Julie C. Meloni

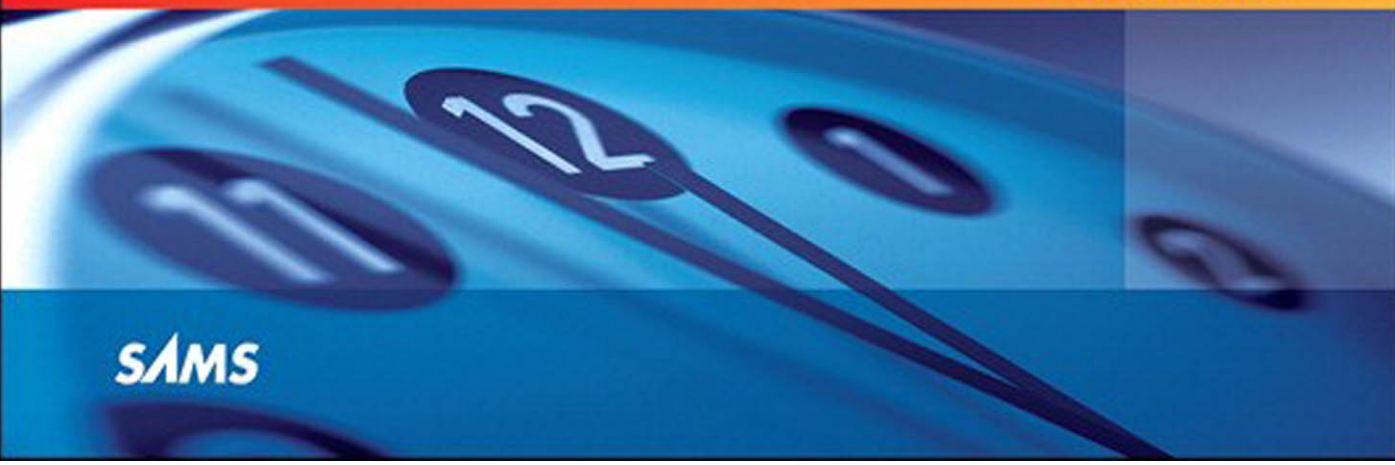
SECOND EDITION

Covers HTML5, CSS3 and jQuery

Sams Teach Yourself

HTML, CSS and JavaScript

All in One



SAMS

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Julie C. Meloni

Sams **Teach Yourself**

HTML, CSS and JavaScript

All in One

SECOND EDITION

SAMS 800 East 96th Street, Indianapolis, Indiana, 46240 USA

Sams Teach Yourself HTML, CSS and JavaScript All in One, Second Edition

Copyright © 2015 by Pearson Education

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: 978-0-672-33714-7
ISBN-10: 0-672-33714-2

Library of Congress Catalog Card Number: 2014945244

Printed in the United States of America

First Printing: October 2014

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 author 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 or from the use of the CD or programs accompanying it.

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 or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact international@pearsoned.com.

Acquisitions Editor
Mark Taber

Managing Editor
Kristy Hart

Project Editor
Elaine Wiley

Copy Editor
Cheri Clark

Indexer
Ken Johnson

Proofreader
Debbie Williams

Technical Editor
Phil Ballard

Publishing Coordinator
Vanessa Evans

Designer
Gary Adair

Cover Designer
Mark Shirar

Senior Compositor
Gloria Schurick

Contents at a Glance

PART I: **Getting Started on the Web**

CHAPTER 1: Understanding How the Web Works

CHAPTER 2: Structuring an HTML Document

CHAPTER 3: Understanding Cascading Style Sheets

CHAPTER 4: Understanding JavaScript

CHAPTER 5: Validating and Debugging Your Code

PART II: **Building Blocks of Practical Web Design**

CHAPTER 6: Working with Fonts, Text Blocks, Lists, and Tables

CHAPTER 7: Using External and Internal Links

CHAPTER 8: Working with Colors, Images, and Multimedia

PART III: **Advanced Web Page Design with CSS**

CHAPTER 9: Working with Margins, Padding, Alignment, and Floating

CHAPTER 10: Understanding the CSS Box Model and Positioning

CHAPTER 11: Using CSS to Do More with Lists, Text, and Navigation

CHAPTER 12: Creating Fixed or Liquid Layouts

PART IV: **Getting Started with Dynamic Sites**

CHAPTER 13: Understanding Dynamic Websites and HTML5 Applications

CHAPTER 14: Getting Started with JavaScript Programming

CHAPTER 15: Working with the Document Object Model (DOM)

CHAPTER 16: Using JavaScript Variables, Strings, and Arrays

CHAPTER 17: Using JavaScript Functions and Objects

CHAPTER 18: Controlling Flow with Conditions and Loops

CHAPTER 19: Responding to Events

CHAPTER 20: Using Windows

PART V: **Advanced JavaScript Programming**

CHAPTER 21: JavaScript Best Practices

CHAPTER 22: Using Third-Party JavaScript Libraries and Frameworks

CHAPTER 23: A Closer Look at jQuery

CHAPTER 24: First Steps Toward Creating Rich Interactions with jQuery UI

CHAPTER 25: AJAX: Remote Scripting

PART VI: Advanced Website Functionality and Management

CHAPTER 26: Working with Web-Based Forms

CHAPTER 27: Organizing and Managing a Website

Table of Contents

CHAPTER 1: **Understanding How the Web Works** **1**

A Brief History of HTML and the World Wide Web ... 1
Creating Web Content ... 2
Understanding Web Content Delivery ... 3
Selecting a Web Hosting Provider ... 6
Testing with Multiple Web Browsers ... 8
Creating a Sample File ... 10
Using FTP to Transfer Files ... 10
Understanding Where to Place Files on the Web Server ... 15
Distributing Content Without a Web Server ... 19
Tips for Testing Web Content ... 20
Summary ... 21
Q&A ... 22
Workshop ... 23

CHAPTER 2: **Structuring an HTML Document** **25**

Getting Prepared ... 25
Getting Started with a Simple Web Page ... 26
HTML Tags Every Web Page Must Have ... 30
Organizing a Page with Paragraphs and Line Breaks ... 32
Organizing Your Content with Headings ... 34
Understanding Semantic Elements ... 36
Using `<header>` in Multiple Ways ... 43
Understanding the `<section>` Element ... 44
Using `<article>` ... 45
Implementing the `<nav>` Element ... 46
When to Use `<aside>` ... 47
Using `<footer>` Effectively ... 49
Summary ... 50
Q&A ... 51
Workshop ... 52

CHAPTER 3: **Understanding Cascading Style Sheets** **55**

How CSS Works ... 56
A Basic Style Sheet ... 57
A CSS Style Primer ... 62
Using Style Classes ... 67
Using Style IDs ... 69
Internal Style Sheets and Inline Styles ... 70
Summary ... 73
Q&A ... 74
Workshop ... 75

CHAPTER 4: **Understanding JavaScript** **77**

Learning Web Scripting Basics ... 77
How JavaScript Fits into a Web Page ... 79
Exploring JavaScript's Capabilities ... 82
Displaying Time with JavaScript ... 84
Testing the Script ... 87
Summary ... 91
Q&A ... 92
Workshop ... 93

CHAPTER 5: **Validating and Debugging Your Code** **95**

Validating Your Web Content ... 95
Debugging HTML and CSS Using Developer Tools ... 98
Debugging JavaScript Using Developer Tools ... 111
Summary ... 117
Q&A ... 118
Workshop ... 118

CHAPTER 6: **Working with Fonts, Text Blocks, Lists, and Tables** **121**

Working with Special Characters ... 122
Boldface, Italics, and Special Text Formatting ... 125
Tweaking the Font ... 128
Using Web Fonts ... 132
Aligning Text on a Page ... 134
The Three Types of HTML Lists ... 138

Placing Lists Within Lists 140
Creating a Simple Table 146
Controlling Table Sizes 150
Alignment and Spanning Within Tables 153
Page Layout with Tables 157
Using CSS Columns 158
Summary 162
Q&A 164
Workshop 165

CHAPTER 7: **Using External and Internal Links** **169**
Using Web Addresses 169
Linking Within a Page Using Anchors 172
Linking Between Your Own Web Content 175
Linking to External Web Content 178
Linking to an Email Address 179
Opening a Link in a New Browser Window 181
Using CSS to Style Hyperlinks 181
Summary 185
Q&A 186
Workshop 187

CHAPTER 8: **Working with Colors, Images, and Multimedia** **189**
Best Practices for Choosing Colors 190
Understanding Web Colors 192
Using Hexadecimal Values for Colors 194
Using CSS to Set Background, Text, and Border Colors 195
Choosing Graphics Software 198
The Least You Need to Know About Graphics 199
Preparing Photographic Images 200
Creating Banners and Buttons 206
Reducing or Removing Colors in an Image 208
Creating Tiled Background Images 209
Creating Animated Web Graphics 211
Placing Images on a Web Page 212
Describing Images with Text 214
Specifying Image Height and Width 216
Aligning Images 216
Turning Images into Links 220
Using Background Images 223
Using Imagemaps 225
Linking to Multimedia Files 231
Embedding Multimedia Files 235
Using Pure HTML5 for Audio and Video Playback 238
Additional Tips for Using Multimedia 241
Summary 242
Q&A 246
Workshop 247

CHAPTER 9: **Working with Margins, Padding, Alignment, and Floating** **249**
Using Margins 250
Padding Elements 257
Keeping Everything Aligned 261
Understanding the `Float` Property 262
Summary 266
Q&A 266
Workshop 267

CHAPTER 10: **Understanding the CSS Box Model and Positioning** **269**
The CSS Box Model 269
The Whole Scoop on Positioning 273
Controlling the Way Things Stack Up 278
Managing the Flow of Text 280
Summary 281
Q&A 282
Workshop 282

CHAPTER 11: **Using CSS to Do More with Lists, Text, and Navigation** **285**
HTML List Refresher 286
How the CSS Box Model Affects Lists 286
Placing List Item Indicators 290
Creating Imagemaps with List Items and CSS 292
How Navigation Lists Differ from Regular Lists 296

Creating Vertical Navigation with CSS ... 296
Creating Horizontal Navigation with CSS ... 307
Summary ... 311
Q&A ... 311
Workshop ... 313

CHAPTER 12: **Creating Fixed or Liquid Layouts** **315**
Understanding Fixed Layouts ... 316
Understanding Liquid Layouts ... 318
Creating a Fixed/Liquid Hybrid Layout ... 321
Considering a Responsive Web Design ... 332
Summary ... 333
Q&A ... 334
Workshop ... 334

CHAPTER 13: **Understanding Dynamic Websites and HTML5 Applications** **337**
Understanding the Different Types of Scripting ... 337
Including JavaScript in HTML ... 338
Displaying Random Content ... 340
Understanding the Document Object Model ... 345
Changing Images Based on User Interaction ... 346
Thinking Ahead to Developing HTML5 Applications ... 348
Summary ... 349
Q&A ... 350
Workshop ... 350

CHAPTER 14: **Getting Started with JavaScript Programming** **353**
Basic Concepts ... 353
JavaScript Syntax Rules ... 360
Using Comments ... 361
Best Practices for JavaScript ... 362
Understanding JSON ... 363
Summary ... 364
Q&A ... 364
Workshop ... 365

CHAPTER 15: **Working with the Document Object Model (DOM)** **367**
Understanding the Document Object Model ... 367
Using `window` Objects ... 368
Working with the `document` Object ... 369
Accessing Browser History ... 372
Working with the `location` Object ... 374
More About the DOM Structure ... 376
Working with DOM Nodes ... 378
Creating Positionable Elements (Layers) ... 380
Hiding and Showing Objects ... 385
Modifying Text Within a Page ... 387
Adding Text to a Page ... 389
Summary ... 391
Q&A ... 392
Workshop ... 392

CHAPTER 16: **Using JavaScript Variables, Strings, and Arrays** **395**
Using Variables ... 395
Understanding Expressions and Operators ... 399
Data Types in JavaScript ... 400
Converting Between Data Types ... 401
Using `String` Objects ... 402
Working with Substrings ... 405
Using Numeric Arrays ... 408
Using String Arrays ... 409
Sorting a Numeric Array ... 411
Summary ... 414
Q&A ... 415
Workshop ... 415

CHAPTER 17: **Using JavaScript Functions and Objects** **419**
Using Functions ... 419
Introducing Objects ... 425
Using Objects to Simplify Scripting ... 427
Extending Built-in Objects ... 432
Using the `Math` Object ... 434
Working with `Math` Methods ... 436

Working with Dates 438
Summary 441
Q&A 441
Workshop 442

CHAPTER 18: **Controlling Flow with Conditions and Loops** **445**
The `if` Statement 445
Using Shorthand Conditional Expressions 448
Testing Multiple Conditions with `if` and `else` 449
Using Multiple Conditions with `switch` 452
Using `for` Loops 453
Using `while` Loops 456
Using `do...while` Loops 457
Working with Loops 457
Looping Through Object Properties 459
Summary 462
Q&A 463
Workshop 463

CHAPTER 19: **Responding to Events** **465**
Understanding Event Handlers 465
Using Mouse Events 471
Using Keyboard Events 474
Using the `onload` and `onunload` Events 477
Using `onclick` to Change a `<div>`'s Appearance 478
Summary 485
Q&A 485
Workshop 486

CHAPTER 20: **Using Windows** **489**
Controlling Windows with Objects 489
Moving and Resizing Windows 493
Using Timeouts 495
Displaying Dialog Boxes 497
Summary 499
Q&A 500
Workshop 500

CHAPTER 21: **JavaScript Best Practices** **503**
Scripting Best Practices 503
Reading Browser Information 514
Cross-Browser Scripting 519
Supporting Non-JavaScript-Enabled Browsers 520
Summary 526
Q&A 526
Workshop 526

CHAPTER 22: **Using Third-Party JavaScript Libraries and Frameworks** **529**
Using Third-Party JavaScript Libraries 529
Adding JavaScript Effects Using a Third-Party Library 534
Using JavaScript Frameworks 537
Summary 539
Q&A 539
Workshop 539

CHAPTER 23: **A Closer Look at jQuery** **541**
Preparing to Use jQuery 541
Becoming Familiar with the `$(document).ready` Handler 542
Selecting DOM and CSS Content 544
Manipulating HTML Content 545
Putting the Pieces Together to Create a jQuery Animation 549
Handling Events with jQuery 553
Summary 554
Q&A 555
Workshop 555

CHAPTER 24: **First Steps Toward Creating Rich Interactions with jQuery UI** **557**
Preparing to Use jQuery UI 557
Using Selectors in jQuery UI 558
Positioning UI Elements with jQuery UI 559
Applying jQuery UI Effects 564
Using jQuery UI Widgets for Advanced Interactions 573

Where to Go from Here ... 585
Summary ... 586
Q&A ... 586
Workshop ... 587

CHAPTER 25: **AJAX: Remote Scripting** **589**
Introducing AJAX ... 589
Using `XMLHttpRequest` ... 592
Creating a Simple AJAX Library ... 594
Creating an AJAX Quiz Using the Library ... 596
Debugging AJAX-Based Applications ... 601
Using jQuery's Built-in Functions for AJAX ... 606
Summary ... 608
Q&A ... 608
Workshop ... 608

CHAPTER 26: **Working with Web-Based Forms** **611**
How HTML Forms Work ... 611
Creating a Form ... 612
Accepting Text Input ... 617
Naming Each Piece of Form Data ... 618
Labeling Each Piece of Form Data ... 618
Grouping Form Elements ... 619
Exploring Form Input Controls ... 621
Using HTML5 Form Validation ... 629
Submitting Form Data ... 631
Accessing Form Elements with JavaScript ... 633
Summary ... 635
Q&A ... 638
Workshop ... 638

CHAPTER 27: **Organizing and Managing a Website** **641**
When One Page Is Enough ... 642
Organizing a Simple Site ... 644
Organizing a Larger Site ... 648
Writing Maintainable Code ... 652
Thinking About Version Control ... 654
Using HTML and CSS Frameworks ... 657
Summary ... 658
Q&A ... 658
Workshop ... 659

INDEX 661

About the Author

Julie C. Meloni is a software development manager and technical consultant living in Washington, D.C. She has written several books and articles on web-based programming languages and database topics, including the bestselling *Sams Teach Yourself PHP*, *MySQL and Apache All in One*.

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.

