

# CHAPTER 3: COMPLEX TAGS

“A complex system that works is invariably found to have evolved from a simple system that worked.”

– John Gall

If you are used to creating heavily formatted documents with a word processor, you may be a little disappointed with the limitations of HTML.

## SIMPLE TAGS

So far, we have examined simple tags. Simple tags are of the form:

```
<tag>text</tag>
```

such as:

```
<h1>This headline is big and bold</h1>
```

The keyword inside the tag is known as the selector:

- `<h1>` is the tag.
- `h1` is the selector.

You can add an argument to a tag:

```
<tag property1="value" property2="value"
. . . propertyx="value">text</tag>
```

where the argument consists of one or more parameters. Each parameter comprises:

- a property, which identifies the affected aspect of the selector; and, usually,
- a value, which specifies which of the possible characteristics this property will assume.

## PROPERTY RULES

The rules for setting properties are not complicated:

- Separate the properties from the selector and from other properties with a space.
- Elements of the argument may appear in any order.
- Values should be placed within double quotation marks.

You may discover that if you forget the quotation marks, neither Internet Explorer nor Netscape® Navigator™ care. Still, you should get in the habit of placing any value within quotation marks.

Although IE and Navigator dominate the browser market, they are by no means alone—and other browsers may not treat the missing quotation marks with the same indifference. In addition, placing values within quotation

marks makes your code easier to read and to edit. Most importantly, though, specifications for some advanced features require values to be encased in quotation marks.

## CHANGE THE FONT

To illustrate the concept of properties, let's examine `<font>` and its argument. Here's its general form:

```
<font argument>text</font>
```

where the *argument* may include:

- `size="+4", "+3", "+2", "+1", "-1", or "-2"`
- `color="#rrggbb"` or `"colorname"` (we will work with colors in Chapter 4)
- `face="Arial"` (or other font name)

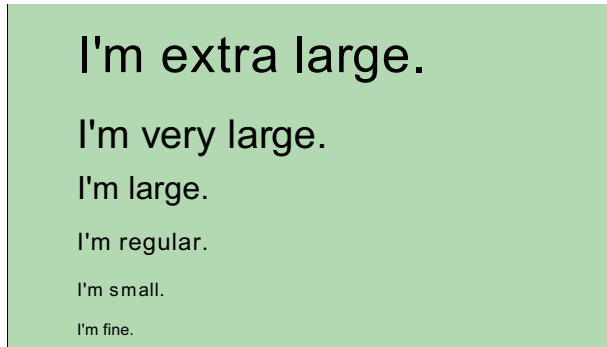
Remember, the parameters of the argument may appear in any order. Parameters should be typed in lower case, to comply with the XHTML standard, but their corresponding values are case-insensitive. You must type a space between the selector and each parameter. Finally, the values—that is, whatever follows an equal sign (=), should be placed in quotation marks. Let's examine these parameters.

## FONT SIZE

Font sizes are relative to the base font size. For example:

```
<font size="+3">I'm extra large.</font>
<br />
<font size="+2">I'm very large.</font>
<br />
<font size="+1">I'm large.</font><br />
I am regular.<br />
<font size="-1">I'm small.</font><br />
<font size="-2">I'm fine.</font>
```

appears as:



### FONT FACE

You can use the `<font>` tag's *face* parameter to specify the font you want to use. If you specify a font that does not exist on your visitor's machine, his or her browser will substitute a default browser font, usually a serif font similar to Times New Roman or a sans serif font similar to Arial.

The solution is to list several fonts. If the first one is not on the visitor's system, it tries the second. If it doesn't have the second font, it tries the third, and so on. For example, you might specify:

```
<font face="Arial, Helvetica, sans-serif">
text</font>
```

In this case, a Windows<sup>®</sup> browser will display *text* in Arial, because all Windows<sup>®</sup> PCs have the Arial font. A Macintosh<sup>®</sup> computer doesn't have the Arial font, so its browser will display *text* in Helvetica<sup>®</sup>. If the platform has neither Arial nor Helvetica<sup>®</sup>, it will display *text* in its generic sans-serif font.

That's fine for the basic Web fonts, but there is no easy solution for decorative fonts. The most common solution is to convert text into a graphic, and then to display the graphic, as demonstrated in Lab 2, 3, 3.

Note that the standard Macintosh fonts<sup>®</sup>—Chicago, Courier, Geneva, Helvetica<sup>®</sup>, Monaco, New York, Palatino<sup>®</sup>, and Times<sup>®</sup>—sometimes have homophones in Windows<sup>®</sup>, but they are not the same:

Courier ≠ Courier New

Times ≠ Times New Roman ≠ Roman

In other words, to invoke Courier New in Windows<sup>®</sup> and Courier in Macintosh<sup>®</sup>, you must specify:

```
<font face="'Courier New', Courier">
```

Note that two-word font names, such as “Courier New” must be placed inside single quotes.

### CREATE LINES

Lines, known as “horizontal rules,” are created with the `<hr />` tag. It is an empty tag, so there is no corresponding turn-off tag. For empty tags, we type a space and a forward slash, `/`, after the selector. The tag has several optional parameters:

```
<hr argument />
```

where the *argument* may include these parameters:

- `size="x"` (height in pixels)
- `width="y"` (in pixels) or `"y%"`
- `align="left"`, `"right"`, or `"center"`
- `color="#rrggbb"` or `"colorname"` (we will work with colors in Chapter 4)
- `noshade`

For example:

```
<hr size="5" width="50%" align="center" />
```

creates a three-dimensional, horizontal line, five pixels high, half the width of its frame, and centered. The tag:

```
<hr />
```

without an argument, creates a three-dimensional, horizontal line that is as wide as its frame.

By default, lines are shaded, creating a three-dimensional effect. Include the *noshade* option to turn this off. The *noshade* parameter is unusual, because it is not set equal to a corresponding value. Just the word *noshade* is all you need.

### CREATING LISTS

Now we turn our attention to so called complex tags. They are not called “complex” because they are complicated. They are called complex because they come in two or more parts.

To illustrate this, we now introduce lists. Three kinds of lists—unordered, ordered, and definitions—are possible.

#### UNORDERED LISTS

To create an unordered list, start with an `unordered list` tag, `<ul>`. For each item on the list, enter the List Item tag, `<li>`, followed by the item:

```
Great Yankees
<ul>
  <li>Mattingly</li>
  <li>Mantle</li>
  <li>Rizzuto</li>
  <li>Berra</li>
  <li>DiMaggio</li>
  <li>Gehrig</li>
  <li>Ruth</li>
</ul>
```

As you can see, each item on the list ends with an `</li>` tag, and you must end the list with a `</ul>` tag. This will appear as:

```
Great Yankees
  • Mattingly
  • Mantle
  • Rizzuto
  • Berra
  • DiMaggio
  • Gehrig
  • Ruth
```

Also note how I pressed `⌘Tab` to indent the `<li>` tags. As you know, HTML completely ignores white space, including tabs—so this has no effect whatsoever on the appearance of the document. We do it because, as tags become more and more complicated, it will help us keep our turn-ons and turn-offs straight. Can you imagine trying to debug this:

```
<ul><li>Mattingly</li><li>Mantle</li><li>
Rizzuto</li><li>Berra</li><li>DiMaggio</li>
<li>Gehrig</li>Ruth</li></ul>
```

even though it will produce identical results?!! (Did you notice there is an `<li>` tag missing just before Ruth!) Indenting complex tags makes it much easier to read, easier to debug, and easier to edit.

### ORDERED LISTS

An ordered list uses the ordered list `<ol>` tag. Items within the list are tagged using the `<li>` tag, just as with unordered lists:

```
<h2>The Standings</h2>
<ol>
  <li>Yankees</li>
```

```
<li>Orioles</li>
<li>Red Sox</li>
<li>Devil Rays</li>
<li>Blue Jays</li>
</ol>
```

will appear as:

```
The Standings:
  1. Yankees
  2. Orioles
  3. Red Sox
  4. Devil Rays
  5. Blue Jays
```

### DEFINITION LISTS

A definition list begins with a `<dl>`, followed by alternating `<dt>` (term) and `<dd>` (definition) entries. The `<dt>` and `<dd>` elements can contain any number of paragraphs or lists. Here is an example:

```
<dl>
  <dt>ALU</dt>
  <dd>arithmetic logic unit</dd>
  <dt>AI</dt>
  <dd>Artificial Intelligence</dd>
  <dt>WYSIWYG</dt>
  <dd>What You See Is What You Get</dd>
</dl>
```

will appear as:

```
ALU
  arithmetic logic unit
AI
  Artificial Intelligence
WYSIWYG
  What You See Is What You Get
```

### HANGING INDENTS

Definition Lists are often ignored. Still, they have some quirky features that can be very useful. Let's look at the previous example, with one change—add the *compact* argument to the `<dl>` tag:

```
<dl compact>
  <dt>ALU</dt>
  <dd>arithmetic logic unit</dd>
  <dt>AI</dt>
  <dd>Artificial Intelligence</dd>
  <dt>WYSIWYG</dt>
  <dd>What You See Is What You Get</dd>
</dl>
```

Here's the result:

```
ALU  arithmetic logic unit
AI   Artificial Intelligence
WYSIWYG
      What You See Is What You Get
```

Do you see how this places everything on the same line? Everything, that is, except the WYSIWYG line. The `<dl>` tag's *compact* parameter places the definition on the same line as the term, *but only if the term is four characters or less!* What good is that?

