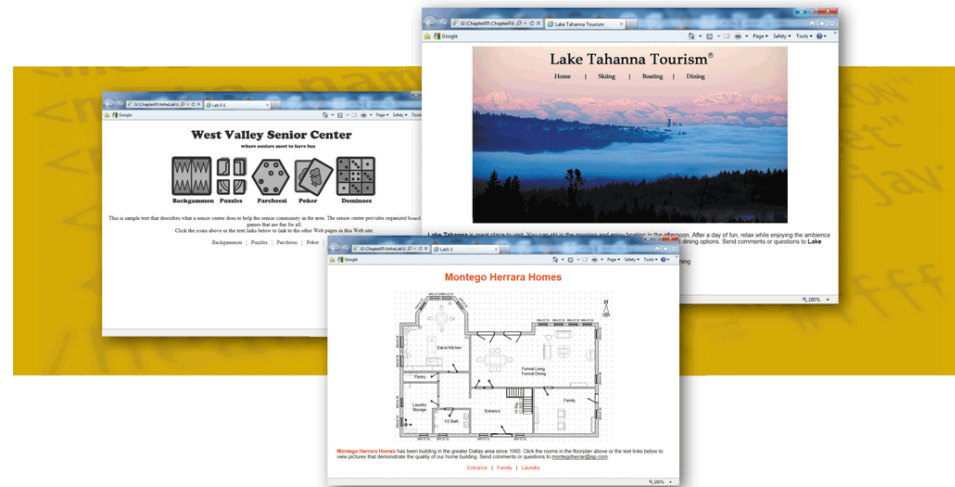


Web Design with HTML5 & CSS3 8th Edition

Chapter 5

Responsive Design Part 1: Designing for Mobile Devices



Chapter Objectives

- Explain the principles of responsive design
- Describe the pros and cons of a mobile website
- Explain the design principles of a mobile website
- Describe a mobile-first strategy
- Define a viewport
- Create a fluid layout

Chapter Objectives (continued)

- Make images flexible
- Use styles for a mobile viewport
- Insert and style a span element
- Insert a viewport meta tag
- Test a responsive site using a device emulator

Exploring Responsive Design

- **Responsive design**
 - **Responsive design** is a website development strategy that strives to provide an optimal user experience of a website regardless of the device or browser used
 - The content in responsive design is easy to read and navigate on devices of three sizes: desktop browser, tablet, and phone

Exploring Responsive Design (continued 1)

- The three concepts of responsive design are:
 1. **Fluid layout:**
 - Applies proportional size measurements to the webpage wireframe and content
 - The **viewport** is the viewing area for the webpage

Exploring Responsive Design (continued 2)

2. Media queries

- Allow the webpage developer to detect the approximate pixel size of the current viewport
- Allow the developer to selectively apply CSS rules that work best for that viewport size

Exploring Responsive Design (continued 3)

3. Flexible images

- They shrink and grow based on the size of the viewport
- They do not have height and width attributes or values in the HTML document
- They use CSS rules to resize the image relative to the wireframe and viewport

Designing for Mobile Devices

- **Mobile websites**

- Techniques emerged to build a completely separate, parallel website optimized for mobile users called a **mobile website** to address problems with viewports
- They are identified with an m. or mo. prefix in the URL
- In contrast, responsive design optimizes the viewing experience for a wide range of devices using one website

Designing for Mobile Devices (continued 1)

- Mobile-first strategy
 - Employs responsive design principles
 - In this, a web developer designs the flexible wireframe and essential content for the smallest viewport first, progressively adding more content as the viewport grows
 - Then, media queries are used to add styles for progressively larger viewports, progressing from tablet to laptop and desktop
 - It is a more productive and effective way to build a website from scratch

Designing for Mobile Devices (continued 2)

- Implementation of the website development approach depends on many factors, such as:
 - the current environment
 - the target audience
 - available resources
 - the time available to tackle the project

Using Fluid Layouts

- **Fluid layout**

- It is also known as a webpage with a liquid layout
- It changes in width based on the size of the viewport
- Responsive designs are based on fluid layouts

Using Fluid Layouts (continued 1)

- **Fixed layouts**

- They do not change in width based on the size of the viewport
- They use fixed measurement units such as pixels to define the width of the areas of the wireframe that “fix” the width of the content regardless of the size of the viewport
- It is implemented by measuring the widths of the wireframe elements and content in relative units such as percentages and ems

