Tutorial 2

Objectives

Session 2.1
- Storyboard various Web site structures
- Create links documents in a Web site
- Understand relative and absolute folder paths
- Work with the base element

ession 2.2
- Mark a location with the id attribute
- Create a link to an id
- Mark an image as a link
- Create an image map
- Remove an image border

ssion 2.3
- Understand URLs
- Link to a site on he Web
- Link to an e-mail ddress
- Work with hypertext ttributes
- Work with metadata

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.

rting Data Files

Tutorial
glosstxt.htm
hometxt.htm
tipstxt.htm
+ 3 graphic files

Demo
demo_mailto.htm
+ 3 graphic files

Review
childtxt.htm
contesttxt.htm
flowertxt.htm
scenictxt.htm
+ 22 graphic files

Case1
colleges.txt
uwlisttxt.htm
+ 1 graphic file

Case2
hometxt.htm
slide1txt.htm
slide2txt.htm
slide3txt.htm
slide4txt.htm
slide5txt.htm
slide6txt.htm
+ 18 graphic files

Case3
classtxt.htm
hometxt.htm
indextxt.htm
memtxt.htm
+ 1 graphic file

Case4
characters.txt
notes.txt
tempest.txt
+ 1 graphic file

Session 2.1

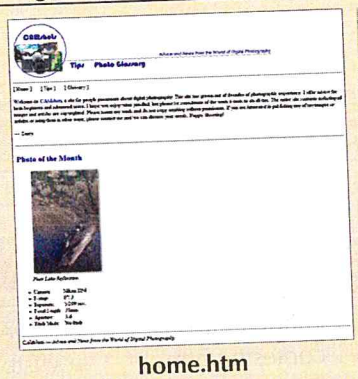
Exploring Web Site Structures

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.

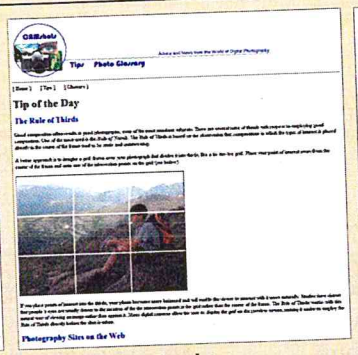
To view Gerry's Web pages:

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.

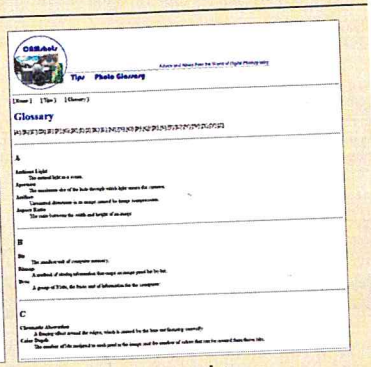
Figure 2-1 Pages in the CAMshots Web site



home.htm
the CAMshots home page



tips.htm
the CAMshots tip of the day



glossary.htm
a partial glossary of photography terminology

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 able to navigate the site without getting lost or missing important information.

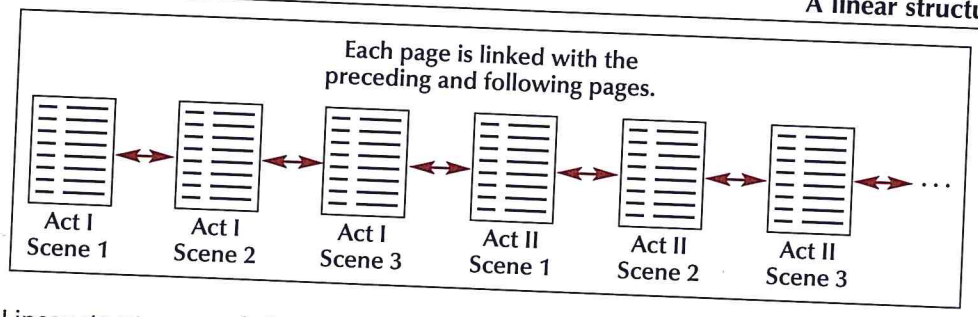
Every Web site starts with a single **home page** that acts as a focal point for the Web site. It is usually the first page that users see. Starting from the home page, you add the links to other

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.

