

Focus on Web Design

Visit a website that interests you. Print the home page or one other pertinent page from the site. Write a one-page summary and reaction to the website you chose to visit. Address the following topics:

- a. What is the purpose of the site?
- b. Who is the intended audience?
- c. Do you believe the site reaches its audience?
- d. Was this site useful to you? Why or why not?
- e. List the colors and/or graphics that are used on the home page of this website: background, backgrounds of page sections, text, logo, navigation buttons, and so on.
- f. How does the use of color and graphics enhance the website?



WEBSITE CASE STUDY

Using Graphics & Visual Elements

Each of the case studies in this section continues throughout most of the text. In this chapter, we add images to the websites, create a new page, and modify existing pages.

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 earlier chapters. Using the existing website as a starting point, you will modify the design of the pages and create a new page, the Music page. You have five tasks in this case study:

1. Create a new folder for this JavaJam case study, and obtain the starter image files.
2. Modify the Home page to display the winding road image as shown in Figure 4.47.
3. Modify the Menu page, shown in Figure 4.48, to be consistent with the Home page.
4. Create a new Music page, as shown in Figure 4.49.
5. Modify the style rules in the `javajam.css` file as needed.



Hands-On Practice Case

Task 1: Create a folder on your hard drive or portable storage device called javajam4. Copy all the files from your Chapter 3 javajamcss folder into the javajam4 folder. Obtain the images used in this case study from the student files. The images are located in the chapter4/casestudystarters/javajam folder. The images are background.gif, greg.jpg, grethumb.jpg, javajamlogo.jpg, melanie.jpg, melaniethumb.jpg, mugs.jpg, and windingroad.jpg. Save them in your javajam4 folder.

Task 2: The Home Page. Launch a text editor, and open the index.html file from your javajam4 folder. Modify the index.html file to look similar to the web page shown in Figure 4.47.

1. Replace the “Relax at JavaJam” text contained within the h2 element with “Follow the Winding Road to JavaJam.”
2. Code an img tag for the windingroad.jpg image above the h2 element in the main content area. Be sure to include the alt, height, and width attributes. Also configure the image to appear to the right of the text content by coding the align="right" attribute on the tag. *Note:* The W3C HTML validator will indicate that the align attribute is invalid. We'll ignore the error for this case study. In Chapter 6 you'll learn to use the CSS float property (instead of the align property) to configure this type of layout.
3. Add a paragraph with the following text below the h2 element:

“We’re a little out of the way, but take a drive down Route 42 to JavaJam today! Indulge in our locally roasted free-trade coffee and home-made pastries. You’ll feel right at home at JavaJam!”
4. Configure the text “JavaJam Coffee House features:” within an h3 element below the paragraph and above the unordered list.

Save and test your new index.html page. It will be similar to Figure 4.47, but you'll notice that a few final touches (including the background image and logo image) are missing; you'll configure these with CSS in Task 5.

Task 3: The Menu Page. Launch a text editor, and open the menu.html page from your javajam4 folder. Modify the menu.html file to look similar to the web page shown in Figure 4.48.

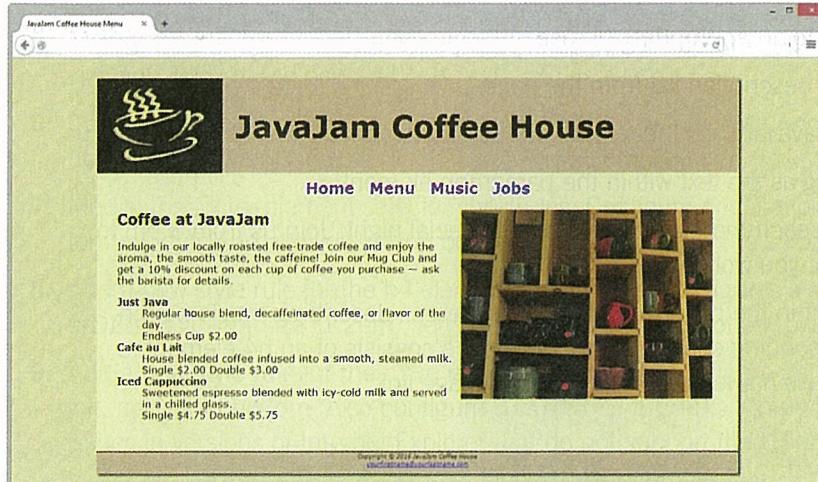


Figure 4.48 JavaJam menu.html

1. Code an img tag for the mugs.jpg image above the h2 element in the main content area. Be sure to include the alt, height, and width attributes. Also configure the image to appear to the right of the text content by coding the align="right" attribute on the tag. Note: The W3C HTML validator will indicate that the align attribute is invalid. We'll ignore the error for this case study. In Chapter 6, you'll learn to use the CSS float property (instead of the align property) to configure this type of layout.

