

Laura Lemay  
Rafe Colburn  
Jennifer Kyrnin

SEVENTH  
EDITION

Covers  
**HTML5, CSS3,**  
and **jQuery**

Sams **Teach Yourself**

# HTML, CSS & JavaScript Web Publishing

in **One Hour** a Day



SAMS

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



Laura Lemay  
Rafe Coburn  
Jennifer Kyrnin

Sams **Teach Yourself**  
**HTML, CSS**  
**& JavaScript**  
Web Publishing

in **One Hour a Day**  
Seventh Edition

**SAMS**

800 East 96th Street, Indianapolis, Indiana 46240

# **Sams Teach Yourself HTML, CSS & JavaScript Web Publishing in One Hour a Day, Seventh Edition**

Copyright © 2016 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-672-33623-2

ISBN-10: 0-672-33623-5

Library of Congress Control Number: 2015918052

Printed in the United States of America

First Printing December 2015

## **Trademarks**

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

## **Warning and Disclaimer**

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

## **Special Sales**

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at [corpsales@pearsoned.com](mailto:corpsales@pearsoned.com) or (800) 382-3419.

For government sales inquiries, please contact [governmentsales@pearsoned.com](mailto:governmentsales@pearsoned.com).

For questions about sales outside the U.S., please contact [international@pearsoned.com](mailto:international@pearsoned.com).

### **Acquisitions Editor**

Mark Taber

### **Managing Editor**

Sandra Schroeder

### **Senior Project Editor**

Tonya Simpson

### **Copy Editor**

Keith Cline

### **Indexer**

Tim Wright

### **Proofreader**

Gill Editorial Services

### **Editorial Assistant**

Vanessa Evans

### **Cover Designer**

Mark Shirar

### **Compositor**

Bronkella Publishing

# Contents at a Glance

Introduction	1
<b>PART I: Getting Started</b>	
<b>1</b> What Is Web Publishing?	7
<b>2</b> Getting Your Tools in Order	23
<b>3</b> Introducing HTML and CSS	41
<b>PART II: Creating Web Pages</b>	
<b>4</b> Learning the Basics of HTML	57
<b>5</b> Organizing Information with Lists	71
<b>6</b> Working with Links	89
<b>PART III: Doing More with HTML and CSS</b>	
<b>7</b> Formatting Text with HTML and CSS	121
<b>8</b> Using CSS to Style a Site	159
<b>9</b> Using Images on Your Web Pages	197
<b>10</b> Building Tables	241
<b>11</b> Using CSS to Position Elements on the Page	287
<b>12</b> Designing Forms	313
<b>13</b> Structuring a Page with HTML5	371
<b>14</b> Integrating Multimedia: Video and Sound	385
<b>15</b> Advanced CSS: Page Layout in CSS	421
<b>16</b> Using Responsive Web Design	443
<b>PART IV: Using JavaScript and jQuery</b>	
<b>17</b> Introducing JavaScript	471
<b>18</b> Using jQuery	499
<b>19</b> Using JavaScript in Your Pages	527
<b>20</b> Working with Frames and Linked Windows	555
<b>PART V: Designing for Everyone</b>	
<b>21</b> Designing for the Mobile Web	575
<b>22</b> Designing for User Experience	605
<b>PART VI: Going Live on the Web</b>	
<b>23</b> How to Publish Your Site	625
<b>24</b> Taking Advantage of the Server	651
<b>25</b> Search Engines and SEO	687
Index	703

# Table of Contents

Introduction	1
--------------	---

## **PART I: Getting Started**

<b>LESSON 1: What Is Web Publishing?</b>	<b>7</b>
--	----------

Thinking Like a Web Publisher	8
The Web Is a Hypertext Information System	8
The Web Is Cross-Platform	9
The Web Is Distributed	9
The Web Is Dynamic	10
The Web Is Interactive	12
Web Browsers	14
What the Browser Does	14
An Overview of Some Popular Browsers	15
Web Servers	18
Uniform Resource Locators	18
Defining Web Publishing Broadly	19
Summary	20
Workshop	20
Q&A	20
Quiz	21
Quiz Answers	21
Exercises	21

<b>LESSON 2: Getting Your Tools in Order</b>	<b>23</b>
--	-----------

Anatomy of a Website	24
Setting Up Your Computer for Web Publishing	26
Text Editors	26
A Web Browser	28
Using the Google Chrome Developer Tools	29
What Do You Want to Do on the Web?	33
Wireframing Your Website	33
What's Wireframing, and Why Do I Need It?	34
Hints for Wireframing	35

Web Hosting .....	36
Using a Content-Management Application .....	36
Setting Up Your Own Web Hosting .....	37
Summary .....	38
Workshop .....	39
Q&A .....	39
Quiz .....	39
Quiz Answers .....	40
Exercises .....	40

### **LESSON 3: Introducing HTML and CSS** **41**

What HTML Is (And What It Isn't) .....	42
HTML Describes the Structure of a Page .....	42
HTML Does Not Describe Page Layout .....	43
Why It Works This Way .....	44
How Markup Works .....	45
What HTML Files Look Like .....	45
Text Formatting and HTML .....	49
HTML Attributes .....	50
Using the <code>style</code> Attribute .....	50
Including Styles in Tags .....	51
A Short History of HTML Standards .....	52
XHTML .....	52
The Current and Evolving Standard: HTML5 .....	53
Summary .....	54
Workshop .....	54
Q&A .....	54
Quiz .....	54
Quiz Answers .....	55
Exercises .....	55

## **PART II: Creating Web Pages**

### **LESSON 4: Learning the Basics of HTML** **57**

Structuring Your HTML .....	58
The <code>&lt;html&gt;</code> Tag .....	58
The <code>&lt;head&gt;</code> Tag .....	59
The <code>&lt;body&gt;</code> Tag .....	59

The Title .....	60
Headings .....	62
Paragraphs .....	64
Comments .....	65
Summary .....	68
Workshop .....	68
Q&A .....	68
Quiz .....	69
Quiz Answers .....	69
Exercises .....	70

**LESSON 5: Organizing Information with Lists** **71**

Lists: An Overview .....	72
Numbered Lists .....	73
Customizing Ordered Lists .....	74
Unordered Lists .....	78
Customizing Unordered Lists .....	78
Definition Lists .....	81
Nesting Lists .....	82
Other Uses for Lists .....	84
Summary .....	85
Workshop .....	87
Q&A .....	87
Quiz .....	88
Quiz Answers .....	88
Exercises .....	88

**LESSON 6: Working with Links** **89**

Creating Links .....	90
The Link Tag: <a> .....	90
Linking Local Pages Using Relative and Absolute Pathnames .....	95
Absolute Pathnames .....	96
Should You Use Relative or Absolute Pathnames? .....	97
Links to Other Documents on the Web .....	98
Linking to Specific Places Within Documents .....	104
Creating Links and Anchors .....	104
The name Attribute of the <a> Tag .....	105
Linking to Elements in the Same Document .....	110

Anatomy of a URL .....	110
Parts of URLs .....	110
Special Characters in URLs .....	112
The <code>rel</code> Attribute .....	113
Kinds of URLs .....	113
HTTP .....	113
Anonymous FTP .....	114
Non-Anonymous FTP .....	115
Mailto .....	115
File .....	116
Summary .....	117
Workshop .....	117
Q&A .....	118
Quiz .....	119
Quiz Answers .....	120
Exercises .....	120

## **PART III: Doing More with HTML and CSS**

<b>LESSON 7: Formatting Text with HTML and CSS</b> .....	<b>121</b>
Character-Level Elements .....	122
Semantic HTML Tags .....	122
Changes to Physical Style Tags in HTML5 .....	124
Character Formatting Using CSS .....	125
The Text Decoration Property .....	125
Font Properties .....	126
Preformatted Text .....	128
Horizontal Rules (or Thematic Breaks) .....	130
Attributes of the <code>&lt;hr&gt;</code> Tag .....	132
Line Break .....	133
Addresses .....	134
Quotations .....	135
Special Characters .....	138
Character Encoding .....	139
Character Entities for Special Characters .....	140
Character Entities for Reserved Characters .....	141



Fonts and Font Sizes .....	142
Summary .....	154
Workshop .....	156
Q&A .....	156
Quiz .....	157
Quiz Answers .....	157
Exercises .....	158
<b>LESSON 8: Using CSS to Style a Site</b> .....	<b>159</b>
Including Style Sheets in a Page .....	160
Creating Page-Level Styles .....	160
Creating Sitewide Style Sheets .....	161
Selectors .....	162
Contextual Selectors .....	162
Classes and IDs .....	163
What Cascading Means .....	165
Units of Measure .....	166
Specifying Colors .....	168
Editing Styles with Developer Tools .....	171
Using Color .....	172
Links .....	173
The Box Model .....	174
Borders .....	175
Margins and Padding .....	177
Controlling Size and Element Display .....	182
Float .....	185
More Selectors .....	189
Pseudo-Classes .....	190
Attribute Selectors .....	193
The <body> Tag .....	194
Summary .....	194
Workshop .....	195
Q&A .....	195
Quiz .....	196
Quiz Answers .....	196
Exercises .....	196

<b>LESSON 9: Using Images on Your Web Pages</b>	<b>197</b>
Images on the Web .....	198
Image Formats .....	198
GIF .....	198
JPEG .....	199
PNG .....	199
SVG .....	200
Inline Images in HTML: The <code>&lt;img&gt;</code> Tag .....	200
Adding Alternative Text to Images .....	201
Images and Text .....	205
Text and Image Alignment .....	207
Wrapping Text Next to Images .....	210
Adjusting the Space Around Images .....	213
Images and Links .....	214
Other Neat Tricks with Images .....	219
Image Dimensions and Scaling .....	219
Image Backgrounds .....	220
Using Images as Bullets .....	224
What Is an Imagemap? .....	226
Getting an Image .....	226
Determining Your Coordinates .....	227
The <code>&lt;map&gt;</code> and <code>&lt;area&gt;</code> Tags .....	230
The <code>usemap</code> Attribute .....	231
Image Etiquette .....	237
Summary .....	237
Workshop .....	238
Q&A .....	238
Quiz .....	239
Quiz Answers .....	239
Exercises .....	240
<b>LESSON 10: Building Tables</b>	<b>241</b>
Creating Tables .....	242
Table Parts .....	242
The <code>&lt;table&gt;</code> Element .....	243
Summarizing the Table .....	243
Rows and Cells .....	244

Empty Cells .....	246
Captions .....	247
Sizing Tables, Borders, and Cells .....	251
Setting Table Widths .....	252
Changing Table Borders .....	253
Cell Padding .....	256
Cell Spacing .....	257
Column Widths .....	259
Table and Cell Color .....	260
Aligning Your Table Content .....	262
Table Alignment .....	263
Cell and Caption Alignment .....	264
Spanning Multiple Rows or Columns .....	266
More Advanced Table Enhancements .....	276
Grouping and Aligning Columns .....	276
Grouping and Aligning Rows .....	279
How Tables Are Used .....	282
Summary .....	282
Workshop .....	284
Q&A .....	285
Quiz .....	285
Quiz Answers .....	285
Exercises .....	286
<b>LESSON 11: Using CSS to Position Elements on the Page</b> .....	<b>287</b>
Positioning Schemes .....	288
Relative Positioning .....	288
Absolute Positioning .....	293
Positioning Properties .....	293
Positioning Properties and Height and Width .....	294
Nesting Absolutely Positioned Elements .....	295
Dynamic Overlays .....	297
Fixed Positioning .....	301
Controlling Stacking .....	303
Creating Drop-Down Menus .....	306
Summary .....	311

Workshop .....	311
Q&A .....	311
Quiz .....	312
Quiz Answers .....	312
Exercises .....	312
<b>LESSON 12: Designing Forms</b> .....	<b>313</b>
Understanding Form and Function .....	314
Using the <code>&lt;form&gt;</code> Tag .....	319
Using the <code>&lt;label&gt;</code> Tag .....	323
Creating Form Controls with the <code>&lt;input&gt;</code> Tag .....	324
Creating Text Controls .....	325
Adding Options to Text Fields with <code>datalist</code> .....	327
Using the New HTML5 Controls .....	328
Creating Password Controls .....	332
Creating Submit Buttons .....	333
Creating Reset Buttons .....	334
Creating Check Box Controls .....	334
Creating Radio Buttons .....	336
Using Images as Submit Buttons .....	337
Creating Generic Buttons .....	337
Hidden Form Fields .....	338
The File Upload Control .....	339
Using Other Form Controls .....	340
Using the <code>button</code> Element .....	340
Creating Large Text-Entry Fields with <code>textarea</code> .....	341
Creating Menus with <code>select</code> and <code>option</code> .....	342
Grouping Controls with <code>fieldset</code> and <code>legend</code> .....	350
Changing the Default Form Navigation .....	351
Using Access Keys .....	352
Creating <code>disabled</code> and <code>readonly</code> Controls .....	352
Displaying Updates with <code>progress</code> and <code>meter</code> .....	354
Applying Cascading Style Sheet Properties to Form Elements .....	359
Planning Your Forms .....	365
Summary .....	365

Workshop .....	367
Q&A .....	367
Quiz .....	368
Quiz Answers .....	368
Exercises .....	369
<b>LESSON 13: Structuring a Page with HTML5</b>	<b>371</b>
A Short History of HTML Page Layout .....	372
Laying Out a Page in HTML5 .....	373
HTML5 Structural Tags .....	375
Sections .....	375
Header .....	376
Footer .....	376
Navigation .....	377
Articles .....	378
Asides .....	379
The Page Outline .....	379
Elements with Their Own Outlines .....	381
Using HTML5 Structural Elements .....	381
Polyfill Scripts .....	381
Summary .....	381
Workshop .....	382
Q&A .....	382
Quiz .....	383
Quiz Answers .....	383
Exercise .....	383
<b>LESSON 14: Integrating Multimedia: Video and Sound</b>	<b>385</b>
Embedding Video the Simple Way .....	386
Advantages and Disadvantages of Hosting Videos on External Sites .....	387
Uploading Videos to YouTube .....	388
Customizing the Video Player .....	389
Other Services .....	390
Hosting Your Own Video .....	391
Video and Container Formats .....	392
Converting Video to H.264 .....	393

Embedding Video Using <video>	396
The <video> Tag	397
Using the <source> Element	399
Embedding Flash Using the <object> Tag	400
Alternative Content for the <object> Tag	403
The <embed> Tag	404
Embedding Flash Movies Using SWFObject	406
Flash Video Players	408
JW Player	408
Using Flowplayer	410
Using the <object> Tag with the <video> Tag	412
Embedding Audio in Your Pages	413
The <audio> Tag	413
Flash Audio Players	414
Summary	416
Workshop	417
Q&A	417
Quiz	418
Quiz Answers	418
Exercises	419
<b>LESSON 15: Advanced CSS: Page Layout in CSS</b>	<b>421</b>
Laying Out the Page	422
The Problems with Layout Tables	422
Writing HTML with Structure	423
Writing a Layout Style Sheet	426
The Floated Columns Layout Technique	433
The Role of CSS in Web Design	435
Style Sheet Organization	436
Site-Wide Style Sheets	439
Summary	439
Workshop	440
Q&A	440
Quiz	441
Quiz Answers	441
Exercises	441

<b>LESSON 16: Using Responsive Web Design</b>	<b>443</b>
What Is Responsive Web Design?	444
History of Responsive Web Design	444
Why RWD Is Important	445
RWD Is More Than Just Changing the Number of Columns	445
Mobile Devices Should Come First	445
Mobile First	446
Affecting the Viewport	447
Planning a Responsive Website	449
Check Your Analytics	449
Try the Site with Your Own Phone	449
Decide What Content Is Critical	449
Writing Media Queries	450
Media Types	450
Media Features	451
Breakpoints	452
Building a Style Sheet with Media Queries	454
Understanding the Mechanics of RWD	454
Adjusting the Layout	454
Making Images and Videos Responsive	460
Building Responsive Tables	463
Responsive Web Design Best Practices	467
Give Everyone the Best Experience	467
Use the Best Breakpoints for Your Website, Not for Devices	468
Be Flexible But Think Small	468
Summary	468
Workshop	469
Q&A	469
Quiz	469
Quiz Answers	469
Exercises	470

**PART IV: Using JavaScript and jQuery**

<b>LESSON 17: Introducing JavaScript</b>	<b>471</b>
Why Would You Want to Use JavaScript?	472
Ease of Use	473
Improving Performance	473
Integration with the Browser	473
The <code>&lt;script&gt;</code> Tag	473
The Structure of a JavaScript Script	474
The <code>src</code> Attribute	474
JavaScript and the Chrome Development Tools	475
The JavaScript Language	476
Operators and Expressions	479
Variables	480
Control Structures	482
Functions	486
Data Types	488
Arrays	489
Objects	489
The JavaScript Environment	491
Events	491
Summary	495
Workshop	495
Q&A	495
Quiz	496
Quiz Answers	496
Exercises	497
<b>LESSON 18: Using jQuery</b>	<b>499</b>
What Are JavaScript Libraries?	500
Getting Started with jQuery	500
Your First jQuery Script	501
Selecting Elements from the Document	503
Binding Events	504
Modifying Styles on the Page	505
Hiding and Showing Elements	505
Retrieving and Changing Style Sheet Properties	507



