LEARNING MADE EASY

Web Coding & Development

ALL-IN-ONE

for dummies

A Wiley Brand

8 Books in one!

Paul McFedries

Web Coding & Development

ALL-IN-ONE

for dummies®
A Wiley Brand

Web Coding & Development

ALL-IN-ONE

by Paul McFedries

for dummies®
A Wiley Brand

Web Coding & Development All-in-One For Dummies®

Published by: **John Wiley & Sons, Inc.,** 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Published simultaneously in Canada

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit https://hub.wiley.com/community/support/dummies.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2018935649

ISBN: 978-1-119-47392-3; ISBN: 978-1-119-47383-1 (ePDF); ISBN: 978-1-119-47379-4 (ePub)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Contents at a Glance

Introduction 1

Book 1: Getting Ready to Code for the Web 5

CHAPTER 1: How Web Coding and Development Work 7

CHAPTER 2: Setting Up Your Web Development Home 21

CHAPTER 3: Finding and Setting Up a Web Host 35

Book 2: Coding the Front End, Part 1: HTML & CSS 47

CHAPTER 1: Structuring the Page with HTML 49

CHAPTER 2: Styling the Page with CSS 79

CHAPTER 3: Sizing and Positioning Page Elements 103

CHAPTER 4: Creating the Page Layout 127

Book 3: Coding the Front End, Part 2: JavaScript 167

CHAPTER 1: An Overview of JavaScript 169

CHAPTER 2: Understanding Variables 183

CHAPTER 3: Building Expressions 197

CHAPTER 4: Controlling the Flow of JavaScript 225

CHAPTER 5: Harnessing the Power of Functions 249

CHAPTER 6: Working with Objects 267

CHAPTER 7: Working with Arrays 291

CHAPTER 8: Manipulating Strings, Dates, and Numbers 311

CHAPTER 9: Debugging Your Code 341

Book 4: Coding the Front End, Part 3: jQuery 363

CHAPTER 1: Developing Pages Faster with jQuery 365

CHAPTER 2: Livening Up Your Page with Events and Animation 387

CHAPTER 3: Getting to Know jQuery UI 411

Book 5: Coding the Back End: PHP and MySQL 433

CHAPTER 1: Learning PHP Coding Basics 435

CHAPTER 2: Building and Querying MySQL Databases 467

CHAPTER 3: Using PHP to Access MySQL Data 493

Book 6: Coding Dynamic Web Pages 507

CHAPTER 1: Melding PHP and JavaScript with Ajax and JSON 509

CHAPTER 2: Building and Processing Web Forms. 533

CHAPTER 3: Validating Form Data 565

Book 7: Coding Web Apps 591

CHAPTER 1: Planning a Web App. 593

CHAPTER 2: Laying the Foundation. 619

CHAPTER 3: Managing Data 637

CHAPTER 4: Managing App Users 673

Book 8: Coding Mobile Web Apps 721

CHAPTER 1: Exploring Mobile-First Web Development 723

CHAPTER 2: Building a Mobile Web App. 739

Index 769

Table of Contents

INTRODUCTION 1
About This Book. 2
Foolish Assumptions. 3
"I've never coded before!". 3
"I have coded before!". 3
Icons Used in This Book 4
Beyond the Book 4

BOOK 1: GETTING READY TO CODE FOR THE WEB. 5

CHAPTER 1: How Web Coding and Development Work 7
The Nuts and Bolts of Web Coding and Development 8
How the web works. 8
How the web works, take two 11
Understanding the Front End: HTML and CSS .12
Adding structure: HTML 13
Adding style: CSS 14
Understanding the Back End: PHP and MySQL. 15
Storing data on the server: MySQL. 16
Accessing data on the server: PHP 16
How It All Fits Together: JavaScript and jQuery. 16
Front end, meet back end: JavaScript. 16
Making your web coding life easier: jQuery 17
How Dynamic Web Pages Work 18
What Is a Web App?. 19
What Is a Mobile Web App? 19
What's the Difference between Web Coding and Web Development? 20

CHAPTER 2: Setting Up Your Web Development Home 21
What Is a Local Web Development Environment? 22
Do You Need a Local Web Development Environment? 22
Setting Up the XAMPP for Windows Development Environment. 23
Installing XAMPP for Windows. 24
Running the XAMPP for Windows Control Panel 26
Accessing your local web server 27
Setting Up the XAMPP for OS X Development Environment. 29
Installing XAMPP for OS X. 29
Running the XAMPP Application Manager. .30
Accessing your local web server 31
Choosing Your Text Editor 33

CHAPTER 3: **Finding and Setting Up a Web Host** 35
Understanding Web Hosting Providers 36
Using your existing Internet provider 36
Finding a free hosting provider 37
Signing up with a commercial hosting provider 37
A Buyer's Guide to Web Hosting 37
Finding a Web Host 40
Finding Your Way around Your New Web Home 41
Your directory and your web address 42
Making your hard disk mirror your web home 42
Uploading your site files 44
Making changes to your web files 45

BOOK 2: CODING THE FRONT END, PART 1: HTML & CSS 47

CHAPTER 1: **Structuring the Page with HTML** 49
Getting the Hang of HTML 50
Understanding Tag Attributes 52
Learning the Fundamental Structure of an HTML5 Web Page 53
Giving your page a title 54
Adding some text 56
Some Notes on Structure versus Style 57
Applying the Basic Text Tags 58
Emphasizing text 58
Marking important text 59
Nesting tags 60
Adding headings 60
Adding quotations 61
Creating Links 62
Linking basics 62
Anchors aweigh: Internal links 63
Building Bulleted and Numbered Lists 65
Making your point with bulleted lists 65
Numbered lists: Easy as one, two, three 67
Inserting Special Characters 68
Inserting Images 69
Carving Up the Page 71
The <header> tag 71
The <nav> tag 72
The <main> tag 73
The <article> tag 74
The <section> tag 74
The <aside> tag 75

The <footer> tag 75
Handling non-semantic content with <div> 76
Handling words and characters with <span> 77

