it sits is important, we recommend the PNG-24 format; however, it has some limitations as we outline in the next section.

*An additional feature supported by the GIF format is simple frame-by-frame animation. The hottest thing on the Web in 1996, the novelty of animated GIFs has now, thankfully, worn off. However, there might be instances where animated GIFs are suitable, for example, for an advertisement banner or animated logo. More complex animations are better achieved using Adobe's Flash, which is beyond the scope of this book.*

So, to summarize, if you're working with graphics, logos, or text set in a particular typeface, choose GIF or PNG, which we cover next.

## PNG: The new (old) kid on the block

Meet the new (old) kid on the block: PNG, or Portable Network Graphic. The PNG format, introduced in 1997 as a patent-free alternative to GIF, is suited to the same sort of imagery as the GIF format introduced in the last section. However, PNGs offer a number of distinct advantages over GIFs, notably a larger color palette and advanced transparency features.

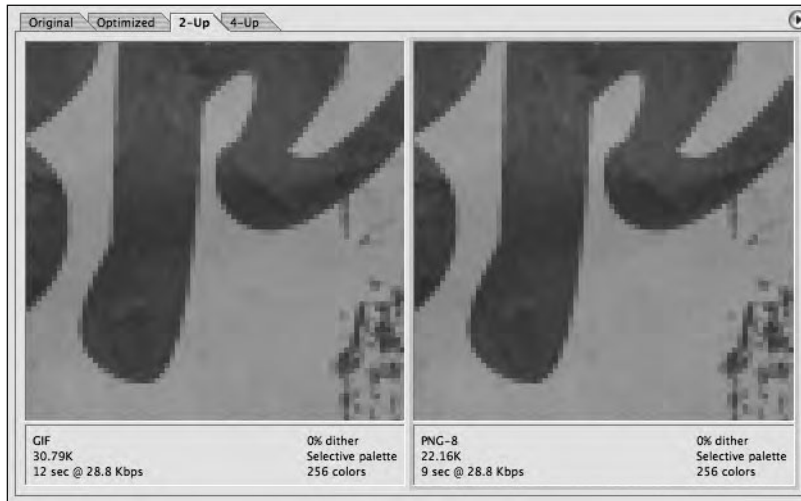
As the advanced features of the PNG format have become more widely supported with the introduction of Internet Explorer 7, PNGs are proving a more desirable alternative to GIFs and are certainly worth considering.



## PNG-8

In practical terms, a PNG-8 is almost identical to a GIF. It has a 256-color palette and is suitable for the same sorts of graphics covered in the discussion on GIFs (i.e., type, logos, line drawings, and icons).

Why use PNG-8 instead of the GIF format? In some cases, PNG-8 may prove more efficient than the GIF format. You can use Photoshop's *Save for Web and Devices* feature to compare sizes and select the appropriate format, as in the example in Figure 5-12. The bottom line: use the image format that looks best at the smallest file size.

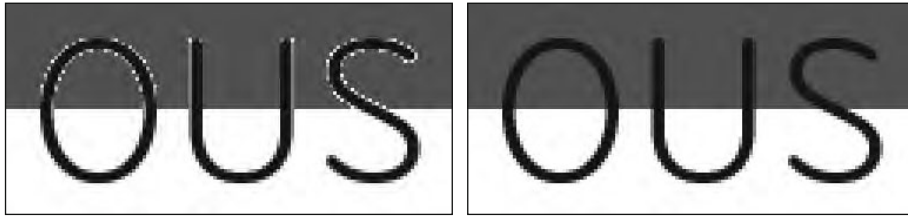


**Figure 5-12.** These two images are barely distinguishable; however, the PNG is smaller in file size.

## PNG-24

Why use PNG-24 instead of PNG-8? The answer is that PNG-24 allows us to display millions of colors compared to PNG-8's limited 256-color palette. It also offers us full alpha transparency. But what exactly is full alpha transparency? Where GIFs or PNG-8s offer 1-bit transparency (i.e., one color is specified as being transparent) PNG-24 allows for more than one color to be transparent, allowing for seamless blends between an image in the foreground and the background of the page.

All this talk of transparency can be a little confusing, so it's best to explain it with an example. Take a look at the two images in Figure 5-13. The image on the left features a PNG-8 image, with transparency limited to a single color. The image on the right features a PNG-24 image, allowing for full alpha transparency. The resulting PNG-24 image allows for a seamless blend between the image in the foreground and the background of the web page.