Using Fluid Layouts (continued 2)

Table 5–1 Common CSS Units of Measurement

Unit	Description	Relative or Fixed?	Common Uses	CSS Examples
em	An em is historically based on the height of the capital letter M of the default font. 1em is typically larger on a desktop browser than on a tablet browser. For example, 1em is usually about 16pt in a desktop browser and about 12pt in a tablet browser.	Relative	Em may be used to scale anything related to textual content such as font size, line sizes, margins, padding. Em sizes are relative to each other. For example, 2em = twice as large as 1em. 0.5em = half as large as 1em.	<pre>p {font-size: 1.0em; line-height: 2.0em; text-indent: 1.8em;} h1 {font-size: 3.0em; margin: 1.0em; padding: 1.5em;}</pre>
%	Percentage. The default font size measurement for most browsers on most devices is 100%.	Relative	Developers use % to measure the widths of the wireframe elements and flexible images. Some use % to measure textual content, too.	<pre>#container {width: 80%;} img {width: 100%;}</pre>
px	One pixel is equal to one dot on the screen. Different screens have different pixel densities.	Pixels on a device are fixed in size, but the number of pixels varies by device	Pixels are commonly used for textual measurements including padding, borders, and margins. Do not use the px measurement for width measurements, as this creates a fixed, unresponsive layout.	<pre>.advertise {border: 1px solid red;}</pre>
pt	points (1pt = 1/72 inch)	Fixed	Points are used to measure font and line sizes in print media.	
cm	centimeters	Fixed	These measurements are not commonly used for webpage development.	Because these measurements are fixed and do not scale based on the size of the viewport, they should not be used within a responsive design.
mm	millimeters			
in	inches			
pc	picas (1pc = 12pt)	Fixed	The pica measurement harkens back to the “pica typewriter,” which produced a Courier fixed-width font, 12pts tall.	

Creating a Fluid Layout

- It is created by using percentages to measure the width of each column in a webpage
- A webpage can have a liquid layout without using media queries
- No matter how small a browser becomes, the content placeholder boxes on the page shrink and grow in response to the size of the viewport

Creating a Fluid Layout (continued 1)

- To make the navigation links more appealing on a webpage, they can be formatted so they look like buttons rather than text links

Creating a Fluid Layout (continued 2)

