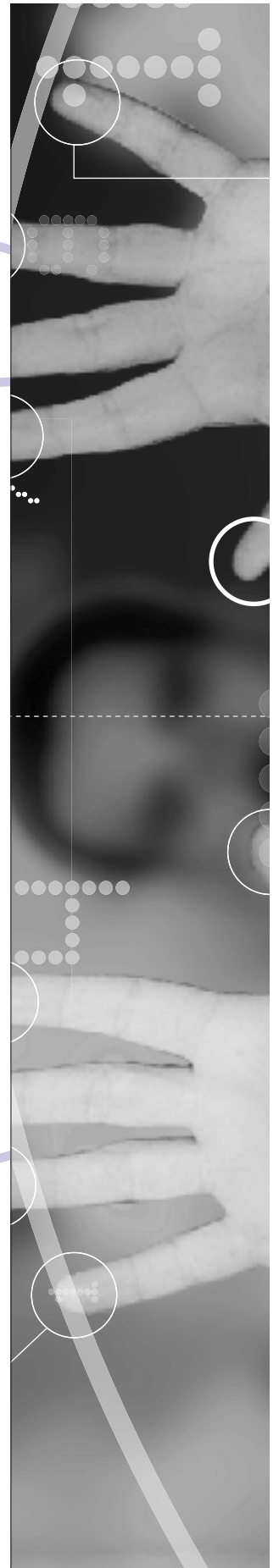


## Part 1: HTML Document Structure

- Task 1: How to Write a Tag
- Task 2: Structuring an HTML Document
- Task 3: Defining Meta Tag Keywords
- Task 4: Defining Meta Tag Descriptions
- Task 5: Defining the Author of a Document Using Meta Tags
- Task 6: Defining Meta Tag Expiration Dates
- Task 7: Refreshing Page Content Using Meta Tags
- Task 8: Defining Meta Tag Robot Values
- Task 9: Controlling the Document Background
- Task 10: Working with Source Code in the Browser



## Task

## 1

## notes

- When HTML was first created, the standard practice was to write tags in uppercase. Over time, this standard changed to lowercase to mimic the syntax of programming languages. Browsers currently treat uppercase and lowercase code identically. However, Extensible Hypertext Markup Language (XHTML), which is destined to replace HTML, is case-sensitive, so XHTML-compliant browsers will see `<P>` and `<p>` as different tags. To make sure your code is always XHTML-compliant, write your code in lowercase.
- The majority of tags in HTML come in pairs: the opening and closing tags. Together, these tags form a container around the page content they define, indicating to your Web browser the beginning and end of a particular element.
- Not all HTML tags have a corresponding closing tag. Some tags only have an opening one. These *empty tags* are used to define elements that don't have logical beginnings and endings. For instance, the line-break tag is written as just `<br />` (there is no closing `</br>` tag).

## How to Write a Tag

**P**rior to computer-assisted publishing, you wrote notes to the manuscript's typesetter directly in the document — hence the phrase *to mark up*. In an electronic text document, like a Web page, you can't scribble in the margins; you need another mechanism. That mechanism is the *tag*. Hypertext Markup Language is based on tags that mark up text-based documents. They instruct Web browsers how to display content. What we'll look at in this task is the basic *syntax* (grammatical rules) for writing HTML tags.

1. To indicate where a given element begins, place the appropriate tag before it. This consists of a certain abbreviation sandwiched by the less-than (`<`) and greater-than (`>`) symbols. For example, to mark up a paragraph, precede the text with the opening-paragraph tag (`<p>`), as shown in Listing 1-1.

```
<p>She stretched herself up on tiptoe, and peeped over the  
edge of the mushroom, and her eyes immediately met those of  
a large blue caterpillar, that was sitting on the top, with  
its arms folded, quietly smoking a long hookah, and taking  
not the smallest notice of her or of anything else.
```

**Listing 1-1:** Placement of the opening-paragraph tag

2. To indicate where an element ends, place the corresponding closing tag at the end. This looks the same as the opening tag, except for the addition of the forward slash, as shown in Listing 1-2.

```
<p>She stretched herself up on tiptoe, and peeped over the  
edge of the mushroom, and her eyes immediately met those of  
a large blue caterpillar , that was sitting on the top, with  
its arms folded, quietly smoking a long hookah, and taking  
not the smallest notice of her or of anything else.</p>
```

