

Foundation PHP for Dreamweaver 8

David Powers



Foundation PHP for Dreamweaver 8

Copyright © 2006 by David Powers

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN (pbk): 1-59059-569-6

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013.
Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710.
Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit www.apress.com.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is freely available to readers at www.friendsofed.com in the Downloads section.

Credits

Lead Editor

Chris Mills

Assistant Production Director

Kari Brooks-Copony

Technical Reviewer

Jason Nadon

Production Editor

Kelly Winquist

Editorial Board

Steve Anglin, Dan Appleman,
Ewan Buckingham, Gary Cornell,
Tony Davis, Jason Gilmore,
Jonathan Hassell, Chris Mills,
Dominic Shakeshaft, Jim Sumser

Compositor

Katy Freer

Proofreader

Nancy Sixsmith

Project Manager

Richard Dal Porto

Indexer

Diane Brenner

Copy Edit Manager

Nicole LeClerc

Cover Artist

Corné van Dooren

Copy Editor

Andy Carroll

Interior and Cover Designer

Kurt Krames

Manufacturing Director

Tom Debolski

CONTENTS

About the Author	x
About the Technical Reviewer	xi
About the Cover Image	xii
Acknowledgments	xiii
Introduction	xiv
Who this book is for	xiv
Do I need to have Dreamweaver 8?	xv
Windows- and Mac-friendly	xv
What if I'm new to Dreamweaver?	xvi
What this book covers	xvi
Support for this book	xvii
Layout conventions	xvii
Chapter 1: So, You Want to Build Dynamic Sites?	1
What makes a site dynamic?	2
Increasing response to the user with server-side technology	3
Why choose PHP/MySQL?	4
How dynamic sites work	5
Using code that acts as a template	5
Taking a peek behind the scenes	7
Building dynamic sites with Dreamweaver	8
An overview of Dreamweaver server behaviors	9
Looking ahead	11

Chapter 2: Dreamweaver and PHP—A Productive Partnership . . . 13

- Taking a quick first look at Dreamweaver 8’s new features 14
 - Improved CSS support 14
 - A more productive coding environment 15
 - Support for PHP 5 16
- Organizing your workspace 17
 - Preset workspace layouts in Windows 17
 - Tabbed documents in Mac OS X 18
 - Saving and sharing customized workspace layouts 20
- Getting the best out of Code view 21
 - Using the Coding toolbar 21
 - Collapsing code sections 22
 - Formatting code with context menus 24
 - Using PHP code hints and auto completion 25
- Comparing different versions of files 26
 - Setting up the File Compare feature 27
 - Using File Compare 27
 - Comparing two local files in the same site 30
 - Comparing two local files in different sites 30
 - Comparing local and remote files 30
 - Problems with File Compare 31
- Creating standards-compliant code 32
 - Choosing the right DTD 32
 - Be aware of potential pitfalls in XHTML 33
 - Changing the DTD in Dreamweaver 8 35
 - Treat the DOCTYPE declaration with respect 38
 - Meet Mark of the Web 40
 - Removing unwanted code cleanly 40
 - Watch that cursor! 41
- The next step 41

Chapter 3: Getting the Work Environment Ready 43

- Deciding where to test your pages 44
 - Checking the suitability of your remote server for testing 45
- Creating a local testing server 45
 - Why not use an all-in-one package? 46
- Deciding where to locate your sites 47
 - Two strategies: centralized or distributed 47
- Setting up on Windows 51
 - Getting Windows to display filename extensions 51
 - Choosing the right web server 52
 - Installing Apache on Windows 52
 - Starting and stopping Apache on Windows 56
 - Changing startup preferences or disabling Apache 58
 - Downloading and installing PHP on Windows 58
 - Adding PHP to your Windows startup procedure 65
 - Configuring Apache to work with PHP 66
 - Avoiding the “Cannot load mysqli extension” error 72

Configuring IIS to work with PHP	73
Upgrading and uninstalling Apache and PHP on Windows	77
Setting up on Mac OS X	78
Starting Apache and testing PHP	78
Upgrading PHP on Mac OS X	80
Configuring PHP to display errors on Mac OS X	82
Checking your PHP configuration (Windows and Mac)	84
Setting up virtual hosts	85
Registering virtual hosts on Windows	86
Registering virtual hosts on Mac OS X	87
Registering virtual directories on IIS	89
Defining a PHP site in Dreamweaver	89
Points to watch when using a remote server for testing	95
Now you're set to go	96

Chapter 4: Getting Ready for the Case Study 99

Creating the basic structure of East-West Seasons	101
Organizing the download files for the case study	101
Installing Lorem and More	103
Analyzing the main page layout	103
Styling the page with CSS	111
Deciding whether to support legacy browsers	111
Attaching the basic styles with <link>	112
Attaching the remaining styles with @import	114
PHP, here we come	119

Chapter 5: Integrating PHP into Your Site 121

Including text and code from other files	122
Using require() instead of include()	127
Points to remember when using includes	129
“Failed to open stream” and other error messages	130
Displaying PHP output with Live Data view	131
Introducing the basics of PHP	133
Using the correct filename extension	134
The core components of PHP syntax	134
Handling unknown values with variables	135
Assigning values to a variable	136
Getting to know the various datatypes	137
Indenting code and use of white space	138
Separating commands with the semicolon	138
Commenting scripts for clarity and debugging	139
Handling text and other output in PHP	142
Choosing single or double quotation marks	143
Using escape sequences in strings	145
Joining strings	146
Adding to an existing string	147

- Doing calculations with PHP 148
 - Arithmetic operators 148
- Making decisions with PHP 151
 - Using if . . . else conditional statements 153
 - Using comparison operators 154
 - Testing more than one condition 156
 - Solid foundations 158

Chapter 6: Getting Feedback from an Online Form 161

- Designing the feedback form 162
- Activating the feedback form 167
 - Understanding how the contact form works 167
 - Organizing related values in arrays 168
 - Identifying elements in the POST array 170
 - Keeping secure with the superglobal arrays 170
- Improving the feedback form 176
 - Getting rid of unwanted backslashes 176
 - Checking required fields for valid input 178
 - Avoiding email injection attacks 178
 - Improving the look of the email 184
 - How long does a variable last? 185
 - Acknowledging the feedback 186
 - Using Balance Braces 191
 - Turning the email regular expression into a snippet 192
- Changing the site's styles automatically 193
 - Using the switch statement for decision chains 193
 - Building an automatic style changer 194
 - Extracting date parts with date() 194
- Automating the navigation menu 198
 - Using loops to handle repetitive tasks 199
 - Loops using while and do . . . while 199
 - The versatile for loop 200
 - Looping through arrays with foreach 200
 - Creating arrays 201
 - Modularizing code with functions 205
 - Naming functions 206
 - Passing values to functions 206
 - Returning values from functions 208
- Halfway review 208

