Contents

Credits vi

About the Author vii

Acknowledgments ix

Introduction xix

Part 1: JavaScript Basics **1**

- Task 1: Creating a `script` Block 2
- Task 2: Hiding Your JavaScript Code 4
- Task 3: Providing Alternatives to Your JavaScript Code 6
- Task 4: Including Outside Source Code 8
- Task 5: Commenting Your Scripts 10
- Task 6: Writing a JavaScript Command 12
- Task 7: Temporarily Removing a Command from a Script 14
- Task 8: Using Curly Brackets 16
- Task 9: Writing Output to the Browser 18
- Task 10: Creating a Variable 20
- Task 11: Outputting a Variable 22
- Task 12: Creating a String 24
- Task 13: Creating a Numeric Variable 26
- Task 14: Performing Math 28
- Task 15: Concatenating Strings 30
- Task 16: Searching for Text in Strings 32
- Task 17: Replacing Text in Strings 34
- Task 18: Formatting Strings 36
- Task 19: Applying Multiple Formatting Functions to a String 38
- Task 20: Creating Arrays 40
- Task 21: Populating an Array 42
- Task 22: Sorting an Array 44
- Task 23: Splitting a String at a Delimiter 46
- Task 24: Calling Functions 48

Task 25: Alerting the User 50
Task 26: Confirming with the User 52
Task 27: Creating Your Own Functions 54
Task 28: Passing an Argument to Your Functions 56
Task 29: Returning Values from Your Functions 58
Task 30: Passing Multiple Parameters to Your Functions 60
Task 31: Calling Functions from Tags 62
Task 32: Calling Your JavaScript Code after the Page Has Loaded 64
Task 33: Using `for` Loops 66
Task 34: Testing Conditions with `if` 68
Task 35: Using Short-Form Condition Testing 70
Task 36: Looping on a Condition 72
Task 37: Looping through an Array 74
Task 38: Scheduling a Function for Future Execution 76
Task 39: Scheduling a Function for Recurring Execution 78
Task 40: Canceling a Scheduled Function 80
Task 41: Adding Multiple Scripts to a Page 82
Task 42: Calling Your JavaScript Code after the Page Has Loaded 84
Task 43: Check If Java Is Enabled with JavaScript 86

Part 2: Outputting to the Browser 89

Task 44: Accessing the `document` Object 90
Task 45: Outputting Dynamic HTML 92
Task 46: Including New Lines in Output 94
Task 47: Outputting the Date to the Browser 96
Task 48: Outputting the Date and Time in a Selected Time Zone 98
Task 49: Controlling the Format of Date Output 100
Task 50: Customizing Output by the Time of Day 102
Task 51: Generating a Monthly Calendar 104
Task 52: Customizing Output Using URL Variables 106
Task 53: Dynamically Generating a Menu 108
Task 54: Replacing the Browser Document with a New Document 110
Task 55: Redirecting the User to a New Page 112
Task 56: Creating a “Page Loading ...” Placeholder 114

Part 3: Images and Rollovers 117

Task 57: Accessing an HTML-Embedded Image in JavaScript 118
Task 58: Loading an Image Using JavaScript 120
Task 59: Detecting `MouseOver` Events on Images 122

Task 60: Detecting Click Events on Images 124
Task 61: Switching an Image Programatically 126
Task 62: Using Multiple Rollovers in One Page 128
Task 63: Displaying a Random Image 130
Task 64: Displaying Multiple Random Images 132
Task 65: Using a Function to Create a Rollover 134
Task 66: Using a Function to Trigger a Rollover 136
Task 67: Using Functions to Create Multiple Rollovers in One Page 138
Task 68: Creating a Simple Rollover Menu System 140
Task 69: Creating a Slide Show in JavaScript 142
Task 70: Randomizing Your Slide Show 144
Task 71: Triggering Slide Show Transitions from Links 146
Task 72: Including Captions in a Slide Show 148
Task 73: Testing If an Image Is Loaded 150
Task 74: Triggering a Rollover in a Different Location with a Link 152
Task 75: Using Image Maps and Rollovers Together 154
Task 76: Generating Animated Banners in JavaScript 156
Task 77: Displaying a Random Banner Ad 158

