New Perspectives on

HTML and XHTML

5th Edition

Comprehensive

Patrick Carey
Developing a Web Site

Creating a Web Site for Digital Photography Enthusiasts

Case | CAMshots

Gerry Hayward is an amateur photographer and digital camera enthusiast. He’s decided to create a Web site named CAMshots, where he can offer advice and information to people who are just getting started with digital photography or who are long-time hobbyists like himself and are looking to share tips and ideas. Gerry’s Web site will contain several pages, with each page dedicated to a particular topic. He has created a few pages for the Web site, but he hasn’t linked them together. He has asked your help in designing his site. You’ll start with only a few pages and then Gerry can build on your work as he adds more information to the site.
Session 2.1

Exploring Web Site Structures

You meet with Gerry to discuss his plans for the CAMshots Web site. Gerry has already created a prototype for the Web site. He’s created three Web pages: one page is the site’s home page and contains general information about CAMshots; the second page contains tips about digital photography; and the third page contains a partial glossary of photographic terms. The pages are not complete, nor are they linked to one another. You’ll begin your work for Gerry by viewing these files in your text editor and browser.

To view Gerry’s Web pages:

1. Start your text editor, and then one at a time, open the hometxt.htm, tipstxt.htm, and glosstxt.htm files, located in the tutorial.02\tutorial folder included with your Data Files.

2. Within each file, go to the comment section at top of the file and add your name and the date in the space provided.

3. Save the files as home.htm, tips.htm, and glossary.htm, respectively, in the tutorial.02\tutorial folder.

4. Take some time reviewing the HTML code within each document so that you understand the structure and content of the files.

5. Start your Web browser, and then one at a time, open the home.htm, tips.htm, and glossary.htm files. Figure 2-1 shows the current layout and appearance of Gerry’s three Web pages.

Gerry wants to create links among the three pages so that users can easily navigate from one page to another. Before you write code for the links, it’s worthwhile to map out exactly how you want the pages to relate to each other, using a technique known as storyboarding. A storyboard is a diagram of a Web site’s structure, showing all the pages in the site and indicating how they are linked together. Because Web sites use a variety of structures, it’s important to storyboard your Web site before you start creating your pages. This helps you determine which structure works best for the type of information your site contains. A well-designed structure ensures that users will be able to navigate the site without getting lost or missing important information.

Every Web site starts with a single home page that acts as a focal point for the Web site. It is usually the first page that users see. Starting from the home page, you add the links to other
pages in the site, creating the site's overall structure. The Web sites you commonly encounter as you navigate the Web use one of several different Web structures. Examine some of these structures to help you decide how to design your own sites.

Linear Structures

If you wanted to create an online version of a famous play, like Shakespeare’s *Hamlet*, one method would be to link the individual scenes of the play in a long chain. Figure 2-2 shows the storyboard for this type of linear structure, in which each page is linked with the pages that follow and precede it. Readers navigate this structure by moving forward and backward through the pages, much as they might move forward and backward through the pages of a book.

Linear structures work for Web sites with a clearly defined order of pages that are small in size. However, they can be difficult to work with as the chain of pages increases in length. An additional problem is that in a linear structure you move farther and farther away from the home page as you progress through the site. Because home pages often contain important general information about the site and its author, this is usually not the best design technique.

You can modify this structure to make it easier for users to return immediately to the home page or other main pages. Figure 2-3 shows this online play with an augmented linear structure, in which each page contains an additional link back to the opening page of each act.
Hierarchical Structures

Another popular structure is the **hierarchical structure**, in which the pages are linked going from the home page down to pages dedicated to specific topics. Those pages, in turn, can be linked to even more specific topics. So, a hierarchical structure allows users to easily move from general to specific and back again. In the case of the online play, you can link an introductory page containing general information about the play to pages that describe each of the play's acts, and within each act you can include links to individual scenes. See Figure 2-4. With this structure, a user can move quickly to a specific scene within the page, bypassing the need to move through each scene in the play.

![Figure 2-4](image_url)

**Figure 2-4**  
A hierarchical structure

Mixed Structures

With larger and more complex Web sites, you often need to use a combination of structures. Figure 2-5 shows the online play using a mixture of the three main structures. The overall form is hierarchical, as users can move from a general introduction down to individual scenes; however, users can also move through the site in a linear fashion, going from act to act and scene to scene. Finally, each individual scene contains a link to the home page, allowing users to jump to the top of the hierarchy without moving through the different levels.
As these examples show, a little foresight can go a long way toward making your Web site easier to use. Also keep in mind that search results from a Web search engine such as Google or Yahoo! can point users to any page in your Web site—not just your home page—so they will need to quickly understand what your site contains and how to navigate it. At a minimum, each page should contain a link to the site's home page or to the relevant main topic page. In some cases, you might want to supply your users with a site index, which is a page containing an outline of the entire site and its contents. Unstructured Web sites can be difficult and frustrating to use. Consider the storyboard of the site displayed in Figure 2-6.
This confusing structure makes it difficult for users to grasp the site's contents and scope. The user might not even be aware of the presence of some pages because there are no connecting links, and some of the links only point in one direction. The Web is a competitive place; studies have shown that users who don’t see how to get what they want within the first few seconds often leave a Web site. How long would a user spend on a site like the one shown in Figure 2-6?

Protected Structures

Sections of most commercial Web sites are off-limits except to subscribers and registered customers. As shown in Figure 2-7, these sites have a password-protected Web page that users must go through to get to the off-limits areas. The same Web site design principles apply to the protected section as the regular, open section of the site.
Storyboarding a protected structure is particularly important to ensure that no unmonitored “back doors” to the protected area are allowed in the site design.

**Creating a Hypertext Link**

Gerry wants his site visitors to be able to move effortlessly among the three documents he’s created. To do that, you’ll link each page to the other two pages. Figure 2-8 provides the storyboard for the simple structure you have in mind.

To create these links, you have to add hypertext links to each of the three documents. Hypertext links are created by enclosing some document content with a set of opening and closing `<a>` tags. The general syntax to create a hypertext link is

```
<a href="reference">content</a>
```

where *reference* is the location being linked to and *content* is the document content that is being marked as a link. The *reference* value can be a page on the World Wide Web, a local file, an e-mail address, or a network server. For example, to create a hypertext link to the tips.htm file, you could enter the following code:

```
<a href="tips.htm">Photography Tips</a>
```

This code marks the text “Photography Tips” as a hypertext link. When rendered by the browser, the words “Photography Tips” will be underlined, providing a visual clue to the user that the text is linked to another document. If the user clicks the text with a mouse, the browser will load the linked document (tips.htm).

Filenames are case sensitive on some operating systems, including the UNIX and Macintosh operating systems. Web servers running on those systems differentiate between a file named tips.htm and Tips.htm. For this reason, you might find that links you create on your computer do not work when you transfer your files to a Web server. To avoid this problem, the current standard is to always use lowercase filenames for all Web site files and to avoid using special characters such as blanks and slashes (/).

At the top of the home.htm, tips.htm, and glossary.htm files, Gerry has already entered the names of each of his three documents. Your first task is to mark these names as hypertext links to each of Gerry’s three files. You’ll start with the names in the home.htm file.
To mark content as hypertext, use

```html
<a href="reference">content</a>
```

where `reference` is the location being linked to and `content` is the document content that is being marked as a link.

---

**To create a hypertext link to a document:**

1. Return to the `home.htm` file in your text editor and locate the second div element at the top of the file.
2. Mark the text “Home” as a hypertext link using a set of `<a>` tags as follows:
   ```html
   <a href="home.htm">Home</a>
   ```
3. Mark the text “Tips” as a hypertext link using the following code:
   ```html
   <a href="tips.htm">Tips</a>
   ```
4. Mark the text “Glossary” as a hypertext link as follows:
   ```html
   <a href="glossary.htm">Glossary</a>
   ```

Figure 2-9 highlights the revised text in the `home.htm` file.

---

5. Save your changes to the file.
6. The two other files have the same headings at the top of the document. Go to the `tips.htm` file in your text editor and repeat Steps 2 through 5 for the Home, Tips, and Glossary titles at the top of that file.
7. Go to the `glossary.htm` file in your text editor and repeat Steps 2 through 5 to mark the titles in that document as hypertext links as well.

Now that you’ve added hypertext links to each of the three documents, test those links in your browser.
8. Reload or refresh the `home.htm` file in your Web browser. As indicated in Figure 2-10, the titles at the top of the page should now be underlined, providing visual evidence that these words are treated as hypertext links.
9. Click the **Tips** link from the list of page names. Verify that the browser loads and displays the Tips page.

10. Click the **Glossary** link. Verify that the Glossary page is opened by the browser.

11. Continue to click the hypertext links from the list, confirming that you can jump from any of the three pages to each of the other two pages.

**Trouble?** If the links do not work, check the spelling of the filenames in the href attributes of the `<a>` tags. Because some Web servers require you to match capitalization in a filename, you should verify this in your attributes as well.

---

**Specifying a Folder Path**

In the links you’ve just created, you specified only the filename and not the location of the file. When you specify only the filename, the browser searches for the file in the same folder as the document containing the hypertext link; however, large Web sites containing hundreds of documents often place those documents in separate folders to make them easier to manage.

As Gerry adds more files to his Web site, he will probably want to use folders to organize the files. Figure 2-11 shows a preview of how Gerry might employ those folders. In this case, the topmost folder is named camshots. Gerry has placed some of his HTML files within the pages folder, which he has then divided into three subfolders named tips, glossary, and articles. He has also created separate folders for the images and video clips used on his Web site. Figure 2-11 displays the location of four HTML files named index.htm, tips1.htm, tips2.htm, and glossary.htm.
To create a link to a file located in a different folder than the current document, you must specify the file’s location, or **path**, so that browsers can find it. HTML supports two kinds of paths: absolute and relative.
Absolute Paths

An **absolute path** specifies a file’s precise location within a computer’s entire folder structure. Absolute pathnames employ the syntax

```
/folder1/folder2/folder3/file
```

where `folder1` is the topmost folder in the computer’s folder tree, followed by `folder2`, `folder3`, and so forth, down to the file you want to link to. Figure 2-12 shows how you would express absolute paths to the four files listed in Figure 2-11.

<table>
<thead>
<tr>
<th>Absolute Path</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/camshots/pages/tips/tips1.htm</code></td>
<td>The tips1.htm file located in the pages/tips subfolder</td>
</tr>
<tr>
<td><code>/camshots/pages/tips/tips2.htm</code></td>
<td>The tips2.htm file located in the pages/tips subfolder</td>
</tr>
<tr>
<td><code>/camshots/pages/glossary/glossary.htm</code></td>
<td>The glossary.htm file located in the pages/glossary subfolder</td>
</tr>
<tr>
<td><code>/camshots/index.htm</code></td>
<td>The index.htm file located in the camshots folder</td>
</tr>
</tbody>
</table>

If files are located on different drives as well as in different folders, you must include the drive letter in the form

```
/drive/folder1/folder2/folder3/file
```

where `drive` is the letter assigned to the drive. For example, the tips1.htm file located on drive C in the `/camshots/pages/tips` folder would have the absolute path:

```
/C/camshots/pages/tips/tips1.htm
```

Remember that you don’t have to include a drive letter if the destination document is located on the same drive as the document containing the link.

