

Figure 4-6. An image within a heading

In general, images within headings look best at the beginning of the heading, aligned with the bottom or middle of the heading text.

4.3 Changing Text Appearance and Meaning

A number of tags change the appearance of and associate hidden meaning with text. In general, these tags can be grouped into two flavors: content-based styles and physical styles.

In addition, the World Wide Web Consortium (W3C) standard for Cascading Style Sheets (CSS) is now well supported by the popular browsers, providing another, more comprehensive way for authors to control the look and layout of their document text. We describe the tag-based text styles in this chapter. See Chapter 8 for details about CSS.

4.3.1 Content-Based Styles

Content-based style tags inform the browser that the enclosed text has a specific meaning, context, or usage. The browser then formats the text in a manner consistent with that meaning, context, or usage. Note the distinction here. Content-based style tags confer meaning, not formatting. Accordingly, they are important for automated processes; machines don't care what the document looks like—at least for now.

Because font style is specified via semantic clues, the browser can choose a display style that is appropriate for the user. Because such styles vary by locale, using content-based styles helps ensure that your documents will have meaning to a broader range of readers. This is particularly important when a browser is targeted at blind or handicapped readers whose display options are radically different from conventional text or are extremely limited in some way.

The current HTML and XHTML standards do not define a format for each content-based style; they only specify that they must be rendered in a manner different from

the regular text in a document. The standards don't even insist that the content-based styles be rendered differently from one another. In practice, you'll find that many of these tags have fairly obvious relationships with conventional print, having similar meanings and rendered styles, and are rendered in the same style and fonts by most browsers.

4.3.2 Physical Styles

We use the word *intent* a lot when we talk about content-based style tags. That's because the meaning conveyed by the tag is more important than the way a browser displays the text. In some cases, however, you might want the text to appear explicitly in some special way—italic or bold, for example—perhaps for legal or copyright reasons. In those cases, use a physical style for the text.

While the tendency with other text-processing systems is to control style and appearance explicitly, with HTML or XHTML you should avoid physical tags except on rare occasions. Provide the browser with as much contextual information as possible. Use the content-based styles. Even though current browsers may do nothing more than display their text in italic or bold, future browsers and various document-generation tools may use the content-based styles in any number of creative ways.

4.4 Content-Based Style Tags

It takes discipline to use HTML/XHTML content-based style tags because it is easier to simply think of how your text should look, not necessarily what it may also mean. Once you get started using content-based styles, your documents will be more consistent and better lend themselves to automated searching and content compilation.

Content-Based Style Tags	
Function	Alter the appearance of text based upon the meaning, context, or usage of the text
Attributes	<code>class</code> , <code>dir</code> , <code>id</code> , <code>lang</code> , <code>onClick</code> , <code>onDblClick</code> , <code>onKeyDown</code> , <code>onKeyPress</code> , <code>onKeyUp</code> , <code>onMouseDown</code> , <code>onMouseMove</code> , <code>onMouseOut</code> , <code>onMouseOver</code> , <code>onMouseUp</code> , <code>style</code> , <code>title</code>
End tags	Never omitted
Contains	<i>text</i>
Used in	<i>text</i>

4.4.1 The <abbr> Tag

First introduced in HTML 4.0, the <abbr> tag indicates that the enclosed text is an abbreviated form of a longer word or phrase. The browser might use this information to change the way it renders the enclosed text or substitute alternative text. Notice that we said *might*—not all of the popular browsers currently do anything to the text enclosed by the <abbr> tag, and we can't predict how other browsers will implement the tag in the future.

4.4.2 The <acronym> Tag

The <acronym> tag indicates that the enclosed text is an acronym, an abbreviation usually formed from the first letter of each word in a name or phrase, such as HTML and IBM. Like <abbr>, not all browsers change the display of the <acronym> content-based style tag's enclosed text.

