

# Focus on Web Design

In this chapter, you have learned how to configure color and text with CSS. In this activity, you will design a color scheme, code an external CSS file for the color scheme, and code an example web page that applies the styles you configured. Use any of the following sites to help you get started with color and web design ideas:

## Psychology of Color

- <https://www.infoplease.com/spot/colors1.html>
- <https://www.empower-yourself-with-color-psychology.com/meaning-of-colors.html>
- <https://www.designzzz.com/infographic-psychology-color-web-designers>

## Color Scheme Generators

- <https://meyerweb.com/eric/tools/color-blend>
- <http://www.colr.org>
- <https://color.adobe.com/create/color-wheel>
- <http://paletton.com>

Complete the following tasks:

- a. Design a color scheme. List three hexadecimal color values in addition to neutral colors such as white (`#FFFFFF`) or black (`#000000`) in your design.
- b. Describe the process you went through as you selected the colors. Describe why you chose these colors. For what type of website would they be appropriate? 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 Bar

See [Chapter 2](#) for an introduction to the JavaJam Coffee Bar Case Study. [Figure 2.32](#) 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 2.33](#) 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: THE WEBSITE FOLDER.** 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 Tahoma, Arial, or any sans-serif font.
2. Styles for the header element selector that configure text color #8C3826, background color #D2B48C, and centered text.
3. Styles for the h1 element selector that configure 200% line height.
4. Styles for the h2 element selector that configure #8C3826 text color.
5. Styles for the nav element selector that configure centered, bold text. *Hint:* Use the CSS `text-align` and `font-weight` properties.
6. 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 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 and 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 the background color and alignment of your page content is different. 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 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 background and alignment.

**Figure 3.30** New menu.html page



**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%, `#FEFAEB` background color, `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?

## Fish Creek Animal Clinic

See [Chapter 2](#) for an introduction to the Fish Creek Animal Clinic Case Study. [Figure 2.36](#) 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 2.37](#) 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](#).

---

**Figure 3.31** New Fish Creek index.html

---

#FFFFFF



#F0F0F0



#5280C5



#AEC3E3



#003366



4. Modify the Services page to be consistent with the new Home page.

5. Configure centered page layout.

## Hands-On Practice Case Study

**TASK 1: THE WEBSITE FOLDER.** 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 #FFFFFF; text color #003366; and Verdana, Arial, or any sans-serif font.
2. Styles for the header element selector that configure background color #003366, text color #AFC3F3 and a serif font such as Georgia or Times New Roman.

#AEC3E3, and a serif font such as Georgia or Times New Roman.

3. Styles for the h1 element selector that configure 200% line height and set the text-indent property to 1em.
4. Styles for the h2 element selector that configure 1.2em font size.
5. Styles for the nav element selector that display text in bold.
6. Styles for the dt element selector that configure #5280C5 text color, size 1.1em font in bold, and a serif font such as Georgia or Times New Roman.
7. Styles for a class named `category` with bold font, text color #5380C5, and a serif font such as Georgia or Times New Roman.
8. 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 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 the content area. Remove the tags from within each dt element, because the CSS will configure the bold font style.
4. Configure the page footer area. Remove the and 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 the background color and alignment of your page content will be different. Don't worry—you'll finish configuring 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 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 and background color.

**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 with a light gray background color. Refer to Hands-On Practice 3.9 if necessary.

1. Launch a text editor, and open the fishcreek.css file. Add a style rule for an id named `wrapper` with `width` set to 90%, `margin-right` set to auto, `margin-left` set to auto, and background color set to #F0F0F0.
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 3.31.
3. 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