Part 4: Working with Forms 161
Task 78: Preparing Your Forms for JavaScript 162
Task 79: Accessing Text Field Contents 164
Task 80: Dynamically Updating Text Fields 166
Task 81: Detecting Changes in Text Fields 168
Task 82: Accessing Selection Lists 170
Task 83: Programmatically Populating a Selection List 172
Task 84: Dynamically Changing Selection List Content 174
Task 85: Detecting Selections in Selection Lists 176
Task 86: Updating One Selection List Based on Selection in Another 178
Task 87: Using Radio Buttons instead of Selection Lists 180
Task 88: Detecting the Selected Radio Button 182
Task 89: Detecting Change of Radio Button Selection 184
Task 90: Updating or Changing Radio Button Selection 186
Task 91: Creating Check Boxes 188
Task 92: Detecting Check Box Selections 190
Task 93: Changing Check Box Selections 192
Task 94: Detecting Changes in Check Box Selections 194
Task 95: Verifying Form Fields in JavaScript 196
Task 96: Using the onSubmit Attribute of the Form Tag to Verify Form Fields 198

Task 97: Verifying Form Fields Using INPUT TYPE="button" Instead of TYPE="submit" 200
Task 98: Validating E-mail Addresses 202
Task 99: Validating Zip Codes 204
Task 100: Validating Phone Numbers 206
Task 101: Validating Credit Card Numbers 208
Task 102: Validating Selection List Choices 210
Task 103: Validating Radio Button Selections 212
Task 104: Validating Check Box Selections 214
Task 105: Validating Passwords 216
Task 106: Validating Phone Numbers with Regular Expressions 218
Task 107: Creating Multiple Form Submission Buttons Using INPUT TYPE="button" Buttons 220
Task 108: Reacting to Mouse Clicks on Buttons 222
Task 109: Using Graphical Buttons in JavaScript 224
Task 110: Controlling the Form Submission URL 226
Task 111: Validating a Numeric Text Field with Regular Expressions 228
Task 112: Encrypting Data before Submitting It 230
Task 113: Using Forms for Automatic Navigation Jumping 232

Part 5: Manipulating Browser Windows 235
Task 114: Using the Window Object 236
Task 115: Popping Up an Alert Dialog Box 238
Task 116: Popping Up Confirmation Dialog Boxes 240
Task 117: Popping Up JavaScript Prompts 242
Task 118: Creating New Browser Windows 244
Task 119: Opening a New Browser Window from a Link 246
Task 120: Setting the Size of New Browser Windows 248
Task 121: Setting the Location of New Browser Windows 250
Task 122: Controlling Toolbar Visibility for New Browser Windows 252
Task 123: Determining the Availability of Scroll Bars for New Browser Windows 254
Task 124: Restricting Resizing of New Browser Windows 256
Task 125: Loading a New Document into a Browser Window 258
Task 126: Controlling Window Scrolling from JavaScript 260
Task 127: Opening a Full-Screen Window in Internet Explorer 262
Task 128: Handling the Parent-Child Relationship of Windows 264
Task 129: Updating One Window's Contents from Another 266
Task 130: Accessing a Form in Another Browser Window 268
Task 131: Closing a Window in JavaScript 270
Task 132: Closing a Window from a Link 272

Task 133: Creating Dependent Windows in Netscape 274
Task 134: Sizing a Window to Its Contents in Netscape 276
Task 135: Loading Pages into Frames 278
Task 136: Updating One Frame from Another Frame 280
Task 137: Sharing JavaScript Code between Frames 282
Task 138: Using Frames to Store Pseudo-Persistent Data 284
Task 139: Using One Frame for Your Main JavaScript Code 286
Task 140: Using a Hidden Frame for Your JavaScript Code 288
Task 141: Working with Nested Frames 290
Task 142: Updating Multiple Frames from a Link 292
Task 143: Dynamically Creating Frames in JavaScript 294
Task 144: Dynamically Updating Frame Content 296
Task 145: Referring to Unnamed Frames Numerically 298

