

5

# Creative Use of PNG Transparency in Web Design

**jeff**  
*croft*

[www2.jeffcroft.com](http://www2.jeffcroft.com)

**Jeff Croft** is a web and graphic designer focused on standards-based development who lives and works in Lawrence, Kansas. As the senior designer at World Online, Jeff works on such award-winning news sites as [www.lawrence.com](http://www.lawrence.com) and [www.ljworld.com](http://www.ljworld.com). Jeff also runs a popular blog and personal site at [www.jeffcroft.com](http://www.jeffcroft.com), where he writes about many topics, including modern web and graphic design.

Jeff has been making the Web about as long as there has been a Web to be made. He created his first web pages in 1994 using SimpleText and Netscape Navigator 1.1N on a Macintosh Performa 600. He started working on the Web full time in 1996 and has been at it ever since.

Although Jeff enjoys technology and gadgets, what truly inspires him about the Web is its ability to communicate and connect people. Jeff is a constant consumer of design, finding inspiration nearly everywhere.

When he's not hunched over a computer, Jeff enjoys photography, music, film, television, and a good night out on the town.



# PNG, GIF, and JPEG

The PNG image has been widely overlooked by the web design community—and mostly for good reason. Until recently, it hasn't been possible to take full advantage of the format and have it work reliably in all browsers. But, with proper PNG support in Internet Explorer 7, and some handy JavaScript and CSS tricks to account for older browsers, we can use PNG images to greatly enhance our design vocabulary.

## What is PNG?

PNG, usually pronounced “ping,” stands for Portable Network Graphics. It is a losslessly compressed bitmap image format. In plain English, it's a way of saving graphic images that reduces file size without reducing image quality. It was originally created as a replacement for the ubiquitous GIF format, which used to require a patent license for producers of imaging software to use it legally (the GIF/LZW patent has since expired, so this is no longer a factor). PNG is also an International Standard (ISO/IEC 15948:2003) and an official W3C recommendation ([www.w3.org/TR/PNG/](http://www.w3.org/TR/PNG/)).

Besides being a freely available format, PNG offers several practical advantages over GIF for the web designer:

- **Greater compression:** For most images, PNG achieves a smaller file size than GIF.
- **Greater color depths:** PNG offers truecolor up to 48 bits, whereas GIF allows only 256-color palettes.
- **Alpha-channel transparency:** Whereas GIF offers only binary transparency, PNG allows for virtually unlimited transparency effects by enabling an alpha channel for transparency.

It's worth mentioning that PNG does not allow for animation, as GIF does. There is a related standard called Multiple-image Network Graphics (MNG, [www.libpng.org/pub/mng/](http://www.libpng.org/pub/mng/)) that does allow both, but it is not widely supported by web browsers or imaging software.

## So why is GIF still so popular?

You're probably wondering why PNG isn't the most commonly used image format on the Web, if it's as good as advertised. The answer, for the most part, lies in misconceptions about the format and the browser support for it.

Because Internet Explorer 6 and lower do not support the full spectrum of PNG's features (including alpha-channel transparency), people seem to believe (albeit erroneously) that Internet Explorer doesn't support PNGs at all, or at least doesn't support transparency. In reality, Internet Explorer 5 and 6 both support enough of the PNG spec to make PNG images functionally equivalent to (or better than) nonanimated GIF images. All other notable browsers—including Firefox, Netscape 6 and higher, Mozilla, Opera 6 and higher, Safari, and Camino—offer full support for PNG transparency.

Besides the misconceptions about browser support, GIF's built-in support for animation was (and continues to be) a key reason for its success. In recent years, however, this use of GIF has become less popular as other technologies (notably Flash) have become more common for animation.

Transparency is a key feature of both GIF and PNG that is often the reason either one gets chosen as a web designer's format of choice for a particular image element. Although PNG offers far more extensive support for transparency, web designers are often required to create GIF versions of the images as well to accommodate older browsers. Using CSS, it's possible (and somewhat commonplace) to send GIF images to older browsers and higher-quality PNGs to browsers that understand them. But the creation of two images is extra work for the web designer, and this has often resulted in people settling on the lowest common denominator, which continues to be GIF images.

In the end, there are several reasons why GIF is still so popular, but most of them are based on misconceptions or use case scenarios that are becoming less and less common. Armed with some new knowledge of how PNG works and how it can be used reliably across browsers, you should be able to take advantage of all that the format offers without having to rely on GIF.

## What about JPEG?

JPEG, the Web's other ubiquitous file format, is almost always a better choice than either PNG or GIF for photographic (or photo-like) images. PNG was not intended to compete with JPEG. JPEG's lossy compression (which results in some reduction in quality each time the image is saved) will produce considerably smaller files than PNG when dealing with photos. PNG, on the other hand, will produce smaller files when the images within are text, line art, logos, flat colors, and so on.

## Some great uses for the humble PNG

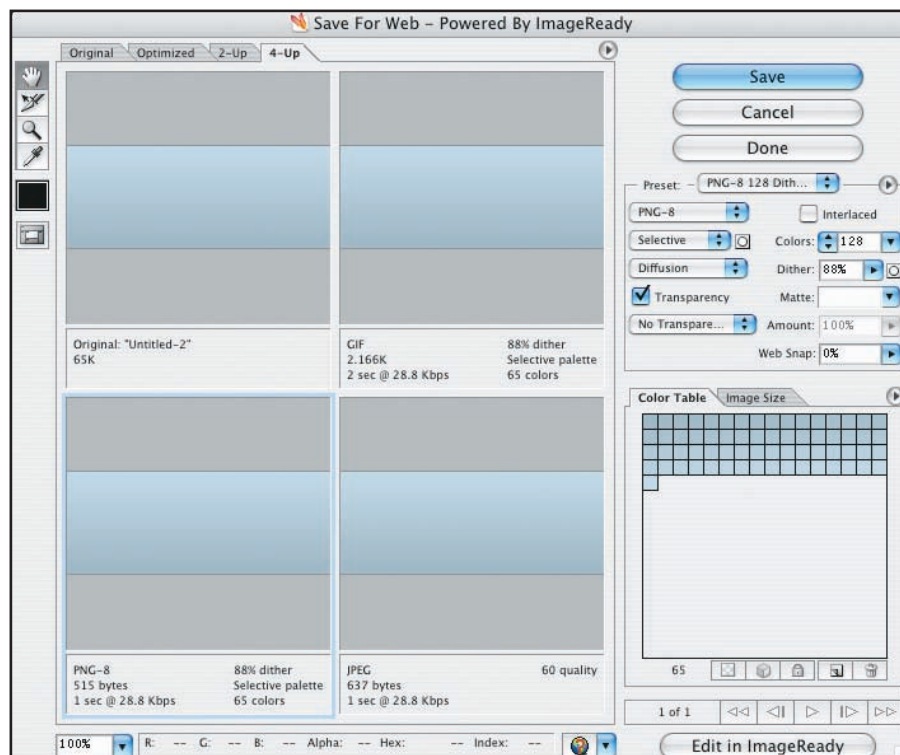
Now let's look at some great uses for the PNG in web design. I've included all the files for each example in a separate folder inside the code download for this chapter, available at [www.friendsofed.com](http://www.friendsofed.com).

### The gradient

In the past few years, the gradient—a smooth transition between two or more colors—has become the web designer's best friend. Especially popular is the subtle, barely noticeable gradient fill, which adds a feeling of depth and texture without being overt and cheesy.

GIF is sometimes a good choice for gradients. If the gradient is a simple two-color fade, GIF usually works just fine. However, the GIF 256-color limit often causes noticeable and unsightly banding across more complex gradient transitions. JPEG, on the other hand, can render quite pleasing gradients, but often at the cost of a higher-than-desired file size. And while JPEG gradients are usually “good enough,” keep in mind that JPEG does use lossy compression, which means the reproduced image is never as high fidelity as the original, uncompressed image.

Consider the typical background gradient style often used for buttons, boxes, and just about anything else. It may look something like Figure 5-1. Clockwise from the top left, we have the original (uncompressed) image, a GIF version, a PNG version, and a JPEG version. You can see that PNG results in the smallest file size (515 bytes). It's about four times smaller than the GIF image. The JPEG is slightly larger than the PNG at 637 bytes, and it is also of lower quality due to the lossy compression (admittedly, the ability for the human eye to detect the difference in quality in this simple example is questionable at best).



