Focus on Web Design

Now that you've had some practice creating responsive web pages, it's a good idea to explore resources on the web about responsive web design best practices. Use the following URLs as a starting point as you research this topic. Write a one-page, double-spaced summary that describes four recommended practices of responsive web design.

- https://www.smashingmagazine.com/2018/02/media-queries-responsive-design-2018/
- https://www.uxpin.com/studio/blog/best-practices-examples-of-excellent-responsive-design/
- https://www.impactbnd.com/blog/responsive-design-best-practices
- https://crossbrowsertesting.com/blog/development/future-responsive-design-2019/
- https://fireart.studio/blog/how-to-design-responsive-website-best-practices/

Website Case Study Modern, Responsive Layout

Each of the following case studies continues throughout most of the text. This chapter configures the website with a modern, responsive layout.

JavaJam Coffee Bar Case Study

In this chapter's case study, you will use the existing JavaJam Coffee Bar (Chapter 6) website as a starting point to create a new version with a responsive layout that implements media queries. You'll practice a Mobile First strategy for responsive design. First, you will configure a page layout that works well in smartphones (test with a small browser window). Then you'll resize the browser viewport to be larger until the design "breaks" and code media queries and additional CSS as needed. Figure 7.54 shows wireframes for three different layouts. The Home page displays will be similar to Figure 7.55.