Here's one way it can be used. Let's say you have a résumé like this:

```
<dl compact>
  <dt>2002</dt>
  <dd>Consultant</dd>
  <dt>1995</dt>
  <dd>Designed and manufactured
  aluminum-based, reflective vehicle
  identification devices at a large
  state facility.</dd>
  <dt>1992</dt>
  <dd>Pharmaceutical wholesaling</dd>
  <dt>1991</dt>
  <dd>Jewelry wholesaling</dd>
</dl>
```

Here is the result. Notice that this not only places jobs on the same line as the year, it also wraps at the indent, creating "hanging indents."

```
2002  Consultant
1995  Designed and manufactured aluminum-
      based, reflective vehicle identification
      devices at a large state facility.
1992  Pharmaceutical wholesaling
1991  Jewelry wholesaling
```

## NESTED LISTS

Lists can be combined and nested in outline form.

```
Groceries:
<ul>
  <li>Meat:</li>
  <ul>
    <li>Hamburger</li>
    <li>Steak</li>
  </ul>
  <li>Cheese:</li>
  <ul>
    <li>Swiss</li>
  </ul>
</ul>
```

```
Groceries:
  • Meat:
    ◦ Hamburger
    ◦ Steak
  • Cheese:
    ◦ Swiss
```

## LIST FORMATTING

### ORDERED LIST FORMATTING

You can change the numbering system for an ordered list.

The full syntax for the `<ol>` tag is:

```
<ol type="x" start="y">
```

where *x* may be:

- "1" (Arabic numeral—the default);
- "A" (uppercase alphabetical);
- "a" (lowercase alphabetical);
- "I" (uppercase Roman numeral); or,
- "i" (lowercase Roman numeral).

and *y* is the start value of the list (the default is 1, A, a, I, or i, depending on the type value).

For example:

```
Groceries:
<ol type="A">
  <li>Eggs</li>
  <li>Milk</li>
  <li>Bread</li>
  <li>Salt</li>
</ol>
```

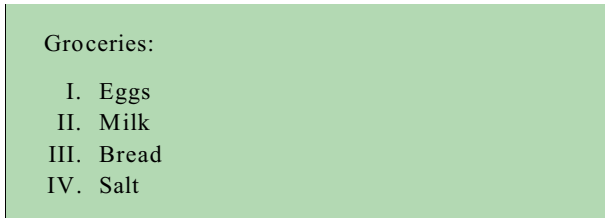
will appear as:



Similarly:

```
Groceries:
<ol type="I">
  <li>Eggs</li>
  <li>Milk</li>
  <li>Bread</li>
  <li>Salt</li>
</ol>
```

will appear as:



### UNORDERED LIST FORMATTING

You can change the bullets that appear in an unordered list with the type parameter. The full syntax for the <ul> tag is:

```
<ul type="z">
```

where z may be:

- "circle";
- "disc"; or,
- "square".

For instance, the code:

```
Groceries:
<ul type="circle">
  <li>Eggs</li>
  <li>Milk</li>
  <li>Bread</li>
</ul>
```

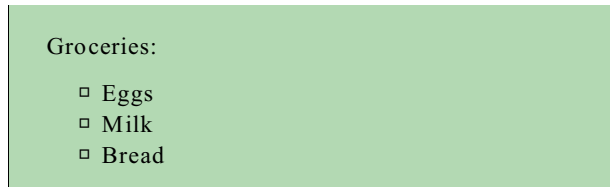
will appear as:



Similarly, the code:

```
Groceries:
<ul type="square">
  <li>Eggs</li>
  <li>Milk</li>
  <li>Bread</li>
</ul>
```

will appear as:



### CASE SENSITIVE

Note that the value of the type parameter is case sensitive:

- type="A" in an ordered list, creates a list of upper-case letters.
- type="a" in an ordered list, creates a list of lower-case letters.
- type="disc" in an unordered list, creates a bullet list.
- type="DISC" does nothing (because it is upper-case)!

### SPECIAL CHARACTERS

Here's a concept: when you publish a document on the *World Wide* Web, it can be read by people *world-wide*. That means it can be read by people who think you are stupid if you wrote the following:

When he arrived at the cafe in Dusseldorf, Francois ordered a souffle for himself and crepes with sauted mushrooms and a jalapeno pate for his fiancee. They shared a

Chateaubriand over entrees while a flugelhorn etude by Bela Bartok played softly. Reexamining his bill with elan, he summoned the garcon. Although they had tickets for Aida—or was it Les Miserables?—he could not resist an apercu of Haagen-Dazs.

At a chateau in Nurnberg, meeting with Nestle officials, he had lost his attache case. He would have to phone his naive protege, Jose, who had coordinated his trip.<sup>1</sup>

Escape sequences are used to create special characters that cannot be typed on a standard keyboard. We can create escape sequences, sometimes called “character entities,” by typing `&`, `#`, the ASCII number, and a `;`. For example, to get José, with the acute accent over the e, you would type:

`Jos&#233;`

Escape sequences *are* case sensitive, unlike tags:

- `&egrave;` is the escape sequence for è
- `&Egrave;` is the escape sequence for È

#### UNDERSTANDING ESCAPE SEQUENCE STRUCTURE

HTML escape sequences follow a pattern. If we examine `&#233;`:

`&`      `#`      `233`      `;`  
 |      |      |      |  
 Ampersand   # sign   ASCII #   Semi-colon

Because remembering these number is a daunting task, HTML gives you an mnemonic equivalent for many symbols. For example, you could also get José by typing:

`Jos&eacute;`

It’s easier to use the mnemonic equivalents if you know the proper name for various diacritics. For example, ~ is a *tilde*, not a “squiggle.” That makes it easier to understand why ñ is produced by `&ntilde;` and ã is `&atilde;`.

DIACRITIC	LOOKS LIKE	APPEARS AS	HTML MNEMONIC
acute accent	´	á	<code>&amp;aacute;</code>
“ring”	°	â	<code>&amp;aring;</code>
cedilla	¸	ç	<code>&amp;ccedil;</code>
circumflex	ˆ	â	<code>&amp;acirc;</code>
grave accent	`	à	<code>&amp;agrave;</code>
ligature	two characters, joined	æ	<code>&amp;aelig;</code>
tilde	~	ñ	<code>&amp;ntilde;</code>
umlaut	¨	ä	<code>&amp;auml;</code>

These mnemonics follow a pattern. For example, if we dissect `&aacute;`; it becomes:

`&`      `a`      `acute`      `;`  
 |      |      |      |  
 Ampersand   Letter   Key word   Semi-colon

See Appendix C for a list of escape sequences and mnemonic equivalents.

#### INTERNATIONAL

An escape sequence can result in some pretty weird looking code:

`Jos&eacute; Ord&oacute;&ntilde;ez`

but it has happy results:

José Ordóñez

The international aspect of the Internet, not to mention the intercultural relations of our ever-shrinking planet, make the effort worthwhile. For example, to many Americans, raised to think that anything you can’t type on a typewriter is foreign and therefore suspect, words with special characters are commonly misspelled.

In Spanish, the tilde—the squiggle over the “ñ” in el Niño—creates a distinct letter. Saying that the letter “ñ” is the same as the letter “n” is like saying the letter “y” is the same as “v.” It’s close, but, as the saying goes, close only counts in horseshoes and hand grenades. Misspelling words, particularly names, is lazy at best and insulting at worst. Make the effort to do it right.

#### NON-BREAKING SPACE

When a word doesn’t fit on a line, browsers automatically wrap the text to the next line. Unlike word processors, browsers do not hyphenate; the break occurs at the nearest space. Sometimes, we do not want word wrap to separate two words, as in the case of names. To prevent this, we can place the non-breaking space character:

`&nbsp;`

between them. For example:

This is another example of how  
Ludwig van&nbsp;Beethoven . . . .

appears as:

This is another example of how Ludwig van Beethoven . . . .

The non-breaking space keeps “van” and “Beethoven” together on the same line, preventing:

This is another example of how Ludwig van Beethoven . . . .

Remember that escape sequences are case sensitive:

**van&NBSP;Beethoven**

will do nothing! It must be in lower case.

### ADDING SPACE

You will recall from Chapter 2 that most browsers ignore multiple `<p>` or `<br />` tags. One way to overcome this limitation is to use the non-breaking space character:

**Hello<br />&nbsp;<br />&nbsp;<br />Wayne!**

will appear as:

Hello

Wayne!

We can also use non-breaking spaces to indent paragraphs. For example:

**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Hi!<br />  
This is not indented.**

will appear as:

Hi!

This is not indented.

## UNIVERSE OF TEMPLATES

Every day you will see some cool new trick at a Web site that you don’t know how to reproduce. You can easily discover how they did it: from your browser’s pull-down menu, select View, and then:

- Document Source;
- Source;
- Page Source; or,
- Frame Source.

(The choice that is available will depend on the browser and the nature of the page.) The active document, and all of its tags, will appear.

## NOTES

1. When he arrived at the café in Düsseldorf, François ordered a soufflé for himself and crêpes with sautéed mushrooms, and jalapeño pâté for his fiancée. They shared a Châteaubriand over entrées while a flügelhorn étude by Béla Bartók played softly. Reexamining his bill with élan, he summoned the garçon. Although they had tickets for Aïda—or was it Les Misérables?—he could not resist an aperçu of Häagen-Dazs.

At a château in Nürnberg, meeting with Nestlé officials, he had lost his attaché case. He would have to phone his naïve protégé, José, who had coordinated his trip.



# CHAPTER 4: COLOR

“Grey is a colour that always seems on the eve of changing to some other colour.”