**Figure 5-1.** Photoshop's Save For Web panel displaying file size differences for the same image in various formats

## The image that needs to work on any background

Sometimes it's necessary to create an image that works equally well on a variety of backgrounds. Some common examples are logos and icons. These situations have traditionally been the domain of the GIF file, but there are several reasons why PNG may be a better choice. PNG is almost always the winner in a file-size shootout on logos and other simple artwork. In addition, PNG's native transparency makes it simple to create a single file that works on top of any background you can throw under it. PNG does offer binary transparency—the same as GIF—but also provides the much more exciting alpha-channel variety, in which pixels can be *partially* transparent, rather than simply on or off. Using the latter does increase the file size—sometimes beyond that of a (binary) transparent GIF—but also allows for antialiasing the edges of your artwork, which makes for a much more elegant placement atop your background.

For the website of KTKA Channel 49 News in Topeka, Kansas ([www.49abcnews.com](http://www.49abcnews.com)), World Online staff crafted beautiful weather iconography to indicate the current conditions in the site's header. But, thanks to a clever bit of programming that causes the header to change from a daytime color scheme to a nighttime version precisely at sunset, the weather images needed to work equally well on different backgrounds. Take a look at Figures 5-2 and 5-3.

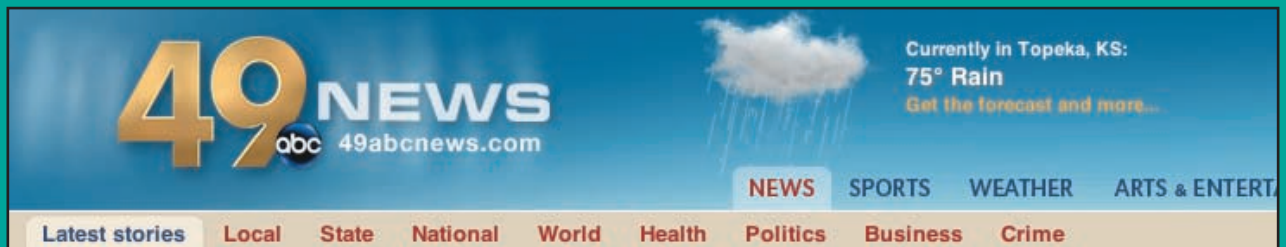


Figure 5-2. [www.49abcnews.com](http://www.49abcnews.com) header, daytime

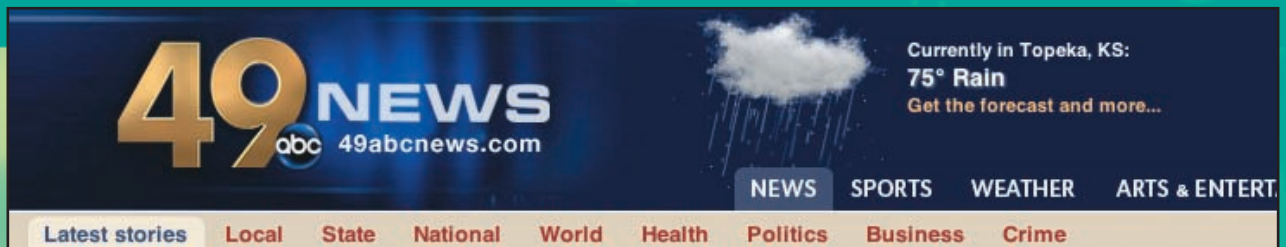


Figure 5-3. [www.49abcnews.com](http://www.49abcnews.com) header, nighttime

By using PNG, I was able to do the designer's work justice whether it appeared on the day or night background. And, should we choose to change the backgrounds at some point, I won't have to remake any weather icons, because the alpha-transparent PNG files will look great on anything.



If I had chosen to use GIF instead, I would have been limited to GIF's binary transparency. The result would have looked like Figure 5-4. I think we can all agree that's not good enough.

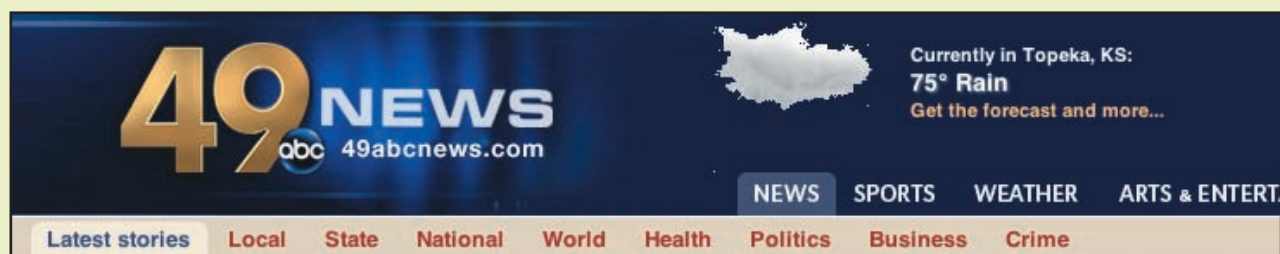


Figure 5-4. www.49abcnews.com header, nighttime, with GIF image instead of PNG

## The translucent HTML overlay

A very common graphic design technique is to overlay a photo or other image with a partially transparent region, usually containing text. This allows for readable text without completely obscuring the view of the image below. Designer Wilson Miner (www.wilsonminer.com) uses this to great effect on the Gingerroot jewelry site (www.simplygingerroot.com), as you can see Figure 5-5.

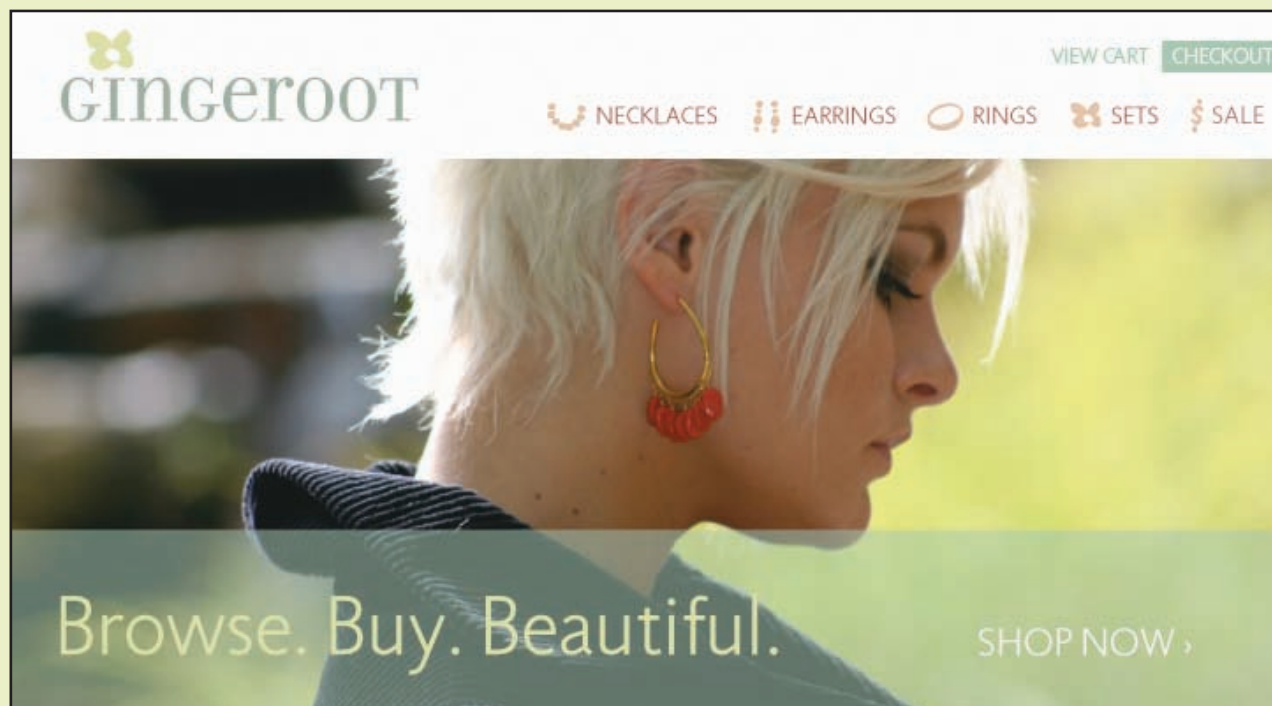


Figure 5-5. www.simplygingerroot.com, designed by the talented Wilson Miner (www.wilsonminer.com)

Wilson includes his transparent region and text in the JPEG image. He created them beforehand in Photoshop. This works fine, and it is totally appropriate for the site's needs. But what if the text in the translucent area needed to change very frequently—perhaps even be different for every visitor? In this case, it wouldn't be practical to put the text in the image. The text would need to be crafted in HTML and CSS. Using PNG's transparency alpha channel, we can emulate Wilson's style without putting the text in the image itself.

