Session 2.2

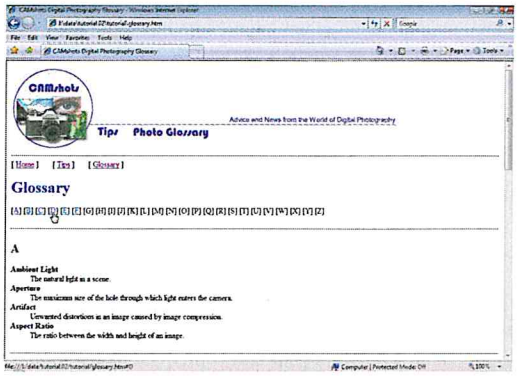
Tip

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.

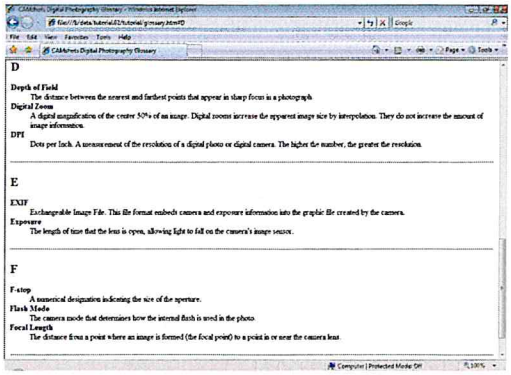
Linking to Locations within Documents

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.

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.

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?

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:

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

   `id="B"`

   Figure 2-15 highlights the revised code.

Adding the id attribute to h2 headings — Figure 2-15

```
<hr />
<h2 id="A">A</h2>
<dl>
   <dt><b>Ambient Light</b></dt>
   <dd>The natural light in a scene.</dd>
   <dt><b>Aperture</b></dt>
   <dd>The maximum size of the hole through which light enters the camera.</dd>
   <dt><b>Artifact</b></dt>
   <dd>Unwanted distortions in an image caused by image compression.</dd>
   <dt><b>Aspect Ratio</b></dt>
   <dd>The ratio between the width and height of an image.</dd>
</dl>

<hr />
<h2 id="B">B</h2>
<dl>
   <dt><b>Bit</b></dt>
   <dd>The smallest unit of computer memory.</dd>
   <dt><b>Bitmap</b></dt>
   <dd>A method of storing information that maps an image pixel bit by bit.</dd>
   <dt><b>Byte</b></dt>
   <dd>A group of 8 bits, the basic unit of information for the computer.</dd>
</dl>
```

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 tag, as shown in Figure 2-16:

   `id="top"`

Adding an id attribute to the div element — Figure 2-16

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

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

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

where *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:

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

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 **glossary.htm** file.
2. After the [character, insert the following opening tag:

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

3. Between the letter A and the] character, insert closing **** tag. Figure 2-17 shows the revised code.

Figure 2-17 Creating a hypertext link for "A"

```
<h1 style="color: teal">Glossary</h1>
<p>
   [<a href="#A">A</a>] [B] [C]
   [D] [E] [F]
   [G] [H] [I]
   [J] [K] [L]
   [M] [N] [O]
   [P] [Q] [R]
   [S] [T] [U]
   [V] [W] [X]
   [Y] [Z]
</p>
```

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.

Figure 2-18 Hypertext links for the list of letters

```
<h1 style="color: teal">Glossary</h1>
<p>
   [<a href="#A">A</a>] [<a href="#B">B</a>] [<a href="#C">C</a>]
   [<a href="#D">D</a>] [<a href="#E">E</a>] [<a href="#F">F</a>]
   [G] [H] [I]
   [J] [K] [L]
   [M] [N] [O]
   [P] [Q] [R]
   [S] [T] [U]
   [V] [W] [X]
   [Y] [Z]
</p>
```

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).