Chapter 7: Putting the Power of a Database Behind Your Pages 211

- Choosing the right version 213
- Explaining the terminology 213
- Installing MySQL on Windows 214
 - Removing an older version of MySQL 215
 - Installing the Windows Essentials version of MySQL 215

Configuring MySQL Windows Essentials	218
Upgrading to a later series of Windows Essentials	224
Performing a straight upgrade from one series to another	224
Running two series in parallel	225
Changing the default table type on Windows Essentials	226
Starting and stopping MySQL manually on Windows	227
Introducing the MySQL monitor on Windows	229
Starting a session	229
Ending your session	229
Setting up MySQL on Mac OS X	230
Downloading and installing the software	230
Adding MySQL to your PATH	232
Amending PATH in the bash shell	232
Amending PATH in the tcsh shell	233
Securing MySQL on Mac OS X	233
Setting the MySQL root password	234
Working with the MySQL monitor on Windows and Mac	236
Using MySQL with a graphical interface	237
phpMyAdmin: A golden oldie	237
Obtaining and installing phpMyAdmin	237
MySQL Administrator and MySQL Query Browser: Snapping at the heels of phpMyAdmin	240
Now to business	241
Chapter 8: Building a Random Quotation Generator	243
Finding your way around phpMyAdmin	244
Creating user accounts for MySQL	245
Granting the necessary user privileges	246
Building the first database table	250
Choosing the right column type	254
Storing text	255
Storing numbers	256
Storing dates and times	256
Storing predefined lists	257
Storing binary data	257
Populating the database table	257
Displaying database content	259
Creating and updating database records	270
A great deal achieved	282
Chapter 9: Working with Multiple Tables	285
Storing related information in separate tables	286
Using foreign keys to link records	286
What foreign key constraints do	288
Splitting the quotations table	289
Deciding on the best structure	296
Selecting records from more than one table	296

The four essential SQL commands	298
SELECT	299
INSERT	301
UPDATE	302
DELETE	302
Managing content with multiple tables	303
Inserting new quotations	304
Inserting new authors	311
Updating authors	315
Deleting authors	320
Updating quotations	324
The mystery of missing records	325
Deleting quotations	328
Chapter review	329

Chapter 10: Using Sessions to Track Visitors and Restrict Access 331

What sessions are and how they work	332
Creating PHP sessions	333
Creating and destroying session variables	333
Destroying a session	334
The “Headers already sent” error	334
Registering and authenticating users	337
Creating a user registration system	337
Building custom server behaviors	346
Completing the user registration form	349
Updating and deleting user records	353
Adapting the Sticky Text Field server behavior	354
Building the update and delete pages	356
Creating a login system	362
Restricting access to individual pages	364
Logging out users	365
Automating the validation process with MX Kollection 3	365
Installing MX Kollection 3	366
Defining your MX Kollection site preferences	367
Building the user registration and login system	370
Assessing the pros and cons of MX Kollection	374
Two approaches to PHP	375

Chapter 11: Displaying a Blog and Photo Gallery 377

Creating the blog back-end	378
Planning and building the required components	378
Building a list of images in a folder	381
Displaying a message when no records are found	384
Formatting dates and time in MySQL	388
Creating striped table rows	391
Finishing the back-end	392

Displaying the blog	396
Extracting the first section of a long item	396
Using Live Data view with a URL parameter	401
Creating an intelligent link	403
Displaying images in a dynamic site	405
Weighing the pros and cons of storing images in a database	405
Positioning dynamically inserted images	406
Getting an image's dimensions dynamically	411
Displaying a photo gallery	413
How the photo gallery is structured	415
Putting the photo gallery together	417
Improving the blog and photo gallery	424
Storing dates in MySQL	425
Validating and formatting dates for database input	425
Mission almost accomplished	429
Chapter 12: Using XSLT to Display Live News Feeds and XML . . .	431
A quick guide to XML and XSLT	432
What an XML document looks like	432
Using client-side and server-side transformations	434
Checking your server's capability	436
Pulling in an RSS news feed	437
How Dreamweaver handles server-side XSLT	438
Choosing a suitable news feed	439
Being a bit more adventurous with XSLT	446
Setting up a local XML source	446
Understanding how XSLT is structured	448
Accessing nested repeating elements	450
Creating conditional regions	451
Sorting elements	454
Formatting elements	454
Displaying output selectively	455
Going further	461
Appendix A: Using Languages Other Than English in MySQL . . .	463
How MySQL uses character sets and collation	463
Setting the right sort order	465
Appendix B: Essential MySQL Maintenance	469
Backing up and transferring a database	469
Updating the privilege tables after an upgrade	474
What to do if you forget your root password	475
Upgrading and uninstalling MySQL on Mac OS X	475
Index.	481

ABOUT THE AUTHOR

David Powers is a professional writer who has been involved in electronic media for more than 30 years, first with BBC radio and television, and more recently with the Internet. He's now written or coauthored five books on PHP, including the highly successful *Foundation PHP 5 for Flash* (friends of ED, ISBN 1-59059-466-5) and *PHP Web Development with Dreamweaver MX 2004* (Apress, ISBN 1-59059-350-2), as well as acting as technical reviewer on some of friends of ED's best-selling titles.

What started as a mild interest in computing was transformed almost overnight into a passion, when David was posted to Japan in 1987 as a BBC correspondent in Tokyo. With no corporate IT department just down the corridor, he was forced to learn how to fix everything himself. When not tinkering with the innards of his computer, he was reporting for BBC TV and radio on the rise and collapse of the Japanese bubble economy.