CHAPTER 2: **Styling the Page with CSS** 79
Figuring Out Cascading Style Sheets 80
Styles: Bundles of formatting options 80
Sheets: Collections of styles 80
Cascading: How styles propagate 81
Getting the Hang of CSS Rules and Declarations 81
Adding Styles to a Page 83
Inserting inline styles 83
Embedding an internal style sheet 84
Linking to an external style sheet 86
Styling Page Text 87
Setting the type size 87
Getting comfy with CSS measurement units 88
Applying a font family 89
Making text bold 91
Styling text with italics 91
Styling links 91
Aligning paragraph text 92
Indenting a paragraph’s first line 92
Working with Colors 93
Specifying a color 93
Coloring text 94
Coloring the background 94
Getting to Know the Web Page Family 95
Using CSS Selectors 96
The class selector 97
The id selector 98
The descendant selector 99
The child selector 99
Revisiting the Cascade 100

CHAPTER 3: **Sizing and Positioning Page Elements** 103
Learning about the CSS Box Model 104
Styling Sizes 105
Adding Padding 107
Building Borders 109
Making Margins 110
Resetting the padding and margin 111
Collapsing margins ahead! 111
Getting a Grip on Page Flow 113

Floating Elements 115
- Clearing your floats 116
- Collapsing containers ahead! 117

Positioning Elements 120
- Using relative positioning 121
- Giving absolute positioning a whirl 122
- Trying out fixed positioning 125

CHAPTER 4: **Creating the Page Layout** 127

What Is Page Layout? 128

Laying Out Page Elements with Floats 128

Laying Out Page Elements with Inline Blocks 132

Making Flexible Layouts with Flexbox 136
- Setting up the flex container 137
- Aligning flex items along the primary axis 139
- Aligning flex items along the secondary axis 140
- Centering an element horizontally and vertically 141
- Laying out a navigation bar with flexbox 143
- Allowing flex items to grow 144
- Allowing flex items to shrink 146
- Laying out content columns with flexbox 149
- Flexbox browser support 152

Shaping the Overall Page Layout with CSS Grid 153
- Setting up the grid container 154
- Specifying the grid rows and columns 154
- Creating grid gaps 155
- Assigning grid items to rows and columns 157
- Aligning grid items 160
- Laying out content columns with Grid 161
- Grid browser support 163

Providing Fallbacks for Page Layouts 164

BOOK 3: CODING THE FRONT END, PART 2: JAVASCRIPT 167

CHAPTER 1: **An Overview of JavaScript** 169

JavaScript: Controlling the Machine 170

What Is a Programming Language? 171

Is JavaScript Hard to Learn? 172

What Can You Do with JavaScript? 173

What Can't You Do with JavaScript? 174

What Do You Need to Get Started? 175

Basic Script Construction 175
- The <script> tag 175
- Handling browsers with JavaScript turned off 176

Where do you put the <script> tag? 176
Example #1: Displaying a message to the user 177
Example #2: Writing text to the page 179
Adding Comments to Your Code 180
Creating External JavaScript Files 181

CHAPTER 2: **Understanding Variables** 183
What Is a Variable? 184
Declaring a variable 184
Storing a value in a variable 185
Using variables in statements 186
Naming Variables: Rules and Best Practices 187
Rules for naming variables 187
Ideas for good variable names 188
Understanding Literal Data Types 189
Working with numeric literals 189
Working with string literals 191
Working with Boolean literals 193
JavaScript Reserved Words 193
JavaScript Keywords 194

CHAPTER 3: **Building Expressions** 197
Understanding Expression Structure 197
Building Numeric Expressions 199
A quick look at the arithmetic operators 199
Using the addition (+) operator 200
Using the increment (++) operator 200
Using the subtraction and negation (-) operators 201
Using the decrement (--) operator 202
Using the multiplication (*) operator 202
Using the division (/) operator 202
Using the modulus (%) operator 204
Using the arithmetic assignment operators 204
Building String Expressions 205
Building Comparison Expressions 208
The comparison operators 208
Using the equal (==) operator 208
Using the not equal (!=) operator 209
Using the greater than (>) operator 209
Using the less than (<) operator 209
Using the greater than or equal (>=) operator 210
Using the less than or equal (<=) operator 210
The comparison operators and data conversion 211
Using the identity (===) operator 212
Using the non-identity (!==) operator 212

- Using strings in comparison expressions .213
- Using the ternary (?:) operator 214

Building Logical Expressions 215

- The logical operators 215
- Using the AND (&&) operator 215
- Using the OR (||) operator. 216
- Using the NOT (!) Operator 217
- Advanced notes on the && and || operators. 217

Understanding Operator Precedence 219

- The order of precedence 220
- Controlling the order of precedence 221

CHAPTER 4: **Controlling the Flow of JavaScript** 225

Understanding JavaScript's Control Structures. 226

Making True/False Decisions with if() Statements 226

Branching with if(). . .else Statements 228

Making Multiple Decisions 229

- Using the AND (??) and OR (||) operators 230
- Nesting multiple if() statements 230
- Using the switch() statement 231

Understanding Code Looping 234

Using while() Loops 235

Using for() Loops 237

Using do. . .while() Loops 241

Controlling Loop Execution 243

- Exiting a loop using the break statement .243
- Bypassing loop statements using the continue statement. 245

Avoiding Infinite Loops. 246

CHAPTER 5: **Harnessing the Power of Functions** 249

What Is a Function?. 250

The Structure of a Function 250

Where Do You Put a Function? 251

Calling a Function 252

- Calling a function when the <script> tag is parsed. 252
- Calling a function after the page is loaded 253
- Calling a function in response to an event 254

Passing Values to Functions. 255

- Passing a single value to a function 256
- Passing multiple values to a function. 257

Returning a Value from a Function. 258

Understanding Local versus Global Variables. 259

- Working with local scope 260
- Working with global scope. 261

Using Recursive Functions 262

CHAPTER 6: **Working with Objects** 267
What Is an Object? 267
The JavaScript Object Hierarchy 269
Manipulating Object Properties 271
Referencing a property 271
Some objects are properties 272
Changing the value of a property 273
Working with Object Methods 273
Playing Around with the window Object 275
Referencing the window object 275
Some window object properties you should know 275
Working with JavaScript timeouts and intervals 276
Interacting with the user 280
Programming the document Object 284
Specifying an element 284
Working with elements 287