I'll start with a photograph of my daughter, Haley Madysan, and put it into a simple XHTML page with some basic CSS styling (this is `haley_example/index.html` in the code download). Note that I'm using an embedded CSS style sheet for demonstration purposes only. In real-world situations, using a linked external style sheet typically provides more flexibility, less code repetition, and more practical file management.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Haley's web site</title>
<style>
  body {
    font-family: "Lucida Grande", Helvetica, Arial, sans-serif;
    background-color: #304251;
    color: #304251;
    margin: 20px auto;
    width: 720px;
  }
  #feature {
    position: relative;
    width: 720px;
    height: 439px;
  }
  #feature-content {
    position: absolute;
    bottom: 0;
    left: 0;
    height: 125px;
    width: 720px;
    background-color: #dfdfdf;
  }
  #feature-content h1 {
    margin: 0;
    padding: 0;
    line-height: 125px;
    padding: 0 30px;
    font-weight: normal;
    font-size: 2.3em;
  }
  #feature-content a {
    float: right;
    font-size: .6em;
    color: #fff;
    text-decoration: none;
    text-transform: uppercase;
  }
</style>
</head>
<body>
  <div id="feature">
    <h1>Sweet. Smart. Beautiful.</h1>
    <a href="#">LEARN MORE</a>
  </div>
</body>
</html>
```

