



*NEW PERSPECTIVES*

COMPREHENSIVE

# HTML5 and CSS3

7th Edition

Carey



# Want to turn C's into A's? Obviously, right?

But the right way to go about it isn't always so obvious. Go digital to get the grades. MindTap's customizable study tools and eTextbook give you everything you need all in one place.

Engage with your course content, enjoy the flexibility of studying anytime and anywhere, stay connected to assignment due dates and instructor notifications with the MindTap Mobile app...

*and most of all...EARN BETTER GRADES.*



TO GET STARTED VISIT  
[WWW.CENGAGE.COM/STUDENTS/MINDTAP](http://WWW.CENGAGE.COM/STUDENTS/MINDTAP)



MindTap®

**NEW PERSPECTIVES ON**

# **HTML5 and CSS3**

*7th Edition*

---

**COMPREHENSIVE**

**Patrick Carey**



---

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

**New Perspectives on HTML5 and CSS3, 7th Edition,  
Comprehensive  
Patrick Carey**

Senior Product Director: Kathleen McMahon

Associate Product Manager: Kate Mason

Associate Content Development Manager: Alyssa  
Pratt

Development Editor: Pam Conrad

Product Assistant: Abby Pufpaff

Senior Content Project Manager:  
Jennifer K. Feltri-George

Art Director: Diana Graham

Manufacturing Planner: Fola Orekoya

Cover image(s): Shutterstock/AGCuesta

Compositor: GEX Publishing Services

#### Notice to the Reader

Publisher does not warrant or guarantee any of the products described herein or perform any independent analysis in connection with any of the product information contained herein. Publisher does not assume, and expressly disclaims, any obligation to obtain and include information other than that provided to it by the manufacturer. The reader is expressly warned to consider and adopt all safety precautions that might be indicated by the activities described herein and to avoid all potential hazards. By following the instructions contained herein, the reader willingly assumes all risks in connection with such instructions. The publisher makes no representations or warranties of any kind, including but not limited to, the warranties of fitness for particular purpose or merchantability, nor are any such representations implied with respect to the material set forth herein, and the publisher takes no responsibility with respect to such material. The publisher shall not be liable for any special, consequential, or exemplary damages resulting, in whole or part, from the readers' use of, or reliance upon, this material.

© 2017 Cengage Learning

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](http://www.cengage.com/permissions)  
Further permissions questions can be emailed to  
[permissionrequest@cengage.com](mailto:permissionrequest@cengage.com)

Library of Congress Control Number: 2016954611

ISBN: 978-1-305-50393-9

#### Cengage Learning

20 Channel Center Street

Boston, MA 02210

USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at:  
[www.cengage.com/global](http://www.cengage.com/global)

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

For your course and learning solutions, visit [www.cengage.com](http://www.cengage.com)

Purchase any of our products at your local college store or at our  
preferred online store [www.cengagebrain.com](http://www.cengagebrain.com)

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

Disclaimer: Any fictional data related to persons or companies or URLs used  
throughout this book is intended for instructional purposes only. At the  
time this book was printed, any such data was fictional and not belonging  
to any real persons or companies.

Microsoft and the Windows logo are registered trademarks of Microsoft  
Corporation in the United States and/or other countries. Cengage Learning  
is an independent entity from Microsoft Corporation, and not affiliated with  
Microsoft in any manner.

# Preface

The New Perspectives Series' critical-thinking, problem-solving approach is the ideal way to prepare students to transcend point-and-click skills and take advantage of all that HTML5 and CSS3 has to offer.

In developing the New Perspectives Series, our goal was to create books that give students the software concepts and practical skills they need to succeed beyond the classroom. We've updated our proven case-based pedagogy with more practical content to make learning skills more meaningful to students. With the New Perspectives Series, students understand *why* they are learning *what* they are learning, and are fully prepared to apply their skills to real-life situations.

*"I love this text because it provides detailed instructions and real-world application examples. It is ideal for classroom and online instruction. At the end of the term my students comment on how much they've learned and put to use outside the classroom."*

—Bernice Howard  
St. Johns River Community  
College

## About This Book

This book provides thorough coverage of HTML5 and CSS3, and includes the following:

- Up-to-date coverage of using HTML5 to create structured websites
- Instruction on the most current CSS3 styles to create visually-interesting pages and captivating graphical designs
- Working with browser developer tools to aid in the creation and maintenance of fully-functioning websites

*New for this edition!*

- Coverage of responsive design techniques to create website designs that can scale to mobile, tablet, and desktop devices.
- Hands-on study of new HTML elements and CSS styles including layouts using flexboxes and grid frameworks.
- Exploration of CSS3 styles for graphic design, including image borders, drop shadows, gradient fills, 2D and 3D transformations, and graphic filters.
- Exploration of responsive design for web tables.
- Coverage of CSS styles for animation and transitions.
- Coverage of JavaScript arrays, program loops, and conditional statements.

## System Requirements

This book assumes that students have an Internet connection, a text editor, and a current browser that supports HTML5 and CSS3. The following is a list of the most recent versions of the major browsers at the time this text was published: Internet Explorer 11, Microsoft Edge 15, Firefox 48.02, Safari 10, Opera 39.0, and Google Chrome 53. More recent versions may have come out since the publication of this book. Students should go to the Web browser home page to download the most current version. All browsers interpret HTML5 and CSS3 code in slightly different ways. It is highly recommended that students have several different browsers installed on their systems for comparison and, if possible, access to a mobile browser or a mobile emulator. Students might also want to run older versions of these browsers to highlight compatibility issues. The screenshots in this book were produced using Google Chrome 53 running on Windows 10 (64-bit), unless otherwise noted. If students are using different devices, browsers, or operating systems, their screens might vary from those shown in the book; this should not present any problems in completing the tutorials.



*"New Perspectives texts provide up-to-date, real-world application of content, making book selection easy. The step-by-step, hands-on approach teaches students concepts they can apply immediately."*

—John Taylor  
Southeastern Technical  
College

## VISUAL OVERVIEW

# The New Perspectives Approach

## Context

Each tutorial begins with a problem presented in a “real-world” case that is meaningful to students. The case sets the scene to help students understand what they will do in the tutorial.

## Hands-on Approach

Each tutorial is divided into manageable sessions that combine reading and hands-on, step-by-step work. Colorful screenshots help guide students through the steps. **Trouble?** tips, which anticipate common mistakes or problems, help students stay on track and continue with the tutorial.

## Visual Overviews

Each session begins with a Visual Overview, a two-page spread that includes colorful, enlarged figures with numerous callouts and key term definitions, giving students a comprehensive preview of the topics covered in the session, as well as a handy study guide.

## PROSKILLS

## ProSkills Boxes

ProSkills boxes provide guidance for applying concepts to real-world, professional situations, involving one or more of the following soft skills: decision making, problem solving, teamwork, verbal communication, and written communication.

## KEY STEP

## Key Steps

Important steps are highlighted in yellow with attached margin notes to help students pay close attention to completing the steps correctly and avoid time-consuming rework.

## INSIGHT

## InSight Boxes

InSight boxes offer expert advice and best practices to help students achieve a deeper understanding of the concepts behind the software features and skills.

## TIP

## Margin Tips

Margin Tips provide helpful hints and shortcuts for more efficient use of the software. The Tips appear in the margin at key points throughout each tutorial, giving students extra information when and where they need it.

## REVIEW

## Assessment

## APPLY

## CHALLENGE

## CREATE

Retention is a key component to learning. At the end of each session, a series of Quick Check questions helps students test their understanding of the material before moving on. Engaging end-of-tutorial Review Assignments and Case Problems have always been a hallmark feature of the New Perspectives Series. Colorful bars and brief descriptions accompany the exercises, making it easy to understand both the goal and level of challenge a particular assignment holds.

## REFERENCE

## Reference

Within each tutorial, Reference boxes appear before a set of steps to provide a succinct summary or preview of how to perform a task. In addition, each book includes a combination Glossary/Index to promote easy reference of material.

## GLOSSARY/INDEX

INTRODUCTORY

COMPREHENSIVE

## Our Complete System of Instruction

### Coverage To Meet Your Needs

Whether you're looking for just a small amount of coverage or enough to fill a semester-long class, we can provide you with a textbook that meets your needs.

- Introductory books contain an average of 5 to 8 tutorials and include essential skills on the books concepts.
- Comprehensive books, which cover additional concepts and skills in depth, are great for a full-semester class, and contain 9 to 12+ tutorials.