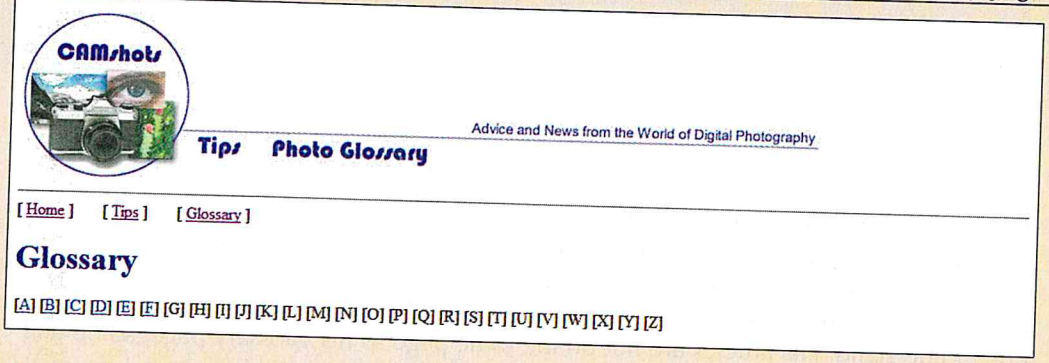
5. Scroll to the bottom of the file and locate the text "Return to Top."
6. Mark the text as hypertext, pointing to the element with an id value of top. See Figure 2-19.

Hypertext link to return to the top of the document Figure 2-19

```
<hr />
<div><a href="#top">Return to Top</a> &#8657;</div>
<hr />
<address>
   CAMshots &#8250;&#8250;&#8250; Tips and News from the World of Digital Photography
</address>
```

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.

Hypertext links in the glossary page Figure 2-20



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.

InSight | 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

```
<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:

```
<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:

```
<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

```
<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

```
<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
  ```
  <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
  ```
  <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

   `id="aperture"`

Inserting an id attribute Figure 2-21

```
<hr />
<h2 id="A">A</h2>
<dl>
   <dt><b>Ambient Light</b></dt>
   <dd>The natural light in a scene.</dd>
   <dt id="aperture"><b>Aperture</b></dt>
   <dd>The maximum size of the hole through which light enters the camera.</dd>
   <dt><b>Artifact</b></dt>
   <dd>Unwanted distortions in an image caused by image compression.</dd>
   <dt><b>Aspect Ratio</b></dt>
   <dd>The ratio between the width and height of an image.</dd>
</dl>
```

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

   `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:

   `<a href="glossary.htm#f-stop">F-stop</a>`
4. Mark "Exposure" as a hypertext link with:

   `<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.

Figure 2-22 Linking to a location within another document

```
<ul>
   <li>Camera:
              
      Nikon D50</li>
   <li><a href="glossary.htm#f-stop">F-stop</a>:
                
      f/7.1</li>
   <li><a href="glossary.htm#exposure">Exposure</a>:
           
      1/200 sec.</li>
   <li><a href="glossary.htm#focal_length">Focal Length</a>:
       
      18mm</li>
   <li><a href="glossary.htm#aperture">Aperture</a>:
             
      3.6</li>
   <li><a href="glossary.htm#flash_mode">Flash Mode</a>:
        No flash</li>
</ul>
```

document file

element id

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.

Figure 2-23 Linked photography terms

Photo of the Month

Pear Lake Reflection

- Camera: Nikon D50
- F-stop: f/7.1
- Exposure: 1/200 sec.
- Focal Length: 18mm
- Aperture: 3.6
- Flash Mode: No flash

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:

```
<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.