**Figure 5-13.** Two images, saved using the PNG format. The image on the left is saved as a PNG-8, the image on the right is saved as a PNG-24.

This can be an extremely useful feature when you want a foreground image to blend into its background to create a particular design effect. However, a word of warning: PNG-24 might look like the answer to your dreams, but it results in files that are much larger than PNG-8s. Of more concern, however, is the fact that Internet Explorer versions 6.0 and earlier do not support full alpha transparency, as shown in Figure 5-14.



**Figure 5-14.** A PNG-24 with transparency rendered in Internet Explorer 6 and 7. In IE 6, on the left, the image background isn't transparent.

Given that IE 6 and its previous incarnations are old browsers being superseded by IE 7 and IE 8, this lack of support for transparency in PNG-24 images will hopefully be consigned to the dustbin of history. We felt it important to warn you of this browser shortcoming nonetheless.

*A word of warning: images formatted as PNGs can sometimes cause display problems. A number of browsers, Internet Explorer 7 included, will not always exactly match a foreground image in PNG format with a background color specified in the CSS on which they are sitting. The result can prove frustrating with an image and a background color not quite matching up. If this specific scenario poses problems for you, we suggest saving your image as a GIF or JPG instead.*

## A pixel is a pixel is a pixel

You might have encountered discussion about the relative merits, or lack of, of saving your images for the Web at different resolutions, or dots per inch (dpi). Different authors will sing the praises of 72 dpi over 96 dpi, or vice versa. In fact, the resolution of your images makes no difference onscreen whatsoever. What does make a difference, however, is the dimensions—in pixels (px)—you generate your images at. Most image editors default to 72 dpi when creating new images—it's perfectly safe to leave this setting alone.

### Size matters

We've looked at image compression to reduce the file sizes of your images. Another aspect of your images to consider is their actual dimensions—their width and height—in pixels. Although it's possible to resize images using the width and height attributes within the `img` element, it's not recommended.

Consider the following scenario. We have a JPG image of Cornelius, the noted chimpanzee archaeologist and historian. The original image supplied is  $1600 \times 1000$  px; however, that's too large to fit into the layout we've designed. We really need the image to be  $400 \times 250$  px to fit. In a moment of laziness, we decide not to scale the image down in our image editor and save a new smaller version. We opt instead to scale it down using the width and height attributes as in the following example:

```

```

Perfect. But is it? This will instruct the browser to scale down the original, large image and display it at a width of 400 px and a height of 250 px—just the size we want. While this might be convenient, there are a number of obvious downsides to this approach.

Forcing the browser to scale down a large image, while possible, results in longer download times and can also cause images to distort badly in many browsers. A better approach is to resize your images using an image editor so that they're sized to the width and height you need them to display.

Consider again browsing the Web on a mobile device with an expensive data plan. Scaling a  $1600 \times 1000$  px image to  $400 \times 250$  px forces users to download 16 times as much data, 16 times the cost—money they'd rather be spending on a mocha-choco-latte.

*You might think that the larger image is only four times bigger, but you'd be mistaken. The large image comprises  $1600 \times 1000$ , or 1,600,000 pixels of data; the small image comprises  $400 \times 250$ , or 100,000 pixels of data, 1/16th smaller.*

## The limitations of bitmaps

One of the downsides of using bitmap images that are by nature resolution dependent is the fact that they don't scale up very well. Scaling up bitmap images usually results in a loss of detail and quality. For this reason, when working with images, it's important to work with original images where possible, scaling them down to the size you require them at.

Take a look at the two close-ups of the mighty King Kong in Figure 5-15. The original on the left at  $200 \times 200$  px is fine for use on the Famous Primates web site. When we scale it up to  $800 \times 800$  px, however, it loses detail and quality.