CHAPTER 7: **Working with Arrays** 291
What Is an Array? 291
Declaring an Array 293
Populating an Array with Data 294
Declaring and populating an array at the same time 295
Using a loop to populate an array 296
Using a loop to work with array data 297
Creating Multidimensional Arrays 299
Using the Array Object 300
The length property 300
Concatenating to create a new array: concat() 301
Creating a string from an array's elements: join() 302
Removing an array's last element: pop() 303
Adding elements to the end of an array: push() 303
Reversing the order of an array's elements: reverse() 304
Removing an array's first element: shift() 305
Returning a subset of an array: slice() 305
Ordering array elements: sort() 306
Removing, replacing, and inserting elements: splice() 308
Inserting elements at the beginning of an array: unshift() 310

CHAPTER 8: **Manipulating Strings, Dates, and Numbers** 311
Manipulating Text with the String Object 311
Determining the length of a string 312
Finding substrings 313
Methods that extract substrings 315

Dealing with Dates and Times 323
Arguments used with the Date object 324
Working with the Date object 324
Extracting information about a date. 325
Setting the date 330
Performing date calculations. 332
Working with Numbers: The Math Object 335
Converting between strings and numbers 336
The Math object's properties and methods. 338

CHAPTER 9: **Debugging Your Code** 341
Understanding JavaScript's Error Types 342
Syntax errors 342
Runtime errors. 342
Logic errors. 343
Getting to Know Your Debugging Tools 344
Debugging with the Console 345
Displaying the console in various browsers 346
Logging data to the Console 346
Executing code in the Console. 347
Pausing Your Code 348
Entering break mode 348
Exiting break mode 350
Stepping through Your Code 350
Stepping into some code 351
Stepping over some code. 351
Stepping out of some code 352
Monitoring Script Values 352
Viewing a single variable value 352
Viewing all variable values 353
Adding a watch expression 354
More Debugging Strategies 355
Top Ten Most Common JavaScript Errors 356
Top Ten Most Common JavaScript Error Messages 359

BOOK 4: CODING THE FRONT END, PART 3: jQUERY 363

CHAPTER 1: **Developing Pages Faster with jQuery** 365
Getting Started with jQuery 366
How to include jQuery in your web page. 366
Understanding the $ function 368
Where to put jQuery code 368

Selecting Elements with jQuery . . . 369
Using the basic selectors . . . 370
Working with jQuery sets . . . 371
Manipulating Page Elements with jQuery . . . 373
Adding an element . . . 374
Replacing an element's HTML . . . 375
Replacing an element's text . . . 376
Removing an element . . . 377
Modifying CSS with jQuery . . . 377
Working with CSS properties . . . 378
Manipulating classes . . . 382
Tweaking HTML Attributes with jQuery . . . 385
Reading an attribute value . . . 385
Setting an attribute value . . . 385
Removing an attribute . . . 386

CHAPTER 2: **Livening Up Your Page with Events and Animation** . . . 387
Building Reactive Pages with Events . . . 388
What's an event? . . . 388
Understanding the event types . . . 389
Setting up an event handler . . . 390
Using jQuery's shortcut event handlers . . . 391
Getting data about the event . . . 393
Preventing the default event action . . . 394
Getting your head around event delegation . . . 396
Turning off an event handler . . . 398
Building Lively Pages with Animation . . . 398
Hiding and showing elements . . . 399
Fading elements out and in . . . 400
Sliding elements . . . 401
Controlling the animation duration and pace . . . 402
Example: Creating a web page accordion . . . 403
Animating CSS properties . . . 406
Running code when an animation ends . . . 408

CHAPTER 3: **Getting to Know jQuery UI** . . . 411
What's the Deal with jQuery UI? . . . 412
Getting Started with jQuery UI . . . 413
Working with the jQuery UI Widgets . . . 415
Dividing content into tabs . . . 415
Creating a navigation menu . . . 418
Displaying a message in a dialog . . . 420
Hiding and showing content with an accordion . . . 422

Introducing jQuery UI Effects. 424
Applying an effect 424
Checking out the effects. 426
Taking a Look at jQuery UI Interactions. 428
Applying an interaction. 428
Trying out the interactions. 429

BOOK 5: CODING THE BACK END: PHP AND MYSQL 433

CHAPTER 1: **Learning PHP Coding Basics** 435
Understanding How PHP Scripts Work 436
Learning the Basic Syntax of PHP Scripts 436
Declaring PHP Variables. 438
Building PHP Expressions. 438
Outputting Text and Tags. 439
Adding line breaks. 440
Mixing and escaping quotation marks. 441
Outputting variables in strings 442
Outputting long strings. 443
Outputting really long strings 444
Working with PHP Arrays 445
Declaring arrays. 445
Giving associative arrays a look. 446
Outputting array values 447
Sorting arrays. 448
Looping through array values 450
Creating multidimensional arrays. 450
Controlling the Flow of Your PHP Code 451
Making decisions with if(). 452
Making decisions with switch() 453
Looping with while() 454
Looping with for() 455
Looping with do. . .while(). 456
Working with PHP Functions 456
Passing values to functions 457
Returning a value from a function 458
Working with PHP Objects 458
Rolling your own objects 458
Creating an object 461
Working with object properties. 461
Working with object methods 462

Debugging PHP 463
Configuring php.ini for debugging 463
Accessing the PHP error log 464
Debugging with echo statements 465
Debugging with var_dump() statements 466

CHAPTER 2: **Building and Querying MySQL Databases** 467
What Is MySQL? 468
Tables: Containers for your data 468
Queries: Asking questions of your data 469
Introducing phpMyAdmin 470
Importing data into MySQL 471
Backing up MySQL data 473
Creating a MySQL Database and Its Tables 473
Creating a MySQL database 473
Designing your table 474
Creating a MySQL table 477
Adding data to a table 479
Creating a primary key 479
Querying MySQL Data 480
What Is SQL? 480
Creating a SELECT query 481
Understanding query criteria 482
Querying multiple tables 485
Adding table data with an INSERT query 490
Modifying table data with an UPDATE query 491
Removing table data with a DELETE query 492

CHAPTER 3: **Using PHP to Access MySQL Data** 493
Understanding the Role of PHP and MySQL in Your Web App 494
Using PHP to Access MySQL Data 495
Parsing the query string 495
Connecting to the MySQL database 497
Creating and running the SELECT query 499
Storing the query results in an array 500
Looping through the query results 501
Incorporating query string values in the query 501
Creating and Running Insert, Update, and Delete Queries 504
Separating Your MySQL Login Credentials 505