- Figure 5–8 shows a webpage with a liquid layout in two widths

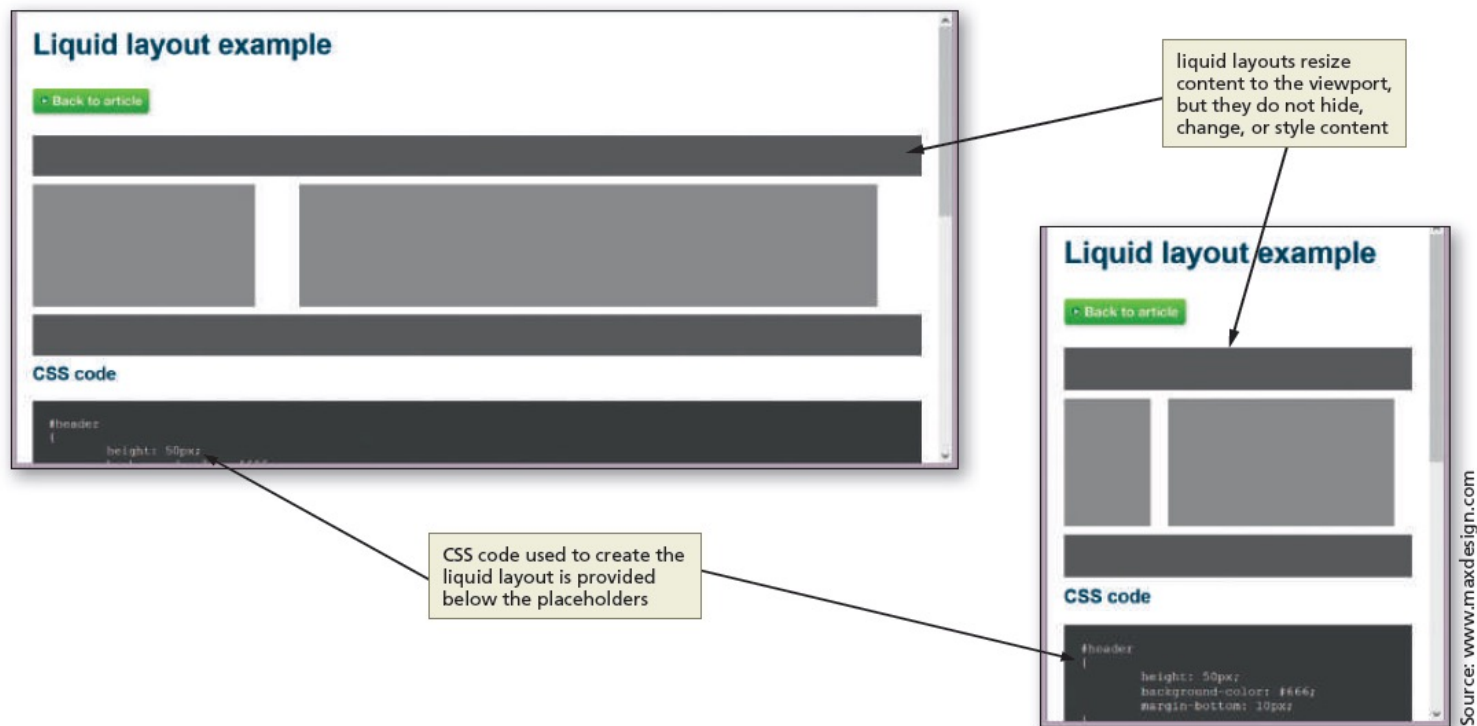


Figure 5–8

Making Images Flexible

- **Flexible image** – It resizes itself to accommodate the size of the viewport and is easy to implement
- **Creating flexible images:**
 1. Delete the `height` and `width` attribute values for the `img` tags in the HTML files
 2. Add styles for the images in the CSS file to provide the desired flexibility such as the following style: `max-width: 100%;`

Making Images Flexible (continued)

- By setting the width of the image to 100%, the image automatically stretches to fill 100% of the width of the container element

Following a Mobile-First Strategy

- It is better to use a single-column layout for a mobile display as this prevents scrolling horizontally
- Styling content for mobile devices requires that each page be analyzed to determine the most important content on the page, and then style that content to attract users of mobile devices

Following a Mobile-First Strategy (continued 1)

- Figure 5–22a depicts a wireframe example for a traditional desktop viewport, while Figure 5–22b depicts all these same areas of content for a mobile design

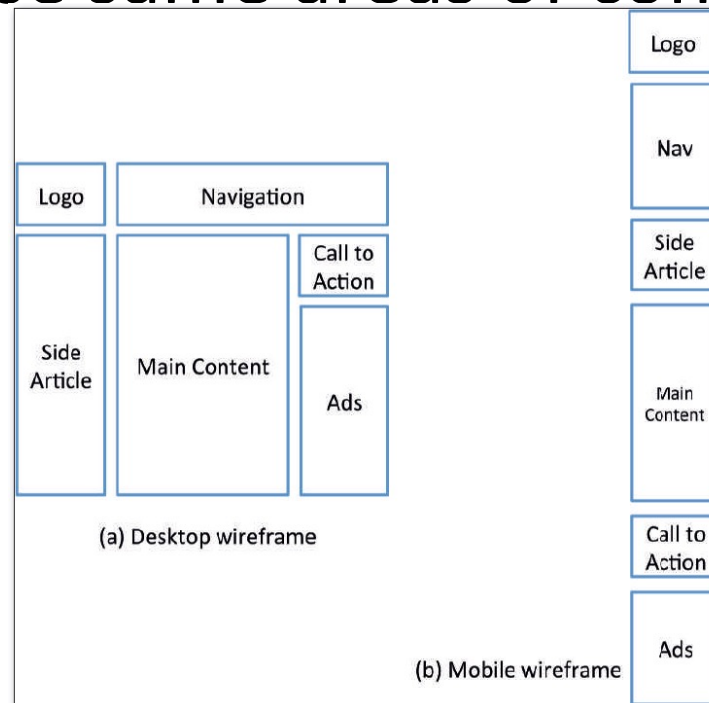


Figure 5–22

Following a Mobile-First Strategy (continued 2)

- Figure 5–23 shows how to hide some webpage areas to create a revised mobile wireframe

