

PHP Solutions: Dynamic Web Design Made Easy

David Powers

PHP Solutions: Dynamic Web Design Made Easy

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-13 (pbk): 978-1-59059-731-6

ISBN-10 (pbk): 1-59059-731-1

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

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

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

Senior Production Editor
Laura Cheu

Technical Reviewer
Samuel Wright

Compositor
Molly Sharp

Editorial Board
Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Keir Thomas, Matt Wade

Artist
April Milne

Proofreader
Liz Welch

Senior Project Manager
Kylie Johnston

Indexer
John Collin

Copy Edit Manager
Nicole Flores

Interior and Cover Designer
Kurt Krames

Copy Editors
Nicole Flores, Ami Knox

Manufacturing Director
Tom Debolski

Assistant Production Director
Kari Brooks-Copony

Cover Photography
David Powers

CONTENTS AT A GLANCE

About the Author	xiii
About the Technical Reviewer	xiv
About the Cover Image	xv
Intro	xvii
Chapter 1: What Is PHP—And Why Should I Care?	3
Chapter 2: Getting Ready to Work with PHP	15
Chapter 3: How to Write PHP Scripts	45
Chapter 4: Lightening Your Workload with Includes	89
Chapter 5: Bringing Forms to Life	117
Chapter 6: Uploading Files	151
Chapter 7: Using PHP to Manage Files	179
Chapter 8: Generating Thumbnail Images	211
Chapter 9: Pages That Remember: Simple Login and Multipage Forms	233
Chapter 10: Setting Up MySQL and phpMyAdmin	261
Chapter 11: Getting Started with a Database	285
Chapter 12: Creating a Dynamic Online Gallery	319
Chapter 13: Managing Content	341
Chapter 14: Solutions to Common PHP/MySQL Problems	381
Chapter 15: Keeping Intruders at Bay	429
Index	444

CONTENTS

About the Author	xiii
About the Technical Reviewer	xiv
About the Cover Image	xv
Intro	xvii
Chapter 1: What Is PHP—And Why Should I Care?	3
Embracing the power of code	5
Creating pages that think for themselves	5
How hard is PHP to use and learn?	8
Can I just copy and paste the code?	9
How safe is PHP?	10
How to use this book	10
Using the download files	11
A note about versions	12
So, let's get on with it	13
Chapter 2: Getting Ready to Work with PHP	15
What you need to write and test PHP pages	16
Checking whether your website supports PHP	16
Choosing a good script editor for PHP	17
Dreamweaver: Visual display of PHP output	17
GoLive CS2: Some useful features	18
EditPlus 2: Versatile text-only editor for Windows	19
BBEdit and TextMate: Script editors for Mac OS X	19
Checking your scripts with a file comparison utility	19
Deciding where to test your pages	20
What you need for a local test environment	20
Individual programs or an all-in-one package?	21

CONTENTS

Setting up on Windows 21

 Getting Windows to display filename extensions 21

 Choosing a web server for Windows 22

 Installing Apache on Windows 22

 Starting and stopping Apache on Windows 24

 Setting up PHP on Windows 24

 Downloading and configuring PHP 24

 Adding PHP to your Windows startup procedure 27

 Configuring Apache to work with PHP 29

 Configuring IIS to work with PHP 32

 Testing PHP on Windows 34

 Troubleshooting 35

Setting up on Mac OS X 35

 Using Apache on Mac OS X 36

 Starting and stopping Apache 36

 Where to locate your web files 37

 Installing PHP on Mac OS X 38

 Using a Mac package for PHP 38

 Configuring PHP to display errors on Mac OS X 39

 Testing PHP on Mac OS X 40

Checking your PHP settings (Windows and Mac) 41

What's next? 43

Chapter 3: How to Write PHP Scripts 45

PHP: The big picture 46

 Telling the server to process PHP 47

 Embedding PHP in a web page 47

 Using variables to represent changing values 48

 Naming variables 50

 Assigning values to variables 50

 Ending commands with a semicolon 51

 Commenting scripts 51

 Single-line comments 52

 Multiline comments 52

 Using arrays to store multiple values 53

 PHP's built-in superglobal arrays 54

 Understanding when to use quotes 55

 Special cases: true, false, and null 56

 Making decisions 57

 Making comparisons 59

 Using indenting and whitespace for clarity 59

 Using loops for repetitive tasks 60

 Using functions for preset tasks 60

 Displaying PHP output 61

 Joining strings together 62

 Working with numbers 62

 Understanding PHP error messages 63

 Now, on with the show 64

PHP: A quick reference	64
Using PHP in an existing website	64
Data types in PHP	64
Doing calculations with PHP	66
Arithmetic operators	66
Determining the order of calculations	67
Combining calculations and assignment	68
Adding to an existing string	68
All you ever wanted to know about quotes—and more	68
How PHP treats variables inside strings	69
Using escape sequences inside double quotes	70
Avoiding the need to escape quotes with heredoc syntax	70
Unraveling the magic quotes tangle	71
Creating arrays	73
Using array() to build an indexed array	74
Using array() to build an associative array	74
Using array() to create an empty array	74
Multidimensional arrays	75
Using print_r() to inspect an array	75
The truth according to PHP	76
Explicit Boolean values	76
Implicit Boolean values	77
Making decisions by comparing two values	77
Testing more than one condition	78
Using the switch statement for decision chains	79
Using the conditional operator	80
Creating loops	80
Loops using while and do... while	81
The versatile for loop	81
Looping through arrays with foreach	82
Breaking out of a loop	83
Modularizing code with functions	83
Passing values to functions	84
Returning values from functions	85
Where to locate custom-built functions	85
PHP quick checklist	85

