CENGAGE

ILLUSTRATED SERIES®

HTML5 and CSS3 Complete

Second Edition

FEATURES:

- Create a website with forms, video, JavaScript, cutting-edge CSS, and more
- New! Covers responsive design
- New! Integrates mobile design and testing

Sasha Vodnik

New! Learning Outcomes

Every 2-page lesson in this book now contains a green **Learning Outcomes box** that states the learning goals for that lesson.

Learning Outcomes
- Locate a custom font online
- Add a `link` element for a custom font
- Add a custom font to a font stack

- **What is a learning outcome?** A learning outcome states what a student is expected to know or be able to do after completing a lesson. Each learning outcome is skills-based or knowledge-based and is *measurable*. Learning outcomes map to learning activities and assessments.
- **How do students benefit from learning outcomes?** Learning outcomes tell students *exactly* what skills and knowledge they are *accountable* for learning in that lesson. This helps students study more efficiently and effectively and makes them more active learners.
- **How do instructors benefit from learning outcomes?** Learning outcomes provide clear, measurable, skills-based learning goals that map to various high-quality learning activities and assessments. A **Learning Outcomes Map**, available for each unit in this book, maps every learning outcome to the learning activities and assessments.

HTML5 and CSS3 Complete

ILLUSTRATED

Second Edition

Sasha Vodnik

CENGAGE Learning®

Australia • Brazil • Mexico • Singapore • United Kingdom • United States

CENGAGE Learning

HTML5 and CSS3—Illustrated Complete
Second Edition
Sasha Vodnik

Product Director: Kathleen McMahon

Product Team Manager: Kristin McNary

Senior Product Manager: Jim Gish

Senior Director of Development: Marah Bellegarde

Product Development Manager: Leigh Hefferon

Senior Content Developer: Marjorie Hunt

Product Assistant: Abigail Pufpaff

Marketing Director: Michele McTighe

Senior Marketing Manager: Eric La Scola

Manufacturing Planner: Fola Orekoya

Developmental Editor: Pamela Conrad

Full Service Project Management: GEX Publishing Services

Proofreader: Gail Marks

Indexer: Alexandra Nickerson

QA Manuscript Reviewers: Danielle Shaw, Jeff Schwartz, and John Freitas

Cover Designer: GEX Publishing Services

Cover Artist: GEX Publishing Services

Cover image: © VikaSuh/Shutterstock.com

Composition: GEX Publishing Services

© 2016 Cengage Learning

Google product screenshots © 2015 Google and Microsoft product screenshots © 2015 Microsoft Corporation

WCN: 01-100-101

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced, transmitted, stored or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the publisher.

For product information and technology assistance, contact us at
Cengage Learning Customer & Sales Support, 1-800-354-9706
For permission to use material from this text or product, submit all requests online at **www.cengage.com/permissions**
Further permissions questions can be emailed to
permissionrequest@cengage.com

Library of Congress Control Number: 2015945493
ISBN: 978-1-305-39404-9

Cengage Learning
20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with employees residing in nearly 40 different countries and sales in more than 125 countries around the world. Find your localrepresentative at **www.cengage.com**

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

For your course and learning solutions, visit **www.cengage.com**

Purchase any of our products at your local college store or at our preferred online store **www.cengagebrain.com**

The websites featured in this book are fictional. Names, characters, businesses, places, are either the products of the author's imagination or used in a fictitious manner. Any resemblance to actual websites, places, or people is purely coincidental.

Trademarks:
Some of the product names and company names used in this book have been used for identification purposes only and may be trademarks or registered trademarks of their respective manufacturers and sellers.

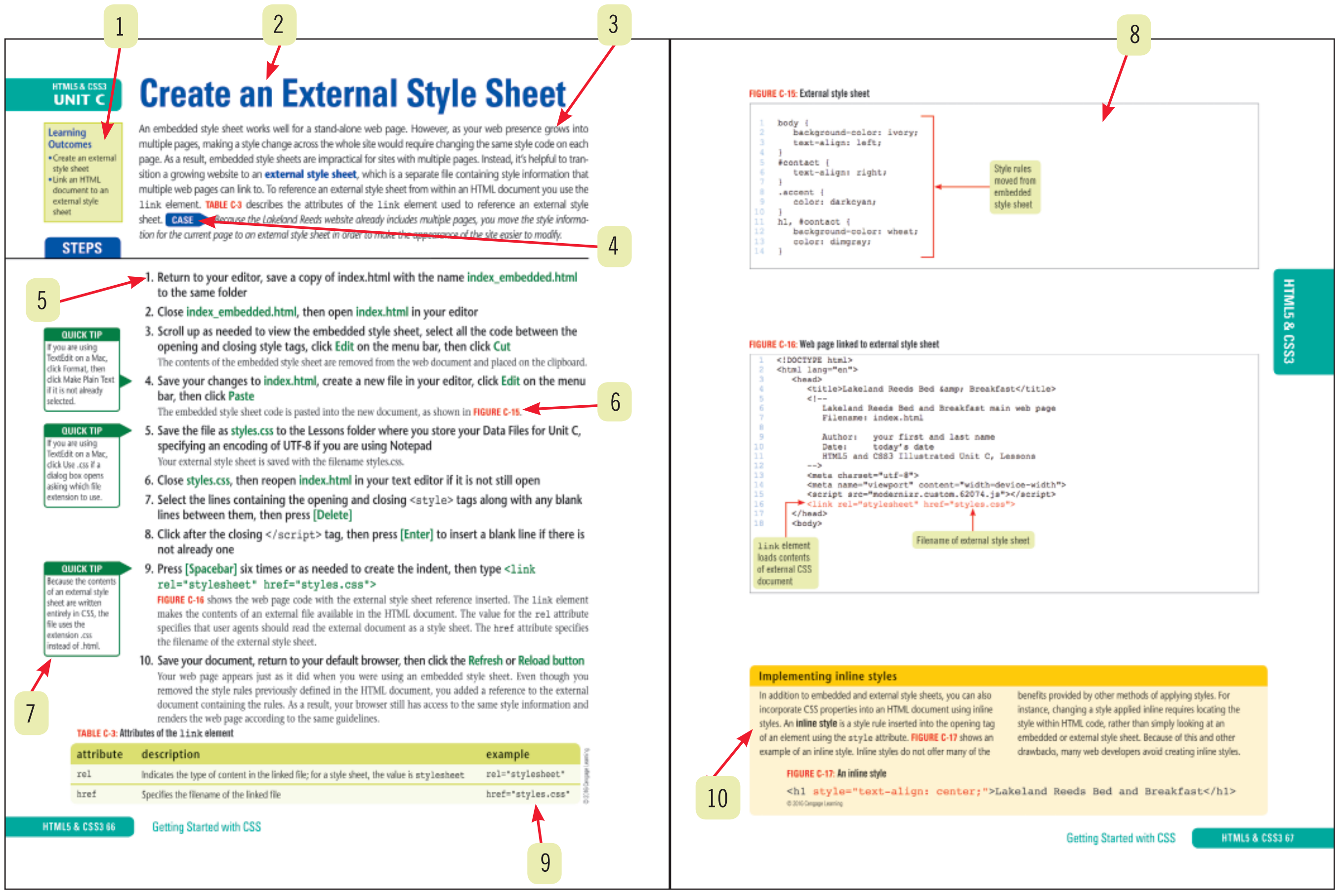
Printed in the United States of America
Print Number: 01 Print Year: 2015

Brief Contents

Contents

Preface

Welcome to *HTML5 and CSS3—Illustrated Complete Second Edition*. This book has a unique design: Each skill is presented on two facing pages, with steps on the left and screens on the right. The layout makes it easy to learn a skill without having to read a lot of text and flip pages to see an illustration.