**Listing 1-2:** Placement of the closing-paragraph tag

3. When you define a tag's *attributes*, which are its individual properties, enter them inside the opening tag and separate them by spaces. The closing tag doesn't get any attributes. For instance, the attribute for aligning a paragraph is written, simply enough, as `align`. Add it to the opening tag as shown in Listing 1-3.

```
<p align>She stretched herself up on tiptoe, and peeped
over the edge of the mushroom, and her eyes immediately
met those of a large blue caterpillar , that was sitting
on the top, with its arms folded, quietly smoking a long
hookah, and taking not the smallest notice of her or of
anything else.</p>
```

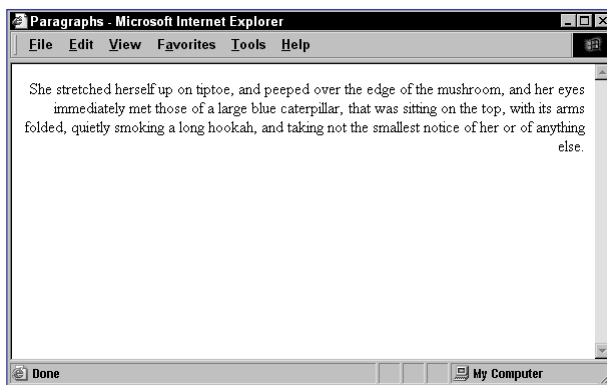
**Listing 1-3:** The opening paragraph tag and its `align` attribute.

4. To set the attribute equal to an appropriate value, define that value by using an equal sign and quotation marks, as shown in Listing 1-4.

```
<p align="right">She stretched herself up on tiptoe,
and peeped over the edge of the mushroom, and her eyes
immediately met those of a large blue caterpillar , that
was sitting on the top, with its arms folded, quietly
smoking a long hookah, and taking not the smallest notice
of her or of anything else.</p>
```

**Listing 1-4:** A properly defined attribute that right-aligns the paragraph text

Figure 1-1 shows how this paragraph appears in the browser.



**Figure 1-1:** The sample paragraph rendered by Internet Explorer

## Task

# 1

### tip

- If what you see when you test your work in a browser doesn't correspond to the code you thought you wrote, chances are you just missed a space between a tag character and its attribute, forgot an equal sign, or omitted a quotation mark.

### cross-reference

- To learn more about XHTML, the next generation of HTML, visit our Web site at [www.wiley.com/compbooks/10simplestepsorless](http://www.wiley.com/compbooks/10simplestepsorless).

## Task

## 2

**notes**

- Indenting the tags for the document title, as we've done in Listing 2-2, has no impact on the way the code is rendered by a browser. However, it greatly improves the readability of your code by others, including yourself.
- The head section defines information about the document that doesn't get displayed in the browser window. You'll learn how to define much of this type of content in Tasks 3-8.

## 4

## Part 1

## Structuring an HTML Document

The simple document template that you are about to build can be used again and again as the starting point for every page you create. All HTML documents share this identical underlying structure — a kind of backbone onto which you build your unique page content. As you learned in the previous task, most HTML tags come in pairs which define the content within them. HTML refers to these as *container tags*. An HTML document's basic structure is really just a series of large containers, inside of which you define the two main sections of your page: the document head and the document body.

1. Open your text editor and begin a new blank document.
2. Type the tag `<html>` at the top of the document. This tag begins the document's primary container. It defines the type of document you're creating: an HTML document.
3. This opening `<html>` tag requires a closing tag, so hit Enter (or Return) twice to move down a few lines and then enter the closing tag, `</html>`. Your document should appear like this:  

```
<html>  
  
</html>
```
4. Place your cursor on the line between the opening and closing tags. Type the tag `<head>`, which defines the head section of the document.
5. Hit Enter (Return) twice and then type `</head>`. Your document should now resemble Listing 2-1.

```
<html>  
<head>  
  
</head>  
</html>
```