4.4.3 The <cite> Tag

The <cite> tag usually indicates that the enclosed text is a bibliographic citation, such as a book or magazine title. By convention, the citation text is rendered in italics. See Figure 4-7 for how Internet Explorer renders this source text:

While kumquats are not mentioned in Melville's
<cite>Moby Dick</cite>, it is nonetheless apparent
that the mighty cetacean represents the bitter
"kumquat-ness" within every man. Indeed, when Ahab
spears the beast, its flesh is tough, much like the noble fruit.

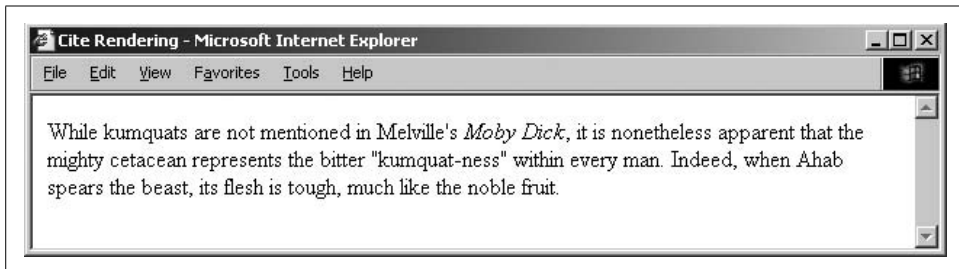


Figure 4-7. Internet Explorer renders <cite> in italics

Use the <cite> tag to set apart any reference to another document, especially those in traditional media, such as books, magazines, journal articles, and the like. If an online version of the referenced work exists, you also should enclose the citation within the <a> tag in order to make it a hyperlink to that online version.

The <cite> tag also has a hidden feature: it enables you or someone else to automatically extract a bibliography from your documents. It is easy to envision a browser that compiles tables of citations automatically, displaying them as footnotes or as a

separate document entirely. The semantics of the `<cite>` tag go far beyond changing the appearance of the enclosed text; they enable the browser to present the content to the user in a variety of useful ways.

4.4.4 The `<code>` Tag

Software code warriors have become accustomed to a special style of text presentation for their source programs. The `<code>` tag is for them. It renders the enclosed text in a monospaced, teletype-style font such as Courier, familiar to most programmers and readers of O'Reilly books such as this one.

This following bit of `<code>`ed text is rendered in a monospaced font style by Firefox, as shown in Figure 4-8 (though the effect is not dramatic, admittedly):

The array reference `a[i]` is identical to the pointer reference `*(a+i)`.

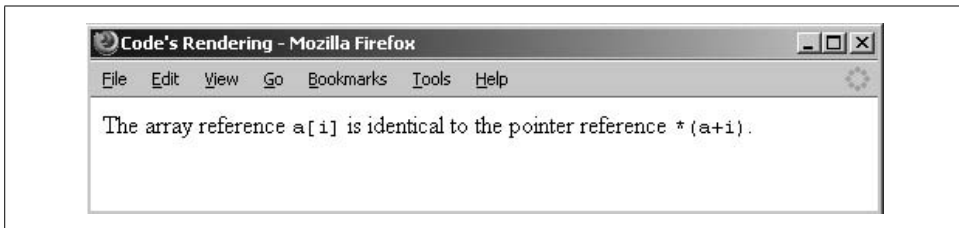


Figure 4-8. Use `<code>` to present computer-speak

You should use the `<code>` tag for text that represents computer source code or other machine-readable content. While the `<code>` tag usually just makes text appear in a monospaced font, the implication is that it is source code, and future browsers may add other display effects.*

For example, a programmer's browser might look for `<code>` segments and perform some additional text formatting, such as special indentation of loops and conditional clauses. If the only effect you desire is a monospaced font, use the `<tt>` tag. If you want to display the programming code in rigidly formatted monospaced text, use the `<pre>` tag. [The `<tt>` Tag, 4.5.10] [`<pre>`, 4.6.5]

4.4.5 The `<dfn>` Tag

Use `<dfn>` to tag defining instances of special terms or phrases. The popular browsers typically display `<dfn>` text in italics. In the future, `<dfn>` might assist in creating a document index or glossary.

* None of the popular browsers format `<code>` segments as a text processor might. Rather, use the `<pre>` tag in conjunction with `<code>` to achieve programming code-like display effects.