HTML5 & CSS3
UNIT C

Create an External Style Sheet

Learning Outcomes
- Create an external style sheet
- Link an HTML document to an external style sheet

An embedded style sheet works well for a stand-alone web page. However, as your web presence grows into multiple pages, making a style change across the whole site would require changing the same style code on each page. As a result, embedded style sheets are impractical for sites with multiple pages. Instead, it's helpful to transition a growing website to an **external style sheet**, which is a separate file containing style information that multiple web pages can link to. To reference an external style sheet from within an HTML document you use the `link` element. TABLE C-3 describes the attributes of the `link` element used to reference an external style sheet. CASE *Because the Lakeland Reeds website already includes multiple pages, you move the style information for the current page to an external style sheet in order to make the appearance of the site easier to modify.*

STEPS

1. Return to your editor, save a copy of index.html with the name index_embedded.html to the same folder
2. Close index_embedded.html, then open index.html in your editor
3. Scroll up as needed to view the embedded style sheet, select all the code between the opening and closing style tags, click Edit on the menu bar, then click Cut
   The contents of the embedded style sheet are removed from the web document and placed on the clipboard.
4. Save your changes to index.html, create a new file in your editor, click Edit on the menu bar, then click Paste
   The embedded style sheet code is pasted into the new document, as shown in FIGURE C-15.
5. Save the file as styles.css to the Lessons folder where you store your Data Files for Unit C, specifying an encoding of UTF-8 if you are using Notepad
   Your external style sheet is saved with the filename styles.css.
6. Close styles.css, then reopen index.html in your text editor if it is not still open
7. Select the lines containing the opening and closing `<style>` tags along with any blank lines between them, then press [Delete]
8. Click after the closing `</script>` tag, then press [Enter] to insert a blank line if there is not already one
9. Press [Spacebar] six times or as needed to create the indent, then type `<link rel="stylesheet" href="styles.css">`
   FIGURE C-16 shows the web page code with the external style sheet reference inserted. The `link` element makes the contents of an external file available in the HTML document. The value for the `rel` attribute specifies that user agents should read the external document as a style sheet. The `href` attribute specifies the filename of the external style sheet.
10. Save your document, return to your default browser, then click the Refresh or Reload button
    Your web page appears just as it did when you were using an embedded style sheet. Even though you removed the style rules previously defined in the HTML document, you added a reference to the external document containing the rules. As a result, your browser still has access to the same style information and renders the web page according to the same guidelines.

QUICK TIP
If you are using TextEdit on a Mac, click Format, then click Make Plain Text if it is not already selected.

QUICK TIP
If you are using TextEdit on a Mac, click Use .css if a dialog box opens asking which file extension to use.

QUICK TIP
Because the contents of an external style sheet are written entirely in CSS, the file uses the extension .css instead of .html.

TABLE C-3: Attributes of the `link` element

attribute	description	example
`rel`	Indicates the type of content in the linked file; for a style sheet, the value is `stylesheet`	`rel="stylesheet"`
`href`	Specifies the filename of the linked file	`href="styles.css"`

© 2016 Cengage Learning

HTML5 & CSS3 66 Getting Started with CSS

FIGURE C-15: External style sheet

```
body {
   background-color: ivory;
   text-align: left;
}
#contact {
   text-align: right;
}
.accent {
   color: darkcyan;
}
h1, #contact {
   background-color: wheat;
   color: dimgrey;
}
```

Style rules moved from embedded style sheet

FIGURE C-16: Web page linked to external style sheet

```
<!DOCTYPE html>
<html lang="en">
   <head>
      <title>Lakeland Reeds Bed &amp; Breakfast</title>
      <!--
         Lakeland Reeds Bed and Breakfast main web page
         Filename: index.html

         Author:   your first and last name
         Date:     today's date
         HTML5 and CSS3 Illustrated Unit C, Lessons
      -->
      <meta charset="utf-8">
      <meta name="viewport" content="width=device-width">
      <script src="modernizr.custom.62074.js"></script>
      <link rel="stylesheet" href="styles.css">
   </head>
   <body>
```

`link` element loads contents of external CSS document

Filename of external style sheet

Implementing inline styles

In addition to embedded and external style sheets, you can also incorporate CSS properties into an HTML document using inline styles. An **inline style** is a style rule inserted into the opening tag of an element using the `style` attribute. FIGURE C-17 shows an example of an inline style. Inline styles do not offer many of the benefits provided by other methods of applying styles. For instance, changing a style applied inline requires locating the style within HTML code, rather than simply looking at an embedded or external style sheet. Because of this and other drawbacks, many web developers avoid creating inline styles.

FIGURE C-17: An inline style

```
<h1 style="text-align: center;">Lakeland Reeds Bed and Breakfast</h1>
```

© 2016 Cengage Learning

Getting Started with CSS HTML5 & CSS3 67

HTML5 & CSS3

1 New! Learning Outcomes box lists measurable learning goals for which a student is accountable in that lesson.

2 Each two-page lesson focuses on a single skill.

3 Introduction briefly explains why the lesson skill is important.

4 A case scenario motivates the steps and puts learning in context.

5 Step-by-step instructions and brief explanations guide students through each hands-on lesson activity.

6 New! Figure references are now in red bold to help students refer back and forth between the steps and screenshots.

7 Tips and troubleshooting advice, right where you need it–next to the step itself.

8 New! Larger screen shots with green callouts and code students type shown in red.

9 Tables provide summaries of helpful information such as element and property syntax.

10 Clues to Use yellow boxes provide useful information related to the lesson skill.

This book is an ideal learning tool for a wide range of learners—the "rookies" will find the clean design easy to follow and focused with only essential information presented, and the "hotshots" will appreciate being able to move quickly through the lessons to find the information they need without reading a lot of text. The design also makes this book a great reference after the course is over! See the illustration on the left to learn more about the pedagogical and design elements of a typical lesson.