So, if you are looking for just the essential skills or more complete in-depth coverage of a topic, we have an offering available to meet your needs. Go to our Web site or contact your Cengage Learning sales representative to find out what else we offer.

### MindTap

MindTap is a personalized learning experience with relevant assignments that guide students to analyze, apply, and improve thinking, allowing you to measure skills and outcomes with ease.

For instructors: personalized teaching becomes yours with a Learning Path that is built with key student objectives. Control what students see and when they see it. Use as-is, or match to your syllabus exactly: hide, rearrange, add, or create your own content.

For students: a unique Learning Path of relevant readings, multimedia, and activities that guide you through basic knowledge and comprehension to analysis and application.

Better outcomes: empower instructors and motivate students with analytics and reports that provide a snapshot of class progress, time in course, engagement, and completion rates.

The MindTap for HTML5 and CSS3 includes coding labs, study tools, and interactive quizzing, all integrated into an eReader that includes the full content of the printed text.

### Instructor Resources

We offer more than just a book. We have all the tools you need to enhance your lectures, check students' work, and generate exams in a new, easier-to-use and completely revised package. This book's Instructor's Manual, Cognition testbank, PowerPoint presentations, data files, solution files, figure files, and a sample syllabus are all available at [sso.cengage.com](http://sso.cengage.com).

## Acknowledgments

I would like to thank the people who worked so hard to make this book possible. Special thanks to my developmental editor, Pam Conrad, for her hard work, attention to detail, and valuable insights, and to Associate Content Developer Manager, Alyssa Pratt, who has worked tirelessly in overseeing this project and made my task so much easier with enthusiasm and good humor. Other people at Cengage who deserve credit are Kathleen McMahan, Sr. Product Manager; Kate Mason, Associate Project Manager; Abby Pufpaff, Product Assistant; Jen Feltri-George, Senior Content Project Manager; Diana Graham, Art Director; Fola Orekoya, Manufacturing Planner; GEX Publishing Services, Composer, as well as Chris Scriver - Sr. MQA Project Leader, and the MQA testers John Freitas and Danielle Shaw.

Feedback is an important part of writing any book, and thanks go to the following reviewers for their helpful ideas and comments: Alison Consol, Wake Technical Community College; Dana Hooper, The University of Alabama; Kenneth Kleiner, Fayetteville Technical Community College; and Laurie Crawford, Franklin University.

I want to thank my wife Joan and my six children for their love, encouragement, and patience in putting up with a sometimes distracted husband and father. This book is dedicated to my grandchildren: Benedict, David, and Elanor.

– Patrick Carey



# BRIEF CONTENTS

## HTML

### Level I Tutorials

- Tutorial 1** Getting Started with HTML5 . . . . . HTML 1  
*Creating a Website for a Food Vendor*
- Tutorial 2** Getting Started with CSS . . . . . HTML 83  
*Designing a Website for a Fitness Club*

### Level II Tutorials

- Tutorial 3** Designing a Page Layout . . . . . HTML 169  
*Creating a Website for a Chocolatier*
- Tutorial 4** Graphic Design with CSS . . . . . HTML 257  
*Creating a Graphic Design for a Genealogy Website*
- Tutorial 5** Designing for the Mobile Web . . . . . HTML 341  
*Creating a Mobile Website for a Daycare Center*

### Level III Tutorials

- Tutorial 6** Working with Tables and Columns. . . . . HTML 433  
*Creating a Program Schedule for a Radio Station*
- Tutorial 7** Designing a Web Form. . . . . HTML 499  
*Creating a Survey Form*
- Tutorial 8** Enhancing a Website with Multimedia . . . . . HTML 585  
*Working with Sound, Video, and Animation*
- Tutorial 9** Getting Started with JavaScript. . . . . HTML 665  
*Creating a Countdown Clock*
- Tutorial 10** Exploring Arrays, Loops, and Conditional Statements. . . . . HTML 735  
*Creating a Monthly Calendar*
- Appendix A** Color Names with Color Values, and HTML Character  
 Entities . . . . . HTML A1
- Appendix B** HTML Elements and Attributes . . . . . HTML B1
- Appendix C** Cascading Styles and Selectors . . . . . HTML C1
- Appendix D** Making the Web More Accessible. . . . . HTML D1
- Appendix E** Designing for the Web . . . . . HTML E1
- Appendix F** Page Validation with XHTML. . . . . HTML F1

### Glossary

REF 1

### Index

REF 11



# TABLE OF CONTENTS

**Preface** ..... iii

---

## HTML LEVEL I TUTORIALS

### Tutorial 1 Getting Started with HTML5

*Creating a Website for a Food Vendor* ..... **HTML 1**

#### **SESSION 1.1** ..... **HTML 2**

Exploring the World Wide Web ..... HTML 4

    Networks ..... HTML 4

    Locating Information on a Network ..... HTML 4

    Web Pages and Web Servers ..... HTML 5

Introducing HTML ..... HTML 5

    The History of HTML ..... HTML 5

Tools for Working with HTML ..... HTML 6

    Testing your Code ..... HTML 7

    Supporting the Mobile Web ..... HTML 7

Exploring an HTML Document ..... HTML 7

    The Document Type Declaration ..... HTML 8

    Introducing Element Tags ..... HTML 9

    The Element Hierarchy ..... HTML 10

    Introducing Element Attributes ..... HTML 11

    Handling White Space ..... HTML 12

    Viewing an HTML File in a Browser ..... HTML 12

    Creating an HTML File ..... HTML 14

Creating the Document Head ..... HTML 15

    Setting the Page Title ..... HTML 15

    Adding Metadata to the Document ..... HTML 16

Adding Comments to your Document ..... HTML 18

Session 1.1 Quick Check ..... HTML 21

#### **SESSION 1.2** ..... **HTML 22**

Writing the Page Body ..... HTML 24

    Using Sectioning Elements ..... HTML 24

    Comparing Sections in HTML4 and HTML5 ..... HTML 26

    Using Grouping Elements ..... HTML 26

    Using Text-Level Elements ..... HTML 29

Linking an HTML Document to a Style Sheet ..... HTML 32

Working with Character Sets and Special Characters ... HTML 33

    Character Encoding ..... HTML 33

    Character Entity References ..... HTML 34

Working with Inline Images ..... HTML 36

    Line Breaks and Other Empty Elements ..... HTML 38

Working with Block Quotes and Other Elements ... HTML 39

Session 1.2 Quick Check ..... HTML 45

#### **SESSION 1.3** ..... **HTML 46**

Working with Lists ..... HTML 48

    Ordered Lists ..... HTML 48

    Unordered Lists ..... HTML 49

    Description Lists ..... HTML 51

    Navigation Lists ..... HTML 55

Working with Hypertext Links ..... HTML 57

    Turning an Inline Image into a Link ..... HTML 59

Specifying the Folder Path ..... HTML 60

    Absolute Paths ..... HTML 61

    Relative Paths ..... HTML 61

    Setting the Base Path ..... HTML 62

Linking to a Location within a Document ..... HTML 63

    Marking Locations with the `id` Attribute ..... HTML 63

    Linking to an `id` ..... HTML 63

    Anchors and the `name` Attribute ..... HTML 63

Linking to the Internet and Other Resources . . . . .	HTML 64
Linking to a Web Resource . . . . .	HTML 65
Linking to an E-Mail Address . . . . .	HTML 65
Linking to a Phone Number . . . . .	HTML 67
Working with Hypertext Attributes. . . . .	HTML 68
Session 1.3 Quick Check . . . . .	HTML 70
Review Assignments . . . . .	HTML 71
Case Problems. . . . .	HTML 74
<b>Tutorial 2 Getting Started with CSS</b>	
<i>Designing a Website for a Fitness Club</i> . . . . .	<b>HTML 83</b>
<b>SESSION 2.1</b> . . . . .	<b>HTML 84</b>
Introducing CSS . . . . .	HTML 86
Types of Style Sheets . . . . .	HTML 86
Viewing a Page Using Different Style Sheets. . .	HTML 87
Exploring Style Rules. . . . .	HTML 90
Browser Extensions. . . . .	HTML 90
Embedded Style Sheets . . . . .	HTML 91
Inline Styles . . . . .	HTML 92
Style Specificity and Precedence. . . . .	HTML 92
Style Inheritance . . . . .	HTML 93
Browser Developer Tools . . . . .	HTML 93
Creating a Style Sheet. . . . .	HTML 95
Writing Style Comments . . . . .	HTML 95
Defining the Character Encoding. . . . .	HTML 96
Importing Style Sheets . . . . .	HTML 96
Working with Color in CSS . . . . .	HTML 97
Color Names. . . . .	HTML 97
RGB Color Values . . . . .	HTML 98
HSL Color Values. . . . .	HTML 99
Defining Semi-Opaque Colors . . . . .	HTML 100
Setting Text and Background Colors . . . . .	HTML 101