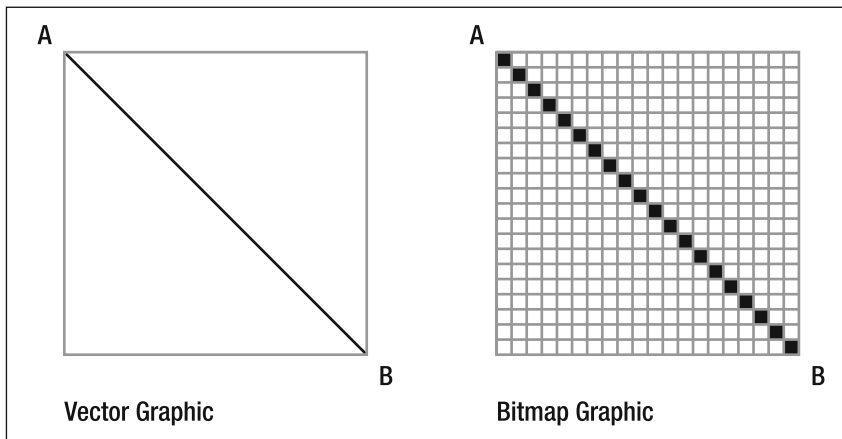
**Figure 5-15.** Scaling up a bitmap image results in a loss of image quality.

Unlike bitmap images, vector graphics can be scaled independently of resolution, as we cover in the following section.

## Vector graphics

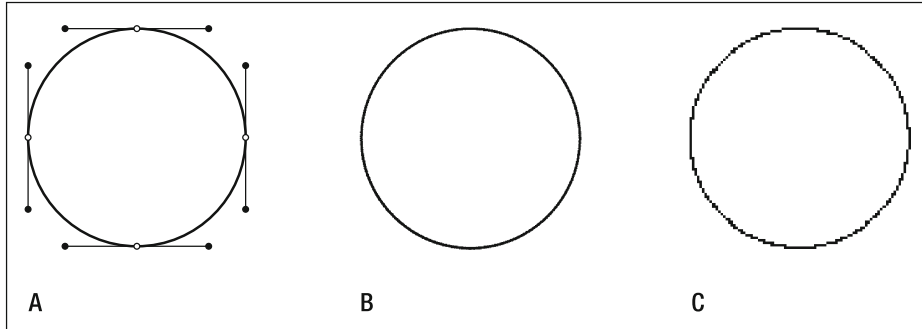
Unlike photographs or other bitmap images, vector graphics do not store pixel data; instead they describe images as a series of mathematical formulae using what are known as Bézier curves or paths.

Let's look at a simple example. Imagine you wanted to draw a line from the top-left corner to the bottom-right corner of a piece of squared paper. A bitmap image, which is essentially a grid of pixels, would draw this line using a series of pixels at a specified resolution, as in the example on the right in Figure 5-16. A vector image, however, would simply specify the start and end points of the line, irrespective of resolution.



**Figure 5-16.** A vector graphic compared to a bitmap graphic

One advantage of vector graphics is that they aren't limited to the size they were created at like bitmap graphics are, as shown in Figure 5-17. Simple sets of instructions—draw a line from point A to point B, or connect four points to create a circle—vector graphics, which are resolution independent, can be scaled up or down to your heart's content. The instructions that make up the image remain the same, regardless of the size it is displayed at. As a result, vector images are often smaller in file size than bitmap images.



**Figure 5-17.** Vector images (A) are resolution independent, while bitmap images (B–C) are tied to the resolution they were saved at.

Vector graphics are often used for the creation of illustrations, logos, or typographic headers. Adobe Illustrator is specifically designed for the creation of vector graphics, although both Fireworks and recent versions of Photoshop have support for vector-based image creation.

One point worth noting is that vector images cannot be added to a web page using the `<img />` tag; they need to be saved in a rasterized (bitmap), web-friendly format first in order to be displayed as an `img` within a browser.

*Scalable Vector Graphics (SVG) is an open standard for displaying vector graphics on the Web. Support for this format is still uneven, with no native support from Internet Explorer.*

## Finding the right image

Perhaps your photography isn't the best in the world, or you'd like to use an illustration, but your drawing skills aren't exactly up to scratch. Have no fear, there are a number of low-cost image resources on the Web sure to offer what you're looking for. In this final section, we introduce you to a few, paying particular attention to iStockphoto ([www.istockphoto.com](http://www.istockphoto.com)) and stock.xchng ([www.sxc.hu](http://www.sxc.hu)), both resources we regularly use. Indeed, iStockphoto is the source of all of the famous primates photographs used in this book.

### Low-cost images

Companies like iStockphoto and stock.xchng offer a variety of high-quality, but free or low-cost, images, perfect for most needs. If your budget is tight or even nonexistent you're likely to find something at either iStockphoto or stock.xchng that will suit your requirements without breaking the bank.