Part 6: Manipulating Cookies 301
Task 146: Creating a Cookie in JavaScript 302
Task 147: Accessing a Cookie in JavaScript 304
Task 148: Displaying a Cookie 306
Task 149: Controlling the Expiry of a Cookie 308
Task 150: Using a Cookie to Track a User's Session 310
Task 151: Using a Cookie to Count Page Access 312
Task 152: Deleting a Cookie 314
Task 153: Creating Multiple Cookies 316
Task 154: Accessing Multiple Cookies 318
Task 155: Using Cookies to Present a Different Home Page for New Visitors 320
Task 156: Creating a Cookie Function Library 322
Task 157: Allowing a Cookie to be Seen for all Pages in a Site 324

Part 7: DHTML and Style Sheets 327
Task 158: Controlling Line Spacing 328
Task 159: Determining an Object's Location 330
Task 160: Placing an Object 332
Task 161: Moving an Object Horizontally 334
Task 162: Moving an Object Vertically 336
Task 163: Moving an Object Diagonally 338
Task 164: Controlling Object Movement with Buttons 340
Task 165: Creating the Appearance of Three-Dimensional Movement 342
Task 166: Centering an Object Vertically 344
Task 167: Centering an Object Horizontally 346

Task 168: Controlling Line Height in CSS 348
Task 169: Creating Drop Shadows with CSS 350
Task 170: Modifying a Drop Shadow 352
Task 171: Removing a Drop Shadow 354
Task 172: Placing a Shadow on a Nonstandard Corner 356
Task 173: Managing Z-Indexes in JavaScript 358
Task 174: Setting Fonts for Text with CSS 360
Task 175: Setting Font Style for Text with CSS 362
Task 176: Controlling Text Alignment with CSS 364
Task 177: Controlling Spacing with CSS 366
Task 178: Controlling Absolute Placement with CSS 368
Task 179: Controlling Relative Placement with CSS 370
Task 180: Adjusting Margins with CSS 372
Task 181: Applying Inline Styles 374
Task 182: Using Document Style Sheets 376
Task 183: Creating Global Style Sheet Files 378
Task 184: Overriding Global Style Sheets for Local Instances 380
Task 185: Creating a Drop Cap with Style Sheets 382
Task 186: Customizing the Appearance of the First Line of Text 384
Task 187: Applying a Special Style to the First Line of Every Element on the Page 386
Task 188: Applying a Special Style to All Links 388
Task 189: Accessing Style Sheet Settings 390
Task 190: Manipulating Style Sheet Settings 392
Task 191: Hiding an Object in JavaScript 394
Task 192: Displaying an Object in JavaScript 396
Task 193: Detecting the Window Size 398
Task 194: Forcing Capitalization with Style Sheet Settings 400
Task 195: Detecting the Number of Colors 402
Task 196: Adjusting Padding with CSS 404

Part 8: Dynamic User Interaction 407

Task 197: Creating a Simple Pull-Down Menu 408
Task 198: Creating Two Pull-Down Menus 410
Task 199: Detecting and Reacting to Selections in a Pull-Down Menu 412
Task 200: Generating a Drop-Down Menu with a Function 414
Task 201: Placing Menu Code in an External File 416
Task 202: Inserting a Prebuilt Drop-Down Menu 418
Task 203: Creating a Floating Window 420
Task 204: Closing a Floating Window 422

Task 205: Resizing a Floating Window 424
Task 206: Moving a Floating Window 426
Task 207: Changing the Content of a Floating Window 428
Task 208: Detecting Drag and Drop 430
Task 209: Moving a Dragged Object in Drag and Drop 432
Task 210: Changing Cursor Styles 434
Task 211: Determining the Current Scroll Position 436
Task 212: Creating an Expanding/Collapsing Menu 438
Task 213: Creating a Highlighting Menu Using Just Text and CSS—No JavaScript 440
Task 214: Creating a Highlighting Menu Using Text, CSS, and JavaScript 442
Task 215: Placing Content Offscreen 444
Task 216: Sliding Content into View 446
Task 217: Creating a Sliding Menu 448
Task 218: Auto-Scrolling a Page 450