Employing Progressive Enhancement . . . . .	HTML 104
Session 2.1 Quick Check . . . . .	HTML 105
<b>SESSION 2.2</b> . . . . .	<b>HTML 106</b>
Exploring Selector Patterns . . . . .	HTML 108
Contextual Selectors . . . . .	HTML 108
Attribute Selectors. . . . .	HTML 111
Working with Fonts . . . . .	HTML 115
Choosing a Font . . . . .	HTML 115
Exploring Web Fonts . . . . .	HTML 118
The @font-face Rule. . . . .	HTML 118
Setting the Font Size. . . . .	HTML 121
Absolute Units . . . . .	HTML 121
Relative Units . . . . .	HTML 121
Scaling Fonts with ems and rems. . . . .	HTML 122
Using Viewport Units . . . . .	HTML 123
Sizing Keywords. . . . .	HTML 123
Controlling Spacing and Indentation . . . . .	HTML 125
Working with Font Styles. . . . .	HTML 127
Aligning Text Horizontally and Vertically. . . . .	HTML 128
Combining All Text Formatting in a Single Style. . .	HTML 128
Session 2.2 Quick Check. . . . .	HTML 131
<b>SESSION 2.3</b> . . . . .	<b>HTML 132</b>
Formatting Lists . . . . .	HTML 134
Choosing a List Style Type . . . . .	HTML 134
Creating an Outline Style. . . . .	HTML 134
Using Images for List Markers . . . . .	HTML 137
Setting the List Marker Position. . . . .	HTML 138
Working with Margins and Padding . . . . .	HTML 139
Setting the Padding Space. . . . .	HTML 140
Setting the Margin and the Border Spaces . . . .	HTML 142



Using Pseudo-Classes and Pseudo-Elements . . . . .	HTML 145
Pseudo-Classes . . . . .	HTML 145
Pseudo-classes for Hypertext . . . . .	HTML 148
Pseudo-Elements . . . . .	HTML 151
Generating Content with CSS . . . . .	HTML 152
Displaying Attribute Values . . . . .	HTML 153
Inserting Quotation Marks . . . . .	HTML 154
Session 2.3 Quick Check . . . . .	HTML 157
Review Assignments . . . . .	HTML 158
Case Problems . . . . .	HTML 160

---

## HTML LEVEL II TUTORIALS

### Tutorial 3 Designing a Page Layout

*Creating a Website for a Chocolatier . . . . .* **HTML 169**

#### **SESSION 3.1 . . . . .HTML 170**

Introducing the <code>display</code> Style . . . . .	HTML 172
Creating a Reset Style Sheet . . . . .	HTML 172
Exploring Page Layout Designs . . . . .	HTML 176
Fixed, Fluid, and Elastic Layouts . . . . .	HTML 176
Working with Width and Height . . . . .	HTML 178
Setting Maximum and Minimum Dimensions . . . . .	HTML 178
Centering a Block Element . . . . .	HTML 181
Vertical Centering . . . . .	HTML 182
Floating Page Content . . . . .	HTML 183
Clearing a Float . . . . .	HTML 187
Refining a Floated Layout . . . . .	HTML 191
Working with Container Collapse . . . . .	HTML 195
Session 3.1 Quick Check . . . . .	HTML 199

#### **SESSION 3.2 . . . . .HTML 200**

Introducing Grid Layouts . . . . .	HTML 202
Overview of Grid-Based Layouts . . . . .	HTML 202

Fixed and Fluid Grids . . . . .	HTML 203
CSS Frameworks . . . . .	HTML 204
Setting up a Grid . . . . .	HTML 204
Designing the Grid Rows . . . . .	HTML 208
Designing the Grid Columns . . . . .	HTML 209
Adding the Page Content . . . . .	HTML 210
Outlining a Grid . . . . .	HTML 216
Introducing CSS Grids . . . . .	HTML 219
Defining a CSS Grid . . . . .	HTML 219
Assigning Content to Grid Cells . . . . .	HTML 220
Session 3.2 Quick Check . . . . .	HTML 223

#### **SESSION 3.3 . . . . .HTML 224**

Positioning Objects . . . . .	HTML 226
The CSS Positioning Styles . . . . .	HTML 226
Relative Positioning . . . . .	HTML 226
Absolute Positioning . . . . .	HTML 227
Fixed and Inherited Positioning . . . . .	HTML 230
Using the Positioning Styles . . . . .	HTML 230
Handling Overflow . . . . .	HTML 240
Clipping an Element . . . . .	HTML 243
Stacking Elements . . . . .	HTML 244
Session 3.3 Quick Check . . . . .	HTML 246
Review Assignments . . . . .	HTML 247
Case Problems . . . . .	HTML 249

### Tutorial 4 Graphic Design with CSS

*Creating a Graphic Design for a Genealogy Website . . . . .* **HTML 257**

#### **SESSION 4.1 . . . . .HTML 258**

Creating Figure Boxes . . . . .	HTML 260
Exploring Background Styles . . . . .	HTML 264
Tiling a Background Image . . . . .	HTML 265
Attaching the Background Image . . . . .	HTML 267

Setting the Background Image Position . . . . .	HTML 267	Review Assignments . . . . .	HTML 331
Defining the Extent of the Background . . . . .	HTML 268	Case Problems . . . . .	HTML 334
Sizing and Clipping an Image . . . . .	HTML 269	<b>Tutorial 5 Designing for the Mobile Web</b>	
The background Property . . . . .	HTML 270	<i>Creating a Mobile Website for a Daycare Center . . . . .</i>	<b>HTML 341</b>
Adding Multiple Backgrounds . . . . .	HTML 272	<b>SESSION 5.1 . . . . .</b>	<b>HTML 342</b>
Working with Borders . . . . .	HTML 273	Introducing Responsive Design . . . . .	HTML 344
Setting Border Width and Color . . . . .	HTML 274	Introducing Media Queries . . . . .	HTML 345
Setting the Border Design . . . . .	HTML 274	The @media Rule . . . . .	HTML 346
Creating Rounded Corners . . . . .	HTML 277	Media Queries and Device Features . . . . .	HTML 347
Applying a Border Image . . . . .	HTML 281	Applying Media Queries to a Style Sheet . . . . .	HTML 349
Session 4.1 Quick Check . . . . .	HTML 285	Exploring Viewports and Device Width . . . . .	HTML 352
<b>SESSION 4.2 . . . . .</b>	<b>HTML 286</b>	Creating a Mobile Design . . . . .	HTML 355
Creating Drop Shadows . . . . .	HTML 288	Creating a Pulldown Menu with CSS . . . . .	HTML 356
Creating a Text Shadow . . . . .	HTML 288	Testing your Mobile Website . . . . .	HTML 359
Creating a Box Shadow . . . . .	HTML 290	Creating a Tablet Design . . . . .	HTML 363
Applying a Color Gradient . . . . .	HTML 296	Creating a Desktop Design . . . . .	HTML 367
Creating a Linear Gradient . . . . .	HTML 296	Session 5.1 Quick Check . . . . .	HTML 371
Gradients and Color Stops . . . . .	HTML 299	<b>SESSION 5.2 . . . . .</b>	<b>HTML 372</b>
Creating a Radial Gradient . . . . .	HTML 301	Introducing Flexible Boxes . . . . .	HTML 374
Repeating a Gradient . . . . .	HTML 305	Defining a Flexible Box . . . . .	HTML 374
Creating Semi-Transparent Objects . . . . .	HTML 307	Cross-Browser Flexboxes . . . . .	HTML 375
Session 4.2 Quick Check . . . . .	HTML 309	Setting the Flexbox Flow . . . . .	HTML 375
<b>SESSION 4.3 . . . . .</b>	<b>HTML 310</b>	Working with Flex Items . . . . .	HTML 377
Transforming Page Objects . . . . .	HTML 312	Setting the Flex Basis . . . . .	HTML 377
Transformations in Three Dimensions . . . . .	HTML 316	Defining the Flex Growth . . . . .	HTML 378
Understanding Perspective . . . . .	HTML 317	Defining the Shrink Rate . . . . .	HTML 379
Exploring CSS Filters . . . . .	HTML 320	The flex Property . . . . .	HTML 381
Working with Image Maps . . . . .	HTML 324	Applying a Flexbox Layout . . . . .	HTML 382
Defining a Client-Side Image Map . . . . .	HTML 324	Reordering Page Content with Flexboxes . . . . .	HTML 388
Applying an Image Map . . . . .	HTML 328	Exploring Flexbox Layouts . . . . .	HTML 390
Session 4.3 Quick Check . . . . .	HTML 330	Aligning Items along the Main Axis . . . . .	HTML 390

