CP 160 - Web Programming and Design

Lab 7: Page Layout

Hands-On Practice 1: Configure vertical navigation

You will configure vertical navigation with an unordered list in this Hands-On Practice. Create a folder named **ch7vert**. Copy the files lighthouseisland.jpg, lighthouselogo.jpg, and starter3.html into your **ch7vert** folder. Display the web page in a browser. It should look similar to **Figure 1**—notice that the navigation area needs to be configured.

Figure 1 Notice that the navigation area needs to be configured.

Launch a text editor and open the **starter3.html** file. Save the file as index.html in your **ch7vert** folder.

1. Review the code for this page, which uses a two-column layout. Examine the nav element and modify the code surrounding the hyperlinks to configure the navigation in an unordered list.

```
<nav>
 <ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="menu.html">Menu</a></li>
  <li><a href="directions.html">Directions</a></li>
  <li><a href="contact.html">Contact</a></li>
 </ul>
```

```
</nav>
```

2. Let's add CSS to the embedded styles to configure the unordered list elements only within the nav element: eliminate the list marker and set the padding to 10 pixels.

```
nav ul { list-style-type: none;
         padding: 10px; }
```

3. Next, configure the anchor tags within the nav element to have 10 pixels of padding, use bold font, and display no underline.

```
nav a { text-decoration: none;
        padding: 10px;          font-weight: bold; }
```

Save your page and test it in a browser. Your page should look similar to **Figure 2**.

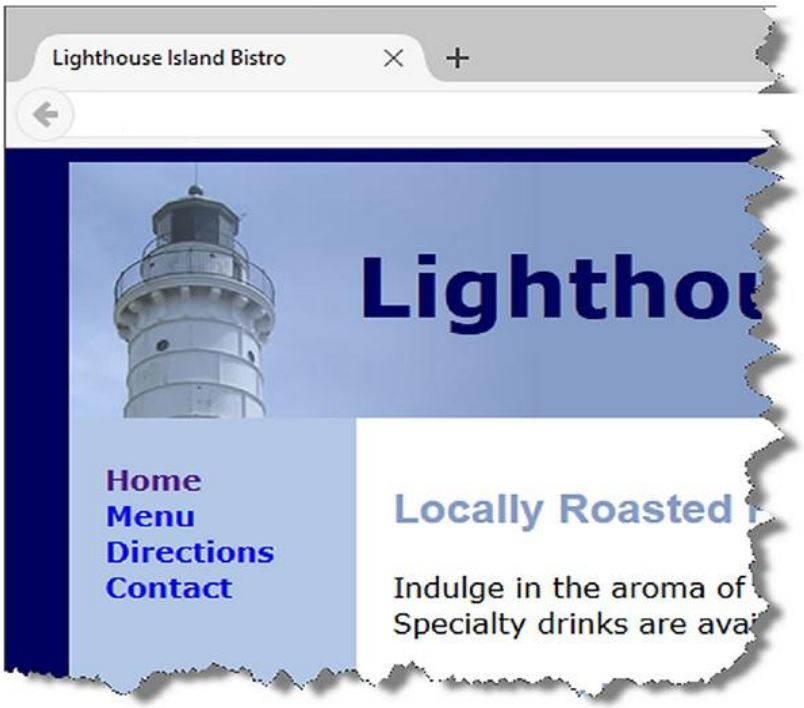

Figure 2 Two-column layout with vertical navigation.

Hands-On Practice 2: Configure horizontal navigation

You will configure horizontal navigation with an unordered list in this Hands-On Practice. Create a folder named **ch7hort**. Copy the files lighthouseisland.jpg, lighthouselogo.jpg, and starter4.html into your ch7hort folder. Display the web page in a browser. It should look similar to **Figure 3**— notice that the navigation area needs to be configured to display in a single line.

