Web Standards Creativity: Innovations in Web Design with XHTML, CSS, and DOM Scripting

Cameron Adams
Mark Boulton
Andy Clarke
Simon Collison
Jeff Croft
Derek Featherstone
Ian Lloyd
Ethan Marcotte
Dan Rubin
Rob Weychert

friendsof ED™
DESIGNER TO DESIGNER™
an Apress® company

Web Standards Creativity: Innovations in Web Design with XHTML, CSS, and DOM Scripting

ISBN-13 (pbk): 978-1-59059-803-0

ISBN-10 (pbk): 1-59059-803-2

Printed and bound in China
9 8 7 6 5 4 3 2 1

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit www.apress.com.

The source code for this book is freely available to readers at www.friendsofed.com in the Downloads section.

Credits

Lead Editor
Chris Mills

Technical Reviewer
Molly Holzschlag

Editorial Board
Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Keir Thomas, Matt Wade

Project Manager
Kylie Johnston

Copy Edit Manager
Nicole Flores

Copy Editors
Marilyn Smith, Nicole Flores

Assistant Production Director
Kari Brooks-Copony

Compositor
Diana Van Winkle, Van Winkle Design Group

Proofreader
Lori Bring

Indexer
Brenda Miller

Interior Designer
Diana Van Winkle, Van Winkle Design Group

Cover Designer
Kurt Krames

Manufacturing Director
Tom Debolski

4

Designing for Outside the Box

andy
clarke

www.malarkey.co.uk
www.stuffandnonsense.co.uk

Based in the UK, **Andy Clarke** (Malarkey) has a background in advertising. He started his own design consultancy, Stuff and Nonsense (`www.malarkey.co.uk`), in 1998. Since then, he has designed sites for Disney Store UK, British Heart Foundation, Save the Children, and the World Wildlife Fund UK.

Andy is passionate about design and about Web Standards; he bridges the gap between design and code. Outside the studio, Andy is a member of the Web Standards Project, where he redesigned the organization's website in 2006 and is an Invited Expert to the W3C's CSS Working Group.

Andy is an internationally known trainer and conference speaker, and he regularly trains designers and developers in the creative applications of Web Standards. He writes about aspects of design and popular culture on his personal website, All That Malarkey (`www.stuffandnonsense.co.uk`), and is the author of *Transcending CSS: The Fine Art of Web Design*, published by New Riders in 2006 (`www.transcendingcss.com`).

Worries?

"Oh baby, what's the matter?" she said. "You look so stressed and worried."

"I am," I replied. "I'm worried that our car needs new tires, we have nothing in the fridge for tea except a paper bag of mushrooms and a tube of tomato puree, *and* my mother is having trouble with her feet again."

Worrying—it's no laughing matter, I can tell you. If worrying about your own problems is not bad enough, worrying about someone else's can be maddening. Still, help is on hand in the form of a new, fictitious service called WorrySome.net.

This will not be your common or garden-variety web application, with enough venture capital to mount a small war and a call center in New Delhi. This site is "by real people, for real people"—people who will worry for you, for a fee, of course. The service will leave you free to live your life, free from worry, as long as you keep on paying the subscription.

WorrySome.net needs a new website, and you are the just the person for the job. It's a tricky one, but don't worry; help is on hand to guide you through making this design into a reality. If you feel at any stage that you are starting to get even the tiniest bit concerned, you can always pay one of WorrySome's worriers to take the weight off your shoulders.

In this chapter, you are going to take the WorrySome.net homepage from design visual to a working prototype with XHTML and CSS. You will start with meaningful, content-out markup, which is always the first step in developing a creative, standards-based design. You will learn how to use powerful CSS selectors and layout techniques to bring this design to life.

Worrying about the Web

Personally, I'm very glad that WorrySome.net can worry on my behalf, because in the past two years, I have been worried about web design. The Web is a youthful, dynamic medium, where people should love to interact with sites that they visit. Creating sites that people love to use is one of the main goals of creative web design. But web designers and developers too often focus on the technical aspects of markup, CSS, Ajax, or on the "science" of usability and accessibility, rather than on connecting with visitors' emotions through good design.

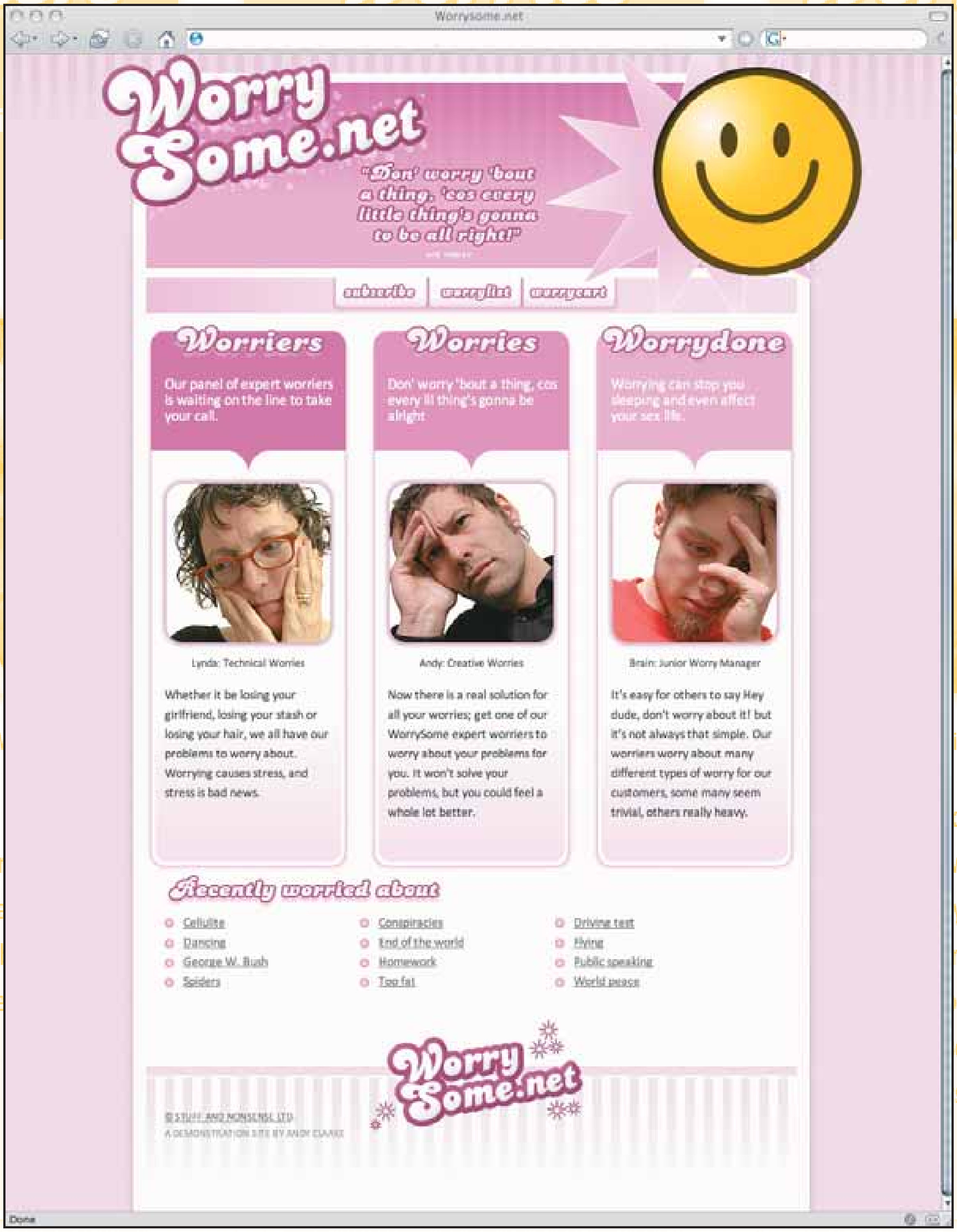
We've seen advances in the tools provided by CSS and wider support for these tools in mainstream web browsers. I hope that we can stop worrying about supporting browsers that are past their sell-by date and look forward by creating new and inspirational designs that break out of the boxes of current thinking.

Designing for WorrySome.net

As WorrySome.net is a novel new way of dealing with the worries of the world, the site calls for a design that breaks away from the familiar conventions of many of today's shiny web applications. The brief calls for this design to be open and friendly. It must make the visitor feel as "welcome as a trusted friend, rather than as a customer."

When creating the look for WorrySome.net, the design took on many forms. I made several experimental layouts with many different interface ideas, only a few of which made it into the final design. Figure 4-1 shows the one that you'll be working on in this chapter.

Let's get started making the design for the WorrySome.net homepage into a reality. You'll use meaningful XHTML markup and CSS, but not just any old CSS. This minimal markup will demand that you use techniques and CSS selectors that you may not yet have implemented in your own work. You can download all of the necessary files from `www.friendsofed.com`.



WorrySome.net

"Don' worry 'bout a thing, 'cos every little thing's gonna to be all right!"

Worriers

Our panel of expert worriers is waiting on the line to take your call.

Lynda: Technical Worries

Whether it be losing your girlfriend, losing your stash or losing your hair, we all have our problems to worry about. Worrying causes stress, and stress is bad news.

Worries

Don' worry 'bout a thing, cos every lil thing's gonna be alright

Andy: Creative Worries

Now there is a real solution for all your worries; get one of our WorrySome expert worriers to worry about your problems for you. It won't solve your problems, but you could feel a whole lot better.

Worrydone

Worrying can stop you sleeping and even affect your sex life.

Brain: Junior Worry Manager

It's easy for others to say Hey dude, don't worry about it! but it's not always that simple. Our worriers worry about many different types of worry for our customers, some many seem trivial, others really heavy.