Aligning Flex Lines . . . . .	HTML 391
Aligning Items along the Cross Axis . . . . .	HTML 392
Creating a Navicon Menu . . . . .	HTML 394
Session 5.2 Quick Check . . . . .	HTML 399
<b>SESSION 5.3 . . . . .</b>	<b>HTML 400</b>
Designing for Printed Media . . . . .	HTML 402
Previewing the Print Version . . . . .	HTML 402
Applying a Media Query for Printed Output. . . . .	HTML 403
Working with the @page Rule . . . . .	HTML 405
Setting the Page Size . . . . .	HTML 405
Using the Page Pseudo-Classes . . . . .	HTML 406
Page Names and the Page Property . . . . .	HTML 406
Formatting Hypertext Links for Printing . . . . .	HTML 412
Working with Page Breaks. . . . .	HTML 415
Preventing Page Breaks . . . . .	HTML 416
Working with Widows and Orphans. . . . .	HTML 418
Session 5.3 Quick Check . . . . .	HTML 421
Review Assignments . . . . .	HTML 422
Case Problems. . . . .	HTML 425

---

## HTML LEVEL III TUTORIALS

### Tutorial 6 Working with Tables and Columns

*Creating a Program Schedule for a Radio Station. . . . .* **HTML 433**

#### **SESSION 6.1 . . . . .**

Introducing Web Tables . . . . . HTML 436

    Marking Tables and Table Rows. . . . . HTML 436

    Marking Table Headings and Table Data . . . . . HTML 438

Adding Table Borders with CSS . . . . . HTML 442

Spanning Rows and Columns . . . . . HTML 447

Creating a Table Caption . . . . . HTML 453

Session 6.1 Quick Check . . . . . HTML 457

#### **SESSION 6.2 . . . . .**

Creating Row Groups . . . . . HTML 460

Creating Column Groups. . . . . HTML 464

Exploring CSS Styles and Web Tables. . . . . HTML 467

    Working with Width and Height. . . . . HTML 468

    Applying Table Styles to Other  
    Page Elements . . . . . HTML 472

Tables and Responsive Design. . . . . HTML 474

Designing a Column Layout . . . . . HTML 478

    Setting the Number of Columns . . . . . HTML 478

    Defining Columns Widths and Gaps. . . . . HTML 481

    Managing Column Breaks . . . . . HTML 484

    Spanning Cell Columns . . . . . HTML 485

Session 6.2 Quick Check . . . . . HTML 488

Review Assignments . . . . . HTML 489

Case Problems. . . . . HTML 491

### Tutorial 7 Designing a Web Form

*Creating a Survey Form. . . . .* **HTML 499**

#### **SESSION 7.1 . . . . .**

Introducing Web Forms. . . . . HTML 502

    Parts of a Web Form. . . . . HTML 502

    Forms and Server-Based Programs . . . . . HTML 503

Starting a Web Form . . . . . HTML 504

    Interacting with the Web Server . . . . . HTML 505

Creating a Field Set. . . . . HTML 507

    Marking a Field Set. . . . . HTML 507

    Adding a Field Set Legend. . . . . HTML 508

Creating Input Boxes. . . . . HTML 510

    Input Types. . . . . HTML 510

    Input Types and Virtual Keyboards . . . . . HTML 514

Adding Field Labels. . . . . HTML 514

Designing a Form Layout. . . . . HTML 516

Defining Default Values and Placeholders . . . . .	HTML 523
Session 7.1 Quick Check. . . . .	HTML 527
<b>SESSION 7.2. . . . .</b>	<b>HTML 528</b>
Entering Date and Time Values. . . . .	HTML 530
Creating a Selection List . . . . .	HTML 531
Working with Select Attributes . . . . .	HTML 533
Grouping Selection Options. . . . .	HTML 535
Creating Option Buttons. . . . .	HTML 537
Creating Check Boxes . . . . .	HTML 540
Creating a Text Area Box. . . . .	HTML 542
Session 7.2 Quick Check . . . . .	HTML 545
<b>SESSION 7.3. . . . .</b>	<b>HTML 546</b>
Entering Numeric Data . . . . .	HTML 548
Creating a Spinner Control . . . . .	HTML 548
Creating a Range Slider . . . . .	HTML 550
Suggesting Options with Data Lists . . . . .	HTML 553
Working with Form Buttons. . . . .	HTML 556
Creating a Command Button. . . . .	HTML 556
Creating Submit and Reset Buttons. . . . .	HTML 556
Designing a Custom Button . . . . .	HTML 559
Validating a Web Form . . . . .	HTML 559
Identifying Required Values. . . . .	HTML 559
Validating Based on Data Type. . . . .	HTML 561
Testing for a Valid Pattern . . . . .	HTML 562
Defining the Length of the Field Value. . . . .	HTML 564
Applying Inline Validation. . . . .	HTML 565
Using the <code>focus</code> Pseudo-Class . . . . .	HTML 565
Pseudo-Classes for Valid and Invalid Data. . . . .	HTML 567
Session 7.3 Quick Check . . . . .	HTML 570
Review Assignments . . . . .	HTML 571
Case Problems. . . . .	HTML 574

## Tutorial 8 Enhancing a Website with Multimedia

*Working with Sound, Video, and Animation . . . . .* **HTML 585**

### **SESSION 8.1. . . . .**

Introducing Multimedia on the Web. . . . .	HTML 588
Understanding Codecs and Containers. . . . .	HTML 588
Understanding Plug-Ins. . . . .	HTML 589
Working with the <code>audio</code> Element . . . . .	HTML 591
Browsers and Audio Formats . . . . .	HTML 591
Applying Styles to the Media Player . . . . .	HTML 594
Providing a Fallback to an Audio Clip . . . . .	HTML 596
Exploring Embedded Objects. . . . .	HTML 598
Plug-In Attributes. . . . .	HTML 598
Plug-Ins as Fallback Options . . . . .	HTML 599
Session 8.1 Quick Check . . . . .	HTML 599

### **SESSION 8.2. . . . .**

Exploring Digital Video . . . . .	HTML 602
Video Formats and Codecs. . . . .	HTML 602
Using the HTML5 <code>video</code> Element . . . . .	HTML 603
Adding a Text Track to Video . . . . .	HTML 607
Making Tracks with WebVTT . . . . .	HTML 608
Placing the Cue Text. . . . .	HTML 611
Applying Styles to Track Cues . . . . .	HTML 612
Using Third-Party Video Players . . . . .	HTML 616
Exploring the Flash Player . . . . .	HTML 617
Embedding Videos from YouTube. . . . .	HTML 618
HTML5 Video Players . . . . .	HTML 619
Session 8.2 Quick Check . . . . .	HTML 621

### **SESSION 8.3. . . . .**

Creating Transitions with CSS. . . . .	HTML 624
Introducing Transitions . . . . .	HTML 624
Setting the Transition Timing. . . . .	HTML 626



Delaying a Transition . . . . .	HTML 629
Creating a Hover Transition . . . . .	HTML 629
Animating Objects with CSS . . . . .	HTML 634
The <code>@keyframes</code> Rule . . . . .	HTML 634
Applying an Animation . . . . .	HTML 637
Controlling an Animation . . . . .	HTML 640
Session 8.3 Quick Check . . . . .	HTML 651
Review Assignments . . . . .	HTML 652
Case Problems . . . . .	HTML 655

## Tutorial 9 Getting Started with JavaScript

<i>Creating a Countdown Clock</i> . . . . .	<b>HTML 665</b>
---	-----------------

### SESSION 9.1 . . . . .HTML 666

Introducing JavaScript . . . . .	HTML 668
Server-Side and Client-Side Programming . . . . .	HTML 668
The Development of JavaScript . . . . .	HTML 669
Working with the <code>script</code> Element . . . . .	HTML 670
Loading the <code>script</code> Element . . . . .	HTML 670
Inserting the <code>script</code> Element . . . . .	HTML 671
Creating a JavaScript Program . . . . .	HTML 673
Adding Comments to your JavaScript Code . . . . .	HTML 673
Writing a JavaScript Command . . . . .	HTML 674
Understanding JavaScript Syntax . . . . .	HTML 675
Debugging your Code . . . . .	HTML 677
Opening a Debugger . . . . .	HTML 677
Inserting a Breakpoint . . . . .	HTML 679
Applying Strict Usage of JavaScript . . . . .	HTML 680
Session 9.1 Quick Check . . . . .	HTML 681