BOOK 6: CODING DYNAMIC WEB PAGES 507

CHAPTER 1: Melding PHP and JavaScript with Ajax and JSON 509
What Is Ajax? 510
Making Ajax Calls with jQuery 511
Learning more about GET and POST requests 511
Handling POST requests in PHP 513
Using .load() to update an element with server data. 514
Using .get() or .post() to communicate with the server 523
Introducing JSON 526
Learning the JSON syntax. 526
Declaring and using JSON variables 527
Returning Ajax Data as JSON Text. 528
Converting server data to the JSON format. 528
Handling JSON data returned by the server 530

CHAPTER 2: Building and Processing Web Forms 533
What Is a Web Form? 534
Understanding How Web Forms Work. 535
Building an HTML5 Web Form. 536
Setting up the form 536
Adding a form button 537
Working with text fields 538
Coding checkboxes 543
Working with radio buttons 548
Adding selection lists 551
Programming pickers 555
Handling and Triggering Form Events 557
Setting the focus 558
Monitoring the focus event 559
Blurring an element 559
Monitoring the blur event 560
Listening for element changes 560
Submitting the Form 561
Triggering the submit event. 562
Preventing the default form submission .562
Preparing the data for submission 563
Submitting the form data 563

CHAPTER 3: Validating Form Data 565
Validating Form Data in the Browser 566
Making a form field mandatory. 566
Restricting the length of a text field 567

Setting maximum and minimum values on a numeric field 568
Validating email fields 569
Making field values conform to a pattern 570
Styling invalid fields 571
Validating Form Data on the Server 574
Checking for required fields 575
Validating text data 578
Validating a field based on the data type .580
Validating against a pattern 582
Regular Expressions Reference 582

BOOK 7: CODING WEB APPS 591

CHAPTER 1: **Planning a Web App** 593
What Is a Web App? 594
Planning Your Web App: The Basics 595
What is my app's functionality? 595
What are my app's data requirements? 596
How will my app work? 597
How many pages will my app require? 597
What will my app's pages look like? 598
Planning Your Web App: Responsiveness 599
Planning Your Web App: Accessibility 605
Planning Your Web App: Security 608
Understanding the dangers 609
Defending your web app 612

CHAPTER 2: **Laying the Foundation** 619
Setting Up the Directory Structure 620
Setting up the public subdirectory 621
Setting up the private subdirectory 623
Creating the Database and Tables 624
Getting Some Back-End Code Ready 626
Defining PHP constants 626
Understanding PHP sessions 627
Securing a PHP session 628
Including code from another PHP file 629
Creating the App Startup Files 630
Creating the back-end initialization file 631
Creating the front-end common files 633
Building the app home page 635

CHAPTER 3: **Managing Data** 637
Handling Data the CRUD Way 638
Starting the web app's data class 639
Creating a data handler script 640
Creating New Data 643
Building the form 643
Sending the form data to the server 648
Adding the data item 649
Reading and Displaying Data 652
Getting the home page ready for data 652
Making an Ajax request for the data 654
Reading the data 655
Displaying the data 656
Filtering the data 657
Updating and Editing Data 661
Deleting Data 668

CHAPTER 4: **Managing App Users** 673
Configuring the Home Page 674
Setting Up the Back End to Handle Users 677
Starting the web app's user class 678
Creating a user handler script 679
Signing Up a New User 682
Building the form 683
Sending the data to the server 685
Sending a verification email 688
Adding the user to the database 689
Verifying the user 690
Signing a User In and Out 696
Checking for a signed-in user 696
Adding the form 697
Checking the user's credentials 700
Signing out a user 704
Resetting a Forgotten Password 704
Deleting a User 714

BOOK 8: CODING MOBILE WEB APPS 721

CHAPTER 1: **Exploring Mobile-First Web Development** 723
What Is Mobile-First Web Development? 724
Learning the Principles of Mobile-First Development 725
Mobile first means content first 725
Pick a testing width that makes sense for your site 726
Get your content to scale with the device 726

Build your CSS the mobile-first way 727
Pick a “non-mobile” breakpoint that makes sense for your content. 727
Going Mobile Faster with jQuery Mobile 729
What is jQuery Mobile?. 729
Adding jQuery Mobile to your web app 730
Working with Images in a Mobile App 731
Making images responsive. 731
Delivering images responsively 732
Storing User Data in the Browser 734
Understanding web storage 735
Adding data to storage 735
Getting data from web storage 736
Removing data from web storage. 737

CHAPTER 2: **Building a Mobile Web App** 739
Building the Button Builder App 740
Getting Some Help from the Web. 741
Building the App: HTML 741
Setting up the home page skeleton 741
Configuring the header. 744
Creating the app menu. 745
Adding the app’s controls. 745
Building the App: CSS 754
Building the App: JavaScript and jQuery 757
Setting up the app data structures 757
Setting the app’s control values 758
Getting the app’s control values 761
Writing the custom CSS code. 763
Running the code. 765
Saving the custom CSS 765
Copying the custom CSS. 766
Resetting the CSS to the default 767

INDEX 769

Introduction

When the web first came to the attention of the world's non-geeks back in the mid-1990s, the vastness and variety of its treasures were a wonder to behold. However, it didn't take long before a few courageous and intrepid souls dug a little deeper into this phenomenon and discovered something truly phenomenal: *They* could make web pages, too!

Why was that so amazing? Well, think back to those old days and think, in particular, of what it meant to create what we now call *content*. Think about television shows, radio programs, magazines, newspapers, books, and the other media of the time. The one thing they all had in common was that their creation was a decidedly *un*common thing. It required a *team* of professionals, a *massive* distribution system, and a *lot* of money. In short, it wasn't something that your average Okie from Muskogee would have any hope of duplicating.

The web appeared to change all of that because learning HTML was within the grasp of anybody who could feed himself, it had a built-in massive distribution system (the Internet, natch), and it required little or no money. For the first time in history, content was democratized and was no longer defined as the sole province of governments and mega-corporations.

Then reality set in.

People soon realized that merely building a website wasn't enough to attract "eyeballs," as the marketers say. A site had to have interesting, useful, or fun content, or people would stay away in droves. Not only that, but this good content had to be combined with a solid site design, which meant that web designers needed a thorough knowledge of HTML and CSS.