It was back in 1995 that David began experimenting with web design, creating and maintaining an 80-page bilingual website for BBC Japanese TV, of which he was Editor for five years. He began the hard way, coding by hand, and then trying all variety of HTML editors, good and bad, before settling on Dreamweaver, which he has used since version 3. The constant demands of updating a large static website prompted the search for a server-side solution, and after experimenting with classic ASP and ColdFusion, David settled on PHP. Since leaving the BBC to work independently, he has built up an online bilingual database of economic and political analysis for Japanese clients of an international consultancy.

When not pounding the keyboard writing books or dreaming of new ways of using PHP and other programming languages, David enjoys nothing better than visiting his favorite sushi restaurant. He has also translated several plays from Japanese.



ABOUT THE TECHNICAL REVIEWER

Jason Nadon started using Dreamweaver around the release of version 2.0. He manages the Ann Arbor Area Macromedia User Group and is an active member of the Macromedia Community. Jason has been in the Information Technology field for the past nine years, and has been building web applications and solutions with Macromedia tools for the past six. He holds several industry certifications and is currently employed by Thomson Creative Solutions as a Web Services Administrator.



ABOUT THE COVER IMAGE

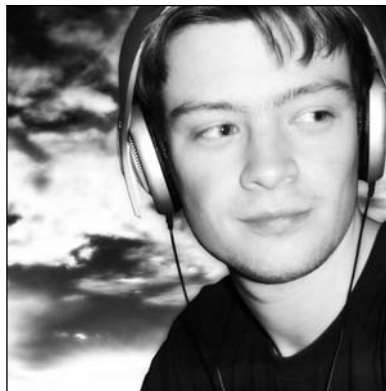
Corné van Dooren designed the front cover image for this book. Having been given a brief by friends of ED to create a new design for the Foundation series, he was inspired to create this new setup combining technology and organic forms.

With a colorful background as an avid cartoonist, Corné discovered the infinite world of multimedia at the age of 17—a journey of discovery that hasn't stopped since. His mantra has always been “The only limit to multimedia is the imagination,” a mantra that is keeping him moving forward constantly.

After enjoying success after success over the past years—working for many international clients, as well as being featured in multimedia magazines, testing software, and working on many other friends of ED books—Corné decided it was time to take another step in his career by launching his own company, *Project 79*, in March 2005.

You can see more of his work and contact him through www.cornevandooren.com or www.project79.com.

If you like his work, be sure to check out his chapter in *New Masters of Photoshop: Volume 2*, also by friends of ED (ISBN: 1-59059-315-4).



ACKNOWLEDGMENTS

Although I'm the only one to get my name on the front cover, like all books, this one would never have seen the light of day without the help and cooperation of many others. First and foremost, my editor, Chris Mills, who endured my interminable phone calls and pointed me in the right direction when I couldn't see the wood for the trees. Chris, a recent Mac convert, also reminded me whenever I'd overlooked the needs of the Mac community. Then there's Richard Dal Porto, who kept things on track while wondering if I'd ever get the final chapter written. (I often wondered myself.) I should also mention Andy Carroll, the copy editor. No author likes to have his carefully crafted prose scrutinized by another, but Andy's vigilance helped remove many ambiguities.

I also received invaluable help from several members of the Dreamweaver development team, who provided many insights into the way Dreamweaver works, and even incorporated some of my suggestions into the final version of Dreamweaver 8. Special thanks must go to one member of the team, Josh Margulis, who helped me unravel the mysteries of the new XSL Transformation server behavior and XPath Expression Builder.

Thanks, too, to Alexandru Costin of InterAKT, who gave me a sneak preview of the updated MX Kollection tools that take working with Dreamweaver 8 to a new level.

Finally, a big thank you to Jason Nadon for his expert technical review. He even tore himself away from the beach on the weekend to work on the book. Now, there's dedication for you . . . or perhaps it was raining. (I'd better not ask.)

INTRODUCTION

Confession time. When I first started developing dynamic websites with Dreamweaver UltraDev 4, I made a complete mess of things. I believed rather naively that Dreamweaver would “do it all” for me, and that all I needed to do was point and click. The first couple of projects actually went well, giving me a false sense of security. As soon as I attempted anything that didn’t fit into the same pattern as the basic tutorials, things began to go horribly wrong. Even though I’ve always been happy working with code, one look inside Code view sent shivers up my spine.

Part of the problem was that I didn’t have a clear grasp of database structure or of what server-side technology really involved. The other part of the problem was that I was treating Dreamweaver as a WYSIWYG (What You See Is What You Get) program. If I saw something wasn’t in the right place, I just highlighted it in Design view and hit *DELETE*. What I didn’t realize was that this left behind messy code that caused even the simplest of applications to break. The more I deleted, the worse it got.

I don’t want that to happen to you. That’s why I wrote this book.

Who this book is for

To be able to develop dynamic websites with a server-side language like PHP, you need to have a good grasp of the principles of web design. You should be familiar with the basics of HTML (Hypertext Markup Language) or XHTML (Extensible Hypertext Markup Language), and preferably have some knowledge of CSS (Cascading Style Sheets). You don’t need to know how to hand-code a web page, but this book involves switching frequently into Dreamweaver’s Code view and making adjustments to the code. Unless you are familiar with the basic structure of a web page, you are likely to find it difficult to follow all the steps.

I assume no prior knowledge of PHP or working with a database. There are detailed instructions on how to set everything up, and I will teach you the basic principles of both PHP and SQL, the language used to interact with databases. The purpose is not to turn you into a “programmer,” but to give you sufficient understanding of what is happening—and why—to help you avoid the mistakes I first made.

I hope this book will also be of value to readers who already have some knowledge of PHP, the MySQL database, or both. Even the chapters that deal with PHP basics are designed to help you work more efficiently in the Dreamweaver 8 environment. And by the end of the book, you will be working on relatively advanced topics. I have included a number of custom functions that the more code-minded reader will probably enjoy deconstructing and adapting. If coding isn't your idea of fun, you can just use the functions and not worry about how they work.

Do I need to have Dreamweaver 8?

This is a book written specifically for Dreamweaver 8, and you're unlikely to get the best value out of it if you're using an earlier version of Dreamweaver. At a pinch, you could probably use Dreamweaver MX 2004, because most of the server behaviors are identical, although bugs in the MX 2004 versions of the User Authentication server behaviors will prevent some parts of Chapter 10 working as described, and you will not be able to do any of Chapter 12, which covers the new—and fascinating—XSL Transformation server behavior. However, you'll also miss out on the important enhancements that have been made to CSS rendering and Code view.