### SESSION 9.2 . . . . .HTML 682

Introducing Objects . . . . .	HTML 684
Object References . . . . .	HTML 685
Referencing Object Collections . . . . .	HTML 685
Referencing an Object by ID and Name . . . . .	HTML 687

Changing Properties and Applying Methods . . . . .	HTML 688
Object Properties . . . . .	HTML 688
Applying a Method . . . . .	HTML 688
Writing HTML Code . . . . .	HTML 689
Working with Variables . . . . .	HTML 693
Declaring a Variable . . . . .	HTML 693
Variables and Data Types . . . . .	HTML 694
Using a Variable . . . . .	HTML 695
Working with Date Objects . . . . .	HTML 695
Creating a Date Object . . . . .	HTML 696
Applying Date Methods . . . . .	HTML 697
Setting Date and Time Values . . . . .	HTML 700
Session 9.2 Quick Check . . . . .	HTML 701

### SESSION 9.3 . . . . .HTML 702

Working with Operators and Operands . . . . .	HTML 704
Using Assignment Operators . . . . .	HTML 704
Calculating the Days Left in the Year . . . . .	HTML 705
Working with the Math Object . . . . .	HTML 707
Using Math Methods . . . . .	HTML 707
Using Math Constants . . . . .	HTML 712
Working with JavaScript Functions . . . . .	HTML 714
Calling a Function . . . . .	HTML 716
Creating a Function to Return a Value . . . . .	HTML 717
Running Timed Commands . . . . .	HTML 718
Working with Time-Delayed Commands . . . . .	HTML 718
Running Commands at Specified Intervals . . . . .	HTML 718
Controlling How JavaScript Works with Numeric Values . . . . .	HTML 720
Handling Illegal Operations . . . . .	HTML 720
Defining a Number Format . . . . .	HTML 721
Converting Between Numbers and Text . . . . .	HTML 721
Session 9.3 Quick Check . . . . .	HTML 723

Review Assignments . . . . .	HTML 724
Case Problems . . . . .	HTML 726

## **Tutorial 10 Exploring Arrays, Loops, and Conditional Statements**

<i>Creating a Monthly Calendar</i> . . . . .	<b>HTML 735</b>
--	-----------------

### **SESSION 10.1 . . . . .HTML 736**

Introducing the Monthly Calendar . . . . .	HTML 738
Reviewing the Calendar Structure . . . . .	HTML 739
Adding the calendar() Function . . . . .	HTML 740
Introducing Arrays . . . . .	HTML 741
Creating and Populating an Array . . . . .	HTML 742
Working with Array Length . . . . .	HTML 745
Reversing an Array . . . . .	HTML 747
Sorting an Array . . . . .	HTML 748
Extracting and Inserting Array Items . . . . .	HTML 749
Using Arrays as Data Stacks . . . . .	HTML 750
Session 10.1 Quick Check . . . . .	HTML 753

### **SESSION 10.2 . . . . .HTML 754**

Working with Program Loops . . . . .	HTML 756
Exploring the for Loop . . . . .	HTML 756
Exploring the while Loop . . . . .	HTML 758
Exploring the do/while Loop . . . . .	HTML 759
Comparison and Logical Operators . . . . .	HTML 760
Program Loops and Arrays . . . . .	HTML 761
Array Methods to Loop Through Arrays . . . . .	HTML 764
Running a Function for Each Array Item . . . . .	HTML 765
Mapping an Array . . . . .	HTML 765
Filtering an Array . . . . .	HTML 766
Session 10.2 Quick Check . . . . .	HTML 769

### **SESSION 10.3 . . . . .HTML 770**

Introducing Conditional Statements . . . . .	HTML 772
Exploring the if Statement . . . . .	HTML 773
Nesting if Statements . . . . .	HTML 775
Exploring the if else Statement . . . . .	HTML 777
Using Multiple else if Statements . . . . .	HTML 778
Completing the Calendar App . . . . .	HTML 780
Setting the First Day of the Month . . . . .	HTML 781
Placing the First Day of the Month . . . . .	HTML 782
Writing the Calendar Days . . . . .	HTML 783
Highlighting the Current Date . . . . .	HTML 785
Displaying Daily Events . . . . .	HTML 787
Managing Program Loops and Conditional Statements . . . . .	HTML 790
Exploring the break Command . . . . .	HTML 790
Exploring the continue Command . . . . .	HTML 790
Exploring Statement Labels . . . . .	HTML 791
Session 10.3 Quick Check . . . . .	HTML 793
Review Assignments . . . . .	HTML 794
Case Problems . . . . .	HTML 796

## **Appendix A Color Names with Color Values, and HTML Character Entities . . . . .HTML A1**

## **Appendix B HTML Elements and Attributes . . . . .HTML B1**

## **Appendix C Cascading Styles and Selectors . . . . .HTML C1**

## **Appendix D Making the Web More Accessible . . . . .HTML D1**

## **Appendix E Designing for the Web . . . . .HTML E1**

## **Appendix F Page Validation with XHTML . . . . .HTML F1**

## **GLOSSARY . . . . . REF 1**

## **INDEX . . . . .REF 11**

## OBJECTIVES

**Session 1.1**

- Explore the history of the web
- Create the structure of an HTML document
- Insert HTML elements and attributes
- Insert metadata into a document
- Define a page title

**Session 1.2**

- Mark page structures with sectioning elements
- Organize page content with grouping elements
- Mark content with text-level elements
- Insert inline images
- Insert symbols based on character codes

**Session 1.3**

- Mark content using lists
- Create a navigation list
- Link to files within a website with hypertext links
- Link to e-mail addresses and telephone numbers