2. Add a paragraph with the following text below the h2 element:

"Indulge in our locally roasted free-trade coffee and enjoy the aroma, the smooth taste, the caffeine! Join our Mug Club and get a 10% discount on each cup of coffee you purchase — ask the barista for details." Save and test your new menu.html page. It will be similar to Figure 4.48, but missing a few final touches (see Task 5).

Task 4: The Music Page. Use the Menu page as the starting point for the Music page. Launch a text editor, and open the menu.html file in the javajam4 folder. Save the file as music.html. Modify the music.html file to look similar to the Music page, as shown in Figure 4.49:



Figure 4.49 JavaJam music.html

1. Change the page title to an appropriate phrase.
2. Delete the image and description list from the page.
3. Configure "Music at JavaJam" as the text within the h2 element.
4. Configure the following as the text within the paragraph element:

"The first Friday night each month at JavaJam is a special night. Join us from 8 pm to 11 pm for some music you won't want to miss!"
5. The rest of the content in the page will consist of two areas describing music performances. The area describing each music performance consists of an h4 element, a div assigned to the class named details, and an image link.

January Music Performance:

- Configure an h4 element with the following text: January
- Code an opening div tag. Assign the div to the class named `details`.
- Configure the `melaniethumb.jpg` as an image link to `melanie.jpg`. Code appropriate attributes on the `` tag.
- Configure the following text within the div after the image link:
Melanie Morris entertains with her melodic folk style.

February Music Performance:

- Configure an h4 element with the following text: February
- Code an opening div tag. Assign the div to the class named `details`.
- Configure the `gregthumb.jpg` as an image link to `greg.jpg`. Code appropriate attributes on the `` tag.
- Configure the following text within the div after the image:
Tahoe Greg is back from his tour. New songs. New stories.

Save the `music.html` file. If you test your page in a browser, you'll notice that it looks different from Figure 4.49—you still need to configure style rules.

Task 5: Configure the CSS. Open `javajam.css` in a text editor. Edit the style rules as follows:

1. Modify the body element selector style rules. Configure `background.gif` as the background image.
2. Modify the style rules for the `wrapper` id. Configure the background color to be `#FEF6C2`. Configure a minimum width of `900px` (use `min-width`). Configure a maximum width of `1280px` (use `max-width`). Use the `box-shadow` property to configure a drop-shadow effect.
3. Modify the header element selector style rules. Remove the `text-align` declaration. Code a declaration to set the height to `150px`. Configure declarations to display the `javajamlogo.jpg` image as a background image without repeating.
4. Modify the `h1` element selector style rules. Remove the `line-height` declaration. Configure declarations to set the top padding to `45px`, left padding to `220px`, and font size to `3em`.
5. Modify the `nav` element selector style rules. Configure declarations for `1.5em` font size size and `10px` of top padding.
6. Code a new style rule to prevent the hyperlinks in the `nav` area from displaying the default underline. Use `nav a { text-decoration: none; }`
7. Modify the footer element selector style rules. Configure declarations for `10px` of bottom padding and a solid `2px` top border (use `#221811` as the color).
8. Add a new style rule for the `h4` element selector that configures a background color (`#D2B48C`), font size (`1.2em`), left padding (`10px`), and bottom padding (`5px`).
9. Add a new style rule for the `main` element selector to configure `2em` of padding on the left, right, and bottom. Also configure `display: block;` (this will prevent rendering issues in versions of Internet Explorer that do not support the HTML5 `main` element).

10. Add a new style rule for the class named `details` to add 20% left and right padding. Notice how this rule adds empty space on either side of the music performance description and image on the `music.html` page.
11. Add a new style rule for the `img` element selector that configures 10px left padding and 10 px right padding.

Save the `javajam.css` file. Test your pages (`index.html`, `menu.html`, and `music.html`) in a browser. If your images do not appear or your image links do not work, examine your work carefully. Use Windows Explorer or Mac Finder to verify that the images are saved in your `javajam4` folder. Examine the `src` attribute on the `` tags to be sure you spelled the image names correctly. Another useful troubleshooting technique is to validate the HTML and CSS code. See Chapters 2 and 3 for Hands-On Practice exercises that describe how to use these validators.

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 Fish Creek. The Home page and Services page were created in earlier chapters. Using the existing website as a starting point, you will modify the design of the pages and create a new page, the Ask the Vet page. You have five tasks in this case study:

1. Create a new folder for this Fish Creek case study, and obtain the starter image files.
2. Modify the Home page to display a logo image and navigation image links as shown in Figure 4.50.
3. Modify the Services page to be consistent with the Home page.
4. Create a new Ask the Vet page, as shown in Figure 4.51.
5. Modify the style rules in the `fishcreek.css` file as needed.