**Listing 2-1:** The head section of your HTML document

6. To create the document title, which appears in the title bar of the browser window, enter `<title>` and `</title>` between the head tags of your document, as shown in Listing 2-2. For example, entering `<title>HTML in 10 Simple Steps or Less</title>` produces what you see in Figure 2-1.

```
<html>
<head>

    <title>HTML in 10 Simple Steps or Less</title>

</head>
</html>
```

**Listing 2-2:** Defining the document title



**Figure 2-1:** The document title displayed on the title bar of the browser

7. The last element to add to your document template is the body section. Between the closing `</head>` and the closing `</html>` tags, enter opening and closing body tags, as shown in Listing 2-3.

```
<html>
<head>
    <title> HTML in 10 Simple Steps or Less </title>
</head>

<body>

</body>
</html>
```

**Listing 2-3:** An HTML document with head and body sections defined.

8. Save your document. You can give it a name like `blank.html` and then use it each time you want to start a new document by opening it, making changes, and resaving the file with a different name.

### *cross-reference*

- Many text editors have features that write these initial document tags for you. See Part 12: TextPad; Part 13: Working with BBEdit; and Part 14: Working with HomeSite.

## Task

## 3

## Defining Meta Tag Keywords

A document's head section often contains descriptive information about the document, referred to as metadata. Using the <meta> tag and its various attributes, you can define such document properties as the author, the expiration date, document key words, and descriptions. When search engines that support metadata read your document, they can use this information to index it in order to return your page when someone does a search on subjects matching the keywords you have defined.

1. In the head section of your document, below the document title, enter the <meta> tag, as shown in Listing 3-1.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta>
</head>
```

**Listing 3-1:** Inserting the <meta> tag

2. Add the name attribute to the <meta> tag and set it equal to "keywords", as shown in Listing 3-2.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords">

</head>

<body>

</body>
</html>
```

**Listing 3-2:** The name attribute set equal to "keywords"

### caution

- If you repeat yourself by using the same or similar keywords, for example "stamp, stamps, stamp collecting," some search engines may view this as a spamming tactic and rank your page low, or not at all.

3. Insert a space and add the `content` attribute, as shown in Listing 3-3.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content>

</head>

<body>

</body>
</html>
```

**Listing 3-3:** Adding the `content` attribute

4. Set the `content` attribute equal to a comma-separated list of keywords pertinent to your page's subject matter, as shown in Listing 3-4.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
  Language, 10 Simple Steps or Less">

</head>

<body>

</body>
</html>
```

**Listing 3-4:** Defining keywords for the `<meta>` tag

5. Because the `<meta>` tag is an empty tag, you want to make sure that the code is both XHTML-compliant and still recognizable to browsers that don't yet support XHTML. To do that, conclude the tag with a forward slash (`/`), placing a space between the last entry in the tag and the forward slash:

```
<meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
```

## Task

# 3

### tips

- The object is not to supply every conceivable keyword you can think of but to tailor your keywords to the specific information contained in the document. Keywords can be single words as well as two- or three-word phrases.
- Work your keywords into your document titles and body text. The first word in your document title should be referenced early in your list of keywords, too, so you probably shouldn't start page titles with words like "The." Any keyword that appears in the text of your document shouldn't be repeated more than seven times in that page.

## Task

## 4

**note**

- What you enter for the `name` and `content` attributes defines something called a *property/value pair*. The `name` attribute defines what the property is, and the `content` attribute defines the value of that property.

## 8

## Part 1

## Defining Meta Tag Descriptions

Search engines use the `<meta>` tag's description of the document for indexing and ranking purposes. Some search engines also display the description entries underneath the links on results pages. Because this text is meant for both human and search engine readability, be sure to write it in a way that entices people to click to your site.

1. In the head section of your document, below the document title, insert another `<meta>` tag.
2. Add the `name` attribute to your `<meta>` tag and set it equal to "description", as shown in Listing 4-1.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description">
</head>

<body>

</body>
</html>
```

**Listing 4-1:** Specifying that this `<meta>` tag contains a document description

3. Press the Spacebar and add the `content` attribute, which accepts your description, as shown in Listing 4-2.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content>
</head>

<body>

</body>
</html>
```

**Listing 4-2:** Adding the `content` attribute



4. Set the content attribute equal to a short piece of descriptive text, as shown in Listing 4-3.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder." >
</head>