Part 9: Handling Events 453
Task 219: Responding to the `onMouseOver` Event 454
Task 220: Taking Action When the User Clicks on an Object 456
Task 221: Responding to Changes in a Form's Text Field 458
Task 222: Responding to a Form Field Gaining Focus with `onFocus` 460
Task 223: Taking Action When a Form Field Loses Focus with `onBlur` 462
Task 224: Post-Processing Form Data with `onSubmit` 464
Task 225: Creating Code to Load When a Page Loads with `onLoad` 466
Task 226: Executing Code When a User Leaves a Page for Another 468
Task 227: Taking Action When a User Makes a Selection in a Selection List 470

Part 10: Bookmarklets 473
Task 228: Downloading and Installing Bookmarklets 474
Task 229: Checking Page Freshness with a Bookmarklet 476
Task 230: Checking for E-mail Links with a Bookmarklet 478
Task 231: E-mailing Selected Text with a Bookmarklet in Internet Explorer 480
Task 232: E-mailing Selected Text with a Bookmarklet in Netscape 482
Task 233: Displaying Images from a Page with a Bookmarklet 484
Task 234: Changing Background Color with a Bookmarklet 486
Task 235: Removing Background Images with a Bookmarklet 488
Task 236: Hiding Images with a Bookmarklet 490
Task 237: Hiding Banners with a Bookmarklet 492
Task 238: Opening All Links in a New Window with a Bookmarklet 494
Task 239: Changing Page Fonts with a Bookmarklet 496

Task 240: Highlighting Page Links with a Bookmarklet 498
Task 241: Checking the Current Date and Time with a Bookmarklet 500
Task 242: Checking Your IP Address with a Bookmarklet 502
Task 243: Searching Yahoo! with a Bookmarklet in Internet Explorer 504
Task 244: Searching Yahoo! with a Bookmarklet in Netscape 506

Part 11: Cross-Browser Compatibility and Issues 509

Task 245: Detecting the Browser Type 510
Task 246: Detecting the Browser Version 512
Task 247: Browser Detection Using Object Testing 514
Task 248: Creating Browser Detection Variables 516
Task 249: Dealing with Differences in Object Placement in Newer Browsers 518
Task 250: Creating Layers with the div Tag 520
Task 251: Controlling Layer Placement in HTML 522
Task 252: Controlling Layer Size in HTML 524
Task 253: Controlling Layer Visibility in HTML 526
Task 254: Controlling Layer Ordering in HTML 528
Task 255: Changing Layer Placement and Size in JavaScript 530
Task 256: Changing Layer Visibility in JavaScript 532
Task 257: Changing Layer Ordering in JavaScript 534
Task 258: Fading Objects 536
Task 259: Creating a Page Transition in Internet Explorer 538
Task 260: Installing the X Cross-Browser Compatibility Library 540
Task 261: Showing and Hiding Elements with X 542
Task 262: Controlling Stacking Order with X 544
Task 263: Changing Text Color with X 546
Task 264: Setting a Background Color with X 548
Task 265: Setting a Background Image with X 550
Task 266: Repositioning an Element with X 552
Task 267: Sliding an Element with X 554
Task 268: Changing Layer Sizes with X 556

Appendix A: JavaScript Quick Reference 559

Appendix B: CSS Quick Reference 593

Index 601

Introduction

Since the mid-1990s when Netscape introduced version 2 of its flagship Netscape Navigator browser, JavaScript has been part of the Web development landscape. Providing a mechanism to implement dynamic interactivity in the browser, without connecting to the server, JavaScript is at the core of the Dynamic HTML model, which allows today's modern browsers to host sophisticated applications and user interfaces.

This book is a recipe book that provides you with quick, digestible examples of how to perform specific tasks using JavaScript. These tasks range from simple tasks such as displaying dynamic output in the browser window to complex tasks such as creating a dynamic, interactive menu system.

This book isn't a tutorial in JavaScript. It is designed to be a useful reference when you are actively engaged in building your Web applications and need quick answers to the question "How do I do this in JavaScript?" For most tasks of low and medium complexity, you will likely find an example in this book. Completing complex tasks can often be achieved by combining more than one sample tasks from the book.

tip

If you don't have any experience with JavaScript, you will probably want to supplement this book with a tutorial introduction to programming in JavaScript. For instance, you might consider *JavaScript for Dummies* by Emily A. Vander Veer (John Wiley & Sons, 0-7645-0633-1).