For example, use the `<dfn>` tag to introduce a new phrase to the reader:

When analyzing annual crop yields, `<dfn>rind spectroscopy</dfn>` may prove useful. By comparing the relative levels of saturated hydrocarbons in fruit from adjacent trees, rind spectroscopy has been shown to be 87% effective in predicting an outbreak of trunk dropsy in trees under four years old.

Notice that we delimit only the first occurrence of “rind spectroscopy” with a `<dfn>` tag in the example. Good style tells us not to clutter the text with highlighted text. As with the many other, content-related and physical style tags, the fewer the better.* As a general style, especially in technical documentation, set off new terms when they are first introduced to help your readers better understand the topic at hand, but resist tagging the terms thereafter.

4.4.6 The `` Tag

The `` tag tells the client browser to present the enclosed text with emphasis. For nearly all browsers, this means the text is rendered in italic. For example, the popular browsers will emphasize by italicizing the words *always* and *never* in the following HTML/XHTML source:

```
Kumquat growers must <em>always</em> refer to kumquats  
as "the noble fruit," <em>never</em> as just a "fruit."
```

Adding emphasis to your text is tricky business. Too little, and the emphatic phrases may be lost. Too much, and you lose the urgency. Like any seasoning, emphasis is best used sparingly.

Although invariably displayed in italic, the `` tag has broader implications as well, and someday browsers may render emphasized text with a different special effect. The `<i>` tag explicitly italicizes text; use it if all you want is italic. Alternatively, you can include text display-altering cascading style definitions in your document. [The `<i>` Tag, 4.5.4]

Besides for emphasis, also consider using `` when presenting new terms or as a fixed style when referring to a specific type of term or concept. For instance, one of O'Reilly's book styles is to specially format file and device names. You might use the `` tag to differentiate those terms from simple italics used for emphasis.

4.4.7 The `<kbd>` Tag

Speaking of special styles for technical concepts, there is the `<kbd>` tag. As you probably already suspect, it is used to indicate text that is typed on a keyboard. Its enclosed text typically is rendered by the browser in a monospaced font.

* If you need convincing that less is better when applying the content-based and physical style tags, try reading a college textbook in which someone has highlighted what he considered important words and phrases with a yellow marker.

The `<kbd>` tag is most often used in computer-related documentation and manuals, such as in this example:

```
Type <kbd>quit</kbd> to exit the utility, or type
<kbd>menu</kbd> to return to the main menu.
```

4.4.8 The `<samp>` Tag

The `<samp>` tag indicates a sequence of literal characters that should have no other interpretation by the user. This tag is most often used when a sequence of characters is taken out of its normal context. For example, the following source:

```
The <samp>ae</samp> character sequence may be converted
to the &aelig; ligature if desired.
```

is rendered by Netscape, for instance, as shown in Figure 4-9.



Figure 4-9. Setting off sample text using the `<samp>` tag

The special HTML reference for the æ ligature entity is `æ` and is converted to its appropriate æ ligature character by most browsers. For more information, see Appendix F.

The `<samp>` tag is not used very often. You should use it in those few cases where special emphasis needs to be placed on small character sequences taken out of their normal context.

4.4.9 The `` Tag

Like the `` tag, the `` tag is for emphasizing text, except with more gusto. Browsers typically display the `` tag differently than the `` tag, usually by making the text bold (versus italic) so that users can distinguish between the two. For example, in the following text, the emphasized “never” appears in italic by Opera, and the `` “forbidden” is rendered in bold characters (see Figure 4-10):

```
One should <em>never</em> make a disparaging remark about the
noble fruit. In particular, mentioning kumquats in conjunction
with vulgar phrases is expressly <strong>forbidden</strong> by
the Association bylaws.
```

If common sense tells us that the `` tag should be used sparingly, the `` tag should appear in documents even more infrequently. `` text is like shouting.