Modifying Content on the Page .....	508
Manipulating Classes .....	508
Manipulating Form Values .....	511
Manipulating Attributes Directly .....	513
Adding and Removing Content .....	514
Special Effects .....	518
AJAX and jQuery .....	520
Using AJAX to Load External Data .....	521
Summary .....	525
Workshop .....	525
Q&A .....	525
Quiz .....	526
Quiz Answers .....	526
Exercises .....	526
<b>LESSON 19: Using JavaScript in Your Pages</b> .....	<b>527</b>
Validating Forms with JavaScript .....	528
Hiding and Showing Content .....	538
The Same Code with jQuery .....	546
Adding New Content to a Page .....	546
Summary .....	551
Workshop .....	551
Q&A .....	551
Quiz .....	552
Quiz Answers .....	552
Exercises .....	553
<b>LESSON 20: Working with Frames and Linked Windows</b> .....	<b>555</b>
What Are Frames? .....	556
Why Were Frames Removed from HTML5? .....	556
What About Iframes? .....	556
Working with Linked Windows .....	557
Browsing Context Keywords .....	557
The <base> Tag .....	562
Inline Frames .....	563
Opening Linked Windows with JavaScript .....	567
Summary .....	571

Workshop .....	572
Q&A .....	572
Quiz .....	572
Quiz Answers .....	572
Exercises .....	573

## **PART V: Designing for Everyone**

### **LESSON 21: Designing for the Mobile Web** **575**

People Browse Differently on Mobile Phones .....	576
Standards Compliance and the Mobile Web .....	577
Progressive Enhancement .....	577
Validating Your Pages .....	578
Writing for the Mobile Web .....	581
Write Clearly and Be Brief .....	582
Organize Your Pages for Quick Scanning .....	582
Make Each Page Stand on Its Own .....	583
Be Careful with Emphasis .....	583
Don't Use Browser-Specific Terminology .....	584
Spell Check and Proofread Your Pages .....	584
Design and Page Layout .....	585
Use Headings as Headings .....	585
Group Related Information Visually .....	586
Use a Consistent Layout .....	586
Using Links .....	587
Mobile Users Tap; They Don't Click .....	587
Use Link Menus with Descriptive Text .....	587
Use Links in Text .....	588
Avoid the "Here" Syndrome .....	588
To Link or Not to Link .....	589
Using Images and Multimedia .....	591
Don't Overuse Images .....	591
Keep Images Small .....	591
Watch Out for Assumptions About Your Visitors' Hardware .....	592
Don't Make Your Videos Annoying .....	592
Avoid Flash .....	593

Making the Most of CSS and JavaScript .....	593
Put Your CSS and JavaScript in External Files .....	593
Location Matters .....	594
Shrink Your CSS and JavaScript .....	594
Take Advantage of Mobile Features .....	595
Geolocation .....	595
Make Phone Calls .....	596
SMS .....	597
Other Good Habits and Hints for Mobile Web Design .....	597
Link Back to Home .....	597
Don't Split Topics Across Pages .....	598
Sign Your Pages .....	598
One Final Secret to Mobile Web Design .....	599
Summary .....	599
Workshop .....	601
Q&A .....	601
Quiz .....	602
Quiz Answers .....	602
Exercises .....	603

## **LESSON 22: Designing for User Experience** **605**

Considering User Experience Level .....	606
Add a Search Engine .....	606
Use Concise, Sensible URLs .....	606
Navigation Provides Context .....	608
Are Your Users Tourists or Regulars? .....	609
Determining User Preferences .....	610
What Is Accessibility? .....	611
Common Myths Regarding Accessibility .....	611
Section 508 .....	612
Alternative Browsers .....	613
Writing Accessible HTML .....	613
Tables .....	614
Links .....	615
Images and Multimedia .....	616
Designing for Accessibility .....	617
Using Color .....	617
Fonts .....	618

Take Advantage of All HTML Tags .....	619
Frames .....	619
Forms .....	619
Validating Your Sites for Accessibility .....	620
Summary .....	621
Workshop .....	621
Q&A .....	622
Quiz .....	623
Quiz Answers .....	623
Exercises .....	624

## **PART VI: Going Live on the Web**

<b>LESSON 23: How to Publish Your Site</b> .....	<b>625</b>
What Does a Web Server Do? .....	626
Other Things Web Servers Do .....	626
How to Find Web Hosting .....	627
Using a Web Server Provided by Your School or Work .....	627
Using a Commercial Web Host .....	628
Commercial Web Builders .....	629
Setting Up Your Own Server .....	629
Free Hosting .....	630
Organizing Your HTML Files for Publishing .....	630
Questions to Ask Your Webmaster .....	630
Keeping Your Files Organized with Directories .....	632
Having a Default Index File and Correct Filenames .....	632
Publishing Your Files .....	633
Moving Files Between Systems .....	633
Troubleshooting .....	636
I Can't Access the Server .....	636
I Can't Access Files .....	636
I Can't Access Images .....	636
My Links Don't Work .....	637
My Files Are Being Displayed Incorrectly .....	638
Promoting Your Web Pages .....	638
Getting Links from Other Sites .....	639
Content Marketing Through Guest Posting .....	639

Promoting Your Site Through Social Media .....	640
Creating a Facebook Page for Your Site .....	642
Site Indexes and Search Engines .....	643
Business Cards, Letterhead, Brochures, and Advertisements .....	644
Finding Out Who's Viewing Your Web Pages .....	644
Log Files .....	644
Google Analytics .....	645
Summary .....	648
Workshop .....	648
Q&A .....	649
Quiz .....	649
Quiz Answers .....	649
Exercises .....	650

## **LESSON 24: Taking Advantage of the Server** **651**

How PHP Works .....	652
Getting PHP to Run on Your Computer .....	653
The PHP Language .....	654
Comments .....	654
Variables .....	655
Arrays .....	656
Strings .....	659
Conditional Statements .....	661
PHP Conditional Operators .....	662
Loops .....	662
foreach Loops .....	663
for Loops .....	664
while and do...while Loops .....	664
Controlling Loop Execution .....	665
Built-In Functions .....	666
User-Defined Functions .....	666
Returning Values .....	667
Processing Forms .....	668
Handling Parameters with Multiple Values .....	669
Presenting the Form .....	674
Using PHP Includes .....	678
Choosing Which Include Function to Use .....	680

Expanding Your Knowledge of PHP .....	681
Database Connectivity .....	682
Regular Expressions .....	682
Sending Mail .....	682
Object-Oriented PHP .....	682
Cookies and Sessions .....	683
File Uploads .....	683
Other Application Platforms .....	683
Microsoft ASP.NET .....	683
Java EE .....	684
Ruby on Rails .....	684
Summary .....	684
Workshop .....	684
Q&A .....	685
Quiz .....	685
Quiz Answers .....	686
Exercises .....	686
<b>LESSON 25: Search Engines and SEO</b> .....	<b>687</b>
What Is SEO? .....	688
Why You Need SEO .....	688
What About Social Media? .....	688
You Can Do Your Own SEO .....	689
Why Don't Search Engines Find Sites Without SEO? .....	689
How Search Engines Work .....	689
Google .....	690
Microsoft Bing .....	691
Yahoo! .....	691
Don't Forget International Searches .....	691
SEO Techniques .....	692
Is Your Site "Friendly?" .....	692
Using Keywords and Keyword Research .....	693
Creating Content for Customers Is the Best SEO .....	694
Myths and Facts About SEO .....	695
Tools for Tracking and Managing SEO .....	696
Using Sitemaps .....	696
The robots.txt File .....	697

- Understanding Canonical Links ..... 698
- Redirecting Duplicate Content ..... 698
- Checking How Your Site Looks to Search Engines ..... 699
- Tracking Your SEO Efforts ..... 700
- Paying for Links ..... 700
- Summary ..... 701
- Workshop ..... 701
  - Q&A ..... 701
  - Quiz ..... 702
  - Quiz Answers ..... 702
  - Exercises ..... 702

**Index**

## About the Authors

**Rafe Colburn** is an author and web developer with more than 15 years of experience building websites. His other books include *Special Edition Using SQL* and *Sams Teach Yourself CGI in 24 Hours*. You can read his blog at <http://rc3.org> or find him on Twitter as @rafeco.

**Jennifer Kyrnin** is an author and web designer who has been working on the Internet since 1995. Her other books include *Sams Teach Yourself Bootstrap in 24 Hours*, *Sams Teach Yourself Responsive Web Design in 24 Hours*, and *Sams Teach Yourself HTML5 Mobile Application Development in 24 Hours*. She can be found at <http://htmljenn.com/> or on Twitter as @htmljenn.

**Laura Lemay** is one of the world's most popular authors on web development topics. She is the original author of *Sams Teach Yourself Web Publishing with HTML*, *Sams Teach Yourself Java in 21 Days*, and *Sams Teach Yourself Perl in 21 Days*.



## We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

*Please note that we cannot help you with technical problems related to the topic of this book.*

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: [feedback@sampublishing.com](mailto:feedback@sampublishing.com)

Mail: Sams Publishing  
ATTN: Reader Feedback  
800 East 96th Street  
Indianapolis, IN 46240 USA

## Reader Services

Register your copy of *Sams Teach Yourself HTML, CSS & JavaScript Web Publishing in One Hour a Day* (ISBN 978-0-672-33623-2) at [informit.com/register](http://informit.com/register) for convenient access to downloads, updates, and corrections as they become available.

# Introduction

Over the past decade, the Web has become completely integrated into the fabric of society. Most businesses have websites, and it's unusual to see a commercial on television that doesn't display a URL. The simple fact that most people know what a URL *is* speaks volumes. People who didn't know what the Internet was several years ago are now reconnecting with their high school friends on Facebook.

Perhaps the greatest thing about the Web is that you don't have to be a big company to publish things on it. The only things you need to create your own website are a computer with access to the Internet and the willingness to learn. Obviously, the reason you're reading this is that you have an interest in web publishing. Perhaps you need to learn about it for work, or you're looking for a new means of self-expression, or you want to post baby pictures on the Web so that your relatives all over the country can stay up-to-date. The question is, how do you get started?

There's more than enough information on the Web about how to publish websites like a seasoned professional. There are tutorials, reference sites, tons of examples, and free tools to make it easier to publish on the Web. However, the advantage of reading this book instead is that all the information you need to build websites is organized in one place and presented in an orderly fashion. It has everything you need to master HTML, publish sites to a server on the Web, create graphics for use on the Web, and keep your sites running smoothly.

But wait, there's more. Other books on how to create web pages just teach you the basic technical details, such as how to produce a boldface word. In this book, you'll also learn why you should be producing a particular effect and when you should use it. In addition, this book provides hints, suggestions, and examples of how to structure your overall website, not just the words on each page. This book won't just teach you how to create a website—it'll teach you how to create a great website and how to get people to come visit it.

In this book, examples are written in valid HTML5 and CSS3 using tags that work in all current browsers wherever possible. Exceptions and caveats are noted whenever I use tags that are obsolete or not included in HTML5.

**NOTE**

Visit our website and register this book at [www.informit.com/](http://www.informit.com/) register for convenient access to any updates, downloads, or errata that might be available for this book.

## Who Should Read This Book

Is this book for you? That depends:

- If you've seen what's out on the Web and you want to contribute your own content, this book is for you.
- If you work for a company that wants to create a website and you're not sure where to start, this book is for you.
- If you're an information developer, such as a technical writer, and you want to learn how the Web can help you present your information online, this book is for you.
- If you're just curious about how the Web works, some parts of this book are for you, although you might be able to find what you need on the Web itself.
- If you've created web pages before with text, images, and links, and you've played with a table or two and set up a few simple forms, you may be able to skim the first half of the book. The second half should still offer you a lot of helpful information.

## What This Book Contains

The lessons are arranged in a logical order, taking you from the simplest tasks to more advanced techniques:

- Part I: Getting Started  
In Part I, you'll get a general overview of the World Wide Web and what you can do with it. You'll also write your first (basic) web page with HTML and CSS.
- Part II: Creating Web Pages  
In Part II, you'll learn how to write simple documents in the HTML language and style them with CSS. You'll learn how to create lists on your pages as well as paragraphs of text, and you'll learn how to link your pages with hypertext links.

- Part III: Doing More with HTML and CSS

In Part III, you'll learn the meat of building web pages. You'll learn how to format text and style a page using CSS. You'll learn how to add images and create tables and forms and place them on your pages. You'll also learn how to lay out your web pages with CSS and make them responsive to the devices that are viewing them.

- Part IV: Using JavaScript and jQuery

In Part IV, you'll learn how you can extend the functionality of your web pages by adding JavaScript to them. First, we provide an overview of JavaScript and of jQuery. We provide some specific JavaScript examples you can use on your own pages. And you learn how to make inline frames and linked windows.

- Part V: Designing for Everyone

Part V gives you hints for creating a well-constructed website, and you'll learn how to design for mobile devices as well as make your site accessible so that it is usable by people with disabilities.

- Part VI: Going Live on the Web

In Part VI, you'll learn how to put your site up on the Web, including how to advertise the work you've done. You'll also learn how to use some of the features of your web server to make your life easier. And you'll get some tips for making your site searchable in the most popular search engines with search engine optimization (SEO).

## What You Need Before You Start

There are lots of books about how to use the Web. This book isn't one of them. We're assuming that if you're reading this book, you already have a working connection to the Internet, you have a modern web browser such as Chrome, Safari, Firefox, Opera, Internet Explorer version 10, or Microsoft Edge, and that you're familiar with the basics of how the Web and the Internet work. You should also have at least a passing acquaintance with some other elements of the Internet, such as email and FTP, because we refer to them in general terms in this book.

In other words, you need to have used the Web to provide content for the Web. If you meet this one simple qualification, read on!

Many of the screenshots in this book are made on a Macintosh computer, but you can do all the work on Windows or a Linux machine if that's what you use. You should just be familiar with how your operating system works and where common programs are located.

## Conventions Used in This Book

This book uses special typefaces and other graphical elements to highlight different types of information.

### Special Elements

Three types of “boxed” elements present pertinent information that relates to the topic being discussed: Note, Tip, and Caution as follows:

#### NOTE

Notes highlight special details about the current topic.

#### TIP

It's a good idea to read the tips because they present shortcuts or trouble-saving ideas for performing specific tasks.

#### CAUTION

Don't skip the cautions. They help you avoid making bad decisions or performing actions that can cause you trouble.

---

### ▼ Task

Tasks demonstrate how you can put the information in a lesson into practice by giving you a real working example.

---

## HTML Input and Output Examples

Throughout the book, we present exercises and examples of HTML input and output.

### Input ▼

---

An input icon identifies HTML code that you can type in yourself.

### Output ▼

---

An output icon indicates the results of the HTML input in a web browser such as Microsoft Internet Explorer.

## Special Fonts

Several items are presented in a monospace font, which can be plain or italic. Here's what each one means:

- `plain mono`—Applied to commands, filenames, file extensions, directory names, and HTML input. For example, HTML tags such as `<table>` and `<p>` appear in this font.
- `mono italic`—Applied to placeholders. A placeholder is a generic item that replaces something specific as part of a command or computer output. For instance, the term represented by `filename` would be the real name of the file, such as `myfile.txt`.

## Workshop

In the “Workshop” section, you can reinforce your knowledge of the concepts in the lesson by answering quiz questions or working on exercises. The Q&A provides additional information that didn't fit in neatly elsewhere in the lesson.

*This page intentionally left blank*

## LESSON 4

# Learning the Basics of HTML

Over the first three lessons, you learned about the World Wide Web, how to prepare to build websites, and why you need to use HTML to create a web page. In Lesson 3, “Introducing HTML and CSS,” you even created your first very simple web page. In this lesson, you learn about each of the basic HTML tags in more depth, and begin writing web pages with headings, paragraphs, and several different types of lists. We focus on the following topics and HTML tags:

- Tags for overall page structure: `<html>`, `<head>`, and `<body>`
- Tags for titles, headings, and paragraphs: `<title>`, `<h1>` through `<h6>`, and `<p>`
- Tags for comments: `<!--...-->`



## Structuring Your HTML

HTML defines three tags that are used to define the page's overall structure and provide some simple header information. These three tags—`<html>`, `<head>`, and `<body>`—make up the basic skeleton of every web page. They also provide simple information about the page (such as its title or its author) before loading the entire thing. The page structure tags don't affect what the page looks like when it's displayed; they're only there to help browsers.