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 stronger.

Please note that we cannot help you with technical problems related to the topic of this book, and that due to the high volume of mail we receive, we might not be able to reply to every message.

When you write, please be sure to include this book's title, edition number, and author, as well as your name and contact information.

Email: feedback@samspublishing.com

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

Reader Services

Visit our website and register this book at **informit.com/register** for convenient access to any updates, downloads, or errata that might be available for this book.

This page intentionally left blank

CHAPTER 4
Understanding JavaScript

The World Wide Web (WWW) began as a text-only medium—the first browsers didn't even support images within web pages. The Web has come a long way since those early days. Today's websites include a wealth of visual and interactive features in addition to useful content: graphics, sounds, animation, and video. Web scripting languages, such as JavaScript, are one of the easiest ways to spice up a web page and to interact with users in new ways.

The first part of this chapter introduces the concept of web scripting and the JavaScript language. As the chapter moves ahead, you'll learn how to include JavaScript commands directly in your HTML documents, and how your scripts will be executed when the page is viewed in a browser. You will work with a simple script, edit it, and test it in your browser, all the while learning the basic tasks involved in creating and using JavaScript scripts.

WHAT YOU'LL LEARN IN THIS CHAPTER:

- What web scripting is and what it's good for
- How scripting and programming are different (and similar)
- What JavaScript is and where it came from
- How to include JavaScript commands in a web page
- What JavaScript can do for your web pages
- Beginning and ending scripts
- Formatting JavaScript statements
- How a script can display a result
- Including a script within a web document
- Testing a script in your browser
- Modifying a script
- Dealing with errors in scripts
- Moving scripts into separate files

Learning Web Scripting Basics

You already know how to use one type of computer language: HTML. You use HTML tags to describe how you want your document formatted, and the browser obeys your commands and shows the formatted document to the user. But because HTML is a simple text markup language, it can't respond to the user, make decisions, or automate repetitive tasks. Interactive tasks such as these require a more sophisticated language: a programming language or a *scripting* language.

Although many programming languages are complex, scripting languages are generally simple. They have a simple syntax, can

perform tasks with a minimum of commands, and are easy to learn. JavaScript is a web scripting language that enables you to combine scripting with HTML to create interactive web pages.

NOTE

Interpreted languages have their disadvantages—they can't execute really quickly, so they're not ideally suited for complicated work, such as graphics. Also, they require the interpreter (in JavaScript's case, usually a browser) in order to work.

Scripts and Programs

A movie or a play follows a script—a list of actions (or lines) for the actors to perform. A web script provides the same type of instructions for the web browser. A script in JavaScript can range from a single line to a full-scale application. (In either case, JavaScript scripts usually run within a browser.)

Some programming languages must be *compiled*, or translated, into machine code before they can be executed. JavaScript, on the other hand, is an *interpreted* language: The browser executes each line of script as it comes to it.

There is one main advantage to interpreted languages: Writing or changing a script is very simple. Changing a JavaScript script is as easy as changing a typical HTML document, and the change is enacted as soon as you reload the document in the browser.

Introducing JavaScript

NOTE

A bit of history: JavaScript was originally called LiveScript and was first introduced in Netscape Navigator 2.0 in 1995. It was soon renamed JavaScript to indicate a marketing relationship with Sun's Java language, although there is no other relationship, structurally or otherwise, between Java and JavaScript.

JavaScript was developed nearly 20 years ago by Netscape Communications Corporation, the maker of the long-defunct Netscape web browser. JavaScript was the first web scripting language to be supported by browsers, and it is still by far the most popular.

JavaScript is almost as easy to learn as HTML, and it can be included directly in HTML documents. Here are a few of the things you can do with JavaScript:

- Display messages to the user as part of a web page, in the browser's status line, or in alert boxes
- Validate the contents of a form and make calculations (for example, an order form can automatically display a running total as you enter item quantities)
- Animate images or create images that change when you move the mouse over them
- Create ad banners that interact with the user, rather than simply displaying a graphic

- Detect the browser in use or its features and perform advanced functions only on browsers that support them
- Detect installed plug-ins and notify the user if a plug-in is required
- Modify all or part of a web page without requiring the user to reload it
- Display or interact with data retrieved from a remote server

You can do all this and more with JavaScript, including creating entire applications. We'll explore the uses of JavaScript throughout this book.

How JavaScript Fits into a Web Page

Using the `<script>` tag, you can add a short script (in this case, just one line) to a web document, as shown in Listing 4.1. The `<script>` tag tells the browser to start treating the text as a script, and the closing `</script>` tag tells the browser to return to HTML mode. In most cases, you can't use JavaScript statements in an HTML document except within `<script>` tags. The exception is event handlers, described later in this chapter.

LISTING 4.1 A Simple HTML Document with a Simple Script

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>The American Eggplant Society</title>
  </head>

  <body>
    <h1>The American Eggplant Society</h1>
    <p>Welcome to our site. Unfortunately, it is still
    under construction.</p>
    <p>We last worked on it on this date:
    <script type="text/javascript">
    <!-- Hide the script from old browsers
    document.write(document.lastModified);
    // Stop hiding the script -->
    </script>
    </p>
  </body>
</html>
```

JavaScript's `document.write` statement, which you'll learn more about later, sends output as part of the web document. In this case, it displays the modification date of the document, as shown in Figure 4.1.

FIGURE 4.1
Using `document.write` to display a last-modified date.

In this example, we placed the script within the body of the HTML document. There are actually four places where you might use scripts:

- **In the body of the page**—In this case, the script's output is displayed as part of the HTML document when the browser loads the page.
- **In the header of the page between the `<head>` tags**—Scripts in the header should not be used to create output within the `<head>` section of an HTML document, since that would likely result in poorly-formed and invalid HTML documents, but these scripts can be referred to by other scripts here and elsewhere. The `<head>` section is often used for functions—groups of JavaScript statements that can be used as a single unit. You will learn more about functions in Chapter 14, "Getting Started with JavaScript Programming."

- **Within an HTML tag, such as** `<body>` **or** `<form>`—This is called an *event handler* and it enables the script to work with HTML elements. When using JavaScript in event handlers, you don't need to use the `<script>` tag. You'll learn more about event handlers in Chapter 14.
- **In a separate file entirely**—JavaScript supports the use of files with the `.js` extension containing scripts; these can be included by specifying a file in the `<script>` tag. While the `.js` extension is a convention, scripts can actually have any file extension, or none.

Using Separate JavaScript Files

When you create more complicated scripts, you'll quickly find that your HTML documents become large and confusing. To avoid this problem, you can use one or more external JavaScript files. These are files with the `.js` extension that contain JavaScript statements.

External scripts are supported by all modern browsers. To use an external script, you specify its filename in the `<script>` tag:

```
<script  type="text/javascript" src="filename.js"></script>
```

Because you'll be placing the JavaScript statements in a separate file, you don't need anything between the opening and closing `<script>` tags—in fact, anything between them will be ignored by the browser.

You can create the `.js` file using a text editor. It should contain one or more JavaScript commands, and only JavaScript—don't include `<script>` tags, other HTML tags, or HTML comments. Save the `.js` file in the same directory as the HTML documents that refer to it.

TIP

External JavaScript files have a distinct advantage: You can link to the same `.js` file from two or more HTML documents. Because the browser stores this file in its cache, this can reduce the time it takes your web pages to display.

Understanding JavaScript Events

Many of the useful things you can do with JavaScript involve interacting with the user, and that means responding to *events*—for example, a link or a button being clicked. You can define event handlers within HTML tags to tell the browser how to respond to an event. For example, Listing 4.2 defines a button that displays a message when clicked.

LISTING 4.2 A Simple Event Handler

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Event Test</title>
  </head>

  <body>
    <h1>Event Test</h1>
    <button type="button"
            onclick="alert('You clicked the button.')">
            Click Me!</button>
  </body>
</html>
```

In various places throughout this book, you'll learn more about JavaScript's event model and how to create simple and complex event handlers.

Exploring JavaScript's Capabilities

If you've spent any time browsing the Web, you've undoubtedly seen lots of examples of JavaScript in action. Here are some brief descriptions of typical applications for JavaScript, all of which you'll explore further, later in this book.

Improving Navigation

Some of the most common uses of JavaScript are in navigation systems for websites. You can use JavaScript to create a navigation tool—for example, a drop-down menu to select the next page to read, or a submenu that pops up when you hover over a navigation link.

When it's done right, this kind of JavaScript interactivity can make a site easier to use, while still remaining usable for browsers that don't support JavaScript (or HTML5/CSS3, which can also be used to create great navigation).

Validating Forms

Form validation is another common use of JavaScript, although the form validation features of HTML5 have stolen a lot of JavaScript's

thunder here as well. A simple script can read values the user types into a form and make sure they're in the right format, such as with ZIP Codes, phone numbers, and e-mail addresses. This type of client-side validation enables users to fix common errors without waiting for a response from the web server telling them that their form submission was invalid. You'll learn how to work with form data in Chapter 26, "Working with Web-Based Forms."

Special Effects

One of the earliest and most annoying uses of JavaScript was to create attention-getting special effects—for example, scrolling a message in the browser's status line or flashing the background color of a page.

These techniques have fortunately fallen out of style, but thanks to the W3C DOM and the latest browsers, some more impressive effects are possible with JavaScript—for example, creating objects that can be dragged and dropped on a page, or creating fading transitions between images in a slideshow. Some developers have HTML5, CSS3, and JavaScript working in tandem to create fully functioning interactive games.

Remote Scripting (AJAX)

For a long time, the biggest limitation of JavaScript was that there was no way for it to communicate with a web server. For example, you could use JavaScript to verify that a phone number had the right number of digits, but not to look up the user's location in a database based on the number.

Now that some of JavaScript's advanced features are supported by most browsers, this is no longer the case. Your scripts can get data from a server without loading a page, or send data back to be saved. These features are collectively known as AJAX (Asynchronous JavaScript and XML), or *remote scripting*. You'll learn how to develop AJAX scripts in Chapter 25, "AJAX: Remote Scripting."

You've seen AJAX in action if you've used Google's Gmail mail application, Facebook, or any online news site that allows you to comment on stories, vote for favorites, or participate in a poll. All of these use remote scripting to present you with a responsive user interface that works with a server in the background.

Displaying Time with JavaScript

One common use of JavaScript is to display dates and times in the browser, and that's where we'll start putting some scripting pieces together. Because JavaScript runs on the browser, the times it displays will be in the user's current time zone. However, you can also use JavaScript to calculate "universal" (UTC) time.

Your script, like most JavaScript programs, begins with the HTML `<script>` tag. As you learned earlier in this chapter, you use the `<script>` and `</script>` tags to enclose a script within the HTML document.

To begin creating the script, open your favorite text editor and type the beginning and ending `<script>` tags as shown:

```
<script type="text/javascript"></script>
```

In this script, you'll use JavaScript to determine the local and UTC times and then display them in the browser. Fortunately, all the hard parts, such as converting between date formats, are built in to the JavaScript interpreter—this is one of the reasons that displaying dates and times is a good starting place for beginners.

NOTE

UTC stands for Universal Time (Coordinated), and is the atomic time standard based on the old GMT (Greenwich Mean Time) standard. This is the time at the prime meridian, which runs through Greenwich, London, England.

CAUTION

Remember to include only valid JavaScript statements between the starting and ending `<script>` tags. If the browser finds anything but valid JavaScript statements within the `<script>` tags, it will display a JavaScript error message.

Storing Data in Variables

To begin the script, you will use a *variable* to store the current date. You will learn more about variables in Chapter 16, "Using JavaScript Variables, Strings, and Arrays," but for now just understand that a *variable* is a container that can hold a value—a number, some text, or, in this case, a date.

To start writing the script, add the following line after the first `<script>` tag. Be sure to use the same combination of capital and lowercase letters in your version because JavaScript commands and variable names are case sensitive.

```
now = new Date();
```

This statement creates a variable called `now` and stores the current date and time in it. This statement and the others you will use in this script use JavaScript's built-in `Date` object, which enables you to conveniently handle dates and times. You'll learn more about working with dates in Chapter 17, "Using JavaScript Functions and Objects."

NOTE

Notice the semicolon at the end of the code snippet creating a variable called `now`. This semicolon tells the browser that it has reached the end of a statement. Semicolons are optional, but using them helps you avoid some common errors. We'll use them throughout this book for clarity.

Calculating the Results

Internally, JavaScript stores dates as the number of milliseconds since January 1, 1970. Fortunately, JavaScript includes a number of functions to convert dates and times in various ways, so you don't have to figure out how to convert milliseconds to day, date, and time.

To continue your script, add the following two statements before the final `</script>` tag:

```
localtime = now.toString();
utctime = now.toGMTString();
```

These statements create two new variables: `localtime`, containing the current time and date in a nice readable format, and `utctime`, containing the UTC equivalent.

NOTE

The `localtime` and `utctime` variables store a piece of text, such as `January 1, 2001 12:00 PM`. In programming parlance, a piece of text is called a *string*.

Creating Output

You now have two variables—`localtime` and `utctime`—which contain the results we want from our script. Of course, these variables don't do us much good unless we can see them. JavaScript includes several ways to display information, and one of the simplest is the `document.write` statement.

The `document.write` statement displays a text string, a number, or anything else you throw at it. Because your JavaScript program will be used within a web page, the output will be displayed as part of the page. To display the result, add these statements before the final `</script>` tag:

```
document.write("<p><strong>Local time:</strong> " + localtime +
"</p>");
document.write("<p><strong>UTC time:</strong> " + utctime +
"</p>");
```

These statements tell the browser to add some text to the web page containing your script. The output will include some brief strings introducing the results, and the contents of the `localtime` and `utctime` variables.

Notice the HTML elements, such as `<p>` and `<strong>`, within the quotation marks—because JavaScript's output appears within a web page, it needs to be formatted using HTML.

NOTE

Notice the plus signs (+) used between the text and variables in the `document.write()` code snippets. In this case, it tells the browser to combine the values into one string of text. If you use the plus sign between two numbers, they are added together.

Adding the Script to a Web Page

You should now have a complete script that calculates a result and displays it. Your listing should match Listing 4.3.

LISTING 4.3 The Complete Date and Time Script

```
<script type="text/javascript">
now = new Date();
localtime = now.toString();
utctime = now.toGMTString();
document.write("<p><strong>Local time:</strong> " + localtime +
"</p>");
document.write("<p><strong>UTC time:</strong> " + utctime +
"</p>");
</script>
```

To use your script, you'll need to add it to an HTML document. If you use the general template you've seen in the chapters so far, you should end up with something like Listing 4.4.