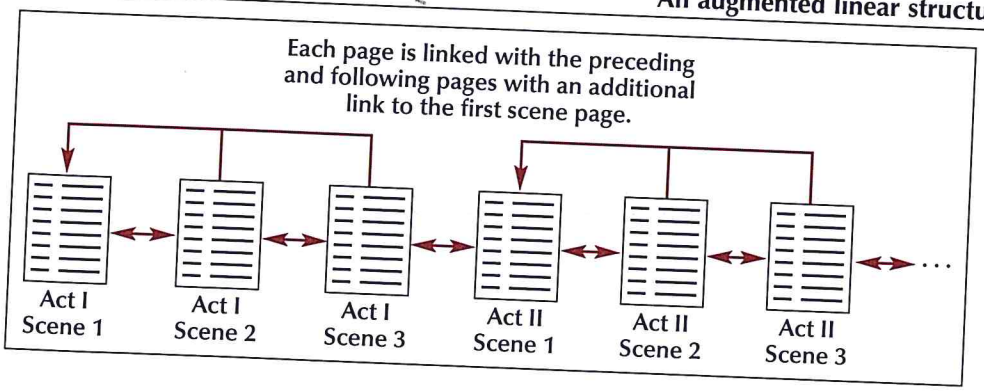
A linear structure Figure 2-2



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.

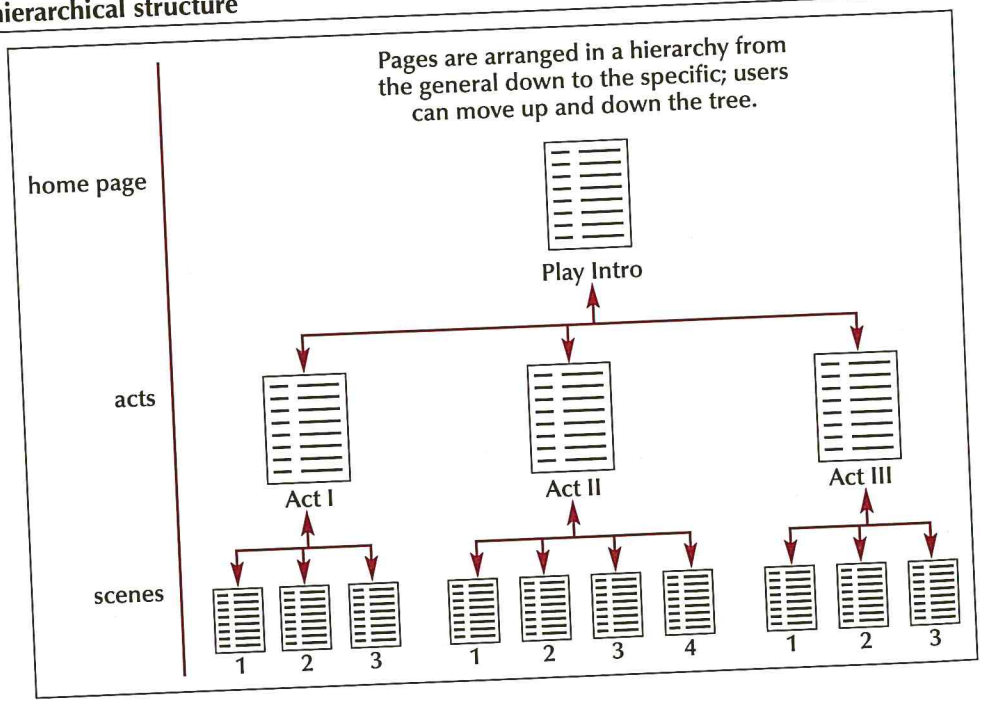
An augmented linear structure Figure 2-3



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.

