# JavaJam Coffee House Case Study – Textbook - Chapter 8

# In this chapter’s case study, you will use the existing JavaJam (Chapter 7) website as a starting point to create a new version of the website that utilizes media queries to configure display for mobile devices. Figure 8.35 displays wireframes for desktop browser, typical table screen, and typical smartphone screen display. When you have finished, the website will look the same in desktop browsers (see Figures 8.36). The mobile displays should be similar to the screen captures in 8.36.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DESKTOP BROWSERwrapper  |  |  |  | | --- | --- | --- | | header | | | |  | |  | | nav | main | | |  | |  | | footer | | | | TABLET DISPLAYwrapper  |  | | --- | | header | |  | | nav | |  | | main | |  | | footer | | SMARTPHONE DISPLAYwrapper  |  | | --- | | header | |  | | nav | |  | | main | |  | | footer | |

# FIGURE 8.35 JavaJam wireframes

# You have five tasks in this case study:

# Copy yourname\_chapter6 JavaJam Coffee House website folder and rename it yourname\_chapter8.

# Edit the javajam.css external style sheet to include media queries and styles needed for appropriate desktop, tablet, and smartphone display.

# Edit the Home page: index.html.

# Edit the Menu page: menu.html.

# Edit the Music page: music.html.

# 

# Desktop Display Tablet Display Smartphone

# FIGURE 8.36 Resize the browser window to approximate the new tablet and smartphone display; desktop, tablet, and mobile.

# Task 1: Copy yourname\_chapter6 JavaJam Coffee House website folder and rename it yourname\_chapter8. Copy the javalogomobile.gif from the student files from the chapter8/starters folder into yourname\_chapter8 folder.

# Task 2: Move all the images files except the favorites icon into a new subfolder called images. Move the stylesheet into a new subfolder called css. Adjust all the links to the CSS file to point to the CSS file’s new location. Adjust all the links within HTML pages and the stylesheet that references images to point to the new location for images using relative paths.

# Task 3: Configure the CSS. Launch a text editor and open the javajam.css external style sheet file.

# Configure Support of HTML5. Add the following style rule to configure most older browsers to render HTML5 block display elements as expected.

# Header, main, nav, footer, figure, figcaption, aside, section, article {display:block;}

# Configure Phone Number Desktop Display. Code the following new styles.

# The mobile class. Set display to none. You’ll apply this class when you edit the home page. (index.html). Your CSS selector will be: .mobile.

# The desktop class. Set display to inline. You’ll apply this class later when you edit the home page. (index.html). Your CSS selector will be: .desktop.

# Configure article. Review the CSS and locate the main article selector and ensure that the overflow is set to auto, and font-size to 90%.

# Configure Tablet Display.

# Code a media query below the existing styles to select for typical tablet device viewport size.

# @media only screen and (max-width: 64em) {

# }

# Configure the following new styles within the media query:

# The body element selector. Set margin to 0 the background color to #D2B48C, and the background image to none.

# The wrapper id selector. Set minimum width to 0. Set width to auto, and box-shadow to none.

# The h1 element selector. Set margin to 0 and configure javalogomobile.gif as the background image.

# The main element selector. Set left margin to 0.

# The nav element selector. Eliminate float. (Hint: Use float: none;), set the width to auto, and configure 0 padding.

# Navigation unordered list. Use a descendant selector to configure ul elements within the navigation area to display centered text.

# Navigation list items. Use a descendant selector to configure li elements within the navigation area with inline display, 0 top and bottom padding, and .75em left and right padding.

# Configure Smartphone Display.

# Code a media query below the existing styles to select for typical smartphone device viewport size.

# @media only screen and (max-width: 37.5em) {

# }

# Configure the following new styles within the media query:

# Navigation list items. Use a descendant selector to configure li elements within the navigation area with inline-block display, 5em width, 120% font-size, centered text, a 2px box shadow (use #330000), #F5F5DC background color, 1% margin, and 2.5% padding.

# Navigation hyperlinks. Use a descendant selector to configure anchor elements within the navigation area with block display. This will provide the user a larger area to tap when selecting a hyperlink. Also configure #333300 text color for the navigation hyperlinks :hover pseudoclass.

# Content hyperlinks. Use descendant selectors to configure :link (#FFFFFF), :visited (#EAEAEA), and :hover (#FFD700) text color for hyperlinks in the main content area.