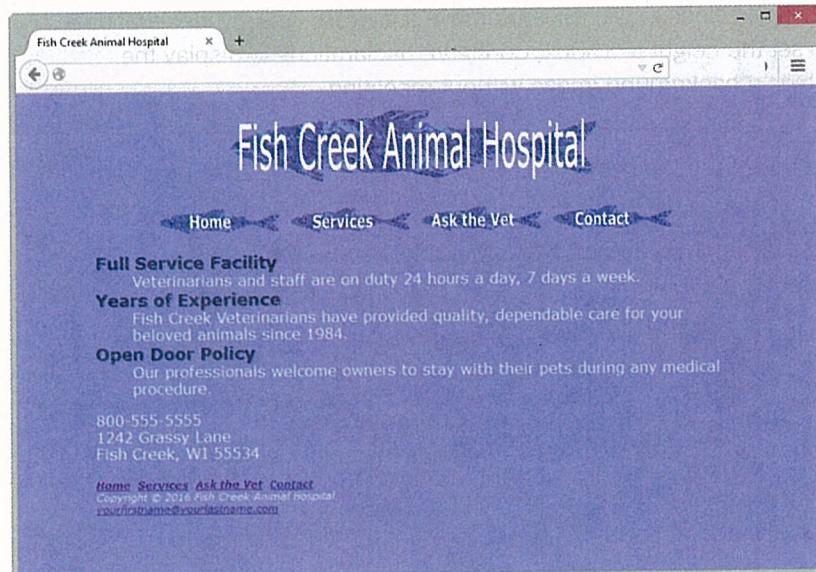


Figure 4.50 New Fish Creek Home page

Hands-On Practice Case

Task 1: Create a folder on your hard drive or portable storage device called fishcreek4. Copy all the files from your Chapter 3 fishcreekcss folder into the fishcreek4 folder. Obtain the images used in this case study from the student files. The images are located in the chapter4/casestudystarters/fishcreek folder. The images are fishcreeklogo.gif, home.gif, services.gif, askthevet.gif, and contact.gif. Save the files in your fishcreek4 folder.

Task 2: The Home Page. Launch a text editor, and open the index.html file from your fishcreek4 folder. Modify the index.html file to look similar to the web page shown in Figure 4.50.

1. Replace the “Fish Creek Animal Hospital” text contained within the h1 element with the fishcreeklogo.gif. Be sure to include the alt, height, and width attributes on the tag for the graphic.
2. Update the navigation area.
 - Since you will be replacing the top navigation with image links, it’s a good idea to provide for accessibility by including a set of text navigation links in the footer section of the web page. Copy the nav element, and paste it inside the footer area above the copyright line.
 - Refer to Figure 4.50, and replace the top navigation text hyperlinks with image links. The home.gif should link to index.html. The services.gif should link to services.html. The askthevet.gif should link to askvet.html. The contact.gif should link to contact.html. Use appropriate attributes on the tag: alt, height, and width.

Save and test your new index.html page. It will be similar to Figure 4.50, but you’ll notice that a few final touches (like the text shadow on the categories) are missing; you’ll configure these with CSS in Task 5.

Task 3: The Services Page. Launch a text editor, and open the services.html page from your fishcreek4 folder. Replace the “Fish Creek Animal Hospital” heading with the fishcreeklogo.gif. Configure the navigation areas in a similar way as the home page. Save and test your new services.html page.



Figure 4.51 Fish Creek askvet.html

Task 4: The Ask the Vet Page. Use the Services page as the starting point for the Ask the Vet page. Launch a text editor, and open the services.html file in the fishcreek4 folder. Save the file as askvet.html. Modify the askvet.html file to look similar to the Ask the Vet page, as shown in Figure 4.51:

1. Change the page title to an appropriate phrase.
2. Delete the unordered list from the page.
3. The page content consists of a paragraph of text followed by a description list that contains a question and an answer.

- a. Configure the text in the paragraph as follows:

Contact us if you have a question that you would like answered here.

- b. The word “Contact” should link to the contact.html page.
 - c. The description list displays the question and answer. The `<dt>` element configures the question. Assign the `<dt>` element to the `category` class used on the Services page. The `<dd>` element configures the answer. The content of the description list follows:

Question: Our dog, Sparky, likes to eat whatever the kids are snacking on. Is it OK for the dog to eat chocolate?”

Answer: Chocolate is toxic to dogs. Please do not feed your dog chocolate. Try playing a game with your children — when you feed them people treats, they can feed Sparky dog treats.

- d. *Hint:* See Appendix C, “Special Characters,” for the character code to display the em dash (—).

Save the askvet.html file. If you test your page in a browser, you’ll notice that it looks different from Figure 4.51—you still need to configure style rules.

Task 5: Configure the CSS. Open fishcreek.css in in a text editor. Edit the style rules as follows:

1. Modify the style rules for the `wrapper` id. Configure a minimum width of 700px (use `min-width`). Configure a maximum width of 1024px (use `max-width`).
2. Modify the style rules for the `h1` element selector. Delete the `line-height` style declaration. Add a new declaration to center the `h1` content (use `text-align:center`).
3. Modify the style rules for the `nav` element selector. Add a new declaration to center the text (use `text-align: center`).
4. Modify the `category` class to display text with a drop shadow (use `text-shadow: 1px 1px 1px #667`).
5. Add a new style rule for the `img` element selector to display no border.
6. Add a new style rule for the navigation in the footer area (use `footer nav` as the selector) to overrule the previous `nav` style rule and configure text to be left-aligned (use `text-align: left`).

