**Hands on Exercise**

Launch a text editor and modify the template.html file to configure a web page as indicated:

1. Configure the text, Image Gallery, within an h1 element, and within the title element.
2. Code a div assigned to the id named gallery. This div will contain the thumbnail images, which will be configured within an unordered list.
3. Configure an unordered list within the div. Code four li elements, one for each thumbnail image. The thumbnail images will function as image links with a :hover pseudo-class that causes the larger image to display on the page. We’ll make this all happen by configuring an anchor element containing both the thumbnail image and a span element that comprises the larger image along with descriptive text. An example of the first li element is <li><a href="photo1.png"><img src="photo1thumb.png" width="100"

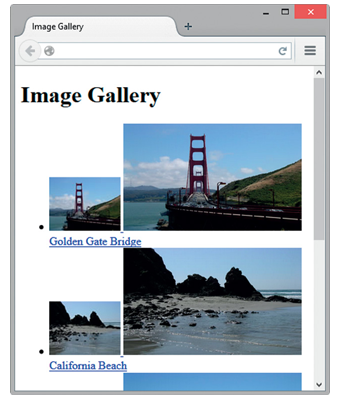
height="75" alt="Golden Gate Bridge">

<span><img src="photo1.png" width="250" height="150"

alt="Golden Gate Bridge"><br>Golden Gate Bridge</span></a>

</li>

1. Configure all four li elements in a similar manner. Substitute the actual name of each image file for the href and src values in the code. Write your own descriptive text for each image. Use photo2.png and photo2thumb.png in the second li element. Use photo3.png and photo3thumb.png in the third li element. Use photo4.png and photo4thumb.png for the fourth li element. Save the file as index.html in the gallery folder. Display your page in a browser. You’ll see an unordered list with the thumbnail images, the larger images, and the descriptive text. [**Figure 7.37**](https://jigsaw.vitalsource.com/books/9780133971026/epub/OPS/xhtml/fileP700049506200000000000000000235D.xhtml#P700049506200000000000000000237E) shows a partial screen capture.



**Figure 7.37 *The web page display before CSS.***

1. Now, let’s add embedded CSS. Open your index.html file in a text editor and code a style element in the head section. The gallery id will use relative positioning instead of the default static positioning. This does not change the location of the gallery but sets the stage to use absolute positioning on the span element in relation to its container (#gallery) instead of in relation to the entire web page document. This won’t matter too much for our very simple example, but it would be very helpful if the gallery were part of a more complex web page. Configure embedded CSS as follows:
   1. Set the gallery id to use relative positioning.

#gallery { position: relative; }

* 1. The unordered list in the gallery should have a width of 250 pixels and no list marker.

#gallery ul { width: 250px; list-style-type: none; }

* 1. Configure the list item elements in the gallery with inline display, left float, and 10 pixels of padding.

#gallery li { display: inline; float: left; padding: 10px; }

* 1. The images in the gallery should not display a border.

#gallery img { border-style: none; }

* 1. Configure anchor elements in the gallery to have no underline, #333 text color, and italic text.

#gallery a { text-decoration: none; color: #333; font-style: italic; }

* 1. Configure span elements in the gallery not to display initially.

#gallery span { display: none; }

* 1. Configure the span elements in the gallery to display *only* when the web visitor hovers the mouse over the thumbnail image link. Set the location of the span to use absolute positioning. Locate the span 10 pixels down from the top and 300 pixels in from the left. Center the text within the span:
  2. #gallery a:hover span { display: block; position: absolute;

top: 10px; left: 300px; text-align: center; }

Save your page and display it in a browser. Your interactive image gallery should work well in modern browsers.