Recently worried about

- Cellulite
- Dancing
- George W. Bush
- Spiders
- Conspiracies
- End of the world
- Homework
- Too fat
- Driving test
- Flying
- Public speaking
- World peace

WorrySome.net

© STUFF AND NONSENSE LTD
A DEMONSTRATION SITE BY ANDY CLARKE

Figure 4-1. WorrySome.net homepage

Stop worrying, start with markup

One of the aims of a designer working on the Web should be to convey meaning. I'm not just talking about the meaning expressed through a visual design to reinforce brand values, but also the meaning conveyed through the XHTML elements chosen for the content. Choosing elements appropriately for their meaning, rather than their visual presentation, will help you to create designs that are as flexible and accessible as they can possibly be. (Of course, you also need to make sure that your markup is as lean and flexible as possible.) You want to make sure that the elements will convey the full meaning of the content when it is viewed without the visual richness provided by your CSS.

Look back at Figure 4-1 and write down the meaning of each of the visual elements on the page. In this design, you see the following elements:

- A branding area that contains the name of the site and its tagline, a lyric from Bob Marley, a master of the laid-back approach to life
- A list of navigation links
- Three headings that are each followed by paragraphs of text and an inline image of a worrier
- A heading that is followed by a list of topics that people often worry about: from conspiracies to George W. Bush (I can't think of any connection there, no sir)
- Site information, commonly containing legal notes, copyright information, and design credits

With this content outline complete, you are ready to flesh out the markup that will be most appropriate to convey its meaning. At this point, you should be concerned only with describing the meaning of this content and not any division elements or presentational markup hacks.

You should begin at the top of your content outline's order and work downwards, so we'll start with the branding.

Adding the content elements

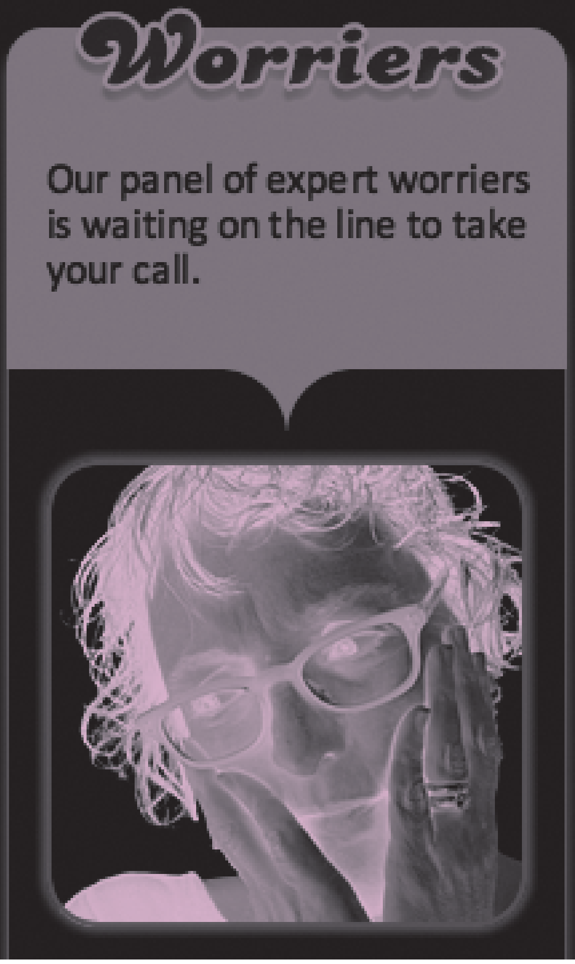
On the homepage of WorrySome, the name of the site can be a good choice for the top-level heading on the page.

```
<h1>Worrysome.net</h1>
```

Many designers will choose to vary this on internal pages, opting instead for a second-level heading and reserving the top-level heading for the page name. On internal pages, this name will likely also include a link back to the homepage, which would be redundant on the homepage.

Next comes the tagline. Written on the steps of Marley's house in Jamaica, where the smokes kept a rollin', this extract from "Three Little Birds" is perfect for a site about chilling out. Flip through the pages of the XHTML specification and look for a `lyric` element if you want. I'll just wait here, singing "sweet songs of melodies pure and true" until you return.

Back already?


Worriers
Our panel of expert worriers is waiting on the line to take your call.

In the absence of a more appropriate element to mark up this poetry from the great man of reggae, a blockquote element will do; after all, he sang those words.

```
<blockquote cite="http://www.bobmarley.com/songs/songs.cgi?threebirds">
<p>Don' worry 'bout a thing, cos every lil thing is gonna be alright.</p>
</blockquote>
```

Notice that the URL source of the quotation has been cited. Although this information will not be visible in a visitor's web browser, you should always cite the sources of any quotations that you reference. For an example of a way to display the URL of a quotation using scripting, see Jeremy Keith's book DOM Scripting: Web Design with JavaScript and the Document Object Model *(friends of ED, ISBN: 1-59059-533-5).*

If you are still in a "Mellow Mood," the list of links to the subscription page, worries list, and shopping cart that form the main navigation are ordered without any importance or weight attached to any particular link. An unordered list is the appropriate choice to mark up these links.

```
<h4>Main navigation</h4>
<ul>
  <li><a href="#" title="Subscribe">Subscribe</a></li>
  <li><a href="#" title="Worrylist">Worrylist</a></li>
  <li><a href="#" title="Worrycart">Worrycart</a></li>
</ul>
```

But what's the heading doing in there? Low-level headings atop lists of navigation links can be a helpful way to inform a visitor who is using a screen reader (or another form of assistive technology) of the purpose of a list. You may choose to make these headings visible or to hide them from view, perhaps by text-indenting them off the screen. This technique has become known as providing an *embedded alternate*.

Now we are really "Jamming," and it is time to move on to the main content of interest on this homepage: the descriptions of the services that this site provides. You will choose a second-level heading for each of the content areas, followed by their related content.

```
<h2>Worriers</h2>
<p>Introduction text</p>
<img src="images/worryone-i.png" alt="Lynda" />
<p>Name and role of worrier</p>
<p>Further descriptive text</p>

<h2>Worries</h2>
<p>Introduction text</p>
<img src="images/worrytwo-i.png" alt="Andy" />
<p>Name and role of worrier</p>
<p>Further descriptive text</p>

<h2>Worry done</h2>
<p>Introduction text</p>
<img src="images/worrythree-i.png" alt="Brain" />
<p>Name and role of worrier</p>
<p>Further descriptive text</p>
```

A third-level heading proudly announces the list of topics that the site's team of expert worriers can take off your shoulders. As this list has been written in alphabetical order, it is debatable whether it would be most appropriate to choose an ordered list, rather than an unordered list. For this example, I have chosen unordered, as no one item is more important than its siblings.

```
<h3>Recently worried about</h3>
<ul>
  <li><a href="#">Worry item</a></li>
  <li><a href="#">Worry item</a></li>
  <li><a href="#">Worry item</a></li>
</ul>
```

You are now almost at the bottom of the page, at the site information and a handy link back to the top to save the visitor's scrolling finger.

```
<p><a href="http://www.stuffandnonsense.co.uk">&copy; Stuff and➥
  Nonsense Ltd.</a> A demonstration site by Andy Clarke</p>
<ul>
  <li><a href="#worrysome-net" title="Top of this page">Top</a></li>
</ul>
```

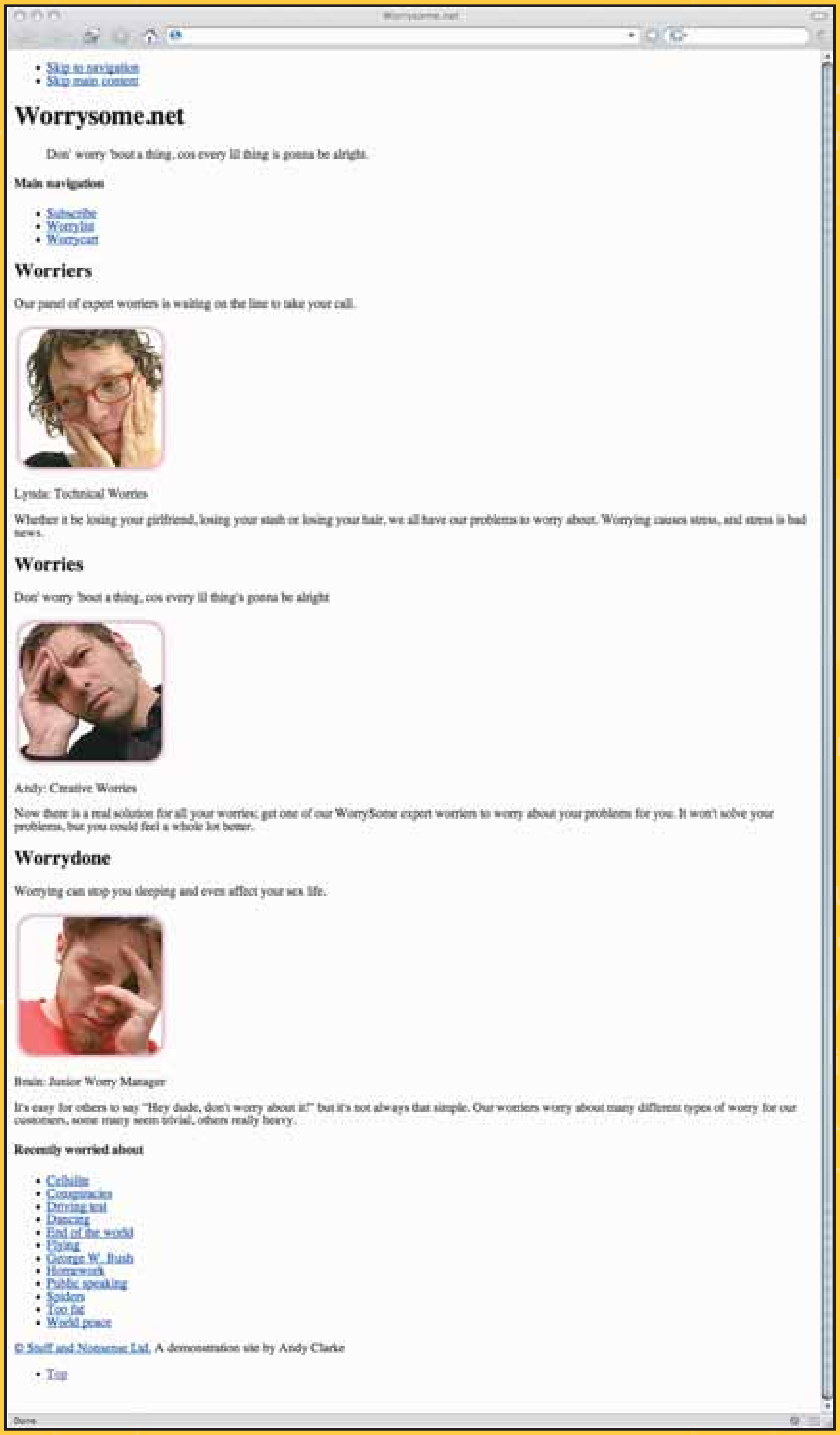
With your meaningful markup neatly written, now is your opportunity to preview your page in your development browser (see Figure 4-2) and to validate your code to ensure that no errors have crept in along the way.