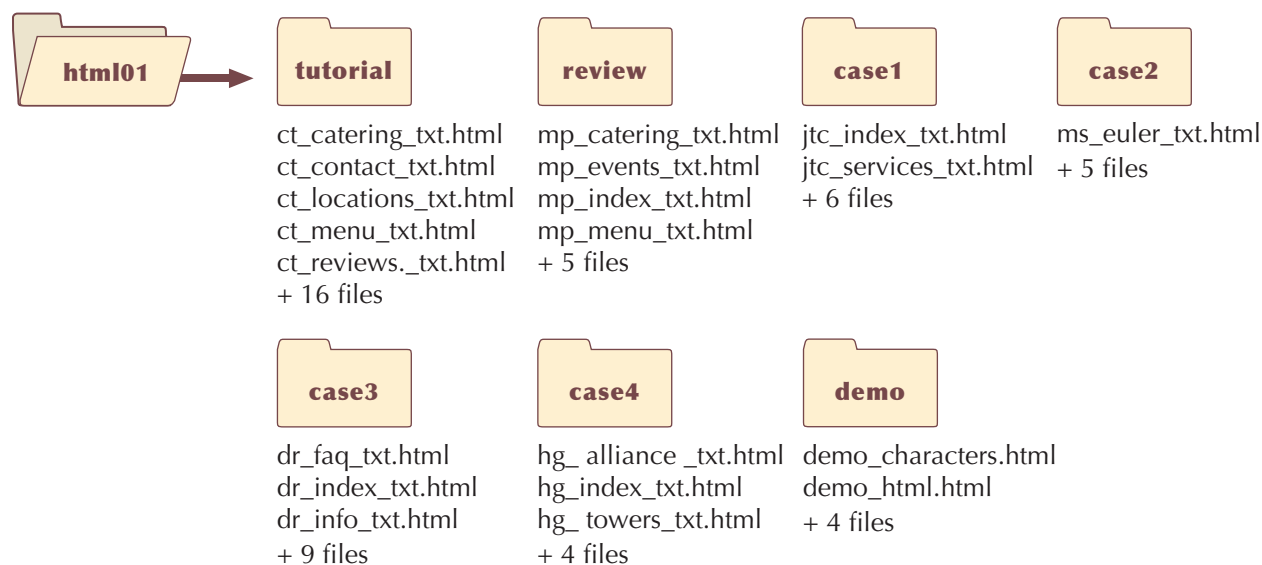
# Getting Started with HTML5

## *Creating a Website for a Food Vendor*

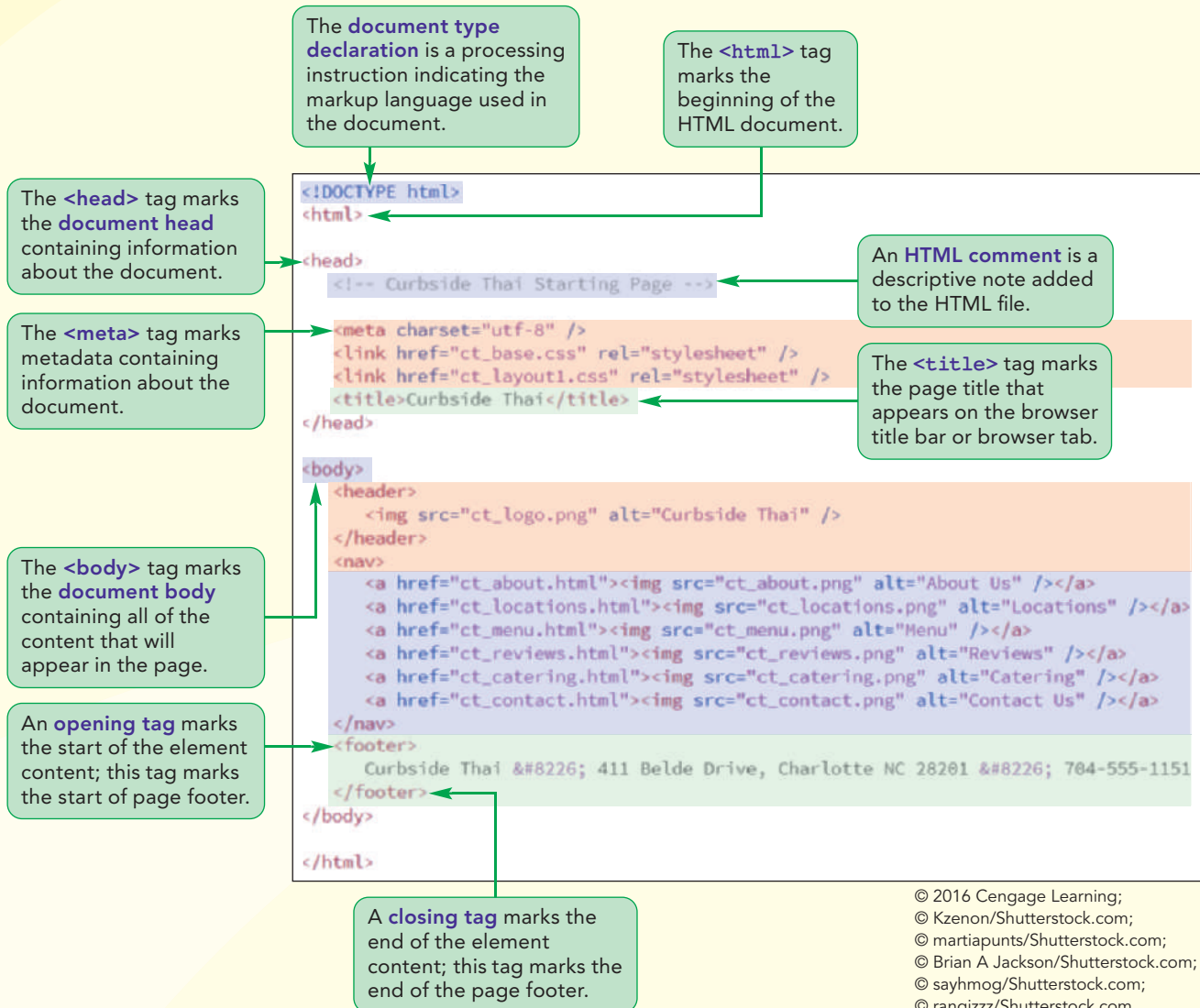
### Case | *Curbside Thai*

Sajja Adulet is the owner and master chef of Curbside Thai, a restaurant owner and now food truck vendor in Charlotte, North Carolina that specializes in Thai dishes. Sajja has hired you to develop the company's website. The website will display information about Curbside Thai including the truck's daily locations, menu, catering opportunities, and contact information. Sajja wants the pages to convey the message that customers will get the same great food and service whether they order in the restaurant or from the food truck. Some of the materials for these pages have already been completed by a former employee and Sajja needs you to finish the job by converting that work into a collection of web page documents. To complete this task, you'll learn how to write and edit HTML5 code and how to get your HTML files ready for display on the World Wide Web.

## STARTING DATA FILES



# Session 1.1 Visual Overview:





# The Structure of an HTML Document

Document as it appears in the browser.



The exact layout of the document elements is determined by a style sheet and not by the document markup.

## Exploring the World Wide Web

It is no exaggeration to say that the World Wide Web has had as profound an effect on human communication as the printing press. One key difference is that operation of the printing press was limited to a few select tradesmen but on the web everyone has his or her own printing press; everyone can be a publisher of a website. Before creating your first website, you'll examine a short history of the web because that history impacts the way you write code for your web pages. You'll start by exploring the basic terminology of computer networks.

### Networks

A **network** is a structure in which information and services are shared among devices known as **nodes** or **hosts**. A host can be any device that is capable of sending and/or receiving data electronically. The most common hosts that you will work with are desktop computers, laptops, tablets, mobile phones, and printers.

A host that provides information or a service to other devices on the network is called a **server**. For example, a print server is a network host that provides printing services and a file server is a host that provides storage space for saving and retrieving files. The device that receives these services is called a **client**. A common network design is the **client-server network**, in which the clients access information provided by one or more servers. You might be using such a network to access your data files for this tutorial.

Networks are classified based on the range of devices they cover. A network confined to a small geographic area, such as within a building or department, is referred to as a **local area network** or **LAN**. A network that covers a wider area, such as several buildings or cities, is called a **wide area network** or **WAN**. Wide area networks typically consist of two or more interconnected local area networks. The largest WAN in existence is the **Internet**, which incorporates an almost uncountable number of networks and hosts involving computers, mobile devices (such as phones, tablets, and so forth), MP3 players, and gaming systems.

### Locating Information on a Network

The biggest obstacle to effectively using the Internet is the network's sheer scope and size. Most of the early Internet tools required users to master a bewildering array of terms, acronyms, and commands. Because network users had to be well versed in computers and network technology, Internet use was largely limited to programmers and computer specialists working for universities, large businesses, and the government.

The solution to this problem was developed in 1989 by Timothy Berners-Lee and other researchers at the CERN nuclear research facility near Geneva, Switzerland. They needed an information system that would make it easy for their researchers to locate and share data on the CERN network. To meet this need, they developed a system of hypertext documents. **Hypertext** is a method of organization in which data sources are interconnected through a series of links or **hyperlinks** that users activate to jump from one data source to another. Hypertext is ideally suited for the Internet because end users don't need to know where a particular document, information source, or service is located—they only need to know how to activate the link. The effectiveness of this technique quickly spread beyond Geneva and was adopted with other networks across the Internet. The totality of these interconnected hypertext documents became known as the **World Wide Web**. The fact that the Internet and the World Wide Web are synonymous in many users' minds is a testament to the success of the hypertext approach.

## Web Pages and Web Servers

Documents on the web are stored on **web servers** in the form of **web pages** and accessed through a software program called a **web browser**. The browser retrieves the document from the web server and renders it locally in a form that is readable on a client device. However, because there is a wide selection of client devices ranging from desktop computers to mobile phones to screen readers that relay data aurally, each web page must be written in code that is compatible with every device. How does the same document work with so many different devices? To understand, you need to look at how web pages are created.

## Introducing HTML

A web page is a simple text file written in **HTML (Hypertext Markup Language)**. You've already read about hypertext, but what is a markup language? A **markup language** is a language that describes the content and structure of a document by "marking up" or tagging, different document elements. For example, this tutorial contains several document elements such as the tutorial title, main headings, subheadings, paragraphs, figures, figure captions, and so forth. Using a markup language, each of these elements could be tagged as a distinct item within the "tutorial document." Thus, a Hypertext Markup Language is a language that supports both the tagging of distinct document elements and connecting documents through hypertext links.

## The History of HTML

In the early years, no single organization defined the rules or **syntax** of HTML. Browser developers were free to define and modify the language in different ways which, of course, led to problems as different browsers supported different "flavors" of HTML and a web page that was written based on one browser's standard might appear totally different when rendered by another browser. Ultimately, a group of web designers and programmers called the **World Wide Web Consortium**, or the **W3C**, settled on a set of standards or specifications for all browser manufacturers to follow. The W3C has no enforcement power, but, because using a uniform language is in everyone's best interest, the W3C's recommendations are usually followed, though not always immediately. Each new version of HTML goes through years of discussion and testing before it is formally adopted as the accepted standard. For more information on the W3C and its services, see its website at [www.w3.org](http://www.w3.org).

