6.9 Bitmap Image Formats: GIF, JPEG, PNG

So far, you've seen examples where if the user clicks a link, the browser opens a new page. It's also possible to create a link where if the user clicks it, the browser opens an image. Usually, "opening an image" causes the image to fill the entire browser window. Normally, that's inappropriate, but as you'll learn in the next chapter, you can use an `iframe` element to create a window "frame," which allows you to open/display an image within just the frame part of the window. But before we talk about the `iframe` element in the next chapter, first we need to talk about image file formats and `img` element details. That will take a while, and the rest of this chapter is devoted to that.

There are two basic categories of image files—bitmap image files and vector graphics files. We'll have more to say about vector graphics files soon enough, but for now we'll focus on bitmap image files. With bitmap image files, an image is comprised of a group of pixels. For example, an *icon*, which is simply a small image file, typically has 16 rows with 16 pixels in each row. Within a bitmap image file, every pixel gets mapped to a particular color value, and each color value is a sequence of bits (where a bit is a 0 or a 1[6]). For a browser to display a bitmap image, it displays each pixel's mapped color. This reliance on mapping color bit values to pixels is the basis for the name *bitmap image*.

The three most common formats for bitmap image files (also called raster image files) on the Web are GIF, JPEG, and PNG. You can see brief descriptions of those formats in **FIGURE 6.9**. We'll provide more details shortly, but first, you should be aware of two other file formats—BMP (for bitmap) and TIFF (for tagged image file format). They're both very popular for graphics applications, but they're generally not used with web pages. Why? BMP files are too large, and TIFF files cannot be viewed by web browsers without a plug-in.[7] Since we're focusing on web pages, we'll refrain from providing details about them.

---

[5] According to my middle school and high school daughters, vlogger Bethany Mota is the coolest, barely twentysomething in the whole world. Formerly bullied, she's now a *motavatour* megastar.

[6] For the purpose of this discussion about bitmap images, all you need to know about bits is that they are 0's and 1's. But from a computer hardware perspective, 0's and 1's are high-energy signals versus low-energy signals. When a computer generates a low-energy signal, that's a 0. When a computer generates a high-energy signal, that's a 1.

[7] A *plug-in* is a software component that adds functionality to an existing software application, like a browser.

Bitmap Image Formats	Description
GIF	Good for limited-color images such as line drawings, icons, and cartoon-like illustrations.
JPEG	Good for high-quality photographs.
PNG	Flexible; good for limited-color images and also high-quality photographs.

FIGURE 6.9 Common formats for web page bitmap image files

GIF Image File Format

Let's begin our discussion of image file formats with GIF files. GIF files use a filename extension of `.gif`. GIF stands for Graphics Interchange Format. Because the G in GIF stands for "graphics" and graphics is pronounced with a hard g, many people in industry (most?) pronounce GIF files with a hard g, so it sounds like the "gif" part of "gift." But don't jump on that bandwagon just yet. For a full pronunciation explanation, see http://www.olsenhome.com/gif. That web page points out that CompuServe, the company that invented the GIF format, documented GIF's pronunciation as "jif," with a soft g, like the peanut butter.[8] If you're reading this book without a teacher, we recommend that you pronounce GIF files as "jif." However, if you've got a teacher, you should follow your teacher's lead.

In creating a GIF file from an original picture, the original picture's colors are mapped to an 8-bit palette of colors. That means each pixel uses 8 bits, and those 8 bits determine the pixel's color. And the entire set of colors forms the image's *color palette*. Each image has its own color palette with its own set of colors. So for the peanut butter image seen here you'd need to capture colors such as red, brown, and blue. Each of those colors would have an 8-bit sequence associated with them, such as 01011010 for red, 11011001 for brown, and 00010110 for blue.

With 8 bits for each color value, there are 256 different ways to arrange the 0's and 1's. You can prove this to yourself by writing all the different permutations of eight bits, or you can just remember that the number of permutations is equal to 2 raised to the power of the number of bits, where $2^8 = 256$. That means each GIF image file can handle a maximum of 256 distinct colors. If a picture has more than 256 distinct colors, then when creating the GIF file from the picture, some of the colors won't be stored accurately. Instead they'll be stored with similar colors that are part of the GIF file's color palette, and that leads to the GIF file's image being a degraded version of the original photograph. Color degradation is nonexistent or imperceptible for limited-color images such as line drawings, icons, or cartoon-like illustrations, and that's why GIF files are good for those types of things and not good for color photographs.

[8] A comment from one of that web page's readers is particularly insightful: "No *decent* coder would pass up an opportunity to inflict a horrid pun on the world. And seeing as peanut butter is one of the principal three programmer foods (the other two being Pepsi and nacho cheese Doritos), the reference is immediately obvious."