# The article element selector. Configure article elements to have 0 margin.

# Article images. Use a descendant selector to configure img elements within an article element to not display.

# The hero id selector. Set the height to auto.

# The mobile class. Set display to inline. You’ll apply this class when you edit the home page. (index.html)

# The desktop class. Set display to none. You’ll apply this class later when you edit the home page. (index.html)

# Save the file as javajam.css in the yourname\_chapter8 folder. Check your syntax with the CSS validator at <http://jigsaw.w3.org/css-validator>. Correct and retest if necessary.

# Task 4: Edit the Home Page. Launch a text editor and open the home page, index.html. Edit the code as follows:

# Configure a viewport meta tag in the head section that configures the width to the device-width and sets the initial-scale to 1.0.

# The home page displays a phone number in the contact information area. Wouldn’t it be handy if a person using a smartphone could click on the phone number to call the coffee house? You can make that happen us using tel: in a hyperlink. Configure a hyperlink assigned to a class named mobile that contains the phone number as shown below:

# <a class=”mobile” href=”<tel:888-555-5555>”>888-555-5555</a>

# But wait a minute, a telephone link could confuse those visiting the site with a desktop browser. Code another phone number directly after the hyperlink. Code a span element assigned to a class named desktop around the phone number as shown here:

# <span class=”desktop”>888-555-5555</span>

# Save the index.html file. Remember that validating your HTML can help you find syntax errors. Validate and correct this page before you continue. Display your page in a browser. While your home page will look unchanged in maximized desktop browsers, as you resize and reduce the browser viewport dimensions, the displays should be similar to the screen captures in Figure 8.36.

# Task 5: Edit the Menu Page. Launch a text editor, and open the menu.html file. Edit the code as follows:

# Configure a viewport meta tag in the head section that configures the width to the device-width and sets the initial-scale to 1.0.

# Save the menu.html file. Remember that validating your HTML can help you find syntax errors. Validate and correct this page before you continue. Display your page in a browser. Resize the browser window to test the media queries. Compare your work with Figure 8.37, which shows screen captures of the menu page.

# 

# 

# FIGURE 8.37 Browser approximation of the menu.html page display; desktop, tablet, and mobile.

# Task 5: Edit the Music Page. Launch a text editor, and open the music.html file. Edit the code as follows:

# Configure a viewport meta tag in the head section that configures the width to the device-width and sets the initial-scale to 1.0.

# 

# FIGURE 8.38 Browser approximation of the music.html page display; desktop, tablet, and mobile.

# 

# Save the music html file. Remember that validating your HTML can help you find syntax errors. Validate and correct this page before you continue. Display your page in a browser. Resize the browser window to test the media queries. Compare your work with Figure 8.38, which shows screen captures of the music page. JavaJam Coffee House is mobile!

# Continue to the next case study on the following page.

# JavaJam Coffee House Case Study – Textbook - Chapter 9

# In this chapter’s case study, you will use the existing JavaJam (Chapter 8) website as a starting point to and modify the Menu page to display information in a table. Your new page will be similar to Figure 9.16 when you have completed this case study.

|  |
| --- |
| You have three tasks in this case study:Use yourname\_chapter8 JavaJam website folder.Modify the style sheet (javajam.css) to configure style rules for the new table.Modify the Menu page to use a table to display information as shown in Figure 9.16. |

# FIGURE 9.16 Menu page with a table.

# Task 1: Use yourname\_chapter8 JavaJam website folder.

# Task 2: Configure the CSS. You will add styles to configure the table on the Menu page. Launch a text editor and open the javajam.css external style sheet file. You will add the new style rules above the media queries.

# Configure the table. Code a new style rule for the table element selector that configures a centered table (use margin: auto;) with an 80% width.

# Configure the table cells. Code a new style rule for the td and th element selectors that configures 10 pixels of padding.

# Configure alternate-row background color. The table looks more appealing if the rows have alternate background colors but is still readable without them. Apply the :nth-of-type CSS3 pseudo-class to configure the odd table rows with a medium brown background color (#D2B48C).

# Save your javajam.css file. Use the CSS Validator (<http://jigsaw.w3.org/css-validator>) to check your syntax. Correct and retest if necessary.

# Task 3: Update the Menu pages. Open the menu.html page for the JavaJam Coffee House website in a text editor. The menu descriptions are configured with a description list. Replace the description list with a table that has three rows and two columns. Use th and td elements where appropriate. Save you page and test it in a browser. If the page does not display as you intended, review your work, validate the CSS, validate the HTML, modify as needed, and test again.