Figure 7.54 JavaJam Coffee Bar wireframes

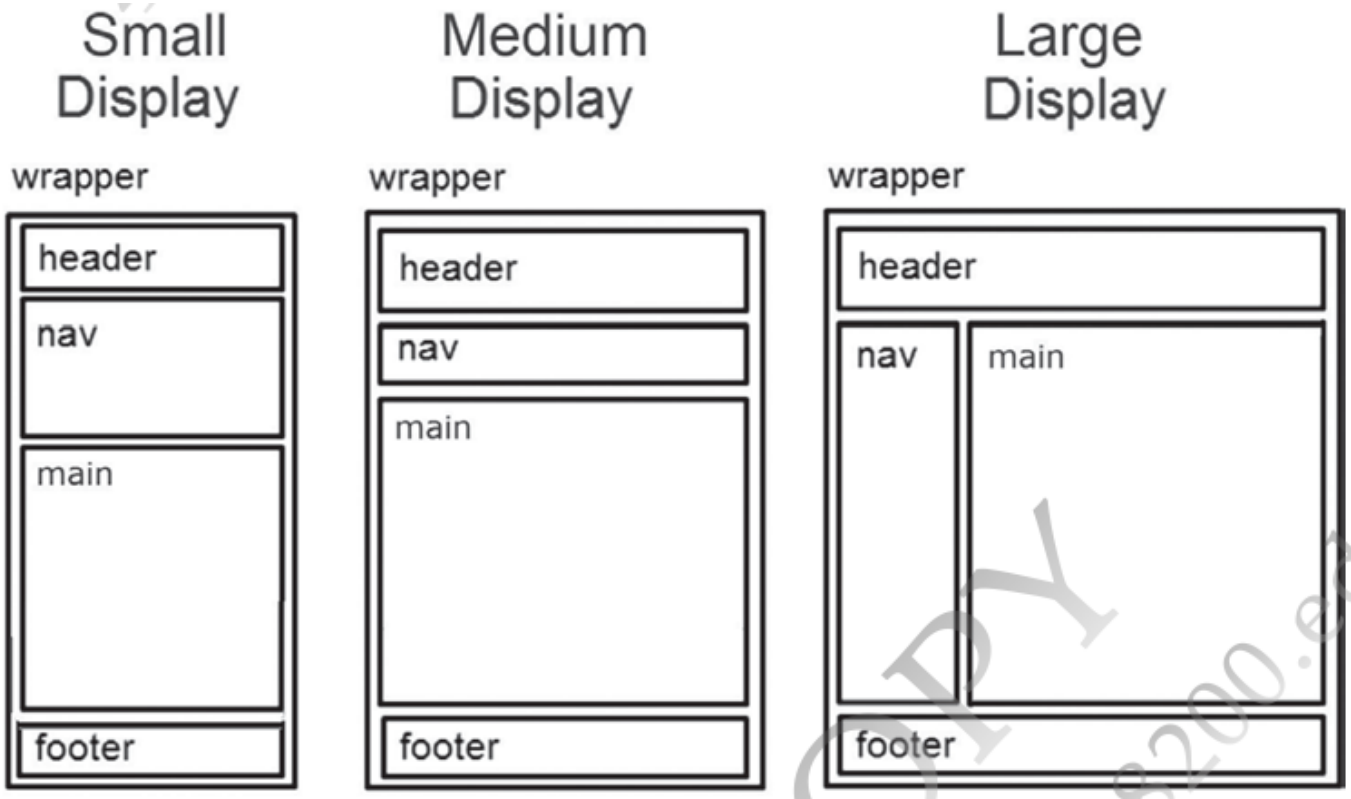

Figure 7.55 **The Home page**

You have four tasks in this case study:

1. Create a new folder for the JavaJam Coffee Bar website.
2. Configure the HTML and CSS needed for pleasing display of a single-column (smartphone) display.

3. Configure the HTML and CSS needed for pleasing display of the web pages on medium sized mobile devices.
4. Configure the CSS needed for a pleasing display of the web pages on large mobile devices and desktops.

TASK 1: Create a folder called javajam7 to contain your JavaJam Coffee Bar website files. Copy the files from the Chapter 6 Case Study javajam6 folder into the javajam7 folder. Copy the files from the chapter7/starters/javajam folder into the javajam7 folder.

TASK 2: CONFIGURE A SMALL SINGLE-COLUMN LAYOUT. First, you will edit the CSS. Then you will edit the Home page and test each page in a browser.

CONFIGURE THE CSS. Launch a text editor and open the javajam.css style sheet. Edit the styles to achieve a layout that displays well on small devices using normal flow (no floats) with full-width block elements.

1. Edit the styles for the body element selector. Remove all declarations for the background image. Set margin to 0 and background color to #D2B48C.
2. Edit the styles for the `wrapper` id selector. Remove all declarations associated with width, margin, and box-shadow.
3. Edit the styles for the header element selector. Change the background image to cup.jpg. Set left padding to 105px and height to 128px.
4. Edit the styles for the h1 element selector. Set the font size to 2em.
5. Edit the styles for the nav element selector. Remove the declarations that configure float, width, font-weight, and padding.
6. Edit the styles for the `nav ul` selector. Configure this selector as a flex container with `flex-direction` set to column. Set 0 margin, 0 padding, and 1.25em font size.
7. Code styles for the `nav li` selector. Set .5em top and bottom padding, 1em left and right padding, 100% width, and a 1px solid bottom border.
8. Remove the style declarations for the `onethird` class selector and `floatleft` class selector.
9. Edit the styles for the main element selector. Remove the declarations that configure margin and overflow.
10. Edit the styles for the hero images. Set the background image for the `homehero` id selector to road.jpg. Set the background image for the `heroguitar` id selector to guitar.jpg. Set the background image for the `heromugs` id selector to threemugs.jpg.
11. Edit the styles for the h3, h3, h4, p, div, and dl element selectors within the main element. Change the left and right padding to 1em.

12. Code styles for the `main ul` selector. Set left padding to 2em;
13. Configure styles for the telephone number to display a hyperlink when on the small display and display plain text otherwise.
    a. Code a style rule for the `mobile` id selector. Set display to `inline`.
    b. Code a style rule for the `desktop` id selector. Set display to `none`.

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

CONFIGURE THE HTML. Modify the pages as indicated.

1. Launch a text editor and open index.html. Save the file when you have completed the following edits.
   a. 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 resort? You can make that happen by using tel: in a hyperlink. Configure a hyperlink assigned to an id named `mobile` that contains the phone number as shown:

```
888-555-5555
```

   However, 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 an id named `desktop` around the phone number as shown:

```
888-555-5555
```

   b. Code a viewport meta tag in the head section that configures the width to the device-width and sets the initial-scale to 1.0.
2. Add a viewport meta tag to the menu.html and music.html files in the same manner as the Home page. Save your files.