### The DOCTYPE Identifier

Although it's not a page structure tag, the XHTML 1.0 and HTML5 standards impose an additional requirement on your web pages. The first line of each page must include a DOCTYPE identifier that defines the HTML version to which your page conforms, and in some cases, the Document Type Definition (DTD) that defines the specification. This is followed by the `<html>`, `<head>`, and `<body>` tags. In the following example, the HTML5 document type appears before the page structure tags:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
..your page content...
</body>
</html>
```

### The `<html>` Tag

The first page structure tag in every HTML page is the `<html>` tag. It indicates that the content of this file is in the HTML language. The `<html>` tag should immediately follow the DOCTYPE identifier (as mentioned in the previous note), as shown in the following example.

All the text and HTML elements in your web page should be placed within the beginning and ending HTML tags, like this:

```
<!DOCTYPE html>
<html>
..your page...
</html>
```

The `<html>` tag serves as a container for all of the tags that make up the page. It is required because both XML and SGML specify that every document have a root element. Were you to leave it out, which you shouldn't do because it would make your page invalid, the browser would make up an `<html>` tag for you so that the page would make sense to its HTML processor.

## The `<head>` Tag

The `<head>` tag is a container for the tags that contain information about the page, rather than information that will be displayed on the page. Generally, only a few tags are used in the `<head>` portion of the page (most notably, the page title, described later). You should never put any of the text of your page into the header (between `<head>` tags).

Here's a typical example of how you properly use the `<head>` tag. (You'll learn about `<title>` later.)

```
<!DOCTYPE html>
<html>
<head>
<title>This is the Title. It will be explained later on</title>
</head>
...your page...
</html>
```

## The `<body>` Tag

The content of your HTML page (represented in the following example as `...your page...`) resides within the `<body>` tag. This includes all the text and other content (links, pictures, and so on). In combination with the `<html>` and `<head>` tags, your page will look something like this:

```
<!DOCTYPE html><html>
<head>
<title>This is the Title. It will be explained later on</title>
</head>
<body>
...your page...
</body>
</html>
```

You might notice here that the tags are nested. That is, both `<body>` and `</body>` tags go inside the `<html>` tags; the same with both `<head>` tags. All HTML tags work this way,

forming individual nested sections of text. You should be careful never to overlap tags. That is, never do something like the following:

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

Whenever you close an HTML tag, make sure that you're closing the most recent unclosed tag. (You'll learn more about closing tags as you go on.)

## NOTE

In HTML, closing some tags is optional. In fact, in HTML 4.0 and earlier, closing tags were forbidden in some cases. The XHTML standard requires your markup to be well-formed XML, which leads to the requirement that all tags be closed. Because the examples shown in this book use HTML5, closing tags will be used only when they are required, but if you are working with XHTML you must close them.

## The Title

Each HTML page needs a title to indicate what the page describes. It appears in the title bar of the browser when people view the web page. The title is stored in your browser's bookmarks and in search engines when they index your pages. Use the `<title>` tag to give a page a title.

`<title>` tags are placed within the `<head>` tag and are normally used to describe the contents of the page, as follows:

```
<!DOCTYPE html><html>
<head>
<title>The Lion, The Witch, and the Wardrobe</title>
</head>
<body>
...your page...
</body>
</html>
```

Each page can have only one title, and that title can contain only plain text; that is, no other tags should appear inside the title.

Try to choose a title that's both short and descriptive of the content. Your title should be relevant even out of context. If someone browsing on the Web follows a random link and ends up on this page, or if a person finds your title in a friend's browser history list, would he have any idea what this page is about? You might not intend the page to be used independently of the pages you specifically linked to it, but because anyone can link to any page at any time, be prepared for that consequence and pick a helpful title.

**NOTE**

When search engines index your pages, each page title is captured and listed in the search results. The more descriptive your page title, the more likely it is that someone will choose your page from all the search results.

Also, because browsers put the title in the title bar of the window, you might have a limited amount of space. (Although the text within the `<title>` tag can be of any length, it might be cut off by the browser when it's displayed.) Here are some examples of good titles:

```
<title>Poisonous Plants of North America</title>
<title>Image Editing: A Tutorial</title>
<title>Upcoming Cemetery Tours, Summer 1999</title>
<title>Installing the Software: Opening the CD Case</title>
<title>Laura Lemay's Awesome Home Page</title>
```

Here are some not-so-good titles:

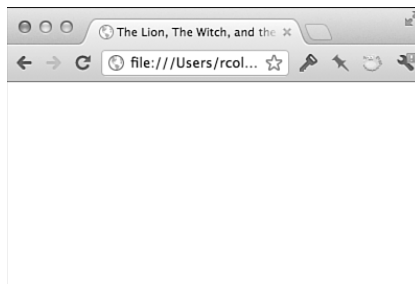
```
<title>Part Two</title>
<title>An Example</title>
<title>Nigel Franklin Hobbes</title>
<title>Minutes of the Second Meeting of the Fourth Conference of the
Committee for the Preservation of English Roses, Day Four, After Lunch</title>
```

Figure 4.1 shows how the following title looks in a browser:

```
<title>The Lion, the Witch, and the Wardrobe</title>
```

**FIGURE 4.1**

The title appears in the tab bar, not on the page.



## Headings

Headings are used to add titles to sections of a page. HTML defines six levels of headings. Heading tags look like the following:

```
<h1>Installing Your Safetee Lock</h1>
```

The numbers indicate heading levels ( $h_1$  through  $h_6$ ). The headings, when they're displayed, aren't numbered. They're displayed in larger and bolder text so that they stand out from regular text.

Think of the headings as items in an outline. If the text you're writing is structured, use the headings to express that structure, as shown in the following code:

```
<h1>Movies</h1>
  <h2>Action/Adventure</h2>
    <h3>Caper</h3>
    <h3>Sports</h3>
    <h3>Thriller</h3>
    <h3>War</h3>
  <h2>Comedy</h2>
    <h3>Romantic Comedy</h3>
    <h3>Slapstick</h3>
  <h2>Drama</h2>
    <h3>Buddy Movies</h3>
    <h3>Mystery</h3>
    <h3>Romance</h3>
  <h2>Horror</h2>
```

Notice that I've indented the headings in this example to better show the hierarchy. They don't have to be indented in your page; in fact, the browser ignores the indenting.

### TIP

Even though the browser ignores any indenting you include in your code, you will probably find it useful to indent your code so that it's easier to read. You'll find that any lengthy examples in this book are indented for that reason, and you'll probably want to carry that convention over to your own HTML code.

Unlike titles, headings can be any length, spanning many lines of text. Because headings are emphasized, however, having many lines of emphasized text might be tiring to read.

A common practice is to use a first-level heading at the top of your page to either duplicate the title or to provide a shorter or less context-specific form of the title. If you have a

page that shows several examples of folding bed sheets—for example, part of a long presentation on how to fold bed sheets—the title might look something like the following:

```
<title>How to Fold Sheets: Some Examples</title>
```

The topmost heading, however, might just be as follows:

```
<h1>Examples</h1>
```

## CAUTION

Don't use headings to display text in boldface type or to make certain parts of your page stand out more. Although the result might look as you intend, the markup will not represent the structure of your page. This comes into play for search engines, accessibility, and some browsers.

Figure 4.2 shows the following headings as they appear in a browser:

## Input ▼

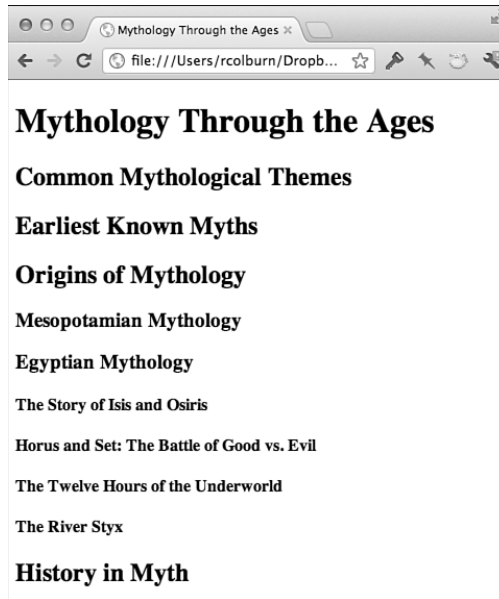
---

```
<h1>Mythology Through the Ages</h1>
  <h2>Common Mythological Themes</h2>
  <h2>Earliest Known Myths</h2>
  <h2>Origins of Mythology</h2>
    <h3>Mesopotamian Mythology</h3>
    <h3>Egyptian Mythology</h3>
      <h4>The Story of Isis and Osiris</h4>
      <h4>Horus and Set: The Battle of Good vs. Evil</h4>
      <h4>The Twelve Hours of the Underworld</h4>
      <h4>The River Styx</h4>
    <h2>History in Myth</h2>
```

---

**Output ▼**

**FIGURE 4.2**  
HTML heading elements.

**TIP**

From a visual perspective, headings 4 through 6 aren't visually interesting, but they do have meaning in terms of the document's structure. If using more than three levels of headings makes sense for the document you're creating, you can use those tags and then use styles to make them appear as you intend.

## Paragraphs

Now that you have a page title and several headings, you can add some ordinary paragraphs to the page.

Paragraphs are created using the `<p>` tag. The Enigern story should look like this:

```
<p>Slowly and deliberately, Enigern approached the mighty dragon.
A rustle in the trees of the nearby forest distracted his attention
for a brief moment, a near fatal mistake for the brave knight.</p>
<p>The dragon lunged at him, searing Enigern's armor with a rapid
blast of fiery breath. Enigern fell to the ground as the dragon
hovered over him. He quickly drew his sword and thrust it into the
dragon's chest.</p>
```

What if you want more (or less) space between your paragraphs than the browser provides by default? The answer is to use CSS. As you'll see, it provides fine control over the spacing of elements on the page, among other things. Figure 4.3 shows what happens when I add another paragraph about Enigern and the dragon to the page. The paragraph breaks are added between the closing and opening `<p>` tags in the text.

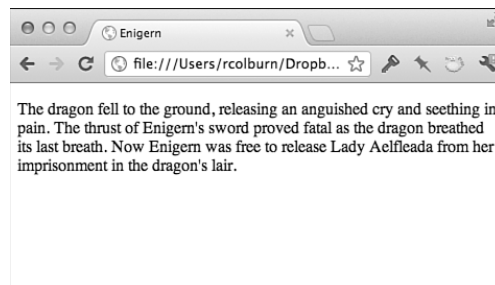
## Input ▼

```
<p>The dragon fell to the ground, releasing an anguished cry and seething in pain. The thrust of Enigern's sword proved fatal as the dragon breathed its last breath. Now Enigern was free to release Lady Aelfleada from her imprisonment in the dragon's lair. </p>
```

## Output ▼

**FIGURE 4.3**

An HTML paragraph.



The closing `</p>` tag, while not required, is important for defining the exact contents of a paragraph for CSS. Most web designers use it automatically, but if you don't need it, you can leave it out of your HTML.

## Comments

You can put comments into HTML pages to describe the page itself or to provide some kind of indication of the status of the page. Some source code control programs store the page status in comments, for example. Text in comments is ignored when the HTML file is parsed; comments never show up onscreen—that's why they're comments. Comments look like the following:

```
<!-- This is a comment -->
```

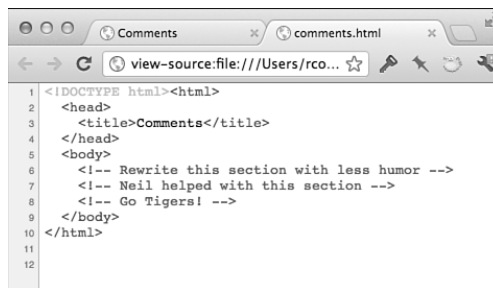


Here are some examples:

```
<!-- Rewrite this section with less humor -->
<!-- Neil helped with this section -->
<!-- Go Tigers! -->
```

As you can see from Figure 4.4, users can view your comments using the View Source functionality in their browsers, so don't put anything in comments that you don't want them to see.

**FIGURE 4.4**  
HTML comments displayed within the source for a page.



```
1 <!DOCTYPE html><html>
2 <head>
3 <title>Comments</title>
4 </head>
5 <body>
6 <!-- Rewrite this section with less humor -->
7 <!-- Neil helped with this section -->
8 <!-- Go Tigers! -->
9 </body>
10 </html>
11
12
```

## ▼ Exercise 4.1: Creating a Real HTML Page

At this point, you know enough to get started creating simple HTML pages. You understand what HTML is, you've been introduced to a handful of tags, and you've even opened an HTML file in your browser. You haven't created any links yet, but you'll get to that soon enough, in Lesson 6, "Working with Links."

This exercise shows you how to create an HTML file that uses the tags you've learned about up to this point. It'll give you a feel for what the tags look like when they're displayed onscreen and for the sorts of typical mistakes you're going to make. (Everyone makes them, and that's why using an HTML editor that does the typing for you is often helpful. The editor doesn't forget the closing tags, leave off the slash, or misspell the tag itself.)

So, create a simple example in your text editor. Your example doesn't have to say much of anything; in fact, all it needs to include are the structure tags, a title, a couple of head-

▼ ings, and a paragraph or two. Here's an example:

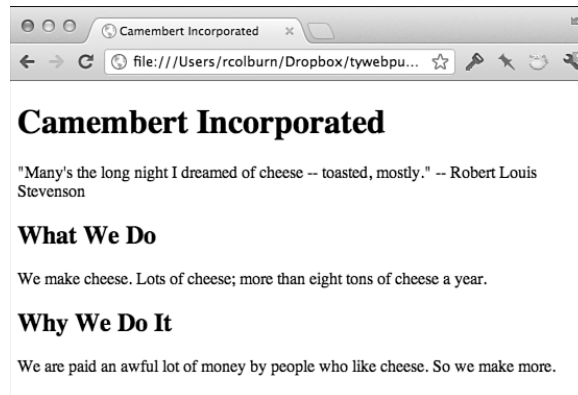
## Input ▼