Save the fishcreek.css file. Test your pages (index.html, services.html, and askvet.html) in a browser. If your images do not appear or your image links do not work, examine your work carefully. Use Windows Explorer or Mac Finder to verify that the images are saved in your fishcreek4 folder. Examine the `src` attribute on the `` tags to be sure you

spelled the image names correctly. Another useful troubleshooting technique is to validate the HTML and CSS code. See Chapters 2 and 3 for Hands-On Practice exercises that describe how to use these validators.

Pacific Trails

See Chapter 2 for an introduction to the Pacific Trails Case Study. Figure 2.38 shows a site map for Pacific Trails. The Home page and Yurts page were created in earlier chapters. Using the existing website as a starting point, you will modify the design of the pages to display a large image on each page, as indicated in the wireframe in Figure 4.52. You will also create a new page, the Activities page. You have five tasks in this case study:

1. Create a new folder for this Pacific Trails case study, and obtain the starter image files.
2. Modify the Home page to display a logo image and scenic photograph as shown in Figure 4.53.
3. Modify the Yurts page to be consistent with the Home page.
4. Create a new Activities page, as shown in Figure 4.54.
5. Modify the style rules in the pacific.css file as needed.

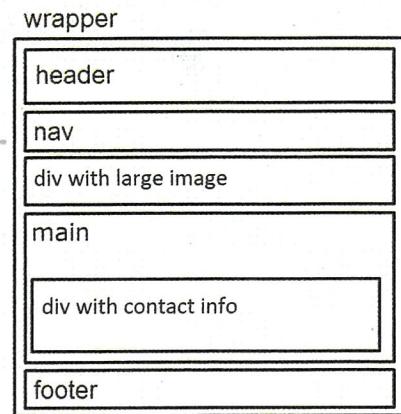


Figure 4.52 New Pacific Trails wireframe

Hands-On Practice Case

Task 1: Create a folder on your hard drive or portable storage device called pacific4.

Copy all the files from your Chapter 3 pacificcss folder into the pacific4 folder. Obtain the images used in this case study from the student files. The images are located in the chapter4/casestudystarters/pacific folder. The images are sunset.jpg, coast.jpg, yurt.jpg, trail.jpg, and background.jpg. Save the files in your pacific4 folder.

Task 2: The Home Page. Launch a text editor, and open the index.html file from your pacific4 folder. Modify the index.html file to look similar to the web page shown in Figure 4.53. Configure a div element to display the coast.jpg image. Code an opening div tag assigned to the id named `homehero` after the closing nav tag. Next, code a closing div tag. As shown in the wireframe in Figure 4.52, this div is located between the nav element and the main element. There is no HTML or text content for this div. The purpose of this div is to display a large image (configured with CSS in Task 5).

Save and validate your new index.html page. It will not yet be similar to Figure 4.53; you'll configure CSS in Task 5.

Task 3: The Yurts Page. Launch a text editor, and open the yurts.html page from your pacific4 folder. Configure a div element to display the yurt.jpg image. Code an opening div tag assigned to the id named `yurthero` after the closing nav tag. Next, code a closing div tag. As shown in the wireframe in Figure 4.52, this div is located between the nav element and the main element. There is no HTML or text content for this div. The purpose of this div is to display a large image (configured with CSS in Task 5). Save and validate your new yurts.html page.