<body>

</body>
</html>
```

**Listing 4-3:** Completing the property/value pair of a <meta> tag description

5. To make the <meta> tag both XHTML-compliant and still recognizable to browsers that don't yet support XHTML, insert a space and forward slash at the end of the tag, as shown:

```
<meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder"
/>
```

## Task

# 4

### tip

- In search engines that make use of <meta> tags, it is this descriptive text, combined with the text you place between your title tags, that potential site visitors see in their search results. Your primary keyword or keyword phrase for this document should be part of your description text. You don't want to pack the description with keywords, or be heavy-handed with text that reads like a late-night infomercial. Remember that this text is for human consumption; there's a reason why infomercials aren't regarded positively as sources of objective information.

### cross-reference

- You can use <meta> tags to instruct a search engine how or even if you want a document to be read by its search engine-updating robots. See Task 8 for more information.

## Task

## 5

**note**

- If you search the Internet for information concerning metadata, you'll find more than a few sites offering to maintain your <meta> tags and guarantee high search engine rankings — for a small fee. Perhaps they've published a book or online document that contains the “secrets” of preparing <meta> tags. Buyer beware: All this information is publicly available (how do you think they got it?), so you can find it out for yourself. Realize, too, that as of this writing the most popular search engine in use is Google, which *does not* make use of metadata whatsoever.

10

Part 1

## Defining the Author of a Document Using Meta Tags

If you want to put your John Hancock on your document, <meta> tags allow you to do this quite simply. To date, none of the search engines that take advantage of metadata specifically target author information, but supplying it does clearly mark who the content author is and who is responsible for updating the page.

1. Enter a <meta> tag into the head section of your document, setting the name attribute equal to author, as shown in Listing 5-1.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder" />
  <meta name="author">
</head>

<body>

</body>
</html>
```

**Listing 5-1:** Set the name attribute equal to “author”.

2. Follow the name attribute and author value with the content attribute:

```
<meta name="author" content>
```

3. Set the content attribute equal to the name of the author, as seen in Listing 5-2.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder" />
  <meta name="author" content="Robert Fuller and Laurie
Ulrich">
</head>

<body>

</body>
</html>
```

**Listing 5-2:** The content attribute set to the author's name

4. To make the <meta> tag both XHTML-compliant and still recognizable to browsers that don't yet support XHTML, insert a space and forward slash at the end of the tag, as shown:

```
<meta name="author" content="Robert Fuller and Laurie
Ulrich" />
```

## Task

# 5

### tip

- You can also include an e-mail address for the content value.

### cross-reference

- Metadata isn't the only thing that appears in the head section of HTML documents. Cascading Style Sheets and JavaScript code goes there too. To learn more, see Parts 9 and 10.

## Task

## 6

## Defining Meta Tag Expiration Dates

**note**

- Greenwich Mean Time (GMT) format uses the three-character abbreviations for the days of the week (Mon, Tue, Wed, Thu, Fri, Sat, Sun), followed by the day, month, full year, and time in hours: minutes:seconds. Of course, it helps if you know what your local time translates to in GMT. You can find out at [www.greenwichmeantime.com](http://www.greenwichmeantime.com).

The default behavior of most browsers is to *cache* (a fancy word for save) the pages it visits so that if you request the page again, it can pull it quickly from your computer's hard drive instead of pulling it off the Internet, which might take more time. Although most browsers allow users to control this behavior, as a developer you can specify the date on which the current content of your page expires. From that point on, browsers visiting the site will have to connect to your server to get the latest version. You can also instruct browsers not to cache your Web pages at all.

1. Insert a <meta> tag in the head section, setting the name attribute equal to `expires`, as shown in Listing 6-1.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder" />
  <meta name="author" content="Robert Fuller and Laurie
Ulrich" />
  <meta name="expires">
</head>

<body>

</body>
</html>
```

**Listing 6-1:** Setting the name attribute equal to `expires`

2. Insert the content attribute as shown:

```
<meta name="expires" content>
```

3. Set the content attribute equal to the expiration date, in Greenwich Mean Time (GMT), as shown in Listing 6-2.
4. To prevent browsers from caching your documents at all, enter a <meta> tag with the name attribute set equal to `pragma` and the content attribute set equal to `no-cache`, as shown in Listing 6-3.