```
<!DOCTYPE html><html>
<head>
<title>Camembert Incorporated</title>
</head>
<body>
<h1>Camembert Incorporated</h1>
<p>"Many's the long night I dreamed of cheese -- toasted, mostly."
-- Robert Louis Stevenson</p>
<h2>What We Do</h2>
<p>We make cheese. Lots of cheese; more than eight tons of cheese
a year.</p>
<h2>Why We Do It</h2>
<p>We are paid an awful lot of money by people who like cheese.
So we make more.</p>
</body>
</html>
```

Save the example to an HTML file, open it in your browser, and see how it came out. Figure 4.5 shows what the cheese factory example looks like.

## Output ▼

**FIGURE 4.5**  
The cheese factory example.



## Summary

HTML, a text-only markup language used to describe hypertext pages on the World Wide Web, describes the structure of a page, not its appearance.

In this lesson, you learned what HTML is and how to write and preview simple HTML files. You also learned about the HTML tags shown in Table 4.1.

**TABLE 4.1** HTML Tags from Lesson 4

Tag	Use
<code>&lt;html&gt; .. &lt;/html&gt;</code>	The entire HTML page
<code>&lt;head&gt; .. &lt;/head&gt;</code>	The head, or prologue, of the HTML page
<code>&lt;body&gt; .. &lt;/body&gt;</code>	All the other content in the HTML page
<code>&lt;title&gt; .. &lt;/title&gt;</code>	The title of the page
<code>&lt;h1&gt; .. &lt;/h1&gt;</code>	First-level heading
<code>&lt;h2&gt; .. &lt;/h2&gt;</code>	Second-level heading
<code>&lt;h3&gt; .. &lt;/h3&gt;</code>	Third-level heading
<code>&lt;h4&gt; .. &lt;/h4&gt;</code>	Fourth-level heading
<code>&lt;h5&gt; .. &lt;/h5&gt;</code>	Fifth-level heading
<code>&lt;h6&gt; .. &lt;/h6&gt;</code>	Sixth-level heading
<code>&lt;p&gt; .. &lt;/p&gt;</code>	A paragraph

## Workshop

You've learned a lot in this lesson, and the following workshop will help you remember some of the most important points. I've anticipated some of the questions you might have in the first section of the workshop.

### Q&A

**Q** In some web pages, I've noticed that the page structure tags (`<html>`, `<head>`, `<body>`) aren't used. Do I really need to include them if pages work just fine without them?

**A** Most browsers handle plain HTML without the page structure tags. The only tag that is required in HTML5 is the `<title>` tag. But it's a good idea to get into the habit of using the structure tags now. Including these tags ensures that browsers handle your markup properly. And, using these tags is the correct thing to do if you want your pages to conform to true HTML format.

**Q Is the `<p>` tag the general-purpose tag for use when styling a page?**

A No. The `<div>` tag is the general-purpose tag for containing content on a page. The `<p>` tag is intended specifically to hold paragraphs of text. There are many tags that are not valid when placed within a `<p>` tag, including `<div>`. You'll learn more about `<div>` in Lesson 7, "Formatting Text with HTML and CSS."

**Q Is it possible to put HTML tags inside comments?**

A Yes, you can enclose HTML tags within comments, and the browser will not display them. In fact, it's common to use comments to temporarily hide sections of a page, especially when testing things. Programmers (and web developers) generally refer to this as "commenting it out."

## Quiz

1. What three HTML tags are used to describe the overall structure of a web page, and what do each of them define?
2. Where does the `<title>` tag go, and what is it used for?
3. How many different levels of headings does HTML support? What are their tags?
4. Why is it a good idea to use two-sided paragraph tags, even though the closing tag `</p>` is optional in HTML?

## Quiz Answers

1. The `<html>` tag indicates that the file is in the HTML language. The `<head>` tag specifies that the lines within the beginning and ending points of the tag are the prologue to the rest of the file. The `<body>` tag encloses the remainder of your HTML page (text, links, pictures, and so on).
2. The `<title>` tag is used to indicate the title of a web page in a browser's title bar and bookmarks. It is also used by search engines. This tag always goes inside the `<head>` tags.
3. HTML supports six levels of headings. Their tags are `<h1 .. /h1>` through `<h6 .. /h6>`.
4. The closing `</p>` tag becomes important when using CSS to style your text. Closing tags also are required for XHTML 1.0.

## Exercises

1. Using the Camembert Incorporated page as an example, create a page that briefly describes topics that you would like to cover on your own website.
2. Create a second page that provides further information about one of the topics you listed in the first exercise. Include a couple of subheadings (such as those shown in Figure 4.2). If you feel really adventurous, complete the page's content and include lists where you think they enhance the page. This exercise will help prepare you for Lesson 5, "Organizing Information with Lists."

# Index

## Symbols

//, JavaScript comments, 474

+ (addition) operator, 479

& (ampersand)

named entities, 140

troubleshooting, 142

/ (division) operator, 479

== (equality) operator, 483

> (greater than symbol),  
troubleshooting, 142

> (greater than) operator, 483

>= (greater than or equal to)  
operator, 483

!= (inequality) operator, 483

< (less than symbol),  
troubleshooting, 142

< (less than) operator, 483

<= (less than or equal to)  
operator, 483

% (modulus) operator, 479

\* (multiplication) operator,  
479

% (percent sign), in email  
addresses, 116

# (pound signs)

CSS and, 164

numbered entities, 141

' ' (quotation marks), href  
attributes, 93

;(semicolon), named entities,  
140

- (subtraction) operator, 479

## A

<a> tag, 93

accesskey attribute, 615

frameset attribute,  
557-562

name attributes and, 105

tabindex attribute, 615

title attribute, 615

<abbr> tags, as logical style  
tags, 123

absbottom alignment  
(images), 207

absolute pathnames, 96-97

linking local pages, 95-98

versus relative pathnames,  
97

absolute positioning,

absolute positioning (CSS),  
293

dynamic overlays, 297-300

nesting elements, 295-297

- page layout, 426-428
  - properties, 293-295
- absolute units (CSS), 166-167
- access keys, forms, 352
- accessibility, 611, 622
  - alternative browsers, 613
  - Building Accessible Websites, 621
  - designing for, 617
    - color, 617
    - fonts, 618
    - forms, 619
    - frames, 619
    - HTML tags, 619
    - linked windows, 619
  - Dive Into Accessibility website, 621
  - myths, 611-612
  - Section 508, 612
  - transcripts, 692
  - validating, 620
  - W3C Web Accessibility Initiative, 621
  - writing HTML, 613
    - images, 616
    - links, 615
- accessing log files, 644-645
- accesskey attribute (<a> tag), 615
- accounts, Twitter, 641
- action attribute, <form> tag, 315, 321, 366
- active state (links), 173
- addition operator (+), 479
- <address> tag, 134-135, 154, 598
- addresses (Web). See URLs
- adjusting layout, 454, 457-459
  - RWD, 445
- Adobe Color CC, 170
- Adobe Photoshop, 198
- Adobe website, 198
- advertising websites, 638
  - brochures, 644
  - business cards, 644
  - links from other sites, 639-640
  - social media, 640-642
- AJAX, 496
  - external data, loading, 521-525
  - jQuery, 520-525
- alert message, JavaScript, 494
- align attributes
  - <hr> tags, 133
  - <img> tag, 209
  - <table> tag, 263
- alignment
  - images, 207-209
  - tables, 263
    - cells, 264-265
    - columns, 276-279
    - rows, 279-282
- alink attribute (<body> tag), 172
- allowfullscreen (<embed> tag), 405
- allowscriptaccess (<embed> tag), 405
- alt attribute, <area> tag, 231
- alt attribute (<img> tag), 201, 237, 616
- alternative content, <object> tag, 403-404
- Amazon.com, 630
- ampersands (&)
  - named entities, 140
  - troubleshooting, 142
- analytics, 700
- anchor tags, 117. See also <a> tag
- anchors, 104
  - creating, 105
  - hash signs and, 108
  - linking in same page, 110
  - sample exercises, 105-109
- anonymous FTP, URLs, 114-115
- anonymous functions, JavaScript, 542
- Apple Safari, 17
- Apple's QuickTime, container format, 392
- <area> tag, 230-231

**arguments**

JavaScript, 486

this, 494

array\_key\_exists() function, 658

**arrays**

JavaScript, 489

PHP, 656-659

associative arrays, 657

indexes, 656

&lt;article&gt; element, 378

articles, guest posting, 639-640

ASCII, 139

&lt;aside&gt; element, 379

ASP.NET, 683

associating image maps with images, 231

associative arrays, PHP arrays, 657

asterisk (\*), multiplication operator, 479

attribute selectors, 193-194

**attributes**

&lt;a&gt; tag

accesskey, 615

tabindex, 615

target, 557-562

title, 615

&lt;area&gt; tag, 230-231

alt, 231

coords, 231

href, 231

shape, 230

&lt;audio&gt; tag, 414

&lt;body&gt; tag, 172

alink, 172

link, 172

text, 172

vlink, 172

&lt;br&gt; tag, clear, 212

&lt;embed&gt; tag, 405

&lt;hr&gt; tag, 132-133

href attributes, links, creating, 90

HTML, 50

&lt;iframe&gt; tag, 571

height, 564

name, 564

src, 564

width, 564

&lt;img&gt; tag, 237

align, 209

alt, 201, 616

height, 219

hspace, 213

src, 201

usemap, 231-232

vspace, 213

width, 219

manipulating jQuery, 513-514

&lt;map&gt; tag, name, 230

name, &lt;a&gt; tags and, 105

&lt;object&gt; tag, 403

style, 51

type attribute, 77

value attributes, 77

&lt;video&gt; tag, 397

audio, embedding, 413-416

audio players, Flash (embedding), 414-416

&lt;audio&gt; tag, 413-416

attributes, 414

authentication, 627

auto margins (CSS), 180

autoplay attribute

&lt;audio&gt; tag, 414

&lt;video&gt; tag, 397

**avoiding**

browser-specific technology, 584

Flash in mobile web design, 593

"here" syndrome, 588



**B**

`<b>` tags, 124, 154

background-color property

table background colors, changing, 261

background-position property (CSS), 222

backgrounds

background images, 221, 224

CSS (cascading style sheets)

background-position, 222

elements, 224

`<img>` tag, 224

tables, background colors, 261-262

Baidu, 691

bandwidth limitations, 631

`<base>` tag, 562-563, 571

baseline alignment (images), 207

best practices for RWD, 467-468

Better-Organized Style Sheet listing (15.7), 438

binding events, JQuery, 504-505

Bing, 691

blank target name, 557

block-level elements

backgrounds, 224

versus character-level elements, 122

`<blockquote>` tag, 135-136, 154

Blogger.com, 37

blue\_page window, 558-562

Bobby (accessibility validator), 620

body (tables), 281

`<body>` tag, 59-60, 172

CSS, applying, 194

CSS box properties and, 181-182

bold text, design tips, 583-584

bold type, headings as, 63

books, Building Accessible Websites, 621

Bookworm Web page code, 151-153

border attribute, 284

`<table>` tag, 253-256

borders

CSS box model, 175-177

table borders, changing, 253-256

bottom property, absolute positioning, 293

bottom property, CSS positioning, 288

box model (CSS), 174

borders, 175-177

clearing elements, 187

controlling size and element display,  
182-185

floating elements, 185-189

margins, 177-182

padding, 177-182

`<br>` tags, 133-134, 155, 212

brackets, HTML tags, 45

Braille browsers, 613

breakpoints, 452-453

brevity in Web page design, 582

brochures, including Web address on, 644

browsers

accessibility, alternative browsers, 613

compatibility, 417

getting image coordinates, 233

JavaScript

compatibility, 496

integration, 473

PHP scripts, dependence, 685

building

responsive tables, 463, 466-467

style sheets with media queries, 454

Building Accessible Websites, 621

built-in functions, PHP, 666

bullets, images, 224-225

business cards, including Web address on, 644

button attribute, `<input>` tag, 367

`<button>` tag, 340, 366

buttons in Developer Tools, 31. *See also* form controls

## C

- Camino, 17
- canonical links, 698
- <caption> tag, 247-251, 283
- captions, tables, 242
  - creating, 247-251
- carriage returns, 634
- cascading (CSS), 165-166
- Cascading Style Sheets (CSS). *See* CSS (Cascading Style Sheets)
- case sensitivity, filenames, 634
- cellpadding attribute (<table> tag), 256
- cells. *See also* columns
  - alignment, 264-265
  - colors, background colors, 261-262
  - creating, 244-246
  - empty cells, creating, 246
  - padding, 256
  - spacing, 257-258
  - spanning multiple rows/columns, 266-269
  - tables, 242
- cellspacing attribute (<table> tag), 257
- celsiusToFahrenheit() function, 680
- character encoding, 139
  - UTF-8, 140
- character entities, 138
  - named entities, 140-141
  - numbered entities, 141
  - reserved characters, 141
- character formatting, design tips, 583-584
- character-level elements, 122
  - versus block-level elements, 122
  - logical style tags, 122-123
  - physical style tags, 124
- check box controls, creating with <input> tag, 334-335
- checkbox attribute, <input> tag, 367
- checked attribute, <input> tag, 335
  - radio, 336
- checking file permissions, 636
- chmod command, 636
- choosing, web servers, 627
  - bandwidth limitations, 631
  - domain parking, 629
  - ISPs (Internet service providers), 628
  - personal servers, 629
  - school servers, 627
  - Web presence providers, 628-629
  - work servers, 627
- Chrome Developer Tools, 171
- Chrome (Google), 15
- circles, imagemap coordinates, 229
- <cite> tag, 154
  - as logical style tags, 123
- clarity in Web page design, 582
- class attribute, overview, 163-164
- classes (CSS)
  - naming conventions, 195
  - overview, 163
- classes, manipulating (jQuery), 508-511
- clear attribute (<br> tag), 212
- clear property (CSS), 187
- clearing text wrap, 211-212
- click to call, 596
- client-side images, 226
  - <area> tag, 230-231
    - associating with images, 231
  - circle coordinates, 229
  - imagemap creation software, 228
  - image selection, 226
  - jukebox image example, 232-236
  - <map> tag, 230-231
  - polygon coordinates, 229
  - rectangle coordinates, 230
  - troubleshooting, 238
- clients, FTP clients, 635

- cloaking, 695
- closing HTML tags, 45, 60, 72
- cm unit (CSS), 167
- Coda, 27
- code listings
  - 15.1 (Using <div> Tags to Create Sections for Positioning), 424-425
  - 15.2 (Style Sheet for Page Layout), 427
  - 15.3 (Style Sheet for Colors and Fonts), 429
  - 15.4 (Moving One Section Before Another), 431
  - 15.5 (Float-Based Layouts in CSS), 433-434
  - 15.6 (Randomly Organized Style Sheet), 437
  - 15.7 (Better-Organized Style Sheet), 438
- <code> tag, 154
  - as logical style tags, 123
- codecs, 416
  - H.264 video codec, 392
    - converting to, 393-396
- <col> tag, 276-277, 283
- <colgroup> tag, 276-277, 283
- color property (CSS), 172
  - table background colors, changing, 261
- Color Schemer, 170
- colors
  - accessibility, 617
  - CSS properties, 173
  - naming, 170
  - page layout, 429-430
  - tables, background colors, 261-262
- colspan attribute, 284
  - <td> tag, 267
  - <th> tag, 267
- columns, 266
  - aligning, 276-279
  - grouping, 276-279
  - width of, 259-260
- commands, chmod, 636
- comments
  - HTML tags, 69
  - JavaScript, 474
  - PHP, 654-655
- comments (HTML), 65-66
- comparison operators (JavaScript), 483
- compatibility, browsers, 417
- complex framesets, creating, 558
- compressing
  - CSS, 594-595
  - JavaScript, 594-595
- conditional operators, PHP, 662
- conditional statements, PHP, 661
- consistent page layout for mobile web, 586
- container formats, video, 392-393
- content
  - adding
    - jQuery, 514-518
    - to sample pages, 145
  - alternative content, <object> tag, 403-404
  - duplicate content, 696
    - redirecting, 698-699
  - formatting. See HTML tags
  - ideas for, 33
  - improving SEO, 694
  - modifying jQuery, 508-518
  - removing jQuery, 514-518
  - splitting topics across pages, 598
  - titles, 61
  - Web pages, adding to, 546-550
  - wireframing, 33-36
- content-management applications, 36-37
- contextual selectors (CSS), 162-163
- control structures, JavaScript, 482-486
- controlling loop execution, PHP, 665-666
- controls attribute (<audio> tag), 414
- controls attribute (<video> tag), 397
- controls (forms), 314
- controls, HTML5, 328-329

- cookies, PHP, 683
- coordinates
  - image coordinates, getting from browsers, 233
  - imagemaps, determining, 227-228
- coordinates (imagemaps)
  - <area> tag, 230-231
  - circle coordinates, 229
  - imagemap creation software, 228
  - <map> tag, 230-231
  - polygon coordinates, 229
  - rectangle coordinates, 230
- coords attribute (<area> tag), 231
- crawlers, 689
- cross-platform compatibility, reasons for, 44
- cross-platform Web compatibility, 9
- cross-site scripting, preventing, 669
- CRUD applications, 674
- CSS (cascading style sheets), 21, 37, 421
  - advantages of, 50
  - applying, 160
  - background properties, block-level elements, 224
  - backgrounds, <img> tag, 224
  - box model, 174
    - borders, 175-177