By using only 8 bits of storage for each pixel, GIF files are able to achieve relatively small file sizes. That's the main benefit of using the GIF file format—small file sizes. File sizes can be reduced further by applying a compression scheme. The GIF file compression algorithm is *lossless* because it does not introduce any color degradation. But be aware that the compression is applied to the GIF file's 256-color-palette image, not to the original picture. As already mentioned, if the original picture has more than 256 distinct colors, then there is indeed information loss in going from the original picture to the GIF file image.

A fun feature of the GIF file format is that it supports simple animation. In rendering an *animated GIF*, the browser displays a sequence of image frames in quick succession. Normally, animated GIF files are configured to display their frame sequences in continuous loops rather than only one time.

JPEG Image File Format

Next up, the JPEG image file format. JPEG stands for Joint Photographic Experts Group, and JPEG is pronounced "jay-peg." JPEG files use a filename extension of `.jpeg` or `.jpg`.

In creating a JPEG file from an original picture, the original picture's colors are mapped to a 24-bit palette of colors. With 24 bits, there are approximately 16 million permutations of 0's and 1's in each color value (2^{24} = 16,777,216). That means approximately 16 million unique colors can be represented, and that's more colors than the human eye can discern. So unless you're an eagle with the ability to distinguish between more than 16,000,000 colors, you should be in good shape with the quality of colors in JPEG files. In other words, for humans, there is effectively no information loss in 24-bit palette JPEG images.

Note this picture of the New York Catskills' famed fall foliage:

Photo courtesy of Allegany County, New York.

Fall on Alma Pond in the Catskills

With the picture's wide variety of colors, a GIF file would have to approximate many of the colors in order to conform to the constraints of its relatively small color palette. But with the JPEG format, all the photograph's original colors can be displayed accurately.

In attempting to produce optimized JPEG files, "lossy" compression schemes are applied. The person who creates the JPEG image file chooses the lossiness of the compression scheme—higher loss means greater color degradation and greater file size reduction. With that in mind, JPEG images can vary widely in terms of color degradation and file size. Nonetheless, for photographs, optimized JPEG files almost always produce a better balance between file size and quality than the other file formats.

PNG Image File Format

PNG stands for Portable Network Graphics, and PNG is pronounced "ping." PNG files use a filename extension of `.png`.

The PNG format was invented in 1996 as an open-source alternative to the GIF format because the GIF format was copyright protected with a patent owned by Unisys. So each time someone made a new GIF file, they were supposed to pay a license fee to Unisys. Oftentimes, GIF file creators didn't bother to pay the license fee, which was illegal. To avoid such illicit activity, the web community eventually invented their own open-source format for image files, and the PNG format was born. After the PNG format's inception in 1996, the GIF copyright expired, so it's now legal to create and use GIF files without fear of retribution. Although the PNG format no longer serves as a licit alternative to the illicit use of the GIF format, the PNG format remains very popular, as it improved upon the GIF format in several ways.

The PNG format provides more flexibility in terms of clarity versus file size. In creating a PNG file from an original picture, you can choose to map each pixel to only a few bits (to create an image file that doesn't require much storage) all the way up to 64 bits per pixel (to create an image file that is very clear). As mentioned earlier, only 24 bits are needed to represent approximately 16 million unique colors, and that's all that most humans can discern. The additional bits for 64-bits-per-pixel PNG files are for things like multiple channels and opacity/transparency levels. Those details are quite technical, and they're beyond the scope of this book.

The PNG format provides more flexibility in terms of transparency. You can create images with different levels of transparency for different parts of an image. GIF images can have only two levels of transparency—completely opaque or completely transparent. PNG images can have 256 levels of transparency.

Note the web page in **FIGURE 6.10**. In particular, note the yellow stucco background behind the aquarium.[9] The background comes from the web page and not from the image file. You can't see the image's original area outside of the aquarium (and the image's rectangular edges) because it's covered over by the web page's background. The browser is able to "cover over" that area because the PNG file is configured to be transparent there. The PNG format allows that sort of complete transparency, but it also allows partial transparency. In the figure, notice the partially transparent glass near the top of the aquarium and the partially transparent water within the aquarium. Note how the glass area is more transparent than the water area. That should make sense when you realize that the glass area is filled with air, which is clearer than the murky aquarium water.

Learning how to create and edit images and add different levels of transparency to them is beyond the scope of this book, but if you're interested, feel free to learn about such things on your own. To get started, you could purchase one of the popular graphics software packages, such as

[9] The yellow stucco might not be the most attractive, but the web page takes its cue from the official PNG website. The PNG authors are known for their creativity and attention to detail...not so much for their fashion sense.