If, like me, your choice of development browser is Firefox, you can find a host of developer extensions that will keep your markup valid as you work, not least of which is Chris Pederick's essential Web Developer Toolbar. You can download the Web Developer Toolbar from `http://chrispederick.com/work/webdeveloper/`.

of expert worriers
on the line to take

Don' worry 'bout a thing, cos every lil thing's gonna be alright

Worrying can stop you sleeping and even affect your sex life.



Worrysome.net

- Skip to navigation
- Skip main content

Worrysome.net

Don' worry 'bout a thing, cos every lil thing is gonna be alright.

Main navigation

- Subscribe
- Worrylist
- Worrycast

Worriers

Our panel of expert worriers is waiting on the line to take your call.

Lynda: Technical Worries

Whether it be losing your girlfriend, losing your stash or losing your hair, we all have our problems to worry about. Worrying causes stress, and stress is bad news.

Worries

Don' worry 'bout a thing, cos every lil thing's gonna be alright

Andy: Creative Worries

Now there is a real solution for all your worries; get one of our WorrySome expert worriers to worry about your problems for you. It won't solve your problems, but you could feel a whole lot better.

Worrydone

Worrying can stop you sleeping and even affect your sex life.

Brain: Junior Worry Manager

It's easy for others to say "Hey dude, don't worry about it!" but it's not always that simple. Our worriers worry about many different types of worry for our customers, some many seem trivial, others really heavy.

Recently worried about

- Cellulite
- Conspiracies
- Driving test
- Dancing
- End of the world
- Flying
- George W. Bush
- Homework
- Public speaking
- Spiders
- Too fat
- World peace

© Stuff and Nonsense Ltd. A demonstration site by Andy Clarke

- Top

Done

Figure 4-2. Previewing the page in a browser to ensure that the content is well ordered when read without syles is your first step in developing visually expressive but accessible designs.

Adding divisions from the content out

In general, web designers' understanding of markup and CSS has developed over recent years, but often our thinking about the ways to accomplish a visual design using CSS has changed little since we worked with tables for layout. Strip away the visual skin of many standards-based sites, their W3C "valid XHTML and CSS" badges glinting in the sunshine, and you will find a mass of nonsemantic and unnecessary `<div>` and `<span>` elements.

Working from the content out means starting with only the structural elements such as headings, paragraphs, and lists. This is an ideal method for keeping your markup free from presentational elements.

Next, you will group only those related elements that you have previously chosen into divisions, giving each an identity that describes the content that it contains.

> *Much has been written on the subject on semantic element naming. Former CSS Samurai John Allsopp has created WebPatterns* (`www.webpatterns.org`)*, a site dedicated to element-naming conventions. My original article on the subject of naming conventions can be found at All That Malarkey* (`www.stuffandnonsense.co.uk/archives/whats_in_a_name_pt2.html`)*.*

To avoid repetition of code, the next example shows only the content areas, rather than reproducing every nuance of markup.

```
<div id="branding">
  <h1>Worrysome.net</h1>
  <blockquote cite="http://www.bobmarley.com/songs/songs.cgi?threebirds ">
    <p>Don' worry 'bout a thing, cos every lil thing is gonna be alright.</p>
  </blockquote>
</div>

<div id="nav_main">
  <h4>Main navigation</h4>
  <ul>
    <li><a href="#" title="Subscribe">Subscribe</a></li>
    <li><a href="#" title="Worrylist">Worrylist</a></li>
    <li><a href="#" title="Worrycart">Worrycart</a></li>
  </ul>
</div>

<div id="content">
  <div id="content_main">
    <div id="worriers">
      <h2>Worriers</h2>
      <p>Introduction text</p>
      <img src="images/worryone-i.png" alt="Lynda" />
      <p>Name and role of worrier</p>
      <p>Further descriptive text</p>
    </div>
```

With the elements and appropriate divisions in your markup complete, but before you start working with CSS, you should take a moment to consider adding one further element. This will help those visitors browsing the site using a screen reader or a small-screen browser that has no support for style sheets.

While it is an impossible dream to expect that this site, or any other, can be fully accessible to *every* visitor, a few unobtrusive additions to your document will help some people enormously. For this example, you will add a short list of skip links, to allow a screen reader user to skip to either the main navigation or the main content.

```
<ul id="nav_access">
<li><a href="#nav_main">Skip to navigation</a></li>
<li><a href="#content_main">Skip main content</a></li>
</ul>
```

You should insert this list directly beneath the opening `<body>` tag and outside the container division.

Satisfying your soul (with CSS)

If markup was your "Punky Reggae Party," transforming this into the visual design layout should be your "Satisfy My Soul" time. (OK, that's the last of the Bob Marley song title malarkey, I promise.)

Implementing design layouts using CSS has come a long way since the early days of the Noodle Incident (`www.thenoodleincident.com`) and The Blue Robot (`www.bluerobot.com`). In those days of practical CSS use, positioning was the method of choice for achieving columns and other layout features. Before long, as designers became more ambitious in their attempts to make complex designs using CSS, positioning gave way to the use of floats. Unfortunately, positioning and its associated `z-index` stacking have since somewhat fallen from favor.

Although positioning is, on a first glance, more difficult to understand, it remains possibly the most powerful of CSS design tools. Implementing the WorrySome.net layout will make heavy use of both positioning and the `z-index`, in combination with floats.

You will also be working with image replacement and a whole host of CSS selectors. Many of these will be familiar to you if you have been working with CSS for some time; others might seem strange. You are about to use selectors that have until recently been the stuff of dreams for web designers. Don't worry "Buffalo Soldier" (sorry, I couldn't resist that), I will explain each new selector and technique as we progress.

Train to Styleville

Until recently, your choice of CSS selectors and techniques that could be used reliably across different browsers was limited to the few simple selectors supported by the world's most used browser, Microsoft's Internet Explorer 6 for Windows. That situation has now changed, thanks to the hard work and passion for standards of Internet Explorer 7's developers, and to the dedication of people like Molly E. Holzschlag of the Web Standards Project (`www.webstandards.org`), who built the bridges between the web developer community and Microsoft. We should all be grateful for their work.

Internet Explorer 7 is far from a perfect browser (what browser is?), but it does level the browser playing field. It has good support for so-called "advanced" CSS selectors, and it fixes the rendering issues and bugs of previous versions that have made the lives of web designers difficult over recent years.

I’m in the mood for style

Despite their broadly consistent rendering of CSS, most browsers will have a different default rendering of a page—the look of the page before author or user styles are applied—using what are known as *user agent* or *browser styles*, also conventionally known as *default* styles. Therefore, you should start by adding a few simple rules to your style sheet to guarantee that the styling of any element is as you, rather than the browser, intends. This example contains a broader range of elements than you will be using to implement WorrySome.net, to enable you to work with any elements that you may require in the future.

```
/*  =reset.css */

body, div, dl, dt, dd, ul, ol, li, h1, h2, h3, h4, h5, h6, pre, form,➥
  fieldset, input, p, blockquote, address, th, td {
margin : 0; padding :0; }

h2, h3, h4, h5, h6 {
font-size : 100%;
font-weight : normal; }

ol, ul {
list-style-type : none; }

table {
border-collapse : collapse;
border-spacing : 0; }

caption, th {
text-align : left; }
fieldset, img { border : 0; }

dt, address, caption, cite, code, dfn, em, i, strong, b, th, var {
font-style : normal;
font-weight : normal; }

q:before, q:after { content :''; }
```

In addition to this reset CSS, I also recommend that you specify `margin` and padding on commonly used text elements.

```
/* =blocktext */
h2, h3, h4, h5, p, ul {
margin : 0 20px;
padding : .5em 0; }
```

Styling WorrySome.net

Whereas many web designers take a granular approach to implementing their designs using CSS, you are going to take an outside-in approach, concentrating first on the outer elements of your design before working on the fine details. You'll begin by applying styles that set up the structure of your layout, starting with html, body, and the container division that encompasses all your content.

A background-color, background-image, and font color applied to the root element html will get the ball rolling. A slim background-image will repeat horizontally (repeat-x) to create the site's striped background.

```
html {
background : #f7d8e8 url(../images/html.png) repeat-x;
color : #333; }
```

Many of the image filenames that I have chosen relate precisely to the elements that they are styling. This approach helps to save time and reduce confusion when returning to a site after several months, or when a number of designers or developers are working together.