```

    }
  </style>
</head>
<body>
  <div id="feature">
    
    <div id="feature-content">
      <h1>Sweet. Smart. Beautiful. <a href="/haley" ↗
title="Haley Madysan Croft">Learn more &raquo;</a></h1>
    </div>
  </div>
</body>
</html>

```

With that, I've more or less duplicated what you saw in Wilson's Gingerroot site, except without any transparency (yet), as shown in Figure 5-6.

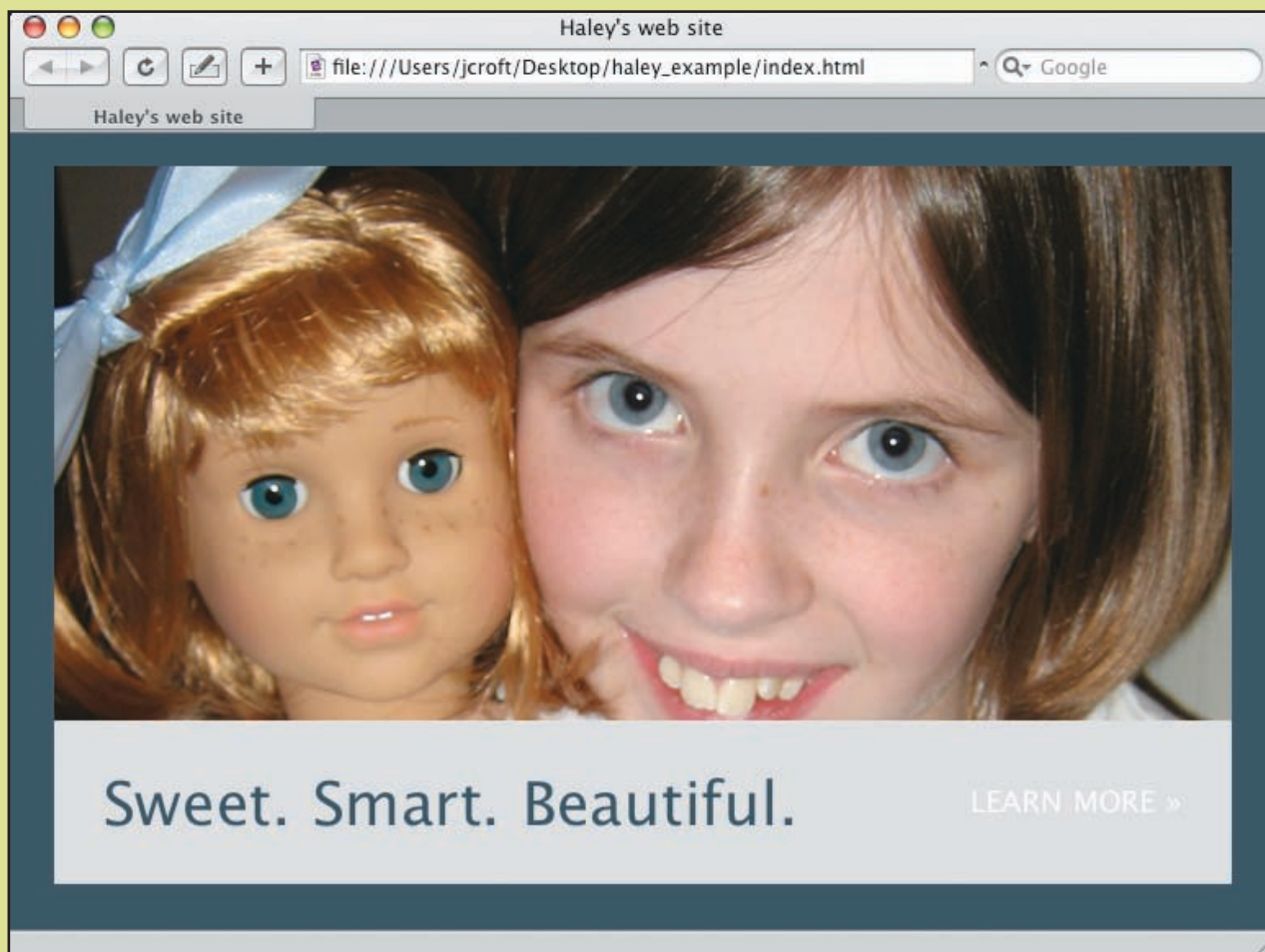


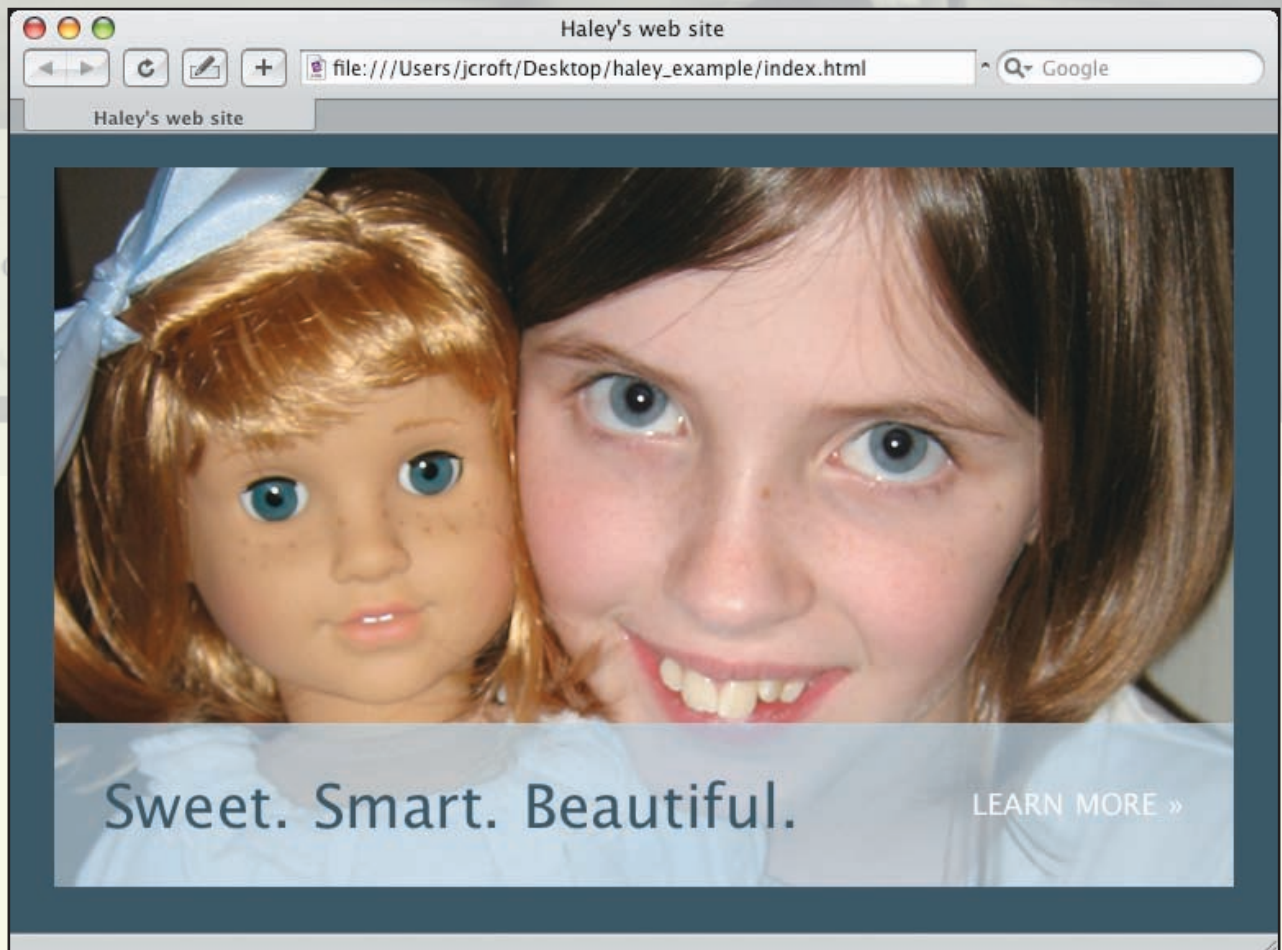
Figure 5-6. Emulating the [www.simplygingerroot.com](http://www.simplygingerroot.com) style with HTML and CSS, but no transparency (yet)



Now I'll create a 1-by-1-pixel image in Photoshop. I fill the image with a shade of light blue and set the layer to 70% opacity. Finally, I save the image using Photoshop's PNG-24 setting, enabling transparency. Then I simply use this image as the background for the overlay, instead of the solid gray you see in Figure 5-6.

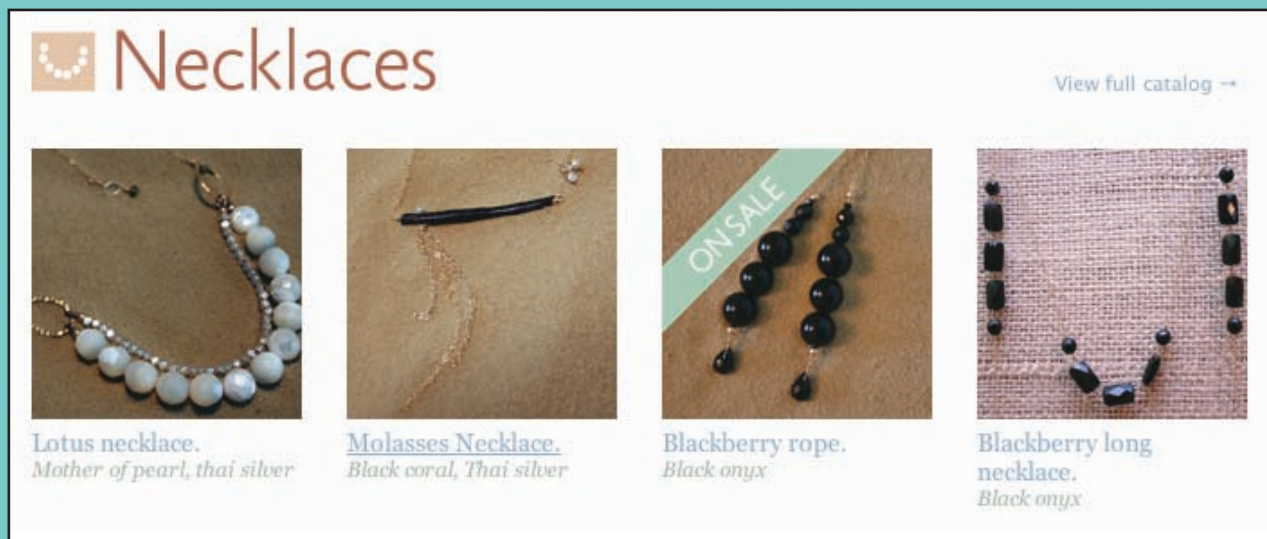
```
#feature-content {  
  position: absolute;  
  bottom: 0;  
  left: 0;  
  height: 125px;  
  width: 720px;  
  background-image: url('transparent.png');  
}
```

The result is quite similar to the original, but with the HTML and CSS text, it becomes more flexible, as shown in Figure 5-7.



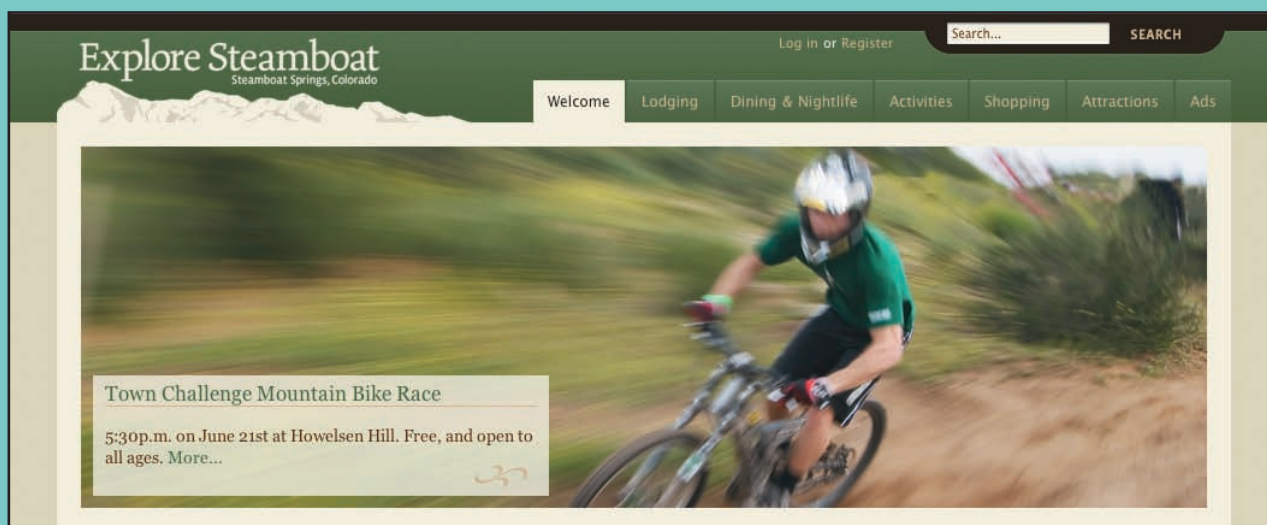
**Figure 5-7.** Adding transparency via the PNG image format nearly duplicates the [www.simplygingerroot.com](http://www.simplygingerroot.com) style.

Wilson Miner actually used the same concept in a different area of the [www.simplygingerroot.com](http://www.simplygingerroot.com) site. On pages that show available products, a transparent PNG image is used to display an On Sale flag in the upper-left corner of sale item product photos, as shown in Figure 5-8. By creating the On Sale image once, and saving it with a transparent background as a PNG image, Wilson avoided the need to create separate versions of every product image with the flag embedded.



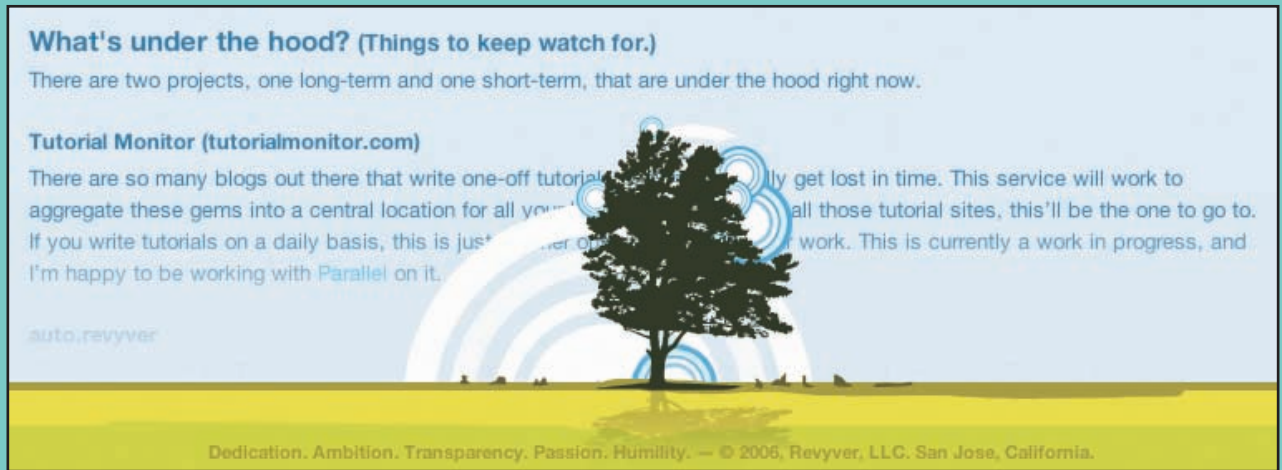
**Figure 5-8.** [www.simplygingerroot.com](http://www.simplygingerroot.com)'s Necklaces section uses a PNG image with a transparent background overlaid on top of the product image to display an On Sale flag in the upper-left corner.

I also used this technique on Explore Steamboat ([www.exploresteamboat.com](http://www.exploresteamboat.com)), a site dedicated to events, entertainment, and activities in Steamboat Springs, Colorado, as shown in Figure 5-9.