    - clearing elements, 187
    - controlling size and element display, 182-185
    - floating elements, 185-189
    - margins, 177-182
    - padding, 177-182
  - <body> tag, 194
  - cascading, 165-166
  - compressing for mobile web, 594-595
  - creating, 50-51
  - CSS files, creating, 161
  - font properties, 126-127, 142
  - form properties, applying styles, 361-364
  - forms, applying to, 359-361
  - HTML and. *See also* HTML
  - HTML tags, combining with, 51
  - layout style sheets, writing, 426-433
  - links, modifying, 173-174
  - list-style-image properties, unordered lists, 80
  - list-style-position properties, unordered lists, 80
  - media queries, 450
    - breakpoints, 452-453
    - media features, 451-452
    - media types, 450-451
    - style sheets, building, 454
  - overview, 159
  - page-level styles, creating, 160
  - placing in external files, 593-594
  - positioning, 288
    - absolute positioning, 293-300
    - fixed positioning, 301-303
    - relative positioning, 288-290, 293
    - static positioning, 288
    - top/left/bottom/right properties, 288
    - z-index property (stacking), 303-306
  - properties
    - background-position, 222
    - color, 172
    - font-family, 156
    - font-size, 156
    - font-style, 156
    - font-variant, 156
    - font-weight, 156
    - retrieving, 507-508
    - text-decoration, 156
  - sample pages, creating, 143-144
    - adding content, 145
    - adding tables of contents, 146
    - Bookworm Web page code, 151-153
    - frameworks, 144-145
    - link menus, 148-149
    - page descriptions, 147

- planning the page, 144
    - signatures, 150-151
    - testing results, 153-154
    - unordered lists, 150
  - selectors, 162, 189-190
    - attribute selectors, 193-194
    - classes, 163-164. *See also* classes (CSS)
    - contextual selectors, 162-163
    - IDs, 163-164. *See also* IDs (CSS)
    - pseudo-classes, 190-193
  - sitewide style sheets, creating, 161-162
  - spacing units, 143
  - <span> tags, 125
  - standards, 195
  - tables, laying out, 422-434
  - text-decoration properties, 125-126
  - troubleshooting, 195
  - units of measure, 166
    - length units, 166-167
    - percentage units, 168
  - URLs, using, 168
  - web design, 435-436
    - organization, 436-437
    - site-wide style sheets, 439
  - .css file extension, 161
  - customizing videos, YouTube, 389
  - CuteFTP, 635
  - Cynthia Says validator, 620
- D**
- Dailymotion.com, 391
  - Dashboard (Google Analytics), 647-648
  - data attribute (<object> tag), 403
  - data types, JavaScript, 488-489
  - database connectivity, PHP, 682
  - <datalist>, 327-328
  - date and time types (HTML5), 330-331
  - <dd> tags, 72, 81, 86
  - default index files, 632-633
  - definition links, 590
  - definition terms. *See* <dt> tags
  - density of keywords, 693
  - deployment, PHP files, 685
  - descendant selector, 189-190
  - descriptive titles, 583
  - design, wireframing, 33-36
  - designing
    - for accessibility, 617
      - color, 617
      - fonts, 618
      - forms, 619
      - frames, 619
      - HTML tags, 619
      - linked windows, 619
    - web pages
      - accessibility. *See* accessibility
      - determining user preferences, 610-611
      - emphasis, 619
      - first time users versus regular users, 609
      - images, image etiquette, 237
      - navigation, 608-609
      - search engines, 606
      - URLs, 606-608
  - designing websites, 581
    - brevity, 582
    - browser-specific terminology, 584
    - clarity, 582
    - consistent layout, 586
    - emphasis, 583-584
    - grouping related information, 586
    - headings, 585
    - images, 591
    - links, 587
    - organizing for quick scanning, 582
    - page validation, 578-580
    - proofreading, 584-585

- spell checking, 584-585
- splitting topics across pages, 598
- standalone pages, 583
- standards compliance, 578-579
- designs, style sheets, 435-436
  - organization, 436-437
  - site-wide, 439
- Developer Tools, 29
  - buttons, 31
  - Inspector, 31-32
  - tabs, 30
  - View Source, 30
- development environments, JavaScript, 495
- <dfn> tag, 154
  - as logical style tags, 123
- dictionaries (PHP), 657
- <dir> tags, 72
- directories, 632
  - folders and, 95
  - index files, web servers, 632
  - in URLs, 112
- disabled form controls, creating, 352-353
- distributed nature of the Web, 9-10
- <div> tag, 423-426
- Dive Into Accessibility website, 621
- division operator (/), 479
- <dl> tags, 81, 86
- Doctorow, Cory, 10
- DOCTYPE identifiers, 58
- DOCTYPE tag, 47
- document object (JavaScript), 491
- document roots, 97
- documentation, JavaScript, 495
- documents (HTML)
  - colors
    - CSS properties, 173
    - naming, 170
  - converting web pages to, 601

- images
  - adding, 200-205
  - aligning with text, 207-209
  - as links, 214-218
  - background images, 221-224
  - bullets, 224-225
  - GIF (Graphics Interchange Format), 198-238
  - Halloween House Web page example, 203-205
  - height/width, 219
  - image etiquette, 237
  - inline images, 200-201, 205-207
  - JPEG (Joint Photographic Experts Group), 199, 238
  - navigation icons, 215-218
  - PNG (Portable Network Graphics), 199
  - preparing for Web, 198
  - scaling, 219
  - spacing around, 213
  - wrapping text around, 210-212
- moving between Web servers, 633
  - carriage returns/line feeds, 634
  - filename restrictions, 633
  - FTP (File Transfer Protocol), 634-635
- organizing, 630
  - default index files, 632-633
  - directories, 632
  - filenames, 632-633
  - Web server setup, 630-631
- text
  - aligning images with, 207-209
  - wrapping around images, 210-212
- text formatting, design tips, 583-584
- DOM (Document Object Model), 503
  - accessing
    - JavaScript methods, 549
    - methods, 543
  - navigating node properties, 545
- domain names, registering, 628-629

- domain parking, 629
- domains, 629
- dotster.com, 629
- drop-down menus, creating, 306-311
- <dt> tags, 72, 81, 86
- Dunbar Project example
  - absolute positioning, 426-428
  - colors/fonts, 429-430
  - floated columns, 433-434
  - HTML sections, 423-426
  - redesigning layout, 432-433
  - reordering sections, 431
  - style sheet, 426-430
  - tables, 422-423
- duplicate content, 696
  - redirecting, 698-699
- dynamic overlays, 297-300
- dynamic nature of Web, 10-11

## E

- echo() function, 674
- editing PHP files, 685
- element displays, CSS box model (controlling), 182-185
- elements
  - backgrounds, 224
  - hiding, JQuery, 505-506
  - JavaScript, hiding/showing, 538-546
  - showing JQuery, 505-506
- elements (HTML), 45
- The Elements of Style, 582
- <em> tag, 102, 154
  - as logical style tags, 122
- em unit (CSS), 167
- em values, 440
- Emacs, 26
- email, sending (PHP), 682
- <embed> tag, 404-405, 416
  - attributes, 405
- embedding
  - audio, 413-416
  - Flash multimedia
    - <object> tag, 400-404
    - SWFObject, 406-408
  - video, 386-391
    - <video> tag, 396-400
- empty cells, creating, 246
- enctype attribute, <form> tag, 323, 339, 366
- Enterprise Edition (Java), 684
- equal sign (=), equality operator, 483
- equality operator (==), 483
- escape codes, HTML reserved characters, 141
- event-driven models of execution, JavaScript, 472
- event handlers
  - JavaScript, 492
  - onblur, 492
  - onchange, 492
  - onclick, 492
  - onfocus, 493
  - onload, 493
  - onmouseover, 493
  - onselect, 493
  - onsubmit, 493, 529
  - onunload, 493
- event.preventDefault() method, 531
- events
  - binding, JQuery, 504-505
  - event handlers, onsubmit, 529
  - JavaScript, 491-495
- ex unit (CSS), 167
- expand all/collapse all links, FAQs (adding to), 547
- explicit navigation links, 589
- expressions, JavaScript, 479-480
- eXtensible HyperText Markup Language (XHTML). See XHTML

eXtensible Markup Language (XML), 52

external data, AJAX (loading), 521-525

external files

    CSS, placing in, 593-594

    JavaScript, placing in, 593-594

    websites, 622

## F

Facebook, advertising websites via, 640-642

FAQs, expand all/collapse all links (adding to), 547

<fieldset> tag, grouping controls, 350-351

<form> tag

    action attribute, 321

    enctype attribute, file, 339

    using, 319-323

feature images, sizing, 220

Fetch, 635

<fieldset> tag, 366

file attribute, <input> tag, 367

file extensions, 632

file formats

    GIF (Graphics Interchange Format), 198-199, 238

    JPEG (Joint Photographic Experts Group), 199, 238

    PNG (Portable Network Graphics), 199

File Not Found errors, 636

File Transfer Protocol (FTP), 111, 634-635

file upload controls, creating with <input> tag, 339-340

file uploads, PHP, 683

file URLs, 116-117

filenames, 632-633

    case sensitivity, 634

    restrictions, 633

files

    determining type of, 626

    file extensions, 632

    File Not Found errors, 636

    filenames

        case sensitivity, 634

        restrictions, 633

    Forbidden errors, 636

    GIF (Graphics Interchange Format), 198-199, 238

    HTML files, 45-48

    index.html files, creating, 632-633

    JPEG (Joint Photographic Experts Group), 199, 238

    log files, 645

    managing, 626

    moving between Web servers, 633

        carriage returns/line feeds, 634

        filename restrictions, 633

        FTP (File Transfer Protocol), 634-635

    organizing, 630

        default index files, 632-633

        directories, 632

        filenames, 632-633

        Web server setup, 630-631

    permissions, checking, 636

    PHP files, 685

    PNG (Portable Network Graphics), 199

    server-side processing, 627

Firefox. *See* Mozilla Firefox

fixed layouts, 426

fixed positioning (CSS), 301-303

Flash audio players, embedding, 414-416

Flash multimedia, embedding

    <object> tag, 400-404

    SWFObject, 406-408

Flash video players, 408

    Flowplayer, 410-412

    JW Player, 408-410



- flashvars (<embed> tag), 405
- Flickr.com, 391
- Float-Based Layouts in CSS listing (15.5), 433-434
- floated columns layout, 433-434
- floating elements, CSS box model, 185-189
- floating frames, 52
- Flowplayer, 410-412, 415
- folders, 95
- font-family property (CSS), 156
- font properties, 127, 142
  - (CSS), 126
- font-size property (CSS), 156
- font-style property (CSS), 156
- font-variant property (CSS), 156
- font-weight property (CSS), 156
- fonts
  - accessibility, 618
  - page layout, 429-430
  - sans-serif fonts, 51
  - serifs, 51
- footer element, 376
- footers (tables), 280
- for loops, 484
  - JavaScript, 484
  - PHP, 664
- Forbidden errors, 636
- foreach loops, PHP, 663
- form controls
  - <button> tag, 340
  - <label> tag, displaying label elements, 323-324
  - creating with <input> tag, 324
    - check box controls, 334-335
    - file upload controls, 339-340
    - generic buttons, 337-338
    - hidden form fields, 338-339
    - password controls, 332-333
    - radio buttons, 336
    - reset buttons, 334
    - submit buttons, 333-334
    - text controls, 325-326
  - disabled controls, creating, 352-353
  - <fieldset> tag, grouping controls, 350-351
  - <legend> tag, grouping controls, 350-351
  - naming, 317
  - navigation
    - access keys, 352
    - default navigation, changing, 351-352
  - <option> tag, menus, creating, 342-345
  - readonly controls, creating, 352-353
  - registration form example, 345-348
  - <select> tag, menus, creating, 342-345
  - submit buttons, images as, 337
  - <textarea> tag, 341-342
- form tags, 314, 366
  - action attribute, 315
  - enctype attribute, 323
  - get method, 315-316
  - method attribute, 315-316, 321-322, 640
  - post method, 315-316
  - using, 319-323
- formatting
  - content. See HTML tags
  - text, HTML pages, 49, 54
- forms, 13, 313-315
  - accessibility, 619
  - check box controls, creating with <input> tag, 334-335
  - controls, 314
  - creating, 314-319
  - CSS (cascading style sheets), applying to, 359-361
  - CSS properties, applying styles, 361-364
  - <form> tag, using, 319-323
  - file upload controls, creating with <input> tag, 339-340
  - form controls. See form controls
  - <form> tag. See form tags

- generic buttons, creating with `<input>` tag, 337-338
  - hidden form fields, creating with `<input>` tag, 338-339
  - HTML, 313
  - `<input>` tag, 316-317
    - type attribute, 316
    - value attribute, 317
  - navigation
    - access keys, 352
    - default navigation, changing, 351-352
  - password controls, creating with `<input>` tag, 332-333
  - PHP, 668-669
    - parameters with multiple values, 669-670
    - presenting forms, 674-676
    - validating forms, 670-672
  - planning, 365
  - radio buttons, creating with `<input>` tag, 336
  - registration form, 13
  - required fields, testing, 531-533
  - reset buttons, creating with `<input>` tag, 334
  - security, 353
  - submit buttons
    - creating with `<input>` tag, 333-334
    - images as, 337
  - tags, 314
  - telephone numbers, validating, 329
  - testing, 368
  - text controls, creating with `<input>` tag, 325-326
  - text form controls, 316
  - validating with JavaScript, 528
    - code listing, 534-535
    - onsubmit event handler, 529
    - this form object, 532
  - values, manipulating, 511-512
  - forms processing, server-side scripts and, 626
  - frames, 555-556
    - accessibility, 619
  - framesets, 557
    - `<base>` tag, 562-563
    - creating, 560-562
    - creating complex, 558
  - inline frames, 563-566
  - linked windows, 557-561
  - magic target names, 557
  - framesets, 52, 557
    - `<base>` tag, 562-563
    - creating, 560-562
    - creating complex, 558
  - frameworks, HTML files (creating), 144-145
  - FTP (File Transfer Protocol), 38, 111, 634-635
    - anonymous FTP, URLs, 114-115
    - navigating FTP servers, 114
    - non-anonymous FTP, URLs, 115
  - FTP Explorer, 635
  - functions
    - `array_key_exists()`, 658
    - `celsiusToFahrenheit()`, 680
    - `echo()`, 674
    - `is_array()`, 658
    - JavaScript, 486-488
      - anonymous functions, 542
    - PHP
      - built-in functions, 666
      - user-defined functions. *See* user-defined functions
    - `popup()`, 568-570
    - `unset()`, 657
- ## G
- generic buttons, creating with `<input>` tag, 337-338
  - geolocation, 595
  - gestures, tapping, 587

- get method (<form> tag), 315-316
  - get method (forms), 321-322, 640
  - getElementsByClassName() method, 500
  - GIF (Graphics Interchange Format), 198-199, 238
  - global attributes (HTML), 50
  - godaddy.com, 629
  - Google, 690
    - Webmaster tools, 696
  - Google Analytics, 645-646
    - Dashboard, 647-648
    - installing, 646-647
    - reports, 647-648
  - Google Chrome, 15
    - supported character encoding, 139
  - Google Chrome Developer Tools, 29
    - buttons, 31
    - Inspector, 31-32
    - tabs, 30
    - View Source, 30
  - Graphics Interchange Format (GIF), 198-199, 238
  - greater than operator (>), 483
  - greater than or equal to operator (>=), 483
  - greater than symbol (>), troubleshooting, 142
  - grouping table columns, 276-279
  - grouping related information, mobile web design, 586
  - guest posting, 639-640
- H**
- H.264 codec versus Ogg Theora, 417
  - H.264 video codec, 392
    - converting to, 393-396
  - Halloween House Web page, 202-205
  - hardware considerations for mobile web design, 592
  - hash signs, anchors and, 108
  - hashes (PHP), 657
  - <head> tag, 59
  - header element, 376
  - heading levels, 62-64
  - heading levels (HTML tags), style attributes and, 51
  - headings
    - boldface, 63
    - importance of, 582
    - mobile web design, 585
    - tables, 242, 279-280
  - height attributes
    - <embed> tag, 405
    - <iframe> tag, 564, 571
    - <img> tag, 219, 592
    - <object> tag, 403
    - <video> tag, 397
  - height of images, 219
  - height property for absolute positioning, 294-295
  - hidden attribute, <input> tag, 367
  - hidden form fields, creating with <input> tag, 338-339
  - hiding elements, JavaScript, 538-546
  - history of HTML page layout, 372-373
  - history object (JavaScript), 491
  - history of RWD, 444
  - hits, websites, 638
  - home pages, 24-25, 40
    - ideas for, 25-26
    - initial visit, 25
    - linking to, 597
  - horizontal rules. See <hr> tag
  - hosting video, 391-395
  - hosting videos, YouTube, 387-388
  - hostnames (URLs), port numbers, 111
  - hover state (links), 173
  - <hr> tag, 130-131, 155
    - attributes, 132-133
    - overview, 131

- href attributes
  - <area> tag, 231
  - links, creating, 90
- hspace attribute (<img> tag), 213
- .htm versus .html, 47
- HTML (hypertext markup language), 37, 42
  - adaptability, reasons for, 44
  - as structurer, 42-43
  - attributes, 50
  - address tags, 134-135, 154
  - <b> tags, 154
  - <blockquote> tag, 135-136, 154
  - <br> tags, 155
  - <cite> tags, 154
  - <code> tags, 154