– G. K. Chesterton

**Y**ou can control five elements of a Web page: the background color, the color of text, and the colors of unvisited links, active links, and visited links.

## PRIMARY COLORS

Primary colors are three colors from which all other colors can be produced. In school, perhaps you were taught that the primary colors are red, yellow, and blue.

Now, it is true that, by using just these three colors, we can produce all the colors of the rainbow. Still, these are *not* the only primary colors. Off-set printers, combining ink to create various hues, use a different set of primary colors: yellow, magenta, and cyan. You may have heard of this set referred to as the subtractive model.

This brings us to computer monitors, which use another set of primary colors: red, green, and blue. This is sometimes called the RGB method or additive model.

## HEXADECIMAL

Colors are often expressed in hexadecimal color codes. In the decimal, or Base-10 system, we count like this:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Hexadecimal is a Base-16 number system. Just as Base-10 doesn't have a unique character for ten, Base-16 doesn't have a unique character for sixteen—but it does have unique characters for ten, eleven, twelve, thirteen, fourteen, and fifteen:

1 2 3 4 5 6 7 8 9 A B C D E F 10


In hexadecimal, B is another way of writing 11 and E is another way of writing 14.

## CHECK THE RESULTS

Let's use the Windows<sup>®</sup> Calculator to check our results. First, let's test the assertion that:

$$14_{10} = E_{16}$$

**Step 1:** Open Calculator:

- From the Taskbar, click  **Start**, Programs, Accessories, Calculator.

**Step 2:** If you are in the Standard view, switch to the Scientific view:

- from the pull-down menu, select **V**iew, **S**cientific, as shown in Figure 12.

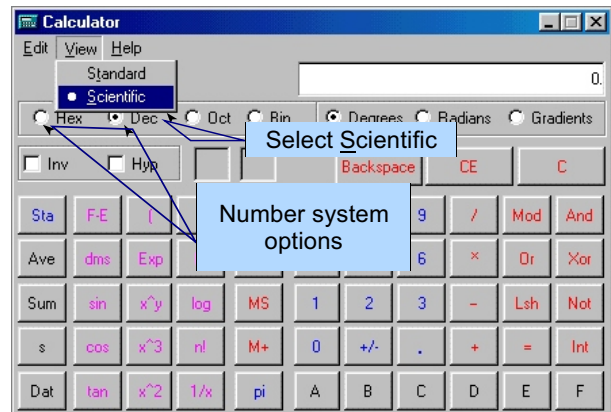


Figure 12

**Step 3:** Click the Dec(imal) option.

**Step 4:** Type:

14

**Step 5:** To convert this decimal to Hexadecimal, click the Hex(adecimal) option.

The result is E. Let's try this again. This time, let's convert a number from Base-10 into Base-16:

**Step 1:** Click the Dec(imal) option.

**Step 2:** Type:

37

**Step 3:** To convert this decimal to Hexadecimal, click the Hex(adecimal) option.

The result is 25.

### LOOKING AHEAD

Obviously, FF is the biggest number we can express with two digits in hexadecimal. What number does this represent in the decimal system?

**Step 1:** Click the Hex(adecimal) option.

**Step 2:** Type:

FF

**Step 3:** Click the Dec(imal) option, and the result is:

255

If anyone ever asks you what your IQ is, say "A0." By their reaction, you may discover theirs.

## HEXADECIMAL COLOR CODES

Hexadecimal color codes begin with a "#" followed by three pairs of characters. Each pair of characters represents a color. The first pair represents the red, the second pair represents the green, and the last pair represents the blue component. The bigger the number, the higher the voltage applied to that color gun of the monitor.

This allows us to create any combination of 255 shades of red, 255 shades of green, and 255 shades of blue—over 16 million distinct colors, expressed in just six characters. Because the human eye can distinguish only about 8 million colors, that ought to be enough.

## NAMED COLORS



In addition to hexadecimal codes, we can specify sixteen colors by name:



## CONFIGURE NETSCAPE®

Before proceeding in this chapter, you must make sure that Netscape® is configured properly:


**Step 1:** From the pull-down menu, select Edit, Prefer-ences.

**Step 2:** In the Category box, click the  (or the ) to the left of Appearance to expand its options.

**Step 3:** Under Appearance, select Colors.

**Step 4:** Depending on the version of Netscape®, either:

- deselect the "Always use my colors . . ." check box; or,
- select the "Always use the colors and background specified by the Web page" option.

**Step 5:** Click  to save your changes and close the dialog.

## BACKGROUND COLOR

To set the background color of a Web page, use the following:

```
<body bgcolor="#rrggbb">
```

where #rrggbb is a hexadecimal color code or a color keyword. For example:

```
<body bgcolor="#000000">
```

produces a page with a black background. This is identical to:

```
<body bgcolor="black">
```

## TEXT COLOR

To set the text color, use:

```
<body text="#ffffff">
```

This produces a page with white text. This is identical to:

```
<body text="white">
```

White text on a white background does not aid reading comprehension.

## HYPERLINK COLORS

As you know, hyperlinks are connections to other Web pages. Browsers keep track of links you have followed and those that you have not. By default, plain text is black, an unvisited link is blue, and a visited link is purple. We can change those colors. A link's status is color-coded:

- An unvisited link has the *link* color.
- As you click a link, it takes on the *alink* (active link) color.
- After you have visited its target, a link assumes the *vlink* (visited link) color.

### UNVISITED LINK

To set the unvisited links color, use:

```
<body link="#00ff00">
```

The unvisited links on this page will be green. This is identical to:

```
<body link="green">
```

### ACTIVE LINK

To set the active link color, use:

```
<body alink="#DA31F9">
```

This produces a page with purple links (lots of red and blue, and just a little green).

### VISITED LINK

To set the active link color, use:

```
<body vlink="#ff0000">
```

This produces a page whose visited links are red. This is identical to:

```
<body vlink="red">
```

## BODY TAG & FONTS

As you can see, the *bgcolor*, *text*, *link*, *vlink* and *alink* are properties of the `<body>` tag. There is only one `<body>` tag, and it belongs after the `</head>`.

Novices often try to affect the color of individual pieces of text by inserting another `<body>` tag and altering its text property. Wrong! Do *not* use more than one `<body>`, or unpleasant effects will ensue. Instead, use the *color* property of the `<font>` tag:

```
<html>
<head><title>My Page</title></head>
<body bgcolor=#ffffff text="blue">
  <h1>Hello!</h1>
  <font color="green">Welcome</font>
  to <font color="red">colors</font>!
</body></html>
```

The results will be similar to Figure 15:



Figure 15

## TILED BACKGROUND

Tiling is a procedure that repeats a very small picture again and again, until it covers your entire screen, giving the illusion of a single, large graphic background.

The code for a tiled background is:

```
<body background="x.gif">
```

where *x.gif* is the name of the image to be tiled.

### USING TILES & BGCOLOR

Obviously, background colors and tiled backgrounds are mutually exclusive. You should always use a *bgcolor* when you use *background*, as in:

```
<body bgcolor="#rrggbb" background="x.gif">
```

There are two reasons for this. First, the background color will fill instantly, and, although it's fast, it still takes a while to load a tiled background image. If you can match the *bgcolor* to the dominant color of the background image, you will give the illusion that your page is loading faster and smoother.

Second, if a user has turned off graphics in his or her browser, background images won't appear, and the browser will display the bgcolor, keeping your overall color scheme intact.

## DITHERING

Different browsers display colors in different ways. If a browser cannot display a color accurately, it may display two or three colors that, when seen from a distance, blend to form the color you want.

As you can see in Figure 16, dithering can create moiré patterns or a grainy look. Dithering makes text difficult to read. The extent of the deleterious effects will depend on the size of the monitor, its resolution, the colors available to the video driver, and the browser of the user.

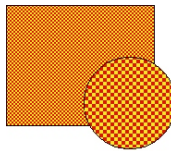


Figure 16

## BROWSER-SAFE COLORS

Of the rainbow of color possibilities, only 216 are accurately displayed by Macintosh® and PC browsers without dithering. These are called “browser-safe colors,” and they are reproduced in Appendix E. When you examine them, you will discover that their red, green, and blue components are multiples of 33—permutations of 00, 33, 66, 99, aa, cc, and ff.

## COLOR ON THE WEB

Here are some Web sites with more information about using color:

### Background Color Information

<http://union.ncsa.uiuc.edu/HyperNews/get/www/html/6/1.html>

### Color Matters

<http://www.colormatters.com/entercolormatters.html>

### ColorMix

<http://www.colormix.com/>

### ColorCenter

<http://www.hidaho.com/colorcenter/cc.html>

### HTML ColorEditor for CGI

<http://www.bbsinc.com/bbs-cgi-bin/colorEditor.cgi>

### A List Apart

<http://www.alistapart.com/stories/color>

### TextColorizer

<http://www.chamisplace.com/tc/>

## DALTONISM

Daltonism, or color blindness, may affect as much as 10 percent of the population. Although monochromatism is rare, partial color blindness, dichromatism, affects 7 percent of men and 1 percent of women. Dichromatism usually manifests itself in the inability to distinguish between shades of red and shades of green, or, more rarely, between blues and yellows.