Relative Paths

When many folders and subfolders are involved, absolute pathnames can be cumbersome and confusing. For that reason, most Web designers prefer to use relative paths. A **relative path** specifies a file’s location in relation to the location of the current document. If the file is in the same location as the current document, the relative path is simply the filename. If the file is in a subfolder of the current document, include the name of the subfolder without the forward slash in the form

```
folder/file
```

where `folder` is the name of the subfolder. To go farther down the folder tree to other subfolders, include those in the relative path separated by forward slashes, i.e.

```
folder1/folder2/folder3/file
```

where `folder1`, `folder2`, `folder3`, and so forth are subfolders of the current folder. Finally, a relative path can go up the folder tree by starting the pathname with a double period (..) followed by a forward slash and the name of the file. The path

```
../file
```

references the `file` document located in the parent folder of the current document. To reference a different folder on the same level of the folder tree, known as a **sibling folder**,
you move up the folder tree using the double period (..) and then down using the name of the sibling folder. The general syntax is

```
../folder/file
```

where `folder` is the name of the sibling folder. Figure 2-13 shows the relative paths to the six files in the tree from Figure 2-11, starting from the `camshots/pages/tips` subfolder.

### Table: Relative Paths

<table>
<thead>
<tr>
<th>Relative Path from the /camshots/pages/tips Subfolder</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>tips1.htm</td>
<td>The tips1.htm file located in the current folder</td>
</tr>
<tr>
<td>tips2.htm</td>
<td>The tips2.htm file located in the current folder</td>
</tr>
<tr>
<td>../glossary/glossary.htm</td>
<td>The glossary.htm file located in the sibling glossary folder</td>
</tr>
<tr>
<td>../../index.htm</td>
<td>The index.htm file located in the parent camshots folder</td>
</tr>
</tbody>
</table>

You should almost always use relative paths in your links. If you have to move your files to a different computer or server, you can move the entire folder structure without having to change the relative pathnames you created. If you use absolute pathnames, you will probably have to revise each link to reflect the new location of the folder tree on the computer.

### Changing the Base

As you’ve just seen, a browser resolves relative pathnames based on the location of the current document. You can change this behavior by specifying a different base or starting location for all relative paths. The code to specify a different base is

```
<base href="path" />
```

where `path` is the folder location that you want the browser to use when resolving relative paths in the current document. The base element has to be added to the head section of the HTML file and will be applied to all hypertext links found within the document.

### Using the Base Element to Set the Default Location of Relative Paths

- To set the default location for a relative path, add the element
  ```html
  <base href="path" />
  ```
  to the document head, where `path` is the folder location that you want the browser to use when resolving relative paths in the current document.

The base element is useful when a single document is moved to a new folder. Rather than rewriting all of the relative paths to reflect the document’s new location, the base element redirects browsers to the document’s old location, allowing any relative paths to be resolved as they were before.
## Managing Your Web Site

Web sites can quickly grow from a couple of pages to dozens or hundreds of pages. As the size of the site increases, it becomes more difficult to get a clear picture of the site's structure and content. Imagine deleting or moving a file in a Web site that contains dozens of folders and hundreds of files. Can you easily project the effect of this change? Will all of your hypertext links still work after you move or delete the file?

To effectively manage a Web site, you should follow a few important rules. The first is to be consistent in how you structure the site. If you decide to collect all image files in one folder, you should follow that rule as you add more pages and images. Web sites are more likely to break down if files and folders are scattered throughout the server without a consistent rule or pattern. Decide on a structure early on and stick with it.

The second rule is to create a folder structure that matches the structure of the Web site itself. If the pages can be easily categorized into different groups, that grouping should also be reflected in the grouping of the subfolders. The names you assign to your files and folder should also reflect their use on the Web site. This makes it easier for you to predict how modifying a file or folder will impact other pages on the site.

Finally, you should document your work by adding comments to each new Web page. Comments are useful not only for colleagues who may be working on the site, but also for the author who has to revisit those files months or even years after creating them. The comments should include:

- The page's filename and location
- The page's author and the date the page was initially created
- A list of any supporting files used in the document, such as image and audio files
- A list of the files and their locations that link to the page
- A list of the files and their locations that the page links to

By following these rules, you can reduce a lot of the headaches associated with maintaining a large and complicated Web site.

You’ve completed your initial work linking the three files in Gerry’s Web site. In the next session, you’ll learn how to work with hypertext links that point to locations within files. If you want to take a break before starting the next session, you can close your files and your Web browser now.

### Session 2.1 Quick Check

1. What is storyboarding? Why is it important in creating a Web page system?
2. What is a linear structure? What is a hierarchical structure?
3. What code would you enter to link the text “Sports Info” to the sports.htm file? Assume that the current document and sports.htm are in the same folder.
4. What’s the difference between an absolute path and a relative path?
5. Refer to Figure 2-11. If the current file is in the camshots/pages/glossary folder, what are the relative paths for the four files listed in the folder tree?
6. What is the purpose of the base element?
Session 2.2

Linking to Locations within Documents

Gerry likes the links you’ve created in the last session and would like you to add some more links to the Glossary page. Recall that the Glossary page contains a list of digital photography terms. The page is very long, requiring users to scroll through the document to find a term of interest. At the top of the page Gerry has listed the letters A through Z. Gerry wants to give users the ability to jump to a specific section of the document by clicking a letter from the list. See Figure 2-14.

Using the id Attribute

To jump to a specific location within a document, you first need to mark that location. One way of doing this is to add an id attribute to an element at that location in the document. The syntax of the id attribute is

```
<element id="id"/>
```

where id is the value of the element id. For example, the following code marks the h2 element with an id value of H:

```
<h2 id="H">H</h2>
```

Note that id names must be unique. If you assign the same id name to more than one element on your Web page, the browser uses the first occurrence of the id name. XHTML documents will be rejected if they contain elements with duplicate ids. Id names are not case sensitive, so browsers do not differentiate between ids named top and TOP.
The Glossary page has only a partial list of the photography terms that Gerry will eventually add to his Web site. For now, you’ll only mark sections in the glossary corresponding to the letters A through F.

To add the id attribute to h2 headings:

1. Return to the glossary.htm file in your text editor.

2. Scroll down the file and locate the h2 heading for the letter A. Within the opening <h2> tag, insert the following attribute:
   
   ```html
   id="A"
   ```

3. Locate the h2 heading for the letter B and insert the following attribute in the opening <h2> tag:
   
   ```html
   id="B"
   ```

   Figure 2-15 highlights the revised code.

4. Continue going down the file, adding id attributes to the opening <h2> heading tags for C, D, E, and F corresponding to the letters of those headings.

For longer documents like the Glossary page, it’s also helpful to the reader to be able to jump directly from the bottom of a long page to the top of the page rather than having to scroll back up. With that in mind, you’ll also add an id attribute marking the element at the top of the page.

To mark the top of the page:

1. Scroll up the glossary.htm file in your text editor and locate the div element directly below the opening <body> tag.

2. Insert the following attribute within the opening <div> tag, as shown in Figure 2-16:
   
   ```html
   <div id="top"/>
   ```

   Figure 2-16 adds an id attribute to the div element.
Linking to an id

Once you've marked an element using the id attribute, you can create a hypertext link to that element using the hypertext link

\[
\text{\textless a href="\#id"\textgreater content\textless/a\textgreater}
\]

where \textit{id} is the value of the id attribute of the element. For example, to create a link to the h2 heading for the letter A in the glossary document, you would enter the following code:

\[
\text{\textless a href="\#A"\textgreater A\textless/a\textgreater}
\]

Use this code to change the entries on the Glossary page to hypertext links pointing to the section of the glossary corresponding to the selected letter.

To change the list of letters to hypertext links:

1. Locate the letter A in the list of letters at the top of the \textit{glossary.htm} file.
2. After the [ character, insert the following opening tag:

\[
\text{\textless a href="\#A"\textgreater}
\]

3. Between the letter A and the ] character, insert closing \textless/a\textgreater tag. Figure 2-17 shows the revised code.

4. Mark the letters B through F in the list as hypertext links pointing to the appropriate h2 headings in the document. Figure 2-18 shows the revised code for the list of letters.

Gerry also wants you to create a hypertext link at the bottom of the file that points to the top (using the id attribute you created in the last set of steps).

5. Scroll to the bottom of the file and locate the text “Return to Top.”
6. Mark the text as hypertext, pointing to the element with an id value of top. See Figure 2-19.
7. Save your changes to the file and then reload or refresh the glossary.htm file in your Web browser.

8. As shown in Figure 2-20, the letters A through F in the alphabetic list are displayed as hypertext links. Click the link for F and verify that you jump down to the end of the document, where the photographic terms starting with the letter F are listed.

9. Click the Return to Top hypertext link and verify that you jump back to the top of the document.

10. Click the other links within the document and verify that you jump to the correct sections of the glossary.

Trouble? The browser cannot scroll farther than the end of the page. So, you might not see any difference between jumping to the E section of the glossary and jumping to the F section.
Working with Anchors

Early browser versions might not support the use of the id attribute as a way of marking document elements. These early browser versions instead used anchors or bookmarks to mark document locations. The syntax of the anchor element is

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

where **anchor** is the name of the anchor that marks the location of the document **content**. For example, to mark the h2 heading with an anchor of “A,” you would enter the following code:

```html
<h2><a name="A">A</a></h2>
```

Marking a location with an anchor does not change your document’s appearance in any way; it merely creates a destination within your document.

You use the same syntax to link to locations marked with an anchor as you would with locations marked with id attributes. To link to the above anchor, you could use the following code:

```html
<a href="#A">A</a>
```

The use of anchors is a deprecated feature of HTML and is not supported in strict applications of XHTML, but you will still see anchors used in older code and in code generated by HTML editors and converters.

Creating Links between Documents

Gerry knows that the glossary will be one of the most useful parts of his Web site, especially for novice photographers. However, he’s also aware that most people do not read through glossaries. He would like to create links from the words he uses in his articles to glossary entries so that readers of his articles can quickly access definitions for terms they don’t understand. His articles are not on the same page as his Glossary page, so he will have to create a link between those pages and specific glossary entries.

To create a link to a specific location in another file, enter the code

```html
<a href="reference#id">content</a>
```

where **reference** is a reference to an HTML or XHTML file and **id** is the id of an element marked within that file. For example, the code

```html
<a href="glossary.htm#D">"D" terms in the Glossary</a>
```

creates a hypertext link to the D section in the glossary.htm file. This assumes that the glossary.htm file is located in the same folder as the document containing the hypertext link. If not, you have to include either the absolute or relative path information along with the filename, as described in the last session.

Reference Window | Linking to an id

- To link to a specific location within the current file, use
  ```html
  <a href="#id">content</a>
  ```
  where **id** is the id value of an element within the document.
- To link to a specific location in another file, use
  ```html
  <a href="reference#id">content</a>
  ```
  where **reference** is a reference to an external file and **id** is the id value of an element in that file.

On Gerry’s home page, he wants to showcase a Photo of the Month, displaying a photo that his readers might find interesting or useful in their own work. Along with the photo, he has included the digital camera settings used in taking the photo. Many of the
camera settings are described on the Glossary page. Gerry suggests that you create a link between the setting name and the glossary entry. The five entries he wants to link to are: F-stop, Exposure, Focal Length, Aperture, and Flash Mode. Your first step is to mark these entries in the glossary using the id attribute.

**To mark the glossary entries:**

1. Return to the `glossary.htm` file in your text editor.
2. Scroll down the file and locate the Aperture definition term.
3. As shown in Figure 2-21, within the opening `<dt>` tag, insert the attribute

   ```html
   id="aperture"
   ```

4. Scroll down the file and locate the Exposure definition term.
5. Within the opening `<dt>` tag, insert the following attribute:

   ```html
   id="exposure"
   ```

6. Go to the F section of the glossary and mark the terms with the following ids:

   - F-stop with the id `f-stop`
   - Flash Mode with the id `flash_mode`
   - Focal Length with the id `focal_length`

7. Save your changes to the `glossary.htm` file.

Next you’ll go to the Home page and create links from these terms in the Photo of the Month description to their entries on the Glossary page.