Next, you should apply styles to the body element. This design will be fixed-width and centered in the browser window.

```
body {
position : relative;
width : 740px;
margin : 20px auto 0 auto;
padding-top : 10px;
background : url(../images/body.png) repeat-y;
font : 88%/1.5 Calibri, Trebuchet, "Trebuchet MS", Helvetica, Arial, sans-serif; }
```

You may be wondering why position : relative; should be applied to the body element. After all, body is unlikely to be positioned or offset from its natural position. Many of the visual elements in this design are absolutely positioned on the page. Applying position : relative; to the body element will establish it as the positioning context for any of its positioned descendents.

Next, it's the outer container division's turn. Once again, this element has been established as a positioning context for its descendents and is centered within body to allow that element's background-image to create the subtle drop shadow for the depth of the page.

```
div[id="container"] {
width : 700px;
margin : 0 auto; }
```

```
        <div id="worries">
          <h2>Worries</h2>
          <p>Introduction text</p>
          <img src="images/worrytwo-i.png" alt="Andy" />
          <p>Name and role of worrier</p>
          <p>Further descriptive text</p>
        </div>

        <div id="worrydone">
          <h2>Worrydone</h2>
          <p>Introduction text</p>
           <img src="images/worrythree-i.png" alt="Brain" />
          <p>Name and role of worrier</p>
          <p>Further descriptive text</p>
        </div>
      </div>

      <div id="content_sub">
        <h3>Recently worried about</h3>
        <ul>
          <li><a href="#">Worry item</a></li>
          <li><a href="#">Worry item</a></li>
          <li><a href="#">Worry item</a></li>
        </ul>
      </div>
    </div>

    <div id="siteinfo">
      <p><a href="http://www.stuffandnonsense.co.uk">&copy; Stuff and Nonsense Ltd. </a>
      A demonstration site by Andy Clarke</p>
      <ul>
        <li><a href="#worrysome-net" title="Top of this page">Top</a></li>
      </ul>
    </div>
```

These appropriately identified divisions not only add the structure that will enable you to develop the visual layout with greater ease, but their identifiers also serve to enhance the meaning of the content that they contain. This minimal use of divisions should always be your approach in a content-based workflow. This places the content, rather than the visual layout, at the center of your thinking when writing your markup.

If additional divisions are required to accomplish any specific design, these should be added one at a time until your design can be achieved. For this design, only one additional container division, wrapped around all of your elements, is required. This container is a common approach for allowing you further styling options in addition to the `html` and `body` elements.

```
<div id="container">
All document content
</div>
```

You may have noticed that, in this markup example, no additional identifier or `class` attributes have been added, despite the complexity of the design. Such presentational attributes should rarely be required, as your markup already contains all the elements and attributes (`href`, `title`, `alt`, and so on) that you should ever need when you take a mature approach to your standards-based design.

If the selector in the example is unfamiliar to you, don't worry. This is an attribute selector, *a type that has largely been avoided by web designers and developers because of a lack of support for it in Internet Explorer 6. An attribute selector is made of three parts:*

- The element that you are selecting (`div` in this example)
- The attribute that you are using to select a specific element (`id` in this example)
- The value of the attribute (`container` in this example)

You could select the same element by using either `div#container` *or even* `#container`, *but that wouldn't be as much fun, would it?*

Styling basic page divisions

With the outer regions in place, you can turn your attention to defining the branding, navigation, and content areas of this design. For each, you will apply basic styling, including box and background properties. Start at the top with the branding area.

```
div[id="branding"] {
height : 200px;
margin-bottom : 10px;
background : #f0a4c7 url(../images/branding.png) repeat-x; }
```

This can be followed by the main navigation's outer division.

```
div[id="nav_main"] {
background : #fedaeb url(../images/nav_main.png) repeat-y; }
```

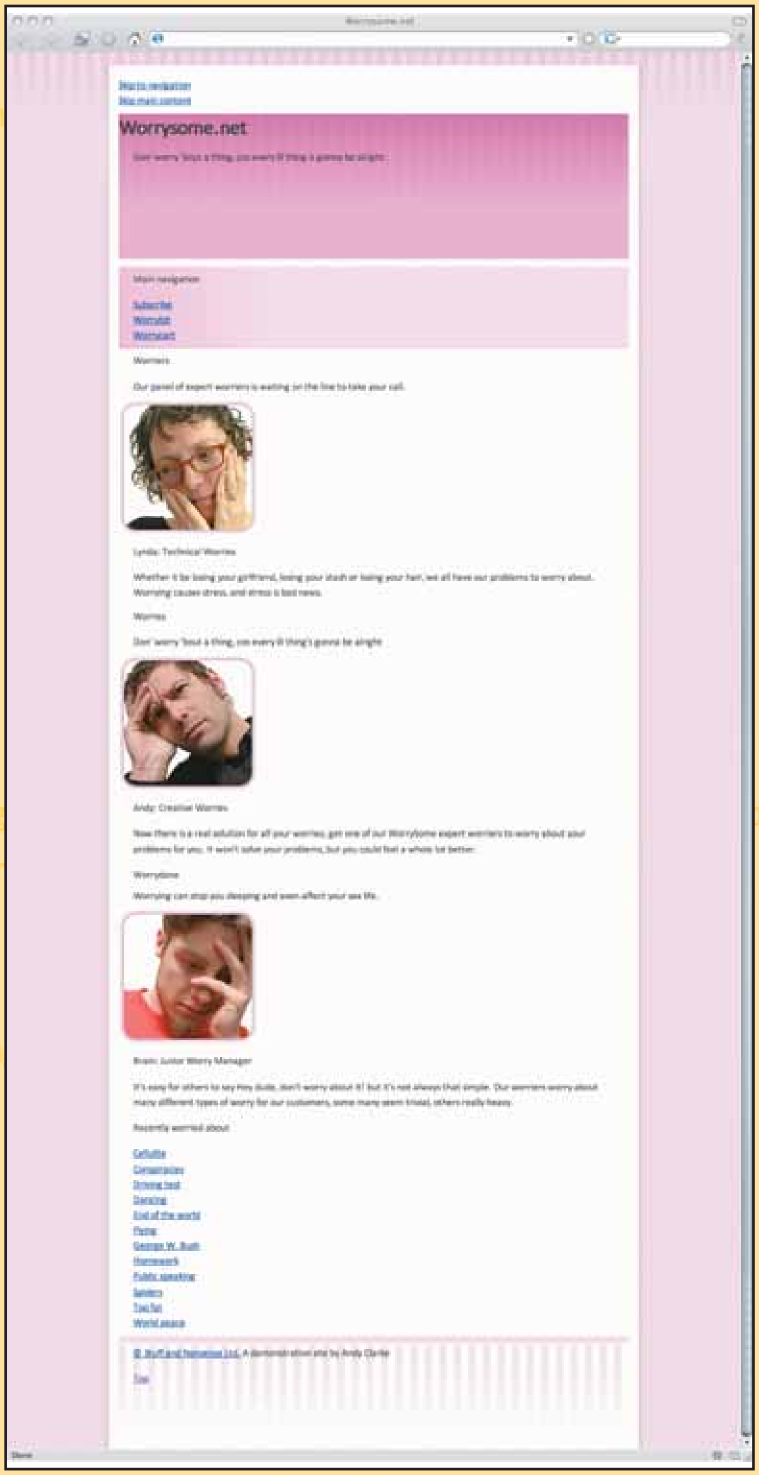
And finish by applying styling to the remaining main divisions:

```
div[id="content"] {
margin-bottom : 80px; }

div[id="content_main"] , div[id="content_sub"] {
width: 100%;  }

div[id="siteinfo"] {
clear : both;
min-height : 120px;
padding: .5em 0;
background : url(../images/siteinfo.png) repeat-x;  }
```

Take a peek at how your design is coming together by loading the page in your development browser (see Figure 4-3). Not too shabby a performance for such little effort, but you're still only a new entry at number 40, and you have a way to go to hit the top of the charts. Don't worry; it won't be as hard as you might think. (END OF CHAPTER EXCERPT).



Worrysome.net

Skip to navigation
Skip main content

Worrysome.net

Don' worry 'bout a thing, cos every lil thing is gonna be alright

Main navigation

Subscribe
Worrylist
Worrycart

Worriers

Our panel of expert worriers is waiting on the line to take your call.

Lynda: Technical Worries

Whether it be losing your girlfriend, losing your stash or losing your hair, we all have our problems to worry about. Worrying causes stress, and stress is bad news.

Worries

Don' worry 'bout a thing, cos every lil thing's gonna be alright

Andy: Creative Worries

Now there is a real solution for all your worries; get one of our WorrySome expert worriers to worry about your problems for you. It won't solve your problems, but you could feel a whole lot better.

Worrydone

Worrying can stop you sleeping and even affect your sex life.

Brain: Junior Worry Manager

It's easy for others to say Hey dude, don't worry about it! but it's not always that simple. Our worriers worry about many different types of worry for our customers, some many seem trivial, others really heavy.

Recently worried about

Cellulite
Conspiracies
Driving test
Dancing
End of the world
Flying
George W. Bush
Homework
Public speaking
Spiders
Top Tat
World peace

© Stuff and Nonsense Ltd. A demonstration site by Andy Clarke

Top

Figure 4-3. Styling the basic page `divs`, `html`, and `body` to create the canvas on which to create the WorrySome.net design

5

Creative Use of PNG Transparency in Web Design

jeff croft

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` and `www.ljworld.com`. Jeff also runs a popular blog and personal site at `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/`).

