

- b. Describe the process you went through as you selected the colors. Describe why you chose these colors. What type of website would they be appropriate for? List the URLs of any resources you used.
- c. Create an external CSS file named color1.css that configures font properties, text color, and background color selections for the document, h1 element selector, p element selector, and footer class, using the colors you have chosen.
- d. Create a web page named color1.html that shows examples of the CSS style rules.



WEBSITE CASE STUDY

Implementing CSS

Each of the case studies in this section continues throughout most of the text. This chapter implements CSS in the websites.

JavaJam Coffee House

See Chapter 2 for an introduction to the JavaJam Coffee House Case Study. Figure 2.30 shows a site map for the JavaJam website. The Home page and Menu page were created in Chapter 2. You will develop a new version of the website that uses an external style sheet to configure text and color. Figure 2.31 depicts the wireframe page layout.

You have the following tasks:

1. Create a new folder for this JavaJam case study.
2. Create an external style sheet named javajam.css that configures the color and text for the JavaJam website.
3. Modify the Home page to utilize an external style sheet to configure colors and fonts.

The new Home page and color swatches are shown in Figure 3.29.



Figure 3.29
New JavaJam
index.html

4. Modify the Menu page to be consistent with the new Home page.
5. Configure centered page layout.

Hands-On Practice Case

Task 1: Create a folder on your hard drive or portable storage device called javajamcss. Copy all the files from your Chapter 2 javajam folder into the javajamcss folder.

Task 2: The External Style Sheet. You will use a text editor to create an external style sheet named javajam.css. Code the CSS to configure the following:

1. Global styles for the document (use the body element selector) with background color #FCEBB6; text color #221811; and Verdana, Arial, or any sans-serif font.
2. Styles for the header element selector that configure background color #D2B48C and centered text.
3. Styles for the h1 element selector that configure 200% line height.
4. Styles for the nav element selector that configure centered, bold text. *Hint:* Use the CSS `text-align` and `font-weight` properties.
5. Styles for the footer element selector that configure background color #D2B48C, small font size (.60em), italics, and centered text.

Save the file as javajam.css in the javajamcss folder. Check your syntax with the CSS validator (<http://jigsaw.w3.org/css-validator>). Correct and retest if necessary.

Task 3: The Home Page. Launch a text editor, and open the index.html file. You will modify this file to apply styles from the javajam.css external style sheet as follows:

1. Add a `<link>` element to associate the web page with the javajam.css external style sheet file.
2. Configure the navigation area. Remove the `` elements which are no longer needed because you have configured bold text with CSS.
3. Configure the page footer area. Remove the `<small>` and `<i>` elements—they are no longer needed since CSS is now used to configure the text.

Save the index.html file, and test it in a browser. Your page should look similar to the one shown in Figure 3.29 except that your page content will be left-aligned instead of centered. Don't worry—you'll center your page layout in Task 5 of this case study.

Task 4: The Menu Page. Launch a text editor, and open the menu.html file. You will modify this file in a similar manner as you modified the home page: Add the `<link>` element and configure the navigation and page footer areas. Save and test your new menu.html page. It should look similar to the one shown in Figure 3.30, except for the alignment.

Task 5: Center Page Layout with CSS. Modify javajam.css, index.html, and menu.html to configure page content that is centered with 80% width (refer to Hands-On Practice 3.9 if necessary):

1. Launch a text editor, and open the javajam.css file. Add a style rule for an id named `wrapper` with `width` set to 80%, `margin-right` set to auto, and `margin-left` set to auto.
2. Launch a text editor, and open the index.html file. Add the HTML code to configure a `div` element assigned to the id `wrapper` that “wraps,” or contains, the code within the `body` section. Save and test your index.html page in a browser and you'll notice that the page content is now centered within the browser viewport as shown in Figure 3.29.

3. Launch a text editor, and open the menu.html file. Add the HTML code to configure a `div` element assigned to the `id wrapper` that “wraps,” or contains, the code within the `body` section. Save and test your menu.html page in a browser and you’ll notice that the page content is now centered within the browser viewport as shown in Figure 3.30.

Experiment with modifying the javajam.css file. Change the page background color, the font family, and so on. Test your pages in a browser. Isn’t it amazing how a change in a single file can affect multiple files when external style sheets are used?