Keep this in mind when you are designing color schemes. Of particular concern are pastel pairs, such as light red text on a light green background. Here are some Web sites that discuss Daltonism in greater detail:

### About Colour Blindness

<http://www.delamare.unr.edu/cb/>

### Color Vision

<http://open.dtcc.cc.nc.us/opticianry/oph121/lesn16.htm>

### Color Vision FAQ

<http://www.cis.rit.edu/mcsl/faq/faq1.shtml>

### Vischeck

<http://www.vischeck.com/>

## CHAPTER 5: LINKS & GRAPHICS

“If you publish simple text, your pages are likely to be boring and seldom visited. If you fill them with beautiful multi-megabyte images, your pages are likely to load slowly and be seldom visited.”

– Bob Metcalfe

In any Windows® application you get Help by pressing **F1**. A list of topics will appear, with key words underlined and highlighted in a contrasting color. This is called “hypertext.” Click a key word, and you automatically call up a document relevant to that word.

### HYPertext

Q: Where is this help file? On your hard disk? On the network? What’s its name?

A: I don’t know and I don’t care. It just works, and that’s all that matters.

That’s the idea behind hypertext. On the Internet, hypertext links, or “hyperlinks,” retrieve documents that may be in your own directory, in a folder on your server, or on a server 6,000 miles away. Where are they? What are their names? The user doesn’t know or care. It just works.

### CREATING HYPERLINKS

Hyperlinks can link to a document on another server, another local file, or even to a specific location within an HTML document. A link from your HTML document to another Web site is called an “external link.” A link from your HTML document to another document in your Web site is called an “internal link.”

#### LINK EXTERNALLY

The location of a document on the Web is specified by its Uniform Resource Locator (URL, pronounced “you-are-ell”). You are familiar with URLs, because you probably type them in your browser’s location bar to navigate. You need not type the “http://” prefix when you enter a URL in a browser, but it must be always be included when you create an external link in HTML. The syntax for a simple external link is:

```
<a href="URL">Caption</a>
```

where *URL* is the full address of the other Web site. The only thing you will see in your document is the word *Caption*, which normally will be underlined and blue.

When you test this document in your browser, click [Caption](#), and your browser will open *URL*.

Again, when you specify a URL in HTML, you must type its full address, including the “http://” prefix. Thus:

```
<a href="waynespies.com">Spies</a>
```

is incorrect. It must be written:

```
<a href="http://waynespies.com">Spies</a>
```

#### LINK INTERNALLY

In addition to an external link to someone else’s site, you can link to other documents in your own site. The syntax for simple internal links is:

```
<a href="doc.htm">Caption</a>
```

where *doc.htm* is the name of the second document, the one to which you are linking. This will work as long as the second document is in the same directory as the first.

Later, in Chapter 12, you will learn how to specify the path wherever it is located, but, for now, keep all documents and images in the same folder.

#### LINK WITH ANCHORS

We can use internal hyperlinks to link to a specific spot in a document. This spot is called the “target,” and the tag that marks it is called the “anchor.”

Using anchors is a two-step process. First, we must create the anchor. The general form of the anchor is:

```
<a name="anchor"></a>
```

where *anchor* is any name that identifies the target.

Next, we must create a link to that anchor. The general form of the link is:

```
<a href="doc.htm#anchor">Caption</a>
```

where *doc.htm* is the name of the file that contains the anchor, and *anchor* is the name of the anchor created in Step 1. A “ # ” must separate the file name and target.

### LINK & ANCHOR EXAMPLE

Let's create a link-and-anchor example:

**Step 1:** Create a simple HTML document with a screen or two of random text.

**Step 2:** Create the anchor for a target in its body:

```
<a name="Bio"></a>
```

**Step 3:** Save the document as:

```
doc1.htm
```

**Step 4:** Create a second HTML document.

**Step 5:** Maneuver within the Body section to where you want the link to appear.

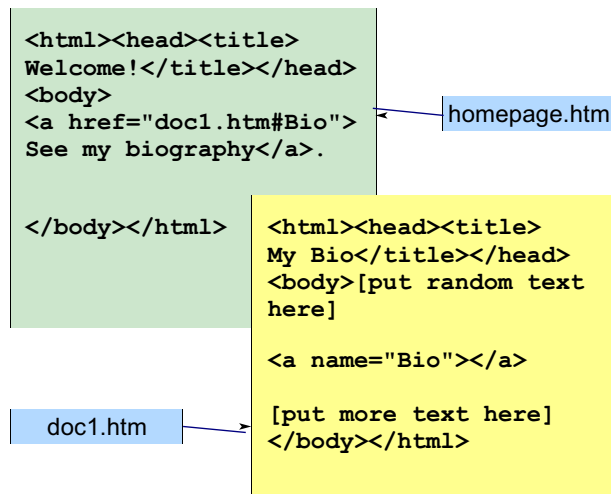
**Step 6:** Type:

```
<a href="doc1.htm#Bio">
See my biography</a>.
```

- As you know, [See my biography](#) is the hyperlink caption.

**Step 7:** Save this second document as:

```
homepage.htm
```



**Step 8:** Open `homepage.htm` in your browser.

When you click [See my biography](#), `doc1.htm` should open exactly at the point you placed the anchor. (If you place the anchor close to the bottom, it will open so that the bottom of the document appears at the bottom of the browser.)

Anchors can be used to create a Table of Contents. Place an anchor at the start of each chapter. Then, create a link to each of those targets in a Table of Contents. When the user clicks the hyperlink in the Table of Contents, the browser will jump down to its corresponding anchor.

Here's a tip for creating anchors. Place the anchor one line above the text you want to display. The extra space will make it easier to read the text.

If the target is in the same document as the link, you may omit the document name in the tag. Thus:

```
<a href="mydoc.htm#bottom">
```

is identical to:

```
<a href="#bottom">
```

if the link and the target are in the same document.

### CASE AWARE

A foolish consistency may be the hobgoblin of little minds, but it is essential when it comes to anchors. Links and their targets may be in UPPER CASE, lowercase, or MiXeD cAsE, *but* the case of the target and its anchor must be identical.

### MAILTO LINKS

When you click a `mailto` link, your e-mail client opens, pre-addressed to the source. The general form of the `mailto` link is:

```
<a href="mailto:x@y.z?subject=s">
```

where:

- `x@y.z` is the e-Mail address
- `s` is the text that you want to appear in the subject line

You must separate the e-Mail address from the word “subject” with a question mark (?). For example:

```
Send me your comments on this book via <a
href="mailto:spies@aya.yale.edu?subject=
Your HTML book">e-Mail</a>.
```

will appear as:

Send me your comments on this book via [e-Mail](#).

When someone clicks [e-Mail](#), his or her browser will open its e-Mail client, pre-addressed to `spies@aya.yale.edu`, with “Your HTML book” in the subject line, like this:

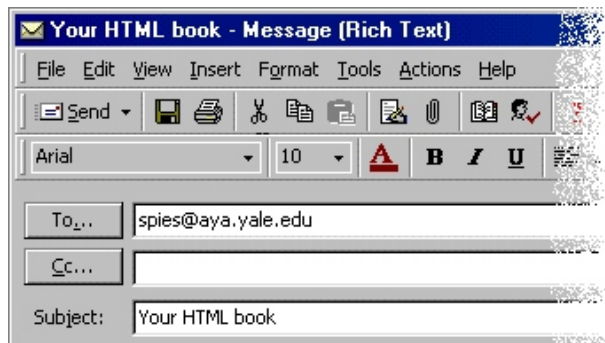


Figure 17

## DOWNLOADING

You can let users download files from your site with a simple link. Use the general form:

```
<a href="xyz.ext">Download XYZ</a>
```

where `xyz.ext` is the name of a program to be downloaded. Now, here is the catch. If the `.ext` extension is recognized by your browser, it will try to display it. If your browser encounters a link to a file with an extension that it does not understand, it will offer to download the file for you. Therefore, you can't use this method to download a file that ends in `.GIF`, `.JPG`, `.HTM`, and so on.

The solution to this problem is to compress the files to be downloaded into a single `.ZIP` file or a self-extracting compressed file, which has an `.EXE` extension.

## SCREEN TIPS

You can use a hyperlink's title parameter to create Screen Tips when the page is viewed in Internet Explorer 4+. Screen tips are those little yellow boxes that pop up as your mouse hovers over a hyperlink. It appears in the general form:

```
<a href="x.htm" title="text">Caption</a>
```

## AESTHETICS

It is considered good form to exclude punctuation marks from links. For example, this:

```
Click <a href="http://xyz.com">here</a>.
```

is preferred to:

```
Click <a href="http://xyz.com">here.</a>
```

Here's an obvious exception:

```
<a href="http://yahoo.com">Yahoo!</a>
```

As awkward as it seems, the exclamation point is part of Yahoo!'s name.

## HOW TO ADD IMAGES

Q.: How do you add images to an HTML document?

A.: *You don't!*

HTML documents are composed of plain, ASCII text. They *do not* and *cannot* include embedded objects or graphics. Instead, they link to an image source. As illustrated in Figure 18, the browser reads the HTML document containing the link to the image source:

```

```

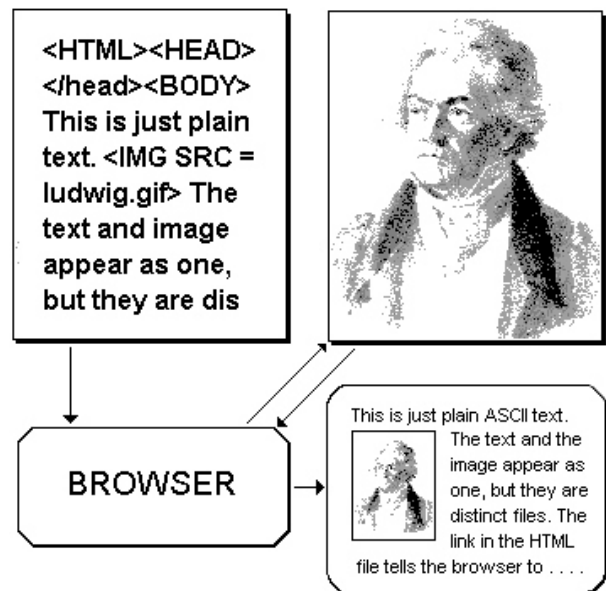


Figure 18

The browser then finds `ludwig.gif` and merges the two documents on the screen. The HTML command for linking to an image source takes this form:

```

```

where `x.gif` is the name of the graphic file. Notice that the `<img>` tag is empty, with no corresponding `</img>`. To comply with the XHTML standard, we end this tag with a space, a slash, and a bracket.