Chapter 4: Lightning Your Workload with Includes 89

Including code from other files	91
Introducing the PHP include commands	91
Choosing the right filename extension for includes	94
Using PHP to identify the current page	96
Creating pages with changing content	103
Preventing errors when an include file is missing	112
Choosing where to locate your include files	114
Security considerations with includes	115
Summary	115

Chapter 5: Bringing Forms to Life 117

How PHP gathers information from a form	118
Understanding the difference between post and get	119
Keeping safe with PHP superglobals	122
Sending email	123
Removing unwanted backslashes from form input	124
Processing and acknowledging the message	125
Validating user input	129
Making sure required fields aren't blank	130
Preserving user input when a form is incomplete	133
Filtering out potential attacks	136
Safely including the user's address in email headers	139
Handling multiple-choice form elements	142
Redirecting to another page	148
Summary	149

Chapter 6: Uploading Files 151

How PHP handles file uploads	152
Checking whether your server supports uploads	153
Adding a file upload field to a form	154
Understanding the \$_FILES array	155
Establishing an upload directory	158
Creating an upload folder for local testing	158
Uploading files	159
Moving the temporary file to the upload folder	159
Removing spaces from filenames	162
Rejecting large files	163
Accepting only certain types of files	167
Preventing files from being overwritten	169
Organizing uploads into specific folders	172
Uploading multiple files	174
Points to watch with file uploads	177

Chapter 7: Using PHP to Manage Files 179

Checking that PHP has permission to open a file	180
Configuration settings that affect file access	180
Creating a file storage folder for local testing	181
Reading and writing files	182
Reading files in a single operation	182
Opening and closing files for read/write operations	187
Reading a file with fopen()	189
Replacing content with fopen()	190
Appending content with fopen()	191
Writing a new file with fopen()	191
Combined read/write operations with fopen()	192
Moving the internal pointer	192

Exploring the file system	195
Inspecting a directory the quick way	195
Opening a directory to inspect its contents	196
Building a drop-down menu of files	197
Automatically creating the next file in a series	200
Opening remote data sources	203
Creating a download link	207
Summary	209
 Chapter 8: Generating Thumbnail Images	 211
Checking your server's capabilities	212
Manipulating images dynamically	213
Making a smaller copy of an image	214
Getting ready	214
Building the script	215
Resizing an image automatically on upload	223
Further improvements	228
Transferring your test files to a remote server	230
Summary	230
 Chapter 9: Pages That Remember: Simple Login and Multipage Forms	 233
What sessions are and how they work	234
Creating PHP sessions	236
Creating and destroying session variables	236
Destroying a session	237
The "Headers already sent" error	237
Using sessions to restrict access	238
Using file-based authentication	241
Encrypting passwords	247
Setting a time limit on sessions	253
Passing information through multipage forms	256
Coming up	258
 Chapter 10: Setting Up MySQL and phpMyAdmin	 261
Why MySQL?	262
Which version?	263
Installing MySQL on Windows	263
Changing the default table type on Windows Essentials	268
Starting and stopping MySQL manually on Windows	268
Using the MySQL monitor on Windows	269
Updating the PHP connector files	270
Troubleshooting	271

CONTENTS

Setting up MySQL on Mac OS X	271
Adding MySQL to your PATH	273
Securing MySQL on Mac OS X	275
Using MySQL with a graphical interface	277
Setting up phpMyAdmin on Windows and Mac	277
Launching phpMyAdmin	280
Logging out of phpMyAdmin	281
Backup and data transfer	281
Looking ahead	283

Chapter 11: Getting Started with a Database 285

How a database stores information	286
How primary keys work	287
Linking tables with primary and foreign keys	288
Breaking down information into small chunks	289
Checkpoints for good database design	289
Setting up the phpsolutions database	290
MySQL naming rules	290
Case sensitivity of names	290
Using phpMyAdmin to create a new database	291
Creating database-specific user accounts	291
Creating a database table	294
Inserting records into a table	296
Choosing the right column type in MySQL	299
Storing text	299
Storing numbers	300
Storing dates and times	300
Storing predefined lists	301
Storing binary data	301
Connecting to MySQL with PHP	301
Checking your remote server setup	302
How PHP communicates with MySQL	303
Connecting with the original MySQL extension	303
Connecting with the MySQL Improved extension	304
Connecting with PDO	304
Building a database connection function	305
Finding the number of results from a query	308
Displaying the results of a query	311
MySQL connection crib sheet	314
Summary	316