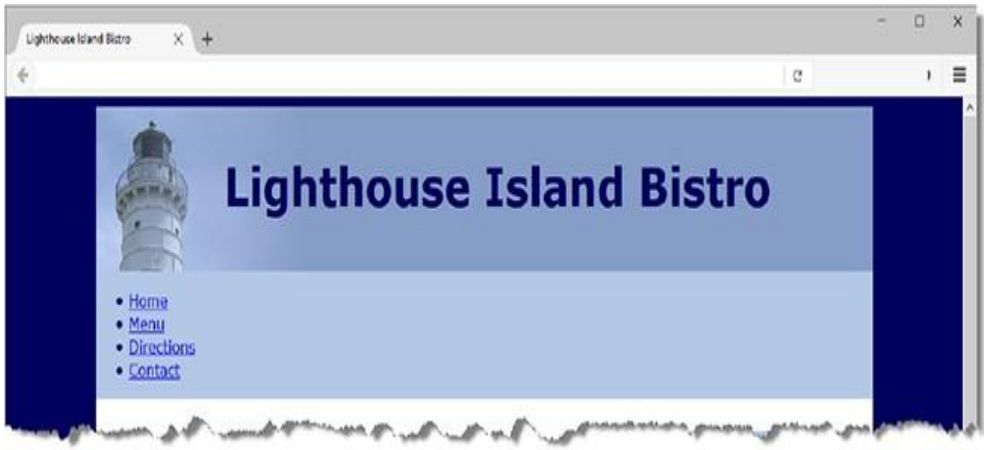

Figure 3 Notice that the navigation area needs to be configured.

Launch a text editor and open the starter4.html file. Save the file as index.html in your **ch7hort** folder.

1. Examine the nav element and notice that it contains an unordered list with navigation hyperlinks. Let's add CSS to the embedded styles to configure the unordered list element within the nav element: eliminate the list marker, center the text, set the font size to 1.5em, and set the margin to 5 pixels.

```
nav ul { list-style-type: none;
         text-align: center;
         font-size: 1.5em;
         margin: 5px; }
```

2. Configure the li elements within the nav element to display as inline elements.

```
nav li { display: inline; }
```

3. Configure the anchor elements within the nav element to display no underline. Also set the left and right padding to 10 pixels.

```
nav a { text-decoration: none;
        padding-left: 10px;          padding-right: 10px; }
```

Save your page and test it in a browser. Your page should look similar to **Figure 4**.

Figure 4 Horizontal navigation within an unordered list.

Hands-On Practice 3: Configure two columns

In this Hands-On Practice, you'll create a new version of the Lighthouse Island Bistro home page with a top header section spanning two columns, content in the left column, navigation in the right column, and a footer section below the two columns. See **Figure 5** for the wireframe. You will configure the CSS in an external style sheet. Create a new folder named **ch7bistro**. Copy the **starter5**.html, lighthouseisland.jpg, and lighthouselogo.jpg files into your **ch7bistro** folder.

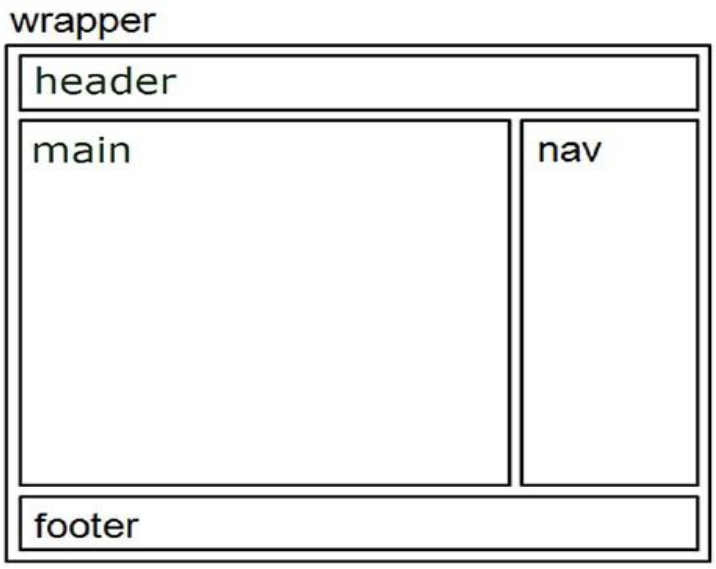

Figure 5 The wireframe for a two-column layout with a top logo area.