LISTING 4.4 The Date and Time Script in an HTML Document

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Displaying Times and Dates</title>
  </head>

  <body>
    <h1>Current Date and Time</h1>
    <script type="text/javascript">
     now = new Date();
     localtime = now.toString();
     utctime = now.toGMTString();
     document.write("<p><strong>Local time:</strong> "
          + localtime + "</p>");
     document.write("<p><strong>UTC time:</strong> " + utctime
          + "</p>");
   </script>
  </body>
</html>
```

Now that you have a complete HTML document, save it with an `.html` extension.

Testing the Script

To test your script, you simply need to load the HTML document you created in a web browser. If you typed the script correctly, your browser should display the result of the script, as shown in Figure 4.2. (Of course, your result won't be the same as mine, but it should be the same as the setting of your computer's clock.)

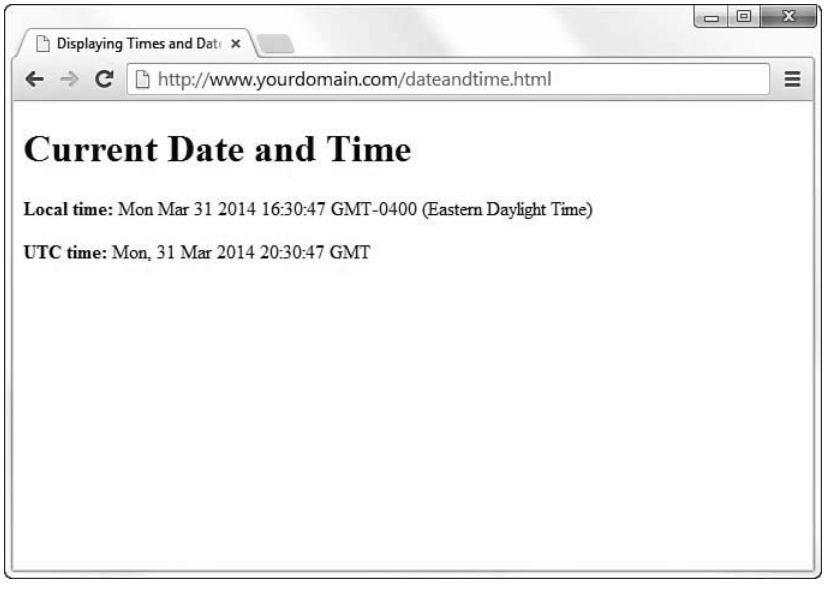


FIGURE 4.2
Using JavaScript to display the date and time.

NOTE

Notepad and other Windows text editors might try to be helpful and add the `.txt` extension to your script. Be sure your saved file has the correct extension.

A note about Internet Explorer: Depending on your security settings, the script might not execute, and your browser might display a security warning. In this case, follow your browser's instructions to allow your script to run. (This happens because the default security settings allow JavaScript in online documents, but not in local files.)

Modifying the Script

Although the current script does indeed display the current date and time, its display isn't nearly as attractive as the clock on your wall or desk. To remedy that situation, you can use some additional JavaScript features and a bit of HTML to display a large clock.

To display a large clock, we need the hours, minutes, and seconds in separate variables. Once again, JavaScript has built-in functions to do most of the work:

```
hours = now.getHours();
mins = now.getMinutes();
secs = now.getSeconds();
```

These statements load the `hours`, `mins`, and `secs` variables with the components of the time using JavaScript's built-in date functions.

After the hours, minutes, and seconds are in separate variables, you can create `document.write` statements to display them:

```
document.write("<p><strong>");
document.write(hours + ":" + mins + ":" + secs);
document.write("</p></strong>");
```

The first statement displays an HTML `<h2>` header tag to display the clock as a second-level header element. The second statement displays the `hours`, `mins`, and `secs` variables, separated by colons, and the third adds the closing `</h2>` tag.

You can add the preceding statements to the original date and time script to add the large clock display. Listing 4.5 shows the complete modified version of the script.

LISTING 4.5 The Date and Time Script with Large Clock Display

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Displaying Times and Dates</title>
  </head>

  <body>
    <h1>Current Date and Time</h1>
    <script type="text/javascript">
     now = new Date();
     localtime = now.toString();
     utctime = now.toGMTString();
     document.write("<p><strong>Local time:</strong> "
          + localtime + "</p>");
     document.write("<p><strong>UTC time:</strong> " + utctime
          + "</p>");
     hours = now.getHours();
     mins = now.getMinutes();
     secs = now.getSeconds();
     document.write("<h2>");
     document.write(hours + ":" + mins + ":" + secs);
```

```
    document.write("</h2>");
  </script>
 </body>
</html>
```

Now that you have modified the script, save the HTML file and open the modified file in your browser. If you left the browser running, you can simply use the Reload button to load the new version of the script. Try it and verify that the same time is displayed in both the upper portion of the window and the new large clock. Figure 4.3 shows the results.

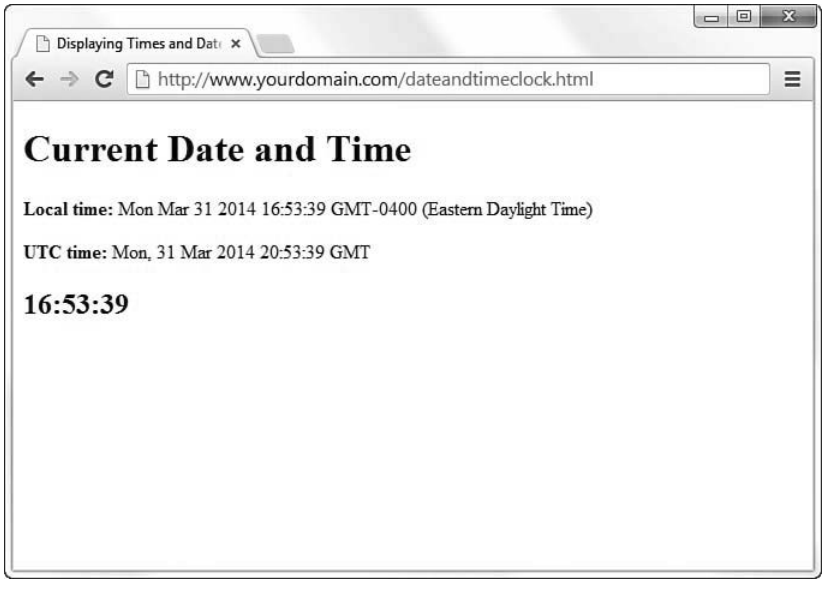


FIGURE 4.3
Displaying the modified Date and Time script.

Dealing with JavaScript Errors

As you develop more complex JavaScript applications, you're going to run into errors from time to time. JavaScript errors are usually caused by mistyped JavaScript statements.

To see an example of a JavaScript error message, modify the statement you added in the preceding section. We'll use a common error: omitting one of the parentheses. Change the last `document.write` statement in Listing 4.5 to read

```
document.write("</h2>";
```

NOTE

The time formatting produced by this script isn't perfect: Hours after noon are in 24-hour time; and there are no leading zeroes, so 12:04 is displayed as 12:4. See Chapter 17 for solutions to these issues.

Save your HTML document again and load the document into the browser. Depending on the browser version you're using, one of two things will happen: Either an error message will be displayed, or the script will simply fail to execute.

If an error message is displayed, you're halfway to fixing the problem by adding the missing parenthesis. If no error was displayed, you should configure your browser to display error messages so that you can diagnose future problems:

- In Firefox, you can also select Tools, JavaScript Console from the menu.
- In Chrome, select Tools, JavaScript Console from the Options menu. A console displays in the bottom of the browser window. The console is shown in Figure 4.4, displaying the error message you created in this example.
- In Internet Explorer, select Tools, Internet Options. On the Advanced page, uncheck the Disable Script Debugging box and check the Display a Notification About Every Script Error box. (If this is disabled, a yellow icon in the status bar still notifies you of errors.)

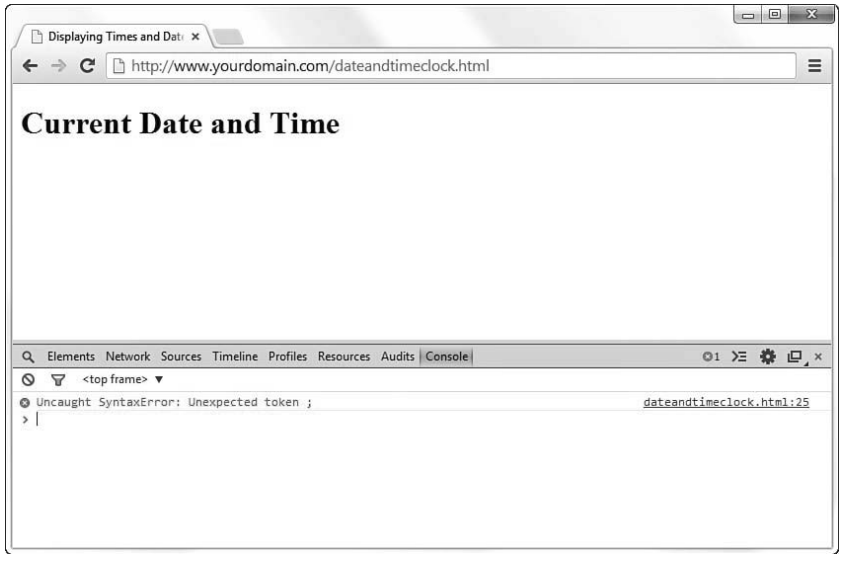


FIGURE 4.4
Showing an error in the JavaScript console in Chrome.

The error we get in this case is `Uncaught SyntaxError` and it points to line 25. In this case, clicking on the name of the script takes you directly to the highlighted line containing the error, as shown in Figure 4.5.

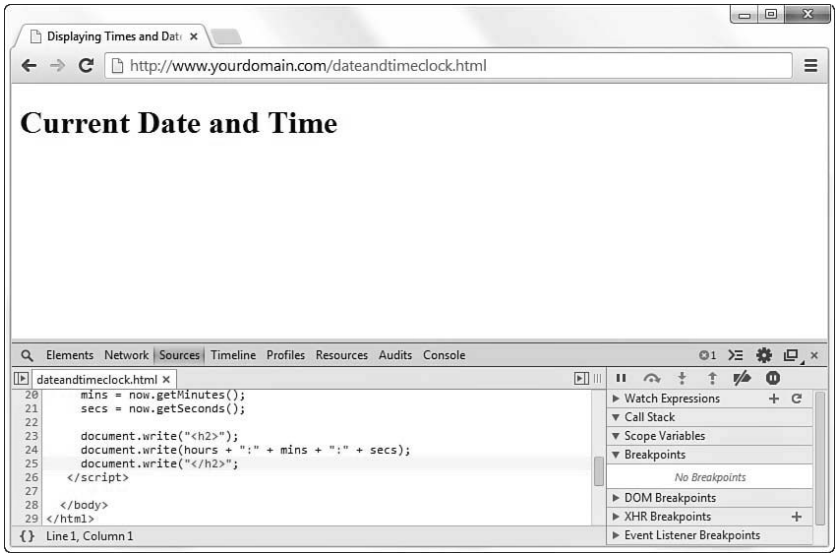


FIGURE 4.5
Chrome helpfully points out the offending line.

Most modern browsers contain JavaScript debugging tools such as the one you just witnessed. You'll learn more about this in the next chapter.

Summary

During this chapter, you've learned what web scripting is and what JavaScript is. You've also learned how to insert a script into an HTML document or refer to an external JavaScript file, what sorts of things JavaScript can do, and how JavaScript differs from other web languages. You also wrote a simple JavaScript program and tested it using a web browser. You also learned how to modify and test scripts, and what happens when a JavaScript program runs into an error.

In the process of writing this script, you have used some of JavaScript's basic features: variables, the `document.write` statement, and functions for working with dates and times.

Now that you've learned a bit of JavaScript syntax, you're ready to continue on to learn all manner and sorts of things about web development before settling in to write interactive websites using client-side scripting.

Q&A

Q. Do I need to test my JavaScript on more than one browser?

A. In an ideal world, any script you write that follows the standards for JavaScript will work in all browsers, and 98% of the time (give or take) that's true in the real world. But browsers do have their quirks, and you should test your scripts in Chrome, Internet Explorer, and Firefox at a minimum.

Q. If I plan to learn PHP, Ruby, or some other server-side programming language anyway, will I have any use for JavaScript?

A. Certainly. JavaScript is the ideal language for many parts of a web-based application, such as form validation. Although PHP, Ruby, and other server-side languages have their uses, they can't interact directly with the user on the client side.

Q. When I try to run my script, the browser displays the actual script in the browser window instead of executing it. What did I do wrong?

A. This is most likely caused by one of three errors. First, you might be missing the beginning or ending `<script>` tags. Check them, and verify that the first reads `<script type="text/javascript">`. Second, your file might have been saved with a `.txt` extension, causing the browser to treat it as a text file. Rename it to `.htm` or `.html` to fix the problem. Third, make sure your browser supports JavaScript, and that it is not disabled in the Preferences dialog.

Q. Why are the `<strong>` and `<br />` tags allowed in the statements to print the time? I thought HTML tags weren't allowed within the `<script>` tags.

A. Because this particular tag is inside quotation marks, it's considered a valid part of the script. The script's output, including any HTML tags, is interpreted and displayed by the browser. You can use other HTML tags within quotation marks to add formatting, such as the `<h2>` tags we added for the large clock display.

Workshop

The workshop contains quiz questions and exercises to help you solidify your understanding of the material covered. Try to answer all questions before looking at the "Answers" section that follows.

Quiz

1. When a user views a page containing a JavaScript program, which machine actually executes the script?
   a. The user's machine running a web browser
   b. The web server
   c. A central machine deep within Netscape's corporate offices

2. What tool do you use to create and edit JavaScript programs?
   a. A browser
   b. A text editor
   c. A pencil and a piece of paper

3. What are variables used for in JavaScript programs?
   a. Storing numbers, dates, or other values
   b. Varying randomly
   c. Causing high-school algebra flashbacks

4. What should appear at the very end of a JavaScript script embedded in an HTML file?
   a. The `<script type="text/javascript">` tag
   b. The `</script>` tag
   c. The `END` statement

Answers

1. **a.** JavaScript programs execute on the web browser. (There is actually a server-side version of JavaScript, but that's another story.)
2. **b.** Any text editor can be used to create scripts. You can also use a word processor if you're careful to save the document as a text file with the `.html` or `.htm` extension.

3. **a.** Variables are used to store numbers, dates, or other values.
4. **b.** Your script should end with the `</script>` tag.

Exercises

- Add a millisecond field to the large clock. You can use the `getMilliseconds` function, which works just like `getSeconds` but returns milliseconds.
- Modify the script to display the time, including milliseconds, twice. Notice whether any time passes between the two time displays when you load the page.

INDEX

Symbols

@font-face feature (CSS3), 132-134

< > tags. *See individual entries indexed according to tag names*

: selectors. *See individual entries indexed according to selector names*

$(document).ready event handler (jQuery JavaScript library), 542-543

\ (forward slashes) and HTML, 170

. (periods), JavaScript objects, 355

+ (plus signs), JavaScript statements, 86

+= operator, 398

; (semicolons), JavaScript statements, 84, 354

A

A Small Orange web hosting provider, 7

<a> tags (HTML), 172-175, 222

absolute addresses and web pages, 170-171

absolute links and web pages, 170

absolute positioning, 273
- display property (CSS), 63
- positioning property, 274-277

accessibility, JavaScript best practices, 513

Adaptive Path, AJAX, 590

addEventListener function, 468

Adjust Hue/Lightness/Saturation tool (GIMP), 204

Adobe Photoshop, 198

AJAX (Asynchronous JavaScript + XML), 83, 589. *See also* JavaScript
- debugging applications, 601-606
- examples of, 591-592
- frameworks, 592
- jQuery and AJAX-related functionality, 606-607
- libraries, 592
  - ajaxRequest function, 595
  - ajaxResponse function, 595
  - creating, 594-595
  - quiz building example, 596-601
  - using, 596
- live searches
  - forms, 602-606
  - front end, 605
  - HTML file example, 603
- quiz building example, 596-601
- requests
  - awaiting responses, 593
  - back end, 590, 603-604
  - creating, 592-593
  - front end, 590, 604-605
  - interpreting response data, 594
  - JavaScript client, 590
  - sending, 593
  - server-side scripts, 590, 603-605
- XML and, 591
- XMLHttpRequest
  - awaiting responses to requests, 593
  - creating requests, 592-593
  - interpretting request response data, 594
  - opening URLs, 593
  - sending requests, 593

alerts (dialog boxes), 499

align property (CSS), 249

aligning
- images
  - horizontal alignment, 216-218
  - vertical alignment, 218-220
- text
  - attributes, 135
  - block-level elements, 136-137
  - tables, 153-156
  - text-align property, 67, 262
  - text-align style rule, 136-137
  - text-decoration property (CSS), 67
- vertical-align property (CSS), 262

alternate text, 214-215

Amazon.com, 592, 648-649
analogous color schemes, 191
anchor objects (DOM), 372
anchor tags
naming, 173
web pages
identifying locations within, 172
linking to anchor locations, 172-175
AngularJS JavaScript framework, 538
animation, 211-212
HTML elements and jQuery JavaScript library, 548-553
jQuery UI library, 566
Apple Safari web browser. *See* Safari web browser
arguments (JavaScript functions), 420
Arial font (text), 129
arithmetic mean, 437
arrays (JavaScript), 340-342
declaring, 408
elements of, accessing, 409
length of, 408-409
numeric arrays, 408, 411-413
string arrays, 409-411
<article> tags (HTML5), 25, 37-41, 45-46
ASCII (American Standard Code for Information Interchange), 475
formatted text, 122
web content, creating, 27, 34
<aside> tags (HTML5), 25, 37-41, 47-49
.asp file extensions, 27
attributes, 135, 169
default values, 136
style attribute and text-align style rule, 136-137
audio
<audio> element, embedding multimedia files, 238-239
HTML5 audio playback, 238-239
auto image loading and web browsers, 21