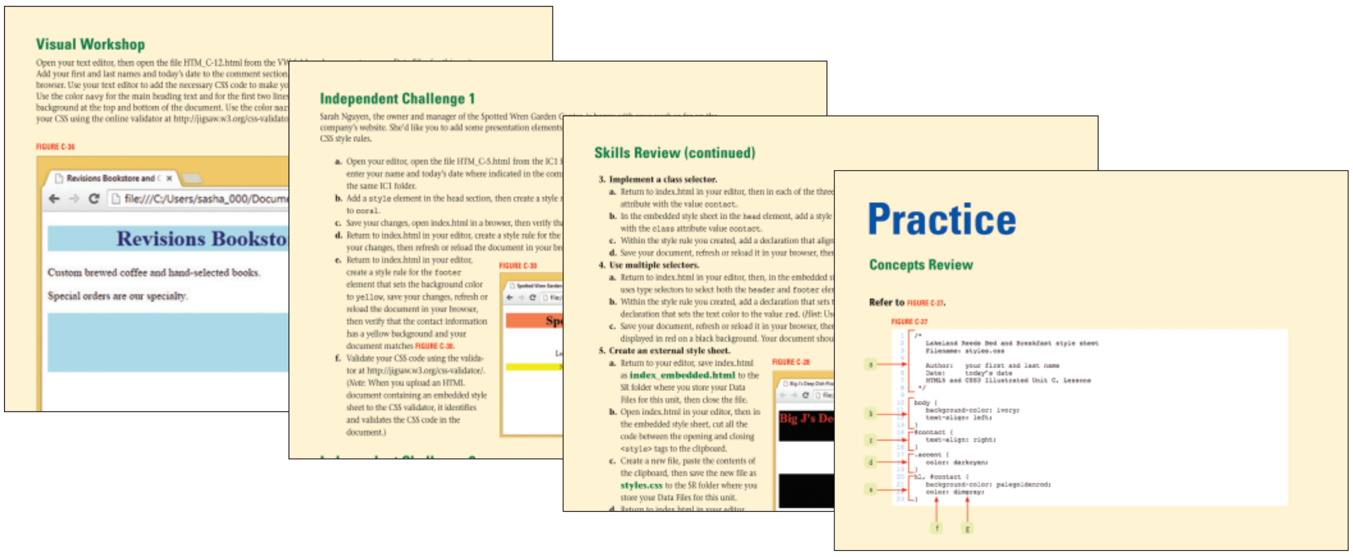
What's New in this Edition

- **Coverage** — This book covers basic to advanced concepts and skills for developing web pages and websites using HTML5 and CSS3. This new edition is fully updated with many enhancements, including new coverage of responsive design, teaching how to build sites that can be accessed by any device; and debugging using browser tools; it also now integrates mobile design and testing throughout.
- **New! Learning Outcomes** — Each lesson displays a green Learning Outcomes box that lists skills-based or knowledge-based learning goals for which students are accountable. Each Learning Outcome maps to a variety of learning activities and assessments.
- **New! Updated Design** — This edition features many new design Improvements to engage students — including larger lesson screenshots with green callouts and code students type formatted in red.
- **New! Independent Challenge 4: Explore** — This new case-based assessment activity allows students to explore new skills and use creativity to solve a problem or create a project.

Assignments

This book includes a wide variety of high-quality assignments you can use for practice and assessment. Assignments include

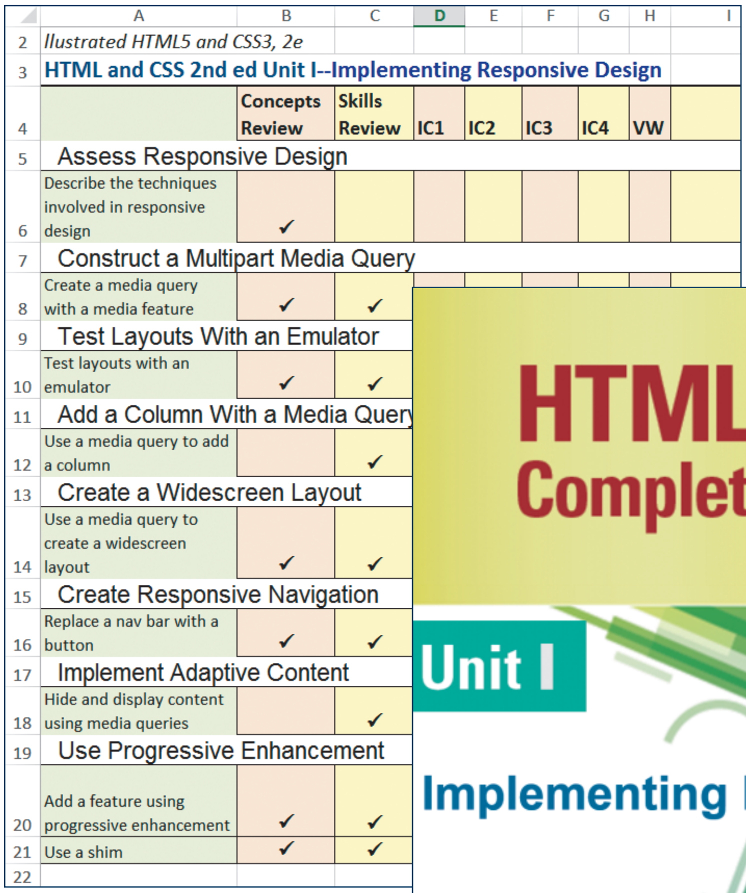
- **Concepts Review** — Multiple choice, matching, and screen identification questions.
- **Skills Review** — Step-by-step, hands-on review of every skill covered in the unit.
- **Independent Challenges 1-3** — Case projects requiring critical thinking and application of the unit skills. The Independent Challenges increase in difficulty. The first one in each unit provides the most hand-holding; the subsequent ones provide less guidance and require more critical thinking and independent problem solving.
- **Independent Challenge 4: Explore** — Case projects that let students explore new skills that are related to the core skills covered in the unit and are often more open ended, allowing students to use creativity to complete the assignment.
- **Visual Workshop** — Critical thinking exercises that require students to create a project by looking at a completed solution; they must apply the skills they've learned in the unit and use critical thinking skills to create the project from scratch.



Instructor Resources

This book comes with a wide array of high-quality technology-based, teaching tools to help you teach and to help students learn. The following teaching tools are available for download at our Instructor Companion Site. Simply search for this text at *login.cengage.com*. An instructor login is required.

- New! **Learning Outcomes Map** — A grid for each unit shows the learning activities and assessments that map to each learning outcome in that unit.
- **Instructor's Manual** — Available as an electronic file, the Instructor's Manual includes lecture notes with teaching tips for each unit.
- **Data Files for Students** — To complete most units, students need Data Files, which are included as an Instructor Resource. Students can also download Data Files on cengagebrain.com.
- **Sample Syllabus** — Prepare and customize your course easily using this sample course outline.
- **PowerPoint Presentations** — Each unit has a corresponding PowerPoint presentation covering the skills and topics in that unit.
- **Figure Files** — The figures in the text are provided on the Instructor Resources site to help you illustrate key topics or concepts.
- **Solution Files** — Solution Files are files that contain the finished project that students create or modify in the lessons or end-of-unit material.
- **Solutions Document** — This document outlines the solutions for the end-of-unit Concepts Review, Skills Review, Independent Challenges and Visual Workshops.
- **Test Banks** — Cengage Learning Testing Powered by Cognero is a full-featured, online assessment system that allows instructors to create tests from publisher-provided content as well as write new questions. With the test generator, you can
  - Create tests from publisher-provided question sets.
  - Edit publisher questions.
  - Write your own questions.
  - Tag questions with learning objectives, rubrics, and other meta-information.
  - Print tests or deliver them online through a learning management system.



Ilustrated HTML5 and CSS3, 2e

HTML and CSS 2nd ed Unit I--Implementing Responsive Design

	Concepts Review	Skills Review	IC1	IC2	IC3	IC4	VW
Assess Responsive Design							
Describe the techniques involved in responsive design	✓						
Construct a Multipart Media Query							
Create a media query with a media feature	✓	✓					
Test Layouts With an Emulator							
Test layouts with an emulator	✓	✓					
Add a Column With a Media Query							
Use a media query to add a column		✓					
Create a Widescreen Layout							
Use a media query to create a widescreen layout	✓	✓					
Create Responsive Navigation							
Replace a nav bar with a button	✓	✓					
Implement Adaptive Content							
Hide and display content using media queries		✓					
Use Progressive Enhancement							
Add a feature using progressive enhancement	✓	✓					
Use a shim	✓	✓					

HTML5 and CSS3 Complete

Second Edition

Unit I

Implementing Responsive Design

Before You Begin

1. Download Data Files

To complete many of the lessons and end-of-unit assignments, you need to use the Data Files that were created specifically for this book. All Data Files are available as part of the Instructor Resources. You can also download Data Files for free at cengagebrain.com. (For detailed instructions, go to www.cengage.com/ct/studentdownload.)

2. Verify System Requirements

To complete the lessons in this book, you need 2 types of applications: an editor and one or more browsers. You can use a plain text editor such as Notepad (part of Windows) or TextEdit (part of Mac OS X), but a code editor is recommended. Free code editors include Aptana Studio 3 (Win/Mac), KomodoEdit (Win/Mac), Notepad++ (Win), and TextWrangler (Mac). You can complete the lessons with a single desktop browser, but real world web development requires testing on a wide variety of browsers and devices. If possible, you should secure access to all of the following desktop browsers: Chrome (Win and Mac), Firefox (Win and Mac), Internet Explorer (Win), and Edge (Win). In addition, if possible, you should make arrangements to test using Safari on iOS (iPhone or iPod Touch) and Chrome on Android.