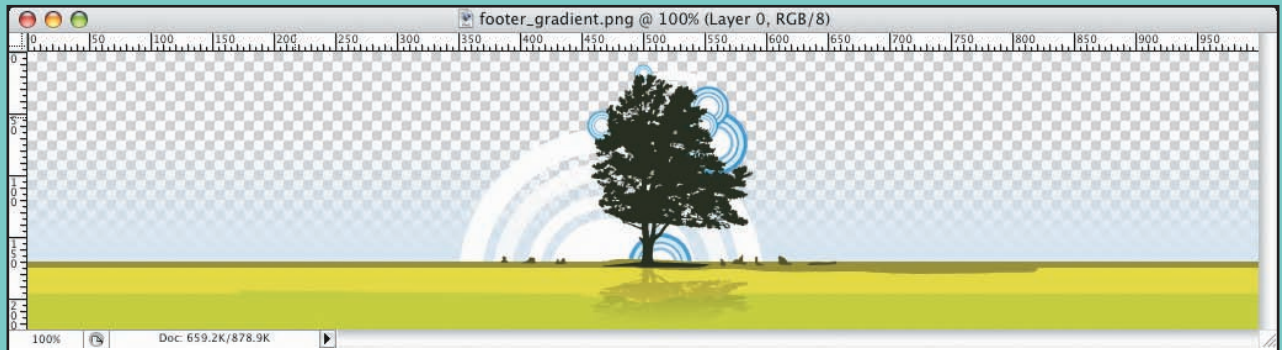


**Figure 5-9.** [www.exploresteamboat.com](http://www.exploresteamboat.com) has a translucent box sitting atop an image by way of transparent PNG.

In another creative example, designer Bryan Veloso ([www.avalonstar.com](http://www.avalonstar.com)) used a transparent PNG image anchored to the bottom of the page to create a “fade-in” effect, in which the text seems to appear out of thin air as you scroll down the page. The effect, found at [www.revyver.com](http://www.revyver.com) (see Figures 5-10 and 5-11) is better seen than described, so be sure to check it out for yourself. Additionally, the tree graphic sits in front of the text content of the page, producing an unexpected visual. It has quite a “wow” factor when you first see it.



**Figure 5-10.** At [www.revyver.com](http://www.revyver.com), designer Bryan Veloso has used a transparent PNG to create a “fade-in” effect as you scroll down the page, and to place his artwork in front of the text content of the page.



**Figure 5-11.** By viewing Bryan’s footer PNG image in Photoshop, we get an idea of how the transparent alpha channel was constructed to achieve the designed effect.

## The watermark

Another common graphic design technique is the subtle watermark overlaid on top of an image. This may be done purely for visual style, but it may also be done as a way of indicating the copyright holder or origin of the image.

On my personal website ([www.jeffcroft.com](http://www.jeffcroft.com), see Figure 5-12), I display a large gallery of photos I’ve taken (currently over 2000 photos). These photos are actually uploaded to Flickr ([www.flickr.com](http://www.flickr.com)), the popular online photo sharing site, and then displayed locally on my site by way of Flickr’s open API.



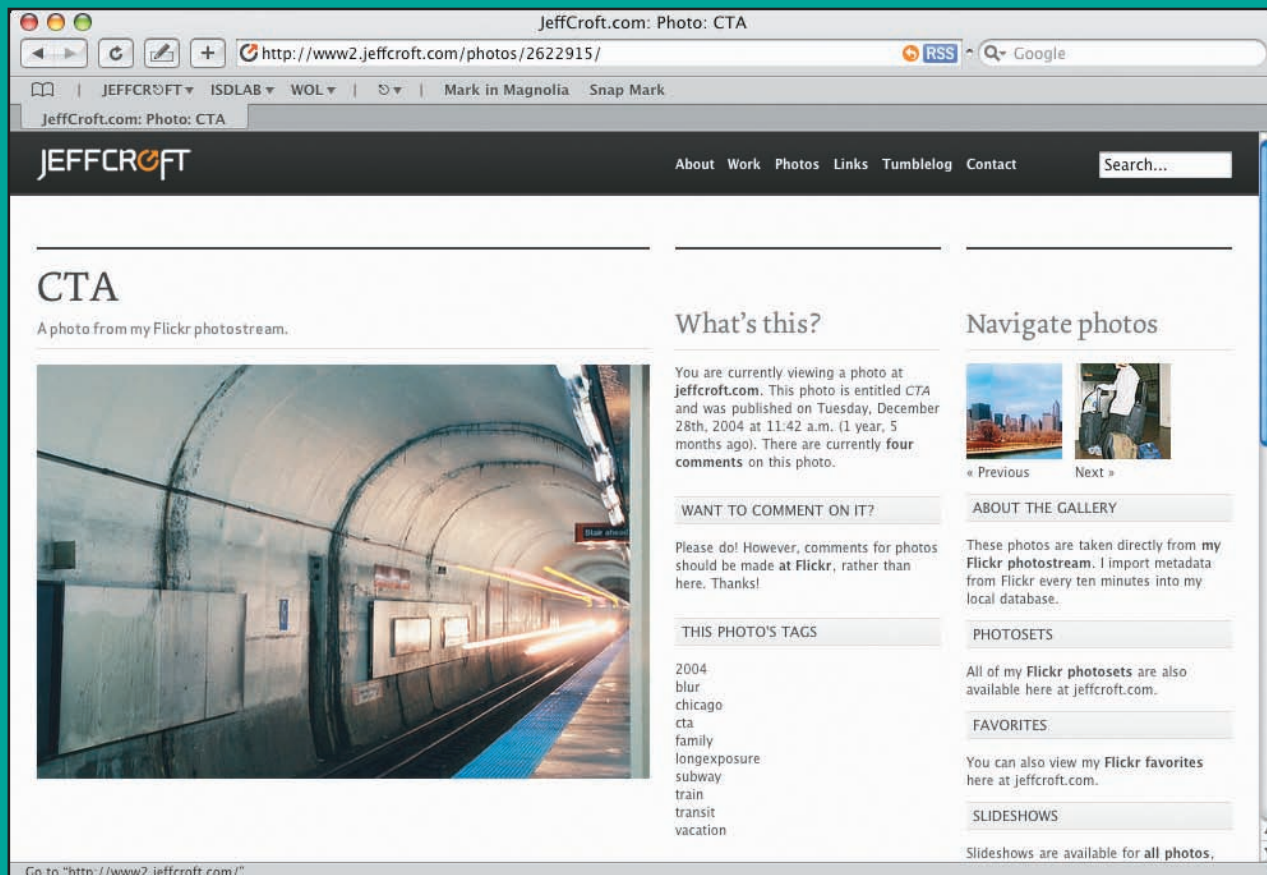


Figure 5-12. Photo detail page on [www.jeffcroft.com](http://www.jeffcroft.com)

What if I wanted to place my personal logo on all of my photographs? Yes, it's possible to open each image in Photoshop, apply the logo, and resave the image. However, this becomes terribly impractical when dealing with thousands of images that are frequently updated—and sometimes updated when I'm nowhere near a computer (for example, when I send photos to Flickr via my cell phone). Wouldn't it be nice if the logo were added automatically? PNG can help do just that.

The HTML used to display the photo in the page looks like this:

```
<a class="photo-container" href="http://www.flickr.com/photos/jcroft/2622915/">
  
</a>
```

I created an 80-by-80-pixel version of my logo in white and then set the opacity to 15% in Photoshop. Saved using Photoshop's standard PNG-24 optimization setting, the 15% translucency is preserved in the resulting PNG image. Then I simply added that image into my HTML as well:

```

<a class="photo-container" href="http://www.flickr.com/photos/jcroft/2622915/">
  
  
</a>

```

A bit of CSS is then used to position it in the right spot:

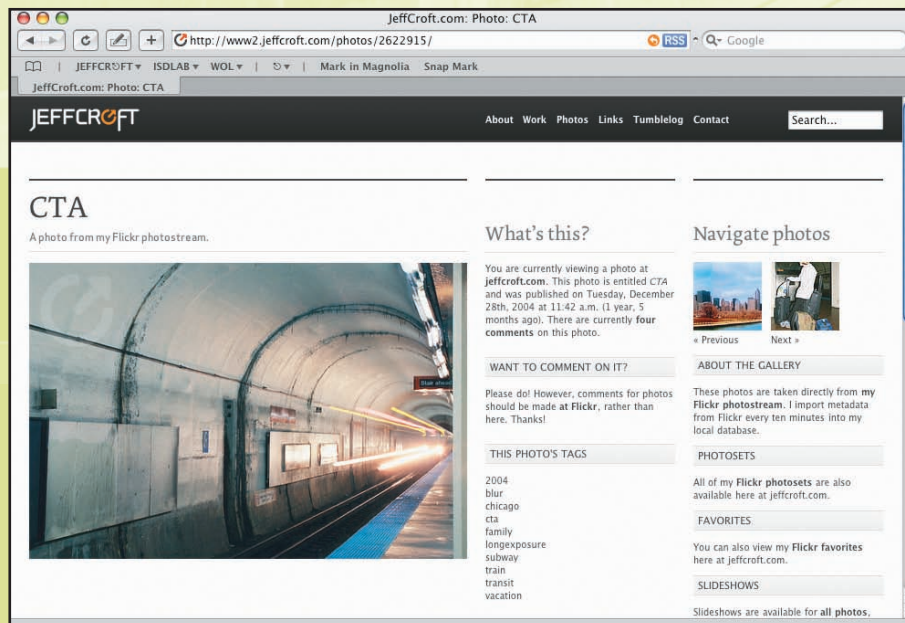
```

a.photo-container {
  position: relative;
  display: block;
}

img.watermark {
  position: absolute;
  top: 2em;
  left: 1em;
}

```

The result is a watermark that appears to be embedded in the photo itself, but is actually a separate PNG image sitting on top of it, as shown in Figure 5-13. By putting this into my templates for my content management system, I get the watermark on every image without having to do it all 2000 plus times.



**Figure 5-13.** Subtle [www.jeffcroft.com](http://www.jeffcroft.com) logo mark appears via transparent PNG in the upper-left corner of the photo.

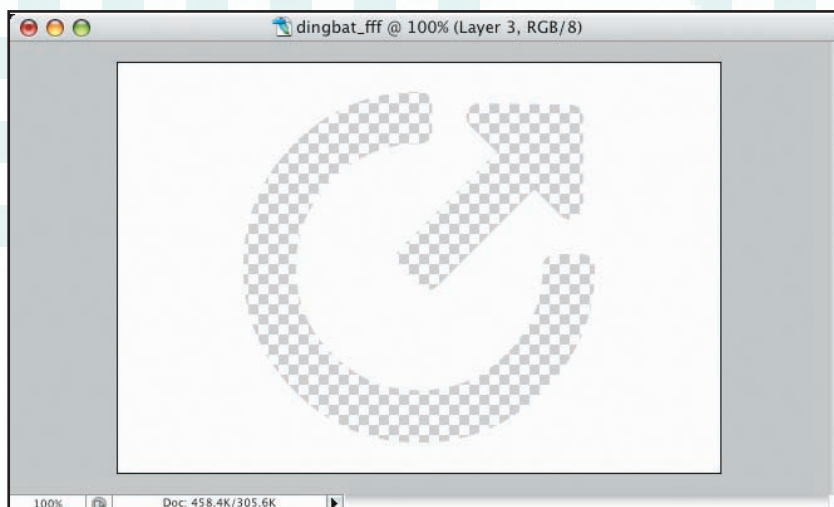
If you wanted to be extra-clever, you could even use DOM scripting to insert the additional (X)HTML markup for the watermark on the fly.



## The mask

Another handy use for the PNG image and its alpha-channel transparency is to mask an image below it. Technically, this is very similar to what you just saw with the watermark, but it achieves a different visual effect.

This time, I'll make a larger version of my logo in Photoshop. Instead of making the logo white, I'll make it transparent and the rest of the image white, because white is the background color of the page, as shown in Figure 5-14.



**Figure 5-14.** Creating an image in Photoshop for use as a transparent PNG mask

As I said, the technical aspects of this are nearly identical to the watermark in the preceding section, starting with the HTML:

```
<a class="photo-container" href="http://www.flickr.com/photos/jcroft/2622915/">
  
  
</a>
```

And then the CSS:

```
a.photo-container {  
  position: relative;  
  display: block;  
}  
  
img.mask {  
  position: absolute;  
  top: 0;  
  left: 0;  
}
```

See Figure 5-15 for the finished result.

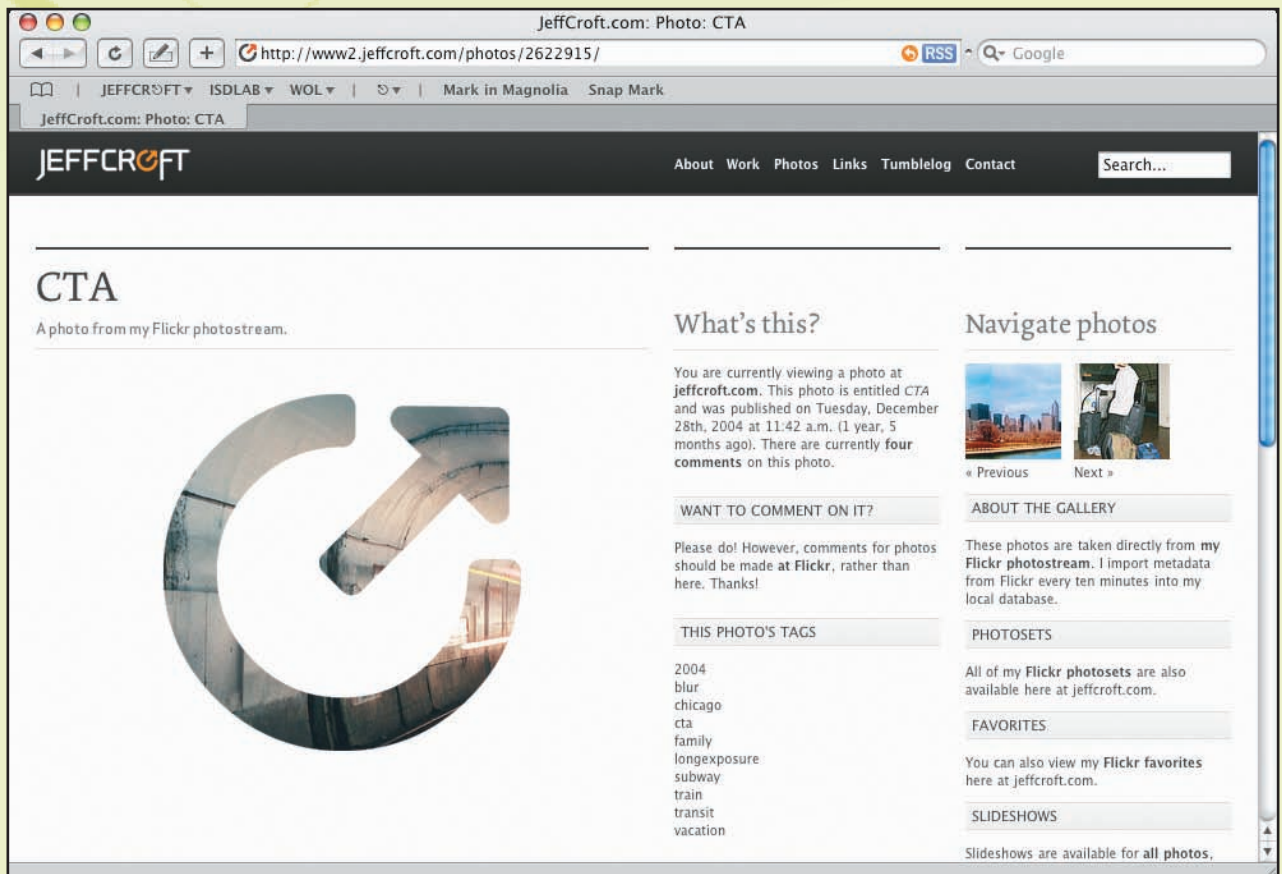


Figure 5-15. The PNG mask is layered on top of the photo to create a “punched-out” effect.

## The color-changing icon

By using the mask concept you saw in the previous example, some people have created icons that can change color, using only CSS. The idea is both simple and ingenious: place a transparent image with the icon symbol “punched-out” over a square, rectangle, or other shape with a flat-color CSS background, and you’ve got yourself an icon. By simply changing the CSS color of the background, you give the impression of the icon changing color.

Perhaps we need a set of icons indicating common sports, as shown in Figure 5-16.

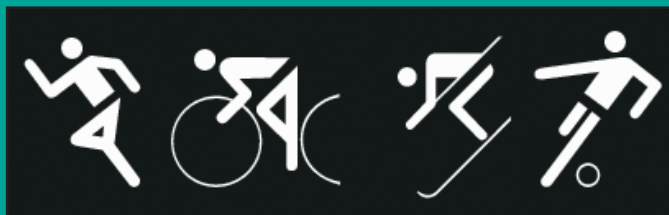


Figure 5-16. Some common sports iconography

I’ve created white images with a transparent punch-out in the shape of the icon symbol, just as I did with the logo in the mask example in the previous section, as shown in Figure 5-17.

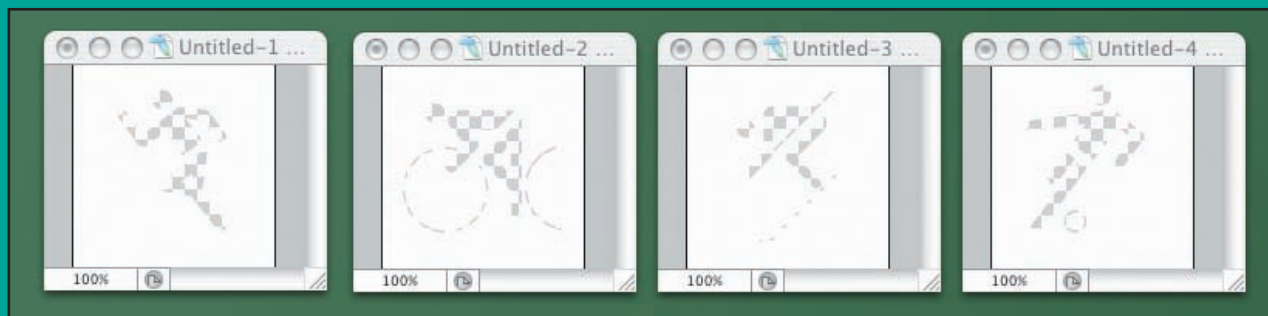


Figure 5-17. Creating “knocked-out” PNG masks for each icon

After scaling the images down to the appropriate size (I chose 48 by 48 pixels), I saved them using Photoshop’s PNG-24 default with transparency enabled. Then I created a simple XHTML file that references each image (sports\_icons\_example/index.html in the code sample download package):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>OMG Sports Icons!</title>
  <style>
    img {
      background-color: #cc0000;
    }
  </style>
</head>
```

```
<body>
  
  
  
  
</body>
</html>
```

I've specified #cc0000 (a strong red) as the background color for images in this file. The result is the red shining through the punched-out shapes, as shown in Figure 5-18. Note the beautifully smooth edges thanks to the antialiasing allowed by PNG's support for partially transparent pixels.



**Figure 5-18.** Transparent PNG images in place as web icons. Notice the CSS background color (red) showing through the image.

As you can see in Figure 5-19, I can change the color of the icon images simply by changing the color value in my CSS:

```
img {
  background-color: #000066;
}
```



**Figure 5-19.** Changing the background color in CSS changes the apparent color of the icon.

This kind of simple color changing can be very handy when you need to rework the design of your site. Instead of remaking all of your icon images, you simply change the color one time in CSS. It's also a great way to achieve a simple mouseover effect for linked images. For example, you could use the red for the standard color and the blue for the hover style (sports\_icons\_example/index\_links.html in the code sample download package):

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>OMG Sports Icons!</title>
<style>
  a:link img,
  a:visited img,
  a:active img {
    background-color: #cc0000;
  }
  a:hover img {
    background-color: #000066;
  }
</style>
</head>
<body>
<a href="/running/"></a>
<a href="/biking/"></a>
<a href="/skiing/"></a>
<a href="/soccer/"></a>
</body>
</html>

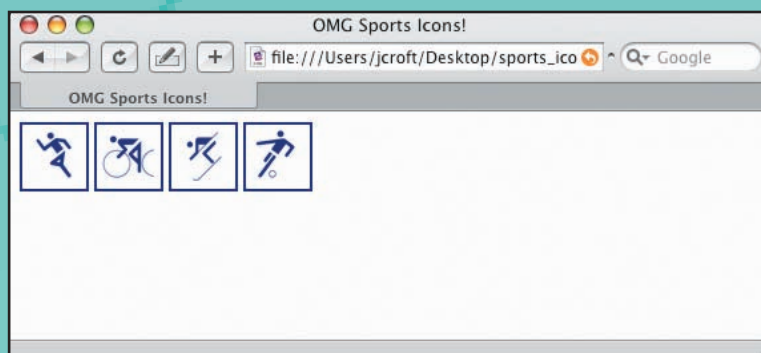
```

You can use additional CSS to do even more creative things with your masked icons, such as adding a border (see Figure 5-20):