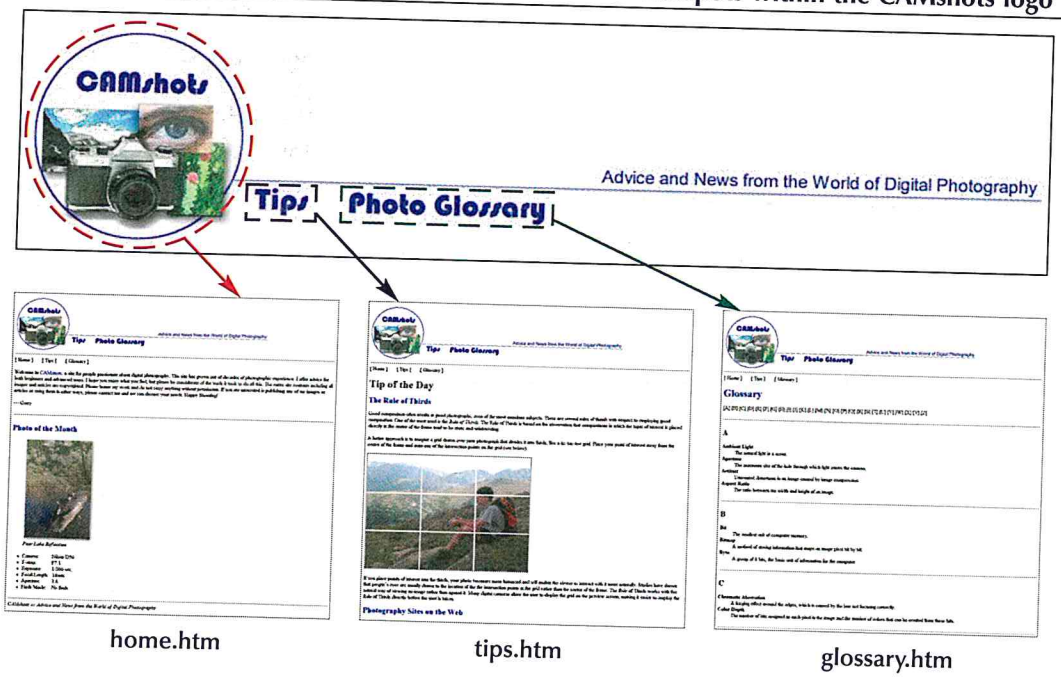
Tip

Always include alternative text for your linked images to allow nongraphical browsers to display a text link in place of the linked image.

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.

Hotspots within the CAMshots logo 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

```
<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:

```
<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

```
<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.

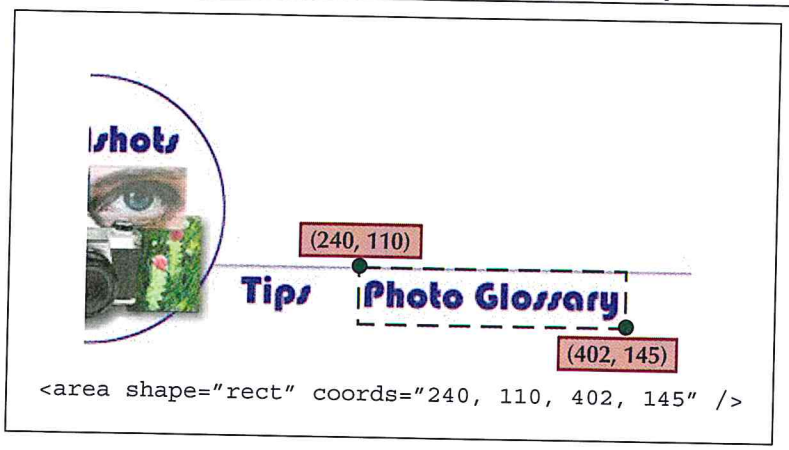
Hotspot coordinates are measured in **pixels**, which are the smallest unit or dot in a digital image or display. Your computer monitor might have a size of 1024 x 768 pixels, which means that the display is 1024 dots wide by 768 dots tall. The CAMshots logo that Gerry uses in his Web site has a dimension 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

```
<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.

Rectangular hotspot and area element Figure 2-25



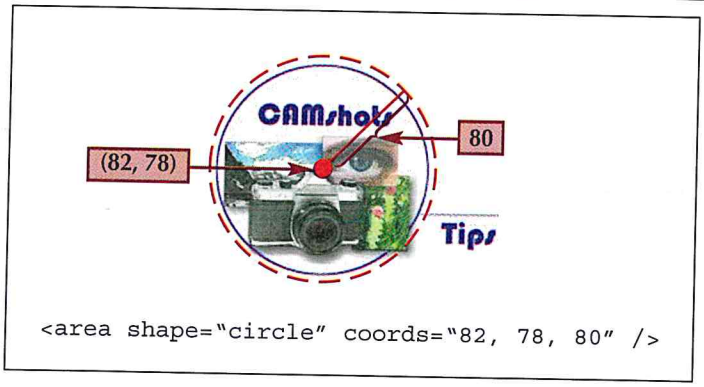
The upper-left corner of the rectangle has the coordinates (240, 110). The lower-right corner is found at the coordinates (402, 145). Coordinates are always expressed relative to the image's top-left corner. A coordinate of (240, 110) refers to a point that is 240 pixels to the right and 110 pixels down from the image's top-left corner.