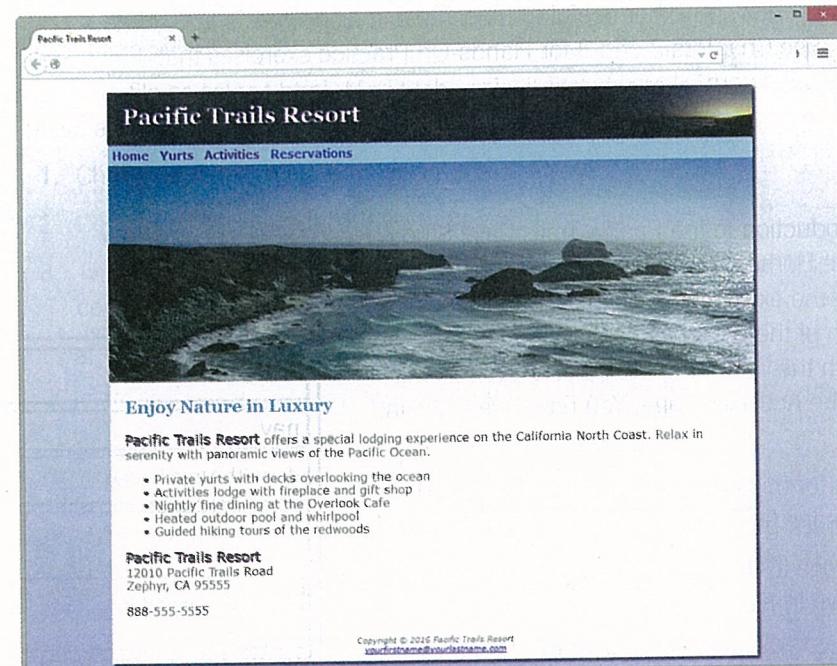


Figure 4.53 New Pacific Trails Resort Home page

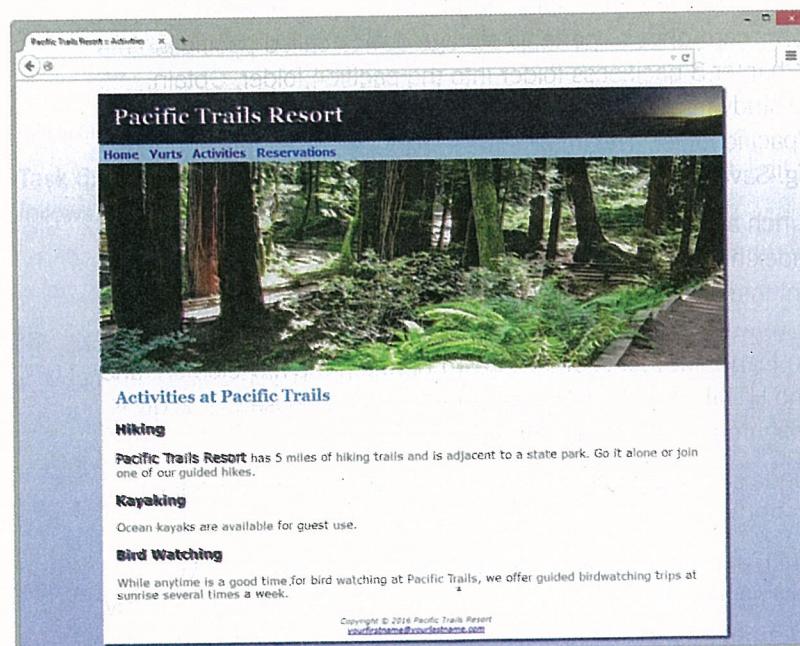


Figure 4.54 Pacific Trails Resort activities.html

Task 4: The Activities Page. Use the Yurts page as the starting point for the Activities page. Launch a text editor, and open the yurts.html file in the pacific4 folder. Save the file as activities.html. Modify the activities.html file to look similar to the Activities page, as shown in Figure 4.54:

1. Change the page title to an appropriate phrase.
2. Change the text in the <h2> to be “Activities at Pacific Trails”.
3. Modify the div assigned to the id *yurthero*. Replace *yurthero* with *trailhero*.
4. Delete the description list from the page.
5. Configure the following text, using h3 tags for the headings and paragraph tags for the sentences:

Hiking

Pacific Trails Resort has 5 miles of hiking trails and is adjacent to a state park. Go it alone or join one of our guided hikes.

Kayaking

Ocean kayaks are available for guest use.

Bird Watching

While anytime is a good time for bird watching at Pacific Trails, we offer guided bird-watching trips at sunrise several times a week.

6. Configure a span element to contain the phrase “Pacific Trails Resort” in the first paragraph on the page. Assign the span to the class named *resort*.

Save the activities.html file. If you test your page in a browser, you’ll notice that it looks different from Figure 4.54; you still need to configure style rules.

Task 5: Configure the CSS. Open pacific.css in in a text editor. Edit the style rules as follows:

1. Modify the body element selector style rules. Configure background.jpg as the background image.
2. Modify the style rules for the *wrapper* id. Configure the background color to be #eeeeee. Configure a minimum width of 700px (use *min-width*). Configure a maximum width of 1024px (use *max-width*). Use the *box-shadow* property to configure a drop-shadow effect.
3. Modify the style rules for the h1 element selector. Configure the sunset.jpg as a background image that displays on the right and does not repeat. Also configure 20 pixels of left padding. Configure a 72-pixel height (the same height as the background image).
4. Modify the style rules for the nav element selector. Configure 5 pixels of padding.
5. Modify the style rules for the footer element selector. Configure 10 pixels of padding.
6. Add a new style rule for the h3 element selector to display #000033 text color.
7. Add a new style rule for the main element selector that configures 20 pixels of left and right padding. Also configure *display: block;* (this will prevent rendering issues in versions of Internet Explorer that do not support the HTML5 main element).