  - CSS (cascading style sheets) and, 43. *See also* CSS (cascading style sheets)
  - <dfn> tags, 154
  - <em> tags, 154
  - files, 45-48
  - formatting text, 54
  - <hr> tags, 155
  - <i> tags, 154
  - <kbd> tags, 154
  - links. *See* links
  - overview, 42
  - page layout sections, 423-426
  - page layouts and, 43
  - parsing, 43
  - <pre> tags, 154
  - <samp> tags, 155
  - sample pages, creating, 47
  - <small> tags, 155
  - <span> tags, 155
  - <strong> tags, 155
  - structure, writing with, 423-425
  - structuring, 58
  - <sub> tags, 155
  - <sup> tags, 155
  - tables. *See* tables
  - tags, <div>, 425
  - <u> tags, 155
  - <var> tags, 155
  - writing accessible HTML, 613
    - images, 616
    - links, 615
- HTML documents
  - colors
    - CSS properties, 173
    - naming, 170
  - images
    - adding, 200-205
    - aligning with text, 207-209
    - as links, 214-218
    - background images, 221, 224
    - bullets, 224-225
    - GIF (Graphics Interchange Format), 198-199, 238
    - Halloween House Web page example, 203-205
    - height/width, 219
    - image etiquette, 237
    - inline images, 200-201, 205-207
    - JPEG (Joint Photographic Experts Group), 199, 238
    - navigation icons, 215-218
    - PNG (Portable Network Graphics), 199
    - preparing for Web, 198
    - scaling, 219
    - spacing around, 213
    - wrapping text around, 210-212
  - moving between Web servers, 633
    - carriage returns/line feeds, 634
    - filename restrictions, 633
    - FTP (File Transfer Protocol), 634-635
  - organizing, 630
    - default index files, 632-633
    - directories, 632

- filenames, 632-633
- Web server setup, 630-631
- text
  - aligning images with, 207-209
  - wrapping around images, 210-212
- text formatting, design tips, 583-584
- HTML elements, 45
- HTML forms. *See* forms
- HTML (Hypertext Markup Language), 21
  - page layout, history, 372-373
- HTML Kit, 27
- HTML pages
  - creating, 46-47
    - viewing results, 47-48
  - links, creating, 91-93
  - local pages, linking, 95-98
  - text formatting, 49
- HTML tags, 43, 58-59, 68, 86
  - `<a>`
    - accesskey attribute, 615
    - tabindex attribute, 615
    - target attribute, 557-562
    - title attribute, 615
  - `<abbr>`, as logical style tags, 123
  - accessibility, 619
  - `<address>`, 598
  - `<area>`, 230-231
  - `<audio>`, 413-416
  - `<b>` tags, 124
  - `<base>`, 562-563, 571
  - `<button>`, form controls, 340
  - `<body>`, 59-60, 172
  - `<br>`, 133-134
  - `<br>`, clear, 212
  - brackets, 45
  - `<button>`, 366
  - `<caption>`, 247-251, 283
  - character entities, 138
    - named entities, 140-141
    - numbered entities, 141
    - reserved characters, 141
  - character-level elements, 122
  - `<cite>` tags, as logical style tags, 123
  - closing, 45, 60, 72
  - `<code>` tags, as logical style tags, 123
  - `<col>`, 276-277, 283
  - `<colgroup>` tag, 276-277, 283
  - comments, 65-66, 69
  - conventions, founding of, 44
  - CSS (cascading style sheets), combining with, 51
  - `<dd>`, 72, 81, 86
  - `<dfn>` tags, as logical style tags, 123
  - `<dir>`, 72
  - `<dl>`, 81, 86
  - DOCTYPE identifiers, 58
  - `<dt>`, 72, 81, 86
  - `<em>`, 102, 122
  - `<embed>` tag, 404-405, 416
  - `<fieldset>`, 366
  - `<form>`, 366
  - `<head>`, 59
  - heading levels, 62-64
    - style attributes and, 51
  - `<hr>`, 130-131
    - attributes, 132-133
  - `<html>`, 58-59
  - `<i>` tags, 124
  - `<iframe>`, 563-565, 571
  - `<img>`, 200-201, 237
    - align attribute, 209
    - alt attribute, 201, 616
    - height attribute, 219
    - hspace attribute, 213
    - src attribute, 201

- usemap attribute, 231-232
- width attribute, 219
- indented code, 62
- `<input>`, 366-367
- `<kbd>` tags, as logical style tags, 123
- `<label>`, 316, 366
  - displaying label elements, 323-324
- `<legend>`, 366
- `<li>`, 72, 86
  - unordered lists, 78
  - type attributes and, 76-77
  - value attributes and, 77
- `<link>`, 161
- lists
  - nesting lists, 82-84
  - numbered lists, 73-77
  - overview, 72-73
  - unordered lists, 78-80
- logical style tags, 122-123
- `<map>`, 230-231
- markup languages, 45
- nested, 59
- nesting, 95
- `<object>`, 400-404, 412-413, 416
- `<ol>`, 73, 86
  - type attributes and, 74-75
- opening, 45
- `<option>`, 366
- origins, 52
- `<p>`, 416
- paragraph tags, 64-65
- `<param>`, 417
- physical style tags, 124
- `<pre>` tags, 128-130
- `<samp>` tags, as logical style tags, 123
- sample pages, creating, 66-67, 143-144
  - adding content, 145
  - adding tables of content, 146
  - Bookworm Web page code, 151-153
  - frameworks, 144-145
  - link menus, 148-149
  - page descriptions, 147
  - planning the page, 144
  - signatures, 150-151
  - testing results, 153-154
  - unordered lists, 150
- `<script>`, JavaScript, 473-474
- `<select>`, 366
- `<small>` tags, 124
- `<source>`, 417
- `<span>` tags, 125
- `<strong>` tags, as logical style tags, 122
- `<style>`, type attribute, 160
- `<sub>` tags, 125
- `<sup>` tags, 125
- `<table>` tags, 243, 283
  - align attribute, 263
  - border attribute, 253-256
  - cellpadding attribute, 256
  - cellspacing attribute, 257
  - width attribute, 253
- `<tbody>`, 281-283
- `<td>`, 244, 284
  - colspan attribute, 267
  - rowspan attribute, 267
  - valign attribute, 264
- `<textarea>`, 341-342, 366
- `<tfoot>`, 280, 283
- `<th>`, 244, 284, 614
  - colspan attribute, 267
  - rowspan attribute, 267
- `<thead>`, 279
- `<thread>`, 283
- `<title>`, 60-61
- `<tr>`, 244, 284
- `<u>` tags, 124
- `<ul>`, 72, 86. *See also* unordered lists

- `<var>` tags, 123
  - `<video>`, 396-400, 412-413, 417
  - HTML5, 53
    - controls, 328-329
    - date and time types, 330-331
    - page layout, 373-374
      - articles, 378
      - asides, 379
      - footers, 376
      - headers, 376
      - implicit sectioning, 380
      - navigation, 377-378
      - sections, 375, 379
    - structural elements, polyfill scripts, 381
  - HTTP
    - status codes, 645
    - URLs, 113-114
  - HTTP servers. *See* Web servers
  - hyperlinks, images, 214-218
  - hypertext, 8. *See also* links
  - Hypertext Markup Language (HTML). *See* HTML (Hypertext Markup Language)
  - hypertext references. *See* href attributes
  - Hypertext Transfer Protocol (HTTP), 111
- I**
- `<i>` tag, 124, 154
  - icons, navigation icons, 215-216, 218
  - `<img>` tag, 200-201
    - height attributes, 592
  - `<input>` tag, 334, 366-367
    - checked attribute, 335
      - radio, 336
    - form control creation, 324
      - check box controls, 334-335
      - file upload controls, 339-340
      - generic buttons, 337-338
      - hidden form fields, 338-339
      - password controls, 332-333
      - radio buttons, 336
      - reset buttons, 334
      - submit buttons, 333-334
      - submit buttons, images as, 337
      - text controls, 325-326
    - type attribute
      - button, 338
      - file, 339
      - hidden, 338
      - radio, 336
    - value attribute, 326, 334
  - id attribute, overview, 163-164
  - IDs (CSS)
    - naming conventions, 195
    - overview, 163-164
  - IE (Internet Explorer). *See* Microsoft Internet Explorer
  - if statement (JavaScript), 482-484
  - `<iframe>` tag, 563-565, 571
  - image attribute, `<input>` tag, 367
  - imagemaps
    - client-side imagemaps, 226
      - `<area>` tag, 230-231
        - associating with images, 231
        - circle coordinates, 229
        - image selection, 226
      - imagemap creation software, 228
      - jukebox image example, 232-236
      - `<map>` tag, 230-231
        - polygon coordinates, 229
        - rectangle coordinates, 230
        - troubleshooting, 238
    - coordinates, determining, 227-228
    - server-side imagemaps, 226
  - images, 198
    - adding to Web pages
      - Halloween House Web page example, 202-205
      - `<img>` tag, 200-201

- aligning with text, 207-209
- as links, 214-218
- background images, 221, 224
- bullets, 224-225
- client-side imagemaps, 226
  - <area> tag, 230-231
  - associating with images, 231
  - circle coordinates, 229
  - image selection, 226
  - imagemap creation software, 228
  - jukebox image example, 232-236
  - <map> tag, 230-231
  - polygon coordinates, 229
  - rectangle coordinates, 230
  - troubleshooting, 238
- coordinates, getting from browsers, 233
- GIF (Graphics Interchange Format), 198-199, 238
- Halloween House Web page example, 202-205
- height/width, 219
- image etiquette, 237
- in mobile web design
  - reducing size of, 591-592
  - when to use, 591
- inline images, 200-201, 205-207
- JPEG (Joint Photographic Experts Group), 199, 238
- making responsive, 460-463
- navigation icons, 215-218
- PNG (Portable Network Graphics), 199
- scaling, 219
- server-side imagemaps, 226
- sizing, 220
- spacing around, 213
- submit buttons, creating with, 337
- troubleshooting, 636-637
- when to use, 591
- wrapping text around, 210-212
- writing accessible HTML, 616
- <img> tag, 237
  - align attribute, 209
  - alt attribute, 201, 616
  - CSS backgrounds, 224
  - height attribute, 219
  - hspace attribute, 213
  - src attribute, 201
  - usemap attribute, 231-232
  - vspace attribute, 213
  - width attribute, 219
- implicit navigation links, 589
- implicit sectioning, 380
- importance of RWD, 445
- improving
  - scannability, 582
  - SEO, 694
- includes, PHP, 678-680
  - choosing includes, 680-681
- indented HTML code, 62
- index files, 632
- indexes
  - PHP arrays, 656
  - of websites, 643
- inequality operator (!=), 483
- inline frames, 563-566
- <input> tag, 316-317
- input types, date, 331
- Inspector, Developer Tools, 31-32
- installing Google Analytics, 646-647
- interactivity of Web, 12
- Internet, media types, 402
- Internet Explorer. See Microsoft Internet Explorer
- Internet service providers (ISPs), 628
- intranets, 9, 24
- IrfanView, 198
- is\_array() function, 658
- ISPs (Internet service providers), 628, 649
- italic text, design tips, 583-584
- iterations, loops, 484



**J**

- Java Enterprise Edition, 684
- JavaScript, 11, 21, 472
  - advantages of, 472, 495
  - alert message, 494
  - anatomy of scripts, 474
  - arguments, 486
  - arrays, 489
  - browsers
    - compatibility, 496
    - integration, 473
  - comparison operators, 483
  - compressing for mobile web, 594-595
  - control structures, 482-486
  - data types, 488-489
  - development environments, 495
  - document object, 491
  - documentation, 495
  - DOM, accessing, 543
  - ease of use, 473
  - elements, hiding/showing, 538-546
  - environment, 491
  - event-driven models of execution, 472
  - event handlers, 492
  - events, 491-495
  - expressions, 479-480
  - for loops, 484
  - form validation, 528
    - code listing, 534-535
    - onsubmit event handler, 529
    - this form object, 532
  - forms, required fields, 531-533
  - functions, 486-488
    - anonymous functions, 542
  - history object, 491
  - if statement, 482-484
  - libraries, 500
    - enabling users, 526
  - jQuery, 500-525
    - slowly loading pages, 525
  - linked windows, opening, 567-571
  - location object, 491
  - looping statements, 484-486
  - objects, 489-490
  - operators, 479-480
  - parameters, 486
  - placing in external files, 593-594
  - reserved words, 481
  - <script> tag, 473-474
    - src attribute, 474
  - server efficiency, 473
  - syntax, 476-496
  - variables, 480-482
  - Web pages, adding content to, 546-550
  - while loops, 485-486
  - window object, 491
- JavaScript Console, 475
- JavaScript Debugger, 475
- JPEG (Joint Photographic Experts Group), 199, 238
- jQuery, selectors, 503
- jQuery JavaScript library, 500-501
  - adding/removing content, 514-518
  - AJAX, 520-525
  - attributes, manipulating, 513-514
  - binding events, 504-505
  - classes, manipulating, 508-511
  - enabling users, 526
  - form values, manipulating, 511-512
  - hiding and showing elements, 505-506
  - modifying page styles, 505-508
  - modifying web page content, 508-518
  - retrieving style sheet properties, 507-508
  - sample script, 501-503

- slowly loading pages, 525
- special effects, 518-520

jukebox imagemap, 232-236

JW Player, 408-410, 415

## K

<kbd> tag, 154

- as logical style tags, 123

keywords, 693, 700

- density, 693
- overoptimization, 693

Komodo Edit, 27

Konqueror, 17

## L

<label> tag, 316, 366

- displaying label elements, 323-324

layout

- absolute positioning, 426-428
- colors/fonts, 429-430
- CSS, positioning, 288-306
- fixed, 426
- floated columns, 433-434
- HTML sections, 423-426
- in HTML5, 373-374
  - articles, 378
  - asides, 379
  - footers, 376
  - headers, 376
  - implicit sectioning, 380
  - navigation, 377-378
  - polyfill elements, 381
  - sections, 375, 379
- liquid, 427
- measurements, 440
- redesigning, 432-433

- reordering sections, 431
- RWD, adjusting, 454-459
- style sheet, 426-430
- tables, 422-423

layout style sheets, writing, 426-433

left property

- absolute positioning, 293
- CSS positioning, 288

<legend> tag, 366

- grouping controls, 350-351

length units (CSS), 167

- overview, 166

less than operator (<), 483

less than or equal to operator (<=), 483

less than symbol (<), troubleshooting, 142

<li> tags, 72, 86

- type attributes and, 76-77
- value attributes and, 77

libraries (JavaScript), 500

- jQuery, 500-501
  - adding/removing content, 514-518
  - AJAX, 520-525
  - binding events, 504-505
  - enabling users, 526
  - hiding and showing elements, 505-506
  - manipulating attributes, 513-514
  - manipulating classes, 508-511
  - manipulating form values, 511-512
  - modifying page styles, 505-508
  - modifying web page content, 508-518
  - retrieving style sheet properties, 507-508
  - sample script, 501-503
  - slowly loading pages, 525
  - special effects, 518-520

line breaks. See <br> tags

line feeds, 634

link attribute (<body> tag), 172

link farms, 696

- link menus, 583
  - creating, 101-103, 148-149
- `<link>` tag, 117, 161
- linked windows, 557
  - accessibility, 619
  - `<base>` tag, 562-563
  - creating, 560-562
  - JavaScript, opening with, 567-571
- links, 117-119. *See also* hypertext
  - active state, 173
  - anchors, 104
    - creating, 105
    - hash signs and, 108
    - linking in same page, 110
    - sample exercises, 105-109
  - canonical links, 698
  - creating, 90
    - `<a>` tags, 90-91
    - sample exercise, 91, 95
  - CSS, modifying with, 173-174
  - hover state, 173
  - images, 214-218
  - in mobile web design, 587-588
    - definition links, 590
    - explicit navigation links, 589
    - “here” syndrome, 588
    - implicit navigation links, 589
    - link menus, 587
    - tangent links, 590
    - when to use, 589-591
  - link menus, creating, 101-103
  - local pages, pathnames, 95-98
  - navigation links, 583
  - paying for, 700
  - remote pages, linking to, 98
    - sample exercise, 99-100
  - to home page, 597
  - troubleshooting, 637
  - Web pages, adding to, 547-548
    - writing accessible HTML, 615
- Links Web browser, 17
- Linode, 630
- liquid layouts, 427
- list-style-image properties, 87
  - unordered lists, 80
- list-style-position properties, unordered lists, 80
- list-style-position property, 87
- list-style property, 87
- list-style-type property, 87
- listings
  - 15.1 (Using `<div>` Tags to Create Sections for Positioning), 424-425
  - 15.2 (Style Sheet for Page Layout), 427
  - 15.3 (Style Sheet for Colors and Fonts), 429
  - 15.4 (Moving One Section Before Another), 431
  - 15.5 (Float-Based Layouts in CSS), 433-434
  - 15.6 (Randomly Organized Style Sheet), 437
  - 15.7 (Better-Organized Style Sheet), 438
- lists, 72-73
  - customizing numbered lists, 75-77
  - HTML tags for, 72-73
    - numbered lists, 73
  - importance of, 582
  - nesting lists, 82-84
  - ordered lists, 73
    - customizing, 74-77
  - other uses for, 84-85
  - unordered lists, 78
    - customizing, 78-80
- Little Brother, 10-11
- loading external data, AJAX, 521-525
- local pages, linking (pathnames), 95-98
- location object (JavaScript), 491
- log files, 645
  - accessing, 644

**logical style tags, 122-123**

- `<abbr>` tags. See `<abbr>` tags
- `<cite>` tags. See `<cite>` tags
- `<code>` tags. See `<code>` tags
- `<dfn>` tags. See `<dfn>` tags
- `<em>` tag, 122
- `<kbd>` tags. See `<kbd>` tags
- `<samp>` tags. See `<samp>` tags
- `<strong>` tags. See `<strong>` tags
- `<var>` tags. See `<var>` tags

**loop attribute**

- `<audio>` tag, 414
- `<video>` tag, 397

**loops**

- for loops, 484
- iterations, 484
- JavaScript, 484-486
- PHP, 662
  - controlling loop execution, 665-666
  - for, 664
  - foreach, 663
  - while and do...while loops, 664-665
- while loops, 485-486

**Lynx, 17**

## M

**magic target names, 557****mailto URLs, 19, 116****management, file management, 626****managing SEO**

- canonical links, 698
- robots.txt file, 697-698
- sitemaps, 696