Chapter 12: Creating a Dynamic Online Gallery 319

Why not store images in a database?	321
Planning the gallery	321
Converting the gallery elements to PHP	323

Building the dynamic elements	326
Passing information through a query string	327
Creating a multicolumn table	330
Paging through a long set of records	332
Selecting a subset of records	332
Navigating through subsets of records	336
Summary	339
 Chapter 13: Managing Content	 341
Keeping your data safe	342
Understanding the danger of SQL injection	342
Basic rules for writing SQL	343
SQL is case-insensitive	343
Whitespace is ignored	343
Strings must be quoted	344
Handling numbers	344
Incorporating variables into SQL queries	344
Direct incorporation	344
MySQL prepared statements	345
PDO prepared statements	346
Setting up a content management system	347
Creating the journal database table	349
Creating the basic insert and update form	350
Inserting new records	351
Linking to the update and delete pages	356
Updating records	360
Deleting records	371
A quick warning about extract()	373
Reviewing the four essential SQL commands	374
SELECT	374
INSERT	377
UPDATE	377
DELETE	378
Security and error messages	378
Summary	379
 Chapter 14: Solutions to Common PHP/MySQL Problems	 381
Displaying a text extract	382
Extracting a fixed number of characters	382
Using PHP	382
Using MySQL	383
Ending an extract on a complete word	383
Extracting the first paragraph	384
Displaying paragraphs	385
Extracting complete sentences	385

CONTENTS

- Let's make a date 388
 - How MySQL handles dates 388
 - Formatting dates in a SELECT query 389
 - Adding to and subtracting from dates 390
 - Working with dates in PHP 392
 - Setting the correct time zone 392
 - Creating a Unix timestamp 393
 - Formatting dates in PHP 394
 - Inserting dates into MySQL 396
- Working with multiple database tables 400
 - Understanding table relationships 400
 - Linking an image to an article 402
 - Selecting records from multiple tables 410
 - Finding records that don't have a matching foreign key 414
 - Creating an intelligent link 416
 - Creating a lookup table 417
 - Setting up the categories and lookup tables 418
 - Inserting new records with a lookup table 418
 - Adding a new category 424
 - Updating records with a lookup table 424
 - Deleting records that have dependent foreign keys 425
- Summary 427
- Chapter 15: Keeping Intruders at Bay 429**
 - Choosing an encryption method 430
 - Using one-way encryption 430
 - Creating a table to store users' details 431
 - Registering new users 431
 - Using two-way encryption 438
 - Creating the table to store users' details 438
 - Registering new users 439
 - User authentication with two-way encryption 440
 - Decrypting a password 441
 - Updating user details 442
 - Where next? 442
- Index. 444**

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. This is the seventh book he has written or co-authored for friends of ED/Apress, including the highly successful *Foundation PHP for Dreamweaver 8* (ISBN: 1-59059-569-6) and *Foundation PHP 5 for Flash* (ISBN: 1-59059-466-5). He is an Adobe Community Expert for Dreamweaver, and provides regular support and advice on PHP and other aspects of web development in several online forums, including friends of ED at www.friendsofed.com/forums.

What started as a mild interest in computing was transformed almost overnight into a passion, when David was posted to Japan in 1987 as BBC correspondent in Tokyo. With no corporate IT department just down the hallway, 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. 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

Samuel Wright is a technical writer and web programmer living near Oxford, England. He is interested in using computers to facilitate routine tasks, and he enjoys learning about new technologies and writing about them. The downside to these interests is spending long hours wrestling with abstruse writing software.

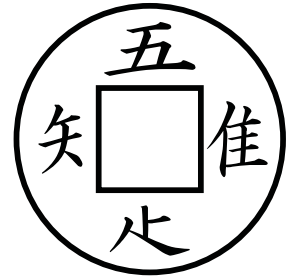
Samuel graduated from the University of Manchester Institute of Science and Technology (UMIST) with a degree in physics, and he has held various positions since. He is currently employed full time at Celoxica as a technical writer.

Samuel runs a music webzine, Lykoszine (www.lykoszine.co.uk), and spends much of his time listening to as much heavy music as he can get his hands on. His remaining time is spent reading, juggling, and hiking.

ABOUT THE COVER IMAGE

The photo on the front cover is a picture I took of the stone water basin behind the monks' quarters at Ryoanji temple in Kyoto, Japan. Ryoanji is perhaps best known for its rock garden—15 stones in a sea of white gravel. It's designated by UNESCO as a World Heritage Site, but was once infamously described by the British travel writer A. A. Gill as “an impractical joke, medieval builder's rubbish.” Although I've visited Ryoanji on several occasions, when I went there in early winter 2005, the garden wall was being restored, so for once it did really look like a builder's yard. Instead of contemplating the rocks and gravel, I spent my time admiring this simple, but beautiful water basin.