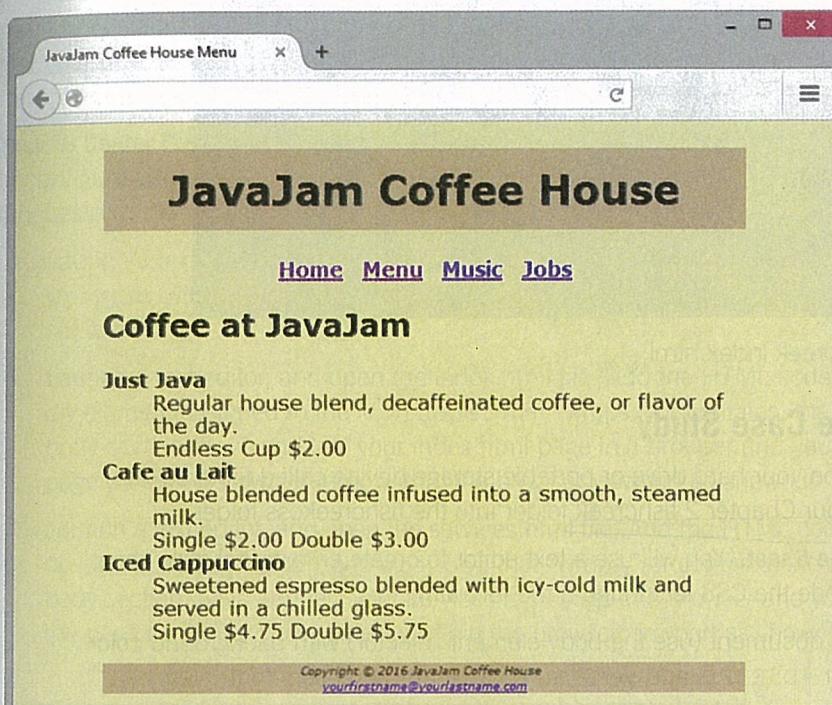


Figure 3.30
New menu.html page

Fish Creek Animal Hospital

See Chapter 2 for an introduction to the Fish Creek Animal Hospital Case Study. Figure 2.34 shows a site map for the Fish Creek website. The Home page and Services page were created in Chapter 2. You will develop a new version that uses an external style sheet to configure text and color. Figure 2.35 depicts the wireframe page layout.

You have the following tasks:

1. Create a new folder for this Fish Creek case study.
2. Create an external style sheet named fishcreek.css that configures the color and text for the Fish Creek website.
3. Modify the Home page to utilize an external style sheet to configure colors and fonts. The new Home page and color swatches are shown in Figure 3.31.
4. Modify the Services page to be consistent with the new Home page.
5. Configure centered page layout.