**caution**

- To get your site listed on a search engine, you must register your site with them. Typically, you submit your site's URL, and at some later point, they scan your site and determine where and how to rank it. Be aware that not every search engine makes use of meta-data. Check with a particular search engine's rules for submitting your site.

## Task 6

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder" />
  <meta name="author" content="Robert Fuller and Laurie
Ulrich" />
  <meta name="expires" content="Mon, 17 February 2003
02:00:00 GMT">
</head>

<body>

</body>
</html>
```

**Listing 6-2:** Expressing the expiration date in GMT

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder" />
  <meta name="author" content="Robert Fuller and Laurie
Ulrich" />
  <meta name="expires" content="Mon, 17 February 2003
02:00:00 GMT">
  <meta name="pragma" content="no-cache">
</head>

<body>

</body>
</html>
```

**Listing 6-3:** Preventing a browser from caching your page with a special <meta> tag

5. To make these <meta> tags both XHTML-compliant and still recognizable to browsers that don't yet support XHTML, insert a space and forward slash at the end of each tag:

```
<meta name="expires" content="Mon, 17 February 2003
02:00:00 GMT" />
  <meta name="pragma" content="no-cache" />
```

*cross-reference*

- See Task 7 to learn how to use meta tags to refresh page content.

## Task

## 7

14

Part 1

## Refreshing Page Content Using Meta Tags

It's possible to modify a browser's behavior using `<meta>` tags. In this task, you're going to generate code that has the same effect as hitting the browser's refresh button. You'll also see how this same code can force the browser to load another document.

### note

- Use the `http-equiv` attribute in place of the `name` attribute when the action being taken retrieves data using the Hypertext Transfer Protocol (`http://`).

- In the head section of your document, below the document title, enter a new `<meta>` tag.
- Add the `http-equiv` attribute and set it equal to `refresh`, as shown in Listing 7-1.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder" />
  <meta name="author" content="Robert Fuller and Laurie
Ulrich" />
  <meta name="expires" content="Mon, 17 February 2003
02:00:00 GMT" />
  <meta http-equiv="refresh">
</head>

<body>

</body>
</html>
```

**Listing 7-1:** Inserting the `http-equiv` attribute

- Follow the `http-equiv` attribute and `refresh` value with the `content` attribute and set it equal to the number of seconds you want the page to remain static before refreshing, as shown in Listing 7-2. In this example, the page will refresh every five seconds.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder" />
  <meta http-equiv="refresh" content="5">
</head>

<body>

</body>
</html>
```

*(continued)*

## Task

## 7

```
<meta name="author" content="Robert Fuller and Laurie
Ulrich" />
<meta name="expires" content="Mon, 17 February 2003
02:00:00 GMT" />
<meta http-equiv="refresh" content="5">
</head>

<body>

</body>
</html>
```

**Listing 7-2:** Setting the number of seconds to wait before a forced refresh

4. To force the browser to load another document after the refresh time elapses, follow the refresh rate value with a semicolon and enter `url=pathname`, where `pathname` equals the file path to a document on your Web server or a complete URL to a document on another site, as shown in Listing 7-3.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder" />
  <meta name="author" content="Robert Fuller and Laurie
Ulrich" />
  <meta name="expires" content="Mon, 17 February 2003
02:00:00 GMT" />
  <meta http-equiv="refresh" content="5;
url=http://www.w3c.org">
</head>

<body>

</body>
</html>
```

**Listing 7-3:** Supplying the URL of another document you want the browser to load after the forced refresh

5. To make your code both XHTML-compliant and still recognizable to browsers that don't yet support XHTML, insert a space and forward slash at the end of the `<meta>` tag:

```
<meta http-equiv="refresh" content="5;
url=http://www.w3c.org" />
```

## Task

## 8

*note*

- The reasons for allowing a robot to crawl your site are obvious. You want the search engine to explore your site fully in order to index everything, baring it all for public consumption. However, there are times when you don't want this to happen; for example, if you have pages you don't want the world knowing about yet, like ones you're in the process of testing, or pages that contain information you only want a limited audience to know about.

## Defining Meta Tag Robot Values

A *robot* is a type of program that search engines use to browse Web site documents and update their databases. Robots that make use of <meta> tag information read the metadata and index it for the search engine. You can control how much or how little of your site a robot reads using the following attributes and values for the <meta> tag.

1. Enter a <meta> tag in the head section of your document, below the document title.
2. Define the name attribute and set it equal to `robots`, as shown in Listing 8-1.