But, alas, eventually even all of that was not enough. To make their websites dynamic and interesting, to make their sites easy to navigate, and to give their sites those extra bells and whistles that surfers had come to expect, something more than content, HTML, and CSS was needed.

That missing link was *code*.

What we've all learned the hard way over the past few years is that you simply can't put together a world-class website unless you have some coding prowess in your site design toolkit. You need to know how to program your way out of the basic problems that afflict most sites; how to use scripting to go beyond the inherent limitations of HTML and CSS; and how to use code to send and receive data from a web server. And it isn't enough just to copy the generic scripts that are available on the web and paste them into your pages. First of all, most of those scripts are very poorly written, and second of all, they invariably need some customization to work properly on your site.

About This Book

My goal in this book is to give you a complete education on web coding and development. You learn how to set up the tools you need, how to use HTML and CSS to design and build your site, how to use JavaScript and jQuery to program your pages, and how to use PHP and MySQL to program your web server. My aim is to show you that these technologies aren't hard to learn, and that even the greenest rookie programmers can learn how to put together web pages that will amaze their family and friends (and themselves).

If you're looking for lots of programming history, computer science theory, and long-winded explanations of concepts, I'm sorry but you won't find it here. My philosophy throughout this book comes from Linus Torvalds, the creator of the Linux operating system: "Talk is cheap. Show me the code." I explain what needs to be explained and then I move on without further ado (or, most of the time, without any ado at all) to examples and scripts that do more to illuminate a concept that any verbose explanations I could muster (and believe me, I can muster verbosity with the best of them).

How you approach this book depends on your current level of web coding expertise (or lack thereof):

- If you're just starting out, begin at the beginning with Book 1 and work at your own pace sequentially through to Books 2 and 3. This will give you all the knowledge you need to pick and choose what you want to learn throughout the rest of the book.
- If you know HTML and CSS, you can probably get away with taking a fast look at Book 2, then settle in with Book 3 and beyond.
- If you've done some JavaScript coding already, I suggest working quickly through the material in Book 3, then dig into Book 4 a little slower if you don't already know jQuery. You'll then be ready to branch out and explore the rest of the book as you see fit.
- If you're a relatively experienced JavaScript programmer, use Books 3 and 4 as a refresher, then tackle Book 5 to learn how to code the back end. I've got a few tricks in there that you might find interesting. After that, feel free to

> consider the rest of the book a kind of coding smorgasbord that you can sample as your web development taste buds dictate.

Foolish Assumptions

This book is not a primer on the Internet or on using the World Wide Web. This is a coding and development book, pure and simple. This means I assume the following:

- You know how to operate a basic text editor, and how to get around the operating system and file system on your computer.
- You have an Internet connection.
- You know how to use your web browser.

Yep, that's it.

"I've never coded before!"

If you've never done a stitch of computer programming before, even if you're not quite sure what programming really is, don't worry about it for a second because I had you in mind when I wrote this book. For too many years programming has been the property of "hackers" and other technowizards. That made some sense because the programming languages they were using — with bizarre names such as C++ and Perl — were exceedingly difficult to learn, and even harder to master.

This book's main coding technologies — HTML, CSS, JavaScript, jQuery, PHP, and MySQL — are different. They're nowhere near as hard to learn as those for-nerds-only languages. I honestly believe that *anyone* can become a savvy and successful web coder, and this book is, I hope, the proof of that assertion. Just follow along, examine my code carefully (particularly in the first few chapters), and practice what you learn, and you *will* master web coding and development.

"I have coded before!"

What if you've done some programming in the past? For example, you might have dipped a toe or two in the JavaScript waters already, or you might have dabbled with HTML and CSS. Will this book be too basic for you? No, not at all. My other main goal in this book is to provide you with a ton of truly *useful* examples that you can customize and incorporate into your own site. The book's first few chapters start slowly to avoid scaring off those new to this programming business. But once you get past the basics, I introduce you to lots of great techniques and tricks that will take your web coding skills to a higher level.

Icons Used in This Book

REMEMBER

This icon points out juicy tidbits that are likely to be repeatedly useful to you — so please don't forget them.

TIP

Think of these icons as the fodder of advice columns. They offer (hopefully) wise advice or a bit more information about a topic under discussion.

WARNING

Look out! In this book, you see this icon when I'm trying to help you avoid mistakes that can cost you time, money, or embarrassment.

TECHNICAL STUFF

When you see this icon, you've come across material that isn't critical to understand but will satisfy the curious. Think "inquiring minds want to know" when you see this icon.

Beyond the Book

Some extra content for this book is available on the web. Go online to find the following:

- **The examples used in the book:** You can find these here:

  ```
  mcfedries.com/webcodingfordummies
  ```

  The examples are organized by book and then by chapter within each book. For each example, you can view the code, copy it to your computer's clipboard, and run the code in the browser.

- **The WebDev Workshop:** To edit the book's examples and try your own code and see instant results, fire up the following site:

  ```
  webdev.mcfedries.com
  ```

  You won't break anything, so feel free to use the site run some experiments and play around with HTML, CSS, JavaScript, and jQuery.

1

Getting Ready to Code for the Web

Contents at a Glance

CHAPTER 1: **How Web Coding and Development Work** 7

CHAPTER 2: **Setting Up Your Web Development Home** 21

CHAPTER 3: **Finding and Setting Up a Web Host** 35

IN THIS CHAPTER

- Learning how the web works
- Understanding the front-end technologies of HTML and CSS
- Understanding the back-end technologies of MySQL and PHP
- Figuring out how JavaScript fits into all of this
- Learning about dynamic web pages, web apps, and mobile web apps

Chapter 1

How Web Coding and Development Work

More than mere consumers of technology, we are makers, adapting technology to our needs and integrating it into our lives.

— DALE DOUGHERTY

The 1950s were a hobbyist's paradise with magazines such as *Mechanix Illustrated* and *Popular Science* showing the do-it-yourselfer how to build a go-kart for the kids and how to soup up a lawnmower with an actual motor! Sixty years later, we're now firmly entrenched in the age of do-it-yourself tech, where folks indulge their inner geek to engage in various forms of digital tinkering and hacking. The personification of this high-tech hobbyist renaissance is the *maker,* a modern artisan who lives to create things, rather than merely consume them. Today's makers exhibit a wide range of talents, but the skill most sought-after not only by would-be makers themselves, but by the people who hire them, is web coding and development.