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/`) 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

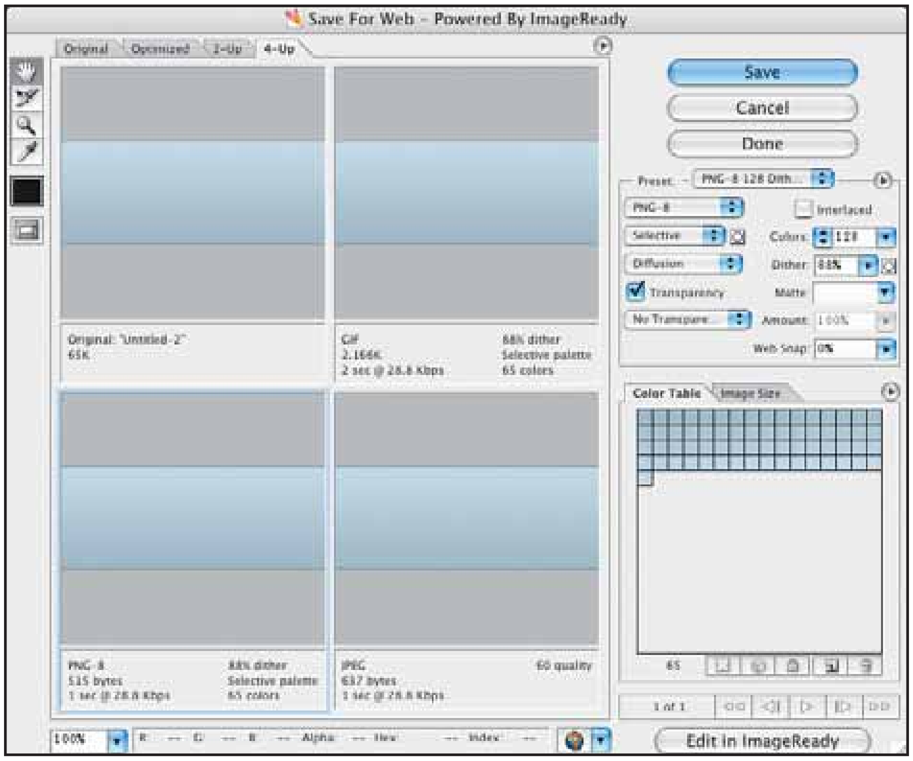
Now let's look at some great uses for the PNG in web design. I've included all the files for each example in a separate folder inside the code download for this chapter, available at `www.friendsofed.com`.

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).



Original: "Untitled-2"
65K

GIF
2.166K
2 sec @ 28.8 Kbps
88% dither
Selective palette
65 colors

PNG-8
515 bytes
1 sec @ 28.8 Kbps
88% dither
Selective palette
65 colors

JPEG
637 bytes
1 sec @ 28.8 Kbps
60 quality

Preset: PNG-8 128 Dith...
PNG-8
Interlaced
Selective
Colors: 128
Diffusion
Dither: 88%
✓ Transparency
Matte:
No Transparency
Amount: 100%
Web Snap: 0%

65
1 of 1
100%
R: -- G: -- B: -- Alpha: -- Hex: -- Index: --

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`), 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.

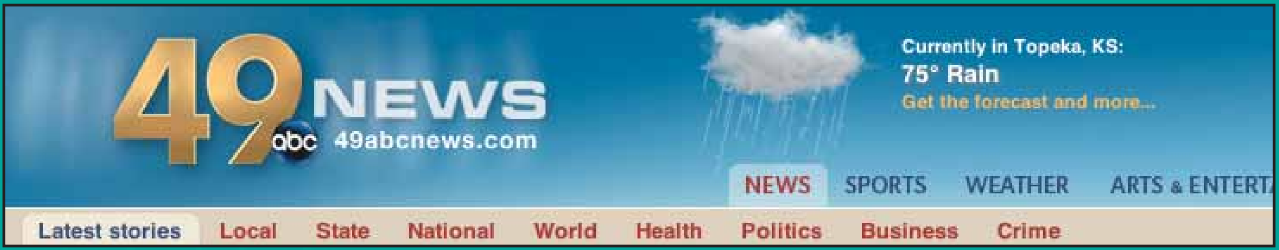

49 NEWS
abc 49abcnews.com
Currently in Topeka, KS:
75° Rain
Get the forecast and more...
NEWS SPORTS WEATHER ARTS & ENTERT
Latest stories Local State National World Health Politics Business Crime

Figure 5-2. `www.49abcnews.com` header, daytime

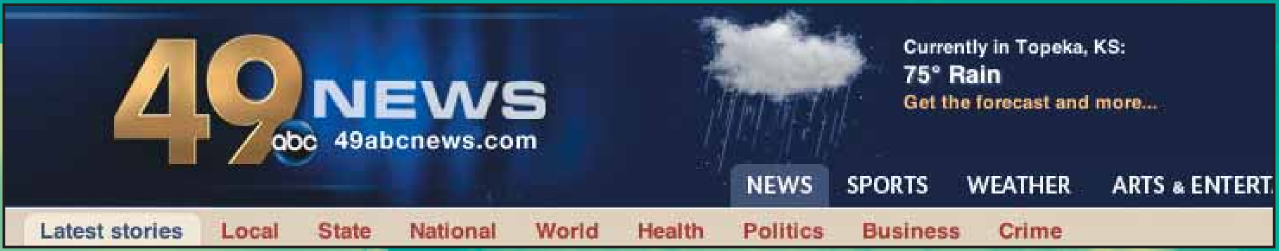
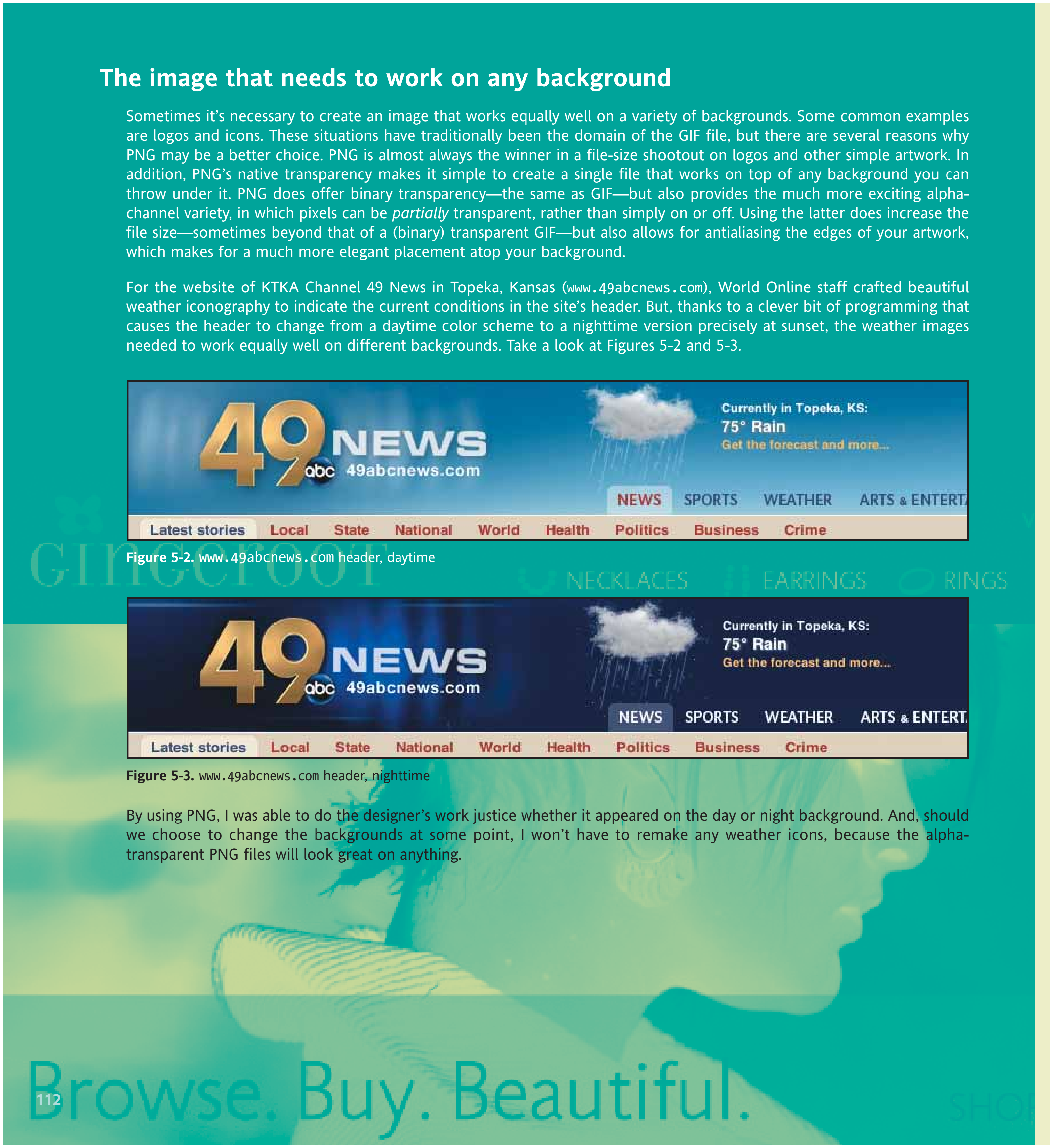

49 NEWS
abc 49abcnews.com
Currently in Topeka, KS:
75° Rain
Get the forecast and more...
NEWS SPORTS WEATHER ARTS & ENTERT
Latest stories Local State National World Health Politics Business Crime

Figure 5-3. `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.