```
<html>
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder" />
  <meta name="author" content="Robert Fuller and Laurie
Ulrich" />
  <meta name="expires" content="Mon, 17 February 2003
02:00:00 GMT" />
  <meta http-equiv="refresh" content="5;
url=http://www.w3c.org" />
  <meta name="robots">
</head>

<body>

</body>
</html>
```

**Listing 8-1:** Setting the name attribute equal to `robots`



**Task 8**

3. To instruct robots to read your entire page and follow all the links within it, follow the name attribute and robots value with the content attribute and set it equal to all, follow, as shown in Listing 8-2.

```
<head>
  <title>HTML in 10 Simple Steps or Less</title>
  <meta name="keywords" content="HTML, Hypertext Markup
Language, 10 Simple Steps or Less" />
  <meta name="description" content="HTML in 10 simple steps
or less. An introductory guide for the beginning coder" />
  <meta name="author" content="Robert Fuller and Laurie
Ulrich" />
  <meta name="expires" content="Mon, 17 February 2003
02:00:00 GMT" />
  <meta http-equiv="refresh" content="5;
url=http://www.w3c.org" />
  <meta name="robots">
<meta name="robots" content="all, follow">
</head>
```

**Listing 8-2:** Code allowing robots to read the entire page and follow all links

4. To instruct robots to read your page, but refrain from following the links within it, set the content attribute equal to all, nofollow:

```
<meta name="robots" content="all, nofollow">
```

5. To prevent robots from reading your page at all, set the content attribute equal to none:

```
<meta name="robots" content="none">
```

6. Insert a space and forward slash at the end of the <meta> tag to make sure your code is both XHTML-compliant and still recognizable to browsers that don't yet support XHTML:

```
<meta name="robots" content="none" />
```

## Task

## 9

**note**

- The default body text color is controlled with the `text` attribute of the `<body>` tag. To learn more, see Task 15.

18

Part 1

## Controlling the Document Background

You can specify a document's background color or background image using two different attributes of the `<body>` tag. Background colors simply fill the entire document. Background images are *tiled* by the browser, meaning they are repeated left to right, top to bottom, filling up the visible space of the browser window.

1. To define a background color for a document, add the `bgcolor` attribute to the `<body>` tag, as shown here:
2. Set the `bgcolor` attribute equal to a hexadecimal color value or predefined color name. Listing 9-1 shows a document with a black background color defined in hexadecimal notation. Figure 9-1 shows the result in a browser.

```
<html>
<head>
  <title>Background Color</title>
</head>

<body bgcolor="#000000" text="white">
<h1>Here we have a black background with white text...</h1>