If you don't have Dreamweaver 8—or any version of Dreamweaver, for that matter—all is not lost. You can download a 30-day trial version from www.macromedia.com/cfusion/tdrc/index.cfm?product=dreamweaver.

All the other software required is either available as part of the Windows or Mac OS X operating systems, or free for download from the Internet. Download locations are given when needed, throughout the course of the book.

Windows- and Mac-friendly

Everything in this book has been tested on both Windows and Mac OS X 10.3 and 10.4. Most of the screenshots have been taken on the Windows version, but I have included separate screenshots or descriptions when referring to features that are either exclusive to Mac OS X or that are substantially different from the Windows version. Fortunately, there are very few differences between the Windows and Mac versions of Dreamweaver 8, so Mac users should have no difficulties in following the instructions.

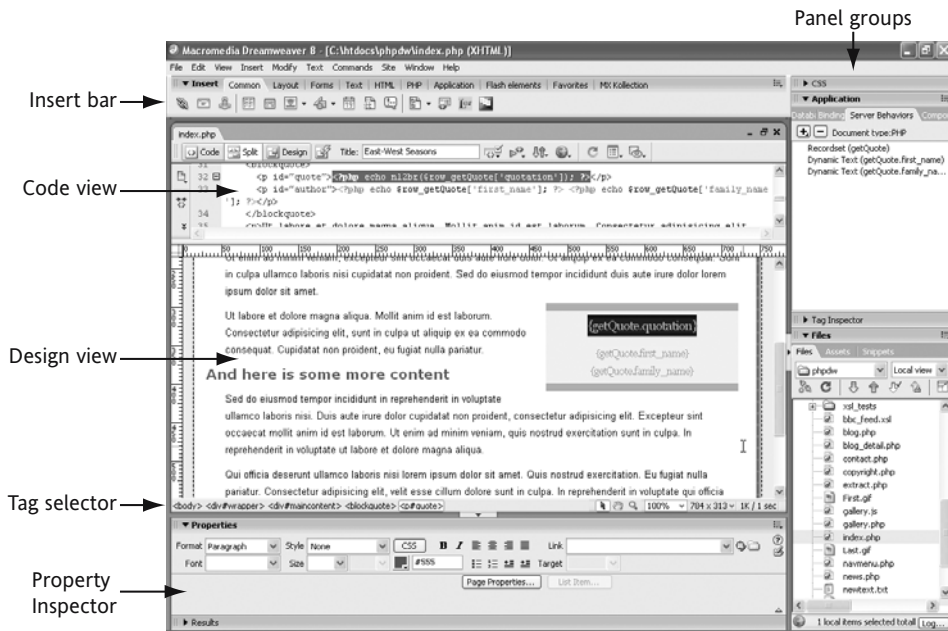
All keyboard shortcut references are given in the order *WINDOWS/MAC*. If you're a newcomer to Mac OS X and don't have a U.S. keyboard, you might be confused by keyboard shortcut references to the *OPT* key. On the U.K. Mac keyboard, at least, this is labeled *ALT*. Whichever Mac keyboard you're using, you can recognize it by the symbol shown alongside.



What if I'm new to Dreamweaver?

This is a Foundation book, so detailed instructions are given for each step. However, I recommend that you familiarize yourself first with the Dreamweaver workspace. Open Dreamweaver Help by pressing *F1*, and take a look at Getting Started with Dreamweaver and Dreamweaver Basics.

The following screenshot shows the Dreamweaver workspace in Windows, using the Designer layout. (The position of each element is identical in the Mac version.) The screenshot shows a PHP document open in Split view, which enables you to view part of the underlying code while seeing a reasonably accurate representation of the final design. As long as you familiarize yourself with the parts of the workspace labeled here, and have a good understanding of web page layout, you should have no difficulty following the instructions in this book. Of the panel groups shown on the right, the two most important ones for building dynamic sites—Applications and Files—are shown expanded.



What this book covers

The book's title really says it all: It's about using PHP with Dreamweaver 8. If you're new to expressions like PHP and server-side programming, Chapter 1 gives you both an overview of what's involved and a brief explanation of what the server behaviors in Dreamweaver 8 are for. Chapter 2 looks at the new features of Dreamweaver 8, with particular emphasis on PHP. Chapters 3 and 7 provide detailed installation instructions for PHP, MySQL, and the phpMyAdmin graphical interface to MySQL.

Chapter 4 sets up the East-West Seasons case study that runs throughout the book, covering the basics of PHP (Chapters 5 and 6) and all the PHP-related server behaviors in Dreamweaver 8 (Chapters 8 through 12). The book also covers building your own server behaviors (in Chapters 10 and 11).

The book focuses exclusively on PHP, so no time is wasted on how something is done differently in ASP, JSP, or ColdFusion. If it's ASP that you're interested in, take a look at *Foundation ASP for Dreamweaver 8* by Omar Elbaga and Rob Turnbull (friends of ED, ISBN 1-59059-568-8), which is equally focused. Take a look, but buy my book first.

Support for this book

All the necessary files for this book can be downloaded from www.friendsofed.com/downloads.php If you run into a problem with any of the instructions, compare your code with the download files using the new File Compare feature in Dreamweaver 8 (as described in Chapter 2). If that doesn't work, check the errata page for this book at www.friendsofed.com/books/1590595696. I try not to make any mistakes, but they occasionally slip through. The errata page is also where you'll find information that came to light after this book went to print.

If you still don't find the answer, post a question in the friends of ED forum at www.friendsofed.com/forums, and I'll try to help you as quickly as possible. Please be as precise as you can about the nature of the problem. It's much easier to answer a question along the lines of "I did this in step 7 on page so and so of *Foundation PHP for Dreamweaver 8*, and this happened" than "Chapter 2 (of unnamed book) doesn't work."

Layout conventions

Important words or concepts are normally highlighted on the first appearance in **bold type**.

Code is presented in fixed-width font.

New or changed code is normally presented in **bold fixed-width font**.

Pseudocode and variable input are written in *italic fixed-width font*.

Menu commands are written in the form Menu ► Submenu ► Submenu.

Where I want to draw your attention to something, I've highlighted it like this:

Ahem, don't say I didn't warn you.