3. Review These Figure Notes

Figures showing code include line numbers. If your editor shows line numbers, they may not exactly match those shown in the figures; instead, use the line numbers in the figures as a rough guide for orienting yourself in your data files. In figures showing newly entered code, that code is displayed in red for easy identification. If your code editor highlights code in one or more colors, this will not match the color shown in the figures.

Acknowledgements

Thanks to the many people who have helped make this book a reality. Jim Gish was consistently encouraging and made sure the resources were available for a top-notch revision. Marjorie Hunt wrangled the schedule and kept her eye on the big picture so I could focus on the details. Pam Conrad went the extra mile as developmental editor to ensure the explanations were clear, the steps all worked as intended, and all the i's were dotted and the t's crossed. The great people at GEX Publishing Services laid everything out on the page and made it look great. Jeff Schwartz, Danielle Shaw, and John Freitas meticulously tested all the steps and let me know what needed fixing before the book went out the door. John McKelvy/mackjackstudio.com and Kenji Oshima/coachkenji.com created the logos and layout prototypes for the fictitious businesses in the lessons and practice projects. Roland McGrath generously provided recent Android devices for testing.

Finally, thanks to my parents, Jim and Diana Vodnik, for helping me to see learning as both an adventure and a priority. And thanks to my husband, Jason Bucy, for his love, support, and help in navigating the balancing act that's a fundamental part of an author's life.

Feedback and Questions

The author welcomes feedback and questions about this book from instructors and students. You can contact Sasha Vodnik on Twitter at *@sashavodnik*.

CENGAGEbrain.com
Buy. Rent. Access.

Access Student Data Files, CourseMate,
and other study tools on **cengagebrain.com**.

For detailed instructions visit
http://solutions.cengage.com/ctdownloads/.

Store your Data Files on a USB drive for maximum efficiency in
organizing and working with the files.

Macintosh users should use a program to expand WinZip or PKZip archives.
Ask your instructor or lab coordinator for assistance.

HTML5 & CSS3 **Unit A**

Getting Started with HTML

CASE You've been hired by Lakeland Reeds Bed and Breakfast to build a website. You can create your own web pages like those found on the World Wide Web using just your computer and a text editor. In this unit, you'll learn about planning and wireframing a website, creating the head and body sections of a web page, adding web page text, and testing your web page in desktop and mobile browsers.

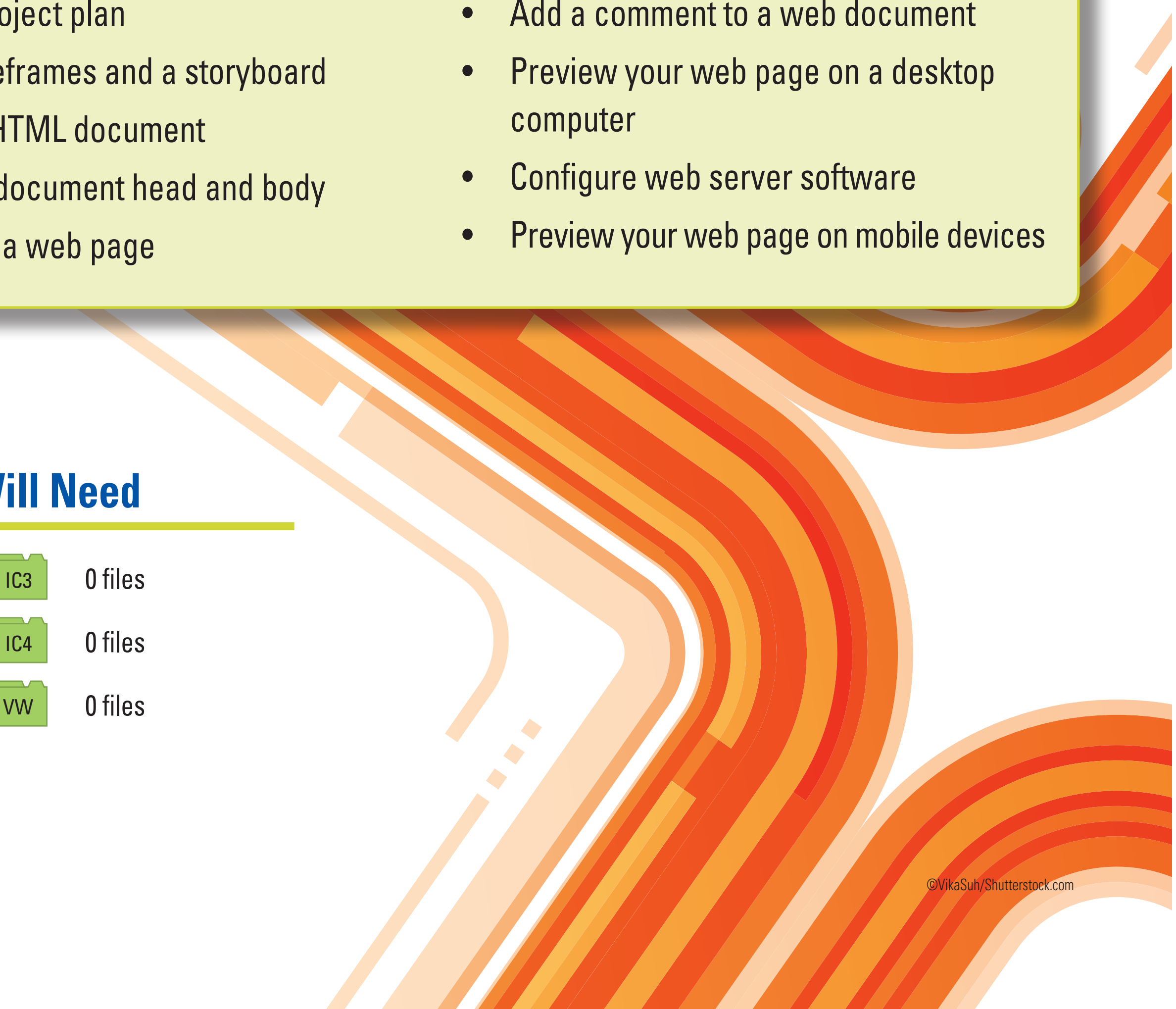
Unit Objectives

After completing this unit, you will be able to:

- Define a project plan
- Create wireframes and a storyboard
- Create an HTML document
- Set up the document head and body
- Add text to a web page
- Add a comment to a web document
- Preview your web page on a desktop computer
- Configure web server software
- Preview your web page on mobile devices

Files You Will Need

L	0 files	IC3	0 files
SR	0 files	IC4	0 files
IC1	0 files	VW	0 files
IC2	0 files		



©VikaSuh/Shutterstock.com

Define a Project Plan

Learning Outcomes
- Identify questions to ask a potential client
- Use a project plan to summarize client responses

People and organizations around the world share information using the **World Wide Web**, or **web** for short. You can make your own information available on the web by creating **web pages**, which are documents formatted to be accessible on the web, and then publishing them as **websites**, which are available to anyone with web access. Whether you intend to make a single web page or a large set of interrelated web pages, making information accessible on the web starts with careful planning. Whether you're brainstorming for a personal site or meeting with a client regarding a site you've been hired to create, your first step is to create a **project plan**, which is also known as a **design document**. A project plan summarizes your client's parameters for the site, including audience, budget, and timeline. **CASE** *You held a planning meeting with Phillip Blaine, owner of Lakeland Reeds, to discuss the components he would like included in his new website.* **FIGURE A-1** *shows the project plan you developed based on client responses at this meeting.*

DETAILS

Important questions to consider when planning a website include the following:

- **What are the goals and objectives of the website?**
  In order to understand the goals and objectives for the site, you want to ask your client a variety of questions. For example, "What is the mission of the organization? Why do you want a website? What are the short-term goals of the website? What are the long-term goals of the website? What do you hope to gain by having a web presence? Who is your target audience? Do your objectives support the needs of your target audience?" The more thorough you are in asking questions of the client, the better prepared you will be to design the website.