But why put it on the cover of a book about PHP? Well, apart from the fact that it's a nice photograph, the crystal clear water trickling into the basin through the bamboo pipe symbolizes for me a constant flow of fresh ideas, a fount of knowledge, just like the Internet. Viewed from above, the water basin also has a fascinating inscription (illustrated alongside).



Read clockwise from the left side, the characters mean *arrow*, *five*, *short-tailed bird*. The final character, at the bottom, has no meaning on its own—and that's the clue. In combination with the square opening of the basin, it forms the character for *sufficient*. In fact, the mouth of the basin is an integral part of the inscription. Each character combines with it to form a completely different one.

Once you unlock the secret, it forms the following sentence: *ware tada taru wo shiru*. Roughly translated, this means “I know only satisfaction” or “I am content with what I have.”

吾唯足知

This is an important concept in Zen philosophy—knowledge for its own sake is sufficient. A person who learns to become content is rich in spirit, even if not in material terms. The more you think about it, the deeper its meaning becomes. Just like the rock garden—if all you can see is a pile of rubble, you have missed the point.

ABOUT THE COVER IMAGE

However, the subtitle of this book is not *Zen and the Art of Website Maintenance* (apologies to Robert M. Pirsig). I want this book to teach you practical skills. At the same time, the inscription on this water basin embodies an important message that applies very much to creating dynamic websites with PHP. The solution to a problem may not always be immediately obvious, but creative thinking will often lead you to the answer. There is no single “right” way to build a dynamic website. The more you experiment, the more inventive your solutions are likely to become.

INTRODUCTION

Dynamic Web Design Made Easy—that’s a pretty bold claim. How easy is easy?

It’s not like an instant cake mix: just add water and stir. Dynamic web design is—well—dynamic. Every website is different, so it’s impossible to grab a script, paste it into a web page, and expect it to work. Building dynamic sites involves diving into the code and adjusting it to your own requirements. If that thought makes you break out in a cold sweat, just relax for a moment. PHP is not difficult, and I’ve written this book very much with the non-programmer in mind.

I’ve done so because I don’t come from a computing background myself. In fact, I went to school in the days before pocket calculators were invented, never mind personal computers. As a result, I don’t assume that you drank in knowledge of arrays, loops, and conditional statements with your mother’s milk. Everything is explained in plain, straightforward language, and I’ve highlighted points where things may go wrong, with advice on how to solve the problem. At the same time, if you’re working with computers and websites, you’re bound to have a certain level of technical knowledge and skill. So I don’t talk down to you either.

Over the years, I’ve read a lot of books about PHP and MySQL. The one thing that’s missing from all of them is any concept of visual design. So I decided to be different. I picked a handful of the best photographs I took on a visit to Japan in late 2005 and incorporated them into a site called Japan Journey (<http://foundationphp.com/phpsolutions/journey/>), which features throughout the book. I wanted to show that sites powered by PHP don’t have to look boring; in fact, they shouldn’t—visual appeal is an essential part of any website. All the pages are built in standards-compliant XHTML and styled with Cascading Style Sheets (CSS). However, the main focus remains firmly on working with PHP and MySQL, teaching you how to add a wealth of dynamic features to a website.

Some of the things you’ll learn by working through this book include the following:

- Displaying random images of different sizes
- Uploading images and automatically making copies that conform to a maximum size
- Creating an online photo gallery
- Building a navigation system to page through a long set of database results

- Displaying a summary of a long article and linking to the full text
- Protecting parts of your site with user authentication

You'll also learn how to process user input from every type of form element—text fields, drop-down menus, check boxes, and so forth. Most important of all, you'll see how a few simple checks can guard your websites and databases from malicious attack.

In this book, I've followed the same technique that has proved successful in *Foundation PHP 5 for Flash* and *Foundation PHP for Dreamweaver 8*. Each chapter takes you through a series of stages in a single project, with each stage building on the previous one. By working through the chapter, you get the full picture of how everything fits together. You can later refer back to the individual stages to refresh your memory about a particular technique. Although this isn't a reference book, Chapter 3 is a primer on PHP syntax, and some chapters contain short reference sections—notably Chapter 7 (reading from and writing to files), Chapter 9 (PHP sessions), Chapter 11 (MySQL data types and connection commands), and Chapter 13 (the four essential SQL commands).

So, to return to the original question: how easy is easy? I have done my best to ease your path, but there is no snake oil or magic potion. It will require some effort on your part. Don't attempt to do everything at once. Add new dynamic features to your site a few at a time. Get to understand how they work, and your efforts will be amply rewarded. Adding PHP and MySQL to your skills will enable you to build websites that offer much richer content and an interactive user experience.

It's been great fun writing this book, and the process has been smoothed all the way by the editorial team at friends of ED/Apress led admirably—as ever—by Chris Mills, the man with the psychedelic stuffed chicken (www.flickr.com/photos/chrisfills/124635002/). Special thanks go also to Samuel Wright for his helpful technical review, Kylie Johnston for keeping the project on an even keel, Nicole Flores and Ami Knox for their sensitive copy editing, Laura Cheu for overseeing the process of turning my words and pictures into the book you're now reading, and everybody else who toiled behind the scenes.