### CASE AWARE, AGAIN

As with links and anchors, a reference to an image—or, indeed, any file—is case sensitive. Links and their images may be in UPPER CASE, lowercase, or MiXeD cAsE, *but* the case of the link and its image’s file name must be identical on Unix servers. This is a terrible trap for the unwary, as this will not effect you when you test your document locally—because Windows® doesn’t care. It’s only when you load your document to a Unix server when the problem occurs, and then it is a doozy to straighten out. Spare yourself the ordeal; match the case of the link and the file name of the image from the start.

### PARAMETERS

The `<img>` tag is an empty tag; it has no turn-off. Optional parameters include:

- `alt="text"`
- `align="top", "center", "bottom", "right", or "left"`
- `width="w"` (in pixels)
- `height="h"` (in pixels)
- `border="b"` (in pixels)
- `hspace="x"` (in pixels)
- `vspace="y"` (in pixels)

Let’s examine these parameters.

### ALTERNATE TEXT

Some visitors to your site will be using a non-graphical browser or will turn off the graphics auto-load feature. To show them know what they’re missing, use the alt parameter:

```

```






where `x` is the name of the graphics file and `text` is the alternate text.

If the user *cannot* receive graphics, the `alt` parameter provides alternative text that appears in lieu of the picture. If the user *can* receive graphics (and is using a newer browser), the alt parameter generates ScreenTips—little yellow boxes—when the mouse passes over the image. This is similar to the ScreenTips generated by the hyperlink title attribute that we saw on page 27.

### ALIGNMENT

The `align` parameter is crude, aligning the image with respect to the adjacent text. Usually we can achieve much finer control over image alignment through the use of tables, which we will investigate in Chapter 8.

---

<code>&lt;img src="fdr.jpg" align="top" /&gt;</code>		Franklin Roosevelt
<code>&lt;img src="jc.jpg" align="center" /&gt;</code>		Jimmy Carter
<code>&lt;img src="b.jpg" align="bottom" /&gt;</code>		George Bush I
<code>&lt;img src="bc.jpg" align="left" /&gt;</code>		President William Jefferson Clinton’s administration was marred by unending personal scandals—both real and imagined—amid an era of unparalleled prosperity.
<code>&lt;img src="rr.jpg" align="right" /&gt;</code>		Ronald Reagan’s adoption of “trickle-down economics” led to the greatest deficit in world history and the destruction of the “Evil Empire”—Soviet communism.

---

### ADD WHITE SPACE

To add space around the image, use the `hspace` (horizontal space), `vspace` (vertical space), and `border` attributes:

```

```

Again, the complements `h`, `v`, and `b` are in pixels.

### CHANGE IMAGE SIZE

You can change the size of a displayed image with the `width` and `height` attributes:

```

```

where:

- *x* is the desired width in pixels; and,
- *y* is the desired height in pixels.

For example:

```

```

will display the file “*g.jpg*” in a rectangle 8 pixels wide and 60 pixels high—regardless of the original size of the picture.

If you specify just one attribute, the other will automatically compensate to preserve the aspect ratio. This is the most common approach.

## FASTER GRAPHICS

The single most common complaint about Web sites is slowly loading pages. The biggest cause of slowly loading pages is the size and nature of graphic images.

### FILE SIZE

Graphics files are huge compared to ASCII text files. As a result, reducing the size of graphics files in a graphics editor can dramatically speed up downloads.

Resizing an image with the *width* or *height* attributes doesn't decrease the file size, but merely reduces the size of its appearance—the “window” through which it is viewed.

You might think that using very tiny files and then blowing them up to a much larger size with *width* or *height* would be the answer. This *would* speed downloads, but the resulting image would not look very good. The *width* and *height* attributes cannot restore information that has been stripped from the file! As a result, they are best used for modifications of less than 10 percent or so.

To maintain the integrity of your image, crop it or resize it as necessary to make it as close the desired size as possible:

- making it too small and resizing it will distort the image;
- making it too large will slow your download.

### IMAGE RESIZING

When a Web page is downloaded by the browser, copies of it and its graphics are saved in your machine's disk cache. That way, once you have visited a page, you can return to it quickly because your browser can reload the image from your cache much faster than downloading it again from the net.

You can use this fact to make your Web pages load faster. For example, it's common for a company's logo to appear large on its homepage, with smaller logos on subsequent pages. Instead of downloading two logos, one large and one small, download only one. Then refer to it as the image source on each page it appears, using the *height* and *width* attributes to create the large one and the small ones.

### IMAGE PRELOADING

As mentioned earlier, your browser maintains a cache of documents and images on your hard drive. When you specify a URL, it checks to see if the document or its graphics are already stored in the cache. Loading data from a cache is obviously much faster than downloading it from the Web.

You can take advantage of this to make your pages load faster. In brief, the strategy is to put images from a graphics-intensive page onto an earlier page that loads quickly. When a browser loads that first page, it will load images and cache them while the visitor is reading the text. Subsequent pages will find that their images are already in the cache, from where they will load quickly.

The trick is to place `<img>` tags at the bottom of the first page. You can make them nearly invisible by setting `width="1" height="1"`. For example,

```

```

loads the entire picture, “*x.jpg*” into the cache, but the `width="1" height="1"` argument forces the browser to display it in a single pixel.

Obviously, this will make your top level page longer to load, but if, as suggested, you place the tiny images at the bottom of the page, following text, the text will load quickly, giving the illusion that the page has loaded.

## CLICKABLE THUMBNAILS

You will remember that a normal link looks like this:

```
<a href="URL">Caption</a>
```

When you click “Caption,” the browser will load the specified URL. You can extend this concept to create clickable images by replacing “Caption” with the image source:

```
<a href="URL"></a>
```

where *pic.gif* is a graphic in your root directory. You can use the *height* and *width* attributes to resize the image as a thumbnail.

With a clickable icon, you can click anywhere on the image, and your browser will whisk you to the specified URL.

### RESOLUTION

You often hear of resolution when working with picture formats and digital cameras. Resolution is the amount of data a picture packs into an individual image. A digital image consists of patterns of dots. Each dot is called a picture element, or pixel. The number of pixels determines the quality of the picture. As a general rule, the higher the resolution, the higher the cost of the camera.

Generally, if you are printing images from a digital camera, the higher the resolution, the better. This isn't true when working with Web graphics. Most monitors display a maximum resolution of 72 dots per inch. Digitizing an image, with either a scanner or a digital camera at a higher resolution increases the size of the file significantly, without noticeably improving its clarity on a monitor.

### MEGAPIXELS

Digital cameras are rated in terms of megapixels. A megapixel is roughly a million pixels. If we do the math, a  $640 \times 480$  image produces 307,200 pixels, which we semantically round to .3 megapixels. Likewise a  $2,048 \times 1,536$  image produces an image of 3,145,728 pixels, which, again, we round semantically to 3 megapixels.

### ON THE SCREEN

Because a traditional monitor displays 72 pixels per inch, a .5 megapixel image (that is, an  $800 \times 600$  image) will measure  $\frac{800}{72} \times \frac{600}{72}$ , or  $11.11" \times 8.33"$ . In other words, a .5 megapixel image is probably adequate for images that you e-mail to others, place in a database, or load to a Web site—wherever it will be displayed on a computer monitor.

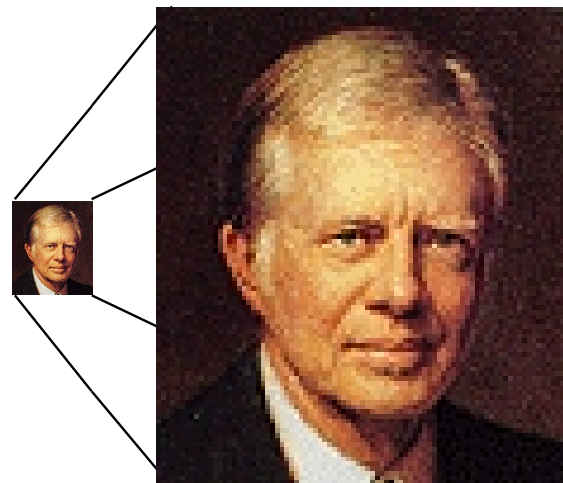
### IN A PRINT

If your goal is to print the image, things are a little different. Because a color printer typically has a 300 dot per inch resolution, that same .5 megapixel image will produce a print that is  $\frac{800}{300} \times \frac{600}{300}$ , or  $2.67" \times 2"$ —much smaller.

It follows that large format, high-quality photos require more pixels. For example, a  $2,832 \times 2,128$  (= 6,026,496, or “6 megapixel”) resolution produces a print that is  $\frac{2832}{300} \times \frac{2128}{300}$ , or  $9.44" \times 7.09"$ .