**manipulating classes, jQuery, 508-511**

- `<map>` tag, 230-231
  - name attribute, 230
- Mapedit program, 228

**margins, CSS box model, 177-182****marketing. See advertising websites****markup languages. See HTML tags****maxlength attribute (`<input>` tag), 325, 333****measurement units (CSS), 166**

- length units, 166-167
- percentage units, 168

**measurements, 440****media attribute, `<link>` tag, 161****media queries, 450**

- breakpoints, 452-453
- media features, 451-452
- media types, 450-451
- style sheets, building, 454

**media types (Internet), 402****menus**

- creating
  - with `<option>` tag, 342-345
  - with `<select>` tag, 342-345
- drop-down menus, creating, 306-311
- link menus, creating, 101-103

**meter gauges, 357-359**

- styling, 359

**method attribute (`<form>` tag), 315-316, 321, 366**

- get method, 315-316
- get/post methods, 321-322, 640
- post method, 315-316

**methods**

- DOM, accessing, 549
- `getElementsByClassName()`, 500

**Microsoft ASP.NET, 683****Microsoft Bing, 691****Microsoft Internet Explorer, 15****middle alignment (images), 207****minus sign (-), subtraction operator, 479****mm unit (CSS), 167**

## mobile web

- compressing CSS and JavaScript, 594-595
- CSS, placing in external files, 593-594
- designing for, 446
- features
  - click to call, 596
  - geolocation, 595
  - SMS, 597
- hardware considerations, 592
- HTML5, head element, 594
- images
  - reducing size of, 591-592
  - when to use, 591
- JavaScript, placing in external files, 593-594
- links, 587
  - definition links, 590
  - explicit navigation links, 589
  - “here” syndrome, 588
  - implicit navigation links, 589
  - in body text, 588
  - link menus, 587
  - tangent links, 590
  - to home page, 597
  - when to use, 589-591
- page layout
  - consistency, 586
  - grouping related information, 586
  - headings, 585
- page signatures, 598-599
- RWD
  - best practices, 467-468
  - images, making responsive, 460-463
  - layout, adjusting, 454, 457-459
  - media queries, 450-454
  - planning for, 449-450
  - tables, making responsive, 463, 466-467
  - videos, making responsive, 460-463
- splitting topics across pages, 598

- standards compliance, 577
  - progressive enhancement, 577-578
  - validating web pages, 578-579
- tapping, 587
- videos
  - Flash, avoiding, 593
  - setting to play automatically, 592
- viewport settings, 447-449
- writing for, 581
  - avoiding browser-specific, 584
  - clarity, 582-583
  - emphasis, 584
  - page organization, 582
  - spell checking, 584-585

## Modernizr, 381

- modulus operator (%), 479
- movies, embedding (SWFObject), 406-408
- moving files between Web servers, 633
  - carriage returns/line feeds, 634
  - filename restrictions, 633
  - FTP (File Transfer Protocol), 634-635
- Moving One Section Before Another listing (15.4), 431
- MozBar, 699
- Mozilla Firefox, 16
- multimedia
  - videos
    - Flash, avoiding, 593
    - setting to play automatically, 592
  - web pages, scaling down, 622
- multiple attribute, <option> tag, 344
- multiplication operator (\*), 479
- myString function (JavaScript), 486
- myths about SEO, 695

**N**

## name attribute

- <a> tags and, 105
- <iframe> tag, 564, 571

- `<map>` tag, 230
  - `<object>` tag, 403
- named entities, 140-141
- names
  - domain names, registering, 629
  - filenames, 632-633
    - case sensitivity, 634
    - restrictions, 633
- naming colors, 170
- naming conventions, CSS classes and IDs, 195
- nav element, 377-378
- navigating DOM, node properties, 545
- navigation (forms)
  - access keys, 352
  - adding to web pages, 608-609
  - defaults, changing, 351-352
  - drop-down menus, creating, 306-311
- navigation icons, 215-218
- navigation links, 583
- nested tags, 59
- nesting
  - absolutely positioned elements, 295-297
  - lists, 82-84
  - tables, 285
  - tags, 95
- Network Solutions website, 629
- networks
  - intranet, 24
  - intranets, 9
- newspapers, online newspapers, 11-12
- node properties, DOM (navigating), 545
- nofollow attribute, 113
- non-anonymous FTP, URLs, 115
- noshade attributes, `<hr>` tags, 133
- Notepad, 26
- Notepad++, 27
- number input type, 329
- numbered entities, 141

- numbered lists
  - customizing, 75-77
  - HTML tags for, 73

## O

- object-oriented PHP, 682
- `<object>` tag, 416
  - alternative content, 403-404
  - attributes, 403
  - embedding Flash multimedia, 400-404
  - `<video>` tag, using with, 412-413
- objects
  - JavaScript, 489-490
  - this form, 532
- `<ol>` tags, 73, 86
  - type attributes and, 74-75
- `<option>` tag, 366
  - menus, creating, 342-345
  - multiple attribute, 344
  - selected attribute, 344
- Ogg Theora container format, 393
- Ogg Theora versus H.264 codec, 417
- onblur event handler, 492
- onchange event handler, 492
- onclick event handler, 492
- onfocus event handler, 493
- online books, 10
- online newspapers, 11-12
- onload event handler, 493
- onmouseover event handler, 493
- onselect event handler, 493
- onsubmit event handler, 493, 529
- onunload event handler, 493
- opening HTML tags, 45
- Opera, 17
- operators
  - JavaScript, 479-480
  - string concatenation operators, 659

- ordered lists, 73
    - customizing, 74-77
    - numbered lists, 73
  - organization, websites, 582
  - organizing
    - HTML documents, 630
      - default index files, 632-633
      - directories, 632
      - filenames, 632-633
      - Web server setup, 630-631
      - style sheets, 436-437
  - overoptimization, 693
- P**
- `<p>` tag, 416
  - padding, CSS box model, 177-182
  - padding (table cells), 256
  - page descriptions, creating, 147
  - page layout
    - absolute positioning, 426-428
    - colors/fonts, 429-430
    - floated columns, 433-434
    - HTML and, 43
    - HTML sections, 423-426
    - in HTML5, 373-374
      - articles, 378
      - asides, 379
      - footers, 376
      - headers, 376
      - implicit sectioning, 380
      - navigation, 377-378
      - polyfill elements, 381
      - sections, 375, 379
    - for mobile web
      - consistent layout, 586
      - grouping related information, 586
      - headings, 585
      - redesigning layout, 432-433
      - reordering sections, 431
      - style sheet, 426-430
      - tables, 422-423
  - page-level styles (CSS), creating, 160
  - pages. *See* Web pages
  - pages (HTML)
    - creating, 46-47
      - viewing results, 47-48
    - text formatting, 49
  - paragraphs, HTML tags for, 64-65
  - `<param>` tag, 417
  - parameters, JavaScript, 486
  - parent target name, 557
  - parsing HTML documents, 43
  - password attribute, `<input>` tag, 367
  - password controls, creating with `<input>` tag, 332-333
  - password protection, 627
  - passwords, creating for forms, 314-319
  - pathnames
    - absolute pathnames, 96-97
    - local pages, linking, 95-98
    - relative pathnames, 96-97
  - paying for search placement, 700
  - pc unit (CSS), 167
  - percent sign (%)
    - in email addresses, 116
    - modulus operator, 479
  - percentage units (CSS), 168
  - percentages, measurements, 440
  - permissions, checking, 636
  - Photoshop, 198
  - PHP, 651-654, 684-685
    - arrays, 656-659
      - associative arrays, 657
      - indexes, 656
    - built-in functions, 666
    - comments, 654-655

- conditional operators, 662
  - conditional statements, 661
  - cookies, 683
  - database connectivity, 682
  - expanding knowledge of, 681
  - file uploads, 683
  - files
    - deploying, 685
    - editing, 685
  - includes, 678-680
    - choosing includes, 680-681
  - loops, 662
    - controlling execution, 665-666
    - for, 664
    - foreach, 663
    - while and do...while loops, 664-665
  - object-oriented PHP, 682
  - processing forms, 668-669
    - parameters with multiple values, 669-670
    - presenting forms, 674-676
    - validating forms, 670-672
  - regular expressions, 682
  - running on your computer, 653-654
  - scripts, browser dependence, 685
  - sending email, 682
  - sessions, 683
  - strings, 659-661
  - user-defined functions, 666-667
    - returning values, 667-668
  - variables, 655-656
- PHP and MySQL Web Development (i), 681
- PHP interpreter, 653
- physical style tags, 124
- Pilgrim, Mark, 621
- pixels, measurements, 440
- planning for RWD, 449-450
- planning pages, 144. *See also* content
- plug-inspage attribute (<embed> tag), 405
- plus sign (+), addition operator, 479
- PNG (Portable Network Graphics), 199
- polyfill elements, 381
- polygons, imagemap coordinates, 229
- popup() function, 568-570
- pop-up windows
  - JavaScript, opening with, 567-571
  - setting up, 569
- port numbers, URLs, 111
- Portable Network Graphics (PNG), 199
- positioning (CSS), 288
  - absolute positioning, 293
    - dynamic overlays, 297-300
    - nesting elements, 295-297
    - properties, 293-295
  - fixed positioning, 301-303
  - relative positioning, 288-293
  - static positioning, 288
  - top/left/bottom/right properties, 288
  - z-index property (stacking), 303-306
- post method (<form> tag), 315-316
- post method (forms), 321-322, 640
- pound signs (#)
  - CSS and, 164
  - numbered entities, 141
- <pre> tags, 128-130, 154
- preformatted text, 128-130
- preload attribute (<audio> tag), 414
- preload attribute (<video> tag), 397
- preloading video files, 592
- preparing images for Web, 198
- presenting forms, PHP, 674-676
- preventing cross-site scripting, 669
- processing forms (PHP), 668-669
  - parameters with multiple values, 669-670
  - presenting forms, 674-676
  - validating forms, 670-672



**programs**

- form validation script, 528
  - code listing, 534-535
  - onsubmit event handler, 529
  - this form object, 532

Mapedit, 228

progress bars, 354-356

progressive enhancement, web pages, 577-578

promoting websites. *See* advertising websites

proofreading, 584-585

**properties**

for absolute positioning, 293-295

CSS (cascading style sheets)

background-position, 222

color, 172

font-family, 156

font-size, 156

font-style, 156

font-variant, 156

font-weight, 156

text-decoration, 156

list-style, 87

list-style-image, 87

list-style-position, 87

list-style-type, 87

style sheets, retrieving, 507-508

z-index property, 303-305

protocols, FTP (File Transfer Protocol), 634-635

protocols (URLs), 111

pseudo-classes, 190-193

pt unit (CSS), 167

publications, online books, 10

publishing websites

file organization, 630

default index files, 632-633

directories, 632

filenames, 632-633

Web server setup, 630-631

moving files between Web servers, 633

carriage returns/line feeds, 634

case sensitivity, 634

filename restrictions, 633

FTP (File Transfer Protocol), 634-635

Web servers, 626

authentication, 627

choosing, 627-629

file management, 626

file types, 626

media types, 626

security, 627

server-side processing, 627

server-side scripts and forms processing, 626

px unit (CSS), 167

**Q**

QuickTime, container format, 392

quotation HTML tags. *See* <blockquote> tag

quotation marks ( ' '), href attributes, 93

**R**

radio attribute, <input> tag, 367

radio buttons, creating with <input> tag, 336

Randomly Organized Style Sheet listing (15.6), 437

range input type, 329

ranking algorithms, 690

readonly form controls, creating, 352-353

rectangles, imagemap coordinates, 230

redirecting duplicate content, 698-699

reducing images for mobile web design, 591-592

referrals, 700

Register.com, 629

- registering domain names, 628-629
- registration form example (form controls), 345-348
- registration forms, 13
- regular expressions, PHP, 682
- rel attribute, 113
  - <link> tag, 161
- relative pathnames, 96
  - versus absolute pathnames, 97
  - linking local pages, 95-98
- relative positioning (CSS), 288-290, 293
- relative units (CSS), 166-167
- rem unit (CSS), 167
- remote pages, linking to, 98
  - sample exercise, 99-100
- reports, Google Analytics, 647-648
- require\_once, 680-681
- reserved characters (character entities), 141
- reserved words, JavaScript, 481
- reset attribute, <input> tag, 367
- reset buttons, creating with <input> tag, 334
- retrieving style sheet properties, JQuery, 507-508
- right alignment (images), 210-211
- right property (CSS positioning), 288
- robots, 689
- robots.txt file, 697-698
- rows. *See also* cells
  - alignment, 279-282
  - creating, 244-246
  - grouping, 279-282
  - tables, 244
- rowspan attribute, 284
  - <td> tag, 267
  - <th> tag, 267
- Ruby on Rails, 684
- RWD, 445
  - best practices, 467-468
  - history of, 444

- images, making responsive, 460-463
- importance of, 445
- layout, adjusting, 454-459
- media queries, 450
  - breakpoints, 452-453
  - media features, 451-452
  - media types, 450-451
  - style sheets, building, 454
- mobile web, viewport settings, 447-449
- planning for, 449-450
- tables, making responsive, 463, 466-467
- videos, making responsive, 460-463

## S

- Safari (Apple), 17
- <samp> tag, 155
  - as logical style tags, 123
- sample script, JQuery, 501-503
- samples
  - creating and formatting pages, 143-144
    - adding content, 145
    - adding tables of contents, 146
    - Bookworm web page code, 151-153
    - frameworks, 144-145
    - link menus, 148-149
    - page descriptions, 147
    - planning the page, 144
    - signatures, 150-151
    - testing results, 153-154
    - unordered lists, 150
  - HTML documents, creating, 47, 66-67
  - linking anchors, 105-109
  - linking remote pages, 99-100
  - link menus, creating, 101-103
  - links, creating, 91-95
  - tables
    - creating, 248-251
    - service specification table, 269-275

- Sams Teach Yourself PHP, MySQL, and Apache All in One*, 681
- sandbox attribute, 566
- sans-serif fonts, 51
- scaling images, 219
- scannability of Web pages, 582
- scanning, 582
- school servers, 627
- SCP (Secure Copy), 38
- screen magnifiers, 613
- screen readers, 613
- `<script>` tag, JavaScript, 473-474
- scripts
  - cross-site scripting, preventing, 669
  - JavaScript
    - code listing, 534-535
    - documentation, 495
    - for loops, 484
    - form validation, 528-529, 532
    - if statement, 483
    - while loops, 485-486
  - PHP, browser dependence, 685
  - server-side scripts and forms processing, 626
- search engines, 643
  - adding to web pages, 606
  - Baidu, 691
  - Bing, 691
  - choosing, 649
  - Google, 690
  - keywords, 693
  - placement, paying for, 700
  - ranking algorithms, 690
  - robots, 689
  - Yahoo!, 691
  - Yandex, 691
- Section 508, 612
- section element, 375
  - implicit sectioning, 379-380
- Secure Copy (SCP), 38
- Secure FTP (SFTP), 38
- Secure Socket Layer (SSL), 627
- security
  - authentication, 627
  - file permissions, checking, 636
  - forms, 353
  - sandbox attribute, 566
  - SSL (Secure Socket Layer), 627
  - Web servers, 627
- `<select>` tag, 366
  - menus, creating, 342-345
  - size attribute, 343
- selected attribute, `<option>` tag, 344
- selectors, 189-190
  - attribute selectors, 193-194
  - jQuery, 503
  - pseudo-classes, 190-193
- selectors (CSS), 162
  - classes, 163-164. *See also* classes (CSS)
  - contextual selectors, 162-163
  - IDs, 163-164
- self target name, 557
- semicolons [;], named entities, 140
- sending email, PHP, 682
- SEO (search engine optimization), 688
  - canonical links, 698
  - cloaking, 695
  - content, duplicate content, 696
  - duplicate content, redirecting, 698-699
  - improving, 694
  - keywords, 693
  - myths, 695
  - need for, 688
  - paying for links, 700
  - robots.txt file, 697-698
  - sitemaps, 696
  - spiders, 689
  - tracking your progress, 700

- SEO-browser.com, 699
- serifs, 51
- server efficiency, JavaScript, 473
- server-side file processing, 627
- server-side images, 226
- server-side scripts and forms processing, 626
- servers. *See* web servers
- servers (web), 24, 626
  - authentication, 627
  - choosing, 627
    - bandwidth limitations, 631
    - domain parking, 629
    - ISPs (Internet service providers), 628
    - personal servers, 629
    - school servers, 627
    - Web presence providers, 628-629
    - work servers, 627
  - file management, 626
  - file types, 626
  - media types, 626
  - moving files between, 633
    - carriage returns/line feeds, 634
    - filename restrictions, 633
    - FTP (File Transfer Protocol), 634-635
  - security, 627
  - server-side file processing, 627
  - server-side scripts and forms processing, 626
  - troubleshooting, 636
- sessions, PHP, 683
- SFTP (Secure FTP), 38
- SGML (Standard Generalized Markup Language), 42
- shape attribute (<area> tag), 230
- shapes, imagemap coordinates
  - circles, 229
  - polygons, 229
  - rectangles, 230
- showing elements, JavaScript, 538-546
- signatures
  - creating, 150-151
  - for mobile web, 598-599
  - web pages, 602
- site indexes, 643
  - Bing, 691
  - choosing, 649
  - Google, 690
  - Yahoo!, 691
- site-wide style sheets, 439