My greatest thanks of all go to you for buying this book. What do you mean you haven't bought it yet? Rush over to the checkout counter and buy it now. Then let the fun begin. If you enjoy what you're doing, then everything becomes easy.

1 WHAT IS PHP—AND WHY SHOULD I CARE?



What this chapter covers:

- Understanding what PHP can do
- Is PHP difficult?
- Is PHP safe?
- Using the download files

One of the first things most people want to know about PHP is what the initials stand for. Then they wish they had never asked. Officially, PHP stands for **PHP: Hypertext Preprocessor**. It's an ugly name that gives the impression that it's strictly for nerds or pro-pellerheads. Nothing could be further from the truth.

PHP is a scripting language that brings websites to life in the following ways:

- Sending feedback from your website directly to your mailbox
- Sending email with attachments
- Uploading files to a web page
- Watermarking images
- Generating thumbnails from larger images
- Displaying and updating information dynamically
- Using a database to display and store information
- Making websites searchable
- And much more . . .

PHP is easy to learn; it's platform-neutral, so the same code runs on Windows, Mac OS X, and Linux; and all the software you need to develop with PHP is open source and therefore free. There was a brief debate on the PHP General mailing list (<http://news.php.net/php.general>) in early 2006 about changing what PHP stands for. Small wonder, then, that it drew the comment that people who use PHP are Positively Happy People. The aim of this book is to help you become one too.

PHP started out as Personal Home Page in 1995, but it was decided to change the name a couple of years later, as it was felt that Personal Home Page sounded like something for hobbyists, and didn't do justice to the range of sophisticated features that had been added. Since then, PHP has developed even further, adding extensive support for object-oriented programming (OOP) in PHP 5. One of the language's great attractions, though, is that it remains true to its roots. You can start writing useful scripts very quickly without the need to learn lots of theory, yet be confident in the knowledge that you're using a technology with the capability to develop industrial-strength applications. Although PHP supports OOP, it's not an object-oriented language, and the scripts in this book concentrate on simpler techniques that are quick and easy to implement. If they help you to achieve what you want, great; if they inspire you to take your knowledge of PHP to the next level, even better.

Make no mistake, though. Using simple techniques doesn't mean the solutions you'll find in these pages aren't powerful. They are.

Embracing the power of code

If you're the sort of web designer or developer who uses a visual design tool, such as Dreamweaver, GoLive, or FrontPage, and never looks at the underlying code, it's time to rethink your approach. You're rapidly becoming an endangered species—and not the furry or cuddly sort that environmentalists will campaign to save from extinction. Good-looking design is definitely a top priority—and always will be—but it's no longer enough on its own. Designers need to have a solid grasp of the underlying structure of their pages. That means a knowledge of Hypertext Markup Language (HTML)—or its more recent incarnation, Extensible Hypertext Markup Language (XHTML)—and Cascading Style Sheets (CSS).

The CSS Zen Garden, cultivated by Dave Shea, played a pivotal role in convincing designers of the power of code. The underlying XHTML of every page showcased at www.csszengarden.com is identical, but as Figure 1-1 shows, the CSS produces stunningly different results. You don't need to be a CSS superhero, but as long as you have a good understanding of the basics of XHTML and CSS, you're ready to take your web design skills to the next stage by adding PHP to your arsenal.

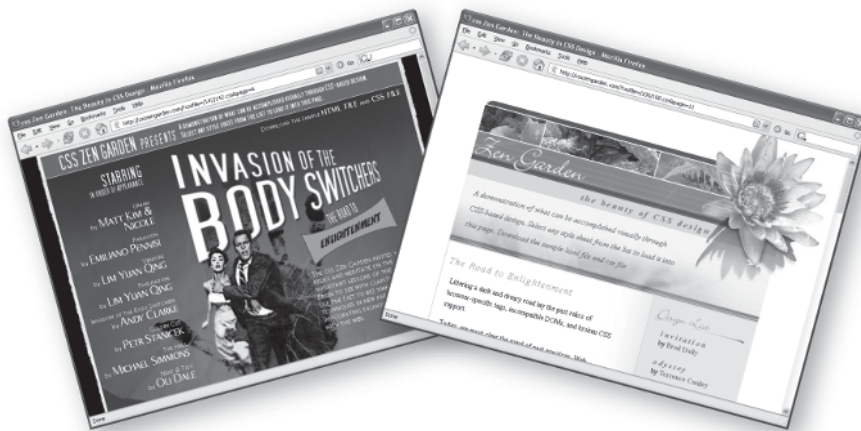


Figure 1-1. CSS Zen Garden has opened the eyes of web designers to the importance of code.