- **Who is the target audience?**
  It can be helpful to know the target audience for a website when choosing a layout and design. Websites should look different based upon who will be visiting the site and why they are interested in the content. Some potential questions to ask about the target audience might be, "Who are the typical members of your audience? What is the mix of genders? What is the age range? What professions are they in? What is the average education level? Why will people visit this website? Will your desktop visitors be using Microsoft Windows, Apple OS X, or another operating system? Which web browsers will they use to view the website? How common is mobile device use among your audience members? Which types of mobile devices do they use?" While your client may not have ready answers to all of these questions, getting even a few answers can help prepare you for the design phase.

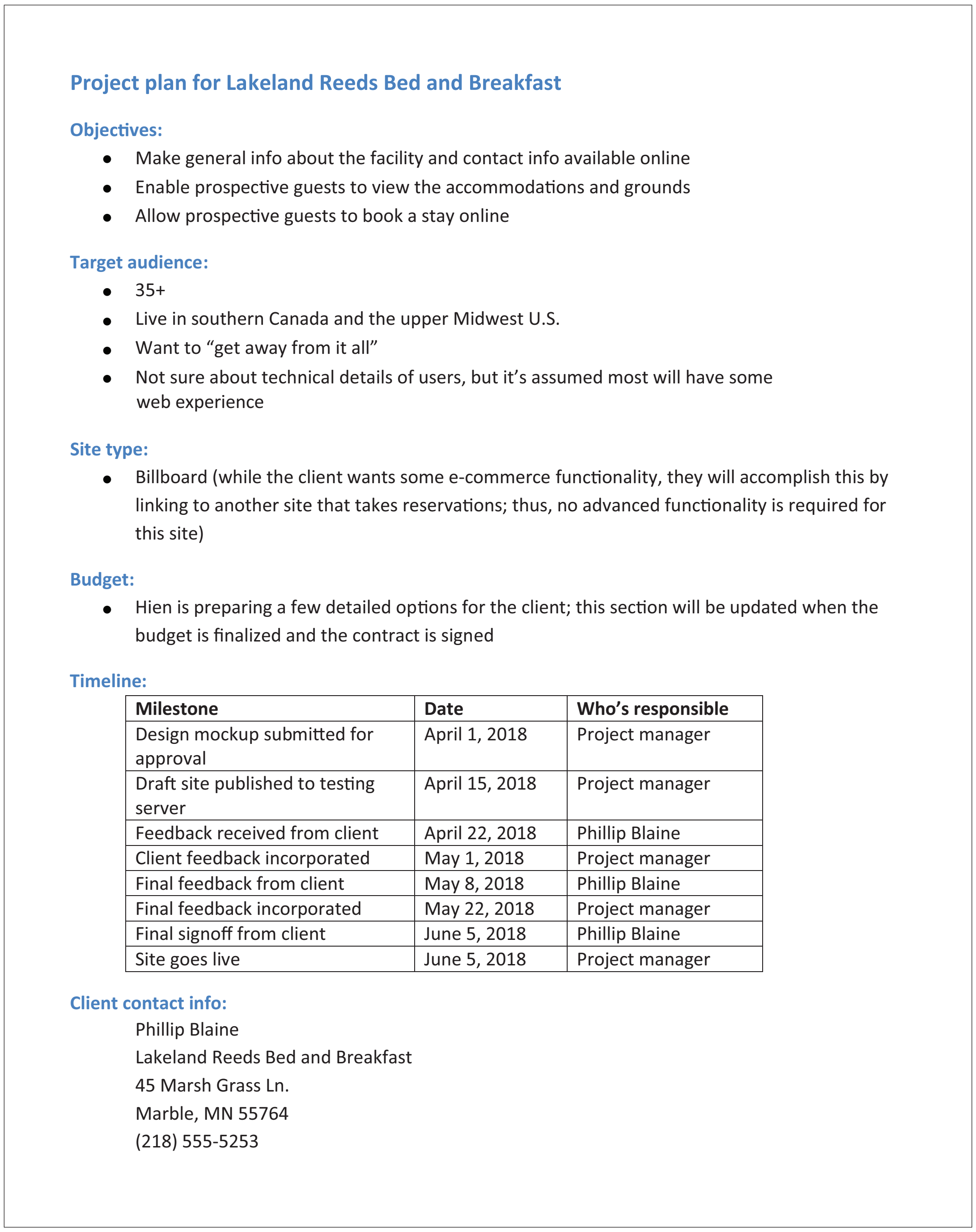
QUICK TIP
Listing exclusions in the project plan will help reduce the potential for scope creep, which is the expansion of a project beyond the original goals and objectives.

- **What type of website is it?**
  Identifying the type of website the owner wants can help to focus the scope of the project. A website usually has one of a small number of main functions: providing a web presence that serves as an online informational brochure, providing important information for special interest groups and nonprofit organizations, showcasing examples of different types of works and designs commonly used by web design individuals and agencies, providing multiple levels of information with page templates, extracting information from databases, or conducting the sale of products or services and other business transactions through the Internet. It is important to clearly define what the site will include, as well as what the site won't include.

- **What is the budget for the project?**
  Every website design project should include a budget that is presented to the client prior to completing any work. The budget should be included in the project plan, which becomes part of the contract.

- **What is the timeline for the project?**
  You should always provide the website owner with a timeline that includes the delivery date of the final website, along with various implementation milestones along the way. The timeline should always identify who is responsible for which tasks.

FIGURE A-1: Lakeland Reeds Bed and Breakfast project plan



Project plan for Lakeland Reeds Bed and Breakfast

Objectives:

- Make general info about the facility and contact info available online
- Enable prospective guests to view the accommodations and grounds
- Allow prospective guests to book a stay online

Target audience:

- 35+
- Live in southern Canada and the upper Midwest U.S.
- Want to "get away from it all"
- Not sure about technical details of users, but it's assumed most will have some web experience

Site type:

- Billboard (while the client wants some e-commerce functionality, they will accomplish this by linking to another site that takes reservations; thus, no advanced functionality is required for this site)

Budget:

- Hien is preparing a few detailed options for the client; this section will be updated when the budget is finalized and the contract is signed

Timeline:

Milestone	Date	Who's responsible
Design mockup submitted for approval	April 1, 2018	Project manager
Draft site published to testing server	April 15, 2018	Project manager
Feedback received from client	April 22, 2018	Phillip Blaine
Client feedback incorporated	May 1, 2018	Project manager
Final feedback from client	May 8, 2018	Phillip Blaine
Final feedback incorporated	May 22, 2018	Project manager
Final signoff from client	June 5, 2018	Phillip Blaine
Site goes live	June 5, 2018	Project manager

Client contact info:

Phillip Blaine
Lakeland Reeds Bed and Breakfast
45 Marsh Grass Ln.
Marble, MN 55764
(218) 555-5253

© 2016 Cengage Learning

Deciding how much to charge

Estimating the amount of time a project will take can be difficult, especially for new web designers. If you work for a web design agency, the budget will typically be developed by your supervisors. If you are a freelance web designer, you must place a value on your time that takes many things into consideration, such as the cost of computer equipment and software, supplying your own insurance, advertising, and other expenses. There really is no set hourly or project fee in this industry, as it varies dramatically depending upon the geographic market, competition, and experience level of the web designer. New web designers often barter, or trade, their skills for products or services offered by a website owner as a means of building a portfolio.