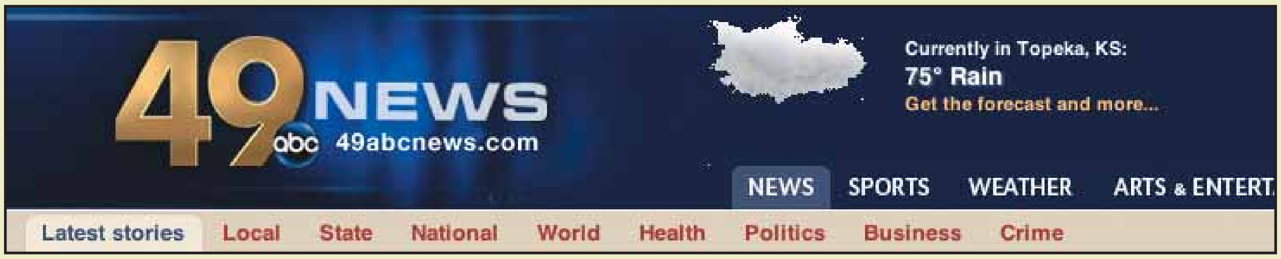


Figure 5-4. `www.49abcnews.com` header, nightime, with GIF image instead of PNG

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 Gingeroot jewelry site (`www.simplygingeroot.com`), as you can see Figure 5-5.

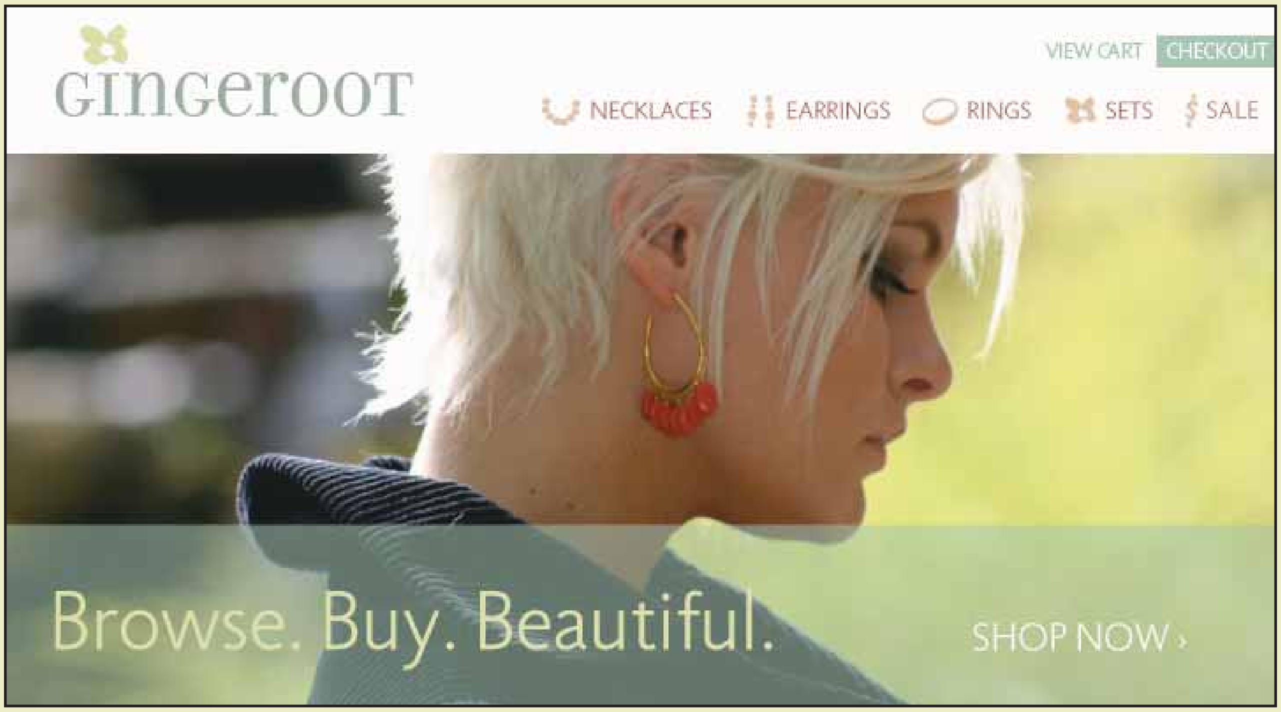


Figure 5-5. `www.simplygingeroot.com`, designed by the talented Wilson Miner (`www.wilsonminer.com`)

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">
      <img src="haley.jpg" alt="Haley Madysan Croft" />
      <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 Gingeroot site, except without any transparency (yet), as shown in Figure 5-6.

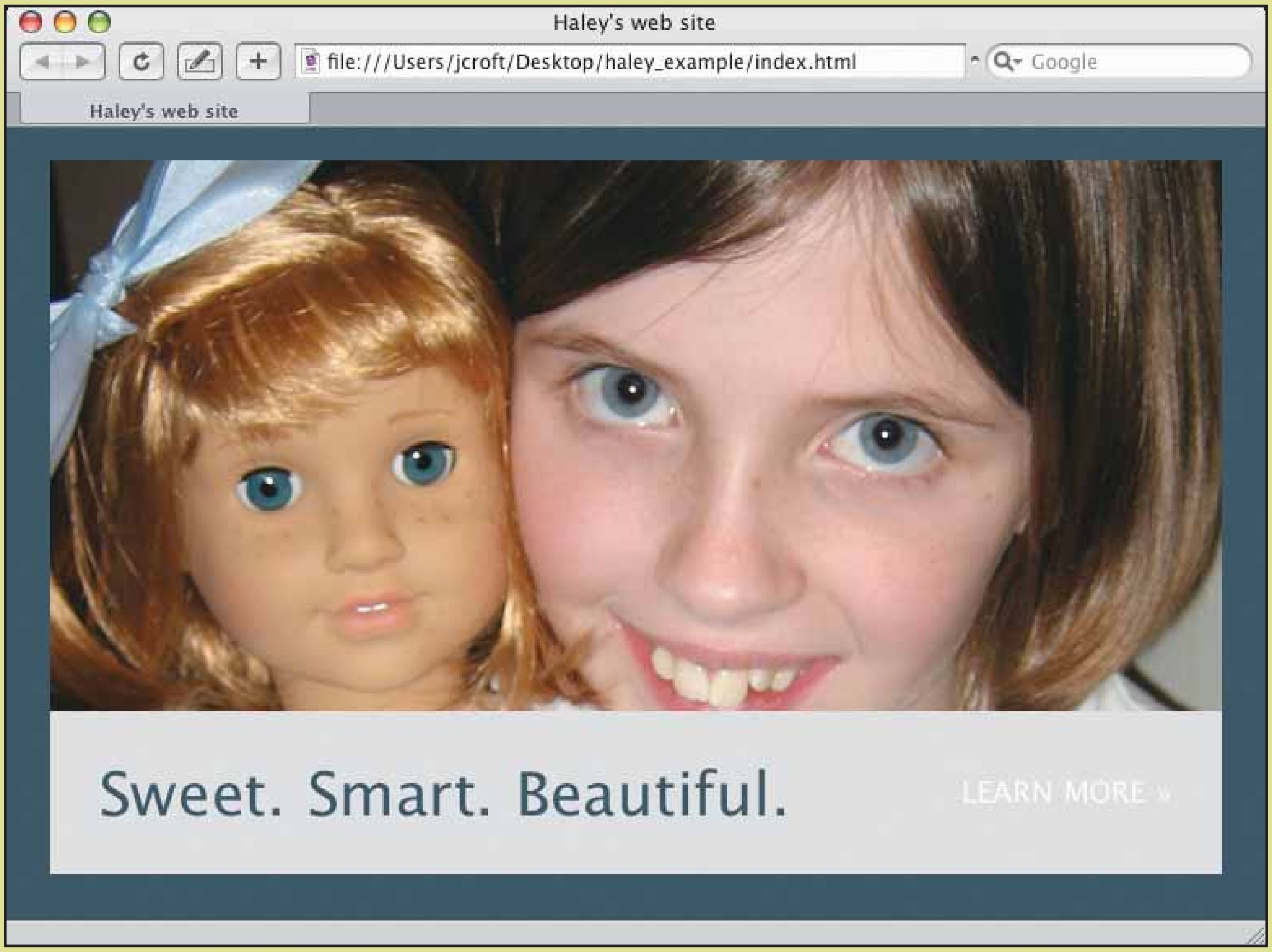


Figure 5-6. Emulating the `www.simplygingeroot.com` style with HTML and CSS, but no transparency (yet)

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.