By 1999, HTML had progressed to the fourth version of the language, **HTML 4.01**, which provided support for multimedia, online commerce, and interactive scripts running within the web page. However, there were still many incompatibilities in how HTML was implemented across different browsers and how HTML code was written by web developers. The W3C sought to take control of what had been a haphazard process and enforce a stricter set of standards in a different version of the language called **XHTML (Extensible Hypertext Markup Language)**. By 2002, the W3C had released the specifications for XHTML 1.1. But XHTML 1.1 was intended to be only a minor upgrade on the way to XHTML 2.0, which would correct many of the deficiencies found in HTML 4.01 and become the future language of the web. One problem was that XHTML 2.0 would not be backward compatible with HTML and, as a result, older websites could not be easily brought into the new standard.

Web designers rebelled at this development and, in response, the **Web Hypertext Application Technology Working Group (WHATWG)** was formed in 2004 with the mission to develop a rival version to XHTML 2.0, called **HTML5**. Unlike XHTML 2.0, HTML5 would be compatible with earlier versions of HTML and would not apply the same strict standards that XHTML demanded. For several years, it was unclear which specification would win out; but by 2006, work on XHTML 2.0 had completely stalled

and the W3C issued a new charter for WHATWG to develop HTML5 as the de facto standard for the next generation of HTML. Thus today, HTML5 is the current version of the HTML language and it is supported by all current browsers and devices. You can learn more about WHATWG and its current projects at [www.whatwg.org](http://www.whatwg.org).

**TIP**

You can find out which browsers and browser versions support the features of HTML5 by going to the website [caniuse.com](http://caniuse.com).

As HTML has evolved, features and code found in earlier versions of the language are often **deprecated**, or phased out, and while deprecated features might not be part of HTML5, that doesn't mean that you won't encounter them in your work—indeed, if you are maintaining older websites, you will often need to interpret code from earlier versions of HTML. Moreover, there are still many older browsers and devices in active use that do not support HTML5. Thus, a major challenge for website designers is writing code that takes advantage of HTML5 but is still accessible to older technology.

Figure 1-1 summarizes some of the different versions of HTML that have been implemented over the years. You can read detailed specifications for these versions at the W3C website.

**Figure 1-1****HTML version history**

Version	Date	Description
HTML 1.0	1989	The first public version of HTML
HTML 2.0	1995	HTML version that added interactive elements including web forms
HTML 3.2	1997	HTML version that provided additional support for web tables and expanded the options for interactive form elements and a scripting language
HTML 4.01	1999	HTML version that added support for style sheets to give web designers greater control over page layout and appearance, and provided support for multimedia elements such as audio and video
XHTML 1.0	2001	A reformulation of HTML 4.01 using the XML markup language in order to provide enforceable standards for HTML content and to allow HTML to interact with other XML languages
XHTML 2.0	discontinued in 2009	The follow-up version to XHTML 1.1 designed to fix some of the problems inherent in HTML 4.01 syntax
HTML 5.0	2012	The current HTML version providing support for mobile design, semantic page elements, column layout, form validation, offline storage, and enhanced multimedia

© 2016 Cengage Learning

This book focuses on HTML5, but you will also review some of the specifications for HTML 4.01 and XHTML 1.1. Note that in the figures that follow, code that was introduced starting with HTML5 will be identified with the label [**HTML5**].

## Tools for Working with HTML

Because HTML documents are simple text files, the first tool you will need is a text editor. You can use a basic text editor such as Windows Notepad or TextEdit for the Macintosh, but it is highly recommended that you use one of the many inexpensive editors that provide built-in support for HTML. Some of the more popular HTML editors are Notepad++ ([notepad-plus-plus.org](http://notepad-plus-plus.org)), UltraEdit ([www.ultraedit.com](http://www.ultraedit.com)), CoffeeCup ([www.coffeecup.com](http://www.coffeecup.com)), BBEdit ([www.barebones.com](http://www.barebones.com)) and ConTEXT ([www.contexteditor.org](http://www.contexteditor.org)). These editors include such features as syntax checking to weed out errors, automatic insertion of HTML code, and predesigned templates with the initial code already prepared for you.

These enhanced editors are a good way to start learning HTML and they will be all you need for most basic projects, but professional web developers working on large websites will quickly gravitate toward using a web **IDE (Integrated Development Environment)**, which is a software package providing comprehensive coverage of all phases of the development process from writing HTML code to creating scripts for programs running on web servers. Some of the popular IDEs for web development include Adobe Dreamweaver ([www.adobe.com](http://www.adobe.com)), Aptana Studio ([www.aptana.com](http://www.aptana.com)), NetBeans IDE ([netbeans.org](http://netbeans.org)) and Komodo IDE ([komodoide.com](http://komodoide.com)). Web IDEs can be very expensive, but most software companies will provide a free evaluation period for you to test their product to see if it meets your needs.

## Testing your Code

### TIP

You can analyze each browser for its compatibility with HTML5 at the website [www.html5test.com](http://www.html5test.com).

Once you've written your code, you can test whether your HTML code employs proper syntax and structure by validating it at the W3C validation website ([validator.w3.org](http://validator.w3.org)). **Validators**, like the one available through the W3C website, are programs that test code to ensure that it contains no syntax errors. The W3C validator will highlight all of the syntax errors in your document with suggestions about how to fix those errors.

Finally, you'll need to test it to ensure that your content is rendered correctly. You should test your code under a variety of screen resolutions, on several different browsers and, if possible, on different versions of the same browser because users are not always quick to upgrade their browsers. What may look good on a widescreen monitor might look horrible on a mobile phone. At a minimum you should test your website using the following popular browsers: Google Chrome, Internet Explorer, Apple Safari, Mozilla Firefox, and Opera.

It is not always possible to load multiple versions of the same browser on one computer, so, in order to test a website against multiple browser versions, professional designers will upload their code to online testing services that report on the website's compatibility across a wide range of browsers, screen resolutions, and devices, including both desktop and mobile devices. Among the popular testing services are BrowserStack ([www.browserstack.com](http://www.browserstack.com)), CrossBrowserTesting ([www.crossbrowsertesting.com](http://www.crossbrowsertesting.com)), and Browsera ([www.browsersa.com](http://www.browsersa.com)). Most of these sites charge a monthly connection fee with a limited number of testing minutes, so you should not upload your code until you are past the initial stages of development.

## Supporting the Mobile Web

Currently, the most important factor impacting website design is the increased use of mobile devices to access the Internet. By the end of 2014, the number of mobile Internet users exceeded the number of users accessing the web through laptop or desktop devices. The increased reliance on mobile devices means that web designers must be careful to tailor their websites to accommodate both the desktop and mobile experience. You'll explore the challenge of designing for the mobile web in more detail in Tutorial 5.

## Exploring an HTML Document

Now that you have reviewed the history of the web and some of the challenges in developing your own website, you will look at the code of an actual HTML file. To get you started, Sajja Adulet has provided you with the `ct_start.html` file containing the code for the initial page users see when they access the Curbside Thai website. Open Sajja's file now.



**TIP**

All HTML files have the file extension .html or .htm.

**To open the ct\_start.html file:**

1. Use the editor of your choice to open the **ct\_start.html** file from the html01 ► tutorial folder.

Figure 1-2 shows the complete contents of the file as viewed in the Notepad++ editor.

**Figure 1-2****Elements and attributes from an HTML document**

The screenshot shows the following HTML code with annotations:

```

<!DOCTYPE html>
<html>

<head>
  <title>Curbside Thai</title>

  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link href="ct_base.css" rel="stylesheet" type="text/css" />
  <link href="ct_layout1.css" rel="stylesheet" type="text/css" />
</head>

<body>
  <header>
    
  </header>
  <nav>
    <a href="ct_about.html"></a>
    <a href="ct_locations.html"></a>
    <a href="ct_menu.html"></a>
    <a href="ct_reviews.html"></a>
    <a href="ct_catering.html"></a>
    <a href="ct_contact.html"></a>
  </nav>
  <footer>
    Curbside Thai &#8226; 411 Belde Drive, Charlotte NC 28201 &#8226; 704-555-1151
  </footer>
</body>
</html>

```

Annotations in the image:

- two-sided tag enclosing element content:** Points to the `<html>` and `</html>` tags.
- empty elements, which do not contain content:** Points to the `<meta>` and `<link>` tags.
- an element attribute:** Points to the `alt="Curbside Thai"` attribute in the `<img>` tag.
- several elements nested within another element:** Points to the `<nav>` and `<header>` elements nested within the `<body>` element.