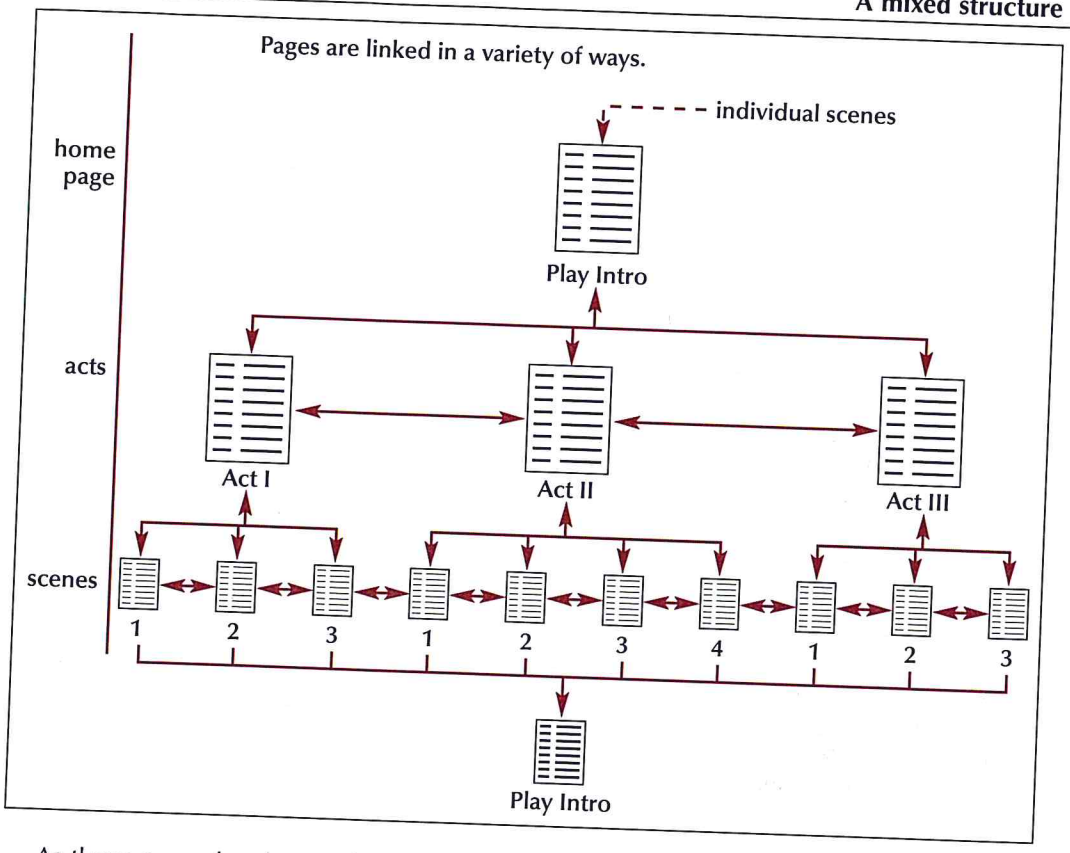
Figure 2-4 **A hierarchical structure**



Mixed Structures

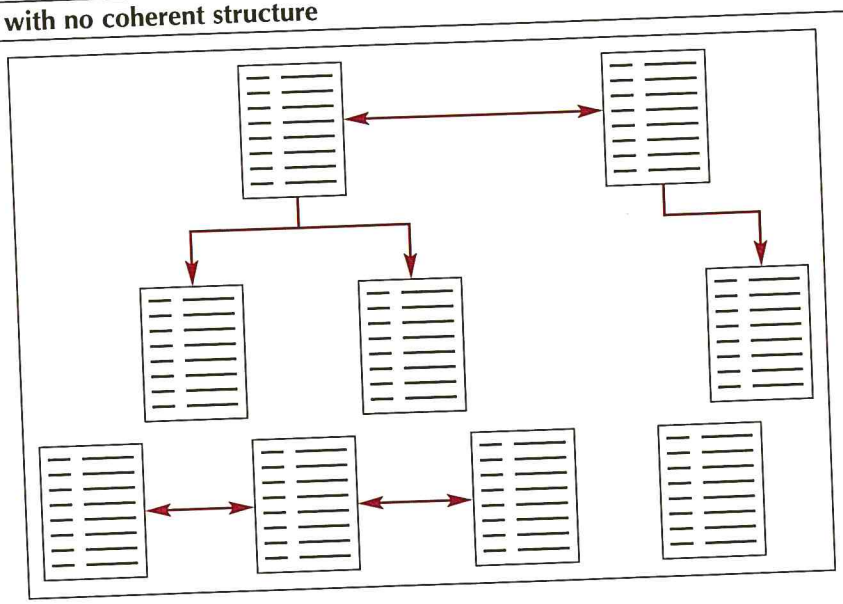
With larger and more complex Web sites, you often need to use a combination of structures. Figure 2-5 shows the online play using a mixture of the three main structures. The overall form is hierarchical, as users can move from a general introduction down to individual scenes; however, users can also 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.

A mixed structure **Figure 2-5**



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.