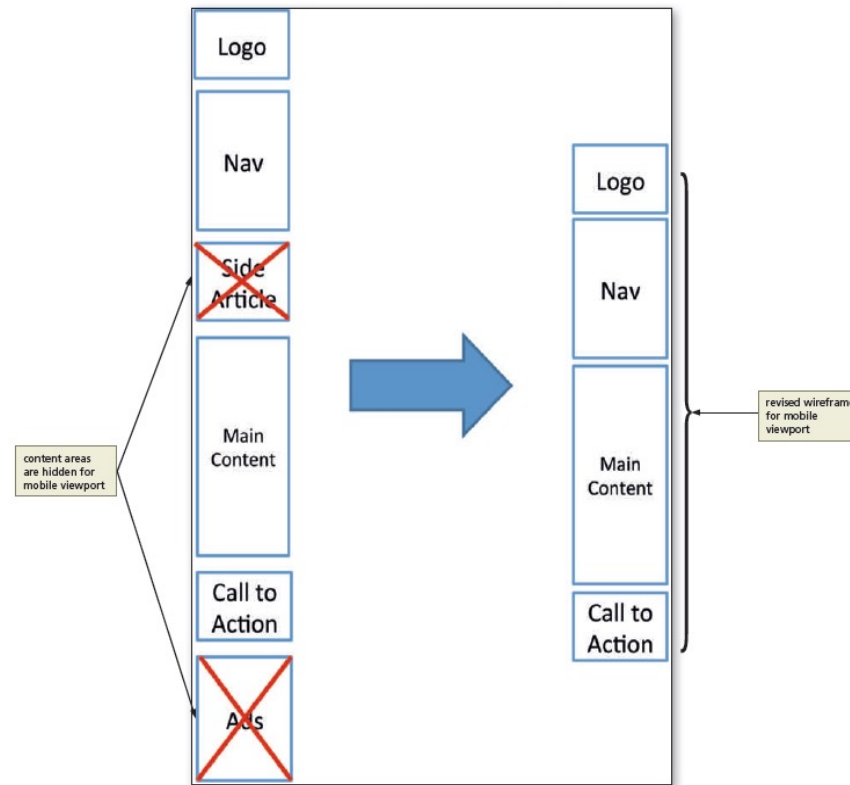


Figure 5–23

Following a Mobile-First Strategy (continued 3)

- Optimize the interface to maximize the mobile user experience
- Some of the key best practices when designing for mobile viewports are as follows:
 1. Make use of 100% of the screen space
 2. Design the navigation to be easy and intuitive
 3. Keep load times minimal. Enhance load times by removing bandwidth-intensive content and streamlining your HTML code

Following a Mobile-First Strategy (continued 4)

4. Display essential page content and hide nonessential page content
5. Make the content easy to access and read
6. Design a simple layout

Using the Span Element

- `span` element
 - Allows the use of CSS to format a span of text separately from its surrounding text
 - The `` and `` are the start and the end tags
 - As an inline element, it can be applied to text within a block
 - A class can be added to a `span` element to style its text

Using the Span Element (continued)

- For example, the following code shows a phone number wrapped in `span` element.

```
<span class="desktop"> (814) 555-9608</span>
```

Adding Meta Tags

- For responsive sites, add the following `meta` tag to the `head` section of each webpage.

```
<meta name="viewport"
content="width=device-width, initial-
scale=1">
```

- This code makes sure that the page *initially loads* in a layout width that matches the viewport of the device

Testing Webpages in Viewports of Different Sizes

Table 5-2 Mobile Device Emulators

Emulator Name	Website	Description
Mozilla Firefox Developer	https://www.mozilla.org/en-US/firefox/developer/	Set of authoring and debugging tools built into Firefox
Chrome Developer's Tools	https://developer.chrome.com/devtools	Set of authoring and debugging tools built into Chrome
Viewport Emulator	http://www.viewportemulator.com/	Online tool to test responsive design website in multiple viewports
iPadPeek	http://ipadpeek.com	Online iPad and iPhone simulator
TestiPhone.com	http://www.testiphone.com/	Online iPhone simulator
iOS Simulator	http://developer.apple.com/	Official Apple iOS simulator — downloadable software
Android Emulator	https://developer.android.com/	Official Android simulator — downloadable software
Windows Phone Emulator	http://msdn.microsoft.com/ (then search for Windows Phone Emulator)	Windows Phone Emulator — downloadable software
Opera Mini Emulator	https://dev.opera.com (then search for Opera Mini Emulator)	Opera Mini Emulator — downloadable software

HTML

8th Edition

Chapter 5 Complete