RESOLUTION	PIXELS	MEGAPIXELS	PRINT SIZE
$640 \times 480$	307,200	0.3	$2.1 \times 1.6$
$800 \times 600$	480,000	0.5	$2.7 \times 2$
$1,280 \times 1,024$	1,310,720	1.3	$4.3 \times 3.4$
$1,600 \times 1,200$	1,920,000	1.9	$5.3 \times 4$
$2,048 \times 1,536$	3,145,728	3.1	$6.8 \times 5.1$
$2,400 \times 1,600$	3,840,000	3.8	$8 \times 5.3$
$2,560 \times 1,920$	4,915,200	5	$8.5 \times 6.4$
$2,832 \times 2,128$	6,026,496	6	$9.4 \times 7.1$

If you tried to take a .5 megapixel image and blow it up electronically, the results will be grainy:



As you can see, blowing up an image digitally has its limitations. Now, some cameras offer both optical zoom and digital zoom. Optical zooming magnifies the image as it is recorded. As a result, optical zoom is an important camera feature. Digital zoom—to the extent that it is used at all—is better left to an image editor on your PC.

## GRAPHIC FORMATS

Even small images create very large files. Very large files obviously take a long time to download. To combat this problem, graphics on the Internet use data compression. The standard compression formats are **.GIF** and **.JPG**.

Both formats are bitmap images, a series of bits, which represent each pixel of the image. Bitmapped graphics cannot be enlarged without their edges becoming jagged in appearance, the so-called “jaggies.” To get an undistorted large image, you need to start with a large image. Even with compression, large images mean large files, which are slow to download. This is just one limitation of these formats.

### .GIF FILES

The Graphics Interchange Format (**.GIF**) was developed by CompuServe® as a way to supply cross-platform, compressed format graphics across its network. CompuServe® licensed the Lempel-Ziv-Welch (LZW) compression format, on which **.GIF**, is based, from Unisys Corp.

In this approach, each pixel’s RGB components are reduced to an index to a 256-color palette. It achieves compression by storing an index to the palette for each pixel instead of the lengthier RGB definition. As a result, the **.GIF** format can display only 256 colors. It compensates for this limitation by dithering images. Dithering is a process that simulates colors beyond **.GIF**’s color palette by tricking the eye.

### INTERLACING

The **.GIF** format also supports interlacing. Interlaced **.GIFs** appear first with low resolution and then improve in resolution until the entire image has arrived, allowing the viewer to get a quick idea of what the picture will look like while waiting for the rest. This creates the illusion that **.GIF** images load faster. Interlaced **.GIFs** are slightly larger than non-interlaced images.

Interlaced **.GIFs** should not be used for background images, because background images will not appear until fully loaded, so interlacing would serve no purpose. Paradoxically, an interlaced image used as a tiled background actually takes longer to appear.

Interlacing is also inappropriate for animated **.GIFs**. With interlacing, an animated **.GIF** appears in low resolu-

tion, and the resulting fuzzy image greatly degrades the animation effect.

### TRANSPARENT .GIFs

Graphics on your page are rectangular. The background of the image may or may not be the same as the page. One variant of **.GIF**, **.GIF89a**—the “89” refers to the year it came out—can be used to make these backgrounds blend seamlessly together. You can specify any one of **.GIF**’s 256 colors, known as the background color, to be transparent, through which the background of the Web page will appear. Such images appear to float upon the page, instead of sitting in a rectangular box, as seen in Figure 26.

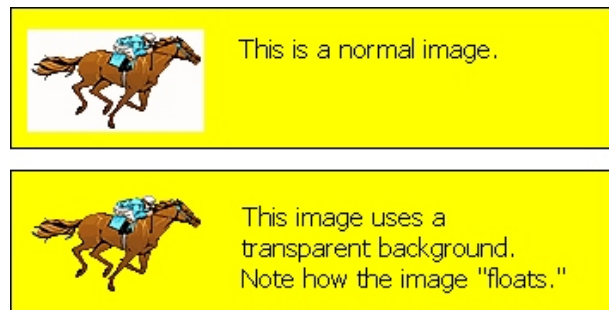


Figure 26

### .JPG FILES

The JPEG (pronounced “JAY-peg”) format, developed by the Joint Photographic Experts Group, is often used for photographic images. These files may have a **.JPG** or **.JPEG** extension. This format supports 24-bit color, allowing it to display millions of colors. As a result, scanning photographs as **.JPG**’s yields superior results to those scanned as 256-color **.GIFs**. In addition, **.JPG** features greater compression than **.GIF** files, enabling quicker downloading and smaller storage requirements.

Still, **.JPG** is not without its disadvantages. Its lossy compression, Discrete Cosine Transformation (DCT), strips the image of some color information. This lossy compression sacrifices a small amount of quality for download speed. Unlike **.GIF** files, **.JPG** does not support interlacing. Instead of appearing all at once, **.JPG** graphics are painted to the screen from the top row to the bottom row. Also, **.JPG** does not support animated graphics or transparent backgrounds.

To address these weaknesses, a new standard, also developed by JPEG, will allow both lossy and lossless compression, through a technology called “wavelet compression.” This approach will allow browsers to download more or less of the wavelet stream, depending on the available bandwidth of your Internet connection. It remains to be seen if this format will become a standard.

## .PNG FILES

In 1995, CompuServe® unexpectedly demanded license fees for the use of the .GIF format. In response, Portable Network Graphics (PNG) was developed in 1996 to provide the benefits of .GIF without the licensing and royalty headaches. PNG, pronounced “ping,” offers several important improvements over .GIF and .JPG:

- 48-bit True Color support. A .PNG image accurately displays and prints 48-bit color across platforms.
- platform-independent gamma correction. Unlike current formats, .PNG images display at the same brightness, regardless of the platform.
- Two-dimensional interlacing. A .PNG image can support more than 65,000 levels of pixel transparency, allowing Web page authors to add anti-aliased type, drop shadows and graduated color fills to their Web pages.

Although .PNG doesn’t yet support animated images, such as animated .GIFs, an extension, Multiple-image Network Graphics (MNG), promises animation features such as frames with multiple images and variable time between frames. MNG reportedly will allow special effects currently unavailable from any format.

The .PNG format also features lossless data compression, called zlib. Lossless compression results in better Web images, and printouts. This compression is not derived from LZW, and is, therefore, free from licensing fees.

Despite its technical superiority, .PNG has made few inroads on the Internet. Although the major browsers support at least some of the .PNG format, their failure to support its more sophisticated features has made it somewhat of a white elephant. Finally, the general availability of fast Internet connections has also reduced the pressure on developers to switch.

One area of continuing confusion is the licensing standards of both .GIF and, more recently, .JPG. Although CompuServe® seemed to lose interest in such lawsuits

following its acquisition by America Online® (and its subsequent Time-Warner amalgam), Unisys stepped up its efforts to collect royalties on .GIF’s underlying technology.

Although .JPG was thought to be in the public domain, Forgent has other ideas. Some experts credit Wen-Hsiung Chen with creating and patenting discrete cosine transform, which underlies .JPG technology. When Chen sold his firm to Forgent, Chen’s intellectual property went with it, and, beginning in 2002, Forgent demanded royalties for .JPG licenses.

## VECTOR-BASED GRAPHICS

These bit-mapped formats divide the picture into tiny dots, each of an individual color. As a result, a bit-mapped image is not scalable; enlarge a .JPG, its details blur and curves become “jaggies.”

The solution would seem to be vector-based graphics. Vector-based graphics define lines and curves mathematically, making them infinitely scalable and much smaller than their bit-mapped counterparts.

This is not a new technology. The problem is that the most popular vector-based formats are proprietary. Until standards evolve and a non-proprietary format is adopted by the major browsers, this superior technology, like .PNG, will sit on the back burner.

## BOTTOM LINE

Despite their obvious limitations, .JPG and .GIF formats promise to dominate Web graphics for the foreseeable future—which, I hasten to add, isn’t too far away. Until then, follow these guidelines:

- Use .JPG for high-resolution images with many gradients, such as black-and-white or color photographs.
- Use .GIF for:
  - line drawings;
  - low-resolution images;
  - animated .GIFs; or,
  - when you need transparent backgrounds.
- Regardless of the format, create the image close to the size you want to display.

- Use the *height* and *width* attributes of the `<img>` tag to fine tune the size—say,  $\pm 10$  percent to minimize bit-mapped distortion.
- Don't use interlacing for animated **.GIFs**.

## NOTES




## LAB 2: WORKING WITH GRAPHICS

In this lab, we'll investigate graphics and so-called "decorative fonts." Next, we'll examine the often overlooked Windows® Paint utility and L-View, a terrific shareware package.

### DECORATIVE FONTS

Back on page 14, I pointed out that the most common way to use exotic fonts is to convert formatted text into a graphic. This isn't a great solution, because it makes the page download much more slowly. On the other hand, if the exotic font is not on the user's system, the browser will substitute another font, with unforeseen results. Until we have a better solution, you need this technique in your bag of tricks:

**Step 1:** Open WordPad.

- Click , and select Programs, Accessories, WordPad.