Figure 2-6 Web site with no coherent structure

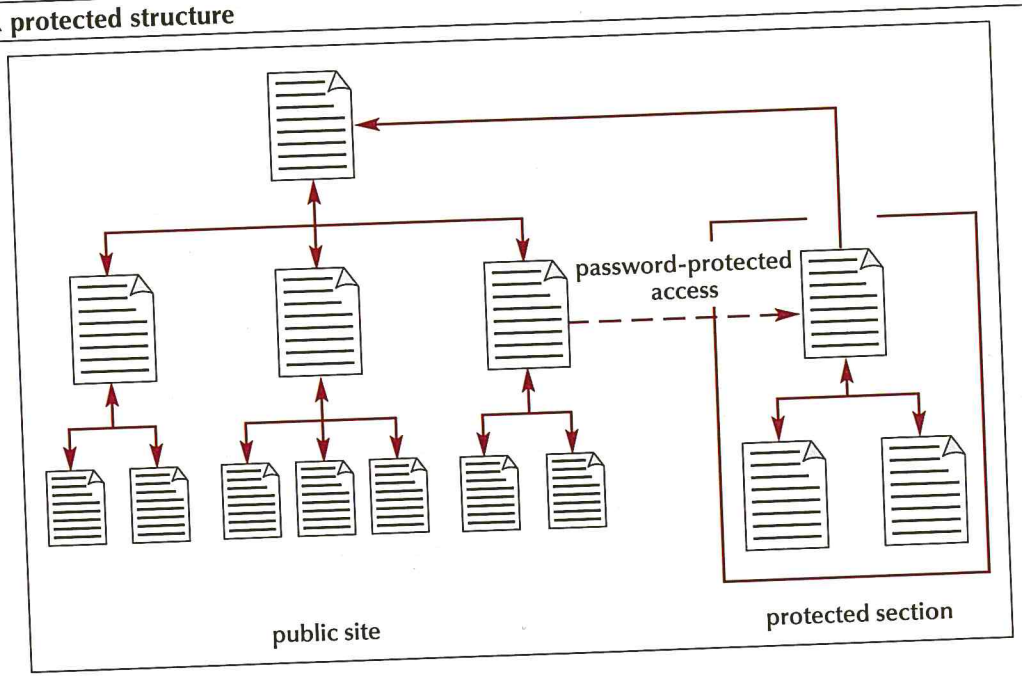


This confusing structure makes it difficult for users to grasp the site's contents and scope. The user might not even be aware of the presence of some pages because there are no connecting links, and some of the links only point in one direction. The Web is a competitive place; studies have shown that 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.

Figure 2-7 A protected structure

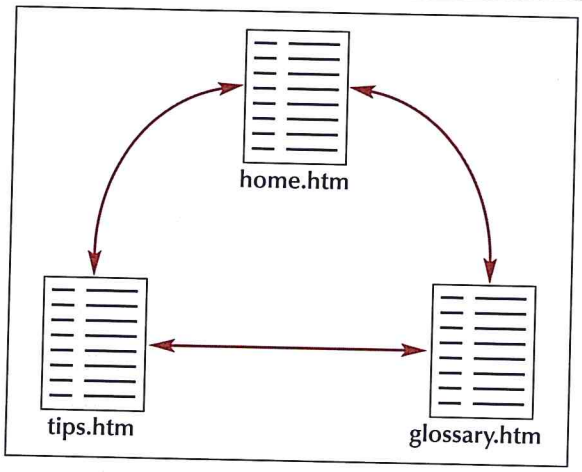


Storyboarding a protected structure is particularly important to ensure that no unmonitored "back doors" to the protected area are allowed in the site design.

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.

Storyboard for the CAMshots Web site Figure 2-8



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.

Tip

Keep your filenames short so that users are less apt to make a typing error when accessing your Web site.