Create Wireframes and a Storyboard

Learning Outcomes
- Define wireframes and storyboards
- Identify wireframe and storyboard components

When you create a web page or a website, it can be helpful to start by getting a clear idea of what you're trying to build. Web designers typically accomplish this by creating a **wireframe**, which is a sketch that outlines the components of each web page and their place in the layout. A designer usually creates one wireframe for each web page or web page type. For a website containing multiple pages, they also create a **storyboard**, which illustrates links among the pages. When there is a web design team working on a project, the people responsible for art or design often create the wireframes and storyboard, and then they hand off these documents to the developers, who use these documents to create a web page or website. **CASE** *You work with Karl Dixon, one of your colleagues in the art department, to create a wireframe and a storyboard for the Lakeland Reeds website based on the project plan.*

DETAILS

When you create a wireframe and a storyboard for a website, be sure to include these main steps:

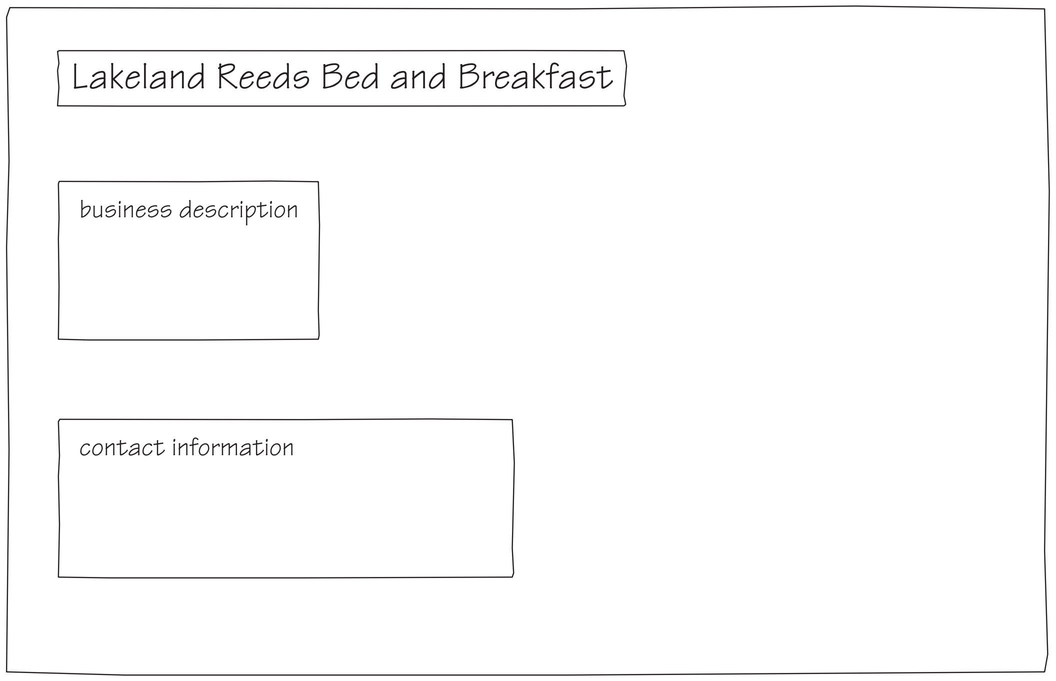
- **Identify components to include**
  Before you start sketching, it's important to get a firm handle on all the elements that the website you're working on must include. A good place to start is your project plan, which should include a thorough inventory of items that must be part of the website; for instance, an existing logo and color scheme that a client already uses in all of its printed materials. You should augment this list with any other essential design elements based on your understanding of the site's target audience and functionality; for example, most multipage websites need a standardized navigation section that provides links to each of the pages.

QUICK TIP
Many common web page designs use columns that are about one third the width of the page to approximate traditional aesthetic proportions.

- **Sketch possible layouts and then select one**
  The next step is to place the elements in a layout that's functional, usable, and, ideally, aesthetically pleasing. This step is often the job of a graphic designer; however, it's a skill that many web developers without artistic backgrounds have obtained with study and practice. Whoever does this step, it often involves a series of sketches that either lays out a set of choices or progressively fine-tunes a theme. For a simple website, a single layout should suffice for all the site's pages; however, if some pages have requirements that are best served by distinct layouts, these layouts need to be finalized in this step as well. The wireframe in **FIGURE A-2** shows the layout for the main page of the Lakeland Reeds website.

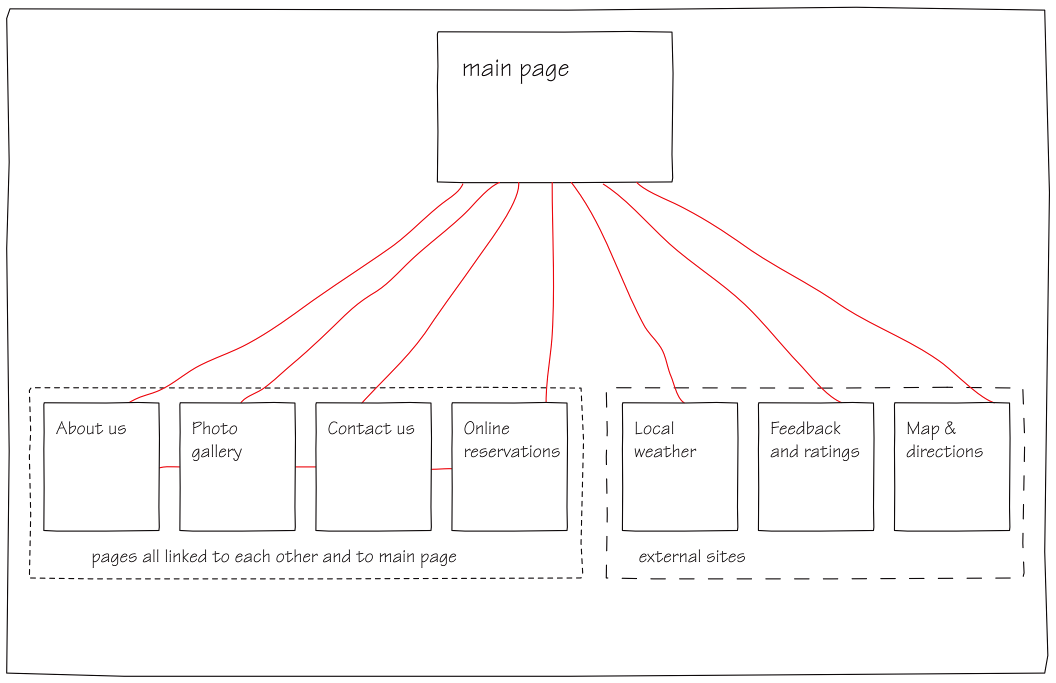
- **Map the relationships among web pages**
  Any time you're creating a website or a single web page with links to other websites, it's helpful to map out the relationships between pages. This map is a crucial tool when you create the navigation system for the website. The storyboard in **FIGURE A-3** lists the pages of the Lakeland Reeds website and illustrates the relationships among them, as well as links to external pages.

FIGURE A-2: Wireframe for Lakeland Reeds main web page



© 2016 Cengage Learning

FIGURE A-3: Storyboard for Lakeland Reeds website



© 2016 Cengage Learning

Creating a website from a template

An alternative to creating a layout for your website is to download a **template**, which is a generic layout that includes a color scheme and element positions but which uses placeholder images and text. Some templates are available to download and use for free, while others must be purchased from the designer. A web developer can simply replace the placeholder items with elements specific to the website being developed. While a template is not as specifically tailored to the companies or topics of the websites where it is used, it can save time in the web development process and can be an invaluable tool when a site needs to be up right away.