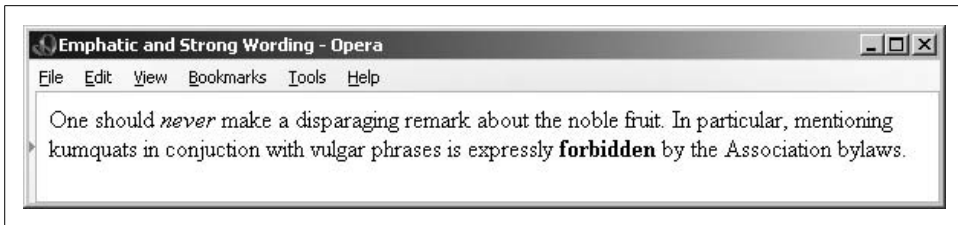


Figure 4-10. Strong and emphasized text are rendered differently

`` text is nothing short of a scream. Like a well-chosen epithet voiced by an otherwise taciturn person, restraint in the use of `` makes its use that much more noticeable and effective.

4.4.10 The `<var>` Tag

The `<var>` tag, another computer-documentation trick, indicates a variable name or a user-supplied value. The tag is often used in conjunction with the `<code>` and `<pre>` tags for displaying particular elements of computer-programming code samples and the like. Browsers typically render `<var>`-tagged text in italics, as shown in Figure 4-11, which displays the following example:

```
The user should type
<pre>
  cp <var>source-file</var> <var>dest-file</var>
</pre>
replacing the <var>source-file</var> with the name of
the source file, and <var>dest-file</var> with the name
of the destination file.
```

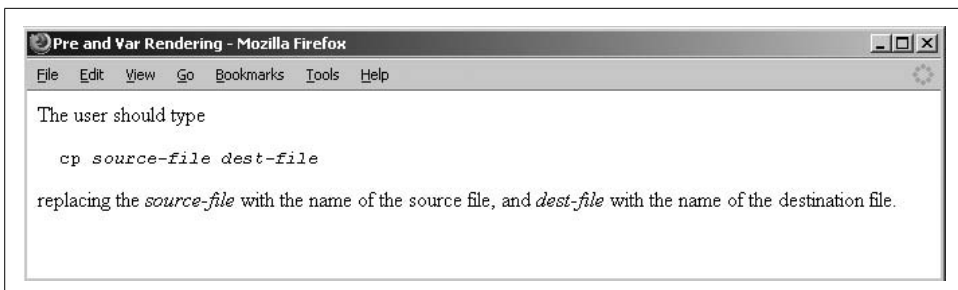


Figure 4-11. The `<var>` tag typically appears in preformatted (`<pre>`) computer code

Like the other computer-programming and documentation-related tags, the `<var>` tag not only makes it easy for users to understand and browse your documentation, but automated systems might someday use the appropriately tagged text to extract information and useful parameters mentioned in your documents. Once again, the more semantic information you provide to your browser, the better it can present that information to the user.

4.4.11 The class, style, id, and title Attributes

Although each content-based tag has a default display style, you can override that style by defining a new look for each tag. You can apply this new look to the content-based tags using either the `style` or the `class` attribute. [Inline Styles: The style Attribute, 8.1.1] [Style Classes, 8.3]

You also may assign a unique identifier (`id`) to the content-based style tag, as well as a less rigorous title, using the respective attributes and their accompanying quote-enclosed string values. [The `id` attribute, 4.1.1.4] [The `title` attribute, 4.1.1.5]

4.4.12 The dir and lang Attributes

The `dir` attribute advises the browser in which direction the text within the content-based style tag should be displayed, and `lang` lets you specify the language used within the tag. [The `dir` attribute, 3.6.1.1] [The `lang` attribute, 3.6.1.2]

4.4.13 Event Attributes

Things happen in and around a content-based tag's content, and, with the respective on attribute and value you may react to that event by displaying a user dialog or activating some multimedia event. [JavaScript Event Handlers, 12.3.3]

4.4.14 Summary of Content-Based Tags

The various graphical browsers render text inside content-based tags in similar fashion; text-only browsers such as Lynx have consistent styles for the tags. Table 4-1 summarizes these browsers' display styles for the native tags. However, stylesheet definitions may override these native display styles.