**To create links to the glossary entries:**

1. Open the `home.htm` file in your text editor.
2. Scroll down the file and locate the F-stop term from the unordered list.
3. Mark “F-stop” as a hypertext link using the following code:

   ```html
   <a href="glossary.htm#f-stop">F-stop</a>
   ```

4. Mark “Exposure” as a hypertext link with:

   ```html
   <a href="glossary.htm#exposure">Exposure</a>
   ```

5. Mark the remaining three entries in the unordered list as hypertext pointing to their corresponding entries on the Glossary page. Figure 2-22 highlights the revised code in the file.
6. Save your changes to the file.

7. Refresh the `home.htm` file in your Web browser. As shown in Figure 2-23, the settings from the Photo of the Month description are now displayed as hypertext links.

8. Click the **F-stop** hypertext link and verify that you jump to the Glossary page with the F-stop entry displayed in the browser window.

9. Return to the **CAMshots home page** and click the hypertext links for the other terms in the list of photo settings, verifying that you jump to the section of the glossary that displays that term’s definition.
Working with Linked Images and Image Maps

A standard practice on the Web is to turn the Web site’s logo into a hypertext link pointing to the home page. This gives users a quick reference point to the home page rather than searching for a link to the home page. To mark an inline image as a hypertext link, you enclose the <img> tag within a set of <a> tags as follows:

```html
<a href="reference"><img src="file" alt="text" /></a>
```

Once the image has been linked, clicking anywhere within the image jumps the user to the linked file.

Introducing Image Maps

When you mark an inline image as a hypertext link, the entire image is linked to the same destination file; however, HTML also allows you to divide an image into different zones, or hotspots, each linked to a different destination. Therefore, a single inline image can be linked to several locations. Gerry is interested in doing this with the CAMshots logo. He would like you to create hotspots for the logo so that if the user clicks anywhere within the CAMshots circle on the left side of the logo, the user jumps to the Home page, while clicking either Tips or Photo Glossary in the logo takes the user to the Tips page or the Glossary page. See Figure 2-24.

To define these hotspots, you create an image map that matches a specified region of the inline image to a specific destination. HTML supports two kinds of image maps: client-side image maps and server-side image maps. You’ll first study how to create a client-side image map.
Client-Side Image Maps

A client-side image map is an image map that is handled entirely by the Web browser running on the user’s computer. Client-side image maps are defined with the map element

```html
<map id="map" name="map">
    hotspots
</map>
```

where map is the name of the image map and hotspots are the locations of the hotspots within the image. Each image map has to be given an id and a name. You have to include both attributes, setting them to the same value, because HTML code requires the name attribute and XHTML requires the id attribute. As long as you include both, your code will work under all browsers. For example, the following code creates a map element named logomap:

```html
<map id="logomap" name="logomap">
    ...
</map>
```

Map elements can be placed anywhere within the body of the Web page because they are not actually displayed by the browser, but used as references for mapping hotspots to inline images. The common practice is to place the map element below the inline image.

Defining Hotspots

The individual hotspots are defined using the area element

```html
<area shape="shape" coords="coordinates" href="reference" alt="text" />
```

where shape is the shape of the hotspot region, coordinates are the list of points that define the boundaries of the region, reference is the file or location that the hotspot is linked to, and text is alternate text displayed for nongraphical browsers. Hotspots can be created in the shape of rectangles, circles, or polygons (multisided figures). So, the shape attribute can have the value rect for a rectangular hotspot, "circle" for a circular hotspot, and “poly” for a polygonal or multisided hotspot. A fourth shape option is “default,” representing the remaining area of the inline image not covered by hotspots. There is no limit to the number of area elements you can add to an image map. Hotspots can also overlap. If they do and the user clicks an overlapping area, the browser opens the link of the first hotspot defined in the map.

Hotspot coordinates are measured in pixels, which are the smallest unit or dot in a digital image or display. Your computer monitor might have a size of 1024 x 768 pixels, which means that the display is 1024 dots wide by 768 dots tall. The CAMshots logo that Gerry uses in his Web site has a dimension of 778 pixels wide by 164 pixels tall. When used with the coords attribute of the area element, the pixel values exactly define the location and size of the hotspot region.

Each hotspot shape has a different set of coordinates that define it. To define a rectangular hotspot, enter

```html
<area shape="rect" coords="x1, y1, x2, y2" ...
```

where x1, y1 are the coordinates of the upper-left corner of the rectangle and x2, y2 are the coordinates of the rectangle’s lower-right corner. Figure 2-25 shows the coordinates of the rectangular region surrounding the Photo Glossary hotspot.
The upper-left corner of the rectangle has the coordinates (240, 110). The lower-left corner is found at the coordinates (402, 145). Coordinates are always expressed relative to the image's top-left corner. A coordinate of (240, 110) refers to a point that is 240 pixels to the right and 110 pixels down from the image's top-left corner.

Circular hotspots are defined using the area element

```html
<area shape="circle" coords="x, y, r" ... />
```

where \(x\) and \(y\) are the coordinates of the center of the circle and \(r\) is the circle’s radius. Figure 2-26 shows the coordinates for a circular hotspot around the CAMshots image from the Web site logo. The center of the circle is located at the coordinates (82, 78) and the circle has a radius of 80 pixels.

Polygonal hotspots are defined with

```html
<area shape="poly" coords="x1, y1, x2, y2, x3, y3, ..." ... />
```

where \((x1, y1), (x2, y2), (x3, y3)\) and so forth define the coordinates of each corner in the multisided shape. Figure 2-27 shows the coordinates for a triangular-shaped hotspot with corners at (30, 142), (76, 80), and (110, 142). With polygonal hotspots, you can create a wide variety of shapes as long you know the coordinates of each corner.
Finally, to define the default hotspot for the image use

```
<area shape="default" coords="0, 0, x, y" ...
```

where x is the width of the inline image in pixels and y is the image’s height. Any spot in the inline image that is not covered by another hotspot will activate the default hotspot link.

To determine the coordinates of a hotspot, you can use either a graphics program such as Adobe Photoshop or image map software that automatically generates the HTML code for the hotspots you define.

In this case, assume that Gerry has already determined the coordinates for the hotspots in his image map and provided them for you. He has three hotspots that he wants you to create, shown earlier in Figure 2-24. The first is a circular hotspot linked to the home.htm file, centered at the point (82, 78) and having a radius of 80 pixels. The second is a rectangular hotspot, linked to the tips.htm file with corners at (168, 110) and (225, 145). The third is also rectangular, linked to the glossary.htm file with corners at (240, 110) and (402, 145). You do not have to create a polygonal hotspot.

You’ll name the image map containing these hotspots logomap.

---

**To create an image map:**

1. Return to **home.htm** file in your text editor.

2. Directly below the `<img>` tag for the CAMshots inline image, insert the following map element:

   ```html
   <map id="logomap" name="logomap">
   </map>
   
3. Within the map element, insert a circular hotspot that points to the home.htm file using the following area element:

   ```html
   <area shape="circle" coords="82, 78, 80" href="home.htm" alt="Home" />
   
4. Directly below the `<area>` tag for the circular hotspot, insert the following two rectangular hotspots pointing to the tips.htm and glossary.htm files:

   ```html
   <area shape="rect" coords="168, 110, 225, 145" href="tips.htm" alt="Tips" />
   <area shape="rect" coords="240, 110, 402, 145" href="glossary.htm" alt="Glossary" />
   
Figure 2-28 highlights the new code in the file.
Creating an image map

To create a client-side image map, insert the map element

```html
<map name="map" id="map">
  hotspots
</map>
```

anywhere within the Web page body, where map is the name and id of the image map and hotspots is a list of hotspot areas defined within the image map.

To add a hotspot to the image map, place the element

```html
<area shape="shape" coords="coordinates" href="reference" alt="text" />
```

within the map element, where shape is the shape of the hotspot region, coordinates are the list of points that define the boundaries of the region, reference is the file or location that the hotspot is linked to, and text is alternate text displayed for nongraphical browsers.

To define a rectangular-shaped hotspot, use the area element

```html
<area shape="rect" coords="x1, y1, x2, y2" ... />
```

where x1, y1 are the coordinates of the upper-left corner of the rectangle and x2, y2 are the coordinates of the rectangle's lower-right corner.

To define a circular hotspot, use

```html
<area shape="circle" coords="x, y, r" ... />
```

where x and y are the coordinates of the center of the circle and r is the circle's radius.

To define a polygonal hotspot, use

```html
<area shape="poly" coords="x1, y1, x2, y2, x3, y3, ..." ... />
```

where (x1, y1), (x2, y2), (x3, y3), and so forth define the coordinates of each corner in the multisided shape.

To define the default hotspot, use

```html
<area shape="default" coords="0, 0, x, y" ... />
```

where x is the width of the inline image in pixels and y is the height in pixels.

To apply an image map to an inline image, add the usemap attribute

```html
<img src="file" alt="text" usemap="#map" />
```

to the img element, where map is the name or id of the map element.

Now that you've defined the image map, your next task is to apply the map to the CAMshots logo.
Applying an Image Map

To apply an image map to an image, add the usemap attribute to the inline image’s `<img>` tag. The syntax is

```
<img src="file" alt="text" usemap="#map" />
```

where `map` is the id or name of the map element. If you place the map element in a separate file, you can reference it using the code

```
<img src="file" alt="text" usemap="reference#map" />
```

where `reference` is a reference to an HTML or XHTML file containing the map element. Unfortunately, most browsers do not support this option, so you should always place the image map in the same file as the inline image. You’ll apply the logomap to the CAM-shots logo and then test it on your Web browser.

To apply the logomap image map:

1. Add the following attribute to the `<img>` tag for the CAMshots logo, as shown in Figure 2-29.
   ```html
   usemap="#logomap"
   ```

   ![Figure 2-29 Applying an image map](image)

   imagemap

   map element name or id

2. Save your changes to the file and reload or refresh the `home.htm` file in your Web browser.

   **Trouble?** Depending on your browser, you might see a border around the CAM-shots logo, which you can ignore for now. You’ll remove it shortly.

3. Click anywhere within the word **Tips** in the logo image and verify that the browser opens the Tips page.

4. Return to the home page and click anywhere within the word **Photo Glossary** to verify that the browser opens the Glossary page.

After changing the logo to a hypertext link, you may have noticed that you have added a border around the image. Hypertext links are usually underlined in the Web page; but with inline images, the image is displayed with a lined border. Gerry would prefer not to have a border because he feels that it detracts from the logo’s appearance. He asks if you can remove the border but still keep the logo functioning as a hypertext link.

To remove the border, you can apply a border-width style to the inline image. By setting the width of the border to zero, you will effectively remove it from the logo. The style attribute to change the width of a border is