Creating pages that think for themselves

PHP is a **server-side language**. That means it runs on the web server, unlike CSS or JavaScript, which run on the client side (that is, the computer of the person visiting your site). This gives you much greater control. As long as the code works on your server, everyone receives the same output. For instance, Chapter 4 shows you how to create a random image generator with PHP. You can do the same thing with JavaScript, but what visitors to your site actually see depends on two things: JavaScript being enabled in their web browser, and the browser they are using understanding the version of JavaScript you have used. With PHP, this doesn't matter, because the dynamic process takes place entirely

on the server and creates the XHTML needed to display the page with a random choice of image. The server chooses the image filename and inserts it into the `<img>` tag before sending the page to the browser. You can even use images of different sizes, because the PHP code detects the dimensions of the image and inserts the correct width and height attributes.

What PHP does is enable you to introduce logic into your web pages. Chapter 3 covers this subject in detail, but this logic is based on alternatives. If it's Wednesday, show Wednesday's TV schedules . . . If the person who logs in has administrator privileges, display the admin menu; otherwise, deny access . . . that sort of thing.

PHP bases some decisions on information that it gleans from the server: the date, the time, the day of the week, information held in the page's URL, and so on. At other times, the decisions are based on user input, which PHP extracts from XHTML forms. As a result, you can create an infinite variety of output from a single script. For example, if you visit my blog at <http://foundationphp.com/blog/> (see Figure 1-2), and click various internal links, what you see is always the same page, but with different content. Admittedly, I tend to write always about the same kinds of subjects, but that's my fault, not PHP's.



Figure 1-2. Blogs are a good example of sites ideally suited to PHP.

Another website that I have created and maintained for several years, a subscription-only Japanese-language site (see Figure 1-3), is driven entirely by PHP. The navigation menu appears on every page of the site, but it's contained in a completely separate file, so if it

ever needs updating, I need to change only one page. Even though the menu is always generated by the same page, a little bit of PHP magic automatically highlights the correct button for the current page. You'll learn how to move an existing navigation bar to an external file and implement automatic highlighting in Chapter 4.

Because the site is subscription-only, users need to log in at the top right of the page to see the content, more than 14,000 articles in Japanese and English stored in a searchable database. When I log in, though, I get to see much more than anyone else: my security setting gives me administrator status, which enables me to insert new articles, edit existing ones, and register new users. You won't be building anything quite so ambitious in this book, but Chapters 9 through 15 teach you how to control access to your site with PHP sessions, as well as how to create a content management system with PHP and the MySQL relational database management system. Don't worry if you haven't worked with MySQL before; Chapter 10 shows you how to install it. Like PHP, it's open source and free for most users.

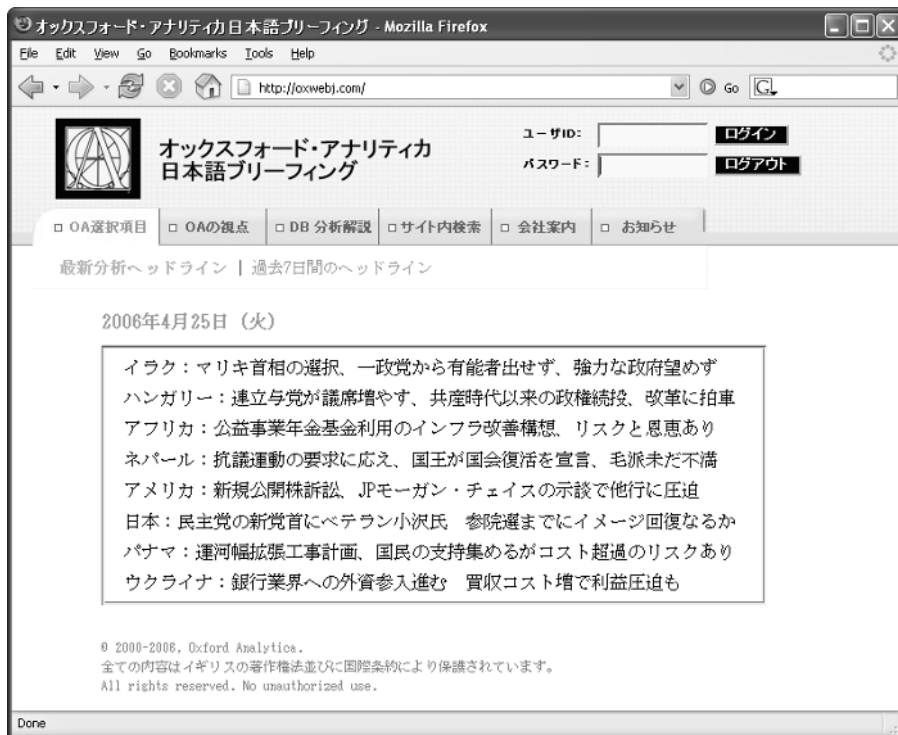


Figure 1-3. PHP not only drives all the logic behind this online database, but also restricts access to subscribers.

Other important uses for PHP in a website are sending email and uploading files, subjects covered in Chapters 5 and 6. By the time you finish this book, you'll wonder how you ever managed without PHP.

So how difficult is it going to be?

How hard is PHP to use and learn?