8. Add a new selector for an id named `homehero`. Code declarations to configure 300px height and to display the `coast.jpg` background image to fill the space (use `background-size: 100% 100%;`) without repeating.
9. Add a new selector for an id named `yurthero`. Code declarations to configure 300px height and to display the `yurt.jpg` background image to fill the space (use `background-size: 100% 100%;`) without repeating.
10. Add a new selector for an id named `trailhero`. Code declarations to configure 300px height and to display the `trail.jpg` background image to fill the space (use `background-size: 100% 100%;`) without repeating.
11. Code a new style rule to prevent the hyperlinks in the `nav` area from displaying the default underline. Use `nav a { text-decoration: none; }`
12. Have you noticed extra empty space between the logo header area and the navigation? Let's do something about that. We'll need to use the CSS margin property, which you'll explore in depth in Chapter 6. Modify the style rules for the `h1` selector to set the bottom margin to 0 with the following code:

```
margin-bottom: 0;
```

Save the `pacific.css` file. Test your pages (`index.html`, `yurts.html`, and `activities.html`) in a browser. The Home page (`index.html`) should look similar to Figure 4.53. The new Activities page (`activities.html`) should look similar to Figure 4.54. If your images do not appear, examine your work carefully. Use Windows Explorer or Mac Finder to verify that the images are saved in your `pacific4` folder. Examine the `src` attribute on the `` tags to be sure you spelled the image names correctly. Another useful troubleshooting technique is to validate the HTML and CSS code. See Chapters 2 and 3 for Hands-On Practice exercises that describe how to use these validators.

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 Path of Light Yoga Studio. The Home page and Classes page were created in earlier chapters. Using the existing website as a starting point, you will modify the design of the pages and create a new page, the Schedule page. You have five tasks in this case study:

1. Create a new folder for this Path of Light Yoga Studio case study, and obtain the starter image files.
2. Modify the Home page to display as shown in Figure 4.55.
3. Modify the Classes page to display as shown in Figure 4.57.
4. Create a new Schedule page, as shown in Figure 4.58.
5. Modify the style rules in the `yoga.css` file as needed.

Hands-On Practice Case

Task 1: Create a folder on your hard drive or portable storage device called `yoga4`. Copy all the files from your Chapter 3 `yogacss` folder into the `yoga4` folder. Obtain the images used in this case study from the student files. The images are located in the `chapter4/casestudystarters/yoga` folder. The images are `lilyheader.jpg`, `yogadoor.jpg`, `yogalounge.jpg`, and `yogamat.jpg`. Save the files in your `yoga4` folder.

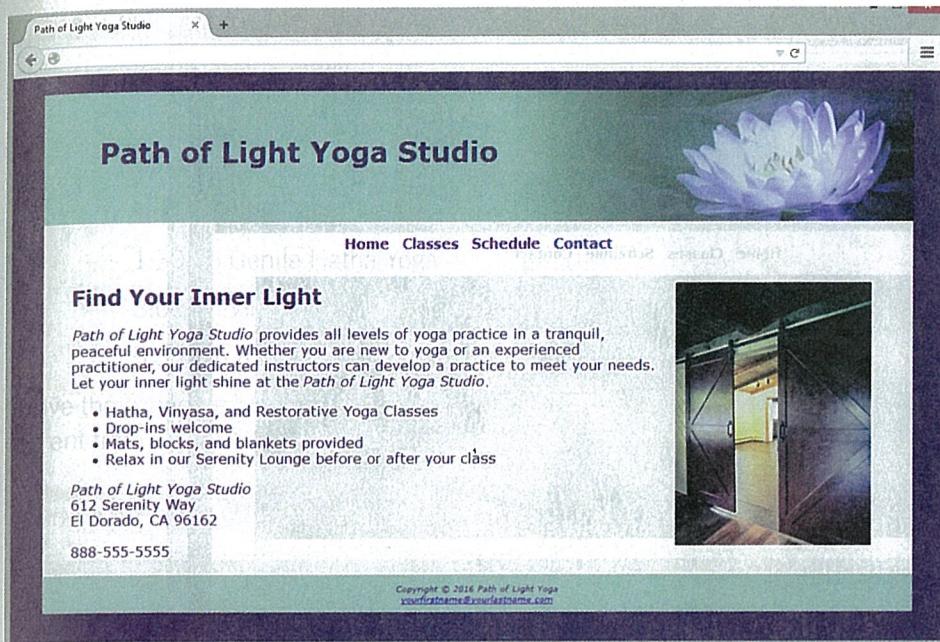


Figure 4.55 Path of Light Yoga Studio Home page

Task 2: The Home Page. Launch a text editor, and open the index.html file from your yoga4 folder. Modify the index.html file to look similar to the web page shown in Figure 4.55.

Code an img tag for the yogadoor.jpg image above the h2 element in the main content area. Be sure to include the alt, height, and width attributes. Also configure the image to appear to the right of the text by coding the align="right" attribute on the tag. Note: The W3C HTML validator will indicate that the align attribute is invalid. We'll ignore the error for this case study. In Chapter 6, you'll learn to use the CSS float property (instead of the align property) to configure this type of layout.

Save and test your new index.html page. It will be similar to Figure 4.55 but you'll notice that a few final touches (including dark page background and lily image in the header) are missing; you'll configure these with CSS in Task 5.