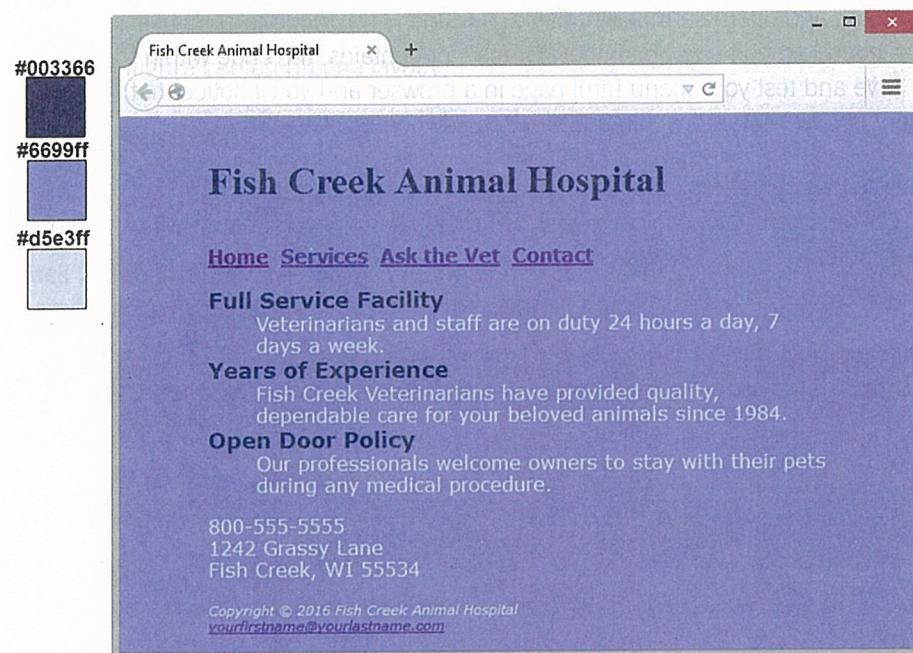


Figure 3.31 New Fish Creek index.html

Hands-On Practice Case Study

Task 1: Create a folder on your hard drive or portable storage device called fishcreekcss. Copy all the files from your Chapter 2 fishcreek folder into the fishcreekcss folder.

Task 2: The External Style Sheet. You will use a text editor to create an external style sheet named fishcreek.css. Code the CSS to configure the following:

1. Global styles for the document (use the body element selector) with background color #6699FF; text color #D5E3FF; and Verdana, Arial, or any sans-serif font.
2. Styles for the header element selector that configure background color #6699FF, text color #003366, and serif font.
3. Styles for the h1 element selector that configure 200% line height.
4. Styles for the nav element selector that display text in bold.
5. Styles for a class named category with bold font, background color #6699FF, text color #003366, and larger font size (1.1em).
6. Styles for the footer element selector with a small font size (.70em) and italic text.

Save the file as fishcreek.css in the fishcreekcss folder. Check your syntax with the CSS validator (<http://jigsaw.w3.org/css-validator>). Correct and retest if necessary.

Task 3: The Home Page. Launch a text editor, and open the index.html file. You will modify this file to apply styles from the fishcreek.css external style sheet as follows:

1. Add a <link> element to associate the web page with the fishcreek.css external style sheet file.
2. Configure the navigation area. Remove the element from the navigation area, because the CSS will configure the bold font style.
3. Configure each <dt> element to apply the category class.
Hint <dt class="category">. Remove the tags, because the CSS will configure the bold font style.

- Configure the page footer area. Remove the `<small>` and `<i>` elements—they are no longer needed since CSS is now used to configure the text.

Save the index.html file, and test in a browser. Your page should look similar to the one shown in Figure 3.31 except that your page content will be left-aligned instead of indented from the margins. Don't worry—you'll configure your page layout in Task 5 of this case study.

Task 4: The Services Page. Launch a text editor, and open the services.html file. You will modify this file in a similar manner: Add the `<link>` element, configure the navigation area and page footer areas, configure the category classes (*Hint:* Use the `` element to contain the name of each service offered), and remove the strong tags.) Save and test your new services.html page. It should look similar to the one shown in Figure 3.32 except for the alignment.

Task 5: Center Page Layout with CSS. Modify fishcreek.css, index.html, and services.html to configure page content that is centered with 80% width. Refer to Hands-On Practice 3.9 if necessary.

- Launch a text editor, and open the fishcreek.css file. Add a style rule for an id named `wrapper` with `width` set to 80%, `margin-right` set to auto, and `margin-left` set to auto.
- Launch a text editor, and open the index.html file. Add the HTML code to configure a `div` element assigned to the id `wrapper` that “wraps,” or contains, the code within the body section. Save and test your index.html page in a browser and you'll notice that the page content is now centered within the browser viewport as shown in Figure 3.31.
- Launch a text editor, and open the services.html file. Add the HTML code to configure a `div` element assigned to the id `wrapper` that “wraps,” or contains, the code within the body section. Save and test your services.html page in a browser and you'll notice that the page content is now centered within the browser viewport as shown in Figure 3.32.