PHP isn't rocket science, but at the same time, don't expect to become an expert in five minutes. If you're a design-oriented person, you may find it takes time to get used to the way PHP is written. What I like about it very much is that it's succinct. For instance, in classic ASP, to display each word of a sentence on a separate line, you have to type out all this:

```
<%@ Language=VBScript %>
<% Option Explicit %>
<%
    Dim strSentence, arrWords, strWord
    strSentence = "ASP uses far more code to do the same as PHP"
    arrWords = Split(strSentence, " ", -1, 1)
    For Each strWord in arrWords
        Response.Write(strWord)
        Response.Write("<br />")
    Next
%>
```

In PHP, it's simply

```
<?php
$sentence = 'ASP uses far more code to do the same as PHP';
$words = explode(' ', $sentence);
foreach ($words as $word) {
    echo "$word<br />";
}
?>
```

That may not seem a big difference, but the extra typing gets very tiresome over a long script. PHP also makes it easy to recognize variables, because they always begin with \$. Most of the functions have very intuitive names. For example, `mysql_connect()` connects you to a MySQL database. Even when the names look strange at first sight, you can often work out where they came from. In the preceding example, `explode()` “blows apart” text and converts it into an array of its component parts. Don't worry if you don't know what variables, functions, or arrays are: they're all explained in Chapter 3, along with the other main things you need to know about the basics of PHP.

Perhaps the biggest shock to newcomers is that PHP is far less tolerant of mistakes than browsers are with XHTML. If you omit a closing tag in XHTML, most browsers will still render the page. If you omit a closing quote, semicolon, or brace in PHP, you'll get an uncomplaining error message like that shown in Figure 1-4. This isn't just a feature of PHP, but of all server-side technologies, including ASP, ASP.NET, and ColdFusion. It's why you need to have a reasonable understanding of XHTML and CSS before embarking on PHP. If the underlying structure of your web pages is shaky to start with, your learning curve with PHP will be considerably steeper.

A missing
parenthesis
turns this . . .



into this.

Figure 1-4. Server-side languages like PHP are intolerant of most coding errors.

PHP isn't like XHTML: you can't choose from a range of PHP editors that generate all the code for you automatically. Dreamweaver does have considerable support for PHP, and it automates a lot of code generation, mainly for integrating web pages with the MySQL database. Even so, most of the techniques in this book still need to be coded by hand in Dreamweaver. For more details of what Dreamweaver can do with PHP, see my book *Foundation PHP for Dreamweaver 8* (friends of ED, ISBN: 1-59059-569-6).

Can I just copy and paste the code?

There's nothing wrong with copying the code in this book. That's what it's there for. Copying is the way we all learn as children, but most of us progress from the copycat stage by asking questions and beginning to experiment on our own. Rather than attempt to teach you PHP by going through a series of boring exercises that have no immediate value to your web pages, I've structured this book so that you jump straight into applying your newfound knowledge to practical projects. At the same time, I explain what the code is for and why it's there. Even if you don't understand exactly how it all works, this should give you sufficient knowledge to know which parts of the code to adapt to your own needs and which parts are best left alone.

If you're completely new to PHP, I suggest that you read at least the first six chapters in the order they appear. Chapter 3 covers all the basics of writing PHP. The first half of the

chapter offers a bird's-eye view of the language and is probably all that you need to read before moving on to work with PHP in the following chapter. But you should come back regularly to the second half of Chapter 3 to fill in the details of PHP syntax. It's also a good idea to work through the PHP Solutions in each chapter in order, because each one builds on what goes before.

If you've already got the basics of PHP under your belt, you'll be able to hop about more freely, picking the solutions that are of more immediate interest to you. However, I recommend that you still read each chapter in its entirety. One of the features of this book is its emphasis on security. You may miss some important information if you read only part of a chapter.

How safe is PHP?

PHP is like the electricity or kitchen knives in your home: handled properly, it's very safe; handled irresponsibly, it can do a lot of damage. One of the inspirations for this book was the spate of email header injection attacks that erupted in late 2005. This type of attack exploits a vulnerability in a popular technique and enables the attacker to turn an online form into a spam relay. Few people were immune. I certainly wasn't, but once I was alerted to the problem, I plugged the hole and stopped the attacks in their tracks. However, day after day, people were sending frantic pleas for help to online forums. Even when they were told how to deal with the problem, their response became even more frantic. Many admitted they didn't know the first thing about any of the code they were using in their websites. For someone building websites as a hobby, this might be understandable, but many of these people were "professionals" who had built sites on behalf of clients. The clients were naturally unhappy when their mailboxes started filling with spam. They were no doubt even unhappier when their domains were suspended by hosting companies fed up with insecure scripts on their servers.

The moral of this story is not that PHP is unsafe; nor does everyone need to become a security expert to use PHP. What is important is to understand the basic principle of PHP safety: *always check user input before processing it*. You'll find that to be a constant theme throughout this book. Most security risks can be eliminated with very little effort. The other important thing is to know enough about scripts that you're using, so that if a problem arises, you can implement any remedies suggested to you by the author of the script or another expert.