Sometimes code won't fit on a single line in a book. Where this happens, I use an arrow like this ➤

This is a very, very long section of code that should be written all on
➤ the same line without a break.





Chapter 1

SO, YOU WANT TO BUILD DYNAMIC SITES?

What this chapter covers:

- Understanding what makes a site dynamic
- Introducing PHP and MySQL—the dynamic duo
- Understanding the interaction between a database and a website
- Learning how Dreamweaver automates communication with a database

Dreamweaver, like the Internet itself, has come a long way in the past few years. It's not only the premier design tool for creating **static** websites (ones in which all the information is hard-coded into each page); since 2002 when it incorporated support for PHP, ASP, ASP.NET, ColdFusion, and JSP, it's also become the tool of choice for many developers of **dynamic** websites. As the title of this book indicates, I'm going to concentrate on using PHP with Dreamweaver 8. So, if you've already made up your mind that PHP is for you, welcome aboard. If you're unsure, I hope to have whetted your appetite sufficiently by the end of this chapter to entice you to stay along for the rest of the ride.

Over the next few pages, I'll explain how dynamic websites work, what's so special about PHP and the MySQL database that integrates so closely with it, and how Dreamweaver 8 can make development of a database-driven website both easy and enjoyable. In subsequent chapters, I'll show you how to use some of the brilliant new features in Dreamweaver 8 and how to set up your computer with the right development environment for dynamic sites, and I'll teach you the basics of working with PHP. Then, it's on to the core theme of the book—creating dynamic web pages that interact with a database. By the time you finish, you should have a solid understanding not only of the tools provided by Dreamweaver, but also of the basic technology involved. The aim is not to turn you into a hardened code warrior, but to give you the knowledge and confidence to dive under the hood when necessary to troubleshoot problems or customize things to your own liking.

First of all, what does the term “dynamic” really mean when applied to a website?

What makes a site dynamic?

Several years ago, **DHTML** (Dynamic HTML) was regarded as *the* next big thing on the Web. Yet you may be surprised to discover that DHTML has never been an official standard. As far as the World Wide Web Consortium (W3C) is concerned, DHTML simply describes “the combination of HTML, style sheets and scripts that allows documents to be animated” (www.w3.org/DOM/faq.html#DHTML-DOM).

In other words, DHTML is what Dreamweaver has always been so good at: automating rollovers and other onscreen effects through JavaScript-based behaviors. Combined with CSS (Cascading Style Sheets), this has become standard on the Web, which may explain why DHTML is no longer the buzzword it once was. Most web developers now just refer to JavaScript and CSS directly by name.

Confusingly, even when a website uses DHTML effects, it's often referred to as “static.” That's because the underlying code is always the same each time it's downloaded from the web server; the dynamic effects are controlled entirely on the user's computer, normally in response to mouse movements or clicking form elements. In formal terms, this is **client-side** technology (see Figure 1-1).

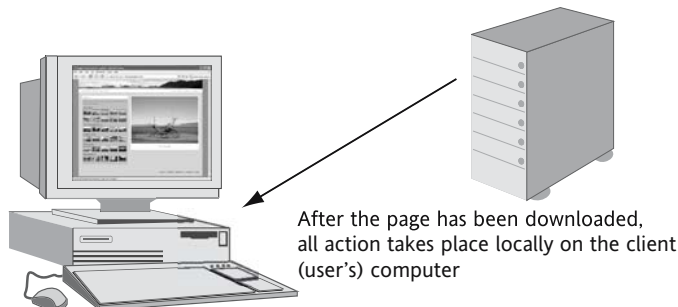


Figure 1-1. Dynamic effects created with JavaScript and CSS involve no direct interaction with the server.

Increasing response to the user with server-side technology

These days, “dynamic”—when applied to a website—usually refers to the ability to change the content or look of a web page *before* it’s downloaded to the user’s computer. In other words, the changes are implemented on the web server. This is known as **server-side** technology, and the languages used to build such sites—like PHP, ASP.NET, ColdFusion, and so on—are called **server-side languages**. Perhaps the best-known example is Amazon.com. Although most pages look very similar, the content has an amazing variety, and it changes in response to user action. Do a search for your favorite author (well, my favorite author), and you’ll be presented with something like Figure 1-2.



Figure 1-2. The Amazon.com website presents information tailored to the user’s requirements through the use of server-side technology.

If you have ever used the Show/Hide Layer behavior in Dreamweaver, you’ll know that it’s possible to put some of your page content in a hidden layer, and reveal it only when the user hovers over an image or clicks on a particular link. Just imagine, though, if Amazon tried to do that! With literally tens, maybe hundreds of thousands of items in the Amazon catalog, it would be impossible to download all that information to your computer and use client-side technology to display just the parts you’re interested in. It makes much more sense to use server-side technology to select the appropriate content and send you no more than necessary. It’s fast, efficient, and very effective.

Amazon.com stores most of its content in a database, and it uses a server-side language to extract and display the relevant information. The system is based on a scripting language known as Perl, but similar functionality is available by choosing any of the popular server-side languages supported by Dreamweaver: ASP (Active Server Pages), ASP.NET, ColdFusion, JSP (JavaServer Pages), and PHP. The initials PHP originally stood for Personal Home Page, but that was considered inappropriate when it developed into the sophisticated language it has become today, so it’s now one of those curious recursive names—PHP Hypertext Processor.

The word **server** will crop up frequently in this book, and it's important to understand that it has two separate, but closely related, meanings. Primarily, it refers to a computer that stores files, such as web pages, and “serves” them to other computers on request. This is known as a file server. A computer that makes a request to a file server is known as a **client**—hence the expression “client-side,” which refers to anything that happens on a user's computer without further intervention by the file server.

The second meaning of server refers to the various types of software that process the requests from visitors to a website. So, in addition to the physical file server, a web server, such as Apache, is needed; and, if a site uses a database, you also need a database server, such as MySQL. Apache and MySQL are software programs that are commonly installed on the same physical computer. I'll show you how they all fit together in the “How dynamic sites work” section later in this chapter.

Dreamweaver uses the term “remote server” to refer to the file server where you upload your web files for display on the Internet—typically a hosting company. The term “testing server” identifies where you test your PHP files before deploying them live on the Internet. Full instructions for setting up a testing server are in Chapter 3.

Why choose PHP/MySQL?