B

<b> tags (HTML), 122, 125-126
Backbone.js JavaScript framework, 538
back end (AJAX requests), 590, 603-604
Back/Forward buttons, adding to documents, 373-374
backgrounds
background-position style property, 224
background-repeat style property, 224
color
background-color property (CSS), 66
background-color style property, 223
CSS and, 195-198
images, 223-224
tiled backgrounds, 209-211
bad websites, examples of, 193
bandwidth, web hosting providers, 7
banners, creating, 206-208
BAWSI.org, website organization, 651
Berners-Lee, Sir Tim, 2
block-level elements, aligning text in, 136-137
block value (CSS display property), 63
blogs, 20
<body> tags, 27, 31-33
boldface text, 125-126
Boolean operators. *See* logical operators (JavaScript)
Boolean values (JavaScript), 400
Bootstrap framework, 657
borders
CSS
border-bottom property, 64-65
border-color property, 64-65
border-left property, 64-65
border property, 64-65
border-right property, 64-65
border-style property, 64-65
border-top property, 64-65
border-width property, 64
box model, 270
tables
color, 195-198
creating, 148-149
spacing, 156
box model (CSS), 269
borders, 270
content, 270
lists and, 286-290
margins, 270
padding, 270
sizing elements, 270-272
**
 tags, 32-33, 128**
break statements (JavaScript), 458
breakpoints, debugging JavaScript, 115-116
browsers (web)
Chrome, 9
cross-browser scripting, 519
140 cross-browser color names, 192
debugging browsers, 520
feature sensing, 509, 519
CSS, 62
debugging, 520
Developer Tools
CSS Inspector, 106-107, 110
debugging CSS, 98, 106-107, 110
debugging HTML, 98-100, 103-105
debugging JavaScript, 111-117
HTML Inspector, 100, 103-105
viewing source code, 110
development of, 2
dialog boxes, displaying, 497-499
distributing, 20
Firefox, 9
graceful degradation, 506
helper applications, defining, 231

history, 2, 372-374
HTML development, 2
images, auto loading, 21
information, reading via JavaScript
dishonest browsers, 517-518
displaying information, 515-516
Internet Explorer, 9, 470
links, opening in new browser windows, 181
lists, displaying, 140
margins, 287-288
non-Internet Explorer event properties, 470
non-JavaScript browsers, 520
avoiding errors, 522-525
detecting, 521
JavaScript optionality, 521-522
<noscript> tags, 521
non-viewable window areas, 316
Opera, 9
padding, 287-288
plug-ins, defining, 231
popularity of, 26
pop-up windows, 181
progressive enhancement, 506-507
QuickTime support, 233
Safari, 9
search engines, 521
sensing. *See* feature sensing
servers, basic browser server interaction, 3-5
text, adjusting font size, 21, 60
websites
comparing, 26
testing, 8-9, 26
windows
creating, 490-491
moving, 493-495
opening/closing, 491-493
resizing, 493-495
timeouts, 495-497

built-in objects, 356, 425, 430-433
date object, 438
converting date formats, 440
creating, 438
local time values, 440
reading date values, 439
setting date values, 439
time zones, 440
definitions, extending, 431
math object, 434-435
generating random numbers, 435-438
rounding decimal numbers, 434
truncating decimal numbers, 435
bulleted lists, 142
buttons, creating, 206-208
buying domain names, 7

C

case-sensitivity, JavaScript syntax, 360
case statements (JavaScript), 453
categorizing elements, 44
CD-ROM, transferring photos to, 200
cells (tables)
creating, 146
sizing, 151-153
Champeon, Steve, 507
character entity, 123
checkboxes (forms), 621-623
child objects (JavaScript), 426
children (DOM), 377
child tags. *See* nested tags (HTML)
Chrome (Google), 9
Console, debugging JavaScript, 111-112
Developer Tools
CSS Inspector, 108-109
debugging JavaScript, 111-117
HTML Inspector, 101-103
error messages, displaying, 90
Sources panel, debugging JavaScript, 114-117
clarity, coding for, 653-654
Classic FTP client
server connections, 14
website connections, 12-13
clear property (CSS) and text property, 281
client-side scripting, 338, 349
clip art, 199
closing/opening browser windows, 491-493
closing slashes (HTML tags), 33
closing tags (HTML), 30
cm value (CSS height/width properties), 63
code (source), viewing via Developer Tools, 110
color
140 cross-browser color names, 192
analogous color schemes, 191
background color
background-color style property, 223
CSS and, 195-198
best practices, 190-191
border color (tables) and CSS, 195-198
color property (CSS), 67
color style rule (CSS), 129
color theory, 191
color wheel, 191
Colorblind Web Page Filter tool, 198
complementary color schemes, 191
graphics, adjusting color in, 204
hexadecimal color codes, 192-195
links and, 192
monitors and, 192
names and case sensitivity, 192
tables and, 155-156

text
color and CSS, 195-198
formatting in style sheets, 56
triadic color schemes, 191
using, 190-191
W3C color standards, 192
columns
CSS, 158-162
fixed/liquid hybrid layouts
defining columns in, 323-325
setting column height in, 326-330
combining string object values, 403-404
command chaining, 549
comments
JavaScript code comments, 340-342, 361-362, 511-512
websites, maintaining code via, 652-653
complementary color schemes, 191
compression (graphics), 199, 205
concatenation operators (JavaScript), 343
conceptualizing web pages, 38-43
conditional expressions, 387, 446-449
conditional operators (JavaScript), 446-447
conditional statements (JavaScript), 356
conditions (for loops), 454
confirmations (dialog boxes), 499
Console (Chrome), debugging JavaScript, 111-112
constructor functions (JavaScript), 428
containers, 376
content
CSS box model, 270
JavaScript best practices, 505
web content
creating, 2-3
delivering, 3-5
publishing, 19-20
selecting web hosting providersl, 6-8, 26
viewing locally, 5
continue statements (JavaScript), 459
continuing loops (JavaScript), 459
control panels, selecting web hosting providers, 7-8
converting
between data types, 401-402
date formats, 440
string case, 405
copyrights, 123, 199
counters (JavaScript), for loops, 453
Creative Commons licenses, 199
cropping images, 200-202
cross-browser scripting
debugging browsers, 520
event handlers and JavaScript, 508-509
feature sensing, 509, 519
CSS (Cascading Style Sheets)
align property, 249
box model, 269
borders, 270
content, 270
lists and, 286-290
margins, 270
padding, 270
sizing elements, 270-272
browser support for, 62
cascading components, 56
clear property and text flow, 281
color, specifying, 195-198
columns, 158-162
creating, 57-62
CSS1, 57
CSS 2, 57
CSS3, 57
CSS Zen Garden and layouts, 315
debugging via Developer Tools, 98, 106-107, 110
definition of, 55-56
 tags, 56, 70
DOCTYPE declarations, 272
external style sheets, 56-62
float property, 249, 262-266, 281
formatting properties, 63
background-color property, 66
border property, 64-65
border-bottom property, 64-65
border-color property, 64-65
border-left property, 64-65
border-right property, 64-65
border-style property, 64-65
border-top property, 64-65
border-width property, 64
color property, 67
font property, 67
font-family property, 66
font-size property, 66
font-style property, 66
font-weight property, 66
line-height property, 67
padding property, 67
text-align property, 66-67
text-decoration property, 67
text-indent property, 66
frameworks, 657
HTML documents, linking to, 61
imagemaps, 292-296
inline styles, 71-72
Inspector, debugging CSS, 106-107, 110
internal style sheets, 56, 70-72
jQuery JavaScript library and, 544
layouts
elastic layouts, 334
fixed layouts, 316-317
fixed/liquid hybrid layouts, 321-332
liquid layouts, 315, 318-321
properties, 63
responsive web design, 332-333
<link /> tag, 61
links
linking to HTML documents, 61
styling, 181-185

lists
horizontal navigation, 307-311
list-style-image property, 286
list-style-position property, 286, 290-292
list-style-type property, 286
navigation lists, 296-311
vertical navigation, 296-306
margin property, 249-257
mouse actions, events and event handling, 478-484
overflow property and text flow, 281
padding property, 249, 257-261
positioning
absolute positioning, 273-277
overlapping elements, 278-280
positioning property, 274
relative positioning, 273-275
z-index property, 278-280
properties, hyphenating, 382
reference guide online resource, 57
selectors, 68
 tags, 70
style classes, 68-69
style ID, 70
style properties, 68
style rules, 56, 60
color style rule, 129
font-family style rule, 129
font-size style rule, 129
font weight style rule, 126
list-style-type style rule, 144
multiple style properties in, 69
viewing, 62
tags, 60
elements and, 56
selectors, 68
text
formatting color, 56
italic text, 67
line-through text, 67
sizing, 60
strikethrough text, 67
text-align property, 262
underline text, 67
validating, 72
vertical-align property, 262
z-index property, 278-280
CSS3 (Cascading Style Sheets version 3), @font-face feature, 132-134
current line and text flow, 280
custom objects (JavaScript), 356
customer service, web hosting providers, 6
Cyberduck FTP client, 12

D

DailyRazor web hosting provider, 7
dashed value (CSS border-style properties), 65
data types (JavaScript), 400-402
date objects (JavaScript)
creating, 438
date formats, converting, 440
date values, 439
local time values, 440
time display example, 84
time zones, 440
debugging
AJAX applications, 601-606
browsers, 520
CSS, 98, 106-107, 110
HTML, 98-100, 103-105
JavaScript, 111-117
decimal numbers
rounding, 434
truncating, 435
declaring variables (JavaScript), 396
decrementing/incrementing variables (JavaScript), 398
definition lists, 138-140, 286
design patterns, JavaScript best practices, 513
Developer Tools
debugging
CSS, 98, 106-107, 110
HTML, 98-100, 103-105
JavaScript, 111-117
source code, viewing, 110
dialog boxes, displaying, 497-499
directories (web content), 169-170
dishonest browsers, 517-518
displaying
document information, 369-371
error messages, 90
time (JavaScript example)
adding scripts to web pages, 86
creating output, 85
Date objects, 84
error handling, 89-91
modifying scripts, 87-89
<script> tags, 84-85
statements, 84-85
testing scripts, 87
variables, 84-85
DisplayKey function, keyboard events, 475-476
display property (CSS), 63
distributing web browsers, 20
** tags**
CSS, 56, 70
HTML, 222
onclick events, 478-484
DOCTYPE declarations, 272
documenting code, 511-512, 652-653
document node (DOM), 379
document objects (DOM)
anchor objects, 372
document information, displaying, 369-371
link objects, 371, 375
methods of, 371

properties of, 369
text, writing within documents, 371
document roots, 14-18
document.write statements (JavaScript), 80, 85
Dojo JavaScript library, 534
DOM (Document Object Model), 367
children, 377
jQuery JavaScript library and, 544
layers
controlling positioning via JavaScript, 381-385
creating, 380-381
moveable layers, 381-385
nodes, 377
document node, 379
methods of, 380
properties, 378-379
objects, 425
anchor objects, 372
document objects, 369-372, 375
hiding/showing, 385-387
history objects, 372-374
link objects, 371, 375
location objects, 374-375
methods, 368
naming, 368
parents, 377
properties, 368
referencing, 368
showing/hiding, 385-387
siblings, 378
window objects, 368, 489-499
parents, 377
readiness and, 542
siblings, 378
structure of, 376-377
text
adding to web pages, 389-391
modifying web pages, 387-388
DOM objects (JavaScript), 356
domain names, purchasing, 7
dotted value (CSS border-style properties), 65
double value (CSS border-style properties), 65
do…while loops (JavaScript), 457
draggable() widget (jQuery UI library), 575-578, 581-584
droppable() widget (jQuery UI library), 575, 580-584
dynamic websites, 337
client-side scripting, 338
DOM, 345-346
images, changing based on user interaction, 346-347
JavaScript
changing images based on user interaction, 346-347
comments in HTML files, 340
displaying random content in HTML files, 340-344
scripting in HTML files, 338-339
server-side scripting, 338
VBScript, 338

E

editors (blog), 20
effects (JavaScript), 83, 534-536
elastic layouts, 334
elements
block-level elements, aligning text in, 136-137
categorizing, 44
definition of, 56
flow content, 44
form elements, grouping, 619-620
else keyword (JavaScript), 448-452
** tags (HTML), 125-126**
email addresses
email address encoders, 180
linking to, 179-180
validating, 629
<embed /> element, 237
embedding multimedia files, 235-238
Ember JavaScript framework, 538
emphasized (italic) text, 67, 125-126
empty tags (HTML), 30
error handling
JavaScript best practices, 510-511
JavaScript scripts, 89-91
non-JavaScript browsers, 522-525
error messages, displaying, 90
errors, avoiding, 524
escaping loops (JavaScript), 458
ESPN.com, website organization, 646
European languages, formatting text for, 122-124
event handlers, 465
$(document).ready handler (jQuery JavaScript library), 542-543
creating, 466-467
defining, 467-468
event objects, 466, 469-470
functions and, 594
JavaScript and, 81-82, 357, 360, 507
best practices, 358
cross-browser scripting, 508-509
W3C event model, 508
jQuery JavaScript library, 542-543, 553-554
keyboard events, 474-476
mouse events
mousestatus function, 473
onClick, 471-474, 478-484
onDblClick, 472
onMouseDown, 472-474
onMouseOut, 471
onMouseOver, 471
onMouseUp, 472-474
rollover images, 471
multiple event handlers, 468
onLoad events, 477
onUnload events, 478
parentheses and, 594

quotation marks and, 466
syntax of, 466
Yahoo! UI Library, 509
event objects, 466, 469-470
expressions (JavaScript), 399
external scripts (JavaScript), 81
external style sheets (CSS), 56-62

F

feature sensing, 509, 519
Fetch FTP client, 12
finding substrings (JavaScript), 407
Firefox web browser, 9, 90
FireFTP FTP client, 11-12
FireZilla FTP client, 12
fixed layouts, 316-317
fixed/liquid hybrid layouts, 321, 330-332
columns
defining in layouts, 323-325
height, 326-330
minimum width, setting, 325-326
structure of, 321-323
Flickr, 212-214
float property (CSS), 249, 262-266, 281
flow content elements, 44
flow control (JavaScript), 445
break statements, 458
case statements, 453
continue statements, 459
do…while loops, 457
for…in loops, 459-462
for loops, 453-455
if statements, 445
conditional expressions, 446-449
conditional operators, 446-447
else keyword, 448-452
logical operators, 447-448
testing multiple conditions, 450-452
infinite loops, 457-458
loops
break statements, 458
continue statements, 459
escaping, 458
switch statements, using multiple conditions, 452-453
while loops, 456-457
flowing text, 280-281
fluid layouts. *See* liquid layouts
folders (web content), 169-171
fonts (text)
Arial font, 129
CSS
color style rule, 129
font property, 67
font-family property, 66
font-family style rule, 129
font-size property, 66
font-size style rule, 129
font-style property, 66
font-weight property, 66
font weight style rule, 126
<font> tags (HTML), 122, 128
foreign languages, 122-124
HTML, customizing in, 128-132
resumes, creating, 130-132
sans-serif font, 129
sizing, 21, 60
special characters, 122-124
Times Roman font, 129
web fonts, 132-134
<footer> tags, HTML5, 25, 37-41, 49-50
foreign languages, formatting text for, 122-124
for loops (JavaScript), 453-455
for…in loops (JavaScript), 459-462
formatting
CSS formatting properties, 63
background-color property, 66
border-bottom property, 64-65
border-color property, 64-65
border-left property, 64-65
border property, 64-65
border-right property, 64-65
border-style property, 64-65
border-top property, 64-65
border-width property, 64
color property, 67
font-family property, 66
font property, 67
font-size property, 66
font-style property, 66
font-weight property, 66
line-height property, 67
padding property, 67
text-align property, 66-67
text-decoration property, 67
text-indent property, 66
style sheets, 56, 60
text
aligning text, 135
boldface text, 125-126
bulleted lists, 142
customizing fonts in HTML, 128-132
definition lists, 138-140, 286
emphasized text, 126
foreign languages, 122-124
italic (emphasized) text, 125-126
multitiered lists, 144-145
nested lists, 140-141, 286
older HTML tags, 122
ordered lists, 138-140, 286
outlines, 141-142
resumes, creating, 130-132
special characters, 122-124
subscript text, 126
superscript text, 126
unordered lists, 138-140, 286
web fonts, 132-134
web pages, creating, 34
forms, 611
checkboxes, 621-623
creating, 612-617

data
displaying in pop-up windows, 633-635
labeling, 618-619
naming, 618
submitting, 631-632
elements
accessing via JavaScript, 633
grouping, 619-620
hidden data, 620
HTML5, 613, 629-630
JavaScript events, 632
passwords, 617
pull-down pick lists, 624-626
radio buttons, 623-624
scrolling lists, 624-626
selection lists, 624-626
sending, 613
text fields, 627-628
user input, accepting, 617-618
validating
HTML5 form validation, 629-630
JavaScript, 82
for statements (JavaScript), 357
Forward/Back buttons, adding to documents, 373-374
forward slashes (\) and HTML, 170
frameworks
AngularJS, 538
Backbone.js, 538
Bootstrap, 657
Ember, 538
Foundation, 657
HTML5 Boilerplate, 657
Knockout, 538
MVC pattern, 537
front end (AJAX requests), 590, 604-605
FTP (File Transfer Protocol), 10, 29
Classic FTP client
server connections, 14
website connections, 12-13
Cyberduck, 12
Fetch, 12
FireFTP, 11-12
FireZilla, 12
Fuchs, Thomas, 533
functions (JavaScript), 80, 419
arguments, 420
calling, 354, 421-422
constructor functions, 428
defining, 420-422
math functions, 435-438
naming, 361
parameters, 354, 420
values, returning, 423-424