Task 3: The Classes Page. It's common for the content pages of a website to have a slightly different structure than the home page. The wireframe shown in Figure 4.56 depicts the structure of the Classes and Schedule pages. Launch a text editor, and open the classes.html page from your yoga4 folder. Configure a div element to display the yogamat.jpg image. As shown in the wireframe in Figure 4.56, this div is located within the main element. Code an opening div tag after the opening main tag. Assign the div to an id named hero. Code an img tag for the yogamat.jpg image. Be sure to include the alt, height, and width attributes. Next, code a closing div tag. Save and test your new classes.html page. If you test your page in a browser, you'll notice that it looks a bit different from Figure 4.57; you still need to configure style rules.

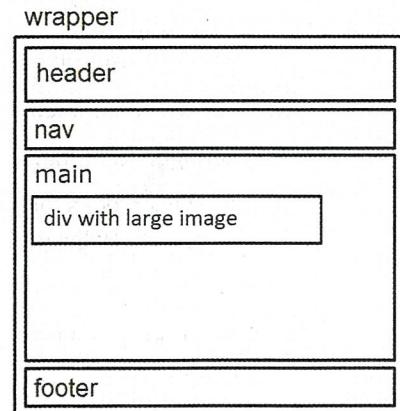


Figure 4.56 New Path of Light Yoga Studio wireframe

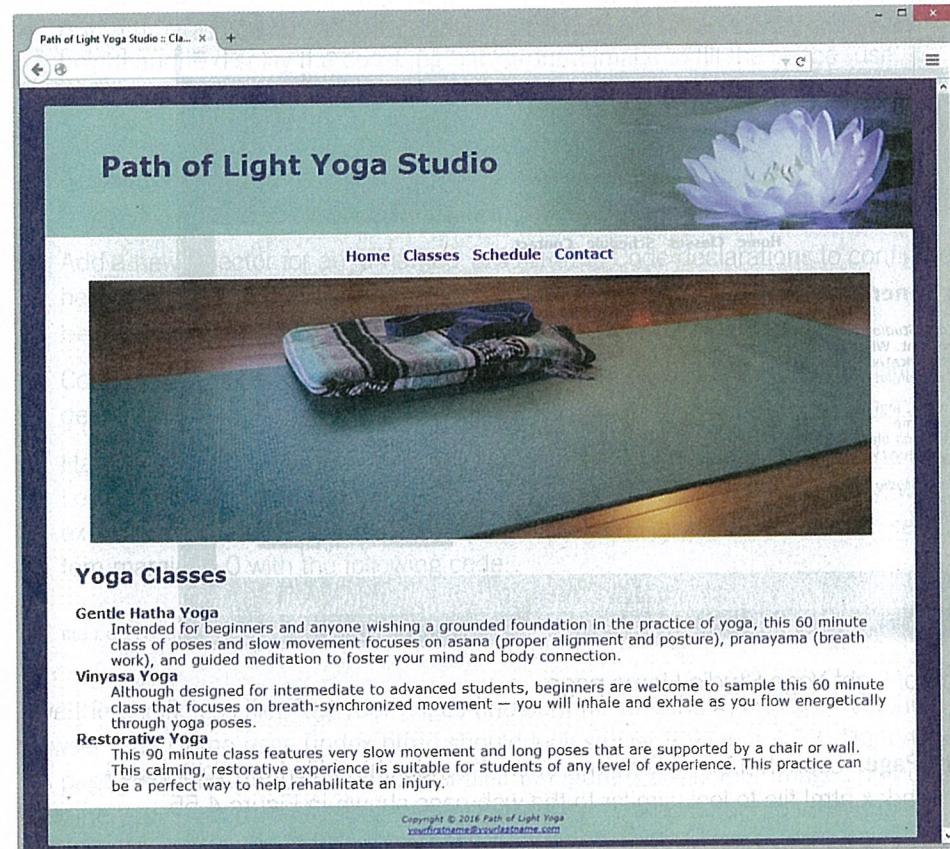


Figure 4.57 Path of Light Yoga Studio Classes Page

Task 4: The Schedule Page. Use the Classes page as the starting point for the Schedule page. Launch a text editor, and open the classes.html file in the yoga4 folder. Save the file as schedule.html. Modify your file to look similar to the Schedule page, as shown in Figure 4.58.

1. Change the title to an appropriate phrase.
2. Modify the img tag to display the yogalounge.jpg image. Configure appropriate alt text.
3. Replace the h2 element text "Yoga Classes" with "Yoga Schedule".
4. Delete the description list.
5. Configure content for the Schedule Page page.
 - Configure a paragraph element that contains the following text:

Mats, blocks, and blankets provided. Please arrive 10 minutes before your class begins. Relax in our Serenity Lounge before or after your class.
 - Configure an h3 element with the following text:

Monday — Friday
 - Configure an unordered list with the following text:

9:00am Gentle Hatha Yoga
10:30am Vinyasa Yoga
5:30pm Restorative Yoga
7:00pm Gentle Hatha Yoga

- Configure an h3 element with the following text:

Saturday & Sunday

- Configure an unordered list with the following text:

10:30am Gentle Hatha Yoga

Noon Vinyasa Yoga

1:30pm Gentle Hatha Yoga

3:00pm Vinyasa Yoga

5:30 pm Restorative Yoga

Save the schedule.html file. If you test your page in a browser, you'll notice that it looks different from Figure 4.58; you still need to configure style rules.