```html
style="border-width: 0"
```
Removing the Border from an Inline Image

- To remove a border from an inline image, add the following attribute to the `<img>` tag:
  ```html
  style="border-width: 0"
  ```

Use the border-width style to remove the border from the CAMshots logo on the three pages of Gerry's Web site.

To set the border width of the CAMshots logo to 0:

1. Return to the `home.htm` file in your text editor.
2. Add the following style attribute to the `<img>` tag for the logo inline image, as shown in Figure 2-30.
   ```html
   style="border-width: 0"
   ```
3. Save your changes to the file.
4. Reload the `home.htm` file in your browser and verify that the border has been removed from the image.

Now that you've created an image map for the logo on the home page, you can create similar image maps for the logos on the Tips and Glossary pages.

To add image maps to the other Web pages:

1. Return to the `tips.htm` file in your text editor.
2. Replace the code within the `div` element for the logo image with the code shown earlier in Figure 2-30. (Hint: You can use the copy and paste feature of your text editor to copy the code from the home.htm file into the tips.htm file.)
3. Save your changes to the file.
4. Go to the `glossary.htm` file in your text editor.
5. As you did for the tips.htm file, replace the code within the `div` element for the logo image with the code from the home.htm file. Save your changes to the file.
6. Return to the `home.htm` file in your Web browser and verify that you can switch among the three Web pages by clicking the hotspots in the CAMshots logo.
7. If you want to take a break before starting the next session, close your files and programs now.
Server-Side Image Maps

The other type of image map you might encounter on the Web is a **server-side image map**, which is stored on the Web server rather than entered into the HTML code of the Web page. When you click a hotspot on a server-side image map, the coordinates of the mouse click are sent to the server, which activates the corresponding link, downloading the page to your Web browser.

The server-side image map was the original HTML standard and is still supported on the Web. However, this map has some limitations compared to client-side image maps. Because the map is located on the server, you cannot test your Web page without server access. Also, server-side image maps might be slower because information must be sent to the server with each mouse click. Finally, unlike client-side image maps, server-side image maps require the use of a mouse. This makes them unsuitable for users with disabilities or users running non-graphical browsers.

To create a server-side image map, enclose the inline image with a hypertext link such as

```html
<a href="map">
  <img src="file" alt="text" ismap="ismap" />
</a>
```

where `map` is the name of a program or file running on the Web server that will handle the image map. The `ismap` attribute tells the Web browser to treat the inline image as an image map.

At this time, you do not foresee a need to use a server-side image map in the CAMshots Web site. In any future projects, you’ll continue to work with client-side maps.

---

InSight | **Writing Effective Hypertext Links**

To make it easier for users to navigate your Web site, you should follow a few key design tips. Write the text of your hypertext links so that they tell the reader exactly what type of document the link points to. For example, the link text

*Click here for more information.*

doesn’t tell the user what type of document will appear when “here” is clicked. In the place of phrases like “click here,” use descriptive link text such as:

*For more information, view a list of frequently asked questions.*

If the link points to a non-HTML file, such as a PDF document, include that information in the link text. If the linked document is extremely large and will take a while to download to the user’s computer, include that information in your link text so that users can decide whether or not to initiate the transfer. The following link text informs users of the size of the video clip before they initiate the link:

*Download the video clip (16 MB).*

Make your link text easy to locate. Because most browsers underline hypertext links, don’t use underlining for other text elements; use italic or boldface fonts instead. Users should never be confused about what is a link and what is not. Also, if you apply a color to your text, do not choose colors that will make the linked text harder to pick out against the Web page background.

Gerry is pleased with the progress you’ve made on his Web site. Adding the links to the glossary and within the CAMshots logo has made his site easier to navigate. However, there are many other sources of information about digital photography and digital cameras that Gerry wants to make available to his readers. In the next session you’ll examine how to create links between his Web site and other sites on the World Wide Web.
Session 2.2 Quick Check

1. Specify the code for marking the text “CAMshots FAQ” as an h2 heading with the id “faq.”
2. Specify the code for marking the text “Read our FAQ” as hypertext linked to an element in the current document with the id “faq.”
3. Specify the code for marking the text “Read our FAQ” as a hypertext link, pointing to an element with the id “faq” in the help.htm file. Assume that help.htm lies in the same folder as the current document.
4. Specify the code for placing an anchor with the name “faq” within the h2 heading “CAMshots FAQ.”
5. For marking locations within a Web page, what is one advantage of using anchors rather than the id attribute? What is one disadvantage?
6. The CAMmap image map has a circular hotspot centered at the point (50, 75) with a radius of 40 pixels pointing to the faq.htm file. Specify the code to create this map element with that circular hotspot.
7. An inline image based on the logo.jpg file with the alternative text “CAMshots” needs to use the CAMmap image map. Specify the code to apply the image map to the image.
8. What attribute do you add to the inline image from the previous question to remove its border?

Session 2.3

Linking to Resources on the Internet

Gerry has a final set of tasks for you. In the tips.htm file, he has listed some of the Web sites he finds useful in his study of photography. He would like to change the entries in this list to hypertext links that his readers can click to quickly access the sites.

Introducing URLs

To create a link to a resource on the Internet, you need to know its URL. A URL, or Uniform Resource Locator, specifies the precise location of a resource on the Internet. Examples of URLs include www.whitehouse.gov, the home page of the President of the United States, and www.w3.org, the home page of the World Wide Web consortium. All URLs share the common form

```
scheme:location
```

where scheme indicates the type of resource referenced by the URL and location is the location of that resource. For Web pages, the location refers to the location of the HTML file; but for other resources, the location might simply be the name of the resource. For example, a link to an e-mail account has the e-mail address as the resource.

The name of the scheme is taken from the protocol used to access the resource. A protocol is a set of rules defining how information is passed between two devices. Your Web browser communicates with Web servers using the Hypertext Transfer Protocol or HTTP. Therefore, the URLs for all Web pages must start with the http scheme. This tells the browser to use http when it tries to access the Web page. Other Internet resources, described in Figure 2-31, use different communication protocols and have different scheme names.

Tip

Because URLs cannot contain blank spaces, avoid blank spaces in Web site file and folder names.
### Internet protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Used To</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>access documents stored locally on a user’s computer</td>
</tr>
<tr>
<td>ftp</td>
<td>access documents stored on an FTP server</td>
</tr>
<tr>
<td>gopher</td>
<td>access documents stored on a gopher server</td>
</tr>
<tr>
<td>http</td>
<td>access Web pages stored on the World Wide Web</td>
</tr>
<tr>
<td>https</td>
<td>access Web pages over a secure encrypted connection</td>
</tr>
<tr>
<td>mailto</td>
<td>open a user’s e-mail client and address a new message</td>
</tr>
<tr>
<td>news</td>
<td>connect to a Usenet newsgroup</td>
</tr>
<tr>
<td>telnet</td>
<td>open a telnet connection to a specific server</td>
</tr>
<tr>
<td>wais</td>
<td>connect to a Wide Area Information Server database</td>
</tr>
</tbody>
</table>

### Linking to a Web Site

The URL for a Web page has the general form

```plaintext
http://server/path/filename#id
```

where `server` is the name of the Web server, `path` is the path to the file on that server, `filename` is the name of the file, and if necessary, `id` is the name of an id or anchor within the file. A Web page URL can also contain specific programming instructions for a browser to send to the Web server (a topic beyond the scope of this tutorial). Figure 2-32 shows the URL for a sample Web page with all of the parts identified.

You might have noticed that a URL like `http://www.camshots.com` doesn’t include any pathname or filename. If a URL doesn’t specify a path, then it indicates the topmost folder in the server’s directory tree. If a URL doesn’t specify a filename, the server will return to the default home page. Many servers use index.html as the filename for the default home page, so a URL like `http://www.camshots.com/index.html` would be equivalent to `http://www.camshots.com`. 
The server name portion of the URL is also called the **domain name**. By studying the domain name you learn about the server hosting the Web site. Each domain name contains a hierarchy of names separated by periods (.), with the topmost level appearing at the end. The top level, called an **extension**, indicates the general audience supported by the Web server. For example, .edu is the extension reserved for educational institutions, .gov is used for agencies of the United States government, and .com is used for commercial sites or general-use sites.

The next lower level appearing before the extension displays the name of the individual or organization hosting the site. A domain name like camshots.com indicates a commercial or general use site owned by CAMshots. To avoid duplicating domain names, the two top-most levels of the domain have to be registered with the IANA (Internet Assigned Numbers Authority) before they can be used. You can usually register your domain name through your Internet Service Provider. Be aware that you will have to pay an annual fee to keep the domain name.

The lowest levels of the domain, which appear farthest to the left in the domain name, are assigned by the individual or company hosting the site. Large Web sites involving hundreds of pages typically divide their domain names into several levels. For example, a large company like Microsoft might have one domain name for file downloads—`downloads.microsoft.com`—and another for customer service—`service.microsoft.com`. Finally, the lowest level of the domain, the first part of the domain name, displays the name of the hard drive or resource storing the Web site files. Many companies have standardized on using “www” as the name of the lowest level in their domain.

Gerry has listed four Web pages that he wants his readers to be able to access. He's provided you with the URLs for these pages, which are shown in Figure 2-33.

<table>
<thead>
<tr>
<th>Web Site</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apogee Photo</td>
<td><a href="http://www.apogeephoto.com">http://www.apogeephoto.com</a></td>
</tr>
<tr>
<td>Outdoor Photographer</td>
<td><a href="http://www.outdoorphotographer.com">http://www.outdoorphotographer.com</a></td>
</tr>
<tr>
<td>PCPhoto</td>
<td><a href="http://www.pcphotomag.com">http://www.pcphotomag.com</a></td>
</tr>
<tr>
<td>Popular Photography and Imaging</td>
<td><a href="http://www.popphoto.com">http://www.popphoto.com</a></td>
</tr>
</tbody>
</table>

To create a link to these Web sites from your document, you need to mark some text as a hypertext link, using the URL of the Web site as the value of the href attribute. So to link the text “Apogee Photo” to the Apogee Photo Web site, you would enter the following code:

```html
<a href="http://www.apogeephoto.com">Apogee Photo</a>
```

Use the information that Gerry has given you to create links to all four of the Web sites listed on his tips page.

**To create links to sites on the Web:**

1. Return to the **tips.htm** file in your text editor.
2. Scroll to the bottom of the file and locate the definition list containing the list of Web sites.
3. Mark the entry for Apogee Photo as a hypertext link using the following code:

```html
<a href="http://www.apogeephoto.com">Apogee Photo</a>
```

4. Mark the remaining three entries in the list as hypertext links pointing to each company's Web site. Figure 2-34 highlights the revised code in the file.

5. Save your changes to the file.

6. Reload or refresh the `tips.htm` file in your Web browser. Figure 2-35 shows the revised list with each entry appearing as a hypertext link.

7. Click each of the links on the page and verify that the appropriate Web site opens.

**Trouble?** To open these sites, you must be connected to the Internet. If you are still having problems, compare your code to the URLs listed in Figure 2-34 to confirm that you have not made a typing error. Also keep in mind that because the Web is constantly changing, the Web sites for some of these links might have changed, or a site might have been removed since this book was published.

Web pages are only one type of resource that you can link to. Before continuing work on the CAMshots Web site, you should explore how to access some of these other resources.

### Linking to FTP Servers

Another method of storing and sharing files on the Internet is through FTP servers. **FTP servers** are file servers that act like file cabinets in which users can store and retrieve data files, much as they store and retrieve files from their own computer. FTP servers transfer information using a communications protocol called **File Transfer Protocol**, or FTP for short. The URL to access an FTP server follows the general format

```
ftp://server/path/
```
where *server* is the name of the FTP server and *path* is the folder path on the server that contains the files you want to access. When you access the FTP site, you can navigate through its folder tree as you would navigate the folders on your own hard disk. Figure 2-36 shows how someone can use Internet Explorer to view the FTP site and how the site appears as a collection of folders that can be opened and viewed.

FTP servers require each user to enter a password and a username to gain access to the server's files. The standard username is anonymous and requires no password. Your browser supplies this information automatically, so in most situations you don't have to worry about passwords and usernames. However, some FTP servers do not allow anonymous access. In these cases, either your browser prompts you for the username and the password, or you can supply a username and password within the URL using the format