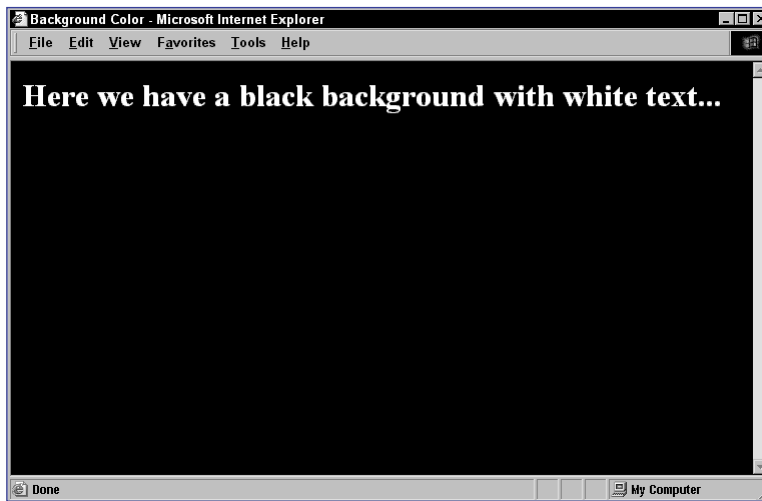
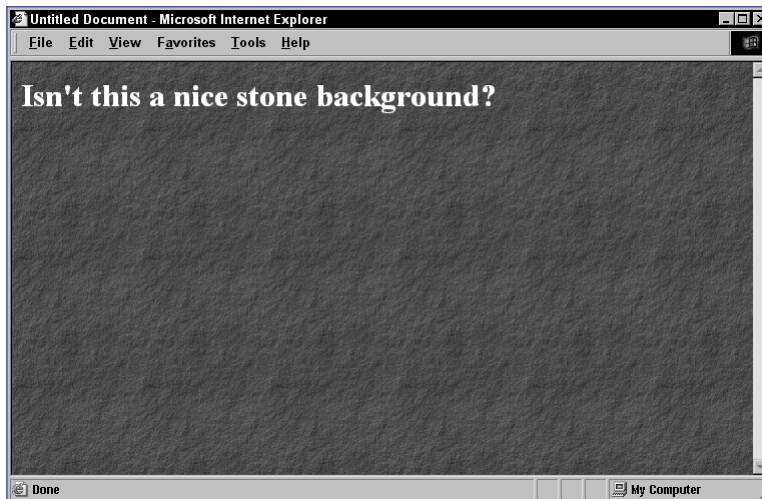
</body>
</html>
```

**Listing 9-1:** Setting the `bgcolor` attribute (and text color)

3. To specify a background image, add the `background` attribute to the `<body>` tag, as shown here:
4. Set the `background` attribute equal to the pathname of the image file on your Web server. Listing 9-2 provides a code sample of a document that makes use of a tiling background texture graphic. Figure 9-2 displays the result in a browser.

