

Bonus Chapter 2

Very Basic HTML Code

On a blog, *code* can mean many things. It can refer to the complicated programming that makes up the software that runs your blog, or it can mean simple styles that make written words look cool when displayed on your blog. In this appendix, you can find out how to make the text in your blog posts look like a million bucks. (You may want to leave the blogging software code to the experts for now.)

The code that you're working with affects only a given blog post; the blog software itself builds the blog pages. So, you don't have to know very many pieces of code — just the code commonly used to format text and photos, or to insert video files.



Most of the code that you need to format your blog posts is HTML (HyperText Markup Language).

Documents formatted by using HTML really consist of nothing but text, which your Internet browser can read and then display. HTML code tells your browser all kinds of things, such as how the text on the screen should be aligned and what text links to what web page. You can easily use HTML. With a few exceptions, each bit of code, or *tag*, includes opening and closing elements, which surround the content that you want to affect. A few tags are stand-alone elements that you insert in only one spot. With HTML, you basically use code to mark up text, photos, and other elements.

Adding Headings, Paragraphs, and Line Breaks

You generally want to break up long blocks of text in your blog posts, maybe even throw in labels here and there. Styling headers, adding line breaks, and making sure that you divide the text of your blog into paragraphs can really improve the flow of your posts.

Headings

HTML has six levels of headings. H1 is the most important and largest, H2 is slightly less important and smaller, and so on. The final heading is H6, but people rarely use that heading level because it produces very small text. For most cases, first- and second-level headers do the job, giving you text larger than the main body text (and these headings usually are in bold). You can implement header tags by inserting code like this:

```
<h1>The most important heading ever</h1>
```

Like a light switch, the tag turns on before the first letter of your header and turns off after the last character.

You use all headers in the same way — just substitute in place of the number 1 a 2, 3, 4, 5, or 6. For example

```
<h2>The second most important heading ever</h2>
```

Figure BC2-1 shows each of the header tags as the browser interprets and displays them.

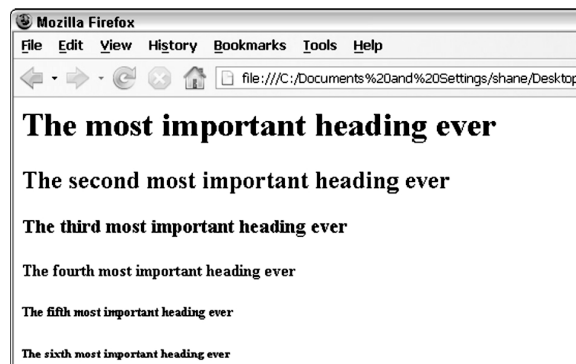


Figure BC2-1:
Headers
from the
largest to
the smallest.



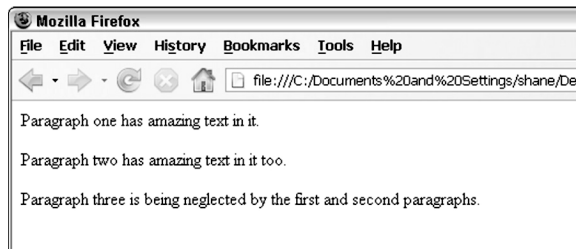
Headers have more than just an effect on how your text looks: Search engines understand that text defined by a header tag is important, so words and phrases that you enclose in a header are marked as especially relevant to your site, and search engine results lists will place your website higher when search terms match words in your headers.

Paragraphs

To create paragraph breaks, you need to start the new paragraph with a `<p>` tag. The closing `</p>` is placed at the end of the paragraph after the text, like this (see Figure BC2-2 for how the browser displays this code):

```
<p>Paragraph one has amazing text in it.</p>
<p>Paragraph two has amazing text in it too.</p>
<p>Paragraph three is being neglected by the first and
second paragraphs.</p>
```

Figure BC2-2:
Text paragraphs, displayed by a web browser.



Most blogging software makes provision for paragraph breaks automatically, but if you ever end up looking at the code, you can recognize these fundamentally important tags. On most websites, paragraph breaks actually produce the effect of two line breaks, so paragraphs are spaced out from each other in the same way that they are in this book.

You can end a line of text and start the next word on a new line by using a `
` tag. You need to insert these stand-alone tags only once: There is no closing `
` tag. To create a line break, here's what you do:

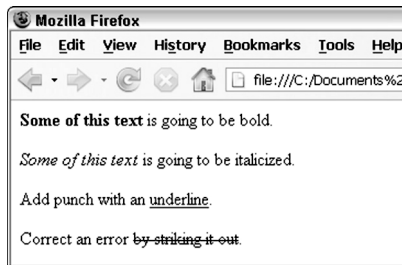
```
Break your line of text<br>
and start again on the next line.
```

Emphasizing Text

Dress up your text by using italics, underlining, bold, or even strikethrough code. You can add these text features by using HTML on/off tags, so you place the first tag at the start of the text that you want to affect and follow the text with the closing tag. Figure BC2-3 shows how each of the following code examples appears in a web browser.