1. Launch a text editor and open the **starter5**.html file. Add a link element to the head section of the web page that associates this file with an external style sheet named bistro.css. A code sample follows:

```
<link href="bistro.css" rel="stylesheet">
```

   Save the file with the name index.html.

2. Launch a text editor and create a new file named **bistro.css** in your ch7bistro folder. Configure the CSS as follows:

   - **The universal selector**: set the box-sizing property to border-box.

```
*{ box-sizing: border-box; }
```

   - **The body element selector**: very dark blue background (#00005D) and Verdana, Arial, or the default sans-serif font typeface

```
body { background-color: #00005D;        font-family: Verdana,
Arial, sansserif; }
```

- **The wrapper id**: centered, take up 80% of the browser viewport, a minimum width of 940px, dark blue text (#000066), and medium-blue (#B3C7E6) background (*this color will display behind the nav section*)

```
#wrapper { margin: 0 auto; width: 80%; min-width: 940px;
           background-color: #B3C7E6; color: #000066; }
```

- **The header element selector**: slate blue (#869DC7) background; very dark blue text (#00005D); 150% font size; 10px top, right, and bottom padding; 155px left padding; height set to 150 pixels; and the lighthouselogo.jpg background image

```
header { background-color: #869DC7;
         color: #00005D;
         font-size: 150%;
         padding: 10px
        10px 10px 155px;
         height: 150px;
         background-repeat: no-repeat;
         background-image: url(lighthouselogo.jpg); }
```

- **The nav element selector**: float on the right, 150px width, bold text, and 0.1em letter spacing

```
nav { float: right; width: 150px; font-
weight: bold;       letter-spacing: 0.1em; }
```

- **The main element selector:** white background (#FFFFFF), black text (#000000), 10 pixels top and bottom padding, and 20 pixels left and right padding, auto overflow, and block display (fixes an Internet Explorer 11 rendering issue).

```
main { background-color: #FFFFFF;
       color: #000000;
        padding: 10px 20px;
       overflow: auto;
       display: block; }
```

- **The footer element selector**: 70% font size, centered text, 10 pixels of padding, a slate blue background color (#869DC7), and `clear` set to `both`.

```
footer { font-size: 70%; text-align: center; padding: 10px;
background-color: #869DC7; clear: both; }
```

Save the bistro.css file. Display index.html in a browser. Your page should look similar to **Figure 6.**

3. Continue editing the bistro.css file to style the h2 element selector and floating image. Configure the h2 element selector with slate blue text color (#869DC7) and Arial or sans-serif font typeface. Configure the `floatright` id to float on the right side with 10 pixels of margin.

```
h2 { color: #869DC7;
     font-family: Arial, sans-serif; }
#floatright { float: right; margin: 10px; }
```

4. Continue editing the bistro.css file and configure the vertical navigation bar.

   ■ Configure the unordered list: eliminate list markers, set zero margin, and set zero padding:

```
nav ul { list-style-type: none; margin: 0; padding: 0; }
```

   ■ Configure hyperlinks: no underline, 20 pixels padding, medium-blue background color (#B3C7E6), and 1 pixel solid white bottom border. Use `display: block;` to allow the web page visitor to click anywhere in the anchor "button" to activate the hyperlink.

```
nav a { text-decoration: none;
        padding: 20px;
        display: block;
        background-color: #B3C7E6;
        border-bottom: 1px solid #FFFFFF;
}
```

   ■ Configure the `:link`, `:visited`, and `:hover` pseudo-classes as follows:

```
nav a:link { color: #FFFFFF; }
nav a:visited { color: #EAEAEA; }
nav a:hover { color: #869DC7;    background-color: #EAEAEA; }
```

Save your file. Display your index.html page in a browser. Move your mouse pointer over the navigation area and notice the interactivity, as shown in **Figure** 7.

Figure 6 The home page with major page sections configured using CSS.