Allowing emerging photographers and illustrators to share their creative work with others, these sites have built up considerable libraries of images available in a variety of resolutions, ideal for use online. Easy to search and offering a huge variety of styles of photos that are, best of all, free or low cost—iStockphoto and stock.xchng are a perfect first port of call when you're looking for a specific image.

### Flickr and Creative Commons

Another option worth considering when looking for images is Flickr ([www.flickr.com](http://www.flickr.com)). Many users of this popular photo-sharing site have licensed their photographs using what's known as a Creative Commons license, which allows you to use them freely under certain conditions.

Creative Commons ([www.creativecommons.org](http://www.creativecommons.org)), established in 2002, offers an alternative to full copyright, allowing content owners and content creators to offer their works under less restrictive licenses.

What this means in practice is that instead of an image having all rights reserved, making it illegal to use without permission, only some rights are reserved. For example, the creator allows you to use their image as long as you give them credit, or the author allows you to use the image for noncommercial projects only. In essence, Creative Commons offers a variety of different license models designed to expand the range of creative works available for others to build upon and to share legally.

Links to all of the preceding web sites and a variety of other online image resources can be found at the book's companion web site:

[www.webstandardistas.com/resources](http://www.webstandardistas.com/resources)

## Summary

So what have we covered? In this chapter we've covered the fundamentals of adding images to the pages of your web site. We've introduced you to the `img` element, which is required to add images to a web page along with its various attributes. We've also introduced a number of image-editing applications and a variety of image formats designed for web delivery. Lastly, we've provided an overview of how to create imagery optimized for delivery on the Web.

In the next chapter we introduce you to links, the elements that make the Web the Web.

5

## Homework: A picture is worth a thousand words

Over the last three chapters, you've created three web pages of progressively varying complexity for Albert I, Miss Baker, and Gordo. As you've worked your way through the homework, we've added a variety of elements to enable you to build web pages that have become progressively richer.

This chapter's homework is to add images to your three monkey pioneer pages.

*We've worked with iStockphoto to enable us to supply you with photographs of all of our famous primates for your homework. Before you download and use these images, here's the small print!*

*Although we're providing you with these images to enable you to undertake each chapter's task, the images remain the copyright of iStockphoto and may not be used for other purposes. iStockphoto is happy for us to offer you these images for educational purposes; however, should you plan on making your Famous Primates web site publicly available, you will need to purchase licenses for the images at a cost of just a few dollars. (Or, alternatively, replace them with images of your own.) We've put a Read Me! file containing the licensing details in the folder with the images.*

You'll be adding two images to each of your three monkey pioneer pages: at the top of the page, above the opening `<h1>`, you'll add the Famous Primates brand, and beneath the text that introduces each monkey you'll add an image of that particular simian.

Do we need to remind you to validate your pages when you're done? We thought not . . .

Once you've added the images to your web pages, the second part of your homework will involve researching some of the image editors we suggested in the chapter.

### 1. Get the images

As mentioned previously, to make your task a little easier and to save you having to track down all the images you'll need, we've added them all to a folder that we've compressed and zipped for you to download.

You can download the folder here; don't forget to read the Read Me! file:

[www.webstandardistas.com/05/primates.zip](http://www.webstandardistas.com/05/primates.zip)

### 2. Add your img elements

Using the examples provided in the chapter, add the Famous Primates brand and the illustrations of Albert I, Miss Baker, and Gordo to your web pages.

As usual, we've been working on the apes' side of the web site and have created a similarly structured page about King Kong, the Eighth Wonder of the World. You can refer to this, using your browser's *View Source* command to see how we've included our images, here:

[www.webstandardistas.com/05/king\\_kong.html](http://www.webstandardistas.com/05/king_kong.html)

### 3. Test!

When you've completed the three pages, check them in your browser to ensure everything's working as expected.

### 4. Research

Research some of the image editors recommended earlier in the chapter. Remember, we've provided a link to some recommended tools at the Web Standardistas web site:

[www.webstandardistas.com/tools](http://www.webstandardistas.com/tools)

Take a look at what each image editor offers and give Adobe's Photoshop Express a whirl; it won't even cost you a penny.

That's it! After you've explored the wonderful world of image editors, put the kettle on and enjoy a cup of *Chifir*' as you prepare yourself for the next chapter.