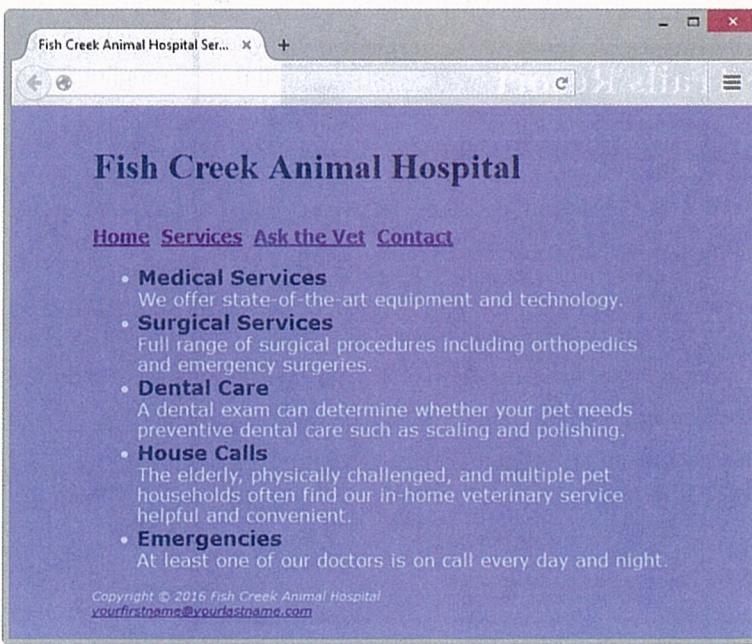


Figure 3.32 New services.html page

Experiment with modifying the fishcreek.css file. Change the page background color, the font family, and so on. Test your pages in a browser. Isn't it amazing how a change in a single file can affect multiple files when external style sheets are used?

Pacific Trails Resort

See Chapter 2 for an introduction to the Pacific Trails Resort Case Study. Figure 2.38 shows a site map for the Pacific Trails Resort website. The Home page and Yurts page were created in Chapter 2. You will develop a new version of this website that uses an external style sheet to configure text and color. Figure 2.39 depicts the wireframe page layout.

You have the following tasks:

1. Create a new folder for this Pacific Trails case study.
2. Create an external style sheet named pacific.css that configures the color and text for the Pacific Trails website.
3. Modify the Home page to utilize an external style sheet to configure colors and fonts. The new Home page and color swatches are shown in Figure 3.33.
4. Modify the Yurts page to be consistent with the new Home page.
5. Configure centered page layout.

Hands-On Practice Case Study

Task 1: Create a folder on your hard drive or portable storage device called pacificcss. Copy all the files from your Chapter 2 pacific folder into the pacificcss folder.



Figure 3.33 New Pacific Trails index.html

Task 2: The External Style Sheet. You will use a text editor to create an external style sheet named pacific.css. Code the CSS to configure the following:

1. Global styles for the document (use the body element selector) with background color #FFFFFF, text color #666666, and Verdana, Arial, or any sans-serif font.
2. Styles for the header element selector that configure background color #000033, text color #FFFFFF, and Georgia or any serif font.
3. Styles for the h1 element selector that configure 200% line height.
4. Styles for the nav element selector that display text in bold and has a sky-blue background color (#90C7E3).
5. Styles for the h2 element selector that configure medium-blue text color (#3399CC) and Georgia or any serif font.
6. Styles for the dt element selector that configure dark-blue text color (#000033) and bold font.
7. Styles for a class named `resort` that configure dark-blue text color (#000033) and 1.2em font size.
8. Styles for the footer element selector with a small font size (.70em) and italic, centered text.

Save the file as pacific.css in the pacificcss folder. Check your syntax with the CSS validator (<http://jigsaw.w3.org/css-validator>). Correct and retest if necessary.

Task 3: The Home Page. Launch a text editor, and open the index.html file. You will modify this file to apply styles from the pacific.css external style sheet as follows:

1. Add a `<link>` element to associate the web page with the pacific.css external style sheet file.
2. Configure the navigation area. Remove the `` element from the navigation area, because the CSS will configure the bold font weight.
3. Find the company name (“Pacific Trails Resort”) in the first paragraph below the h2. Configure a span that contains this text. Assign the span element to the `resort` class.
4. Look for the company name (“Pacific Trails Resort”) directly above the street address. Configure a span that contains this text. Assign the span element to the `resort` class.
5. Configure the page footer area. Remove the `<small>` and `<i>` elements—they are no longer needed since CSS is now used to configure the text.

