

# Chapter 5

---

RESPONSIVE DESIGN PART 1: DESIGNING FOR MOBILE DEVICES

# Chapter Objectives

---

01

Explain the principles of responsive design

02

Describe the pros and cons of a mobile website

03

Explain the design principles of a mobile website

04

Describe a mobile-first strategy

05

Define a viewport

06

Create a fluid layout

# Chapter Objectives (continued)

---

01

Make images flexible

02

Use styles for a mobile viewport

03

Insert and style a span element

04

Insert a viewport meta tag

05

Test a responsive site using a device emulator

# Responsive Design – The Viewport

---

## Viewport:

- User's visible area of a web page
- Varies with the device
  - Smaller on a mobile phone than a computer screen

## Before tablets and mobile phones:

- Web pages were designed only for computer screens
- It was common for web pages to have a static design and a fixed size

## After tablets and mobile phones:

- Fixed size web pages were too large to fit the viewport
- Browsers scaled down the entire web page to fit the viewport

# Responsive Web Design



Web design that allows webpages to be viewed in response to the size of the device being viewed



Makes web pages look good on all devices



Uses only HTML and CSS

## Chapter 5 - Create fluid layouts

- Add style rules for a mobile viewport
  - All viewports use these style rules by default

## Chapter 6

- Add style rules that target tablet or desktop viewports
  - Create a media query for each viewport
  - Web pages can be viewed as a range of screen sizes

# Responsive Web Design

---

CSS and HTML are used to resize, hide, shrink, enlarge, or move the content to make it look good on any screen:



# Setting The Viewport - Meta Tag

---

Add the following <meta> viewport element in the head section:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- Makes sure that the page initially loads in a layout width that matches the viewport of the device

A <meta> viewport element tells the browser how to control the page's dimensions and scaling

- The **width=device-width** sets the width of the page to follow the screen-width of the device (which will vary depending on the device)
- The **initial-scale=1.0** sets the initial zoom level when the page is first loaded by the browser



# Concepts of Responsive Design



Fluid layout



Media queries



Flexible images

# Concepts of Responsive Design: Fluid Layout

## Fluid (Liquid) layout:

- Changes in width based on the size of the viewport
- Responsive designs are based on fluid layouts

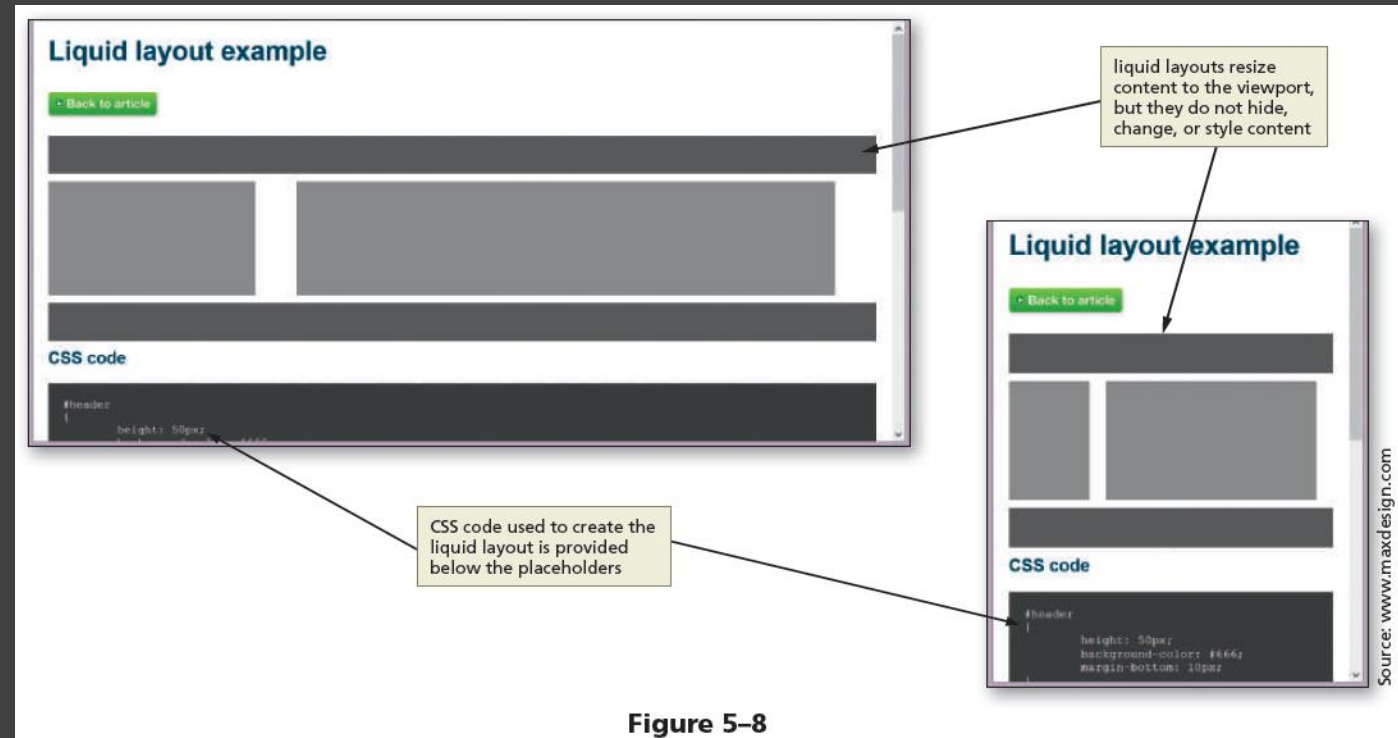


Figure 5-8

## Creating a Fluid Layout

Created by using percentages to measure the width of each column in a webpage

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

# Concepts of Responsive Design: Media Queries

---

## Media queries

- Detects the approximate size of the current viewport
- Selectively apply CSS rules that work best for that viewport size

# Flexible Images

---

## Flexible image

- Resizes itself to accommodate the size of the viewport

## Creating flexible images:

- Do not use height and width attributes in the <img> tags in the HTML document
- Use CSS rules to resize the image relative to the viewport
- By setting the width of the image to 100%, the image automatically stretches to fill 100% of the width of the container element

```
/* Style to create a fluid image */  
img {  
    max-width: 100%;  
}
```



# Mobile-First Strategy

Employs responsive design principles

- Design the content for the smallest viewport first, progressively adding more content as the viewport grows
- Media queries are used to add styles for progressively larger viewports, progressing from tablet to laptop and desktop

It is better to use a single-column layout for a mobile display as this prevents scrolling horizontally

# Span Element

---

- Inline element that can be applied to text within a block
- Allows CSS to format a span of text separately from its surrounding text
- A class can be added to a `span` element to style its text

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

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

# The End

---

Short slides but complicated demo.

Don't forget... GET STARTED ON TERM PROJECT! Week 5 is [index.html](#)

Chapter 5 quiz and homework due next week as usual