Arguments of an almost religious nature often break out when discussing which is the “best” server-side technology. All the languages supported by Dreamweaver are good, but it's a good idea to pick one and get to know it well. Once you have become proficient at one server-side language, you'll find the transition to another a lot easier, because they share many elements in common.

Strictly speaking, it *is* possible to mix different server-side technologies in the same website, as long as the server supports them. I ran a hybrid ASP-PHP site for several months because I had an existing site that I wanted to convert from ASP to PHP. Instead of waiting until everything had been converted, I did the changeover in two stages. But this is an unusual scenario. It worked because I kept the two parts of the site completely separate from each other.

So, why choose PHP/MySQL in preference to the others? Well, I wouldn't have slaved over a hot keyboard night and day to write this book if I didn't think PHP and MySQL were pretty special. And I'm not alone: PHP is the Web's most widely available server-side language. As of September 2005, it was in use on more than 22.1 million domains (www.php.net/usage.php). MySQL is also reputed to be the most popular open source database, with more than 6 million active installations (www.mysql.com). Because PHP is a server-side language, it needs to work in connection with a web server. More often than not, you'll find it in combination with Apache, the software that runs more than two out of every three web servers in the world today (http://news.netcraft.com/archives/web_server_survey.html).

Apache, PHP, and MySQL run on just about every operating system, including Windows, Mac OS X, and Linux. This flexibility is one of the great advantages of developing with PHP/MySQL. Let's quickly review the others:

- **Cost:** They're free. Don't be fooled into thinking this means they're just for hobbyists. PHP 5 has full object-oriented capability. It can also work with all leading database systems (Dreamweaver's automatically generated PHP code, however, supports PHP only in combination with MySQL). MySQL is used by many leading organizations, including NASA, the U.S. Census Bureau, Yahoo!, and the New York Stock Exchange. The fact that more than two out of every three web servers run on Apache speaks for itself.

- **Open source:** As open source technologies, all three benefit from a rapid upgrade policy based on need rather than commercial pressures. If a bug or security risk is identified, the input of many volunteers helps the core development teams solve problems rapidly. Future versions are available for beta testing by anybody who wants to take part, and they aren't declared stable until they really are. The same thriving community offers assistance and advice to newcomers and experienced programmers alike.
- **Cross-platform capability:** You can develop on your personal computer and deploy exactly the same code on the production server, even if it's running on a different operating system.
- **Security:** Although it's impossible to predict future developments, Apache servers are less vulnerable to virus attacks. Access to sensitive content can be restricted through PHP session control (this is the subject of Chapter 10).

One thing missing from that list is “ease of learning.” That's not because they're difficult—far from it. All are relatively easy to pick up, but they do require a bit of effort on your part. If you have experience with other programming languages, your progress is likely to be much faster than if you are a complete beginner. This book is designed to ease your progress, whatever your level of expertise.

Although Apache is the web server that I recommend for developing with PHP, you can also use Microsoft IIS on some Windows systems. Instructions for setting up both will be given in Chapter 3.

How dynamic sites work

Although I refer specifically to Apache, PHP, and MySQL, exactly the same principles lie behind all server-side languages and database-driven websites. The web server and the databases may be different, but the component parts all fit together in the same way.

With an ordinary web page, the information is fixed at the time of design. All text, links, and images are hard-coded into the underlying XHTML (Extensible Hypertext Markup Language). Although dynamic effects, such as rollovers, can be achieved with JavaScript or CSS, the basic content remains unchanged until the designer alters it in an editor like Dreamweaver and uploads it again to the server. As was shown in Figure 1-1, there's no direct interaction with the server once the page has been loaded; everything takes place on the client computer.

Using code that acts as a template

Dynamic web pages built with a server-side language like PHP work in a very different way. Instead of all content being embedded in the underlying code, much of it is automatically generated by the server-side language or drawn from a database. A simple way of thinking about it in Dreamweaver terms is that a dynamic page is rather like a template. Those parts of the page that you want to remain constant are created in standard XHTML, while dynamic elements are scripted in a server-side language.

That *doesn't* mean you need to use Dreamweaver templates to work with a server-side language, and I won't be covering their use in this book. You *can* use templates with PHP if you want, but often a single PHP document is all you need to create in order to display hundreds, even thousands, of different pages on your website. (Think how many books must be in the Amazon catalog, yet every page has the same basic structure.) You build the basic structure in XHTML, and you rely on PHP to generate the actual content dynamically, usually—but not always—drawing the content from a database.

To show you a simple example of how this works, consider a copyright notice at the bottom of a web page. In a static website, the code would probably look something like this:

```
<div id="footer">&copy; 1999-2005 David Powers</div>
```

The problem is that you need to change the second date on January 1 every year. Of course, you could use a client-side solution, such as JavaScript, but there are two problems with this approach:

- Date functions in JavaScript rely on the user's computer clock being set correctly.
- If the user has disabled JavaScript, the function will fail, almost certainly making a mess of your design, unless you build in preventive measures.

A server-side solution eliminates both problems: The server makes sure the date is right, and it inserts the necessary code before sending it to the client. If your server clock isn't set correctly, the date will still be wrong, but you have much more control over the clock on your own server than those on the computers of your visitors.

So, how does the server-side approach work? You create a web page as normal, but instead of giving it an `.html` extension (the default for static web pages on both Windows and Mac in Dreamweaver 8), you give the filename a special extension that instructs the web server to process any dynamic code before sending the page to the browser. Inside the page itself, you insert the necessary code where you want the dynamic data to be displayed. In the case of PHP, you would give the filename a `.php` extension and rewrite the previous copyright code like this:

```
<div id="footer">&copy; 1999-<?php echo date('Y'); ?> David  
➤ Powers</div>
```

You won't be able to run this code on your local computer until you have installed PHP and a web server as described in Chapter 3.

Figure 1-3 shows how the web server handles the code. Anything between opening and closing PHP tags (`<?php` and `?>`) is sent to the PHP engine, converted into XHTML output, and sent to the browser with the rest of the content. If you open the source view in a browser, the PHP code is nowhere to be seen. It always remains on the server (hence the name *ww* “server-side technologies”). In this particular example, what lies between the PHP tags is an instruction to get the current year and display it.