Figure 7 CSS pseudo-classes add interactivity to the page.

Hands-On Practice 4: Configure for optimal screen display and printing

In this Hands-On Practice, you'll rework the Lighthouse Island Bistro page to use external style sheets and be configured for optimal screen display and printing. Create a new folder named **ch7print**. Copy the files from your ch7bistro folder into the ch7print folder.

1. Launch a text editor and open the index.html file. Examine the source code and locate the style element. Copy the CSS between the style tags and paste into a new text document named bistro.css. Save the bistro.css file in the ch7print folder.
2. Edit the index.html file and edit the link tag in the head section that associates the web page with the bistro.css file to specify screen display (use `media="screen"`).
3. Edit the index.html file and add another link tag that associates the web page with a file named bistroprint.css for printing (use `media="print"`). Save the index.html file.
4. Launch a text editor and open bistro.css. Since you want to keep most of the styles for printing, you will start by creating a new version of the external style sheet. Save bistro.css with the name of bistroprint.css in the ch7print folder. You will modify three areas on this style sheet: the header selector, the main selector, and the nav selector. ■ Modify the header styles to print using black text in 20 point font size:

   ```
   header { color: #000000; font-size: 20pt; }
   ```

   ■ Modify the main element area to print using a serif typeface in a 12 point font size:

   ```
   main { font-family: "Times New Roman",
   serif; font-size: 12pt; }
   ```

   ■ Modify the navigation area to not display:

   ```
   nav { display: none; }
   ```

   Save your file.

5. Test your work. Display your index.html file in a browser. Select Print from the menu. Your display should look similar to the page shown in **Figure 8**. The header and content font sizes have been configured. The navigation does not display.

Figure 8 The print preview display of the web page.

Hands-On Practice 5: Practice with positioning and create an interactive image gallery

In this Hands-On Practice, you will create the interactive image gallery web page shown in **Figure 9**. Copy the following images located in the **starters** folder into a folder named **gallery**: photo1.jpg, photo2.jpg, photo3.jpg, photo4.jpg, thumb1.jpg, thumb2.jpg, thumb3.jpg, and thumb4.jpg. 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. Code a div assigned to the id named `gallery`. This div will contain the thumbnail images, which will be configured within an unordered list. 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 follows:

```
<li><a href="photo1.jpg"><img src="thumb1.jpg" width="100"
height="75" alt="Golden Gate Bridge">
    <span><img src="photo1.jpg" width="250" height="150"
     alt="Golden Gate Bridge"><br>Golden Gate Bridge</span></a>
</li>
```

2. 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.jpg and thumb2.jpg in the second li element. Use photo3.jpg and thumb3.jpg in the third li element. Use photo4.jpg and thumb4.jpg 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 9** shows a partial screen capture.
3. Now, let's add embedded CSS. Open your index.html file in a text editor and code a style element in the head section. Code embedded CSS as follows:
   a. Configure the universal selector with box-sizing set to border-box

```
* { box-sizing: border-box; }
```

   b. 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 example, but it would be very helpful if the gallery were part of a more complex web page. Set the gallery id to use relative positioning.

```
#gallery { position: relative; }
```

c. The gallery unordered list should be 280px wide with no list marker.

```
#gallery ul { width: 280px; liststyle-type: none; }
```

d. Configure the gallery li elements with inline display, left float, and 10px padding.

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

e. The images in the gallery should not display a border.

```
#gallery img { border-style: none; }
```

f. Configure gallery anchor elements to have no underline with italic #333 text.