```

img {
  background-color: #000066;
  padding: 2px;
}

```



**Figure 5-20.** Using CSS to further enhance the appearance of the icons by adding a border

Dan Cederholm ([www.simplebits.com](http://www.simplebits.com)) wrote about a very similar technique in his blog way back in 2003 ([www.simplebits.com/notebook/2003/07/24/magic\\_icons\\_for\\_lazy\\_people\\_like\\_me.html](http://www.simplebits.com/notebook/2003/07/24/magic_icons_for_lazy_people_like_me.html)), and PJ Onori ([www.somerandomdude.net](http://www.somerandomdude.net)) gives away a great set of icons (called Sanscons) based on it at his website ([www.somerandomdude.net/srd-projects/sanscons/](http://www.somerandomdude.net/srd-projects/sanscons/)).



However, I've reversed their concept by using a white image with a punched-out symbol. They used a white symbol on a transparent background, which can be very effective as well. In this case, your CSS background color shines through as the square or rectangle surrounding the symbol, rather than as the symbol itself.

Both Dan and PJ used transparent GIFs instead of PNGs. This is completely appropriate for their needs, as the style of the icons they created are bitmap, pixelly looking little guys. By using PNGs, you can apply the same technique, but take advantage of antialiased edges and partial transparency for use in more detailed icons.

## OK, but what browsers does it work in?

I know what you're thinking: all of this PNG transparency business seems really nice, but is it practical?

The good news is that almost all modern browsers fully support PNG images, including the alpha-channel transparency I've taken great advantage of in the examples. Safari (all versions), Firefox (all versions), Opera (version 6 and higher), Netscape (version 6 and higher), and Mozilla (all versions) will happily do everything I've asked it to. The bad news is that the one browser I haven't mentioned is the one your users are most likely to have: Internet Explorer.

Internet Explorer 6 and lower do not support the alpha-channel transparency built into the PNG format. Since this has been the browser of choice (or non-choice) for the vast majority of web surfers for several years now, this glaring hole has kept many web designers away from PNG. But, with the release of Internet Explorer 7, we finally have full support of PNG alpha transparency in all of the major browsers. What's more, there are ways to make alpha-transparent PNGs work in Internet Explorer 6 and lower. So, if you want to use these effects, nothing is stopping you. Internet Explorer 6 and its older siblings make it a little more work than it should be, but it's certainly possible.

## The Internet Explorer workaround: AlphaImageLoader

Internet Explorer includes several proprietary filters. They're used in CSS, but they're not part of any official CSS specification. In other words, they're *not* a web standard. It's unfortunate that Internet Explorer 6 and lower do not fully support the PNG image format (which is recommended by the W3C), but Microsoft does include a filter that overcomes this weakness: AlphaImageLoader.

According to Microsoft's official web page on the topic (<http://msdn.microsoft.com/workshop/author/filter/reference/filters/alphaimageloader.asp>), AlphaImageLoader "displays an image within the boundaries of the object and between the object background and content." In other words, AlphaImageLoader does load a PNG image with its full transparency intact, but it loads it as a layer of its own, underneath the content of the object to which it is applied. PNG images loaded this way act similarly to background images, rather than foreground images (although they actually sit on top of the object's background).

In short, you can simply apply the AlphaImageLoader CSS to an `img` element and have it produce the desired results. Doing so will load the image, transparency intact, but will also load the image again—as the foreground content of the object—with no transparency (thus obscuring your transparent version).

You can't use a transparent PNG as a CSS background image for an (X)HTML element (say, a `<div>`) and expect the AlphaImageLoader filter to make it work as expected in Internet Explorer. Remember that AlphaImageLoader inserts your image in between the background and foreground of the object. So, while it will load your image in all its transparent glory, it will still load it as the CSS background image as well, and without your beautiful translucent pixels.



## A real-world use of AlphamageLoader

Let's go back to one of the earlier examples and try to make Internet Explorer load the image properly. Remember Channel 49, the TV station in Topeka? Sure you do. Figure 5-21 shows how the site looks in Internet Explorer 6.

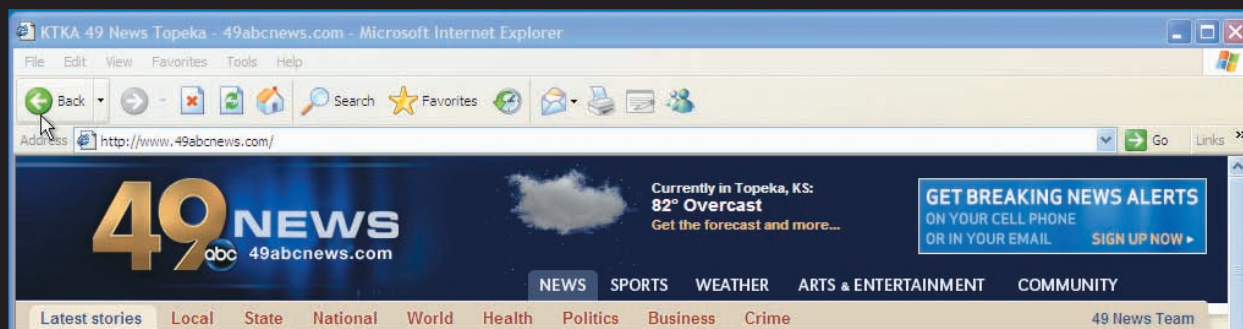


Figure 5-21. www.49abcnews.com header, displayed in Internet Explorer 6 for Windows, with PNG transparency intact

The HTML for the weather part of the header looks as you might expect:

```
<div id="weather">
  <a href="/weather/"></a>
  <p><a href="/weather/">Currently in Topeka, KS:<br />
<strong>82&deg; Overcast</strong><br /><span>Get the forecast
and more...</span></a></p>
</div>
```

You can see the image in question is a PNG, and yet Internet Explorer loads it flawlessly. The secret ingredient here is JavaScript. I've actually used a bit of DOM scripting to remove the `img` element on the fly and replace it with a `div` element that—you guessed it—has the AlphamageLoader CSS applied to it. The JavaScript is referenced within *conditional comments*, another handy but completely nonstandard idiom Microsoft has built into Internet Explorer. Conditional comments let you target code at particular versions of Internet Explorer. The code is then ignored by all other browsers, so it doesn't affect them. In the `<head>` element of `www.49abcnews.com`, you'll find:

```
<!--[if lte IE 6]>
  <script src="http://media.49abcnews.com/js/fixWeatherPng.js" type="text/javascript"></script>
<![endif]-->
```

Thanks to the first line, `if lte IE6`, this script element will be included in the rendered document only if it is being displayed by a version of Internet Explorer less than or equal to (that's what `lte` stands for) 6. All other browsers, including the forthcoming Internet Explorer 7, will ignore it completely.

So what's in the JavaScript file `fixWeatherPng.js`, anyway? Take a look:

```
window.attachEvent("onload", fixWeatherPng);

function fixWeatherPng() {
  var img = document.getElementById("weatherImage");
  var src = img.src;
  img.style.visibility = "hidden";
```

```
var div = document.createElement("DIV");
div.style.filter = "progid:DXImageTransform.Microsoft.↵
AlphaImageLoader(src='" + src + "', sizing='scale')";

// Some 49abcnews.com-specific CSS styling omitted for brevity.

img.replaceNode(div);
}
```

Let's analyze what this script is doing step by step. First, we tell the browser that we want to run the `fixWeatherPng` function when the page loads. The rest of the script is the function itself.

To start off with, we find the image we're dealing with by its `id` attribute and store it in a variable called `img`. We save the `src` attribute (the URL to the image file) in a variable called `src`. Then we hide the `img` element by setting its `visibility` CSS property to `hidden`.

Next, we create a new `div` element and store it in a variable we'll call `div`. We'll apply the `AlphaImageLoader` filter to it, using the URL from the `src` variable we saved before.

Finally, we replace the original `img` element (which is hidden) with the newly created `div` element, which has the `AlphaImageLoader` goodness attached.

Using DOM scripting to insert your `AlphaImageLoader`-filtered bits on the fly has the side benefit of keeping the unpretty-but-necessary invalid CSS out of your CSS files. Likewise, it keeps the nonsemantic `div` elements out of your (X)HTML markup. And since all of this is referenced only within conditional comments, there's no chance of other browsers choking on Microsoft's proprietary code.

If you must do something invalid, at least you can abstract it and keep it quarantined from everything that doesn't need it.

## Conclusion

PNG, as an image file format, offers many technical advantages over the much more widely used GIF. In fact, the advantages are so great that PNG should have taken over as the file format of choice for just about all nonphotographic images long ago. But Internet Explorer's lack of proper support for some of PNG's more exciting features, such as alpha-channel transparency, has resulted in many web developers shying away from it. But there are two very good reasons why you shouldn't be scared of PNG.

First, even Internet Explorer 6 and earlier versions' less-than-perfect support for PNG still offers everything you can get with GIF (save for animation, of course). And PNG almost always results in smaller files for faster transfer and less bandwidth usage.

Second, Internet Explorer 7 offers full support for PNG's alpha-channel transparency. The effects that can be achieved with the complete gamut of opacity options are virtually endless. I suspect designers who find interesting ways to use transparent PNGs, such as those outlined in this chapter, will open the doors to a whole new level of style not widely seen online to date. I've given you a handful of ideas on how you can get creative with PNGs and transparency, but don't stop there. Find your own, as well!