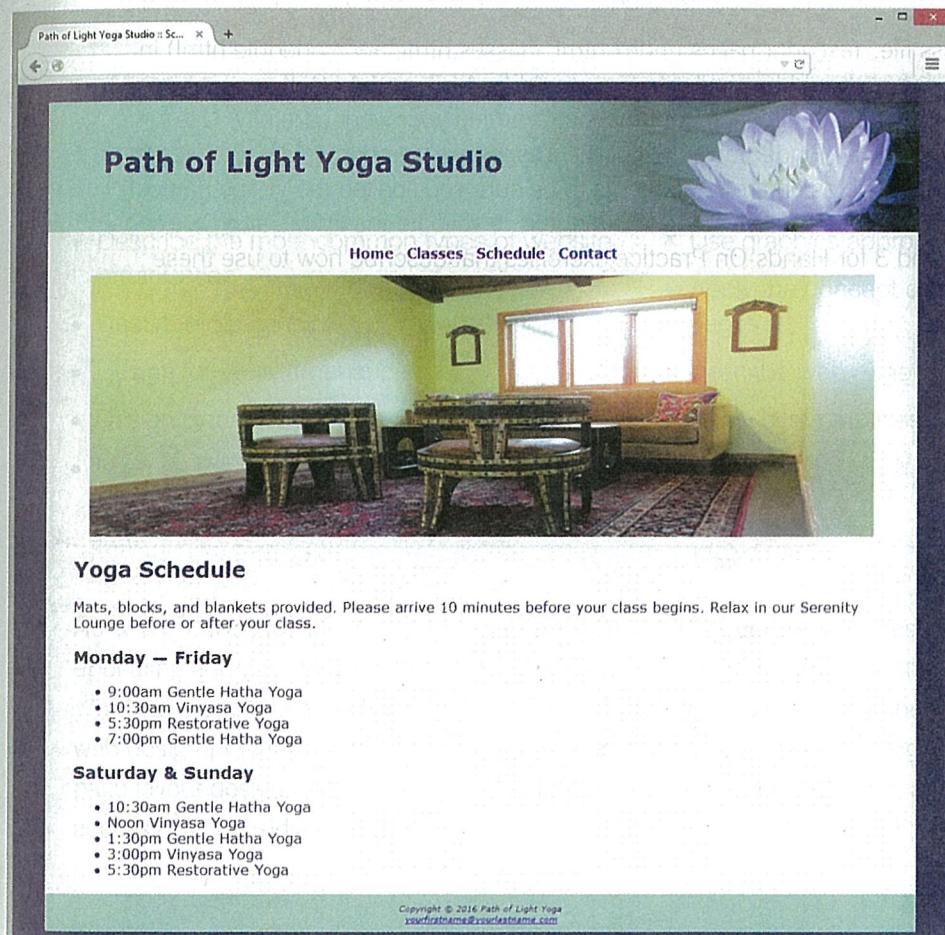


Figure 4.58 Path of Light Yoga Studio schedule.html

Task 5: Configure the CSS. Open yoga.css in a text editor. Edit the style rules as follows:

1. Modify the style rules for the body element selector to configure a very dark background color (#3F2860).
2. Modify the style rules for the #wrapper id. Configure #F5F5F5 as the background color. Configure a minimum width of 1000px (use min-width) and, a maximum width of 1280px (use max-width).

3. Modify the style rules for the header element selector. Remove the `text-align` declaration. Configure `lilyheader.jpg` as a background image that displays on the right without repeating. Set height to 150px.
4. Modify the style rules for the `h1` element selector. Remove the `line-height` declaration. Configure 50px top padding and 2em left padding.
5. Modify the style rules for the `nav` element selector. Configure 1em padding.
6. Modify the style rules for the `footer` element selector. Configure 1em padding.
7. Configure styles for the `main` element selector. Set left and right padding to 2em. You may need to nudge Internet Explorer to display the page as intended by adding the `display: block;` declaration (see Chapter 6).
8. Configure styles for the `img` element selector. Set left and right padding to 1em.
9. Configure styles for an id selector named `hero`. Set `text-align` to center.

Save the `yoga.css` file. Test your pages (`index.html`, `classes.html`, and `schedule.html`) in a browser. Your pages should be similar to Figures 4.55, 4.57, and 4.58. If your images do not appear or your image links do not function, examine your work carefully. Use Windows Explorer or Mac Finder to verify that the images are saved in your `yoga` folder. Examine the `src` attributes on the `` tags to be sure you spelled the image names correctly. Another useful troubleshooting technique is to validate the HTML and CSS code. See Chapters 2 and 3 for Hands-On Practice exercises that describe how to use these validators.