  - creating, 161-162
- sitemaps, 696
- sites. *See* websites
- size attribute
  - <hr> tags, 132
  - <input> tag, 325, 333
  - <select> tag, 343
- sizes, CSS box model (controlling), 182-185
- sizing images, 219-220
- slash (/), division operator, 479
- <small> tag, 124, 155
- smartphones, 576. *See also* mobile web
- SMS, 597
- SmugMug.com, 391
- social media, 688
  - advertising website on, 640-642
- sound, embedding, 413-416
- source code, displaying in Developer Tools, 30
- source element, <video> tag, 399-400
- <source> tag, 417
- spacing
  - images, 213
  - table cells, 257-258
- span attribute, 284
  - <colgroup> tag, 276
  - <span> tag, 125, 155
- special characters, 112, 138. *See also* character entities
- special effects, web pages, 518-520

- spell checking, 584-585
  - spiders, 689
  - splitting topics across pages, 598
  - src attribute
    - <audio> tag, 414
    - <embed> tag, 405
    - <iframe> tag, 564, 571
    - <img> tag, 201, 237
    - <script> tag, 474
    - <video> tag, 397
  - SSL (Secure Socket Layer), 627
  - stacking (CSS), 303-306
  - stand-alone web pages, designing, 583
  - Standard Generalized Markup Language (SGML), 42
  - standards, CSS, 195
  - standards compliance, websites, 578-579
  - statements
    - if statement, JavaScript, 482-484
    - looping statements, 484-486
  - static positioning (CSS), 288
  - status codes (HTTP), 645
  - stopping text wrap, 211-212
  - string concatenation operators, 659
  - strings, PHP, 659-661
  - <strong> tag, 155
    - as logical style tags, 122
  - structural elements in HTML5, polyfill scripts, 381
  - structure
    - HTML, writing with, 423-425
    - of web pages, 373-374
  - structuring HTML, 58
  - style attribute, alternatives to, 160
  - style attributes, 51
  - Style Sheet for Colors and Fonts listing (15.3), 429
  - Style Sheet for Page Layout listing (15.2), 427
  - style sheets. *See also* CSS
    - building with media queries, 454
    - creating, 50-51
    - HTML tags, combining with, 51
    - laying out, 426-430
    - properties
      - background-position, 222
      - color, 172
      - retrieving, 507-508
    - web design
      - organization, 436-437
      - site-wide, 439
    - web designs, 435-436
  - <style> tag, type attribute, 160
  - styles
    - CSS (cascading style sheets)
      - background-position, 222
      - color, 172
    - web pages, JQuery, 505-508
  - styling meter gauges, 359
  - <sub> tag, 125, 155
  - submit attribute, <input> tag, 367
  - submit buttons, creating with <input> tag, 333-334
    - images as submit buttons, 337
  - submit buttons (forms), 317
  - subtraction operator (-), 479
  - <sup> tag, 125, 155
  - SWFObject, 414-416
    - Flash movies, embedding, 406-408
  - syntax, JavaScript, 476-490
- ## T
- tabindex attribute, 351
    - <a> tag, 615
  - <table> tag, 243, 283
    - align attribute, 263
    - border attribute, 253-256

- cellpadding attribute, 256
  - cellspacing attribute, 257
  - width attribute, 253
- tables, 241-242, 282-283
- alignment, 263
    - cells, 264-265
    - rows, 279-282
  - body of, defining, 281
  - borders, changing, 253-256
  - captions, 242
    - creating, 247-251
  - cells, 242
    - alignment, 264-265
    - creating, 244-246
    - empty cells, 246
    - padding, 256
    - spacing, 257-258
    - spanning multiple rows/columns, 266-269
  - colors, background colors, 261-262
  - columns
    - aligning, 276-279
    - grouping, 276-279
    - width of, 259-260
  - creating, 242
    - sample exercise, 248-251
    - <table> tag, 243
  - footers, 280
  - headings, 242, 279-280
  - laying out, 422-423
  - making responsive, 463, 466-467
  - nesting, 285
  - rows
    - alignment, 279-282
    - creating, 244-246
    - grouping, 279-282
  - sample exercises, service specification table, 269-275
  - troubleshooting, 285
  - when to use, 282
- tables of content, adding to sample pages, 146
- tabs in Developer Tools, 30
- tags
- brackets, 45
  - closing, 45
  - CSS (cascading style sheets), combining with, 51
  - datalist, 327-328
  - HTML, 43
    - <address>, 598
    - <div>, 425
  - markup languages, 45
  - opening, 45
  - origins, 52
  - progress, 354-356
- tags (HTML)
- <a>
    - accesskey attribute, 615
    - tabindex attribute, 615
    - target attribute, 557-562
    - title attribute, 615
  - <area>, 230-231
  - <base>, 562-563, 571
  - <body>, 172
  - <br>, clear attribute, 212
  - <iframe>, 565, 571
  - <img>, 237
    - align attribute, 209
    - alt attribute, 201, 616
    - height attribute, 219
    - hspace attribute, 213
    - src attribute, 201
    - usemap attribute, 231-232
    - vspace attribute, 213
    - width attribute, 219
  - <map>, 230-231
    - name attribute, 230
  - <th>, 614
- tangent links, 590

- tapping, 587
- target attribute
  - <a> tag, 557-562
  - <base> tag, 562, 571
- target names, 557
- <tbody> tag, 281-283
- <td> tag, 244-246, 284
  - colspan attribute, 267
  - rowspan attribute, 267
  - valign attribute, 264
- Teach Yourself PHP, MySQL, and Apache All in One*, 681
- telephone numbers, validating, 329
- terms (glossary lists). See <dt> tags
- testing
  - forms, 368
  - required fields, forms, 531-533
  - results, 153-154
- text, 583. See also content
  - aligning images with, 207-209
  - character formatting, design tips, 583-584
  - links, 588
  - proofreading, 584-585
  - spell checking, 584-585
  - wrapping around images, 210-212
- text attribute
  - <body> tag, 172
  - <input> tag, 366
- text controls, creating with <input> tag, 325-326
- text-decoration properties, 126
  - CSS, 125, 156
  - font, 126-127
- text editors, 26-27
- text formatting, HTML pages, 49
- text form controls, 316
- text messaging, SMS, 597
- text wrapping, stopping, 211-212
- <textarea> tag, 366
  - form controls, 341-342
- TextEdit, 26
- texttop alignment (images), 207
- TextWrangler, 27
- <tfoot> tag, 280, 283
- <th> tag, 244, 246, 284, 614. See also tables
  - colspan attribute, 267
  - rowspan attribute, 267
  - valign attribute, 264
- <thead> tag, 279-280
- this (argument), 494
- this form object, 532
- <thead> tag, 283
- title attribute
  - <a> tag, 615
  - <link> tag, 161
  - <title> tag, 60-61
- titles, descriptive titles, 583
- tools, W3C Validator, 578-580
- top alignment (images), 207
- top levels, 97
- top property
  - absolute positioning, 293
  - CSS positioning, 288
- top target name, 558
- <tr> tag, 244, 246, 284. See also tables
- tracking SEO, 700
- transcripts, 692
- Transmit, 635
- troubleshooting
  - client-side imagemaps, 238
  - CSS, 195
  - file access, 636
  - file display errors, 638
  - images, 636-637
  - links, 637
  - tables, 285
  - Web server access, 636
  - websites, 636-638
- Tumblr.com, 37

Twitter, 641  
type attribute, 77  
    button, 338  
    checkbox, 334  
    <embed> tag, 405  
    file, 339  
    hidden, 338  
    <input> tag, 316, 325, 333, 366  
    <link> tag, 161  
    <object> tag, 403  
    radio, 336  
    <style> tag, 160  
TypePad.com, 37

## U

<u> tag, 124-155  
<ul> tags, 72, 86  
    unordered lists, 78  
uniform resource locators (URLs). See URLs  
units of measure (CSS), 166  
    length units, 166-167  
    percentage units, 168  
UNIX, conventions of, 96  
unordered lists, 78  
    creating, 150  
    customizing, 78-80  
unset() function, 657  
uploading videos  
    YouTube, 388-389  
URLs, 18-19, 110-113, 117-119  
    adding to web pages, 606-608  
    anonymous FTP, 114-115  
    CSS and, 161  
    CSS, using URLs, 168  
    directories, 112  
    file, 116-117  
    hostnames, port numbers, 111

HTTP, 113-114  
    mailto, 116  
    Mailto URLs, 19  
    non-anonymous FTP, 115  
    protocols, 111  
    special characters, 112  
usenap attribute (<img> tag), 231-232  
user-defined functions, PHP, 666-667  
    returning values, 667-668  
user preferences, determining when designing  
    web pages, 610-611  
Using <div> Tags to Create Sections for  
    Positioning listing (15.1), 424-425  
UTF-8, 140

## V

validating  
    accessibility, 620  
    forms, PHP, 670-672  
    telephone numbers, 329  
    Web pages, W3C Validator, 578-580  
validating forms with JavaScript, 528  
    code listing, 534-535  
    onsubmit event handler, 529  
    this form object, 532  
validators, Cynthia Says, 620  
valign attribute, 264  
value attribute, 77  
    <input> tag, 317, 326, 334  
values  
    em, 440  
    forms, manipulating, 511-512  
<var> tag, 155  
    as logical style tags, 123  
variables, 667  
    JavaScript, 480-482  
    PHP, 655-656  
vh unit (CSS), 167



Viddler.com, 391

video

container formats, 392-393

customizing, YouTube, 389

embedding, 386-391

Flash, 400-404

SWFObject, 406-408

<video> tag, 396-400

Flash, avoiding in mobile web design, 593

H.264 video codec, 392

converting to, 393-396

hosting, 391-395

YouTube, 387-388

making responsive, 460-463

Ogg Theora container format, 393

setting to play automatically, 592

uploading YouTube, 388-389

web pages, embedding, 386-391

video players (Flash), 408

Flowplayer, 410-412

JW Player, 408-410

<video> tag, 417

attributes, 397

embedding video, 396-400

<object> tag, using with, 412-413

source element, 399-400

VideoPress.com, 391

View Source functionalities, 66

viewport settings on mobile devices, 447-449

Vimeo, 386, 390-391, 416-417

visual styles

HTML tags, evolution of, 44

vlink attribute (<body> tag), 172

vspace attribute (<img> tag), 213

## W

W3 Validator, 578-580

W3C (World Wide Web Consortium), 20, 621

web, 577

control of, 20

cross-platform compatibility, 9

decentralized nature of, 9-10

dynamic nature of, 10-11

hypertext, 8

interactivity, 12

W3C (World Wide Web Consortium), 20

Web Accessibility Initiative (W3C), 621

Web browsers, 14

Camino, 17

exclusive content development, 14

FTP servers, navigating, 114

Google Chrome, 15

as HTML formatters, 43

Konqueror, 17

Links, 17

Lynx, 17

Microsoft Internet Explorer, 15

Mozilla Firefox, 16

multiple browser usage, advantages of, 15

Opera, 17

Safari, 17

websites, standards compliance, 578-579

web builders, 629

web design

style sheets, 435-436

organization, 436-437

site-wide, 439

Web documents. See Web pages

web hosting, 36

choosing, 627-630

commercial, 628

content-management applications, 36-37

free, 630

setting up, 37-38

web pages

accessibility, 620-621

audio, embedding, 413-416

- colors, naming, 170
- content, adding to, 546-550
- creating progressive enhancement, 577-578
- design, 581
  - brevity, 582
  - browser-specific terminology, 584
  - clarity, 582
  - consistent layout, 586
  - emphasis, 583-584
  - grouping related information, 586
  - headings, 585
  - images, 591
  - organizing for quick scanning, 582
  - page validation, 578-580
  - proofreading, 584-585
  - spell checking, 584-585
  - splitting topics across pages, 598
  - stand-alone pages, 583
  - standards compliance, 578-579
- designing
  - accessibility. *See* accessibility
  - determining user preferences, 610-611
  - emphasis, 619
  - for first-time users versus regular users, 609
  - images, 237
  - navigation, 608-609
  - search engines, 606
  - URLs, 606-608
- documents, converting to, 601
- elements, hiding/showing, 538-546
- Flash multimedia, embedding, 406-408
- forms, validating with JavaScript, 528-529, 535
- frames, 555-556
  - inline frames, 563-566
  - magic target names, 557
- framesets, 557
  - <base> tag, 562-563
  - creating, 560-562
- Halloween House example, 202-205
- images
  - adding, 200-205
  - aligning with text, 207-209
  - as links, 214-218
  - background images, 221, 224
  - bullets, 224-225
  - GIF (Graphics Interchange Format), 198-199, 238
  - Halloween House web page example, 203-205
  - height/width, 219
  - image etiquette, 237
  - inline images, 200-201, 205-207
  - JPEG (Joint Photographic Experts Group), 199, 238
  - navigation icons, 215-218
  - PNG (Portable Network Graphics), 199
  - preparing for Web, 198
  - scaling, 219
  - spacing around, 213
  - wrapping text around, 210-212
- links, adding to, 547-548
- modifying content, JQuery, 508-518
- modifying styles, JQuery, 505-508
- multimedia, scaling down, 622
- remote pages, 98
- signatures, 602
- slowly loading, JavaScript, 525
- special effects, JQuery, 518-520
- structure, 373-374
- text formatting, design tips, 583-584
- URLs, 110-113
- validating, W3 Validator, 578-580
- video
  - embedding, 386-400
  - hosting, 391-395
- Web presence providers, 628-629
- Web publishing, text editor, 26-27

- web servers, 18, 24, 626
  - authentication, 627
  - choosing, 627
    - bandwidth limitations, 631
    - domain parking, 629
    - ISPs (Internet service providers), 628
    - personal servers, 629
    - school servers, 627
    - Web presence providers, 628-629
    - work servers, 627
  - directories, index files, 632
  - file management, 626
  - file types, 626
  - FTP servers, navigating, 114
  - log files, 644
  - media types, 626
  - moving files between, 633
    - carriage returns/line feeds, 634
    - filename restrictions, 633
    - FTP (File Transfer Protocol), 634-635
  - security, 627
  - server-side file processing, 627
  - server-side scripts and forms processing, 626
  - troubleshooting, 636
- weblogs, 630
- WebM container format, 393
- webmasters, 630
- Webmaster Tools, 696
- websites, 10, 24, 40
  - accessibility, 620-622
  - Adobe, 198
  - advertising, 638
    - brochures, 644
    - business cards, 644
    - links from other sites, 639
    - social media, 640-642
  - anatomy of, 24-26
  - Bing, 691
  - Building Accessible Websites, 621
  - change to websites, 24
  - content, wireframing, 33-36
  - content ideas, 33
  - CuteFTP, 635
  - design, 581
    - brevity, 582
    - browser-specific terminology, 584
    - clarity, 582
    - consistent layout, 586
    - emphasis, 583-584
    - grouping related information, 586
    - headings, 585
    - images, 591
    - organizing for quick scanning, 582
    - page validation, 578-580
    - proofreading, 584-585
    - spell checking, 584-585
    - splitting topics across pages, 598
    - stand-alone pages, 583
    - standards compliance, 578-579
  - Dive Into Accessibility, 621
  - external files, 622
  - Fetch, 635
  - forms, 13, 528-529, 535
  - FTP Explorer, 635
  - Google, 690
  - hits, 638
  - home pages, 24
  - initial visit, 25
  - IrfanView, 198
  - links, 587
  - log files, 645
  - Network Solutions, 629
  - page validation, W3 Validator, 578-580
  - publishing
    - file organization, 630-633
    - moving files between Web servers, 633-635

- server-side scripts and forms processing, 626
  - Web servers, 626-629
  - search engines, 643
    - Bing, 691
    - Google, 690
    - Yahoo!, 691
  - Section 508, 612
  - Transmit, 635
  - troubleshooting, 636
    - file access, 636
    - file display errors, 638
    - images, 636-637
    - links, 637
    - server access, 636
  - URLs, 110-113
  - W3C Validator, 579
  - W3C Web Accessibility Initiative, 621
  - web hosting, 36
    - content-management applications, 36-37
    - setting up, 37-38
  - web pages, 24
  - Yahoo!, 691
  - WHATWG, 53
  - when to use images, 591
  - when to use links, 589-591
  - while and do...while loops, PHP, 664-665
  - while loops, 486
    - JavaScript, 485-486
  - width
    - of images, 219
    - table columns, 259-260
  - width attributes
    - <embed> tag, 405
    - <iframe> tag, 564, 571
    - <img> tag, 219
    - <hr> tags, 132
    - <object> tag, 403
    - <table> tag, 253
    - <video> tag, 397
  - width property for absolute positioning, 294-295
  - window object (JavaScript), 491
  - windows, linked windows, 557
    - <base> tag, 562-563
    - creating, 560-562
  - Windows, Notepad, 26
  - wireframing, 40
  - WordPress.com, 37
  - work servers, 627
  - World Wide Web. *See* web
  - World Wide Web Consortium (W3C), 20
  - wrapping text around images, 210-212
  - writeParagraph function (JavaScript), 486
  - writing
    - accessible HTML, 613
      - images, 616
      - links, 615
    - articles, guest posting, 639-640
    - for mobile web, 581
      - avoiding browser-specific, 584
      - clarity, 582-583
      - emphasis, 584
      - page organization, 582
      - spell checking, 584-585
  - WYSIWYG (What You See Is What You Get)
    - versus HTML, 42
- ## X
- XAMPP, 653
  - XHTML, 52
    - <hr> tags, converting to XHTML, 131
  - XHTML 1.0, 58. *See also* HTML tags
    - <br> tags, converting, 134
  - XML (eXtensible Markup Language), 52

**Y**

Yahoo!, 691

Yandex, 691

yellow\_page window, 558-562

YouTube, 386, 416-417

- customizing videos on, 389

- embedding video, 386-391

- hosting videos on, 387-388

- uploading videos to, 388-389

**Z**

z-index property (CSS), 303-306

*This page intentionally left blank*