```
ftp://username:password@server/path
```

where *username* and *password* are a username and password that the FTP server recognizes. It is generally not a good idea, however, to include usernames and passwords in URLs, as it can allow others to view your sensitive login information. It's better to let the browser send this information or to use a special program called an **FTP client**, which can encrypt or hide this information during transmission.

### Linking to a Local File

HTML is a very useful language for creating collections of linked documents. Many software developers have chosen to distribute their online help in the form of HTML files. The Web site for their help files then exist locally on the user's computer or network. If the Web site needs to reference local files (as opposed to files on the Internet or another wide area network), the URL needs to reflect this fact. The URL for a local file has the general form

```
file://server/path/filename
```
where *server* is the name of the local network server, *path* is the path on that server to
the file, and *filename* is the name of the file. If you’re accessing a file from your own
computer, the server name can be omitted and replaced by an extra slash (/). So, a file
from the documents/articles folder might have the URL:

```
file:///documents/articles/tips.htm
```

If the file is on a different disk within your computer, the hard drive letter would be
included in the URL as follows:

```
file://D:/documents/articles/tips.htm
```

Unlike the other URLs you’ve examined, the “file” scheme in this URL does not imply
any particular communication protocol; instead, the browser retrieves the document
using whatever method is the local standard for the type of file specified in the URL.

### Linking to an E-Mail Address

Many Web sites use e-mail to allow users to communicate with a site’s owner, sales rep-
resentative, or technical support staff. You can turn an e-mail address into a hypertext
link, so that a user can click the link starting an e-mail program and automatically inserting
the e-mail address into the “To” field of a new outgoing message. The URL for an
e-mail address follows the form

```
mailto:address
```

where *address* is the e-mail address. To create a hypertext link to the e-mail address
ghayward@camshots.com, you could use the following URL:

```
mailto:ghayward@camshots.com
```

Although the mailto protocol is not technically an approved communication protocol, it
is supported by almost every Web browser.

The mailto protocol also allows you to add information to the e-mail, including the
subject line and the text of the message body. To add this information to the link, you use
the form

```
mailto:address?header1=value1&header2=value2& ...
```

where *header1, header2,* etc. are different e-mail headers and *value1, value2,* and so on
are the values of the headers. So to create the e-mail message

```
TO: ghayward@camshots.com
SUBJECT: Test
BODY: This is a test message
```

you would use the following URL:

```
mailto:ghayward@camshots.com?Subject=Test&Body=This%20is%20a%20test%20message
```

Notice that the spaces in the message body “This is a test message” have been replaced
with %20 characters. This is necessary because URLs cannot contain blank spaces. To
preserve information about blank spaces, URLs use escape characters, which are sym-
bols that represent characters including nonprintable characters such as spaces, tabs, and
line feeds. Escape characters use many of the same values as HTML character codes,
though the syntax of escape characters is different. So, when the browser receives the following character string in a URL such as

This%20is%20a%20test%20message

it interprets the %20 escape character as a blank space and resolves the string as

This is a test message

Figure 2-37 lists some of the escape characters that can be used in any URL in place of printable or nonprintable characters.

<table>
<thead>
<tr>
<th>Escape Character Code</th>
<th>Character</th>
<th>Escape Character Code</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>%20</td>
<td>space</td>
<td>%5B</td>
<td>[</td>
</tr>
<tr>
<td>%0D%0A</td>
<td>new line</td>
<td>%5D</td>
<td>]</td>
</tr>
<tr>
<td>%3C</td>
<td>&lt;</td>
<td>%60</td>
<td>‘</td>
</tr>
<tr>
<td>%3E</td>
<td>&gt;</td>
<td>%3B</td>
<td>;</td>
</tr>
<tr>
<td>%23</td>
<td>#</td>
<td>%2F</td>
<td>/</td>
</tr>
<tr>
<td>%25</td>
<td>%</td>
<td>%3F</td>
<td>?</td>
</tr>
<tr>
<td>%7B</td>
<td>{</td>
<td>%3A</td>
<td>:</td>
</tr>
<tr>
<td>%7D</td>
<td>}</td>
<td>%40</td>
<td>@</td>
</tr>
<tr>
<td>%7C</td>
<td>\</td>
<td>%3D</td>
<td>=</td>
</tr>
<tr>
<td>%5C</td>
<td>^</td>
<td>%24</td>
<td>$</td>
</tr>
<tr>
<td>%7E</td>
<td>~</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To further explore how to convert an e-mail message into a URL, you can experiment with a demo page.

To view the e-mail demo:

1. Use your Web browser to open the demo_mailto.htm file from the tutorial.02\demo folder included with your Data Files.
2. Scroll down the page, and in the TO: input box, enter the e-mail address ghayward@camshots.com.
3. Type CAMshots Message in the SUBJECT input box.
4. Type the following in the BODY input box:
   This is a message generated by the CAMshots Web site for Gerry Hayward.
5. Click the Generate URL button to create the URL for this e-mail message.
   As shown in Figure 2-38, the demo page generates the URL for the e-mail message. All of the blank spaces in the mail message have been replaced with the %20 escape character.
6. Click the **Test the URL** button at the bottom of the page. As shown in Figure 2-39, the browser opens the user's e-mail program, with the e-mail fields already filled in, based on the text of the URL.

**Trouble?** Your e-mail window might look different depending on the e-mail program installed on your computer. If you do not have access to an e-mail program, you might not see any result or you might receive an error message after clicking the **Test the URL** button.
7. Close the message window without saving the message.

8. Continue experimenting with the demo page, exploring the effects of different e-mail messages on the URL text. Close the demo page when you are finished.

Gerry wants you to add to a link to his e-mail address on the CAMshots home page. This gives people who read his site the ability to contact him with additional questions or ideas.

To link to an e-mail address on Gerry’s home page:

1. Return to the home.htm file in your text editor.
2. Go to the first paragraph and locate the text “contact me.”
3. Mark “contact me” as a hypertext link using the following code, as shown in Figure 2-40:
   ```html
   <a href="mailto:ghayward@camshots.com?subject=CAMshots%20Message">contact me</a>
   ```

   Adding an e-mail link to the CAMshots home page
   
   Figure 2-40

4. Save your changes to the file.
5. Refresh the home.htm file in your browser. Verify that the text “contact me” in the opening paragraph now appears as a hypertext link.
6. Click contact me and verify that your e-mail program displays a message with ghayward@camshots.com as the recipient and CAMshots Message as the subject.
7. Close your message window without saving the message.
E-Mail Links and Spam

Use caution when adding e-mail links to your Web site. While it may make it more convenient for users to contact you, it also might make you more vulnerable to spam. **Spam** is unsolicited e-mail sent to large numbers of people, promoting products, services, and in some cases inappropriate Web sites. Spammers create their e-mail lists by scanning discussion groups, stealing Internet mailing lists, and using programs called **e-mail harvesters** to scan HTML code for the e-mail addresses contained in mailto URLs. Many Web developers have removed e-mail links from their Web sites in order to foil these harvesters, replacing the links with Web forms that submit e-mail requests to a secure server. If you need to include an e-mail address on your Web page, you can take a few steps to reduce your exposure to spammers:

- Replace the text of the e-mail addresses with inline images that are more difficult for e-mail harvesters to read.
- Write a program to scramble any e-mail addresses in the HTML code, unscrambling the e-mail address only when it is clicked by the user.
- Replace the characters of the e-mail address with escape characters. For example, you can replace the "@" symbol with the escape sequence %40.

There is no quick and easy solution to this problem. Fighting spammers is an ongoing battle, and they have proved very resourceful in overcoming some of the defenses people have created. As you develop your Web site, you should carefully consider how to handle e-mail addresses and review the most current methods for safeguarding that information.

Linking to Various Internet Resources

- The URL for a Web page has the form
  
  http://server/path/filename#id
  
  where **server** is the name of the Web server, **path** is the path to a file on that server, **filename** is the name of the file, and if necessary **id** is the name of an id or anchor within the file.

- The URL for an FTP site has the form
  
  ftp://server/path/filename
  
  where **server** is the name of the FTP server, **path** is the folder path, and **filename** is the name of the file.

- The URL for an e-mail address has the form
  
  mailto:address?header1=value1&header2=value2& ... 
  
  where **address** is the e-mail address; **header1**, **header2**, etc. are different e-mail headers; and **value1**, **value2**, and so on are the values of the headers.

- The URL to reference a local file has the form
  
  file://server/path/filename
  
  where **server** is the name of the local server or computer, **path** is the path to the file on that server, and **filename** is the name of the file. If you are accessing a file on your own computer, the server name is replaced by a third slash (/).

Working with Hypertext Attributes

HTML provides several attributes to control the behavior and appearance of your links. Gerry suggests that you study a few of these to see whether they would be effective in his Web site.
Opening a Secondary Window or Tab

By default, each page you open replaces the contents of the current page in the browser window. This means that when Gerry’s readers click on one of the four external links listed on the tips page, they leave the CAMshots Web site. To return to the Web site, users would have to click their browser’s Back button.

Gerry wants his Web site to stay open when a user clicks one of the links to the external Web sites. Most browsers allow users to open multiple browser windows or multiple tabs within the same browser window. Gerry suggests that links to external sites be opened in a second browser window or tab. He wants these external sites to be displayed in a second browser window or tab. This arrangement allows continual access to his Web site, even as users are browsing other sites.

To force a document to appear in a new window or tab, add the target attribute to the <a> tag. The general syntax is

```html
<a href="url" target="window">content</a>
```

where window is a name assigned to the new browser window or browser tab. The value you use for the target attribute doesn’t affect the appearance or content of the page being opened; the target simply identifies the different windows or tabs that are currently open. You can choose any name you wish for the target. If several links have the same target name, they all open in the same location, replacing the previous content. HTML also supports several special target names, described in Figure 2-41.

<table>
<thead>
<tr>
<th>Target Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>Opens the link in a new window or tab named target</td>
</tr>
<tr>
<td>_blank</td>
<td>Opens the link in a new, unnamed window or tab</td>
</tr>
<tr>
<td>_self</td>
<td>Opens the link in the current browser window or tab</td>
</tr>
</tbody>
</table>

Whether the new page is opened in a tab or in a browser window is determined by the browser settings. It cannot be set by the HTML code.

Opening a Link in a New Window or Tab

- To open a link in a new browser window or browser tab, add the attribute `target="window"` to the `<a>` tag, where window is a name assigned to the new browser window or tab.

Gerry suggests that all of the external links from his page be opened in a browser window or tab identified with the target name “new.”

To specify a link target:

1. Return to the tips.htm file in your text editor.
2. Scroll to the bottom of the file and locate the four links to the external Web sites.
3. Within each of the opening `<a>` tags, insert the following attribute, as shown in Figure 2-42.

   ```html
target="new"
   ```
Setting a target for a hyperlink

4. Save your changes to the file.

5. Refresh the tips.htm file in your browser. Click each of the four links to external Web sites and verify that each opens in the same new browser window or tab.

6. Close the secondary browser window or tab.

You should use the target attribute sparingly in your Web site. Creating secondary windows can clutter up the user’s desktop. Also, because the page is placed in a new window, users cannot use the Back button to return to the previous page in that window; they must click the browser’s program button or the tab for the original Web site. This confuses some users and annoys others. Many Web designers now advocate not using the target attribute at all, leaving the choice of opening a link in a new tab or window to the user. Note that the target attribute is not supported in strict XHTML-compliant code.

Creating a Tooltip

If you want to provide additional information about a link on your Web page, you can add a tooltip to the link. A tooltip is descriptive text that appears when a user positions the mouse pointer over a link. Figure 2-43 shows an example of a tooltip applied to one of Gerry’s links.

To create the tooltip, add the title attribute to the opening <a> tag in the form