```
<html>
<head>
  <title>Background Images</title>
</head>

<body background="images/bg_stone.jpg">
<h1>Isn't this a nice stone background?</h1>
</body>
</html>
```

**Listing 9-2:** The background attribute**Figure 9-1:** White text on a black background**Figure 9-2:** White text over a stone background

## Task 9

### *cross-references*

- To see a hexadecimal color chart and learn more about hexadecimal notation, see our Web site at [www.wiley.com/compbooks/10simplestepsorless](http://www.wiley.com/compbooks/10simplestepsorless).
- Each listing here shows an example of a heading tag. To learn more, see Task 11.

## Task

## 10

## Working with Source Code in the Browser

### notes

- While it's fine to copy source code to examine and learn from, do not plagiarize another developer's HTML. You could run into potential legal issues with the site's owner.
- Because Notepad is a text editor, you can make changes to the code right there. Of course, if you're viewing a Web page on the Internet, changes you make to the code won't have an effect on the actual Web site.

All major Web browsers allow you to view the source code of documents you view — an extremely useful feature. For example, imagine you're surfing the Internet and you come across a page you're really impressed with. To see how it was built, just view the source HTML code. Granted, if this book is your first foray into HTML you may not understand what you're looking at, but you will in time. Each browser has slightly different commands and it supplies slightly different options. Here's how you can view source code using Netscape Navigator and Microsoft Internet Explorer.

1. While viewing a page in Netscape Navigator, go to the View menu and select Page Source. This opens the Source window, as shown in Figure 10-1. From here you can examine the source code, copying and pasting the code into a text editor if you wish.

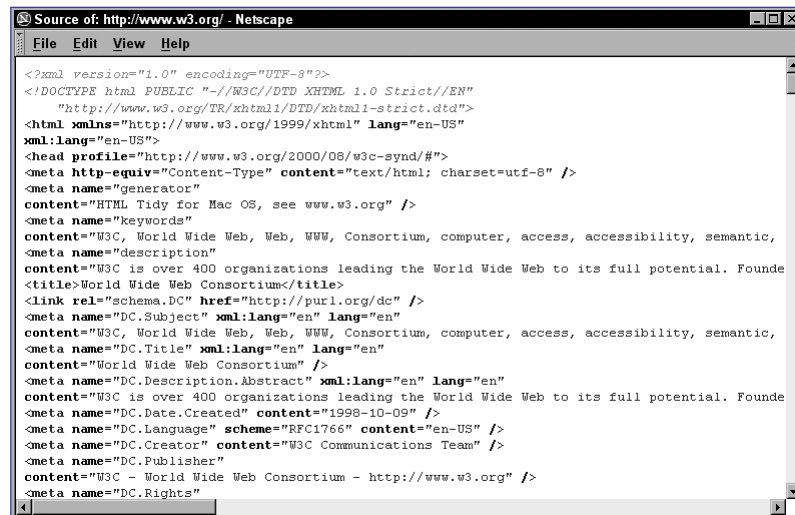


Figure 10-1: The Source window in Netscape Navigator

2. To save the source code from the Source window, select File ⇨ Save Page As and enter a filename.
3. In Microsoft Internet Explorer, select View ⇨ Source. This opens the source code of the document in Notepad, as shown in Figure 10-2.
4. To save the source code, choose File ⇨ Save As from the Internet Explorer window. This opens the Save Web Page dialog box shown in Figure 10-3.
5. In the Save Web Page dialog box, set the Save As Type pop-up menu to Web Page Complete, as shown in Figure 10-4.

6. Choose a location on your hard drive to save the file and click Save. A copy of the HTML document and a folder containing all the associated images and media are saved to the location you chose.

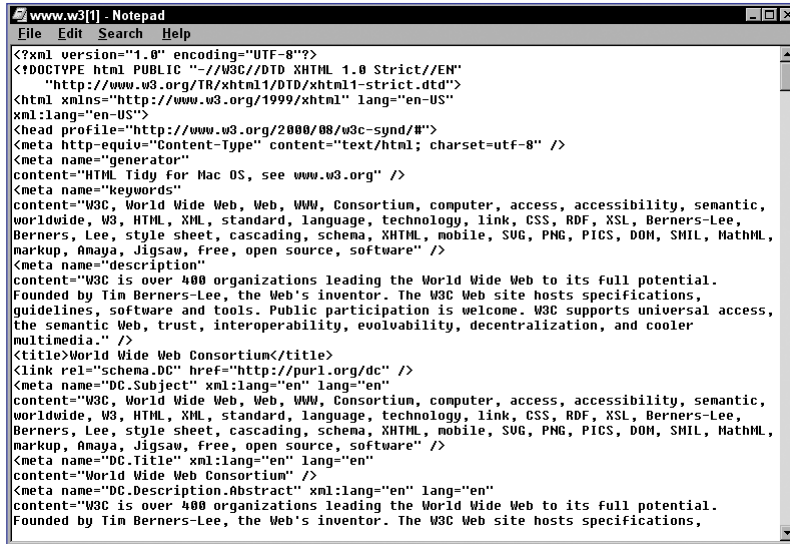


Figure 10-2: Source code in Notepad

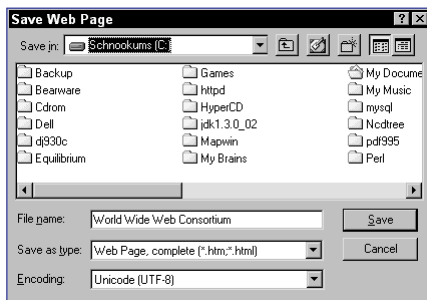


Figure 10-3: The Save Web Page dialog box in Internet Explorer

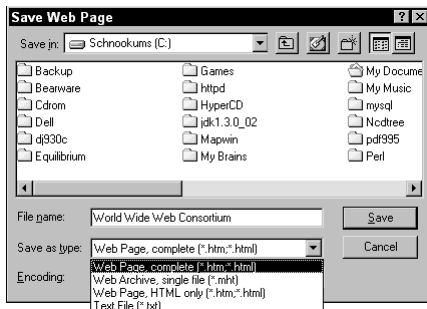


Figure 10-4: Choosing Web Page Complete in the Save As Type menu

## Task

## 10

## tips

- If you are viewing a page in Internet Explorer that's located on your computer, you can open the HTML document in Notepad, make edits, and choose Save As from Notepad's File menu, and add a .htm or .html extension to the file. By clicking the browser's refresh button, you'll see your changes take effect.
- Internet Explorer offers an extremely useful feature for copying an entire Web page to your computer that not only saves a copy of the HTML document but also saves all the images and other media into a folder beside the document. This allows you to later open the file locally and see the document in its entirety. If you then make changes, you can see how they affect all the content. Simply choose Save As, and select Web Page, Complete from the Save As Type menu in the dialog box.