TEST THE WEB PAGES. Display your index.html file in a browser. This layout is intended for narrow mobile screens. Resize your browser to be narrower until your display is similar to the Small Display shown in Figure 7.55, which simulates mobile display. Test the menu.html and music.html files in a similar manner.

TASK 3: CONFIGURE A MEDIUM LAYOUT. Edit the CSS and the content pages to configure a more pleasing display on a wider viewport, setting 600px as the breakpoint for the first media query. When you test your web pages and trigger the media query, the layout in the Medium Display wireframe in Figure 7.54 will be implemented and your pages should look similar to the Medium Display in Figures 7.55, 7.56, and 7.57.

Figure 7.56 The Menu page

Figure 7.57 The Music page

CONFIGURE THE CSS. Launch a text editor and open the javajam.css style sheet. Place your cursor below the existing styles. Code a media query that is triggered when the minimum width is 600px or greater. Code the following styles within the media query.

1. Code styles for the header element selector. Configure centered text and 0 left padding.
2. Code styles for the h1 element selector. Set font size to 3em.
3. Code styles for the `nav ul` selector. Configure the flex container with rows that do not wrap. Also set `justify-content` to `space-around`.
4. Code styles for the nav li selector. Set the bottom border to `none`.
5. Code styles for the hero images. Configure the `homehero` id selector with 50vh height and the hero.jpg background image. Configure `heromugs` id selector with the heromugs.jpg background image. Configure the `heroguitar` id selector with the heroguitar.jpg background image.
6. Code styles for the `flow` id selector. Configure a flex container. The flex direction is `row`.
7. Code styles for the phone number. Configure the `mobile` id selector with display set to `none`. Configure the `desktop` id selector with display set to `inline`.
8. Code styles for the `details` class selector. Configure a flex container. The flex direction is row.
9. Code styles for the h4 element selector. Set left and right margin to 10%.

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

EDIT THE HTML. You need to rework the content area on the Menu and Music pages

1. Launch a text editor and open menu.html. Locate the section elements and remove the `class="onethird"` code from each. Code a div assigned to an id named `flow` that contains all section elements. Save the file.
2. Launch a text editor and open music.html. Locate the img tags and remove the `class="floatleft"` code. Each img is followed by some descriptive text. Enclose each group of descriptive text within a paragraph element. Save the file.

TEST THE WEB PAGES. Display your menu.html file in a browser. You should be able to resize your browser viewport and obtain a display similar to the Medium Display in Figure 7.56. Test the index.html and music.html files in a similar manner.

TASK 4: CONFIGURE A LARGE LAYOUT. Edit the CSS to configure a second media query with a 1024px breakpoint that will configure a grid layout with two columns. When you test your web pages and trigger the media query, the layout in the Large Display wireframe in Figure 7.54 will be implemented and your pages should look similar to the Large Display in Figures 7.55, 7.56, and 7.57.

CONFIGURE THE CSS. Launch a text editor and open the javajam.css style sheet. Place your cursor below the existing styles. Configure a media query that is triggered when the minimum width is 1024px or greater. Within the media query, configure a feature query to check for support of grid layout. Code the following styles within the feature query.

1. Configure the grid areas.
   a. Code styles for the header element selector: set `grid-area` to header.
   b. Code styles for the nav element selector: set `grid-area` to nav.
   c. Code styles for the main element selector: set `grid-area` to main.
   d. Code styles for the footer element selector: set `grid-area` to footer.
2. Configure the `wrapper` id selector as a grid container. Use the `grid-template` property to describe the grid layout shown for large display in Figure 7.54. Use 200px for the width of the navigation area. The CSS follows:

```
#wrapper { display: grid;
           grid-template:
                "header header"
                "nav   main"
                "footer   footer"
                / 200px; }
```

3. Configure the navigation area. Code styles for the `nav ul` selector to set the value `column` for `flex-direction`.
4. Configure the header area. Code styles for the header selector to set coffeelogo.jpg as the background image.

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

TEST THE WEB PAGES. Display your index.html file in a modern browser. You should be able to resize your browser viewport and obtain a display similar to the Large Display in Figure 7.55