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

g. Configure span elements in the gallery not to display initially.

```
#gallery span { display: none; }
```

h. Configure the span elements in the gallery to display *only* when the web visitor hovers the mouse pointer 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:

```
#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. Compare your work to **Figure 10**.

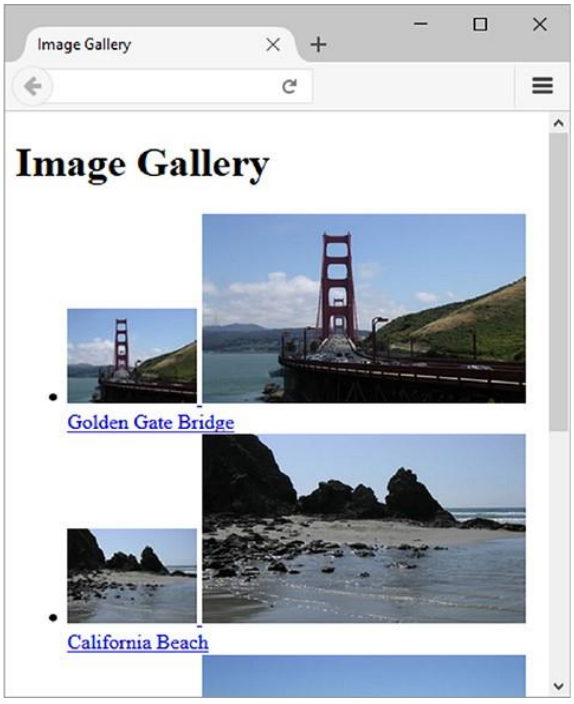

Figure 9 The web page display before CSS.

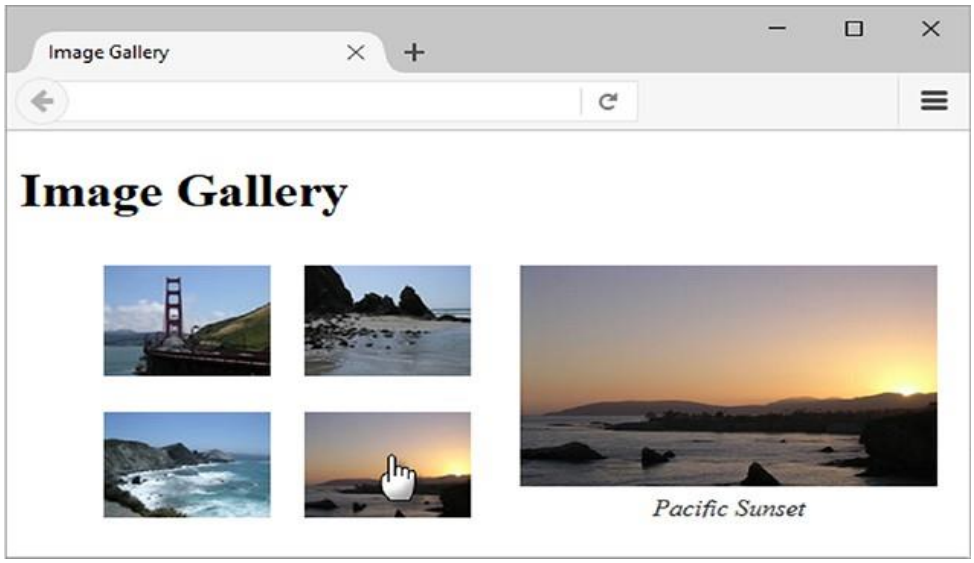

Figure 10 An interactive image gallery with CSS.

Lab Completion / Submission:

Complete all the lab practices. Take the screenshots of your completed webpages; put them into a single word file and submit it to **Blackboard -> CP160 -> Assessments -> Lab7 / Assignment 7**; due date: today

After-lab Assignment:

1. Create a one-paragraph conclusion of what you have learned during the lab today.
2. Write the CSS for an id with the following characteristics: fixed position, light gray background color, bold font weight, and 10 pixels of padding.
3. Write the CSS for a class that is relatively positioned. This class should appear 15 pixels from the left. Configure the class to have a light-green background.
4. Google CSS page layout tutorials and find out one CSS technique which is not covered in this and previous chapters. Create a web page to provide the URL of the tutorial, the name of the website and description of the new technique.

Submit to **Blackboard -> CP160 -> Assessments -> Lab 7 / Assignment 7;** due date: 1 week from today.

Project: Implement the website designed in your proposal.

Please start to work on implementing your website using the HTML and CSS code level techniques learned in this course. More instructions on how to submit the project will be provided.