```
<a href="url" title="text">content</a>
```
where text is the text that appears in the tooltip. To create the tooltip shown in Figure 2-43, you would enter the following HTML code:

```html
<a href="glossary.htm" title="Study photo terminology in the CAMshots glossary">
  Glossary
</a>
```

Note that because some browsers do not support this feature, you should not place crucial information in a tooltip.

### Creating a Semantic Link

The text of a hypertext link should always describe the type of document that will be called up by the link. You can also use the rel and rev attributes to add information about the link. The rel attribute describes the relation of the current document to the linked document. For example, in the link to the Glossary page, Gerry could insert the following rel attribute:

```html
<a href="glossary.htm" rel="glossary">Glossary</a>
```

The rev attribute describes the reverse relationship: how the linked document views the current document. For example, if you’re linking to the Glossary page from the home page, the reverse relation is “home” (because that is how the Glossary page views the home page). The HTML code would be:

```html
<a href="glossary.htm" rel="glossary" rev="home">Glossary</a>
```

Links containing the rel and rev attributes are called **semantic links** because the tag contains information about the relationship between the link and its destination. This information is not designed for the user, but for the browser. A browser could display all hypertext links marked having a rel value of glossary with a special icon. The browser could also collect all of the hypertext links within the Web page and place them within a customized toolbar. Few browsers currently take advantage of these attributes, but future browsers may do so.

Although rel and rev are not limited to a fixed set of attribute values, the specifications for HTML and XHTML include a proposed list of rel and rev names. Figure 2-44 shows some of these proposed relationship values.
Link relations for the rel and rev attributes

<table>
<thead>
<tr>
<th>Link Relation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alternate</td>
<td>A substitute version of the current document, perhaps in a different language or in a different medium</td>
</tr>
<tr>
<td>appendix</td>
<td>An appendix</td>
</tr>
<tr>
<td>bookmark</td>
<td>A bookmark in a collection of documents</td>
</tr>
<tr>
<td>chapter</td>
<td>A document serving as a chapter in a collection of documents</td>
</tr>
<tr>
<td>contents</td>
<td>A table of contents</td>
</tr>
<tr>
<td>copyright</td>
<td>A copyright statement</td>
</tr>
<tr>
<td>glossary</td>
<td>A glossary</td>
</tr>
<tr>
<td>help</td>
<td>A help document</td>
</tr>
<tr>
<td>index</td>
<td>An index</td>
</tr>
<tr>
<td>next</td>
<td>The next document in a linear sequence of documents</td>
</tr>
<tr>
<td>prev</td>
<td>The previous document in a linear sequence of documents</td>
</tr>
<tr>
<td>section</td>
<td>A document serving as a section in a collection of documents</td>
</tr>
<tr>
<td>start</td>
<td>The first document in a collection of documents</td>
</tr>
<tr>
<td>top</td>
<td>The Web site’s home page</td>
</tr>
<tr>
<td>stylesheet</td>
<td>An external style sheet</td>
</tr>
<tr>
<td>subsection</td>
<td>A document serving as a subsection in a collection of documents</td>
</tr>
</tbody>
</table>

At this point, Gerry decides against using the rel and rev attributes on his Web site. However, he’ll keep them in mind as an option as his Web site expands in size and complexity.

Using the Link Element

Another way to add a hypertext link to your document is to add a link element to the document’s head. Link elements are created using the one-sided tag

```html
<link href="url" rel="text" rev="text" target="window" />
```

where the href, rel, rev, and target attributes serve the same purpose as in the <a> tag. For example, to use the link element to create semantic links to the three pages of Gerry’s Web site, you could add the following link elements to the heading of each document:

```html
<link rel="top" href="home.htm" />
<link rel="help" href="tips.htm" />
<link rel="glossary" href="glossary.htm" />
```

Because they are placed within a document’s head, link elements do not appear as part of the Web page. Instead, if the browser supports them, link elements are displayed in a browser toolbar. Figure 2-45 shows how the three link elements described above would appear in the Opera’s Navigation toolbar. If you click an entry on the toolbar, the browser loads the referenced page.
The advantage of the link element is that it places the list of links outside of the Web page, freeing up page space for other content. Also, because the links appear in a browser toolbar, they are always easily accessible to users. Currently, Opera is one of the few browsers with built-in support for the link element. Third party software exists to provide this support for Internet Explorer and Firefox. Because no single list of relationship names is widely accepted, you must check with each browser's documentation to find out what relationship names it supports. Until link elements are embraced by more browsers, you should use them only if you duplicate that information elsewhere on the page.

**Working with Metadata**

Gerry is happy with the work you've done on the design for his CAMshots Web site. Now he wants to start working on getting the site noticed. When someone searches for “digital photography tips” or “camera buying guide,” will they find Gerry's Web site? There are thousands of photography sites on the Web. Gerry knows he needs to add a few extra touches to his home page to make it more likely that the site will be picked up by major search engines such as Yahoo! and Google.

Optimizing a Web site for search engines can be a long and involved process. For the best results, Web authors often turn to companies that specialize in making sites appear more prominently in search engines. CAMshots is a hobby site and Gerry does not want to invest any money in improving the site's visibility, but he would like to do a few simple things that would help.

**Using the Meta Element**

To be noticed on the Web, a site needs to include information about itself so the search engines can read it and add the site to their search indices. Information about the site is called metadata. You can add metadata to your Web pages by adding a meta element to the head section of the document. The syntax of the meta element is

```
<meta name="text" content="text" scheme="text" http-equiv="text" />
```

where the name attribute specifies the type of metadata, the content attribute stores the metadata value, the scheme attribute defines the metadata format, and the http-equiv attribute is used to attach metadata or commands to the communication stream between the Web server and the browser. There are three uses of the meta element:

- To store information about the document that can be read by the author, other users, or the Web server.
To control how the browser handles the document, including forcing the browser to automatically refresh the page at timed intervals.

To assist Web search engines in adding the document to their search index.

For example, the following meta element stores the name of the Web page's author:

```html
<meta name="author" content="Gerry Hayward" />
```

For search engines, you should include metadata describing the site and the topics it covers. This is done by adding a meta element containing the site description and another meta element with a list of keywords. The following two elements would summarize the CAMshots Web site for any search engines running on the Web:

```html
<meta name="description" content="CAMshots provides advice on digital cameras and photography" />
<meta name="keywords" content="photography, cameras, digital imaging" />
```

Figure 2-46 lists some other examples of metadata that you can use to describe your document.

<table>
<thead>
<tr>
<th>Meta Name</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
<td><code>&lt;meta name=&quot;author&quot; content=&quot;Gerry Hayward&quot; /&gt;</code></td>
<td>Supplies the name of the document author</td>
</tr>
<tr>
<td>classification</td>
<td><code>&lt;meta name=&quot;classification&quot; content=&quot;photography&quot; /&gt;</code></td>
<td>Classifies the document</td>
</tr>
<tr>
<td>copyright</td>
<td><code>&lt;meta name=&quot;copyright&quot; content=&quot;&amp;copy; 2011 CAMshots&quot; /&gt;</code></td>
<td>Provides a copyright statement</td>
</tr>
<tr>
<td>description</td>
<td><code>&lt;meta name=&quot;description&quot; content=&quot;Digital photography and advice&quot; /&gt;</code></td>
<td>Provides a description of the document</td>
</tr>
<tr>
<td>generator</td>
<td><code>&lt;meta name=&quot;generator&quot; content=&quot;Dreamweaver&quot; /&gt;</code></td>
<td>Indicates the name of the program that created the HTML code for the document</td>
</tr>
<tr>
<td>keywords</td>
<td><code>&lt;meta name=&quot;keywords&quot; content=&quot;photography, cameras, digital&quot; /&gt;</code></td>
<td>Provides a list of keywords describing the document</td>
</tr>
<tr>
<td>owner</td>
<td><code>&lt;meta name=&quot;owner&quot; content=&quot;CAMshots&quot; /&gt;</code></td>
<td>Indicates the owner of the document</td>
</tr>
<tr>
<td>rating</td>
<td><code>&lt;meta name=&quot;rating&quot; content=&quot;general&quot; /&gt;</code></td>
<td>Provides a rating of the document in terms of its suitability for minors</td>
</tr>
<tr>
<td>reply-to</td>
<td><code>&lt;meta name=&quot;reply-to&quot; content=&quot;ghayward@camshots.com (G. Hayward)&quot; /&gt;</code></td>
<td>Supplies a contact e-mail address and name for the document</td>
</tr>
</tbody>
</table>

In recent years, search engines have become more sophisticated in evaluating Web sites. In the process, the meta element has decreased in importance. However, it is still used by search engines when adding a site to their indexes. Because adding metadata requires very little effort, you should still include meta elements in your Web documents.
Working with Metadata

- To document the contents of your Web page, use the meta element
  <meta name="text" content="text" />
  where the name attribute specifies the type of metadata and the content attribute stores the metadata value.
- To add metadata or a command to the communication stream between the Web server and Web browser, use
  <meta http-equiv="text" content="text" />
  where the http-equiv attribute specifies the type of data or command attached to the communication stream and the content attribute specifies the data value or command.

Having discussed metadata issues with Gerry, he asks that you include a few meta elements to describe his new site.

To add metadata to Gerry's document:
1. Return to the home.htm file in your text editor.
2. Directly below the opening <head> tag, insert the following meta elements, as shown in Figure 2-47:
   
   ```html
   <meta name="author" content="your name" />
   <meta name="description" content="A site for sharing information on digital photography and cameras" />
   <meta name="keywords" content="photography, cameras, digital imaging" />
   ```

3. Close the file, saving your changes.

Applying Metadata to the Communication Stream

Describing your document is not the only use of the meta element. As you learned earlier, servers transmit Web pages using a communication protocol called HTTP. You can add information and commands to this communication stream with the meta element's http-equiv attribute. One common use of the http-equiv attribute is to force the browser to refresh the Web page at timed intervals, which is useful for Web sites that publish scoreboards or stock tickers. For example, to automatically refresh the Web page every 60 seconds, you would apply the following meta element:

```
<meta http-equiv="refresh" content="60" />
```

Another use of the meta element is to redirect the browser from the current document to a new document. This might prove useful to Gerry someday if he changes the URL of his
site’s home page. As his readers get accustomed to the new Web address, he can keep the old address online, automatically redirecting readers to the new site. The meta element to perform an automatic redirect has the general form

```html
<meta http-equiv="refresh" content="sec;url=url" />
```

where sec is the time in seconds before the browser redirects the user and url is the URL of the new site. To redirect users after five seconds to the Web page at http://www.camshots.com, you could enter the following meta element:

```html
<meta http-equiv="refresh" content="5;url=www.camshots.com" />
```

Another use of the http-equiv attribute is to specify the character set used by the document. (For a discussion of character sets, see Tutorial 1.) This is particularly useful for international documents in which the browser might need to know the character set being used to correctly interpret the document. The syntax to specify the character set for an HTML document is

```html
<meta http-equiv="Content-Type" content="text/html;charset=char-set" />
```

where char-set is the character set used by the document. So to indicate that the browser uses the ISO-8859-1 character set, you would include the following meta element in the document’s header:

```html
<meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-1" />
```

With the Web expanding its international presence, many Web developers advocate always including metadata about the character set so there is no ambiguity in the interpretation of the character encoding used in the document.

At this point, Gerry does not need to use the meta element to send data or commands through the HTTP communication protocol. However, he will keep this option in mind if moves the site to a new address.

Gerry is happy with the Web site you’ve started. He’ll continue to work on the site and will come back to you for more assistance as he adds new pages and elements. For now you can close any open files or applications used to create the site.

---

### Review | Session 2.3 Quick Check

1. What are the five parts of a URL?
2. Specify the code to link the text “White House” to the URL http://www.whitehouse.gov, with the destination document displayed in a new unnamed browser window.
3. Specify the code to link the text “Washington” to the FTP server at ftp.uwash.edu.
4. Specify the code to link the text “President” to the e-mail address president@whitehouse.gov.
5. What attribute would you add to a hypertext link to display the popup title “Tour the White House”?
6. What attribute would you add to a link specifying that the destination is the next page in a linear sequence of documents?
7. Specify the code to add the description “United States Office of the President” as metadata to a document.
8. Specify the code to automatically refresh the document every 5 minutes.
In this tutorial you explored some of the issues involved in creating a Web site with several linked pages. The first session began with an overview of storyboarding as a tool for designing and maintaining complex Web site structures. The session then turned to creating a simple Web site involving three Web pages linked together with the `<a>` tag element. The second session focused on creating links to locations within documents, first examining how to mark a location by using the id attribute and the anchor element. It then covered how to create links to these locations from within the same document and from within another document. The second session concluded by examining how to use inline images and image maps to create links to several documents. The third session expanded the discussion of hypertext by showing how to create links to sites on the World Wide Web and non-Web locations, including FTP sites and e-mail addresses. The third session then examined how to set different hypertext attributes to control how the browser displays and reacts to hypertext links. The session and the tutorial concluded by discussing the uses of the meta element for conveying information to Web search engines.

**Key Terms**

- absolute path
- augmented linear structure
- client-side image map
- domain name
- e-mail harvester
- escape characters
- extension
- File Transfer Protocol
- FTP
- FTP server
- hierarchical structure
- home page
- hotspot
- HTTP
- Hypertext Transfer Protocol
- image map
- linear structure
- metadata
- mixed structure
- protected structure
- protocol
- relative path
- semantic link
- server-side image map
- sibling folder
- site index
- spam
- storyboard
- tooltip
- Uniform Resource Locator
- URL
Practice | Review Assignments


Gerry has been working on the CAMshots Web site for a while. During that time, the site has grown in popularity with amateur photographers. Gerry wants to host a monthly photo contest to highlight the work of his colleagues. Each month Gerry will pick the three best photos from different photo categories. He’s asked for your help in creating the collection of Web pages highlighting the winning entries. Gerry has already created four pages. The first page contains information about the photo contest; the next three pages contain the winning entries for child photos, scenic photos, and flower photos. Although Gerry has already entered much of the page content, he needs you to work on creating the links between and within each page. Figure 2-48 shows a preview of the photo contest’s home page.
Complete the following:

1. Use your text editor to open the `contesttxt.htm`, `childtxt.htm`, `scenictxt.htm`, and `flowertext.htm` files from the tutorial.02\review folder included with your Data Files. Enter your name and the date within each file, and then save them as `contest.htm`, `child.htm`, `scenic.htm`, and `flower.htm`, respectively, in the same folder.

2. Go to the `child.htm` file in your text editor. Locate the inline image within the first div element at the top of the file. Directly below the inline image insert an image map with the following properties:
   - Set the id and name of the image map to `contestmap`.
   - Add a polygonal hotspot pointing to the `child.htm` file containing the points (457, 84), (474, 63), (549, 63), and (566, 84). Specify “Child Photos” as the alternate text for the hotspot.
   - Add a polygonal hotspot pointing to the `flower.htm` file containing the points (554, 84), (571, 63), (646, 63), and (663, 84). Specify “Flower Photos” as the alternate text for the hotspot.
   - Add a polygonal hotspot pointing to the `scenic.htm` file containing the points (651, 84), (668, 63), (743, 63), and (760, 84). Specify “Scenic Photos” as the alternate text for the hotspot.
   - Add a circular hotspot pointing to the `contest.htm` file centered at the point (82, 82) and having a radius of 78 pixels. Specify “Contest Results” as the alternate text for the hotspot.

3. Apply the `contestmap` image map to the logo image at the top of the page. Set the width of the border to 0.

4. Locate the three h2 elements naming the three child photo winners. Assign the h2 elements the ids `child1`, `child2`, and `child3`, respectively.

5. Save your changes to the `child.htm` file.

6. Go to the `scenic.htm` file in your text editor. Repeat Steps 2 and 3 for the logo image at the top of the page.

7. Assign the ids `scenic1`, `scenic2`, and `scenic3` to the three h2 elements located farther down in the file. Save your changes to the document.

8. Go to the `flower.htm` file in your text editor and repeat the same edits you applied to the `child.htm` and `scenic.htm` files. Assign the ids `flower1` through `flower3` to the three h2 headings located at the bottom of the document. Save your changes.

9. Go to the `contest.htm` file in your text editor. Repeat Steps 2 and 3 for the logo image at the top of the page.

10. Scroll to the definition list at the bottom of the file. Mark the definition term “Child Photos” as a hypertext link pointing to the `child.htm` file. Mark the definition term “Flower Photos” as a link to the `flower.htm` file. Mark the term “Scenic Photos” as a link to the `scenic.htm` file.

11. Following each definition term is a definition description containing three thumbnail images of the winning photos. Mark the nine thumbnail images as hypertext links pointing to the larger images (contained in the `child.htm`, `flower.htm`, and `scenic.htm` files). For example, mark the first child photo (thumb1.jpg) as a hypertext link pointing to the h2 element with the id `child1` in the `child.htm` file. Set the border width of each of the nine thumbnail images to 0.

12. Scroll up and locate the fourth paragraph. Mark the text “Gerry Hayward” as a hypertext link to an e-mail message sent to ghayward@camshots.com with the subject line “Photo Contest.”

Copyright 2009 Cengage Learning, Inc. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part.
13. Go to the sixth paragraph and mark the text “BetterPhoto.com” as a hypertext link pointing to the URL http://www.betterphoto.com. Set the attribute of the link so that it opens in a new browser window or tab.

14. Save your changes to the contest.htm file.

15. Open contest.htm in your Web browser. Verify that the e-mail link opens a new mail message window with the subject line “Photo Contest.” Verify that the link to BetterPhoto.com opens that Web site in a new browser window or tab. Verify that the three links to the photo pages are connected to the child.htm, scenic.htm, and flower.htm files. Finally, click each of the nine thumbnail images at the bottom of the page and verify that they connect to the larger image of the photo.

16. Open child.htm in your Web browser. Verify that the Scenic Photos link at the top of the page is connected to the scenic.htm file. Navigate forward and backward through the three photo pages by clicking the links at the top of each page. Verify that on each page you can return to the contest page by clicking the contest logo.

17. Submit your completed files to your instructor.

Apply | Case Problem 1

Apply your knowledge of hypertext links to create a directory of universities and colleges.

Data Files needed for this Case Problem: colleges.txt, highered.jpg, and uwlisttxt.htm

HigherEd Adella Coronel is a guidance counselor for Eagle High School in Waunakee, Wisconsin. She wants to take her interest in helping students choose colleges to the Web by starting a Web site called HigherEd. She’s come to you for help in creating the site. The first page she wants to create is a simple directory of Wisconsin colleges and universities. She’s created the list of schools, but has not yet marked the entries in the list as hypertext links. Also, the list is very long, so she has broken it down into three categories: private colleges and universities, technical colleges, and public universities. Because of the length of the page, she wants to include hypertext links that allow students to jump down to a specific college category. Figure 2-49 shows a preview of the page you’ll create for Adella.
Complete the following:

1. In your text editor, open the uwlisttxt.htm file from the tutorial.02\case1 folder included with your Data Files. Enter your name and the date in the comment section of the file. Save the file as uwlist.htm in the same folder.

2. Mark each of the school entries on the page as a hypertext link. Use the URLs provided in the colleges.txt file. (Hint: Use the copy and paste feature of your text editor to efficiently copy and paste the URL text.)

3. Adella wants the links to the school Web sites to appear in a new tab or window. Because there are so many links on the page, add a base element to the document header specifying that all links will open by default in a new browser window or tab named “collegeWin.”

4. Add the id names “private,” “technical,” and “public” to the three h2 headings that categorize the list of schools.

5. Create hyperlinks from the entries in the category list at the top of the page to the three headings.

6. For each of the hypertext links you marked in Step 5, set the link to open in the current browser window and not in a new browser window or tab.

7. Save your changes to the file.

8. Open uwlist.htm in your Web browser and verify that the school links all open in the same browser window or tab and that the links within the document to the different school categories bring the user to those locations on the page but not in a new window tab.

9. Submit your completed files to your instructor.

**Case Problem 2**

Apply your knowledge of HTML to create a slide show Web site.


Lakewood School  
Tasha Juroszek is a forensics teacher at Lakewood School, a small private school in Moultrie, Georgia. Tasha has just finished directing her students in *Fiddler on the Roof Jr.* and wants to place a slide show of the performances on the Web. She has already designed the layout and content of the pages, but needs help to finish the slide show. She has asked you to add hypertext links between the slide pages and the site’s home page. Figure 2-50 shows a preview of one of the slide pages on the Web site.
Complete the following:

1. Use your text editor to open the `hometxt.htm`, and `slide1txt.htm` through `slide6txt.htm` files from the tutorial.02\case2 folder included with your Data Files. Enter your name and the date in the comment section of each file. Save the files as `home.htm` and `slide1.htm` through `slide6.htm`, respectively.

2. Return to the `slide1.htm` file in your text editor. At the top of the page are five buttons used to navigate through the slide show. Locate the inline image for the home button (home.jpg) and mark it as a hypertext link pointing to the home.htm file.
3. There are six slides in Tasha’s slide show. Mark the start button as a hypertext link pointing to the slide1.htm file. Mark the end button as a link to the slide6.htm file. Link the back button to slide1.htm, the first slide in the show. Link the forward button to the slide2.htm file.

4. Directly below the slide show buttons are thumbnail images of the six slides. Link each thumbnail image to its slide page.

5. Set the border width of each linked image to 0, except the thumbnail image for slide1. Set the border width of that thumbnail to 5.

6. Save your changes to the file.

7. Repeat Steps 2 through 6 for the five remaining slide pages. Within each page, set the navigation buttons to go back and forth through the slide show. For the slide6.htm file, the forward button should point to the slide6.htm file since it is the last slide in the show. The border width of each linked image should be set to 0 except the border width of the current slide, which should be set to 5.

8. Go to the home.htm file in your text editor. Go to the second paragraph and mark the text “slide show” as a hypertext link pointing to the slide1.htm file.

9. Go to the end of the second paragraph and mark the phrase “contact me” as a hypertext link pointing to the following e-mail message:
   TO: tashajur@lakewood.edu
   SUBJECT: Photo CD
   BODY: Please send me a copy of the photos.

10. Save your changes to the file.

11. Load the home.htm file in your Web browser. Test the links in the Web site and verify that they work correctly.

12. Submit your completed files to your instructor.

**Challenge**

**Case Problem 3**

Broaden your knowledge of HTML by exploring how to use anchors and pop-up titles in a Web site for a health club.

Data Files needed for this Case Problem: classtxt.htm, diamond.jpg, hometxt.htm, indextxt.htm, and memtxt.htm