Save the index.html file, and test in a browser. Your page should look similar to the one shown in Figure 3.33 except that your page content will be left-aligned instead of indented from the margins. Don’t worry—you’ll configure your page layout in Task 5 of this case study.

Task 4: The Yurts Page. Launch a text editor, and open the yurts.html file. You will modify this file in a similar manner: Add the `<link>` element, configure the navigation area, and configure the page footer area. Delete the strong tags contained within each dt element. Save and test your new yurts.html page. It should look similar to the one shown in Figure 3.34 except for the alignment.

Task 5: Center Page Layout with CSS. Modify pacific.css, index.html, and yurts.html to configure page content that is centered with 80% width. Refer to Hands-On Practice 3.9 if necessary.

1. Launch a text editor, and open the `pacific.css` file. Add a style rule for an id named `wrapper` with `width` set to 80%, `margin-right` set to auto, and `margin-left` set to auto.
2. Launch a text editor, and open the `index.html` file. Add the HTML code to configure a `div` element assigned to the id `wrapper` that “wraps,” or contains, the code within the `body` section. Save and test your `index.html` page in a browser and you’ll notice that the page content is now centered within the browser viewport as shown in Figure 3.33.
3. Launch a text editor and open the `yurts.html` file. Add the HTML code to configure a `div` element assigned to the id `wrapper` that “wraps,” or contains, the code within the `body` section. Save and test your `yurts.html` page in a browser and you’ll notice that the page content is now centered within the browser viewport as shown in Figure 3.34.

Experiment with modifying the `pacific.css` file. Change the page background color, the font family, and so on. Test your pages in a browser. Isn’t it amazing how a change in a single file can affect multiple files when external style sheets are used?

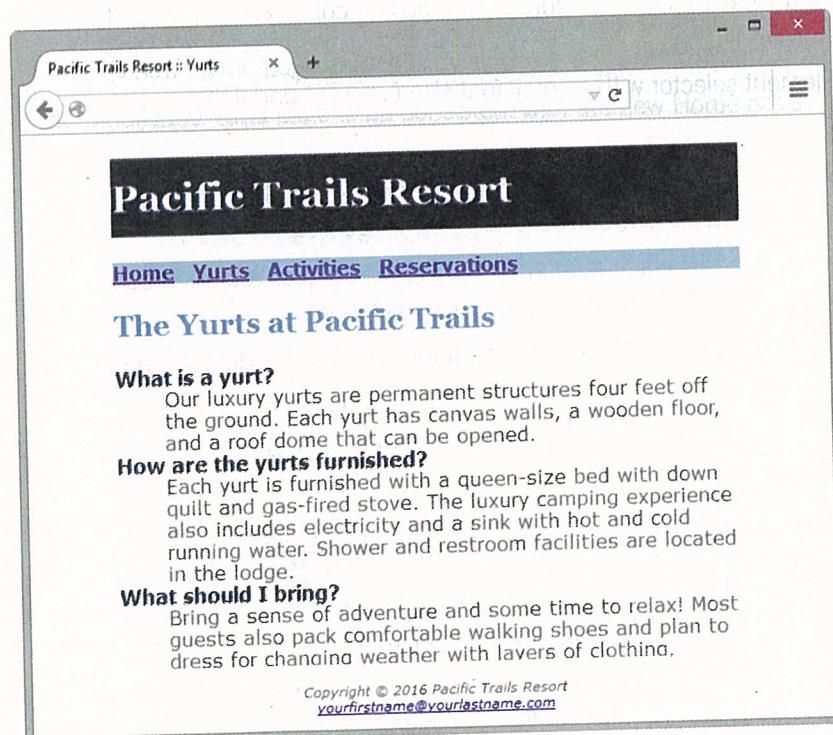


Figure 3.34 New `yurts.html` page

Path of Light Yoga Studio

See Chapter 2 for an introduction to the Path of Light Yoga Studio Case Study. Figure 2.42 shows a site map for the Path of Light Yoga Studio website. The Home page and Classes page were created in Chapter 2. You will develop a new version of this website that uses an external style sheet to configure text and color. Figure 2.43 depicts the wireframe page layout.

You have the following tasks:

1. Create a new folder for this Path of Light Yoga Studio case study.
2. Create an external style sheet named `yoga.css` that configures the color and text for the Path of Light Yoga Studio website.