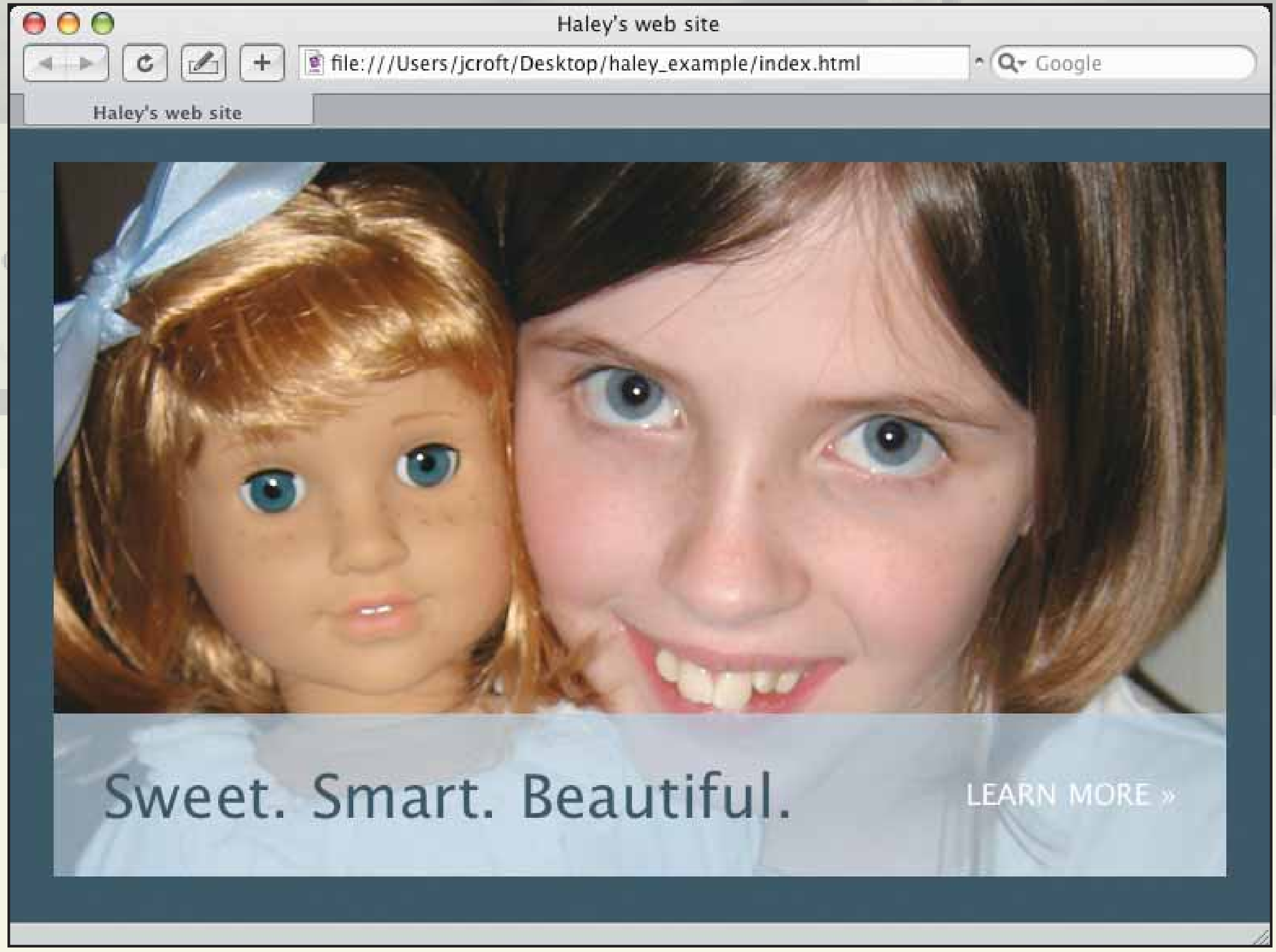


Figure 5-7. Adding transparency via the PNG image format nearly duplicates the `www.simplygingeroot.com` style.

Wilson Miner actually used the same concept in a different area of the www.simplygingeroot.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.

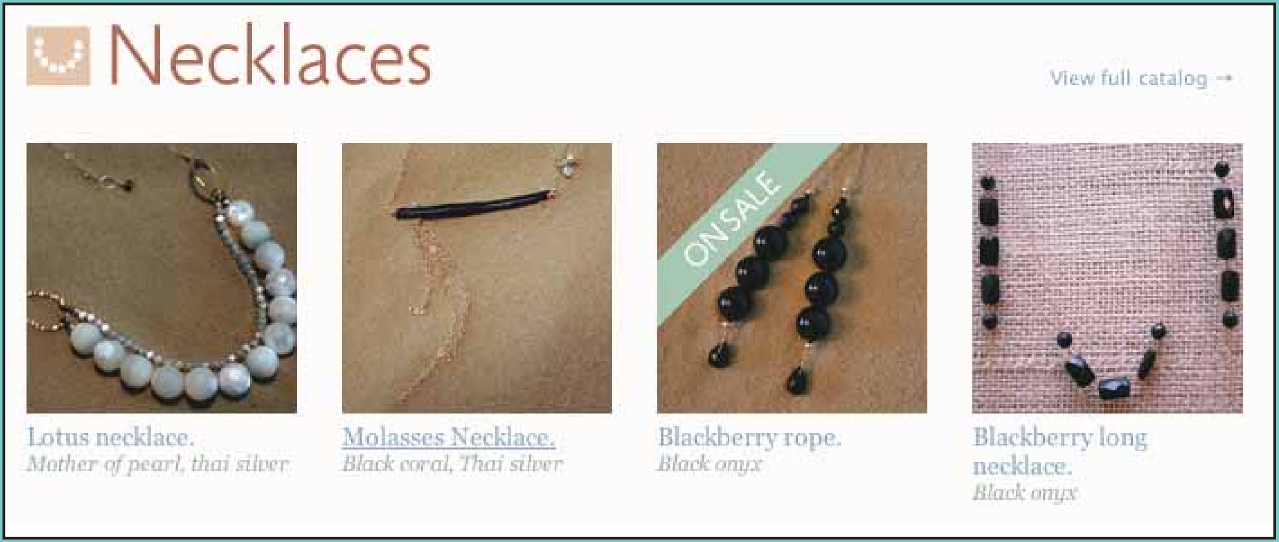


Figure 5-8. www.simplygingeroot.com's Necklaces section uses a PNG image with a transparent background overlaid on top of the product image to display an On Sale flag in the upper-left corner.

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

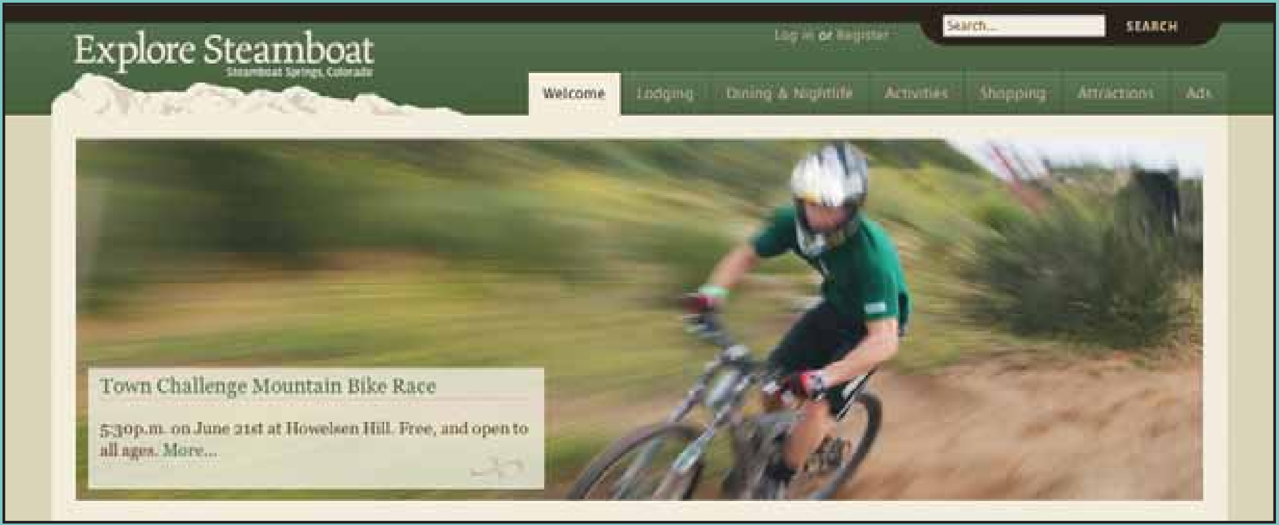
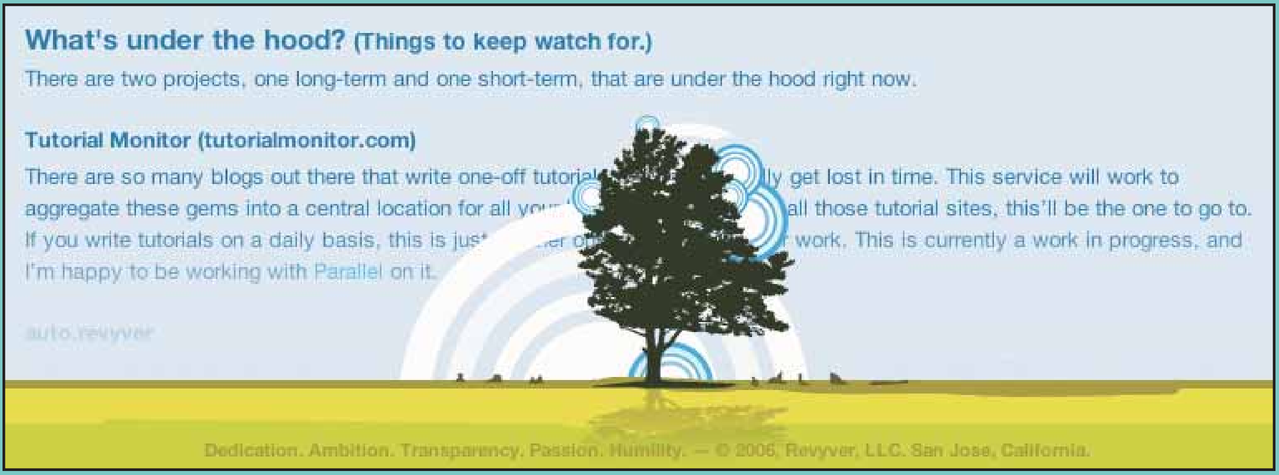

Explore Steamboat
Steamboat Springs, Colorado
Log in or Register
Search...
SEARCH
Welcome
Lodging
Dining & Nightlife
Activities
Shopping
Attractions
Ads
Town Challenge Mountain Bike Race
5:30p.m. on June 21st at Howelsen Hill. Free, and open to all ages. More...

Figure 5-9. 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) 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 (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.



What's under the hood? (Things to keep watch for.)

There are two projects, one long-term and one short-term, that are under the hood right now.