G

Garrett, Jesse James, 590
get methods (JavaScript ojects), 439
Gickr, animated graphics, 211
.GIF (Graphics Interchange Format), 208
animated graphics, 211-212
tiled backgrounds, 209-211
transparent images, 209
GIMP (GNU Image Manipulation Program), 198
Adjust Hue/Lightness/Saturation tool, 204
banners, creating, 206-208
buttons, creating, 206-208
images
adjusting color, 204
cropping, 201-203
.JPEG compression, 205
resizing, 203-204
Red Eye Removal, 204
Git website version control, 656
global variables (JavaScript), 396
Gmail (Google), 522, 591
GMT (Greenwich Mean Time), JavaScript time display example, 84
Google searches, 4
Google Chrome web browser. *See* Chrome (Google)
Google Images, 212
Google Picasa, 199
graceful degradation, 506. *See also* progressive enhancement
graphics
Adobe Photoshop, 198
aligning
horizontal alignment, 216-218
vertical alignment, 218-220
animated graphics, 211-212
backgrounds, 224
background-image style property, 223
tiled backgrounds, 209-211
banners, creating, 206-208
buttons, creating, 206-208
CD-ROM, transferring graphics to, 200
clip art, 199
color, adjusting, 204
compression, 199
copyrighted graphics, 199
Creative Commons licenses, 199
cropping, 200-202
file sizes, 199
Flickr, 212-214
.GIF, 208
animated graphics, 211-212
tiled backgrounds, 209-211
transparent graphics, 209
GIMP, 198
adjusting graphic color, 204
banners, 206-208
buttons, 206-208
cropping graphics, 201-203
JPEG compression, 205
resizing graphics, 203-204
Google Images, 212
Google Picasa, 199
height/width, specifying, 216

image maps, 225-230, 292-296
.JPEG
compression, 205
tiled backgrounds, 209-211
links, turning graphics into, 220-223
Pixlr, 199
.PNG, 209
Red Eye Removal, 204
republishing, 214
resizing, 203-204
resolution, 199-200
responsive graphics, 216
rollover graphics, 471
software, choosing, 198
storing, 213
text descriptions, 214-215
transparent graphics, 209
user interaction, changing graphics based on, 346-347
uses of, 200
web pages
grabbing graphics from, 199
placing graphics on web pages, 212-214
Windows Media Video, linking to, 232-233
groove value (CSS border-style properties), 65
grouping
form elements, 619-620
statements (JavaScript). *See* loops (JavaScript)

H

<head> tags, 27, 31-33, 80
<header> tags (HTML5), 25, 36, 39-44
heading tags (HTML), 34-36
headings (tables), creating, 146
height
CSS box model, adjusting in, 270-272
fixed/liquid hybrid layouts, setting column height in, 326-330
height property (CSS), 63
images, specifying height in, 216
Hello World HTML file, creating, 10
help
CSS reference guide online resource, 57
web hosting providers, selecting, 6
helper applications, defining, 231
hexadecimal color codes, 192-195
hiding/showing
DOM objects, 385-387
form data, 620
HTML elements via jQuery JavaScript library, 547-548
history objects (DOM), 372-374
horizontal image alignment, 216-218
horizontal navigation, 307-311
horizontal rule tags (HTML), 33
<hr /> tags, 33
HTML (HyperText Markup Language)
AJAX
live search forms, 602
quiz building example, 596-597
attributes, 135-137, 169
<audio> element, embedding multimedia files, 238-239
Bootstrap framework, 657
character entity, 123
containers, 376
CSS
box model, 269-272, 286-290
external style sheets, 61-62
linking style sheets to HTML documents, 61
debugging
Developer Tools, 98-100, 103-105
HTML Inspector, 100, 103-105
development of, 2
<embed /> element, embedding multimedia files, 237
event handlers
JavaScript, 507-509
Yahoo! UI Library, 509
files
creating, 10, 27-36
index pages, 18-19
managing, 16-18
organizing, 16-18
transferring, 10-14, 29
viewing, 29
forms, 611
accessing elements via JavaScript, 633
checkboxes, 621-623
creating, 612-617
displaying data in pop-up windows, 633-635
grouping elements, 619-620
hidden data, 620
JavaScript events, 632
labeling form data, 618-619
naming form data, 618
passwords, 617
pull-down pick lists, 624-626
radio buttons, 623-624
scrolling lists, 624-626
selection lists, 624-626
sending, 613
submitting form data, 631-632
text fields, 627-628
text input, 617-618
user input, 617-618
forward slashes (\), 170
frameworks, 657
FTP clients, 10-14
<head> tags and functions, 80
Hello World sample file, 10
history of, 2
.html file extensions, 27
HTML-compatible word processors, 27

images
imagemaps, 228-230
placing images on web pages, 212-214
Inspector, debugging HTML, 100, 103-105
JavaScript
adding scripts to HTML documents, 86
advantages over HTML, 367
comments in HTML files, 340
displaying random content in HTML files, 340-344
scripting in HTML files, 338-339
jQuery JavaScript library and, 545-546
animating elements, 548-553
hiding/showing elements, 547-548
.js files, linking to, 81
layouts
elastic layouts, 334
fixed layouts, 316-317
fixed/liquid hybrid layouts, 321-332
liquid layouts, 315, 318-321
responsive web design, 332-333
links
absolute links, 170
anchor tags, 172-175
images as, 181
linking between web content, 175-178
linking to email addresses, 179-180
linking to external web content, 178-179
opening links in new browser windows, 181
relative-root links, 171
styling via CSS, 181-185
lists
bulleted lists, 142
definition lists, 138-140, 286
list-style-type style rule, 142-145
multitiered lists, 144-145
nested lists, 140-141, 286
ordered lists, 138-140, 286
outline creation, 141-142
unordered lists, 138-140, 286
marked-up text, 3
multiple conditions, testing, 450-451
Notepad, creating HTML files, 27
<object> element, embedding multimedia files, 235-237
outlines, creating, 141-142
pseudoclasses, 182-185
style rules, 136, 142-145
tables
aligning within, 153-156
borders, 148-149
cells, 146, 151-153
color, 155-156
creating, 146-150
headings, 146
images, 156
page layouts, 157
rows, 146
sizing, 150-153
spacing borders, 156
spanning within, 156
tags, 5, 28
<a> tags, 172-175, 222
attributes, 135, 169
 tags, 122, 125-126
 tags, 122, 125-126
<body> tags, 27, 31-33
</body> tags, 33

 tags, 32-33, 128
closing slashes, 33
closing tags, 30
 tags, 222
 tags, 125-126
 tags, 125-126
empty tags, 30
event handlers, 81, 358
 tags, 122, 128
 tags, 122
formatting and older HTML tags, 122
<head> tags, 27, 31-33
</head> tags, 33
heading tags, 34-36
horizontal rule tags, 33
<hr /> tags, 33
<html> tags, 27, 31-33
</html> tags, 33
 tags, 122, 125
 tags, 122, 125
<img> tags, 212-214
lang attribute, 31
line breaks, 32-33
naming files with, 27
nested tags, 141
opening tags, 30
<p> tags, 31-33
paragraphs, 32-33
<pre> tags, 126-127
</pre> tags, 126-127
pseudoclasses, 182-185
saving files with, 27
<section> tags, 177
<strike> tags, 126
 tags, 125-126
 tags, 125-126
<sub> tags, 126
</sub> tags, 126
<sup> tags, 126
</sup> tags, 126
<table> tags, 146
<tbody> tags, 150
<td> tags, 146-149
<tfoot> tags, 150
<th> tags, 146, 149
<thread> tags, 150
<title> tags, 27, 31-33, 36
</title>tags, 33
<tr> tags, 146-147
<u> tags, 126

text, formatting, 122
aligning text, 135
boldface text, 125-126
customizing fonts, 128-132
definition lists, 138, 286
emphasized text, 126
foreign languages, 122-124
italic text, 125-126
multitiered lists, 144
nested lists, 140-141, 286
older HTML tags, 122
ordered lists, 138, 286
outlines, 141
resumes, 130-132
special characters, 122-124
subscript text, 126
superscript text, 126
unordered lists, 138, 286
web fonts, 132-134
whitespace, 32
validating, 95-96, 105
<video> element, embedding multimedia files, 240-241
web content
absolute addresses, 170-171
absolute links, 170
anchor tags, 172-175
creating, 2-3
delivering, 3-5
directories, 169-170
folders, 169-171
images as links, 181
linking between, 175-178
linking to email addresses, 179-180
linking to external web content, 178-179
opening links in new browser windows, 181
organizing, 169-170
publishing, 19-20
relative addresses, 170-171
relative-root addresses, 170
relative-root links, 171
selecting web hosting providers, 6-8, 26
styling links via CSS, 181-185
website architectures, creating, 171
whitespace, 32
Word, creating HTML files, 27
word processors and HTML compatibility, 27
WYSIWYG editors, creating HTML files, 27
HTML5 (HyperText Markup Language version 5)
application development, 348-349
Boilerplate framework, 657
client-side scripting, 349
forms, 613, 629-630
JavaScript, 349
multimedia
audio playback, 238-239
video playback, 238-241
Outline tool, 43
semantic elements, 25, 36-50
server-side scripting, 349
tags
<article> tags, 25, 37-41, 45-46
<aside> tags, 25, 37-41, 47-49
<footer> tags, 25, 37-41, 49-50
<header> tags, 25, 36, 39-44
<nav> tags, 25, 37-41, 46-47
<section> tags, 25, 37-41, 44-45
web pages, conceptualizing, 38-43
hyperlinks. *See* links
hyphenating CSS properties, 382

I

<i> tags (HTML), 122, 125
if statements (JavaScript), 445
conditional expressions, 387, 446-449
conditional operators, 446-447
logical operators, 447-448
multiple conditions, testing
else keyword, 448-452
HTML files, 450-451
JavaScript files, 451-452
images
aligning
horizontal alignment, 216-218
vertical alignment, 218-220
animated images, 211-212
auto loading in web browsers, 21
background images, 224
background-image style property, 223
tiled backgrounds, 209-211
banners, creating, 206-208
buttons, creating, 206-208
CD-ROM, transferring images to, 200
clip art, 199
color, adjusting, 204
compression, 199
copyrighted images, 199
Creative Commons licenses, 199
cropping, 200-202
file sizes, 199
Flickr, 212-214
.GIF, 208
animated images, 211-212
tiled backgrounds, 209-211
transparent images, 209
Google Images, 212
Google Picasa, 199
height/width, specifying, 216
imagemaps, 225-230, 292-296

.JPEG
compression, 205
tiled backgrounds, 209-211
links, turning images into, 181, 220-223
Pixlr, 199
PNG, 209
Red Eye Removal, 204
republishing, 214
resizing, 203-204
resolution, 199-200
responsive images, 216
rollover images, 471
storing, 213
tables and, 156
text descriptions, 214-215
transparent images, 209
user interaction, changing images based on, 346-347
uses of, 200
web pages
grabbing images from, 199
placing images on web pages, 212-214
Windows Media Video, linking to, 232-233
<img> tags (HTML), 212-214
in value (CSS height/width properties), 63
increment expressions (for loops), 454
incrementing/decrementing variables (JavaScript), 398
indenting
code, 653-654
text, web page creation, 34
index pages, HTML file management, 18-19
indexes (JavaScript), for loops, 453
infinite loops (JavaScript), 457-458
initial expressions (for loops), 454
inline styles (CSS), internal style sheets and, 71-72
inline value (CSS display property), 63
input controls and forms
checkboxes, 621-622
radio buttons, 623-624
selection lists, 624-626
text fields, 627-628
inset value (CSS border-style properties), 65
Inspector, debugging
CSS, 106-107, 110
HTML, 100, 103-105
internal style sheets (CSS), 56, 70-72
Internet service providers (ISP), selecting, 26
Internet Explorer, 9
DOCTYPE declarations, 272
error messages, displaying, 90
event properties, 470
interpreted languages, 78
ISP (Internet Service Providers), selecting, 26
italic (emphasized) text, 67, 125-126

J

JavaScript
accessibility, 513
AJAX, 83, 537
jQuery and AJAX-related functionality, 606-607
live search forms, 604-606
quiz building example, 598-600
requests, 590, 604-605
arrays, 340-342
accessing elements of, 409
declaring, 408
length of, 408-409
numeric arrays, 408, 411-413
string arrays, 409-411
best practices, 358, 362-363, 503-504
accessibility, 513
behvaior, 505
comments, 361-362, 511-512
content, 505
design patterns, 513
documenting code, 511-512
error handling, 510-511
event handlers, 507-509
graceful degradation, 506
overusing JavaScript, 504-505
presentation, 505
progressive enhancement, 506-507
reusing code, 514
usability, 512
web standards and browser specificity, 509
break statements, 458
breakpoints, debugging, 115-116
browsers
reading information on, 515-518
specificity and web standards, 509
capabilities of, 78, 82
case statements, 453
client-side scripting, 338
comments, 340-342, 511-512
concatenation operators, 343
conditional expressions, 446-449
conditional operators, 446-447
continue statements, 459
cross-browser scripting
debugging browsers, 520
feature sensing, 509, 519
CSS, jQuery JavaScript library, 544
data types, 400-402
Date objects, time display example, 84
debugging, 111-117
design patterns, 513
development of, 78
documenting code, 511-512
document.write statements, 80, 85
Dojo library, 534