3. Modify the Home page to utilize an external style sheet to configure colors and fonts. The new Home page and color swatches are shown in Figure 3.35.
4. Modify the Classes page to be consistent with the new Home page.
5. Configure centered page layout.

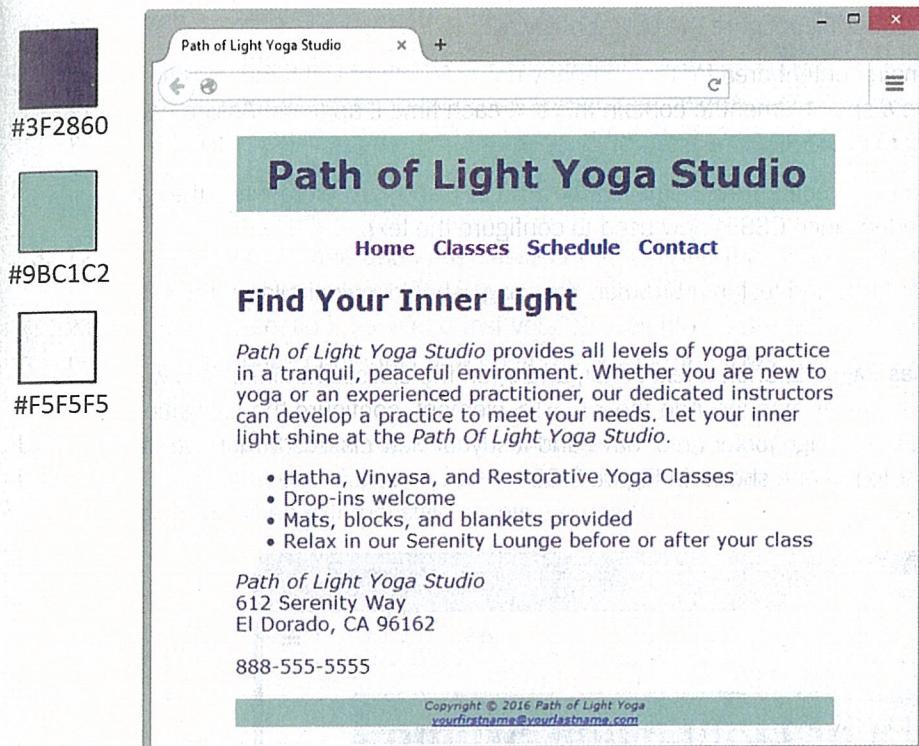


Figure 3.35 New Path of Light Yoga Studio index.html

Hands-On Practice Case Study

Task 1: Create a folder on your hard drive or portable storage device called yogacss. Copy all the files from your Chapter 2 yoga folder into the yogacss folder.

Task 2: The External Style Sheet. You will use a text editor to create an external style sheet named `yoga.css`. Code the CSS to configure the following:

1. Global styles for the document (use the `body` element selector) with background color `#F5F5F5`; text color `#3F2860`; and Verdana, Arial, or any sans-serif font.
2. Styles for the `header` element that configure background color `#9BC1C2` with centered text.
3. Styles for the `h1` element selector that configure 200% line height.
4. Styles for the `nav` element selector that configure centered and bold font.
5. Styles for the anchor elements within the `nav` area to eliminate the default underline (Hint: use the `nav a` selector).
6. Styles for a class named `studio` that configures italic text.
7. Styles for the `footer` element selector with `#9BC1C2` background color, small font size (`.60em`), and italic, centered text.

Save the file as `yoga.css` in the `yogacss` folder. Check your syntax with the CSS validator (<http://jigsaw.w3.org/css-validator>). Correct and retest if necessary.

Task 3: The Home Page. Launch a text editor, and open the index.html file. You will modify this file to apply styles from the yoga.css external style sheet.

1. Add a `<link>` element to associate the web page with the `yoga.css` external style sheet file.
2. Configure the navigation area. Remove the `` element from the navigation area, because the CSS will configure the bold font style.
3. Look in the main content area for the company name (“Path of Light Yoga Studio”) and configure a `span` element to contain this text each time it appears. Assign each `span` element to the `studio` class.
4. Configure the page footer area. Remove the `<small>` and `<i>` elements—they are no longer needed since CSS is now used to configure the text.