Reference Window | Marking a Hypertext Link

- To mark content as hypertext, use
  `<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:

```
<a href="home.htm">Home</a>
```

3. Mark the text "Tips" as a hypertext link using the following code:

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

4. Mark the text "Glossary" as a hypertext link as follows:

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

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

Figure 2-9 Marking hypertext links in the home.htm file

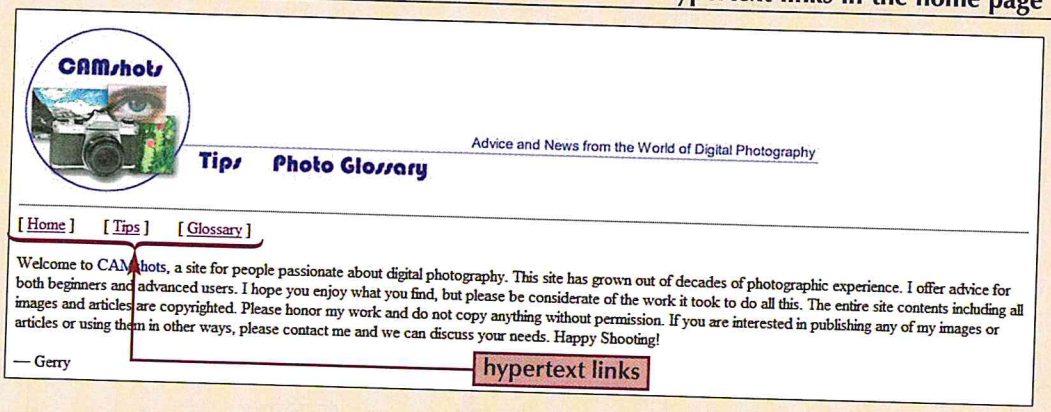
```
<body>
   <div>
      <img src="camshots.jpg" alt="CAMshots" />
   </div>
   <hr />

   <div>
      [ <a href="home.htm">Home</a> ]
           
      [ <a href="tips.htm">Tips</a>  ]
           
      [ <a href="glossary.htm">Glossary</a>  ]
   </div>
```

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.

Hypertext links in the home page Figure 2-10



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.

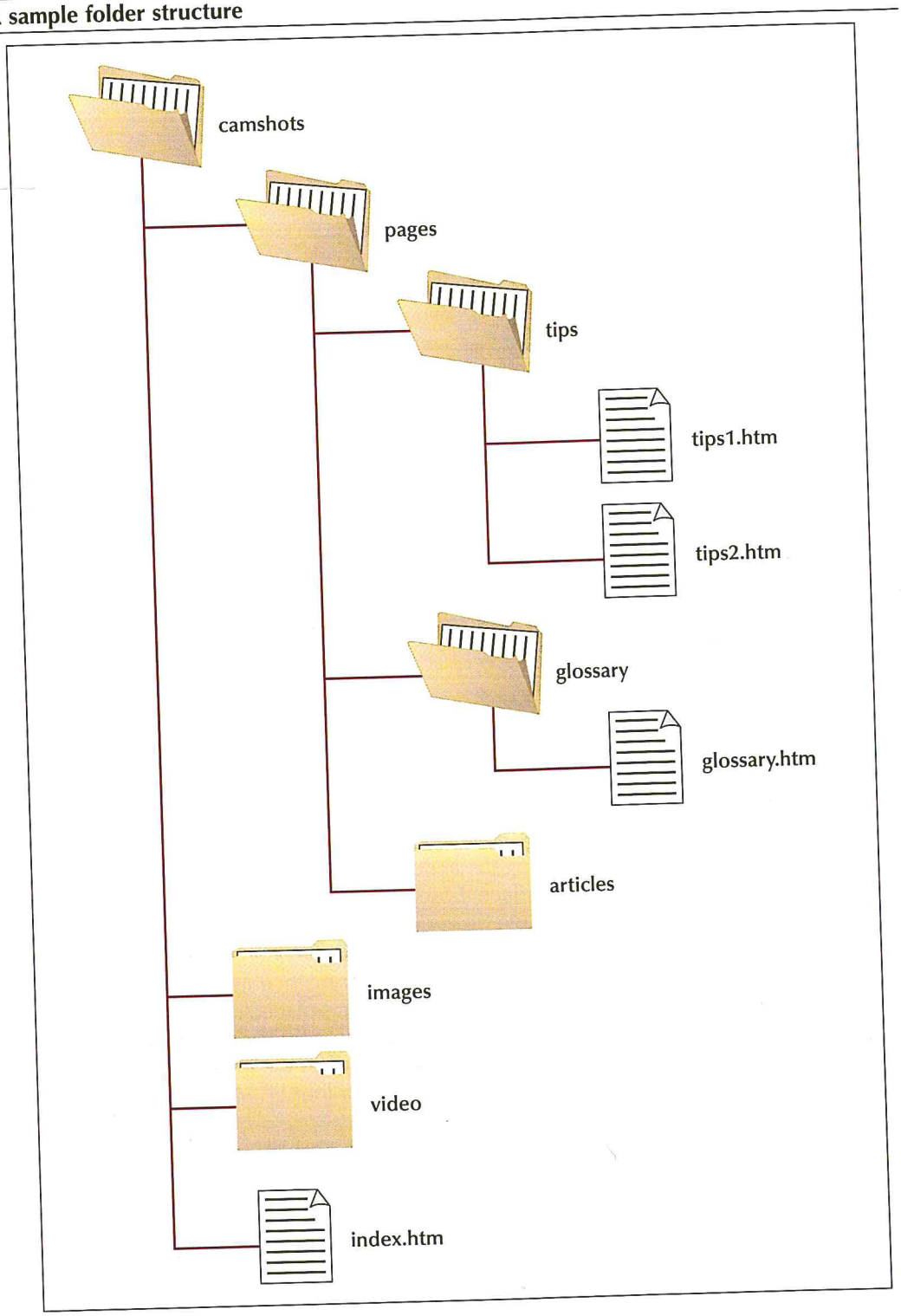
Trouble? If the links do not work, check the spelling of the filenames in the href attributes of the <a> tags. Because some Web servers require you to match capitalization in a filename, you should verify this in your attributes as well.