DOM, 345-346, 367
adding text to web pages, 389-391
anchor objects, 372
children, 377
document objects, 369-372, 375
hiding/showing objects, 385-387
history objects, 372-374
jQuery JavaScript library, 544
layers, 380-385
link objects, 371, 375
location objects, 374-375
modifying text in web pages, 387-388
naming objects, 368
nodes, 377-380
object methods, 368
object properties, 368
objects, 425
parents, 377
referencing objects, 368
showing/hiding objects, 385-387
siblings, 378
structure of, 376-377
window objects, 368, 489-499
do…while loops, 457
effects, 83, 534-536
else keyword, 448-452
error handling, 89-91, 510-511, 522-525
event handlers, 81-82, 357-360, 465, 507
creating, 466-467
cross-browser scripting, 508-509
defining, 467-468
event objects, 466, 469-470
keyboard events, 474-476
mouse events, 471-474, 478
multiple event handlers, 468
onclick events, 478-484
onLoad events, 477
onUnload events, 478
quotation marks and, 466
syntax of, 466
W3C event model, 508
expressions, 399
external scripts, 81
flow control, 445
break statements, 458
case statements, 453
continue statements, 459
continuing loops, 459
do…while loops, 457
escaping loops, 458
for loops, 453-455
for…in loops, 459-462
if statements, 445-452
infinite loops, 457-458
switch statements, 452-453
while loops, 456-457
for loops, 453-455
for…in loops, 459-462
forms
accessing elements, 633
events, 632
validating, 82
frameworks, 537-538
functions, 80, 419
arguments, 420
calling, 354, 421-422
constructor functions, 428
defining, 420-422
math functions, 435-438
naming, 361
parameters, 354, 420
returning values, 423-424
Gmail and, 522
graceful degradation, 506
history of, 78
HTML
comments, 340
displaying random content in, 340-344
JavaScript advantages over, 367
jQuery JavaScript library, 545-553
scripting in, 338-339
HTML5, 349
if statements, 445
conditional expressions, 446-449
conditional operators, 446-448
else keyword, 448-452
testing multiple conditions, 450-452
images, changing based on user interaction, 346-347
infinite loops, 457-458
jQuery library, 531-532
$(document).ready event handler, 542-543
AJAX-related functionality, 606-607
animating elements, 548-553
command chaining, 549
CSS content, 544
DOM content, 544
downloading, 541
event handling, 553-554
hiding/showing elements, 547-548
HTML content, 545-553
storing, 541
jQuery UI library
animation easing, 566
applying effects, 564-573
:data() selector, 558
downloading, 557-558
:focusable() selector, 559
positioning elements, 559-564
storing, 557
:tabbable selector, 559
visibility effects, 570-573
widgets, 573-585
.js files, 81
JSON, 363-364, 591
layers, controlling positioning of, 381-385

libraries, 529-530
adding effects via, 534-535
building scripts, 535-536
Dojo, 534
jQuery, 531-532, 541-554, 606-607
jQuery UI, 557-585
Prototype, 532-533
Script.aculo.us, 533-536
using effects, 535
Yahoo! UI Library, 534
logical operators, 447-448
loops
break statements, 458
continue statements, 459
escaping, 458
modifying scripts, 87-89
modulo operators, 437
multiple conditions, testing, 451-452
non-JavaScript browsers, 520
avoiding errors, 522-525
detecting, 521
JavaScript optionality, 521-522
<noscript> tag, 521
objects
built-in objects, 356, 425, 430-435, 438-440
child objects, 426
creating, 426, 429-430
custom objects, 356
date objects, 438-440
defining, 427-428
defining methods, 428-429
DOM objects, 356, 425
math objects, 434-438
methods, 355-356, 426, 439
naming, 361
properties, 355, 426
prototypes, 432-433
simplifying scripting via, 427-430
operators, precedence of, 399-400
order of script operation, determining, 359-360
output creation, time display example, 85
overusing, 504-505
parseFloat() function, 402
parseInt() function, 402
progressive enhancement, 363, 506-507
Prototype library, 532-533
remote scripting, 83
reusing code, 514
Script.aculo.us library, 533
<script> tags, 79-81, 84-85
statements
conditional statements, 356
for statements, 357
function calls, 354
loops, 357
objects, 355-356, 361
plus signs (+) in, 86
semicolons, 84
time display example, 84-85
stepping into/out of code, 115
stepping over code, 115-116
strings
calculating length of, 404-405
converting case, 405
string arrays, 409-411
string objects, 402-404, 433
substrings, 405-407
switch statements, 452-453
syntax, 403
case-sensitivity, 360
functions, 361
objects, 361
reserved words, 361
spacing (whitespace), 361
variables, 361
testing
multiple conditions, 450-452
scripts, 87
text editors, 87
time display example, 84-91
toLowerCase() method, 405
toUpperCase() method, 405
.txt file extension, 87
unobtrusive scripting, 503, 523-525
usability, 80-81, 512
variables, 355, 361, 395
assigning values to variables, 397-398
declaring, 396
global variables, 396
incrementing/decrementing, 398
local variables, 396
localtime variable, 85
naming, 395-396
scope of, 396
time display example, 84-85
UTC variable, 85
web pages, adding scripts to, 79-80, 86
websites, navigating, 82
while loops, 456-457
Yahoo! UI Library, 534

join() method, 413

.JPEG (Joint Photographic Experts Group) files
compression, 205
tiled backgrounds, 209-211

jQuery JavaScript library, 531-532
$(document).ready event handler, 542-543
AJAX-related functionality, 606-607
command chaining, 549
CSS content, 544
DOM content, 544
downloading, 541

event handling, 553-554
HTML content, 545-546
animating elements, 548-553
hiding/showing elements, 547-548
storing, 541
jQuery UI library, 557
animation easing, 566
applying effects, 564-573
:data() selector, 558
downloading, 557-558
:focusable() selector, 559
positioning elements, 559-564
storing, 557
:tabbable selector, 559
visibility effects, 570-573
widgets, 573, 585
draggable() widget, 575-578, 581-584
droppable() widget, 575, 580-584
mouse interaction widget, 574-575
.js files, 81
JSON (JavaScript Object Notation), 363-364, 591
.jsp file extensions, 27

K–L

keyboard events, 474-476
Knockout JavaScript framework, 538
Koch, Peter-Paul, 520

labeling form data, 618-619
lang attribute (HTML tags), 31
languages (foreign), formatting text for, 122-124
layers (DOM)
creating, 380-381
moveable layers, 381-385
positioning, controlling via JavaScript, 381-385
layouts
elastic layouts, 334
fixed layouts, 316-317
fixed/liquid hybrid layouts, 330-332
defining columns in, 323-325
setting column height, 326-330
setting minimum width, 325-326
structure of, 321-323
layout properties (CSS), 63
liquid layouts, 318-321
responsive web design, 332-333
web resources, 315
leading (text), line-height property (CSS), 67
libraries
AJAX, 592
ajaxRequest function, 595
ajaxResponse function, 595
creating, 594-595
quiz building example, 596-601
JavaScript, 529-530
Dojo, 534
effects, 534-535
jQuery, 531-532, 541-554, 606-607
jQuery UI, 557-585
Prototype, 532-533
Script.aculo.us, 533-536
scripts, 535-536
Yahoo! UI Library, 534
line breaks, web page creation, 32-33
line-height property (CSS), 67
line-through text, 67
<link /> tags (CSS), 61
link objects (DOM), 371, 375
links
absolute links, 170
anchor tags, 172-175
color and, 192
documents, 371, 375
email addresses, 179-180
images, 181
linking to Windows Media Video, 232-233
turning images into links, 220-223
.js files, linking to, 81
multimedia files, 231-233
opening, 181
relative-root links, 171
styling via CSS, 181-185
web content
linking between, 175-178
linking to external web content, 178-179
Windows Media Video, linking images to, 232-233
liquid layouts, 315, 318-321
list-item value (CSS display property), 63
lists
bulleted lists, 142
CSS box model, 286-290
definition lists, 138-140, 286
list item indicators, 290-292
list-style-image property (CSS), 286
list-style-position property (CSS), 286, 290-292
list-style-type property (CSS), 286
list-style-type style rule, 142-145
multitiered lists, 144-145
navigation lists, 296-311
nested lists, 140-141, 286
ordered lists, 138-140, 286
outlines, building via lists, 141-142
unordered lists, 138-140, 286
LiveScript, JavaScript development, 78
live search
example, 606
forms, creating via AJAX, 602-606
requirements for, 606

loading web content, timing of, 21
local time values, date object (JavaScript) and, 440
local variables (JavaScript), 396
localtime variable (JavaScript), 85
location objects (DOM), 374-375
logical operators (JavaScript), 447-448
loops (JavaScript), 357
break statements, 458
continue statements, 459
continuing, 459
do…while loops, 457
escaping, 458
for…in loops, 459-462
for loops, 453-455
infinite loops, 457-458
while loops, 456-457
LunarPages web hosting provider, 7

M

managing
domain names, 7
HTML files, 16-19
web page headings, 34-35
websites
coding clarity, 653-654
comments, 652-653
documenting code, 652-653
indenting code, 653-654
maintainable code, 652-654
version control, 654-656
margins
browsers and, 287-288
CSS box model, 270
margin property (CSS), 249-257
marked-up text, 3
math object (JavaScript)
decimal numbers
rounding, 434
truncating, 435
math functions, 435-438
random numbers, generating, 435-438
methods
DOM nodes, 379-380
DOM objects, 368
document objects, 371
history objects, 372
location objects, 375
JavaScript objects, 355-356, 426
defining, 428-429
get methods, 439
prototypes, 432-433
string objects and, 433
Microsoft Internet Explorer web browser. *See* Internet Explorer
MIME types, multimedia files, 236
mm value (CSS height/width properties), 64
modifying
JavaScript scripts, 87-89
text in web pages, 387-388
modulo operators (JavaScript), 437
monitors
color and, 192
resolution, 207
mouses
events and event handling
mousestatus function, 473
onClick event handler, 471-475, 478-484
onDblClick event handler, 472
onMouseDown event handler, 472-474
onMouseOut event handler, 471
onMouseOver event handler, 471
onMouseUp event handler, 472-474
mouse interaction widget (jQuery UI library), 574-575
rollover images, 471
moveable layers (DOM), 381-385
moving browser windows, 493-495
Mozilla Firefox web browser. *See* Firefox web browser
multimedia, 242
audio files, HTML5 playback, 238-239
creating, 232
defining, 189
embedding files, 235-238
linking to files, 231-233
MIME types, 236
<object> element, 235-237
QuickTime support, 233
streaming files, 235
video files, HTML5 playback, 238-241
multiple event handlers, 468
multitiered lists, 144-145
MVC (Model-View-Controller) JavaScript framework pattern, 537

N

naming
anchor tags, 173
DOM objects, 368
files with HTML tags, 27
form data, 618
JavaScript
functions, 361
objects, 361
variables, 361, 395-396
NaN (Not a Number), 402
<nav> tags, HTML5, 25, 37-41, 46-47
navigating websites, 82
navigation lists
horizontal navigation, 307-311
primary navigation, 296
regular lists vs, 296
secondary navigation, 296

vertical navigation, 296-299
multilevel vertical navigation, 302-306
single-level vertical navigation, 300-302
nested lists, 140-141, 286
nested tags (HTML), 141
nodes (DOM), 377-378
document node, 379
methods of, 380
properties, 378-379
none values (CSS)
border-style properties, 65
display property, 63
non-viewable window areas (browsers), 316
<noscript> tag (JavaScript), detecting non-JavaScript browsers, 521
Notepad, creating HTML files, 27
null values (JavaScript), 401
numbers
arithmetic mean, 437
decimal numbers
rounding, 434
truncating, 435
numeric arrays (JavaScript), 408, 411-413
numeric data types (JavaScript), 400
random numbers, generating, 435-438

O

<object> element, embedding multimedia files, 235-237
objects
DOM, 425
anchor objects, 372
document objects, 369-372, 375
hiding/showing, 385-387
history objects, 372-374
link objects, 371, 375
location objects, 374-375
methods, 368, 371-372, 375
naming, 368
parents, 377
properties, 368-369, 372-375
referencing, 368
showing/hiding, 385-387
siblings, 378
window objects, 368, 489-499
JavaScript
built-in objects, 356, 425, 430-434, 438-440
child objects, 426
creating, 426
custom objects, 356
date objects, 438-440
defining, 427-428
instance creation, 429-430
math objects, 434-438
methods, 355-356, 426-429, 432-433, 439
naming, 361
properties, 355, 426, 432-433
prototypes, 432-433
scripting, 427-430
string objects, 402-404, 433
onClick event handler, 471-474, 478
onclick event (JavaScript) and mouse actions, 478-484
onDblClick event handler, 472
online resources, CSS
browser support, 62
reference guide, 57
onLoad events, 477
onMouseDown event handler, 472-474
onMouseOut event handler, 471
onMouseOver event handler, 471
onMouseUp event handler, 472-474
onUnload events, 478
opening/closing browser windows, 491-493
opening tags (HTML), 30
operators (JavaScript), precedence of, 399-400
Opera web browser, 9
optional JavaScript coding, 521-522
ordered lists, 138-140, 286
organizing
HTML files
document roots, 16-18
index pages, 18-19
web content, 169-170
web page headings, 34-35
websites, 641
Amazon.com, 648-649
BAWSI.org, 651
ESPN.com, 646
larger websites, 648-651
Peet's Coffee & Tea, 649
simple websites, 644-648
single-page websites, 642-643
Outline tool (HTML5), 43
outlines, building via lists, 141-142
outset value (CSS border-style properties), 65
overflow property (CSS) and text flow, 281
overlapping elements, 278-280
overusing JavaScript, 504-505

P

<p> tags, 31-33
padding
browsers and, 287-288
CSS box model, 270
padding property (CSS), 67, 249, 257-261
paragraphs, web page creation, 32-33
parameters (JavaScript functions), 354, 420
parent folders, 171
parents (DOM), 377
parseFloat() function (JavaScript), 402

parseInt() function (JavaScript), 402
passwords and forms, 617
Peet's Coffee & Tea, website organization, 649
periods (.), JavaScript objects, 355
peripherals
monitors
color and, 192
resolution, 207
mouses
events and event handling, 471-474, 478-484
mouse interaction widget (jQuery UI library), 574-575
rollover images, 471
photos, 200
aligning, 216-220
background photos, 223-224
cropping, 200-202
Flickr, 212-214
Google Images, 212
height/width, specifying, 216
imagemaps, 225-230, 292-296
links, turning photos into, 220-223
Red Eye Removal, 204
republishing, 214
resizing, 203-204
responsive images, 216
storing, 213
text descriptions, 214-215
user interaction, changing photos based on, 346-347
web pages, placing photos on, 212-214
Windows Media Video, linking to, 232-233
Photoshop (Adobe), 198
PHP, 612
AJAX live search forms, 603-604
.php file extensions, 27
Picasa (Google), 199
Pixlr, 199
plain text, 27, 34, 122
plug-ins, defining, 231
plus signs (+), JavaScript statements, 86
PNG (Portable Network Graphics), 209
pop-up windows, 181, 633-635
positionable elements. *See* layers (DOM)
positioning
absolute positioning, 273-277
overlapping elements, 278-280
relative positioning, 273-275
<pre> tags (HTML), 126-127
presentation, JavaScript best practices, 505
pricing web hosting providers, 7
primary navigation, 296
programming languages and strings, 85
progressive enhancement, 363, 506-507. *See also* graceful degradation
prompts (dialog boxes), 499
properties
DOM nodes, 378-379
DOM objects, 368
document objects, 369
history objects, 372
location objects, 374-375
JavaScript objects, 355
prototypes, 432-433
values, 426
Prototype JavaScript library, 532-533
pseudoclasses, 182-185
pt value (CSS height/width properties), 64
publishing web content
blog publication, 20
local publication, 19-20
pull-down pick lists (forms), 624-626
pull quotes, 48
purchasing domain names, 7
px value (CSS height/width properties), 64

Q–R

QuickTime, web browser support, 233
QuirksMode, debugging code, 520
quiz building example (AJAX), 596-601
quotes (pull), 48

radio buttons (forms), 623-624
random content, displaying in HTML files via JavaScript, 340-344
random numbers, generating, 435-438
readiness and DOM, 542
Red Eye Removal, 204
registered trademark symbol, 123
relationship properties (DOM nodes), 379
relative addresses and web pages, 170-171
relative positioning, 63, 273-275
relative-root addresses and web pages, 170
relative-root links and web pages, 171
reliability, web hosting providers, 6
remote scripting. *See* AJAX
republishing images, 214
reserved words and JavaScript syntax, 361
resizing
browser windows, 493-495
images, 203-204
resolution
graphics, 199-200
screen, 207
responsive images, 216
responsive web design, 332-333
reusing JavaScript code, 514
ridge value (CSS border-style properties), 65
rollover images, 471
rounding decimal numbers, 434
rows (tables), creating, 146
Ruby on Rails and Prototype JavaScript library, 533