Have you ever visited a website and thought, "Hey, I can do better than that!"? Have you found yourself growing tired of merely reading text and viewing images that someone else has put on the web? Is there something creative in you — stories, images, expertise, opinions — that you want to share with the world? If you answered a resounding "Yes!" to any of these questions, then congratulations: You have everything you need to get started with web coding and development. You have, in short, the makings of a maker.

The Nuts and Bolts of Web Coding and Development

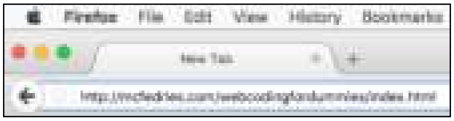
If, as the King said very gravely in Lewis Carroll's *Alice in Wonderland*, it's best to "begin at the beginning," then you've come to the right place. My goal here is to get you off on the right foot by showing you what web coding and web development are.

How the web works

Before you can understand web coding and development, you need to take a step back and understand a bit about how the web itself works. In particular, you need to know what happens behind the scenes when you click a link or type a web page address into your browser. Fortunately, you don't need to be a network engineer to understand this stuff, because I can explain the basics without much in the way of jargon. Here's a high-level blow-by-blow of what happens:

1. **You tell the web browser the web page you want to visit.**

   You do that either by clicking a link to the page or by typing the location — known as the *uniform resource locator* or *URL* (usually pronounced "you-are-ell," but also sometimes "earl") — into the browser's address bar (see Figure 1-1).



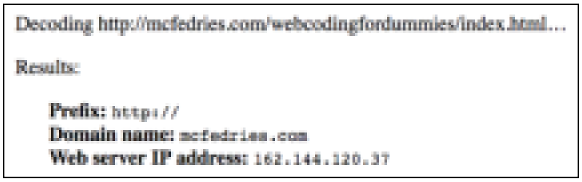
http://mcfedries.com/webcodingfordummies/index.html

FIGURE 1-1: One way to get to a web page is to type the URL in the browser's address bar.

2. **The browser decodes the URL.**

   Decoding the URL means two things: First, it checks the prefix of the URL to see what type of resource you're requesting; this is usually `http://` or `https://`, both of which indicate that the resource is a web page. Second, it gets the

URL's domain name — the `something.com` or `whatever.org` part — and asks the *domain name system* (DNS) to translate this into a unique location — called the IP (Internet Protocol) address — for the web server that hosts the page (see Figure 1-2).



Decoding http://mcfedries.com/webcodingfordummies/index.html…

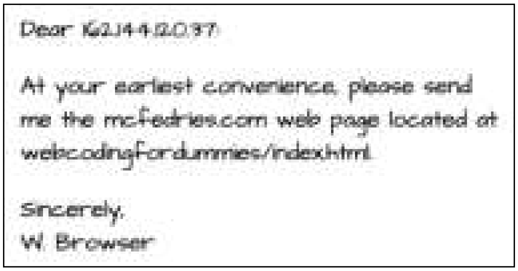
Results:

Prefix: `http://`
Domain name: `mcfedries.com`
Web server IP address: `162.144.120.37`

FIGURE 1-2: The browser extracts the prefix, domain, and the server address from the URL.

3. **The browser contacts the web server and requests the web page.**

   With the web server's unique IP address in hand, the web browser sets up a communications channel with the server and then uses that channel to send along a request for the web page (see Figure 1-3).



Dear 162.144.120.37

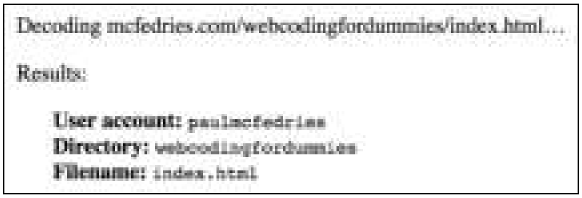
At your earliest convenience, please send me the mcfedries.com web page located at webcodingfordummies/index.html

Sincerely,
W. Browser

FIGURE 1-3: The browser asks the web server for the web page.

4. **The web server decodes the page request.**

   Decoding the page request involves a number of steps. First, if the web server is shared between multiple user accounts, the server begins by locating the user account that owns the requested page. The server then uses the page address to find the directory that holds the page and the file in which the page code is stored (see Figure 1-4).



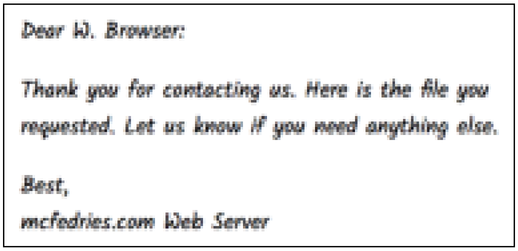
Decoding mcfedries.com/webcodingfordummies/index.html…

Results:

User account: `paulmcfedries`
Directory: `webcodingfordummies`
Filename: `index.html`

FIGURE 1-4: The server uses the page request to get the account, directory, and filename.

5. **The web server sends the web page file to the web browser (see Figure 1-5).**



Dear W. Browser:

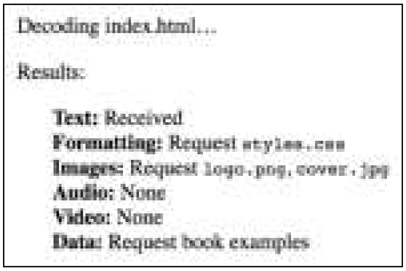
Thank you for contacting us. Here is the file you requested. Let us know if you need anything else.

Best,
mcfedries.com Web Server

FIGURE 1-5: The web server sends the requested web page file to the browser.

6. **The web browser decodes the web page file.**

   Decoding the page file means looking for text to display, instructions on how to display that text, and other resources required by the page, such as images and fonts (see Figure 1-6).