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.

Figure 2-11 A sample folder structure



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.

Tip

To make your Web site easier to maintain, organize your folders to match the organization of the pages on the Web site and group images and other media files within folders separate from your HTML files.

Absolute paths Figure 2-12

Absolute Path	Interpretation
/camshots/pages/tips/tips1.htm	The tips1.htm file located in the pages/tips subfolder
/camshots/pages/tips/tips2.htm	The tips2.htm file located in the pages/tips subfolder
/camshots/pages/glossary/glossary.htm	The glossary.htm file located in the pages/glossary subfolder
/camshots/index.htm	The index.htm file located in the camshots folder

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.

Figure 2-13 Relative paths

Relative Path from the /camshots/pages/tips Subfolder	Interpretation
tips1.htm	The tips1.htm file located in the current folder
tips2.htm	The tips2.htm file located in the current folder
../glossary/glossary.htm	The glossary.htm file located in the sibling glossary folder
../../index.htm	The index.htm file located in the parent camshots folder

Tip

You can reference the current folder using a single period (.) character.

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.

Reference Window | Using the Base Element to Set the Default Location of Relative Paths

- To set the default location for a relative path, add the element
  `<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 | InSight

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 | Review

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

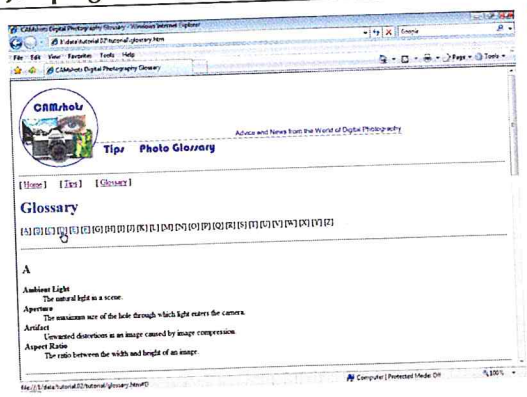
Linking to Locations within Documents

Tip

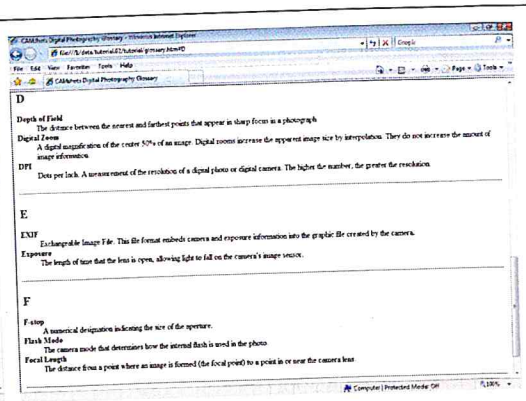
In general, Web pages should not span more than one or two screen heights. Studies show that long Web pages are often skipped by busy users.

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.

Figure 2-14 Jumping to a location within a Web page



clicking the letter D from the alphabetical list ...



...jumps the user to the D section of the glossary

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

```
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.

Reference Window | Defining an Element id

- To define the id of a specific element in a Web document, use the attribute
  `id="id"`
  where *id* is the value of the element id.