S

Safari web browser, 9
sans-serif font (text), 129
saving files
files with HTML tags, 27
.js files, 81
scaling images, 203-204
screen resolution, 207
Script.aculo.us JavaScript library, 533-536
<script> tags (JavaScript), 79-81, 84-85
scripting
AJAX, 537, 589
ajaxRequest function, 595
ajaxResponse function, 595
back end, 590, 603-604
debugging applications, 601-606
examples of, 591-592
frameworks, 592
front end, 590, 604-605
JavaScript client, 590
jQuery and AJAX-related functionality, 606-607
libraries, 592-601
live search forms, 602-606
quiz building example, 596-601
requests, 590-594, 603-605
server-side scripts, 590, 603-605
XML and, 591
XMLHttpRequest, 592-594
client-side scripting, 338, 349
cross-browser scripting
debugging browsers, 520
event handlers and JavaScript, 508-509
feature sensing, 509, 519
effects (third-party libraries), 534-536
error handling, 510-511
frameworks
AngularJS, 538
Backbone.js, 538
Ember, 538
Knockout, 538
MVC pattern, 537
graceful degradation, 506
interpreted languages, 78
JavaScript
accessibility, 513
adding scripts to web pages, 86
adding to web pages, 79-80
advantages over HTML, 367
AJAX, 83, 590, 598-600, 604-607
arrays, 408-413
best practices, 358, 362-363, 503-514
break statements, 458
breakpoints, 115-116
capabilities of, 78, 82
case statements, 453
comments, 361-362, 511-512
conditional expressions, 446-449
conditional operators, 446-447
continue statements, 459
continuing loops, 459
cross-browser scripting, 509, 519-520
data types, 400-402
Date objects, 84
debugging via Developer Tools, 111-117
design patterns, 513
development of, 78
do…while loops, 457
documenting code, 511-512
document.write statements, 80, 85
Dojo library, 534
DOM, 367-391, 489-499
effects, 83
else keyword, 448-452
error handling, 89-91
escaping loops, 458
event handlers, 81-82, 357-360, 465-478, 507-509
events, 81
expressions, 399
external scripts, 81
flow control, 445-462
for loops, 453-455
for…in loops, 459-462
form events, 632-633
functions, 80, 354, 361, 419-424, 428, 435-438
Gmail and, 522
history of, 78
if statements, 445-452
infinite loops, 457-458
jQuery library, 531-532, 541-554, 606-607
jQuery UI library, 557-585
.js files, 81
JSON, 363-364, 591
libraries (third-party), 529-536, 541
logical operators, 447-448
modifying scripts, 87-89
navigating websites, 82
non-JavaScript browsers, 520-525
objects, 355-356, 361, 425-435, 438-440
operators, 399-400
order of script operation, 359-360
output, 85
overusing, 504-505
parseFloat() function, 402
parseInt() function, 402
plus signs (+) in statements, 86
progressive enhancement strategies, 363
Prototype library, 532-533
reading browser information, 515-518

remote scripting, 83
saving js files, 81
<script> tags, 79-81, 84-85
Script.aculo.us library, 533
simplifying, 427-430
statements, 84-85, 353-357, 361
stepping into code, 115
stepping out of code, 115-116
stepping over code, 115-116
strings, 402-411, 433
switch statements, 452-453
syntax, 360-361
testing scripts, 87
time display example, 84-91
toLowerCase() method, 405
toUpperCase() method, 405
unobtrusive scripting, 503, 523-525
usability, 512
using, 80-81
validating forms, 82
variables, 84-85, 355, 361, 395-398
web standards and browser specificity, 509
while loops, 456-457
Yahoo! UI Library, 534
JSON, 591
languages, 77
PHP, 612
progressive enhancement, 506-507
remote scripting. *See* AJAX
reusing code, 514
server-side scripts, 338
AJAX requests, 590, 603-605
HTML5, 349
text editors, 87
.txt file extension, 87
unobtrusive scripting, 503, 523-525

scrolling lists (forms), 624-626
search engines, 521
searches
Google searches, 4
live search forms, creating via AJAX, 606
HTML forms, 602
JavaScript front end, 604-605
PHP back end, 603-604
secondary navigation, 296
<section> tags
HTML, 177
HTML5, 25, 37-41, 44-45
selection lists (forms), 624-626
selectors (CSS), 68
semantic elements (HTML5), 36-38
<article> tags, 25, 39-41, 45-46
<aside> tags, 25, 39-41, 47-49
<footer> tags, 25, 39-41, 49-50
<header> tags, 25, 39-44
<nav> tags, 25, 39-41, 46-47
recommendations, 39
<section> tags, 25, 39-41, 44-45
semicolons (;), JavaScript statements, 84, 354
server-side scripts, 338
AJAX requests, 590, 603-605
HTML5, 349
servers
browsers, basic server interaction, 3-5
document roots, 14-18
FTP client connections, 14
space, 6
uptime, 6
web hosting providers, selecting, 6
shorthand conditional expressions (JavaScript), 448-449
siblings (DOM), 378
single-page websites, 642-643
sizing
browser windows, 493-495
cells (tables), 151-153
elements (CSS box model), 270-272
images, 203-204
tables, 150-153
text
font-size style rule (CSS), 129
style sheets, 60
skeleton pages. *See* templates, web page creation
solid value (CSS border-style properties), 65
sorting
numeric arrays (JavaScript), 411-413
string arrays (JavaScript), 411
source code, viewing via Developer Tools, 110
source editors, blogs, 20
Sources panel (Chrome), debugging JavaScript, 114-117
spacing (whitespace), JavaScript syntax, 361
** tags (CSS), 70**
spanning with tables, 156
special characters (symbols), 122-124
special effects (JavaScript), 83
splitting string arrays (JavaScript), 410-411
statements (JavaScript), 353
conditional statements, 356
for statements, 357
function calls, 354
loops, 357
objects, 355
built-in objects, 356
custom objects, 356
DOM objects, 356
naming, 361
plus signs (+) in, 86
semicolons (;), 84
time display example, 84-85
variables, 355, 361
naming, 361
Stephenson and Prototype JavaScript library, Sam, 532-533
stepping into JavaScript code, 115

stepping out of JavaScript code, 115-116
stepping over JavaScript code, 115-116
streaming multimedia files, 235
<strike> tags (HTML), 126
strikethrough text, 67
strings (JavaScript), 85, 401
case, converting, 405
length of, calculating, 404-405
string arrays, 409
sorting, 411
splitting, 410-411
string objects
adding methods to, 433
assigning values, 403-404
combining values, 403-404
creating, 402-403
substrings
finding, 407
getting single characters, 406-407
using parts of strings, 405-406
** tags (HTML), 125-126**
strong text. *See* boldface text
style attribute and text-align style rule, 136-137
style classes (CSS), 68-69
style ID (CSS), 70
style properties (CSS), 68
style rules, 56, 60
color style rule, 129
default values, 136
font weight style rule, 126
font-family style rule, 129
font-size style rule, 129
list-style-type style rule, 142-145
multiple style properties in, 69
text-align, 136-137
viewing, 62
style sheets
align property, 249
box model, 269
borders, 270
content, 270
lists and, 286-290
margins, 270
padding, 270
sizing elements, 270-272
browser support for, 62
cascading components, 56
clear property and text flow, 281
color, specifying, 195-198
columns, 158-162
creating, 57-62
CSS1, 57
CSS 2, 57
CSS3, 57
CSS Zen Garden and layouts, 315
debugging via Developer Tools, 98, 106-107, 110
definition of, 55-56
 tags, 56, 70
DOCTYPE declarations, 272
external style sheets, 56-62
float property, 249, 262-266, 281
formatting properties, 63
background-color property, 66
border property, 64-65
border-bottom property, 64-65
border-color property, 64-65
border-left property, 64-65
border-right property, 64-65
border-style property, 64-65
border-top property, 64-65
border-width property, 64
color property, 67
font property, 67
font-family property, 66
font-size property, 66
font-style property, 66
font-weight property, 66
line-height property, 67
padding property, 67
text-align property, 66-67
text-decoration property, 67
text-indent property, 66
frameworks, 657
HTML documents, linking to, 61
imagemaps, 292-296
inline styles, 71-72
Inspector, debugging CSS, 106-107, 110
internal style sheets, 56, 70-72
jQuery JavaScript library and, 544
layouts
elastic layouts, 334
fixed layouts, 316-317
fixed/liquid hybrid layouts, 321-332
liquid layouts, 315, 318-321
properties, 63
responsive web design, 332-333
<link /> tag, 61
links
linking to HTML documents, 61
styling, 181-185
lists
horizontal navigation, 307-311
list-style-image property, 286
list-style-position property, 286, 290-292
list-style-type property, 286
navigation lists, 296-311
vertical navigation, 296-306
margin property, 249-257
mouse actions, events and event handling, 478-484
overflow property and text flow, 281
padding property, 249, 257-261
positioning
absolute positioning, 273-277
overlapping elements, 278-280

positioning property, 274
relative positioning, 273-275
z-index property, 278-280
properties, hyphenating, 382
reference guide online resource, 57
selectors, 68
 tags, 70
style classes, 68-69
style ID, 70
style properties, 68
style rules, 56, 60
color style rule, 129
font-family style rule, 129
font-size style rule, 129
font weight style rule, 126
list-style-type style rule, 144
multiple style properties in, 69
viewing, 62
tags, 60
elements and, 56
selectors, 68
text
formatting color, 56
italic text, 67
line-through text, 67
sizing, 60
strikethrough text, 67
text-align property, 262
underline text, 67
validating, 72
vertical-align property, 262
z-index property, 278-280
<sub> tags (HTML), 126
subscript text, 126
substrings
finding, 407
parts of strings, using, 405-406
single characters, getting, 406-407
Subversion website version control, 656
<sup> tags (HTML), 126
superscript text, 126
switch statements (JavaScript)
multiple conditions, using, 452-453
syntax of, 453
symbols (special characters), 122-124
syntax, JavaScript, 403
case-sensitivity, 360
functions, 361
objects, 361
reserved words, 361
spacing (whitespace), 361
variables, 361

T

<table> tags (HTML), 146
tables
aligning within, 153-156
borders
creating, 148-149
spacing, 156
specifying color via CSS, 195-198
cells
creating, 146
sizing, 151-153
color in, 155-156, 195-198
creating, 146-150
columns, 158-162
headings, creating, 146
images in, 156
page layout via, 157
rows, creating, 146
sizing, 150-153
spanning within, 156
tags
CSS, 60
elements and, 56
<link /> tags, 61
selectors, 68
 tags, 70, 478-484
<section> tags, 177
, CSS, 70
HTML, 5, 28
<a> tags, 172-175, 222
attributes, 135, 169
 tags, 122, 125-126
 tags, 122, 125-126
<body> tags, 27, 31-33
</body> tags, 33

 tags, 32-33, 128
closing slashes, 33
closing tags, 30
containers, 376
 tags, 222
 tags, 125-126
 tags, 125-126
empty tags, 30
event handlers, 81
 tags, 122, 128
 tags, 122
<head> tags, 27, 31-33, 80
</head> tags, 33
heading tags, 34-36
horizontal rule tag, 33
<hr /> tags, 33
<html> tags, 27, 31-33
</html> tags, 33
 tags, 122, 125
 tags, 122, 125
<img> tags, 212-214
lang attribute, 31
line breaks, 32-33
naming files with, 27
nested tags, 141
older HTML tags, formatting and, 122
opening tags, 30
<p> tags, 31-33
paragraphs, 32-33
<pre> tags, 126-127
</pre> tags, 126-127
pseudoclasses, 182-185
saving files with, 27
<strike> tags, 126

 tags, 125-126
 tags, 125-126
<sub> tags, 126
</sub> tags, 126
<sup> tags, 126
</sup> tags, 126
<table> tags, 146
<tbody> tags, 150
<td> tags, 146-149
<tfoot> tags, 150
<th> tags, 146, 149
<thread> tags, 150
<title> tags, 27, 31-33, 36
</title> tag, 33
<tr> tags, 146-147
<u> tags, 126
HTML5
<article> tags, 37
<aside> tags, 37
<footer> tags, 37
<header> tags, 36
<nav> tags, 37
<section> tags, 37
semantic elements, 36-38
JavaScript
<script> tags, 79-81, 84-85
<tbody> tags (HTML), 150
<td> tags (HTML), 146-149
templates, web page creation, 31
testing
AJAX quiz building example, 600-601
JavaScript scripts, 87
web content, 20-21
websites, multiple web browsers, 8-9, 26
text
aligning
attributes, 135
block-level elements, 136-137
tables, 153-156
text-align style rule, 136-137
alternate text, 214-215
ASCII formatted text, 27, 34, 122
color
color property (CSS), 67
CSS and, 195-198
CSS columns, 158-162
documents, writing text within, 371
editors, 87
flowing text, 280-281
fonts
Arial font, 129
color style rule (CSS), 129
font property (CSS), 67
font-family property (CSS), 66
font-family style rule (CSS), 129
font-size property (CSS), 66
font-size style rule (CSS), 129
font-style property (CSS), 66
font-weight property (CSS), 66
font weight style rule (CSS), 126
sans-serif font, 129
Times Roman font, 129
formatting
aligning text, 135
boldface text, 125-126
bulleted lists, 142
color, 56
customizing fonts in HTML, 128-132
definition lists, 138-140, 286
emphasized text, 126
foreign languages, 122-124
italic text, 125-126
multitiered lists, 144-145
nested lists, 140-141, 286
ordered lists, 138-140, 286
outlines, 141-142
resumes, 130-132
special characters, 122-124
subscript text, 126
superscript text, 126
unordered lists, 138-140, 286
web fonts, 132-134
web page creation, 34
forms
accepting text input in, 617-618
passwords, 617
text fields, 627-628
graphics, 214-215
HTML, whitespace, 32
indenting, web page creation, 34
italic (emphasized) text, 67
leading, line height property (CSS), 67
line breaks, web page creation, 32-33
line-through text, 67
lists
definition lists, 138-140
ordered lists, 138-140
unordered lists, 138-140
paragraphs, web page creation, 32-33
plain text, 27, 34, 122
sizing, 60, 129
strikethrough text, 67
style sheets
formatting color, 56
sizing, 60
text-align property (CSS), 67
text-align style rule, 136-137
text-decoration property (CSS), 67
text-indent property (CSS), 66
underline text, 67
web browsers, adjusting font size settings, 21
web pages
adding to web pages, 389-391
modifying text in, 387-388
<tfoot> tags (HTML), 150
<th> tags (HTML), 146, 149
third-party JavaScript libraries, 529-531
Dojo, 534
effects, 534-536
jQuery, 531-532
$(document).ready event handler, 542-543

AJAX-related functionality, 606-607
animating elements, 548-553
command chaining, 549
CSS content, 544
DOM content, 544
downloading, 541
event handling, 553-554
hiding/showing elements, 547-548
HTML content, 545-553
storing, 541
jQuery UI, 557-585
Prototype, 532-533
Script.aculo.us, 533-536
scripts, building, 535-536
Yahoo! UI Library, 534
<thread> tags (HTML), 150
tiled backgrounds, 209-211
time, displaying (JavaScript example)
creating output, 85
Date objects, 84
error handling, 89-91
scripts
adding to web pages, 86
modifying, 87-89
<script> tags, 84-85
testing, 87
statements, 84-85
variables, 84-85
timeouts (browser windows), 495-497
Times Roman font (text), 129
time zones and date objects (JavaScript), 440
timing loading of web content, 21
<title> tags, 27, 31-33, 36
toLowerCase() method (JavaScript), 405
tooltips, 215
toUpperCase() method (JavaScript), 405
<tr> tags (HTML), 146-147
trademark (registered/unregistered) symbols, 123
transferring HTML files, 10-14, 29
transparent images, 209
triadic color schemes, 191
truncating decimal numbers, 435
.txt file extensions, 87

U

<u> tags (HTML), 126
underline text and style sheets, 67
unobtrusive scripting, 503, 522-525
unordered lists, 138-140, 286
unregistered trademark symbol, 123
uptime and servers, 6
URL (Uniform Resource Locators), opening, 593
usability, JavaScript best practices, 512
USB (Universal Serial Bus) drivers and web browser distribution, 20
user input in forms
accepting input, 617-618
passwords, 617
UTC (Universal Time Coordinated)
JavaScript time displaying example, 84
UTC variable (JavaScript), 85

V

validating
email addresses, 629
forms
HTML5, 629-630
JavaScript, 82
style sheets, 72
web content, 95-96, 105
variables (JavaScript), 355
declaring, 396
global variables, 396
incrementing/decrementing, 398
local variables, 396
localtime variable, 85
naming, 361, 395-396
scope of, 396
time display example, 84-85
UTC variable, 85
values, assigning to variables, 397-398
VBScript (Microsoft), client-side scripting, 338
version control and websites, 654-656
vertical image alignment, 218-220
vertical navigation, 296-299
multilevel vertical navigation, 302-306
single-level vertical navigation, 300-302
video
HTML5 video playback, 238-241
<video> element, embedding multimedia files, 240-241
viewing
CSS style rules, 62
HTML files, 29
web pages, 29
visibility effects (jQuery UI library), 570-573
visual editors and blogs, 20