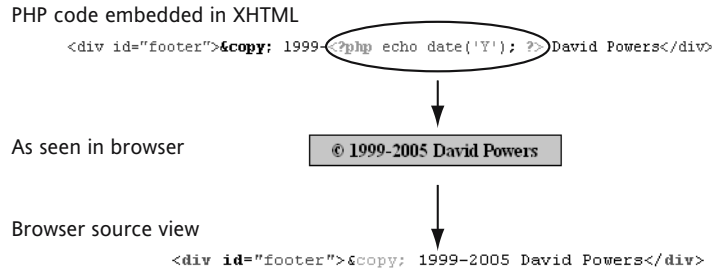


Figure 1-3. The server-side code remains exclusively on the server, and ordinary XHTML output is sent to the browser.

Consequently, as soon as the New Year is rung in—even at the stroke of midnight—the correct year will be displayed without any need to update the underlying code. In fact, as I'll show you in Chapter 5, you can instruct the server to add the same block of code to every page on a site. So, you can make the changes to just one page that contains the common code, rather than needing to update every single one. In this respect, it's similar to the principle of CSS, where a single stylesheet can control an entire website, no matter how many pages it contains.

This example involves no user interaction; it simply displays the year. But server-side technology really comes into its own when responding to user input. In Chapter 6, I'll show you how you can use PHP to email the contents of an online feedback form to your own mailbox; and in the second half of the book, you'll work with a database to create a random quotation generator (Chapters 8 and 9), a blog (Chapter 11), and an image gallery (Chapter 11), as well as looking at how to password-protect selected parts of your site (Chapter 10) and display input from a live news feed (Chapter 12).

Pretty powerful stuff, this server-side technology—and Dreamweaver takes much of the hard work out of it, particularly when it comes to working with a database.

Taking a peek behind the scenes

Without getting too technical, let's take a quick look behind the scenes to see how a dynamic page is served up when a request comes from a user. Figure 1-4 shows the sequence of events.

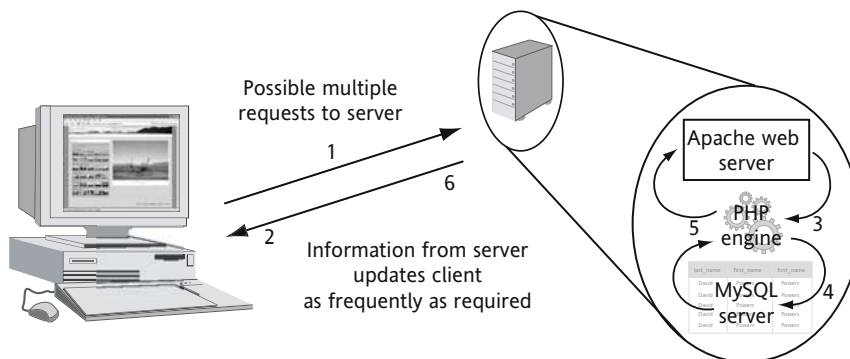


Figure 1-4. A diagrammatic representation of what happens when a client computer makes a request to PHP

1. The client computer sends a request to the web server (normally Apache).
2. If the requested page contains no PHP script, the XHTML, CSS, images, and any other page assets are sent to the client.
3. If the requested page contains any PHP, the web server passes the script to the PHP engine for processing.
4. If the PHP script initiates a database query, a request is sent to the MySQL server.
5. The results of the database query are sent back to the PHP engine for processing and insertion into the page.
6. The web server assembles the entire page as XHTML and sends it to the client, together with any images or other assets.

Once the client receives the data from the server, the page is displayed in the browser, which is ready to receive any input from the user.

There's a lot going on here, and this process can be repeated many times as the user interacts with the page. Sometimes all six stages are required, and, at other times, PHP may simply do all the processing itself without having to query the database. Although it sounds complicated, PHP, MySQL, and Apache can search through many thousands of records in a fraction of a second, so the user may be totally unaware of anything going on in the background. The biggest delay is often caused by a slow Internet connection or network bottlenecks.

Apache, PHP, and MySQL are frequently located on the same computer (although in large operations they may be distributed across several). Apache and MySQL run unobtrusively in the background, consuming very few resources, and are ready to spring into action whenever a request comes in.

*Programs like Apache and MySQL that run in the background are normally referred to as **daemons** on Unix, Linux, and Mac OS X. Windows calls them **services**.*

Building dynamic sites with Dreamweaver

Dreamweaver 8 has a lot of new features to improve your productivity when working with PHP. I'll cover these in detail in the next chapter, but while I'm talking about databases, I'd like to take a look ahead to what lies in store for you in the second half of the book. Dreamweaver makes connecting to a database a breeze. It creates the code that displays the results of a database query in seconds, and it automatically sets up navigation systems to page through a long set of results. (You'll learn how to do both of these in Chapter 8.)

PHP is the language you'll be using to communicate with your MySQL database, and although I'll also be showing you how to write your own PHP, Dreamweaver takes care of much of the complex coding on your behalf by offering an extensive suite of server behaviors. You use server behaviors in a very similar way to the JavaScript-driven behaviors that you might have used in Dreamweaver for validating a form or creating rollover effects.

An overview of Dreamweaver server behaviors

You can access Dreamweaver's server behaviors in three different ways:

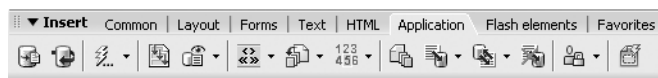
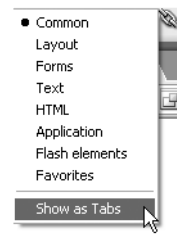
- By clicking the plus (+) button in the Server Behaviors panel, located in the Application panel group
- By selecting the Insert ► Application Objects menu option
- By clicking the appropriate icon in the Application category of the Insert bar

You won't be able to use either of the first two methods until you have defined a PHP site in Dreamweaver (covered in Chapter 3), but the Insert bar is always visible, and it's where you find a lot of the most commonly used features in Dreamweaver. If you're new to Dreamweaver, you can learn all about the Insert bar by selecting Help ► Getting Started with Dreamweaver. When the help window opens, select Dreamweaver Basics ► Getting to know the Dreamweaver 8 workspace ► The Insert bar. Let's take a quick look at the Application category of the Insert bar, so you have an idea of what Dreamweaver can do when working with a dynamic data.