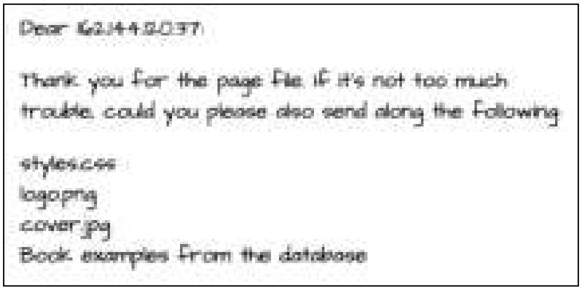
Decoding index.html…

Results:

Text: Received
Formatting: Request `styles.css`
Images: Request `logo.png`, `cover.jpg`
Audio: None
Video: None
Data: Request book examples

FIGURE 1-6: The web browser scours the page file to see if it needs anything else from the server.

7. **If the web page requires more resources, the web browser asks the server to pass along those resources (see Figure 1-7).**



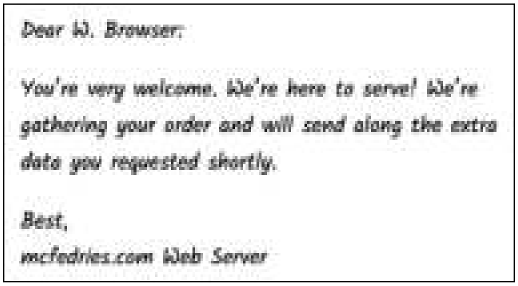
Dear 62.144.0.037:

Thank you for the page file. If it's not too much trouble, could you please also send along the following:

styles.css
logo.png
cover.jpg
Book examples from the database

FIGURE 1-7: The web browser goes back to the server to ask for the other data needed to display the web page.

8. **For each of the requested resources, the web server locates the associated file and sends it to the browser (see Figure 1-8).**



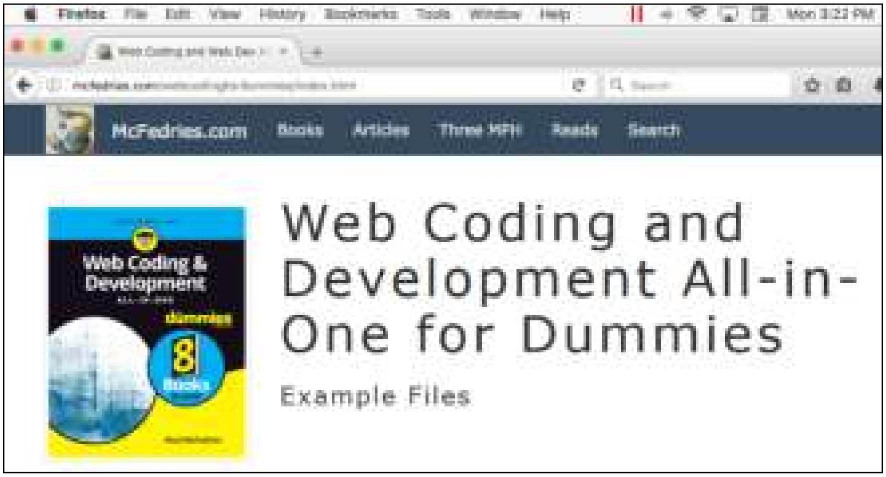
Dear W. Browser:

You're very welcome. We're here to serve! We're gathering your order and will send along the extra data you requested shortly.

Best,
mcfedries.com Web Server

FIGURE 1-8: The web server sends the browser the rest of the requested files.

9. **The web browser gathers up all the text, images, and other resources and displays the page in all its digital splendor in the browser's content window (see Figure 1-9).**



Web Coding and Development All-in-One for Dummies

Example Files

FIGURE 1-9: At long last, the web browser displays the web page.

How the web works, take two

Another way to look at this process is to think of the web as a giant mall or shopping center, where each website is a storefront in that mall. When you request a web page from a particular site, the browser takes you into that site's store and asks the clerk for the web page. The clerk goes into the back of the store, locates the page, and hands it to the browser. The browser checks the page and asks for any other needed files, which the clerk retrieves from the back. This process is repeated until the browser has everything it needs, and it then puts all the page pieces together for you, right there in the front of the store.

This metaphor might seem a bit silly, but it serves to introduce yet another metaphor, which itself illustrates one of the most important concepts in web development. In the same way that our website store has a front and a back, so, too, is web development separated into a front end and a back end:

- **Front end:** That part of the web page that the web browser displays in the browser window. That is, it's the page stuff you see and interact with.
- **Back end:** That part of the web page that resides on the web server. That is, it's the page stuff that the server gathers based on the requests it receives from the browser.

As a consumer of web pages, you only ever deal with the front end, and even then you only passively engage with the page by reading its content, looking at its images, or clicking its links or buttons.

However, as a maker of web pages — that is, as a web developer — your job entails dealing with both the front end and the back end. Moreover, that job includes coding what others see on the front end, coding how the server gathers its data on the back end, and coding the intermediate tasks that tie the two together.

Understanding the Front End: HTML and CSS

As I mention in the previous section, the *front end* of the web development process involves what users see and interact with in the web browser window. It's the job of the web developer to take a page design — which you might come up with yourself, but is more often something cooked up by a creative type who specializes in web design — and make it web-ready. Getting a design ready for the web means translating the design into the code required for the browser to display the page somewhat faithfully. (I added the hedge word "somewhat" there because it's not always easy to take a design that looks great in Photoshop or Illustrator and make it look just as good on the web. However, with the techniques you learn in this book, you'll almost always be able to come pretty close.)

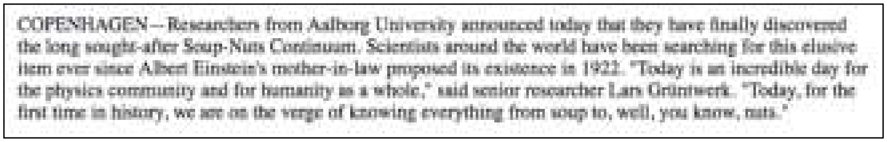
You need code to create the front end of a web page because without it your page will be quite dull. For example, consider the following text:

```
COPENHAGEN—Researchers from Aalborg University announced today
  that they have finally discovered the long sought-after
  Soup-Nuts Continuum. Scientists around the world have been
```

```
   searching for this elusive item ever since Albert Einstein's
   mother-in-law proposed its existence in 1922.

"Today is an incredible day for the physics community and for
   humanity as a whole," said senior researcher Lars Grüntwerk.
   "Today, for the first time in history, we are on the verge of
   knowing everything from soup to, well, you know, nuts."
```

If you plop that text onto the web, you get the result shown in Figure 1-10. As you can see, the text is very plain, and the browser didn't even bother to include the paragraph break.