Table 4-1. Content-based tags

Tag	Netscape Navigator	Internet Explorer	Mozilla Firefox	Opera	Lynx
<abbr>	N/A	N/A	N/A	N/A	N/A
<acronym>	N/A	N/A	N/A	N/A	N/A
<cite>	<i>italic</i>	<i>italic</i>	<i>italic</i>	<i>italic</i>	monospace
<code>	monospace	monospace	monospace	monospace	monospace
<dfn>	<i>italic</i>	<i>italic</i>	<i>italic</i>	<i>italic</i>	N/A
	<i>italic</i>	<i>italic</i>	<i>italic</i>	<i>italic</i>	monospace
<kbd>	monospace	monospace	monospace	monospace	monospace
<samp>	monospace	monospace	monospace	monospace	monospace
	bold	bold	bold	bold	monospace
<var>	<i>italic</i>	<i>italic</i>	<i>italic</i>	<i>italic</i>	monospace

4.4.15 Allowed Content

Any content-based style tag may contain any item allowed in text, including conventional text, anchors, images, and line breaks. In addition, other content-based and physical style tags can be embedded within the content.

4.4.16 Allowed Usage

Any content-based style tag may be used anywhere an item allowed in text is used. In practice, this means you can use the ``, `<code>`, and other similar tags anywhere in your document except inside `<title>`, `<listing>`, and `<xmp>` tagged segments. You can use text style tags in headings, too, but their effects may be overridden by the effects of the heading tags themselves.

4.4.17 Combining Content-Based Styles

It may have occurred to you to combine two or more of the various content-based styles to create interesting and perhaps even useful hybrids. Thus, an emphatic citation might be achieved with:

```
<cite><em>Moby Dick</em></cite>
```

In practice, Dr. Frankenstein, the browser usually ignores the monster—as you can test by typing and viewing the example yourself, “Moby Dick” gets the citation without emphasis.

The HTML and XHTML standards do not require the browser to support every possible combination of content-based styles and do not define how the browser should handle such combinations. Someday maybe; for now, it’s best to choose one tag.

4.5 Physical Style Tags

The current HTML and XHTML standards currently provide nine physical styles: bold, italic, monospaced, underlined, strikethrough, larger, smaller, superscripted, and subscripted text. Much to our relief, Internet Explorer has stopped supporting a tenth physical style, “blinking” text. We wish the others would “get it.” All physical style tags require ending tags.

As we discuss physical tags in detail, keep in mind that they convey an acute styling for the immediate text. For more comprehensive, document-wide control of text display, use stylesheets (see Chapter 8).

4.5.1 The `` Tag

The `` tag is the physical equivalent of the `` content-based style tag, but without the latter’s extended meaning. The `` tag explicitly boldfaces a character or segment of text that is enclosed between it and its corresponding end tag (``). If a boldface font is not available, the browser may use some other representation, such as reverse video or underlining.

Physical Style Tags

Function	Specify physical styles for text
Attributes	class, dir, id, lang, onClick, onDbClick, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, style, title
End tags	Never omitted
Contains	<i>text</i>
Used in	<i>text</i>

4.5.2 The <big> Tag

The <big> tag makes it easy to increase the size of text. It couldn't be simpler: the browser renders the text between the <big> tag and its matching </big> ending tag one font size larger than the surrounding text. If that text is already at the largest size, <big> has no effect. [, 4.10.3]

Even better, you can nest <big> tags to enlarge the text. Each <big> tag makes the text one size larger, up to a limit of size seven, as defined by the font model.

4.5.3 The <blink> Tag (Obsolete Extension)

Text contained between the <blink> tag and its end tag, </blink>, does just that: it blinks on and off. Firefox, for example, simply and reiteratively reverses the background and foreground colors for the <blink>-enclosed text. Neither the HTML nor the XHTML standard includes <blink>. Originally, it was supported as an extension only by Netscape Navigator versions before version 6; then it was dropped in version 6, and was reinstated in versions 7 and later. Opera and Firefox support it, too—only Internet Explorer eschews it. You should, too.