Create an HTML Document

Learning Outcomes

- Enter a DOCTYPE declaration
- Add basic tags to an HTML document
- Save an HTML document

Web pages are written in **Hypertext Markup Language (HTML)**, which is a standardized format for specifying the structure of a web page. An HTML document consists solely of text. As a result, you can create a web page in a text editor such as Notepad, which is included with Windows, or TextEdit, which is part of Mac OS X. To create a web page, you enter text that you want to display on the page along with HTML codes known as **tags**, which specify how a browser should treat each item in the document. An HTML tag always starts with an opening angle bracket (<) and ends with a closing angle bracket (>).The text between the angle brackets specifies the HTML element type being applied to the selection. While most tags occur in pairs, some tags, known as **one-sided tags**, are used by themselves. In addition, every document starts with a **DOCTYPE declaration**, which is text that resembles a tag and provides information about the programming language used to write the page. CASE *You create the basic structure of the main page for the Lakeland Reeds website.*

STEPS

QUICK TIP
To write HTML in TextEdit on a Mac, you need to change two program preferences. Press [command][,] to open the Preferences dialog box. On the New Document tab, click Plain text. On the Open and Save tab, uncheck Add ".txt" extension to plain text files.

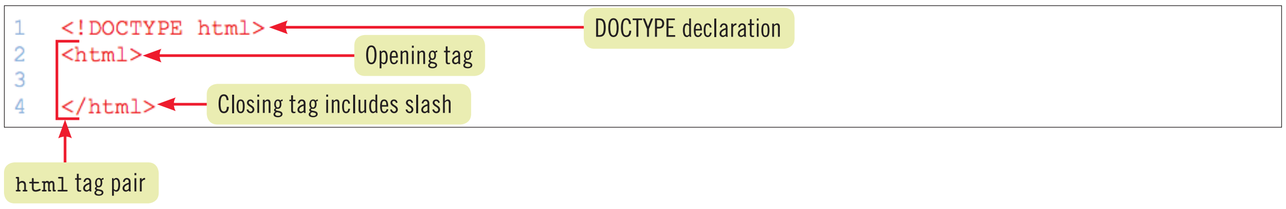
TROUBLE
If you don't have the empty folder structure for storing this unit's files, create a Lessons folder within the drive and folder where you store your Data Files for Unit A, then save your file to the Lessons folder.

1. **Start your text editor**
   A new, blank document opens.
2. **Type `<!DOCTYPE html>`, then press [Enter]**
   This DOCTYPE declaration lets browsers know that the document contents are written in HTML.
3. **Type `<html>`, press [Enter] twice, then type `</html>`, as shown in FIGURE A-4**
   A tag pair assigns meaning to a web page **element**, which is a specific component of the page, such as a paragraph or a heading. You place the **opening tag** at the start of the element you are marking and the **closing tag** at the end. A closing tag is the same as its corresponding opening tag except that the opening angle bracket is followed by a slash (/). The `html` tag pair marks the beginning and the end of the web page.
4. **Click File, then click Save**
   The Save As dialog box opens.
5. **Navigate to the drive and folder where you store your Data Files for Unit A, then open the Lessons folder**
6. **In the File name box (Windows) or Save As box (Mac), type index.html**
   The standard name for the main page of a website is "index." The .htm or .html extension signifies that a file is written in HTML.
7. **If you are using Notepad in Windows, click the Save as type list arrow, then click All Files (*.*)**
8. **If you are using TextEdit on a Mac, click the If no extension is provided, use ".txt" box to uncheck it**
9. **Click Save**
   The index.html file is saved to your storage location.

Using other web page creation software

Many other programs are available that allow you to create web pages visually by clicking buttons and using drag-and-drop to place items on a page. However, creating your first web pages by entering HTML directly—sometimes referred to as **hand-coding**—is one of the best ways to become familiar with HTML and the underlying structure of a web page. Many professional developers use a **code editor**, which is a text editor that is optimized for writing code in a programming language. FIGURE A-5 shows the web page in Notepad++, a free code editor available for Windows, and points out some features common to code editors. Other popular free code editors include Aptana Studio 3 (Win and Mac), Komodo Edit (Win and Mac), and TextWrangler (Mac).

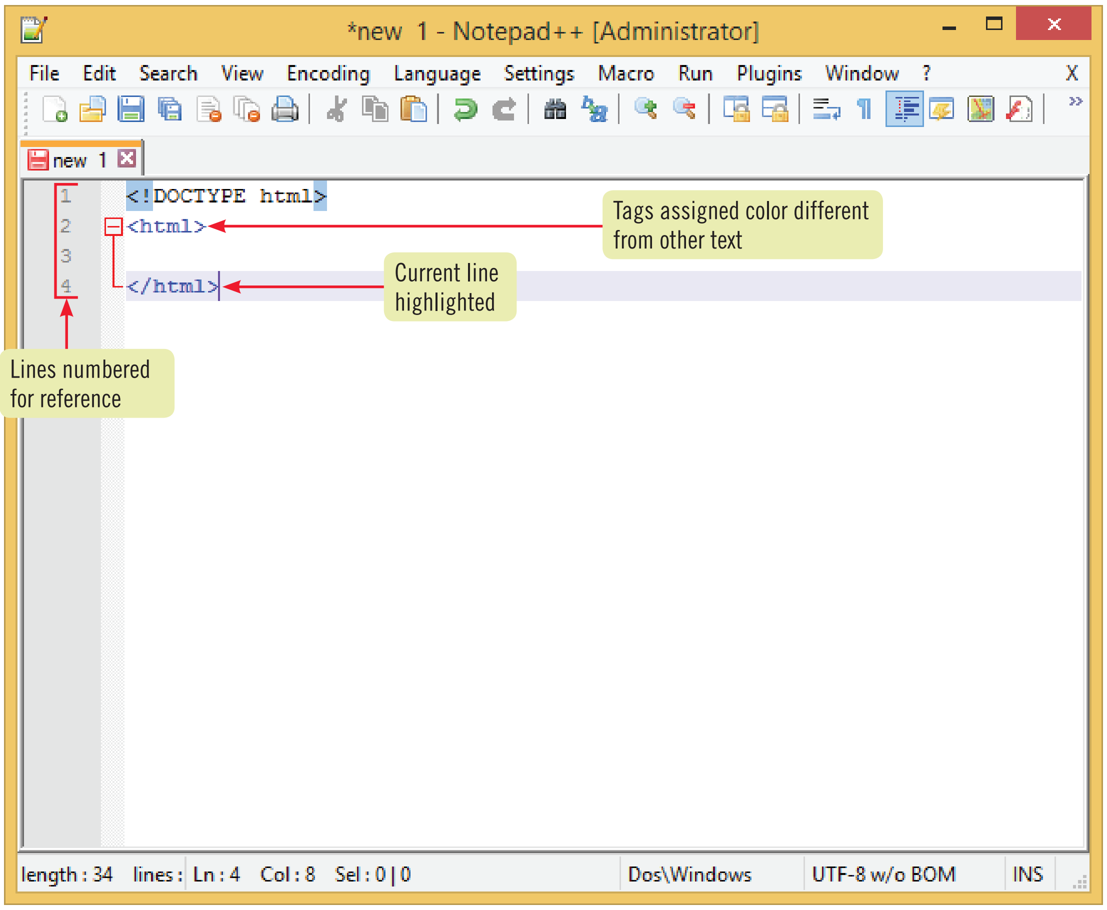
FIGURE A-4: The basic structure of the web page



```
<!DOCTYPE html>
<html>

</html>
```

FIGURE A-5: The basic structure of the web page in Notepad++



```
<!DOCTYPE html>
<html>

</html>
```

```
length : 34    lines : Ln : 4    Col : 8    Sel : 0 | 0          Dos\Windows      UTF-8 w/o BOM      INS
```

*new 1 - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 1
Tags assigned color different from other text
Current line highlighted
Lines numbered for reference

Courtesy: Don Ho 2014

Understanding the difference between HTML and XHTML

Extensible Hypertext Markup Language (XHTML) is a version of HTML that conforms to the rules of Extensible Markup Language (XML). Although most modern web documents are written in HTML, some organizations require XML-compatible code, and therefore use XHTML instead of HTML. HTML and XHTML have only a handful of differences, and utilities are available that can automatically convert code from one to the other. In general, HTML syntax is less strict than XHTML code, and some aspects of code that you can omit in HTML are required in XHTML code. Except where otherwise stated, all the code you write in this book will conform to the rules of HTML.