Tutorial Monitor (tutorialmonitor.com)

There are so many blogs out there that write one-off tutoria … lly get lost in time. This service will work to aggregate these gems into a central location for all yo … all those tutorial sites, this'll be the one to go to. If you write tutorials on a daily basis, this is jus … er o … work. This is currently a work in progress, and I'm happy to be working with Parallel on it.

auto.revyver

Dedication. Ambition. Transparency. Passion. Humility. — © 2006, Revyver, LLC. San Jose, California.

Figure 5-10. At 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.

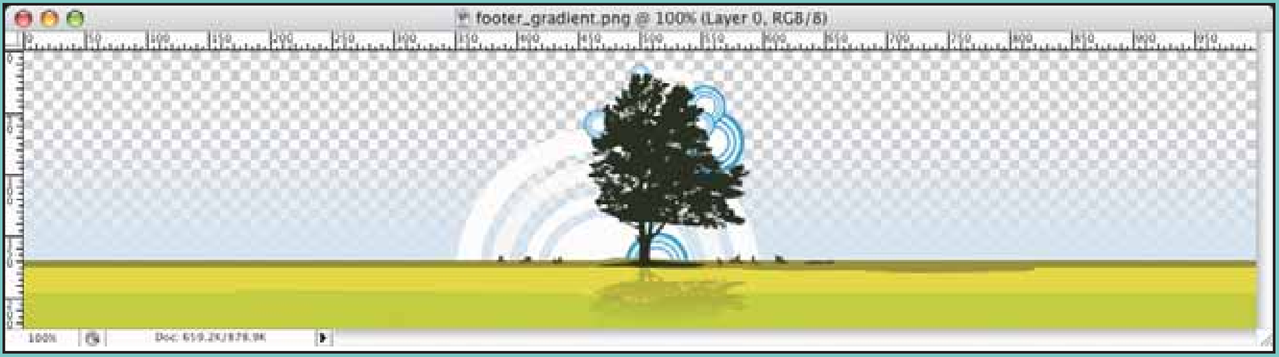
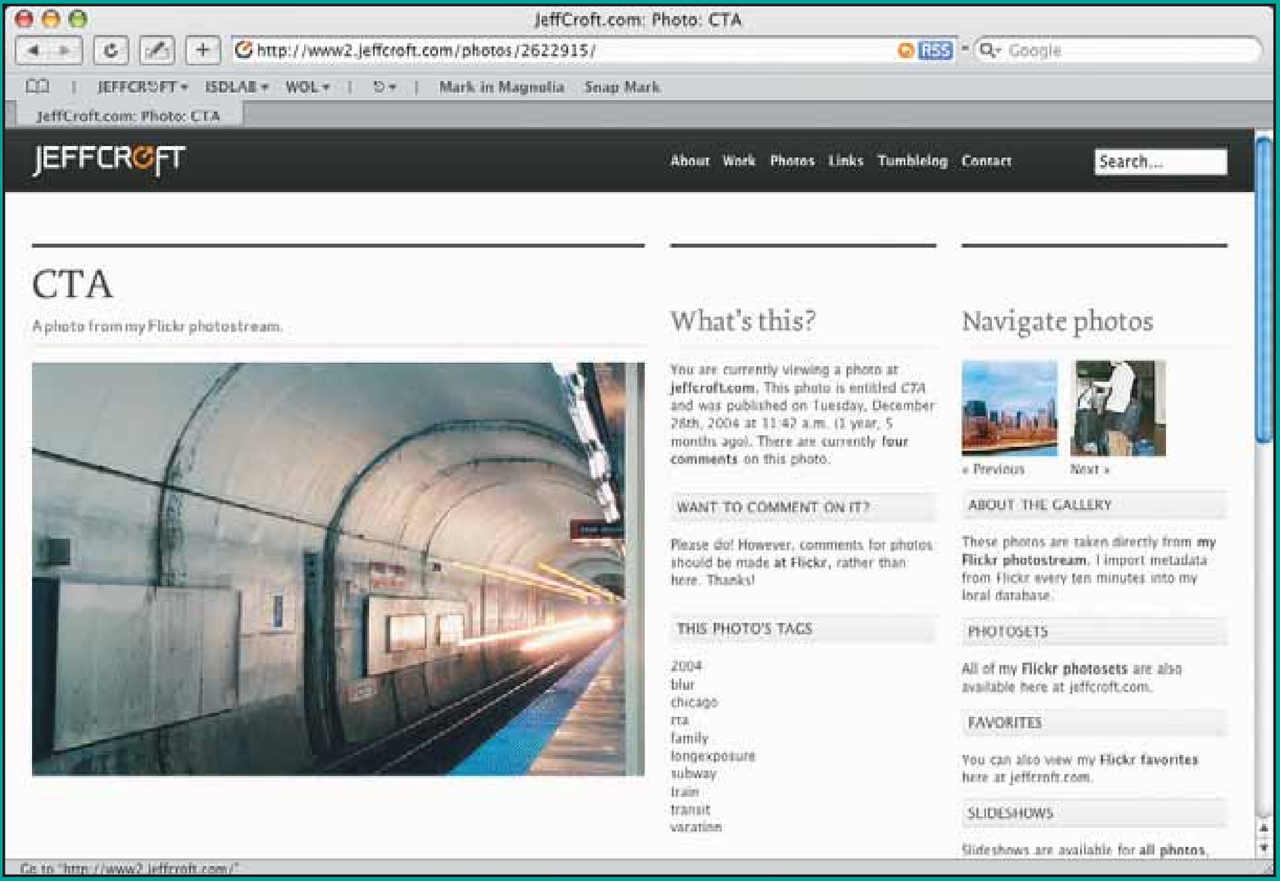


Figure 5-11. By viewing Bryan’s footer PNG image in Photoshop, we get an idea of how the transparent alpha channel was constructed to achieve the designed effect.

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, 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), the popular online photo sharing site, and then displayed locally on my site by way of Flickr’s open API.


JeffCroft.com: Photo: CTA
http://www2.jeffcroft.com/photos/2622915/
Google
JEFFCROFT ISDLAB WOL Mark in Magnolia Snap Mark
JEFFCROFT
About Work Photos Links Tumblelog Contact
Search...
« Previous
Next »

CTA

A photo from my Flickr photostream.

What's this?

You are currently viewing a photo at **jeffcroft.com.** This photo is entitled *CTA* and was published on Tuesday, December 28th, 2004 at 11:42 a.m. (1 year, 5 months ago). There are currently **four comments** on this photo.

WANT TO COMMENT ON IT?

Please do! However, comments for photos should be made **at Flickr**, rather than here. Thanks!

THIS PHOTO'S TAGS

2004
blur
chicago
cta
family
longexposure
subway
train
transit
vacation

Navigate photos

ABOUT THE GALLERY

These photos are taken directly from **my Flickr photostream.** I import metadata from Flickr every ten minutes into my local database.

PHOTOSETS

All of my **Flickr photosets** are also available here at jeffcroft.com.

FAVORITES

You can also view my **Flickr favorites** here at jeffcroft.com.

SLIDESHOWS

Slideshows are available for **all photos**,

Go to "http://www2.jeffcroft.com/"

Figure 5-12. Photo detail page on `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/">
  <img class="full-size-photo" src=http://static.flickr.com/2/2622915_8b78c1207d.jpg➥
alt="CTA, a photo by Jeff Croft" />
</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/">
  <img class="full-size-photo" src="http://static.flickr.com/2/2622915_8b78c1207d.jpg"➥
alt="CTA, a photo by Jeff Croft " />
  <img class="watermark" src="http://media.jeffcroft.com/img/core/jeffcroft_logo_watermark.png"➥
alt="Watermark" />
</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.

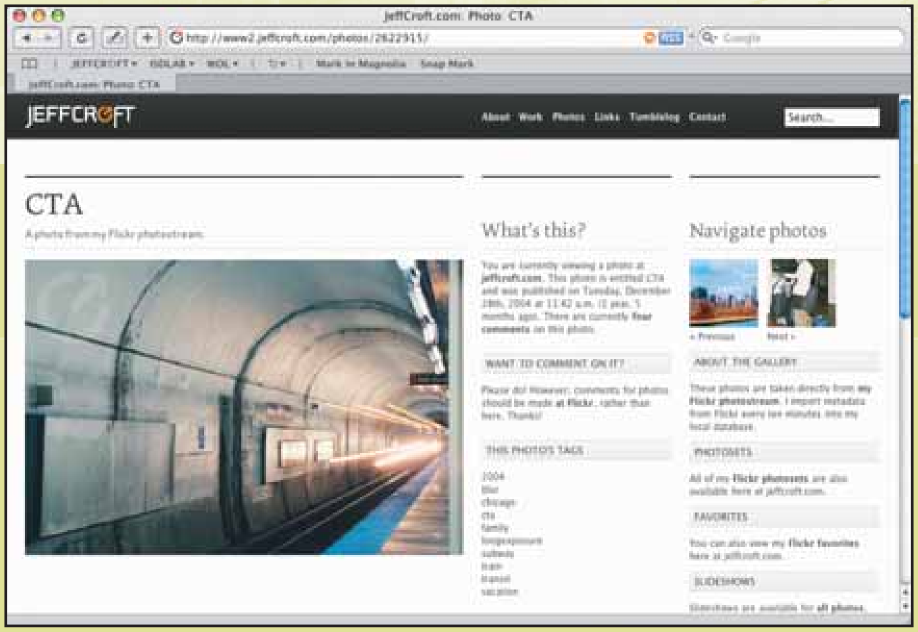


Figure 5-13. Subtle `www.jeffcroft.com` logo mark appears via transparent PNG in the upper-left corner of the photo.

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. (END OF CHAPTER EXCERPT).