About the Book

This book is divided into 11 parts:

Part 1: JavaScript Basics

This part provides tasks that illustrate some fundamental JavaScript techniques and skills. If you have never used JavaScript before, this part is for you. It provides examples that illustrate the basics of creating scripts and using JavaScript.

Part 2: Outputting to the Browser

This part covers some core techniques for using JavaScript to generate dynamic output to the browser window, including outputting dynamic values such as dates.

Part 3: Images and Rollovers

Using JavaScript, you can manipulate images, producing effects such as rollover effects and random slide shows. The tasks in this part illustrate techniques for working with images from JavaScript.

Part 4: Working with Forms

Forms involve more than just submitting data to the server. This part illustrates how to create dynamic client-side forms in the browser and to build forms that work with the user without contacting the server.

Part 5: Manipulating Browser Windows

This part provides tasks that illustrate the creation and closing of windows, how to manage the attributes of those windows, and how to work with frames. All these features are key to developing sophisticated user interfaces with JavaScript.

Part 6: Manipulating Cookies

Normally, cookies are created by your server and sent to the browser for storage. The browser then sends them back to the server when the user connects to that server. Now with JavaScript, you can create cookies and access them later without any interaction with the server.

Part 7: DHTML and Style Sheets

JavaScript is part of a threesome that forms Dynamic HTML. The other parts are the Domain Object Model and cascading style sheets. The tasks in this part show you how to work with the DOM and style sheets.

Part 8: Dynamic User Interaction

This part provides tasks that illustrate some of the most popular uses of JavaScript for dynamic user interaction—from creating pull-down menus to producing floating windows and handling drag-and-drop user interaction.

Part 9: Handling Events

JavaScript is an event-driven scripting language. This means you don't create linear programs but instead can write your programs to respond to events. Events might be the user clicking on a button or the completion of a task by the browser, such as completing loading of the current document.

Part 10: Bookmarklets

Bookmarklets are an interesting application of JavaScript that combines JavaScript with the bookmarks or favorites feature of browser. Bookmarklets are short, self-contained JavaScript scripts that perform some useful task that you can add to your favorites or bookmarks and then run at any time by selecting the relevant favorite or bookmark.

Part 11: Cross-Browser Compatibility and Issues

As JavaScript has become more advanced and its features have expanded, browser compatibility has become an issue. As would be expected, different browser vendors have different ideas about the right way to do things in their implementations of JavaScript. The result is a plethora of browsers with subtle differences in the way JavaScript works. The tasks in this part provide you with some techniques for handling these browser differences in your applications.

The appendices provide you quick references to JavaScript and cascading style sheets you can consult in developing your applications when you need reminders of the correct property, method, or style attribute name.

Finally, the complete source code for each task can be found on the companion Web site at www.wiley.com/10stepsorless. This makes it easy for you to try the code illustrated in the task or adapt the code for your own purposes.

Conventions Used in this Book

As you go through this book, you will find a few unique elements. We'll describe those elements here so that you'll understand them when you see them.

Code

If a single line of code is too long to appear as one line in the printed book, we'll add the following symbol to indicate that the line continues: ⮐

Text You Type and Text on the Screen

Whenever you are asked to type in text, the text you are to type appears in bold, like this:

Type in this address: **111 River Street**.

When we are referring to URLs or other text you'll see on the screen, we'll use a monospace font, like this:

Check out `www.wiley.com`.

Icons

A number of special icons appear in the margins of each task to provide additional information you might find helpful.

note

The Note icon is used to provide additional information or help in working in JavaScript.

tip

The Tip icon is used to point out an interesting idea or technique that will save you time, effort, money, or all three.

caution

The Caution icon is used to alert you to potential problems that you might run into when working in JavaScript.

cross-reference

Although this book is divided into tasks to make it easy to find exactly what you're looking for, few tasks really stand completely alone. The Cross-Reference icon provides us the opportunity to point out other tasks in the book you might want to look at if you're interested in this task.