Figure BC2-3:

Use bold, italics, underlining, and strikethrough in your HTML to make your text pop.



Here are the codes for each kind of text:

✓ **Bold:** `` or ``

```
<b>Some of this text</b> is bold.
```

✓ *Italics:* `<i>` or ``

```
<i>Some of this text</i> is italicized.
```

✓ Underline: `<u>`

```
Add punch with an <u>underline</u>.
```

✓ Strikethrough: `<strike>`

```
Correct an error <strike>by striking it out</strike>.
```



Even though the preceding list shows you how to underline text in your blog posts, I don't actually recommend that you use this style. As you know, underlining commonly indicates links online, so your readers may be very confused to see underlined text that isn't also a link.

Linking Up

In HTML, links are defined with the `<a>` tag (the *a* stands for anchor). But that isn't all! This special tag also gets what's called an attribute and a value, to go along with the basic tag. The *attribute*, `href`, indicates to the browser that what comes next is a hypertext reference — in this case, a web page. The *value* is the actual web page that the code links to, enclosed in quotation marks. Here's how all that looks in action:

```
This link goes to <a href=&#x201C;<url href="http://
www.google.com"><span cssStyle="font-
family:monospace">http://www.google.com</
span></url>&#x201D;>Google</a>.
```

If you post the preceding line in a blog, the word *Google* appears as a clickable link that leads to the Google website.



Make sure that you use the full URL when you create a link — including the `http://`! Unless you include that prefix, the browser doesn't understand that it needs to find another website. Instead, it looks for a document on your website called `http://www.google.com`.

Also, when you turn off the tag, you need to turn off only the main tag, not the attribute or value. Those extras are turned off automatically when you use the closing `` tag.

You can also use this tag around an image to make the image clickable to a web page. Simply place the tag around your image code, like so:

```
<a href=&#x201C;<url href="http://www.google.com"><span
cssStyle="font-family:monospace">http://
www.google.com</span></url>&#x201D;><img
src=&#x201C;googlelogo.gif&#x201D;></a>
```

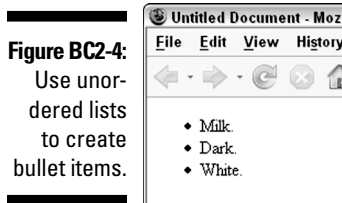
Making Lists

Lists can let people know about your favorite kinds of candy, music, or whatever. With a little help from your friendly HTML, you can create unordered lists or ordered lists. You create all lists, no matter the flavor, by using a series of HTML tags. These tags make lists a bit tricky to implement, so pay close attention to those closing tags — they're easy to forget.

An *unordered list* is a series of bulleted items. You can use this kind of list for items that don't require numbering. For example, if you want to create a list of cat breeds, you don't need to list them in any particular order. Unordered lists use the `` and `` tags. Set up the code as I do in this list of kinds of chocolate:

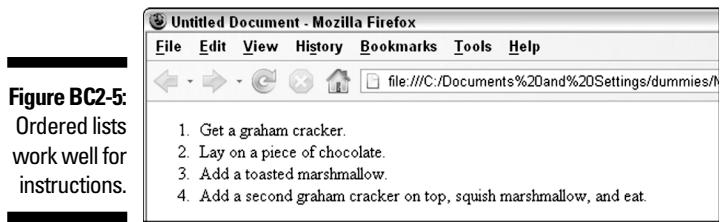
```
<ul>
<li>Milk.</li>
<li>Dark.</li>
<li>White.</li>
</ul>
```

The `` tag stands for *unordered list*, and it turns on at the beginning of the list and turns off following the last item in the list. The `` tag stands for *list item*; it turns on at the beginning of each new item and turns off at the end of that item. Figure BC2-4 shows an unordered list.



An ordered list contains items that use numbers instead of bullet points and are perfect for steps lists or creating lists with content that needs to be ranked. Format an ordered list the same way you format an unordered list (as discussed in the preceding section), but substitute the `` tag for the `` tag (see Figure BC2-5):

```
<ol>
<li>Get a graham cracker.</li>
<li>Lay on a piece of chocolate.</li>
<li>Add a toasted marshmallow.</li>
<li>Add a second graham cracker on top, squish marshmallow, and eat.</li>
</ol>
```



You can embed a list inside another list. In fact, you can even embed a different kind of list into a list. Each time you embed a new list, the web browser creates additional indenting, giving you a nice stair-stepped look in your final page. Here's how you format the code that places an unordered sub-list inside an ordered list in a s'mores recipe:

```
<ol>
<li>Get a graham cracker.</li>
<li>Lay on a piece of chocolate. It can be:
  <ul>
    <li> Milk</li>
    <li> Dark</li>
    <li> White</li>
  </ul>
</li>
<li>Add a toasted marshmallow.</li>
<li>Add a second graham cracker on top, squish
marshmallow, and eat.</li>
</ol>
```