**Diamond Health Club, Inc.** You work for Diamond Health Club, a health club in Boise, Idaho that has been serving active families for 25 years. The director, Karen Padilla, has asked you to help work on their Web site. The site contains three pages: the home page describing the club, a page listing classes offered, and a page describing the various membership options. You need to add links within the main page and add other links connecting the pages. Because this Web site will need to support older browsers, you will have to use the anchor tag to mark specific locations in the three documents. Karen would also like you to create pop-up titles for some of the links in the site to supply additional information about the links to the users.

Finally, this new site will replace the old company Web site. Karen wants to keep the old Web site address and redirect users automatically to the new home page. She wants you to insert the code required to do this.

Figure 2-51 shows a preview of the completed home page.
Complete the following:

1. Use your text editor to open the hometxt.htm, indextxt.htm, classtxt.htm, and memtxt.htm files from the tutorial.02\case3 folder included with your Data Files. Enter your name and the date in the comment section of each file. Save the files as home.htm, index.htm, classes.htm, and members.htm respectively.

2. Go to the index.htm file. Use the <a> tag to add the anchor names fac, hours, and staff to the h3 headings “Facilities,” “Hours,” and “For More Information, E-mail our Staff.”

3. Scroll up to the top of the file. Below the logo image at the top of the page, add an image map with the following properties:
   - Give the image map a name and id of diamondmap.
   - Create a rectangular hotspot with the coordinates (225, 7) and (333, 40). Point the hotspot to the classes.htm file with the alternate text “Classes.” Add the tooltip “View our classes.”
• Create a rectangular hotspot with the coordinates (258, 44) and (437, 82). Point the hotspot to the members.htm file with the alternate text “Memberships.” Add the tooltip “View our membership options.”

• Create a default hotspot for the inline image. (Hint: the image is 548 pixels wide and 150 pixels tall.) Point the default hotspot to the index.htm file with the alternate text “Home Page.” Add the tooltip “Return to the Home Page.”

4. Apply the diamondmap hotspot to the logo image. Remove the border around the inline image.

5. In the list at the top of the page, mark “Facilities” as a link pointing to the fac anchor within the index.htm document. Mark “Staff” as a link pointing to the staff anchor within the index.htm file. Mark “Hours” as a link pointing to the hours anchor within the index.htm file.

6. Add the tooltip “Learn more about our facilities” to the Facilities link. Add the tooltip “Meet the DHC staff” to the Staff link. Add the tooltip “View the DHC hours of operation” to the Hours link.

7. Go to the staff list at the bottom of the page. Format each name as a link that points to the individual’s e-mail address. The e-mail addresses are:
   Ty Stoven: tstoven@dmond-health.com
   Yosef Dolen: ydolen@dmond-health.com
   Sue Myafin: smyafin@dmond-health.com
   James Michel: jmichel@dmond-health.com
   Ron Chi: rchi@dmond-health.com
   Marcia Lopez: mlopez@dmond-health.com

8. Save your changes to the file.

9. Go to the members.htm file in your text editor and repeat Steps 3 through 6.

10. Use the <a> tag to add anchors named “ind” to the “Individual memberships” h3 heading, “fam” to the “Family memberships” h3 heading, and “temp” to the “Temporary memberships” h3 heading.

11. Format the phrase “e-mail Ron Chi” in the first paragraph as a link pointing to Ron Chi’s e-mail address. Save your changes to the file.

12. Go to the classes.htm file in your text editor and repeat Steps 3 through 6 for the entries at the top of that page.

13. Use the <a> tag to add the following anchors to h3 headings in the file: “senior” for “Senior Classes,” “adult” for “Adult Classes,” “teen” for “Teen Classes,” and “child” for “Children’s Classes.”

14. Format the phrase “e-mail Marcia Lopez” in the first paragraph as a link pointing to Marcia Lopez’s e-mail address. Save your changes to the file.

15. Return to the index.htm file in your text editor. Within the first paragraph, link the word “children” to the child anchor in the classes.htm file. Link the word “teens” to the teen anchor in the classes.htm file. Link the word “adults” to the adult anchor in classes.htm. Finally, link “seniors” to the senior anchor in classes.htm.

16. Within the second paragraph of index.htm, link the word “individual” to the ind anchor in the members.htm file. Link the word “family” to the fam anchor in members.htm. Finally, link the first occurrence of the word “temporary” to the temp anchor in members.htm.
17. Go to the head section of the document and add the following metadata directly below the opening <head> tag:
   - The description: “The Diamond Health Club is your year-round source for fun family health.”
   - The keywords: health club, exercise, family, seattle
18. Save your changes to the file.
19. Go to the home.htm file in your text editor. Within the head section, insert a meta element to redirect the browser to the index.htm file after a 5 second delay.
20. Mark the phrase “this link to our new Web site” as a hypertext link pointing to the index.htm file. Save your changes to the file.
21. Open the home.htm file in your Web browser. Verify that the browser loads the index.htm file after a 5 second delay.
22. Once the index.htm file is loaded, verify that all of your links work correctly, including the links that point to sections within documents and the links within the image map. Verify that tooltips appear as you move your mouse pointer over the links at the top of each page. (Note: Internet Explorer does not currently support tooltips found within image map hotspots.)
23. Submit your completed files to your instructor.

Create | Case Problem 4

Test your knowledge of HTML and use your creativity to design a Web site documenting a Shakespeare play.

Data Files needed for this Case Problem: characters.txt, notes.txt, tempest.jpg, and tempest.txt

Mansfield Classical Theatre  Steve Karls continues to work as the director of Mansfield Classical Theatre in Mansfield, Ohio. The next production he plans to direct is The Tempest. Steve wants to put the text of this play on the Web, but he also wants to augment the dialog of the play with notes and commentary. However, he doesn’t want his commentary to get in the way of a straight-through reading of the text, so he has hit on the idea of linking his commentary to key phrases in the dialog. Steve has created text files containing an excerpt from The Tempest as well as his commentary and other supporting documents. He would like you to take his raw material and create a collection of linked pages.

Complete the following:

1. Create HTML files named tempest.htm, commentary.htm, and cast.htm and save them in the tutorial.02\case4 folder included with your Data Files. Add comment tags to the head section of each document containing your name and the date. Add an appropriate page title to each document.
2. Using the contents of the tempest.txt, notes.txt, and characters.txt text files, create the body of the three Web pages in Steve’s Web site. The design of these pages is left to your imagination and skill. Make the pages easy to read and visually interesting. You can supplement the material on the page with appropriate material you find on your own.
3. Use the tempest.jpg file as a logo for the page. Create an image map from the logo pointing to the tempest.htm, commentary.htm, and cast.htm files. The three rectangular boxes on the logo have the following coordinates for their upper-left and lower-right corners:
   - The Play: (228, 139) (345, 173)
   - Commentary: (359, 139) (508, 173)
   - The Cast: (520, 139) (638, 173)
Use this image map in all three of the Web pages from this Web site.

Copyright 2009 Cengage Learning, Inc. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part.
4. Create links between the dialog on the play page and the notes on the commentary page. The notes contain line numbers to aid you in linking each line of dialog to the appropriate note.

5. Create a link between the first appearances of each character’s name from the tempest.htm page with the character’s description on the cast.htm page.

6. Include a link to Steve Karl’s e-mail address on the tempest.htm page. Steve’s e-mail address is stevekarls@mansfieldct.com. E-mail sent to Steve’s account from this Web page should have the subject line “Comments on the Tempest.”

7. Add appropriate meta elements to each of the three pages documenting the page’s contents and purpose.

8. Search the Web for sites that would provide additional material about the play. Add links to these pages on the tempest.htm page. The links should open in a new browser window or tab.

9. Submit your completed files to your instructor.

Session 2.1

1. Storyboarding is the process of diagramming a series of related Web pages, taking care to identify all links among the various pages. Storyboarding is an important tool in creating Web sites that are easy to navigate and understand.

2. A linear structure is one in which Web pages are linked from one to another in a direct chain. Users can go to the previous page or the next page in the chain, but not to a page in a different section of the chain. A hierarchical structure is one in which Web pages are linked from general to specific topics. Users can move up and down the hierarchy tree.

3. <a href="sports.htm">Sports Info</a>

4. An absolute path indicates the location of the file based on its placement in the computer. A relative path indicates the location of the file relative to the location of the current document.

5. glossary.htm
   ../tips/tips1.htm
   ../tips/tips2.htm
   ../index.htm

6. The base element specifies the default location that the browser should use to resolve all relative paths.

Session 2.2

1. <h2 id="faq">CAMshots FAQ</h2>

2. <a href="#faq">Read our FAQ</a>

3. <a href="help.htm#faq">Read our FAQ</a>

4. <h2><a name="faq">CAMshots FAQ</a></h2>

5. Anchors are supported by older browsers. Some older browsers do not support using the id attribute to mark a location in a document. However, use of anchor tags has been deprecated, so it is not supported in strict applications of XHTML. Also, because it is deprecated, use of the anchor tag may be phased out in future browser releases.
6. `<map name="CAMmap" id="CAMmap">
   <area type="circle" coords="50, 75, 40" href="faq.htm" />
</map>
7. `<img src="logo.jpg" alt="CAMshots" usemap="#CAMmap"/>
8. `style="border-width: 0"

**Session 2.3**

1. The protocol, the hostname, the folder name, the filename, and the anchor name or id.
2. `<a href="http://www.whitehouse.gov target="_blank">White House</a>
3. `<a href="ftp://ftp.uwash.edu">Washington</a>
4. `<a href="mailto:president@whitehouse.gov">President</a>
5. `title="Tour the White House"
6. `rel="next"
7. `<meta name="description" content="United States Office of the President"/>
8. `<meta http-equiv="refresh" content="300"/>

**Ending Data Files**

glossary.htm  
home.htm  
tips.htm  
+ 3 graphic files

child.htm  
contest.htm  
flower.htm  
scenic.htm  
+ 22 graphic files

uwlist.htm  
+ 1 graphic file

home.htm  
slide1.htm  
slide2.htm  
slide3.htm  
slide4.htm  
slide5.htm  
slide6.htm  
+ 18 graphic files

home.htm  
index.htm  
members.htm  
+ 1 graphic file

classes.htm  
cast.htm

commentary.htm  
tempest.htm  
+ 1 graphic file
Reality Check

The Web has become an important medium for advertising products and one’s self. By making your resume available online, you can quickly get prospective employers information they need to make a hiring decision. There are many sites that will assist you in writing and posting your resume. They will also, for a fee, present your online resume to employers in your chosen field. Assuming you don’t want to pay to use such a site, you can also create your own Web site containing your employment history and talents. In this exercise, you’ll use the skills and tasks you learned in Tutorials 1 and 2 to design your own Web site and create an online resume.

1. Collect material on yourself that would be useful in an online resume. You should include material for a page on your employment history, talents and special interests, a general biography, and a summary of the main points of your resume.
2. Create a storyboard outlining the pages on your Web site. Clearly indicate the links between the pages. Make sure that your site is easy to navigate no matter which page the user starts on.
3. Collect or create graphical image files to make your site interesting to the viewer. If you obtain graphics from the Web, be sure to follow all copyright restrictions on the material.
4. Start designing your site’s home page. It should include an interesting and helpful logo. The home page should be brief and to the point, summarizing the main features of your resume. Its height should not be greater than two screens.
5. Add other pages containing more detailed information. Each page should have a basic theme and topic. The pages should follow a unified theme and design.
6. Use boldface fonts and italics to highlight important ideas. Do not overuse these page elements; doing so can distract from your page’s readability rather than enhancing it.
7. Use numbered and bulleted lists to list the main points in your resume.
8. Use block quotes to highlight recommendations from colleagues and former employers.
9. Use horizontal rules to divide longer pages into topical sections.
10. If there are sites on the Web that would be relevant to your online resume (such as the Web sites of former or current employers), include links to those sites.
11. Include a link to your e-mail address. Write the e-mail address link so that it automatically adds an appropriate subject line to the e-mail message.
12. Save your completed Web site and present it to your instructor.