Save the `index.html` file, and test in a browser. Your page should look similar to the one shown in Figure 3.35.

Task 4: The Classes Page. Launch a text editor, and open the `classes.html` file. You will modify this file in a similar manner: Add the `<link>` element, configure the navigation area, and configure the page footer area. Save and test your new `classes.html` page. It should look similar to the one shown in Figure 3.36.

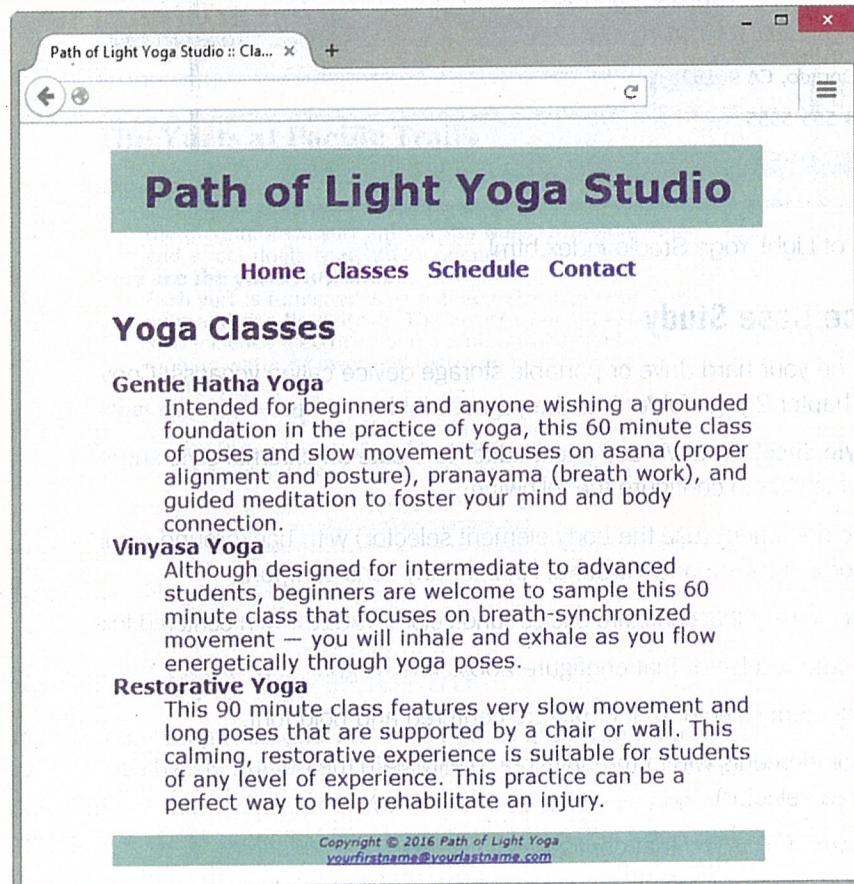


Figure 3.36 New `classes.html` page

Task 5: Center Page Layout with CSS. Modify `yoga.css`, `index.html`, and `classes.html` to configure page content that is centered with 80% width. Refer to Hands-On Practice 3.9 if necessary.

1. Launch a text editor, and open the `yoga.css` file. Add a style rule for an id named `wrapper` with `width` set to 80%, `margin-right` set to auto, and `margin-left` set to auto.
2. Launch a text editor, and open the `index.html` file. Add the HTML code to configure a `div` element assigned to the id `wrapper` that “wraps,” or contains, the code within the body section. Save and test your `index.html` page in a browser and you’ll notice that the page content is now centered within the browser viewport as shown in Figure 3.35.
3. Launch a text editor, and open the `classes.html` file. Add the HTML code to configure a `div` element assigned to the id `wrapper` that “wraps,” or contains, the code within the body section. Save and test your `classes.html` page in a browser and you’ll notice that the page content is now centered within the browser viewport as shown in Figure 3.36.

Experiment with modifying the `yoga.css` file. Change the page background color, the font family, and so on. Test your pages in a browser. Notice how a change in a single file can affect multiple files when external style sheets are used.