**Step 2:** Type:

**Welcome!**

**Step 3:** Select the word:

**Welcome!**

**Step 4:** From the pull-down menu, select Format, Font.

**Step 5:** Select a font, a font size, and a color, as shown in Figure 27, and click .

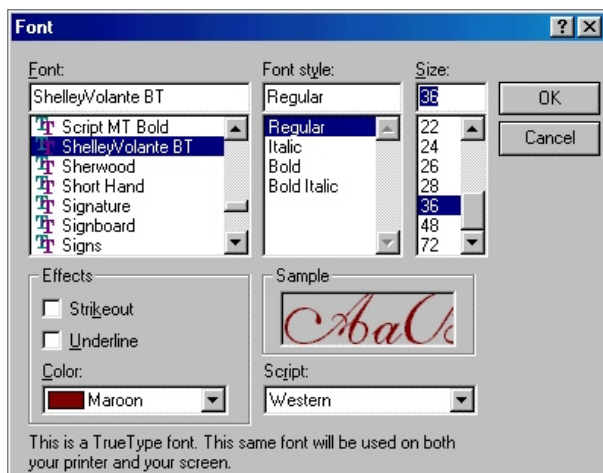


Figure 27

- Your document should resemble Figure 28.






Figure 28



**Step 6:** Press  .

- In Windows®, this copies the image from the active window to Clipboard.

**Step 7:** Open Paint:

- Click , and select Programs, Accessories, Paint.

**Step 8:** Press   to paste the image from Clipboard into the Paint document

**Step 9:** Select the image of the formatted text with one of the selection tools,  .



- Use the free-form selection tool, , to select irregular shapes. It works like the pencil or paintbrush. Click and drag the mouse around the outline of the area to be selected.
- Use the regular selection tool, , to select rectangular shapes. This works like the Rectangle tool. Click one corner of the area to be selected, and drag to the opposing corner, as shown in Figure 32.



Figure 32

**Step 10:** From the pull-down menu, select Edit, Copy To.

**Step 11:** In the resulting Save As dialog, expand the “Save as type” drop-down box and select “256 Color Bitmap (\*.bmp;\*.dib)”.

- The .GIF format can support only 256 colors.

**Step 12:** Name the document and save it in the folder of your choice.

## CONVERTING THE BIT-MAP

Now we have the image, but it has been saved as a bit-mapped graphic, with a .BMP extension. To use it on our Web site, we must convert it to either the .JPG or .GIF format—simply changing the file extension will not work.

*If you use Windows® 95 or NT, skip to the next section on L-View.* If you use Windows® 98, ME or XP, you can change the image’s format from within Paint:

**Step 13:** Back in Paint, press **Ctrl O**, and open the image that you saved in the previous step.

**Step 14:** From the pull-down menu, select File, Save As.

**Step 15:** Navigate to the folder that contains the other files for your Web site.

**Step 16:** In the Save as type drop-down box, shown in Figure 33, select CompuServe GIF (\*.GIF).

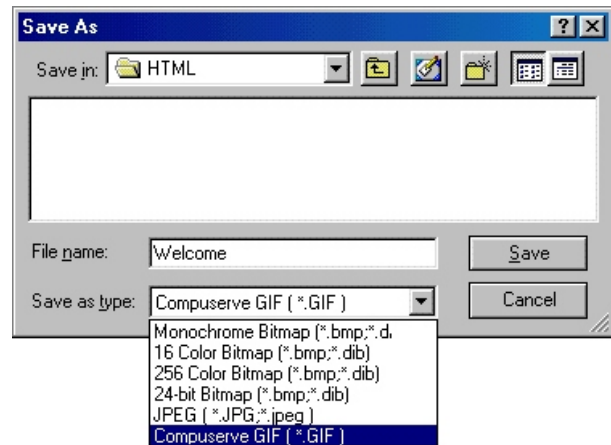


Figure 33

**Step 17:** Click **Save** to save in the .GIF format.

## L-VIEW

There are several wonderful image editors on the market. Unfortunately, they are as high-priced as they are high-powered. If you are a graphics artist, by all means, get one and master it. When we work with images on the Internet, there are a handful of features that we need, whether or not we are graphics artists, such as:

- converting formats;
- cropping images;
- resizing images; and,
- creating transparent backgrounds.

L-View is available as shareware, and it is terrific for our purposes.

### OPEN AN IMAGE

To open an image in L-View:

**Step 1:** From the pull-down menu, select File, Open.

**Step 2:** Select the format of the image you seek in the Files of type drop-down list.

- L-View supports these formats:
 

➤ BMP	➤ GIF	➤ JPG
➤ JFIF	➤ JPEG	➤ TGA
➤ PPM	➤ PGM	➤ PBM
➤ TIF	➤ PCX	➤ TIFF

- If you are unsure of the type, select All files (\*.\*)

**Step 3:** Select an image.

- It will appear in the Preview window in Color or, if you prefer, in Grayscale (black and white).



Figure 34

**Step 4:** Click **Open**.**ZOOM**

You can zoom in by clicking the image or zoom out by right-clicking the image.

**CONVERT FORMATS**

L-View can convert an image from one format into another. This means you can create a .BMP image in Paint, and then convert it to a .JPG or .GIF, or convert a .JPG or .GIF to a .BMP (or any other format, for that matter):

**Step 1:** From the pull-down menu, select **F**ile, **S**ave **A**s.

**Step 2:** Select the new format in the Save as type drop down list, as shown in Figure 35.




Figure 35

**Step 3:** Click **Save**, and L-View will change the format and change the file name extension.

**CROPPING**

To crop a picture in L-View:

**Step 1:** Click the Crop tool icon, .

**Step 2:** Select the image to be cropped, as in Figure 37.

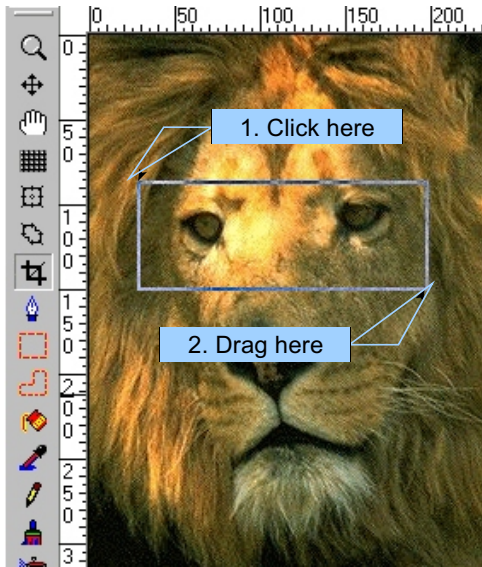


Figure 37

**Step 3:** From the pull-down menu, select Image, Crop, as shown in Figure 38.

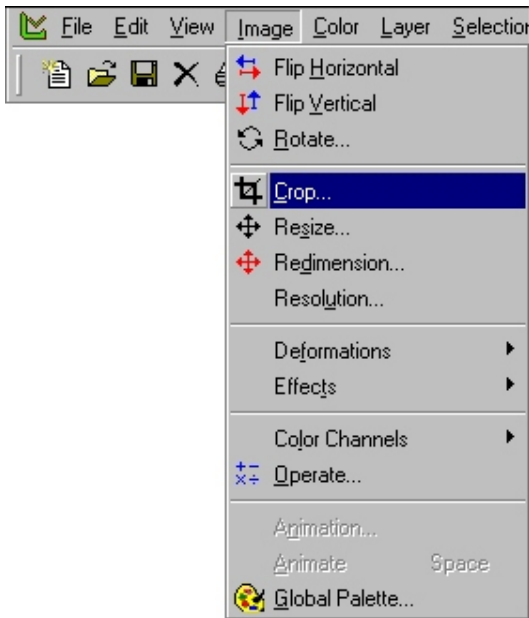


Figure 38

**Step 4:** Click **Crop now**, as shown in Figure 39.

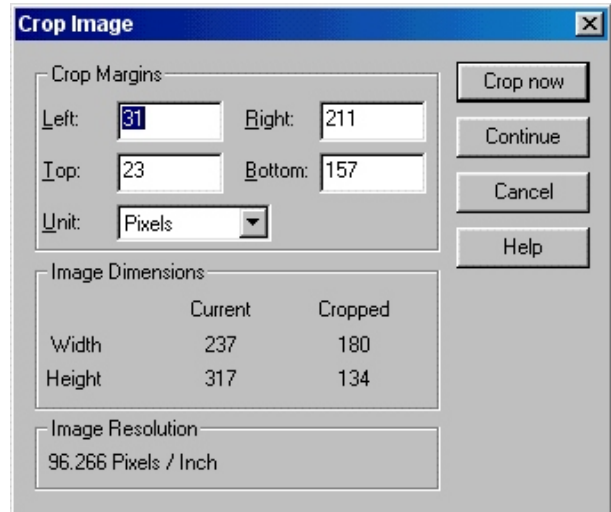


Figure 39

**RESIZE IMAGE**

To make your pages load quicker, reduce the image to the size it will be on your page.

**Step 1:** From the pull-down menu, select Image, Resize.

**Step 2:** In Figure 40, specify the new size.

- If you deselect the Preserve Aspect Ratio check box, you will distort the image.

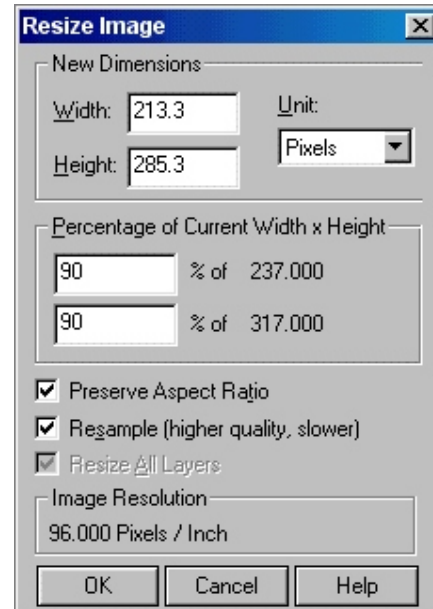



Figure 40

**Step 3:** Click  to apply your changes and close the dialog.

### TRANSPARENT COLORS

As you know, .GIF 89 supports transparent backgrounds. You can use L-View to identify that color:

**Step 1:** From the pull-down menu, select Color, Color Depth.

- If necessary, convert the image to a palette-based format, as shown in Figure 41.