Set Up the Document Head and Body

Learning Outcomes

- Define element nesting
- Add a head section to an HTML document
- Add a body section to an HTML document

Within the `html` tag pair for a web page, the document contents are divided into two sections. The **head section** contains elements that specify information about the web page but are not displayed in the browser window, such as the page title that appears in a browser tab. The contents of the **body section** are visible in the main window of a web browser and include elements like headings and paragraphs. Both the `head` and `body` tag pairs are located within, or **nested** within, the `html` tag pair. Most elements in the code for a web page are nested within one or more other elements. CASE *As you continue creating the structure for the Lakeland Reeds web page, you add the head and body sections to the page.*

STEPS

QUICK TIP
To read more about any HTML tag used in this book, see Appendix A.

1. **Click the blank line between the opening and closing `html` tags, press [Spacebar] three times, then type `<head>`**
   This is the opening tag for the `head` tag pair. Adding a fixed number of spaces before a nested tag makes it appear indented. As your web page code becomes longer and more complex, these indentations make it easier to identify the beginning and end of an element at a glance. For the code you enter in this book, you'll use three spaces for each indent level. FIGURE A-6 shows HTML code with several layers of nested tags that are formatted with indentations.
2. **Press [Enter] twice, press [Spacebar] three times, then type `</head>`**
   This creates the closing tag for the `head` tag pair.
3. **Press [Enter], press [Spacebar] three times, then type `<body>`**
   This creates the opening tag for the `body` tag pair.
4. **Press [Enter] twice, press [Spacebar] three times, then type `</body>`**
   This creates the closing tag for the `body` tag pair.
5. **Save your work**
   FIGURE A-7 shows the completed web page structure including the head and body sections.

QUICK TIP
To save your work without using the menus, you can press [Ctrl][S] (Windows) or [command][S] (Mac).

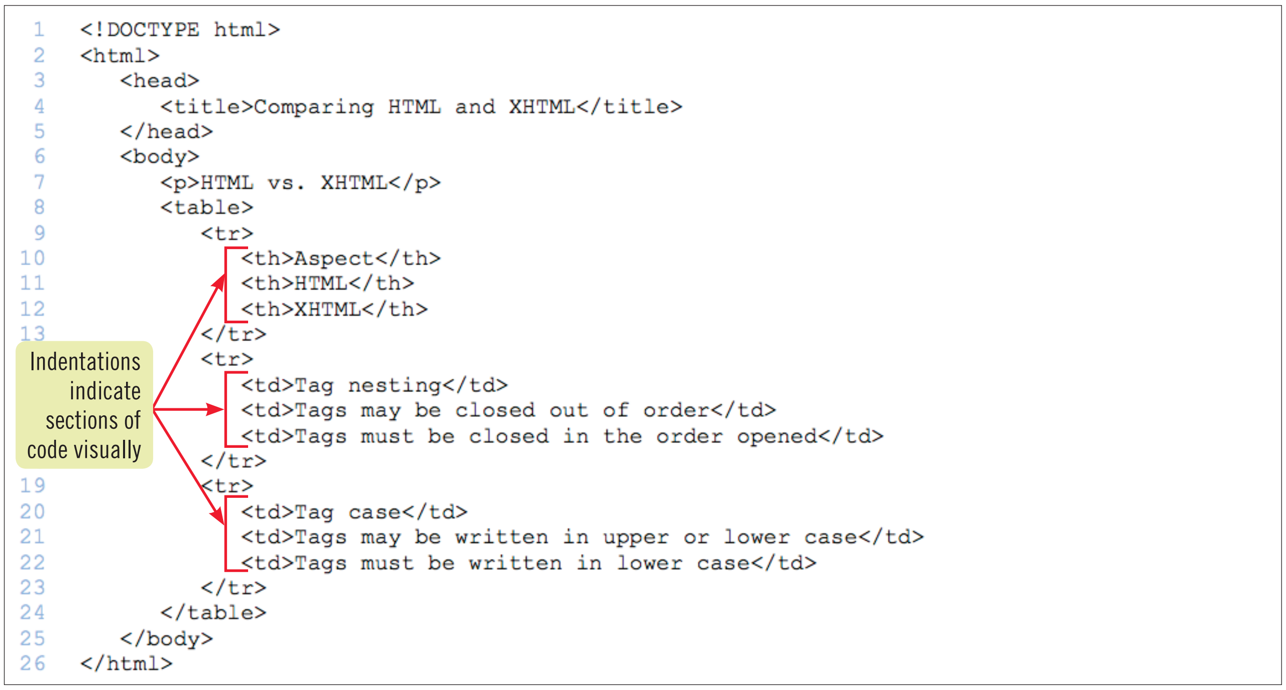


FIGURE A-6: HTML code containing multiple layers of nested elements

```
<!DOCTYPE html>
<html>
   <head>
      <title>Comparing HTML and XHTML</title>
   </head>
   <body>
      <p>HTML vs. XHTML</p>
      <table>
         <tr>
            <th>Aspect</th>
            <th>HTML</th>
            <th>XHTML</th>
         </tr>
         <tr>
            <td>Tag nesting</td>
            <td>Tags may be closed out of order</td>
            <td>Tags must be closed in the order opened</td>
         </tr>
         <tr>
            <td>Tag case</td>
            <td>Tags may be written in upper or lower case</td>
            <td>Tags must be written in lower case</td>
         </tr>
      </table>
   </body>
</html>
```

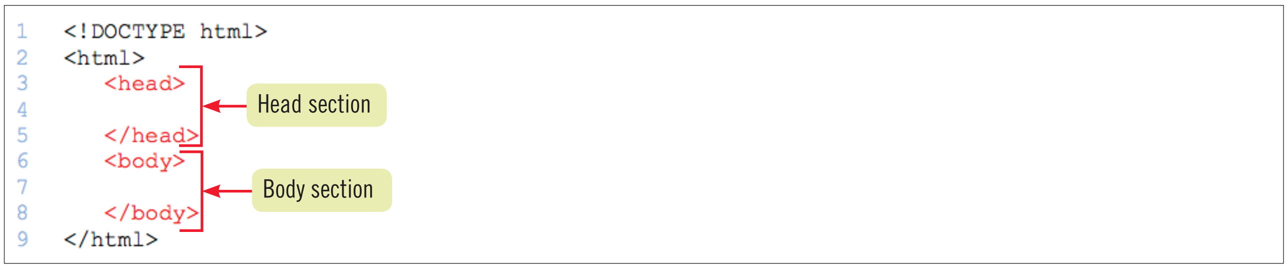


FIGURE A-7: Completed web page structure

```
<!DOCTYPE html>
<html>
   <head>

   </head>
   <body>

   </body>
</html>
```

Describing nesting relationships

An element nested within another element is called a **child element** of the enclosing element, and the enclosing element is known as the **parent element**. In the code shown in **FIGURE A-6**, the `head` element is both a child of the `html` element and the parent of the `title` element. In addition, the `html` element is the **grandparent element** of the `title` element, which can be referred to as a **grandchild element** of the `html` element. Two elements that are both children of the same parent element are known as **sibling elements**. The `head` element and the `body` element are sibling elements, because they are both children of the `html` element.