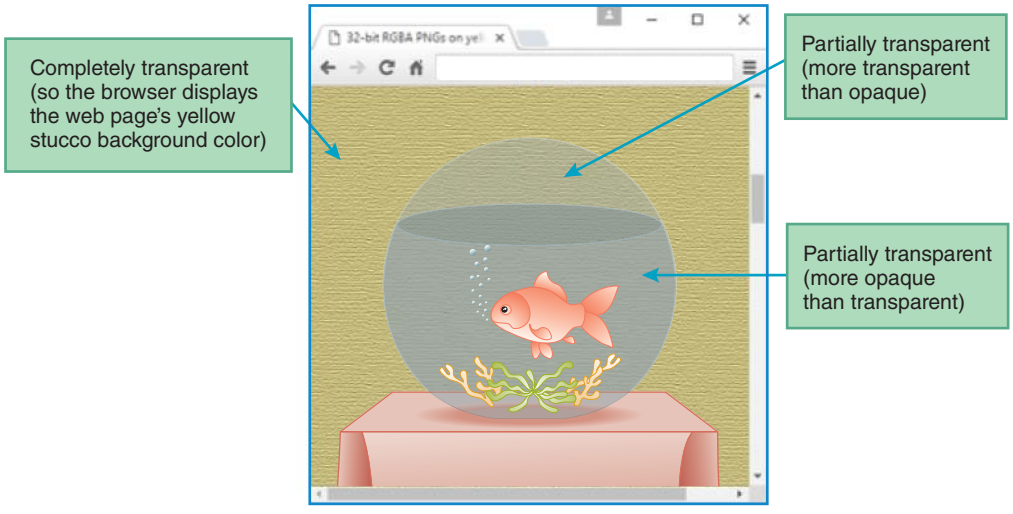

FIGURE 6.10 Aquarium PNG image with multiple levels of transparency

Adobe Photoshop or Corel PaintShop Pro. Or as an alternative, you could download and use the GIMP graphics software package for free.

Image Format Comparison

When you implement a web page with an image(s), you'll need to decide which image format is most appropriate. There are no absolute rules. What follows are guidelines.

If your image is a photograph, you should normally choose the JPEG format because a JPEG file will yield a higher quality image, and it will tend to be smaller in size. How can a JPEG file be smaller than a GIF file when a JPEG file uses 24 bits for each color and a GIF file uses 8 bits for each color? For photographs, the JPEG compression scheme is much more effective than the GIF compression scheme. The JPEG compression often reduces the size so it's less than the size of a comparable original GIF file and even less than a compressed version of the GIF file.

For limited-color images such as line drawings, icons, and cartoon-like illustrations, choose the GIF or PNG format because that will lead to smaller files than if the JPEG format is used. The PNG format is flexible—in addition to being a good choice for limited-color images, it can also be used for high-quality photographic images.

As mentioned earlier, the GIF format supports simple animation. The PNG format does not support animation, but in 2001, an offshoot of the PNG working group formally introduced an extension to the PNG format called MNG that does support animation. Here are a few advantages that MNG files have over animated GIF files:

- Significantly improved compression
- Ability to move images relative to the other images in the animation's sequence of images
- Ability to use not only PNG images, but also JPEG-oriented (JNG format) images

6.10 `img` Element

Now that you know about image file formats, it's time to focus on the nuts and bolts of how to display an image on a web page. In Chapter 4, we introduced the `img` element, which displays images. In this section, we'll review what you learned and provide more details.

Let's jump right into an `img` element example:

```
<img src="../images/winkingSmiley.gif"
  alt="Winking Smiley Face" width="50" height="50">
```

The `src` attribute specifies the image's filename and the location of the image's file. In this `img` element, the filename is `winkingSmiley.gif`. The location is indicated by the path that comes before the filename. The path that comes before `winkingSmiley.gif` is `../images/`. Do you remember what the `..` is for? The `..` says to move up to the parent directory of the current directory. After `..` comes `/images`, which says to go down to the `images` directory. That's where the `winkingSmiley.gif` file is supposed to reside.

Quick quiz: How would you specify `src`'s value if `winkingSmiley.gif` was in the same directory as the current web page? Before reading on, try to figure it out for yourself. The answer is… `src="winkingSmiley.gif"`. The key is to realize that if the image file and current web page are in the same directory, then there's no need for a path.

Referring back to the `img` code fragment, note the `alt` attribute and its `"Winking Smiley Face"` value. The HTML5 standard states that the `alt` attribute is required. The `alt` attribute's value should normally be a description of the picture. The `alt` stands for "alternative" because it provides an alternative for displaying the image in case the image is unviewable. Formally, we refer to the `alt` attribute as a *fallback mechanism* because it provides content for the browser to fall back on if the image can't be displayed. That content is referred to as *fallback content*.