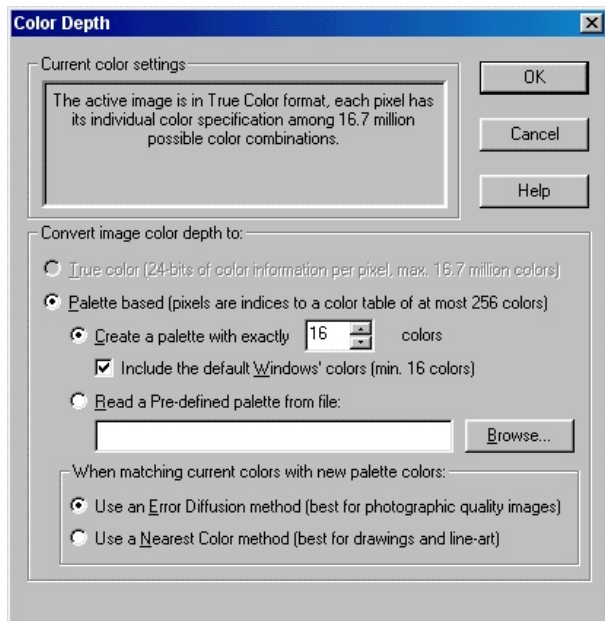


Figure 41

**Step 2:** Click  to close the dialog.

**Step 3:** From the pull-down menu, select Color, Palette Operations, Transparency.

**Step 4:** Select the Layer uses palette transparency check box, as shown in Figure 42.

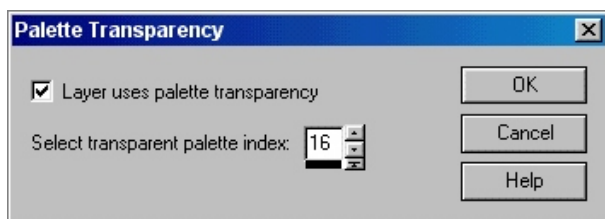


Figure 42

**Step 5:** Either:

- click on the desired background color in the image, or,
- click the slider.

In either case, the image will change, reflecting the absence of the selected color.

**Step 6:** From the pull-down menu, select the File, Save As.

**Step 7:** Select the GIF-89 format.

**Step 8:** Click .

### DOWNLOAD L-VIEW

As you work more with L-View, you will see that we have not even scratched the surface of its features. Still, this overview gives the essentials that we need for Web work.

L-View is shareware. You may download it for a 21-day trial, after which you must remove it from your system or purchase it.

### THE EDITING PROBLEM

Assume that you have this image in your site:

## Brahms: More Than Lullabies



When you look at the Web page in your browser, you decide that it should be cropped, to eliminate Bach. You might take these steps:

**Step 1:** Launch your graphics editor.

**Step 2:** Open the xyz.gif.

**Step 3:** Crop and/or edit the picture.

**Step 4:** Save the results.

**Step 5:** Return to your browser, and reload the page.

Th results are:

## Brahms: More Than Lullabies



Why didn't the change appear? Because of your cache. Before we solve the problem of the cache, let's make sure we understand it first.

### CACHE

If you had to download the same images every time you revisited a page, your Internet experience would be considerably slower than it already is. Your browser solves this problem by storing all downloaded text and images in caches. There are two types of caches:

- memory cache, and,
- disk cache.

The memory cache stores text and graphics from the current session. When you close your browser, the memory cache writes its contents to the disk cache.

When it opens a page with graphics, your browser checks to see if it has that image stored already in the cache. Opening an image from a cache on your PC is obviously much faster than downloading it each time. As a result, increasing your cache size makes your Web experience faster.<sup>1</sup>

### Back to Our Problem

This means is that when you edit the Brahms picture and reload the page, your browser looks to see if the Brahms picture is in the cache. It is, so it reloads it from the cache *instead of from the source*. Of course, the source contains the edited image, not the cache! To solve this, we have two choices; we must either:

- force the browser to do what is known as a “fresh reload,” or,
- turn off your browser's cache.

Either approach will result in the page appearing the way you intended:

## Brahms: More Than Lullabies



### FORCE A FRESH RELOAD

When you press **Ctrl R**, your browser reloads your document from the cache. To force it to download the entire page from the source:



- press **Ctrl Shift R**, or,
- hold down **Shift** as you click **Reload** (Netscape®), or **Refresh** (IE).

This also replaces the old components of the page that are in the cache with fresh ones from the source.

### TURN OFF NETSCAPE® CACHE

By default, the Netscape® memory cache is set at 1,024 KB, and its disk cache is set at 5,000 KB (=5 MB). To turn off its cache, as you test your Web page design:

**Step 1:** From the pull-down menu, select Edit, Preferences.

**Step 2:** Click the  or the  to the left of the Advanced category folder to expand its options.

**Step 3:** Select Cache.



**Step 4:** Under the options for how often the cached document is compared to the document on network, select Every time.

**Step 5:** Click **OK** to save your changes and close the dialog.

### RESET THE NETSCAPE® CACHE


If you use Netscape® to access the Internet, be sure to turn the cache back on when you are finished editing your Web site:

**Step 1:** From the pull-down menu, select Edit, Prefer-  
ences.

**Step 2:** Click the  or the  to the left of the Advanced category folder to expand its options.

**Step 3:** Select Cache.

**Step 4:** Select Once per session.

**Step 5:** Click  to save your changes and close the dialog.

### TURN OFF IE'S CACHE


Each time you open a Web site, IE caches text and graphics to its own cache folder:

**C:\Windows\Temporary Internet Files**

By default, IE's cache is set at 3 percent of your hard disk space. IE caches Web pages until the cache is full, when it automatically deletes the oldest material first.

To turn off IE's cache, as you test your Web page design:

**Step 1:** From the pull-down menu, select Tools, Internet Options.

**Step 2:** On the General tab click .

**Step 3:** To turn off the cache, select Every visit to the page, as shown in Figure 49.

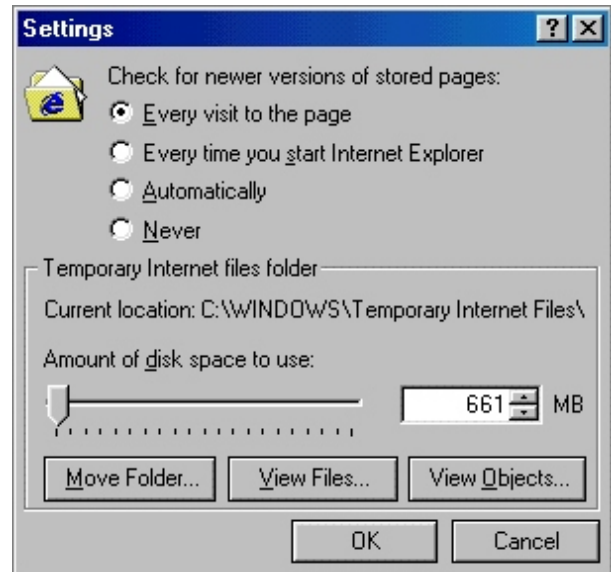




Figure 49

**Step 4:** Click  twice to save your changes and close the dialogs.

### RESET THE IE CACHE

If you use IE to access the Internet, be sure to turn the cache back on when you are finished editing your Web site:

**Step 1:** In Figure 49, select Automatically.

**Step 2:** Click  twice to save your changes and close the dialogs.

### ALIASED TEXT

If you create text with graphics software, you often have the choice of using aliased or anti-aliased fonts:

- Aliased text has sharp edges, which display as “jaggies” on curved letters.
- Anti-aliased text adds light-colored pixels around the edges, to give a smoother appearance.

Anti-aliased type is generally preferred, unless you are using a dark background, which accentuates its light-colored edges, creating a strange halo effect.

## SPACER .GIF

The bigger the image, the bigger the file, and the longer the download. It stands to reason that the smallest image—1 pixel × 1 pixel—would be very small, and it is—under a kilobyte! Because it’s so small to download and because it is sizeable, a one-pixel image is often used to position elements on a page:

**Step 1:** Create a .GIF file measuring 1 pixel × 1 pixel.

**Step 2:** Set that pixel’s color as a transparent background.

**Step 3:** Set the *height* and *width* attributes to size the image in your document.

- Greatly resizing a standard image would distort it, of course, but because it is transparent, resizing our little “spacer” can’t hurt it.
- For example, if you name such an image “spacer.gif,” you might use it to position the word “Hello!” 80 pixels from the left of the screen:

```
Hello!
```

## AN HR SPACING TRICK

You may remember the Horizontal Rule tag, `<hr />`, which we discussed on page 14. We usually use this to create horizontal lines, but we can also use it to control vertical spacing. Let’s say that you want text to appear 20 points below a picture or other text. Instead of using the spacer.gif approach, you might try to create a horizontal line that is one pixel wide, but 20 pixels tall—and the same color as the background:

```
<hr color="white" width="1" align="right"
size="20" />
```

## NOTES