**Trouble?** Depending on your editor and its configuration, the text style applied to your code might not match that shown in Figure 1-2. This is not a problem. Because HTML documents are simple text files, any text styles are a feature of the editor and have no impact on how the document is rendered by the browser.

2. Scroll through the document to become familiar with its content but do not make any changes to the text.

## The Document Type Declaration

The first line in an HTML file is the document type declaration or doctype, which is a processing instruction indicating the markup language used in the document. The browser uses the document type declaration to know which standard to use to display the content. For HTML5, the doctype is entered as

```
<!DOCTYPE html>
```

You might also see the doctype entered in lowercase letters as

```
<!doctype html>
```

Both are accepted by all browsers. Older versions of HTML had more complicated doctypes. For example, the doctype for HTML 4.01 is the rather foreboding

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

You might even come across older HTML files that do not have a doctype. Because early versions of HTML did not require a doctype, many browsers interpret the absence of the doctype as a signal that the page should be rendered in **quirks mode**, based on styles and practices from the 1990s and early 2000s. When the doctype is present, browsers will render the page in **standards mode**, employing the most current specifications of HTML. The difference between quirks mode and standards mode can mean the difference between a nicely laid-out page and a confusing mess, so, as a result, you should always put your HTML5 file in standards mode by including the doctype.

## Introducing Element Tags

The fundamental building block in every HTML document is the **element tag**, which marks an element in the document. A **starting tag** indicates the beginning of that element, while an **ending tag** indicates the ending. The general syntax of a two-sided element tag is

```
<element>content</element>
```

where *element* is the name of the element, *content* is the element's content, `<element>` is the starting tag, and `</element>` is the ending tag. For example, the following code marks a paragraph element:

```
<p>Welcome to Curbside Thai.</p>
```

Here the `<p></p>` tags are the starting and ending HTML tags that indicate the presence of a paragraph and the text *Welcome to Curbside Thai.* comprises the paragraph text.

Not every element tag encloses document content. **Empty elements** are elements that are either nontextual (such as images) or contain directives to the browser about how the page should be treated. An empty element is entered using one of the following forms of the **one-sided element tag**:

```
<element />
```

or

```
<element>
```

For example, the following `br` element, which is used to indicate the presence of a line break in the text, is entered with the one-sided tag:

```
<br />
```

Note that, while this code could also be entered as `<br>`, the ending slash `/>` form is the required form in XHTML documents as well as other markup languages. While HTML5 allows for either form, it's a good idea to get accustomed to using the ending slash `/>` form if you intend to work with other markup languages in the future. We'll follow the `/>` convention in the code in this book.

Elements can contain other elements, which are called **nested elements**. For example, in the following code, the `em` element (used to mark emphasized text) is nested within the paragraph element by placing the `em` markup tag completely within the `p` markup tag.

### Proper syntax:

```
<p>Welcome to <em>Curbside Thai</em>.</p>
```

Note that when nesting one element inside of another, the entire code of the inner element must be contained within the outer element, including opening and closing tags. Thus, it would not be correct syntax to place the closing tag for the `em` element outside of the `p` element as in the following code:

### Improper syntax:

```
<p>Welcome to <em>Curbside Thai</p>.</em>
```

Now that you've examined the basics of tags, you'll look at how they're used within an HTML file.

## The Element Hierarchy

The entire structure of an HTML document can be thought of as a set of nested elements in a hierarchical tree. At the top of the tree is the `html` element, which marks the entire document. Within the `html` element is the `head` element used to mark information about the document itself and the `body` element used to mark the content that will appear in the web page. Thus, the general structure of an HTML file, like the one shown in Figure 1-2, is

```
<!DOCTYPE html>
<html>
  <head>
    head content
  </head>

  <body>
    body content
  </body>
</html>
```

where *head content* and *body content* are nested elements that mark the content of the document head and body. Note that the `body` element is always placed after the `head` element.

### Creating the Basic Structure of an HTML File

- To create the basic structure of an HTML file, enter the tags

```
<!DOCTYPE html>
<html>
  <head>
    head content
  </head>

  <body>
    body content
  </body>
</html>
```

where *head*, *content*, and *body content* contain nested elements that mark the content of the head and body sections.

## Introducing Element Attributes

### TIP

Attributes can be listed in any order but they must come after the element name and be separated from each other by a blank space; each attribute value must be enclosed within single or double quotation marks.

Elements will often contain one or more **element attributes**. Each attribute provides additional information to the browser about the purpose of the element or how the element should be handled by the browser. The general syntax of an element attribute within a two-sided tag is

```
<element attr1="value1" attr2="value2" ...>
    content
</element>
```

Or, for a one-sided tag

```
<element attr1="value1" attr2="value2" ... />
```

where *attr1*, *attr2*, and so forth are attributes associated with *element* and *value1*, *value2*, and so forth are the corresponding attribute values. For example, the following code adds the `id` attribute with the value "intro" to the `<p>` tag in order to identify the paragraph as an introductory paragraph.

```
<p id="intro">Welcome to Curbside Thai.</p>
```

HTML editors will often color-code attributes and their values. The attributes in Figure 1-2 are rendered in a blue font while the corresponding attribute values are rendered in magenta.

Each element has its own set of attributes but, in addition to these element-specific attributes, there is a core set of attributes that can be applied to almost every HTML element. Figure 1-3 lists some of the most commonly used core attributes; others are listed in Appendix B.

Figure 1-3

Commonly used core HTML attributes

Attribute	Description
<code>class="text"</code>	Defines the general classification of the element
<code>dir="ltr rtl auto"</code>	Defines the text direction of the element content as left-to-right, right-to-left, or determined by the browser
<code>hidden</code>	Indicates that the element should be hidden or is no longer relevant <b>[HTML5]</b>
<code>id="text"</code>	Provides a unique identifier for the element
<code>lang="text"</code>	Specifies the language of the element content
<code>style="definition"</code>	Defines the style or appearance of the element content
<code>tabindex="integer"</code>	Specifies the tab order of the element (when the tab button is used to navigate the page)
<code>title="text"</code>	Assigns a title to the element content

© 2016 Cengage Learning

Some attributes do not require a value, so, as a result, HTML supports **attribute minimization** in which no value is shown in the document. For example, the `hidden` attribute used in the following code does not require a value, its mere presence indicates that the marked paragraph should be hidden in the rendered page.

```
<p hidden>Placeholder Text</p>
```

Attribute minimization is another example of how HTML5 differs from other markup languages such as XHTML in which minimization is not allowed and all attributes must have attribute values.

### Adding an Attribute to an Element

- To add an attribute to an element, enter

```
<element attr1="value1" attr2="value2" ...>
  content
</element>
```

where *attr1*, *attr2*, and so forth are HTML attributes associated with *element* and *value1*, *value2*, and so forth are the corresponding attribute values.

## Handling White Space

Because an HTML file is a text file, it is composed only of text characters and white-space characters. A **white-space character** is any empty or blank character such as a space, tab, or line break. When the browser reads an HTML file, it ignores the presence of white-space characters between element tags and makes no distinction between spaces, tabs, or line breaks. Thus, a browser will treat the following two pieces of code in exactly the same way:

```
<p>Welcome to <em>Curbside Thai</em>.</p>
```

and

```
<p>
  Welcome to <em>Curbside Thai</em>.
</p>
```

The browser will also collapse consecutive occurrences of white-space characters into a single occurrence. This means that the text of the paragraph in the following code is still treated as “Welcome to Curbside Thai” because the extra white spaces between “Curbside” and “Thai” are ignored by the browser.

```
<p>
  Welcome to <em>Curbside      Thai</em>.
</p>
```

The bottom line is that it doesn’t matter how you lay out your HTML code because the browser is only interested in the text content and not how that text is entered. This means you can make your file easier to read by indenting lines and by adding extra white-space characters to separate one code block from another. However, this also means that any formatting you do for the page text to make the code more readable, such as tabs or extra white spaces, is *not* transferred to the web page.

## Viewing an HTML File in a Browser

The structure of the HTML file shown in Figure 1-2 should now be a little clearer, even if you don’t yet know how to interpret the meaning and purpose of each of element and attribute. To see what this page looks like, open it within a web browser.

### To open the `ct_start.html` file in a web browser:

1. Open your web browser. You do not need to be connected to the Internet to view local files stored on your computer.