Earlier, we said the image might be unviewable. There could be a problem with the file, such as it might not exist, it might be corrupted, or it might be in a different location than what's specified in the `src` attribute's path. Other reasons for not being able to view a file have nothing to do with the file; they have to do with the user. To make web page downloads faster, users might disable the ability to download images on their browsers. Some users are visually impaired and use screen readers to process web page content. Screen readers ignore displayable things like images and instead rely on the fallback content to know what to read aloud to the user.

Browsers are free to use the `alt` fallback content as they see fit. Before HTML5, prior versions of HTML specified that the `alt` attribute's value should be displayed as a *tooltip* (text that pops up when the user hovers the mouse over an item), but that's no longer the case. With HTML5, it's the `title` attribute that is used for tooltips. Well, formally, the HTML5 standard says the `title` attribute is for "advisory information," and that has led browsers to use the `title` attribute for tooltips. To avoid redundancy, current browsers no longer use the `alt` attribute for tooltips. Instead, they use the `alt` attribute as fallback content for things like screen readers attempting to render `img` elements.

An `img` element's `width` and `height` attributes specify the image's size in pixels. In implementing an `img` element, you should find the image's actual width and height values and use those values for the `img` element's width and height attributes. Perhaps the easiest way to find an image's actual

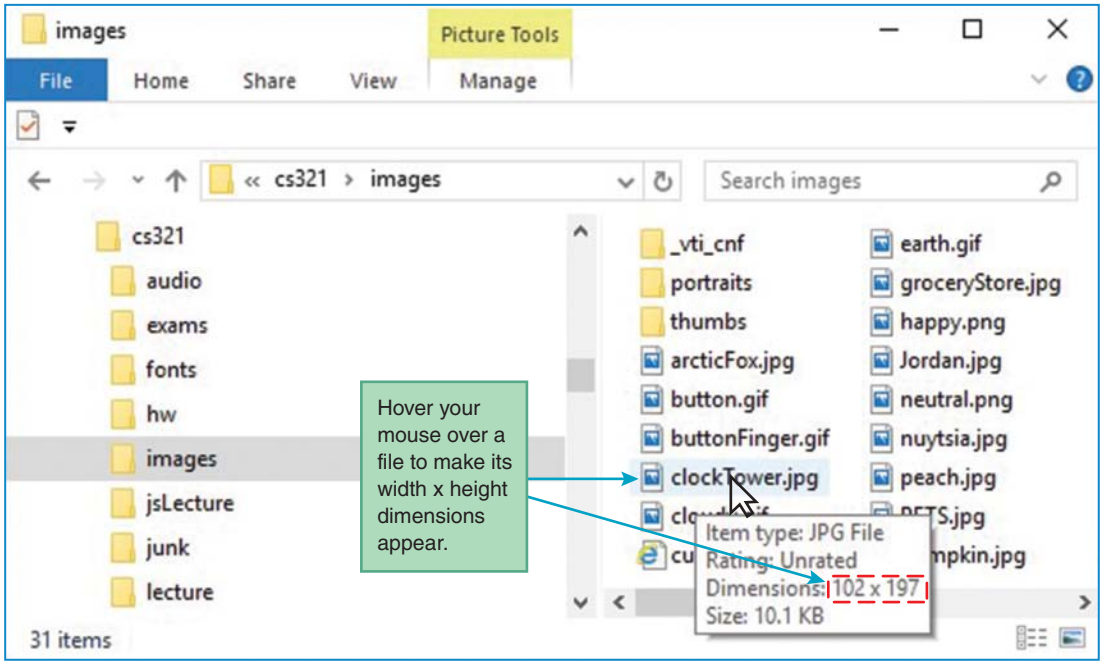

FIGURE 6.11 How to find pixel dimensions for an image file

width and height values is to find the image file in File Explorer, hover your mouse over it, and look for its dimensions. FIGURE 6.11 shows how that's done for the Clock Tower web page's image file.

If you omit the `img` element's `width` and `height` attributes, then the image will display with its natural size, which is a good thing. But specifying its natural size explicitly is better because that improves download speed. How is that, you ask? The size and position of the image affects the layout of the other items on the web page. For example, the width of an image determines whether the subsequent item on the web page gets positioned at the right of the image or on the next row. If the image's dimensions are known early during the web page download process, then those dimensions can be used to display the web page's other items in their proper positions. On the other hand, if the image's dimensions are not specified, then the browser has to wait for the image file to be downloaded and displayed before it knows where to position the other items. Avoiding the wait can be very helpful because, typically, image files (and audio and video files, covered in the next chapter) take much longer to download than other items on the web page. If you specify width and height values that don't match the image's original pixel dimensions, the picture will be displayed with the specified dimensions, but the quality of the resulting picture normally will degrade. So try to always use the correct dimensions.