W

W3C color standards, 192
W3C CSS Validator, 72
W3C event model, 508
W3C Validation Service, 96-97, 105
web browsers
Chrome, 9
CSS support, 62
Developer Tools
CSS Inspector, 106-107, 110
debugging CSS, 98, 106-107, 110

debugging HTML, 98-100, 103-105
debugging JavaScript, 111-117
HTML Inspector, 100, 103-105
viewing source code, 110
development of, 2
distributing, 20
Firefox, 9
graceful degradation, 506
helper applications, defining, 231
history of, 2
HTML development, 2
images, auto loading, 21
Internet Explorer, 9
lists, displaying, 140
Opera, 9
plug-ins, defining, 231
popularity of, 26
progressive enhancement, 506-507
QuickTime support, 233
Safari, 9
servers, basic browser server interaction, 3-5
text, adjusting font size settings, 21
websites
comparing, 26
testing, 8-9, 26

web content
absolute addresses, 170-171
aligning
align property (CSS), 249
text-align property (CSS), 262
vertical-align property (CSS), 262
clear property (CSS) and text flow, 281
color
140 cross-browser color names, 192
best practices, 190-191
Colorblind Web Page Filter tool, 198
hexadecimal color codes, 192-195
names and case sensitivity, 192
using, 190-191
W3C color standards, 192
comparing, 26
creating, 2-3
ASCII text, 27, 34
comparing web content HTML codes, 36
formatting text, 34
HTML tags, 27-33
indenting text, 34
line breaks, 32-33
organizing content via headings, 34-35
overview of, 29
paragraphs, 32-33
plain, text, 27, 34
templates, 31
CSS box model, 269
borders, 270
content, 270
lists and, 286-290
margins, 270
padding, 270
sizing elements, 270-272
delivering, 3-5
directories, 169-170
floating elements, 249
float property (CSS), 249, 262-266, 281
folders, 169-171
forms, 611
accessing elements via JavaScript, 633
checkboxes, 621-623
creating, 612-617
displaying data in pop-up windows, 633-635
grouping elements, 619-620
hidden data, 620
JavaScript events, 632
labeling form data, 618-619
naming form data, 618
passwords, 617
pull-down pick lists, 624-626
radio buttons, 623-624
scrolling lists, 624-626
selection lists, 624-626
sending, 613
submitting form data, 631-632
text fields, 627-628
text input, 617-618
user input, 617-618
graphics
adjusting color, 204
Adobe Photoshop, 198
aligning graphics, 216-220
animated graphics, 211-212
background graphics, 223-224
banners, 206-208
buttons, 206-208
choosing software, 198
clip art, 199
compression, 199
copyrights and, 199
Creative Commons licenses, 199
cropping, 200-202
file sizes, 199
Flickr, 212-214
.GIF, 208-212
GIMP, 198, 201-208
Google Images, 212
Google Picasa, 199
grabbing from web pages, 199
graphics, placing on web pages, 212-214
height, 216
imagemaps, 225-230
.JPEG, 205, 209-211
Pixlr, 199
.PNG, 209
Red Eye Removal, 204
republishing, 214

resizing, 203-204
resolution, 199-200
text descriptions, 214-215
tiled backgrounds, 209-211
transparent graphics, 209
turning graphics into links, 220-223
uses of, 200
width, 216
links
absolute links, 170
anchor tags, 172-175
email addresses, 179-180
images as, 181
linking between, 175-178
linking to external web content, 178-179
opening in new browser windows, 181
relative-root links, 171
styling via CSS, 181-185
lists
CSS box model and, 286-290
horizontal navigation, 307-311
navigation lists, 296-311
placing list item indicators, 290-292
vertical navigation, 296-306
loading, timing loads, 21
managing
coding clarity, 653-654
comments, 652-653
documenting code, 652-653
indenting code, 653-654
maintainable code, 652-654
version control, 654-656
margins
browsers, 287-288
margin property (CSS), 249-257
organizing, 169-170, 641-642
larger websites, 648-651
simple websites, 644-648
single-page websites, 642-643
overflow property (CSS) and text flow, 281
padding
browsers, 287-288
padding property (CSS), 249, 257-261
progressive enhancement, 506-507
publishing, 19-20
random content, displaying via JavaScript, 340-344
relative addresses, 170-171
relative-root addresses, 170
responsive web design, 332-333
style sheets
creating, 57-62
definition of, 55
external style sheets, 56-62
formatting properties, 63-67
formatting text color, 56
inline styles, 71-72
internal style sheets, 56, 70-72
layout properties, 63
linking to HTML documents, 61
selectors, 68
sizing text, 60
style classes, 68-69
style ID, 70
style properties, 68
style rules, 56, 60-62, 69
validating, 72
tables
aligning within, 153-156
borders, 148-149
cells, 146, 151-153
color in, 155-156
creating, 146-150
CSS columns, 158-162
headings, 146
images in, 156
page layout via, 157
rows, 146
sizing, 150-153
spacing borders, 156
spanning within, 156
testing, 20-21
text
adding to web pages, 389-391
flowing, 280-281
modifying, 387-388
text, formatting, 122
aligning text, 135
boldface text, 125-126
customizing fonts in HTML, 128-132
definition lists, 138, 286
emphasized text, 126
foreign languages, 122-124
italic (emphasized) text, 125-126
multitiered lists, 144
nested lists, 140-141, 286
older HTML tags, 122
ordered lists, 138, 286
outlines, 141
resumes, 130-132
special characters, 122-124
subscript text, 126
superscript text, 126
unordered lists, 138, 286
web fonts, 132-134
transferring, FPT, 29
validating, 95-96, 105
viewing, 5, 29
web hosting providers, selecting, 6-8, 26
website architectures, 171
web design, progressive enhancement, 506-507
web fonts, 132-134
web hosting providers
A Small Orange, 7
bandwidth, 7
control panels, 7-8
customer service, 6
DailyRazor, 7

domain names, purchasing, 7
LunarPages, 7
pricing, 7
reliability, 6
selecting, 6-8, 26
server space, 6
web pages. *See also* web content
absolute addresses, 170-171
aligning (CSS)
align property, 249
text-align property, 262
vertical-align property, 262
clear property (CSS) and text flow, 281
color
140 cross-browser color names, 192
best practices, 190-191
Colorblind Web Page Filter tool, 198
hexadecimal color codes, 192-195
names and case sensitivity, 192
using, 190-191
W3C color standards, 192
conceptualizing, 38-43
creating
ASCII text, 27, 34
comparing web page HTML codes, 36
formatting text, 34
HTML tags, 27-33
indenting text, 34
line breaks, 32-33
organizing content via headings, 34-35
overview of, 29
paragraphs, 32-33
plain text, 27, 34
templates, 31
CSS box model, 269
borders, 270
content, 270
lists and, 286-290
margins, 270
padding, 270
sizing elements, 270-272
directories, 169-170
elements, defining, 56
floating elements, 249
float property (CSS), 249, 262-266, 281
folders, 169-171
forms, 611
accessing elements via JavaScript, 633
checkboxes, 621-623
creating, 612-616
displaying data in pop-up windows, 633-635
grouping elements, 619-620
hidden data, 620
JavaScript events, 632
labeling form data, 618-619
naming form data, 618
passwords, 617
pull-down pick lists, 624-626
radio buttons, 623-624
scrolling lists, 624-626
selection lists, 624-626
sending, 613
submitting form data, 631-632
text fields, 627-628
text input, 617-618
user input, 617-618
graphics
adjusting color, 204
Adobe Photoshop, 198
aligning graphics, 216-220
animated graphics, 211-212
background graphics, 223-224
banners, 206-208
buttons, 206-208
choosing software, 198
clip art, 199
compression, 199
copyrights and, 199
Creative Commons licenses, 199
cropping, 200-202
file sizes, 199
Flickr, 212-214
.GIF, 208-212
GIMP, 198, 201-208
Google Images, 212
Google Picasa, 199
grabbing from web pages, 199
height, 216
imagemaps, 225-230
.JPEG, 205, 209-211
links, turning graphics into, 220-223
Pixlr, 199
placing on web pages, 212-214
.PNG, 209
Red Eye Removal, 204
republishing, 214
resizing, 203-204
resolution, 199-200
text descriptions, 214-215
tiled backgrounds, 209-211
transparent graphics, 209
uses of, 200
width, 216
JavaScript, adding to web pages, 79-80
links
absolute links, 170
anchor tags, 172-175
email addresses, 179-180
images as, 181
linking between web pages, 175-178
linking to external web pages, 178-179
opening in new browser windows, 181
relative-root links, 171
styling via CSS, 181-185

lists
CSS box model and, 286-290
horizontal navigation, 307-311
navigation lists, 296-311
placing list item indicators, 290-292
vertical navigation, 296-306
loading, timing loads, 21
managing
coding clarity, 653-654
comments, 652-653
documenting code, 652-653
indenting code, 653-654
maintainable code, 652-654
version control, 654-656
margins
browsers, 287-288
margin property (CSS), 249-257
multimedia, 242
<audio> element, 238-239
audio/video playback via HTML5, 238-241
creating, 232
defining, 189
<embed /> element, 237
embedding files, 235-238
linking to files, 231-233
MIME types, 236
<object> element, 235-237
QuickTime support, 233
streaming files, 235
<video> element, 240-241
organizing, 169-170, 641
larger websites, 648-651
simple websites, 644-648
single-page websites, 642-643
overflow property (CSS) and text flow, 281
padding
browsers and, 287-288
padding property (CSS), 249, 257-261
progressive enhancement, 506-507
random content, displaying via JavaScript, 340-344
relative addresses, 170-171
relative-root addresses, 170
responsive web design, 332-333
scripts, adding to web pages, 86
style sheets
creating, 57-62
definition of, 55
external style sheets, 56-62
formatting properties, 63-67
formatting text color, 56
inline styles, 71-72
internal style sheets, 56, 70-72
layout properties, 63
linking to HTML documents, 61
selectors, 68
sizing text, 60
style classes, 68-69
style ID, 70
style properties, 68
style rules, 56, 60-62, 69
validating, 72
tables
aligning within, 153-156
borders, 148-149
cells, 146, 151-153
color in, 155-156
creating, 146-150
CSS columns, 158-162
headings, 146
images in, 156
page layout via, 157
rows, 146
sizing, 150-153
spacing borders, 156
spanning within, 156
text
adding to web pages, 389-391
flowing, 280-281
modifying, 387-388
text, formatting, 122
aligning text, 135
boldface text, 125-126
customizing fonts in HTML, 128-132
definition lists, 138, 286
emphasized text, 126
foreign languages, 122-124
italic text, 125-126
multitiered lists, 144
nested lists, 140-141, 286
older HTML tags, 122
ordered lists, 138, 286
outlines, 141
resumes, 130-132
special characters, 122-124
subscript text, 126
superscript text, 126
unordered lists, 138, 286
web fonts, 132-134
transferring FTP, 29
validating, 95-96, 105
viewing, 29
website architectures, 171
web scripting. *See* scripting
web servers
document roots, 14-18
FTP client connections, 14
websites
align property (CSS), 249
text-align property (CSS), 262
vertical-align property (CSS), 262
architectures, creating, 171
bad website examples, 193
clear property (CSS) and text flow, 281
color
140 cross-browser color names, 192
best practices, 190-191
Colorblind Web Page Filter tool, 198

hexadecimal color codes, 192-195
names and case sensitivity, 192
using, 190-191
W3C color standards, 192
comparing, 26
connecting to, Classic FTP client, 12-13
CSS box model, 269
borders, 270
content, 270
lists and, 286-290
margins, 270
padding, 270
sizing elements, 270-272
dynamic websites, 337
changing images based on user interaction, 346-347
client-side scripting, 338
DOM, 345-346
JavaScript, 338-347
server-side scripting, 338
VBScript, 338
floating elements, 249
float property (CSS), 249, 262-266, 281
forms, 611-612
accessing elements via JavaScript, 633
checkboxes, 621-623
creating, 612-617
displaying data in pop-up windows, 633-635
grouping elements, 619-620
hidden data, 620
JavaScript events, 632
labeling form data, 618-619
naming form data, 618
passwords, 617
pull-down pick lists, 624-626
radio buttons, 623-624
scrolling lists, 624-626
selection lists, 624-626
sending, 613
submitting form data, 631-632
text fields, 627-628
text input, 617-618
user input, 617-618
graphics
adjusting color, 204
Adobe Photoshop, 198
aligning graphics, 216-220
animated graphics, 211-212
background graphics, 223-224
banners, 206-208
buttons, 206-208
choosing software, 198
clip art, 199
compression, 199
copyrights, 199
Creative Commons licenses, 199
cropping, 200-202
file sizes, 199
Flickr, 212-214
.GIF, 208-212
GIMP, 198, 201-208
Google Images, 212
Google Picasa, 199
grabbing from web pages, 199
height, 216
imagemaps, 225-230
.JPEG, 205, 209-211
Pixlr, 199
placing on web pages, 212-214
.PNG, 209
Red Eye Removal, 204
republishing, 214
resizing, 203-204
resolution, 199-200
specifying height/width, 216
text descriptions, 214-215
tiled backgrounds, 209-211
transparent graphics, 209
turning graphics into links, 220-223
uses of, 200
width, 216
links
absolute links, 170
anchor tags, 172-175
email addresses, 179-180
images as, 181
linking between web pages, 175-178
linking to external web pages, 178-179
opening in new browser windows, 181
relative-root links, 171
styling via CSS, 181-185
lists
CSS box model, 286-290
horizontal navigation, 307-311
navigation lists, 296-311
placing list item indicators, 290-292
vertical navigation, 296-306
managing
coding clarity, 653-654
comments, 652-653
documenting code, 652-653
indenting code, 653-654
maintainable code, 652-654
version control, 654-656
margins
browsers, 287-288
margin property (CSS), 249-257
navigating, 82
organizing, 641-642
Amazon.com, 648-649
BAWSI.org, 651
ESPN.com, 646
larger websites, 648-651
Peet's Coffee & Tea, 649
simple websites, 644-648
single-page websites, 642-643
overflow property (CSS) and text flow, 281

padding
browsers, 287-288
padding property (CSS), 249, 257-261
progressive enhancement, 506-507
single-page websites, 642-643
style sheets
creating, 57-62
definition of, 55
external style sheets, 56-62
formatting properties, 63-67
formatting text color, 56
inline styles, 71-72
internal style sheets, 56, 70-72
layout properties, 63
linking to HTML documents, 61
selectors, 68
sizing text, 60
style classes, 68-69
style IDs, 70
style properties, 68
style rules, 56, 60-62, 69
validating, 72
testing, 8-9, 26
text
adding to web pages, 389-391
flowing text, 280-281
modifying, 387-388
web content
absolute addresses, 170-171
directories, 169-170
folders, 169-171
organizing, 169-170
relative addresses, 170-171
relative-root addresses, 170

web sockets, 349

while loops (JavaScript), 456-457

whitespace (spacing)
HTML, 32
JavaScript syntax, 361

widgets (jQuery UI library), 573-574, 585
draggable() widget, 575-578, 581-584
droppable() widget, 575, 580-584
mouse interaction widget, 574-575

width
CSS box model, adjusting in, 270-272
fixed/liquid hybrid layouts, setting minimum width in, 325-326
images, specifying width in, 216
width property (CSS), 63

window objects (DOM), 368, 489
browser windows
creating, 490-491
moving, 493-495
opening/closing, 491-493
resizing, 493-495
timeouts, 495-497
dialog boxes, displaying, 497-499
properties of, 489-490

windows
browser windows
creating, 490-491
moving, 493-495
non-viewable window areas, 316
opening/closing, 491-493
resizing, 493-495
timeouts, 495-497
pop-up windows, displaying form data in, 633-635

Windows Media Video, linking images to, 232-233

Word, creating HTML files, 27

WordPress Theme Gallery and layouts, 315

WWW (World Wide Web), development of, 2

WYSIWYG (what-you-see-is-what-you-get) editors, creating HTML files, 27

X–Z

XHTML (Extensible HyperText Markup Language), 3, 31, 61

XML (Extensible Markup Language) and AJAX, 537, 591, 598

XMLHttpRequest, 592-594

Yahoo! Developer Network, JavaScript design patterns, 513

Yahoo! UI Library, 509, 534

z-index property (CSS), 278-280

Zen Garden (CSS), 315