The Insert bar is displayed by default at the top of the workspace. If you can't see it, select Window ► Insert or press *CTRL+F2/⌘+F2*. Normally, the Insert bar displays only the last category you were working in, as shown here:



The other categories are accessed from the drop-down menu at the left end of the Insert bar. I find it a nuisance to open the menu every time I want to select another category, and prefer to have the categories shown as tabs that can be seen at all times. To select this option, choose Show as Tabs at the bottom of the drop-down menu (as shown alongside). With the Application tab selected, the Insert bar will then look like this:



If you prefer to revert to the default display, right-click/*CTRL*-click the title bar of the Insert bar, and select Show as Menu. Throughout the book, I'll use the tabbed interface, so if you're using the default layout, just select the appropriate category from the drop-down menu. There's no difference in functionality, but the tabbed interface takes up an extra 20 pixels of vertical space on your monitor; so if you have a small monitor, the menu interface will give you a precious bit of extra real estate.

As you can see, the Application tab of the Insert bar has a lot of icons, some of which activate drop-down menus. I don't intend to go into minute detail about what each one does—it's easier to understand their functions by actually using them—but the following quick rundown will give you an insight into what you can expect to have learned by the end of this book.

Recordset and Stored Procedure One of the most fundamental operations when working with a database is retrieving records that match your search criteria. This may be a list of products in an online catalog, a staff list, or even the latest entries in a blog. Dreamweaver gathers the results of your search into a **recordset** (in other words, a set of records), ready for you to display in your web page. Clicking the Recordset icon opens the dialog box where you define which records you want a recordset to contain. Dreamweaver then creates all the necessary PHP code, as well as a query written in SQL (Structured Query Language), the language used to interact with all major relational databases. Most of the process is automatic, although you will need some knowledge of SQL to build more complex queries. I'll be covering the main SQL commands as you progress through the second half of the book.



Stored procedures are very similar to recordsets, except they use preset queries stored inside your database. Stored procedures aren't available in MySQL prior to version 5.0, so Dreamweaver doesn't yet support them for PHP/MySQL, although it is hoped that an update will eventually add this feature. Until then, the Stored Procedure icon serves no purpose when developing a PHP site.



Dynamic Data This icon activates a drop-down menu with six options that enable you to populate a variety of web page elements, such as tables, drop-down menus, and text fields with data from your database. Normally, the data will be taken from a recordset. Although this option appears on the Insert bar, it's normally much quicker to access the same lightning bolt icon in the Property inspector or in a dialog box. You will become very familiar with this icon in a variety of contexts by the time you reach the end of this book.



Repeated Region and Show Region When displaying a list of products in a static web page, you normally need to create a table and insert every item individually. That's not necessary with PHP. You just create a single row, insert the appropriate dynamic data, and then apply Dreamweaver's Repeated Region server behavior, which loops through all the results and creates the code automatically. You can also choose whether to display all results, or limit the page to just a specific number. (Recordset Paging, described in the next section, automates the creation of a navigation system to move back and forth through a long set of results.) Repeated Regions don't apply only to tables; you can also use them with <div>s, paragraphs, or any other integral page element.

Sometimes, your recordset may not contain any relevant results. The Show Region server behavior creates the "smart" code that decides whether to display certain parts of the page when that happens. You'll learn how to use the Repeated Region server behavior in Chapter 8 and Show Region in Chapter 11.



Recordset Paging, Go to Detail Page, and Display Record Count These server behaviors enable you to help visitors find their way around your website. The most useful of these is Recordset Paging, which automates the construction of a navigation system that lets users move backwards and forwards through a long set of database results by spreading them over several pages. Recordset Paging is covered in Chapter 8.

Master Detail Page Set, Insert Record, Update Record, and Delete Record The Master Detail Page Set is probably my least favorite server behavior. It's a good idea, but it's complicated to set up, and very inflexible. I'll show you a better solution using other server behaviors in Chapter 11. The three other server behaviors—Insert Record, Update Record, and Delete Record—are the bread and butter of working with a database. The names are self-explanatory. They're very efficient and easy to use. You'll learn how to use them in Chapters 8 and 9.



User Authentication This set of server behaviors makes light work of password-protecting sections of your site, such as a members-only area or the administrative back-end of a database. You'll use this suite of server behaviors in Chapter 10.



XSL Transformation This is a major new feature in Dreamweaver 8 that makes working with XML very easy. You can either convert entire existing XHTML pages to incorporate XML data, or you can create what Dreamweaver 8 calls an XSLT Fragment for incorporation into a PHP page. Suitable XML sources can be files that you create yourself or a remote source, such as an RSS feed. I'll show you how to incorporate a live news feed into your website using an XSLT Fragment and PHP in Chapter 12. (And if you're confused by all this alphabet soup, everything will be explained at the same time.) Although the code created by Dreamweaver 8 is compatible with both PHP 4 and PHP 5, you will need access to PHP 5 to make full use of this feature, as XSLT support is not enabled by default in PHP 4.



Looking ahead

As you can see, Dreamweaver puts at your disposal an impressive arsenal of tools that automate a large part of building a database-driven website with PHP and MySQL. It's important to realize, though, that automatic code generation is only part of the story. No computer program, however sophisticated, can completely automate everything for you.

Although working with a database is what puts real power behind dynamic websites, PHP is used for much more. One of its most practical uses is gathering information from an online form and emailing it. Sadly, this isn't one of the functions that Dreamweaver automates for you.

So, rather than diving straight into working with databases and server behaviors, I intend to show you first how to understand the basics of PHP, and use that knowledge to build an online contact form. As well as being of immediate practical use, this knowledge will ease the transition to working with databases and customizing Dreamweaver's automatically generated code.

Dreamweaver 8 has some excellent new features that make diving under the hood into the underlying code an experience to be enjoyed—and not something to fear. We'll take an in-depth look at them in the next chapter. Then, in Chapter 3, I'll show you how to set up the development environment for working with PHP in Dreamweaver. The rest of the book will be centered on a case study called East-West Seasons that will be used to demonstrate the transition from a static website to a dynamic one—setting that up is the subject of Chapter 4. Chapters 5 and 6 will be devoted to the basics of PHP and getting that contact form online. Then, from Chapter 7 onward, you'll be working with databases. There's a lot to cover, so let's get on with it!