COPENHAGEN—Researchers from Aalborg University announced today that they have finally discovered the long sought-after Soup-Nuts Continuum. Scientists around the world have been searching for this elusive item ever since Albert Einstein's mother-in-law proposed its existence in 1922. "Today is an incredible day for the physics community and for humanity as a whole," said senior researcher Lars Grüntwerk. "Today, for the first time in history, we are on the verge of knowing everything from soup to, well, you know, nuts."

FIGURE 1-10: Text-only web pages are dishwater-dull.

So, if you can't just throw naked text onto the web, what's a would-be web developer to do? Ah, that's where you start earning your web scout merit badges by adding code that tells the browser how you want the text displayed. That code comes in two flavors: structure and formatting.

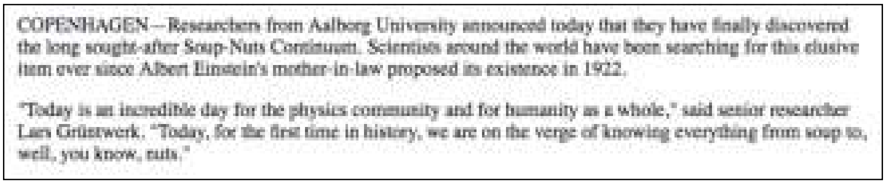
Adding structure: HTML

The first thing you usually do to code a web page is give it some structure. This means breaking up the text into paragraphs, adding special sections such as a header and footer, organizing text into bulleted or numbered lists, dividing the page into columns, and much more. The web coding technology that governs these and other web page structures is called (deep breath) *Hypertext Markup Language,* or *HTML,* for short.

HTML consists of a few dozen special symbols called *tags* that you sprinkle strategically throughout the page. For example, if you want to tell the web browser that a particular chunk of text is a separate paragraph, you place the `<p>` tag (the `p` here is short for paragraph) before the text and the `</p>` tag after the text.

In the code that follows, I've added these paragraph tags to the plain text that I show earlier. As you can see in Figure 1-11, the web browser displays the text as two separate paragraphs, no questions asked.

```
<p>
COPENHAGEN—Researchers from Aalborg University announced today
   that they have finally discovered the long sought-after
   Soup-Nuts Continuum. Scientists around the world have been
   searching for this elusive item ever since Albert Einstein's
   mother-in-law proposed its existence in 1922.
</p>
<p>
"Today is an incredible day for the physics community and for
   humanity as a whole," said senior researcher Lars Grüntwerk.
   "Today, for the first time in history, we are on the verge of
   knowing everything from soup to, well, you know, nuts."
</p>
```



COPENHAGEN—Researchers from Aalborg University announced today that they have finally discovered the long sought-after Soup-Nuts Continuum. Scientists around the world have been searching for this elusive item ever since Albert Einstein's mother-in-law proposed its existence in 1922.

"Today is an incredible day for the physics community and for humanity as a whole," said senior researcher Lars Grüntwerk. "Today, for the first time in history, we are on the verge of knowing everything from soup to, well, you know, nuts."

FIGURE 1-11: Adding paragraph tags to the text separates the text into two paragraphs.

REMEMBER

HTML is one of the fundamental topics of web development, and you learn all about it in Book 2, Chapter 1.

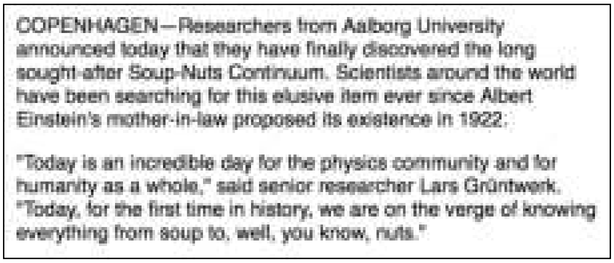
Adding style: CSS

HTML takes care of the structure of the page, but if you want to change the formatting of the page, then you need to turn to a second front-end technology: *cascading style sheets,* known almost universally as just *CSS*. With CSS in hand, you can play around with the page colors and fonts, you can add margins and borders around things, and you can mess with the position and dimensions of page elements.

CSS consists of a large number of *properties* that enable you to customize many aspects of the page to make it look the way you want. For example, the `width` property lets you specify how wide a page element should be; the `font-family` property enables you to specify a typeface for an element; and the `font-size` property lets you dictate the type size of an element. Here's some CSS code that applies all three of these properties to every `p` element (that is, every `<p>` tag) that appears in a page (note that `px` is short for pixels):

```
p {
    width: 700px;
    font-family: sans-serif;
    font-size: 24px;
}
```

When used with the sample text from the previous two sections, you get the much nicer-looking text shown in Figure 1-12.



COPENHAGEN—Researchers from Aalborg University announced today that they have finally discovered the long sought-after Soup-Nuts Continuum. Scientists around the world have been searching for this elusive item ever since Albert Einstein's mother-in-law proposed its existence in 1922.

"Today is an incredible day for the physics community and for humanity as a whole," said senior researcher Lars Grüntwerk. "Today, for the first time in history, we are on the verge of knowing everything from soup to, well, you know, nuts."

FIGURE 1-12: With the judicious use of a few CSS properties, you can greatly improve the look of a page.

REMEMBER

CSS is a cornerstone of web development. You learn much more about it in Book 2, Chapters 2, 3, and 4.

Understanding the Back End: PHP and MySQL

Many web pages are all about the front end. That is, they consist of nothing but text that has been structured by HTML tags and styled by CSS properties, plus a few extra files such as images and fonts. Sure, all these files are transferred from the web server to the browser, but that's the extent of the back end's involvement.

These simple pages are ideal when you have content that doesn't change very often, if ever. With these so-called *static* pages, you plop in your text, add some HTML and CSS, perhaps point to an image or two, and you're done.

But there's another class of page that has content that changes frequently. It could be posts added once or twice a day, or sports or weather updates added once or twice an hour. With these so-called *dynamic* pages, you might have some text, HTML, CSS, and other content that's static, but you almost certainly don't want to be updating the changing content by hand.

Rather than making constant manual changes to such pages, you can convince the back end to do it for you. You do that by taking advantage of two popular back-end technologies: MySQL and PHP.