Circular hotspots are defined using the area element

```
<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.

Circular hotspot and area element Figure 2-26

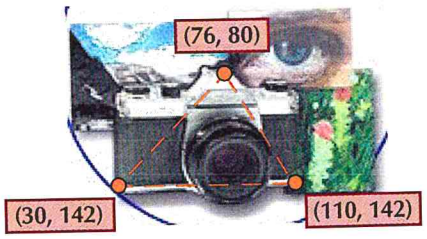


Polygonal hotspots are defined with

```
<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.

Figure 2-27 Polygonal hotspot and area element




```
<area shape="poly" coords="30, 142, 76, 80, 110, 142" />
```

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:

```
<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:

```
<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:

```
<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 — Figure 2-28

hotspots

```
<body>
   <div>
      <img src="camshots.jpg" alt="CAMshots" />
      <map id="logomap" name="logomap">
         <area shape="circle" coords="82, 78, 80"
               href="home.htm" alt="Home" />
         <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" />
      </map>
   </div>
   <hr />
```

5. Save your changes to the file.

Creating a Client-Side Image Map

Reference Window

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

```
<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

```
<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

```
<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

```
<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

```
<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

```
<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

```
<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" />
```

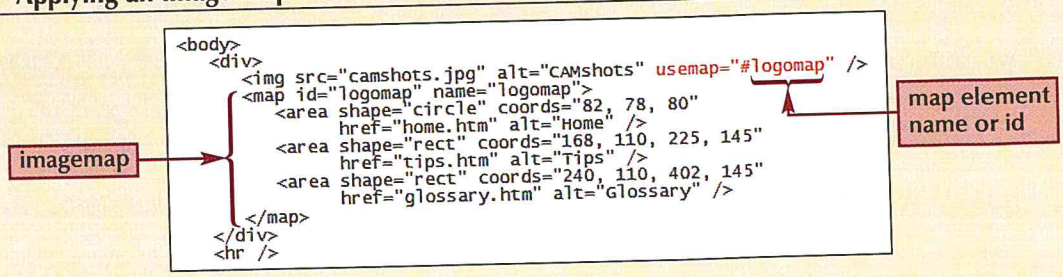
where *reference* is a reference to an HTML or XHTML file containing the map element. Unfortunately, most browsers do not support this option, so you should always place the image map in the same file as the inline image. You'll apply the logomap to the CAMshots logo and then test it on your Web browser.

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.

```
usemap="#logomap"
```

Figure 2-29 Applying an image map



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 CAMshots 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.

Tip

If you need to be compatible with older browsers, use the attribute border="0" in place of the border-width style. Note that the border attribute has been deprecated and is not supported in strict applications of XHTML.

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

```
style="border-width: 0"
```

Removing the Border from an Inline Image

Reference Window

- To remove a border from an inline image, add the following attribute to the <img> tag:
  `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.

   `style="border-width: 0"`

Removing an inline image border — Figure 2-30

```
<body>
   <div>
      <img src="camshots.jpg" alt="CAMshots" usemap="#logomap" style="border-width: 0" />
      <map id="logomap" name="logomap">
         <area shape="circle" coords="82, 78, 80"
               href="home.htm" alt="Home" />
         <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" />
      </map>
   </div>
   <hr />
```

set the width of the image border to 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 nongraphical browsers.

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

```
<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 | Review

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

Tip

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

```
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.

Figure 2-31 Internet protocols

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

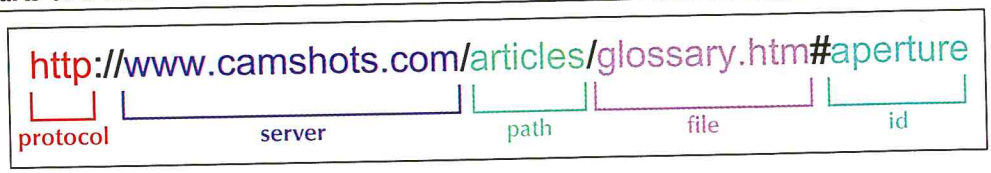
Linking to a Web Site

The URL for a Web page has the general form

`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.

Figure 2-32 Parts of a URL



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*.