We cannot effectively reproduce the animated effect in these static pages, but it is easy to imagine and best left to the imagination, too. Blinking text has two primary effects: it gets your readers' attention and then promptly annoys them to no end. Forget about blinking text.

4.5.4 The <i> Tag

The <i> tag is like the content-based style tag. It and its necessary end tag (</i>) tell the browser to render the enclosed text in an italic or oblique typeface. If the typeface is not available to the browser, highlighting, reverse video, or underlining might be used.

4.5.5 The <s> Tag (Deprecated)

The <s> tag is an abbreviated form of the <strike> tag supported by all current browsers even though it is deprecated in HTML 4 and XHTML. In other words, the “s” stands for shy: don’t use it; <s> will go away, eventually.

4.5.6 The <small> Tag

The <small> tag works just like its <big> counterpart (see [The <big> Tag, 4.5.2]), except it decreases the size of text instead of increasing it. If the enclosed text is already at the smallest size supported by the font model, <small> has no effect.

As you can with <big>, you can nest <small> tags to sequentially shrink text. Each <small> tag makes the text one size smaller than the containing <small> tag, to a limit of size 1.

4.5.7 The <strike> Tag (Deprecated)

The popular browsers put a line through (“strike through”) text that appears inside the <strike> tag and its </strike> end tag. Presumably, it is an editing markup that tells the reader to ignore the text passage, reminiscent of the days before typewriter correction tape. You’ll rarely, if ever, see the tag in use today: it is deprecated in HTML 4 and XHTML, just one step away from complete elimination from the standard.

4.5.8 The <sub> Tag

The text contained between the _{tag and its} end tag gets displayed half a character’s height lower, but in the same font and size as the current text flow. Both <sub> and its <sup> counterpart are useful for math equations and in scientific notation, as well as with chemical formulæ.

4.5.9 The <sup> Tag

The ^{tag and its} end tag superscript the enclosed text; it gets displayed half a character’s height higher, but in the same font and size as the current text flow. This tag is useful for adding footnotes to your documents, along with exponential values in equations. When you use it in combination with the <a> tag, you can create nice, hyperlinked footnotes:

```
The larval quat  
weevil<a href="footnotes.html#note74"><sup><small>74</small></sup></a> is a
```

This example assumes that *footnotes.html* contains all your footnotes, appropriately delimited as named document fragments.

4.5.10 The <tt> Tag

Like the <code> and <kbd> tags, the <tt> tag and its necessary </tt> end tag direct the browser to display the enclosed text in a monospaced typeface. For those browsers

that already use a monospaced typeface, this tag may make no discernible change in the presentation of the text.

4.5.11 The `<u>` Tag (Deprecated)

This tag tells the browser to underline the text contained between the `<u>` and the corresponding `</u>` tag. The underlining technique is simplistic, drawing the line under spaces and punctuation as well as the text. This tag is deprecated in HTML 4 and XHTML, but the popular browsers support it.

The same display effects for the `<u>` tag are better achieved by using stylesheets, covered in Chapter 8.

4.5.12 The `dir` and `lang` Attributes

The `dir` attribute lets you advise the browser in which direction the text within the physical tag should be displayed, and `lang` lets you specify the language used within the tag. [The `dir` attribute, 3.6.1.1] [The `lang` attribute, 3.6.1.2]

4.5.13 The `class`, `style`, `id`, and `title` Attributes

Although each physical tag has a defined style, you can override that style by defining your own look for each tag. You can apply this new look to the physical tags using either the `style` or the `class` attribute. [Inline Styles: The `style` Attribute, 8.1.1] [Style Classes, 8.3]

You also may assign a unique ID to the physical style tag, as well as a less rigorous title, using the respective attribute and accompanying quote-enclosed string value. [The `id` attribute, 4.1.1.4] [The `title` attribute, 4.1.1.5]

4.5.14 Event Attributes

As with content-based style tags, user-initiated mouse and keyboard events can happen in and around a physical style tag's contents. The browser recognizes many of these events if it conforms to current standards, and with the respective `on` attribute and value, you may react to the event by displaying a user dialog box or activating some multimedia event. [JavaScript Event Handlers, 12.3.3]

4.5.15 Summary of Physical Style Tags

The various graphical browsers render text inside the physical style tags in a similar fashion. Table 4-2 summarizes these browsers' display styles for these tags. Stylesheet definitions may override these native display styles.

Table 4-2. Physical style tags

Tag	Meaning	Display style
	Bold contents	Bold
<big>	Increased font size	Bigger text
<blink> (obsolete)	Alternating fore- and background colors	Blinking text
<i>	Italic contents	<i>Italic</i>
<small>	Decreased font size	Smaller text
<s>, <strike> (deprecated)	Strikethrough text	Strike
<sub>	Subscripted text	subscript
<sup>	Superscripted text	superscript
<tt>	Teletypewriter style	monospaced
<u> (deprecated)	Underlined contents	Underlined

The following HTML source example illustrates some of the various physical tags as rendered by Firefox (see Figure 4-12):

```
Explicitly <b>boldfaced</b>, <i>italicized</i>, or
<tt>teletype-style</tt> text should be used
<big><big>sparingly</big></big>.
Otherwise, drink <strike>lots</strike> 1x10<sup>6</sup>
drops of H<sub><small><small>2</small></small></sub>O.
```

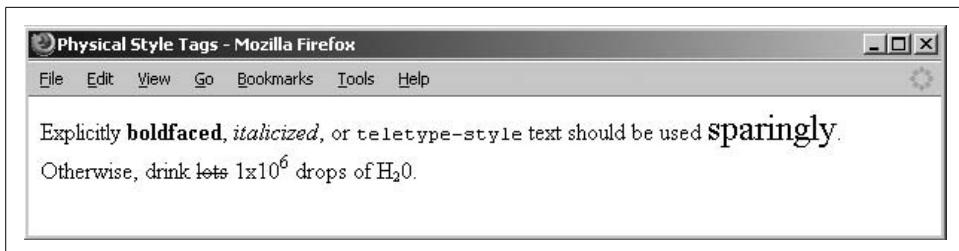


Figure 4-12. Use physical text tags with caution

4.5.16 Allowed Content

Any physical style tag may contain any item allowed in text, including conventional text, anchors, images, and line breaks. You can also combine physical style tags with other content-based tags.

4.5.17 Allowed Usage

You can use any physical style tag anywhere you can use an item allowed in text. In general, this means anywhere within a document, except in the <title>, <listing>, and <xmp> tags. You can use a physical style tag in a heading, but the browser will probably override and ignore its effect in lieu of the heading tag.

4.5.18 Combining Physical Styles

You will probably have better luck combining physical tags than you might have combining content-based tags to achieve multiple effects. For instance, all the popular browsers render the following in bold and italic typeface:

```
<b><i>Thar she blows!</i></b>
```

Other browsers may elect to ignore such nesting. The standards require the browser to “do its best” to support every possible combination of styles, but do not define how the browser should handle such combinations. Although most browsers make a good attempt at doing so, do not assume all combinations will be available to you.

4.6 Precise Spacing and Layout

CSS notwithstanding, the original concept of HTML is for specifying document content without indicating format; to delineate the structure and semantics of a document, not how that document is to be presented to the user. Normally, you should leave word wrapping, character and line spacing, and other presentation details up to the browser. That way, the document’s content—its rich information, not its good looks—is what matters. When looks matter more, such as for commercial presentations, look to stylesheets for layout control (see Chapter 8).

4.6.1 The
 Tag

The
 tag interrupts the normal line filling and word wrapping of paragraphs within an HTML or XHTML document. It has no ending tag with HTML;* it simply marks the point in the flow where a new line should begin. Most browsers simply stop adding words and images to the current line, move down and over to the left margin, and resume filling and wrapping.

	
Function	Inserts a line break into a text flow
Attributes	class, clear, id, style, title
End tag	None in HTML; </br> or <br ... /> in XHTML
Contains	Nothing
Used in	<i>text</i>

* With XHTML, put the end inside the start tag:
. See Chapter 16 for details.