How to use this book

PHP books tend to fall into three broad categories: beginner's tutorials, cookbooks for experienced users, and project-based books. This book tries to steer a middle course. It assumes no prior knowledge of PHP or MySQL, but is intended to be of equal value to designers and developers who already have some experience of these technologies. The approach I have taken is to explain each section of code in sufficient detail so that readers of all levels should be able to follow. However, the basic reference material is concentrated in Chapter 3, so more advanced readers shouldn't find themselves needing to wade through stuff they already know.

Because the book is aimed at web designers, most of the material centers on the Japan Journey site shown in Figure 1-4 (you can also view it online at <http://foundationphp.com/phpsolutions/site>). It's not intended to be a book-long case study that you're expected to build chapter by chapter. Most PHP books concentrate solely on code and pay zero attention to design, so the idea is to show you that pages built with PHP don't need to look ugly. You also see how to integrate PHP into an existing website. The emphasis is on enhancing your sites rather than building complex PHP applications from scratch.

Using the download files

PHP sites need to be located where the scripts can be processed by the web server. Normally, this means keeping them in a folder inside the Apache document root or an IIS virtual directory. Full instructions for setting up a local test environment are given in the next chapter. If you follow the recommendations there, Windows users should create a folder called `C:\htdocs\phpsolutions` if using Apache or create a virtual directory called `phpsolutions` in IIS. On Mac OS X, the `phpsolutions` folder should be located inside the `Sites` subfolder of your home folder.

A ZIP file containing the code for this book is available for download at www.friendsofed.com—it contains the following four folders:

- `assets`: CSS for the Japan Journey site and other pages
- `downloads`: All the source files arranged by chapter
- `images`: The images used on the Japan Journey site and other pages
- `includes`: Originally empty

Copy these four folders and their contents to the `phpsolutions` folder. When working with the example files in Chapter 3, view them in your browser by typing the following URL into the browser address bar on Windows (using the actual filename instead of *filename.php*):

```
http://localhost/phpsolutions/downloads/ch03/filename.php
```

On Mac OS X, use the following URL (using your own Mac username instead of *username* and the actual filename instead of *filename.php*):

```
http://localhost/~username/phpsolutions/downloads/ch03/filename.php
```

Most of the code for Chapter 4 and beyond should be copied from the appropriate subfolder of the `downloads` folder into the main `phpsolutions` folder (the Japan Journey site root). Where a page undergoes several changes in the course of a chapter, I have numbered the different versions like this: `index01.php`, `index02.php`, and so on. When copying a file into the site root, remove the number from the filename, so `index02.php` becomes `index.php`. If you are using a program like Dreamweaver, which prompts you to update links when moving files from one folder to another, do *not* update them. The files are all designed to pick up the correct images and stylesheets when located in the site root. I have done this so that you can use a file comparison utility to compare your code with mine (instructions for how to do this are in the next chapter).

The download files for each chapter contain a complete set of all files, apart from the images and stylesheets, which are common to all chapters. This means you can safely move back and forth through the book and always have the right files to work with. Each chapter gives instructions about which files to use and whether they need to be copied to a particular folder. The URL for the Japan Journey site on Windows is

```
http://localhost/phpsolutions/index.php
```

On Mac OS X the URL is

```
http://localhost/~username/phpsolutions/index.php
```

The layout of the Japan Journey site is controlled by CSS. Since this is a book about PHP, it doesn't go into details about the style rules or classes, although the stylesheets are fully commented. To brush up on your CSS skills, take a look at *Web Designer's Reference: An Integrated Approach to Web Design with XHTML and CSS* by Craig Grannell (friends of ED, ISBN: 1-59059-430-4) and *CSS Mastery: Advanced Web Standards Solutions* by Andy Budd (friends of ED, ISBN: 1-59059-614-5).

A note about versions

New versions of open source software are often released at a fast and furious pace. Most of the time, the new versions are just bug fixes, and the basic software is installed and operates in exactly the same way as in the previous versions. Sometimes, though, what should be a minor version upgrade results in significant changes that can confuse newcomers. This book is based on the following versions:

- Apache 2.2.3 and Apache 2.0.59 (Windows), Apache 1.3.33 (Mac)
- PHP 5.2.0 Release Candidate 4 (Windows), PHP 5.1.6 (Mac)
- MySQL 5.0.24
- phpMyAdmin 2.8.2.4

New versions will inevitably come out during the lifetime of this book. My advice is to install the most recent version available for your operating system. As this book was about to go to press, the PHP development team was in the final stages of testing PHP 5.2.0, the first official version compatible with Apache 2.2 on Windows. However, Mac OS X still ships with the Apache 1.3 series as the default installation. Quite honestly, the 1.3 series is more than adequate for a local testing environment.

By the time you read this, the Windows version of PHP should support Apache 2.2, but in case of an unforeseen hitch, the instructions in the next chapter cover both Apache 2.0 and 2.2. If there are any significant changes to the installation or operation of PHP, MySQL, or phpMyAdmin, they will be posted on the friends of ED website at www.friendsofed.com or my website at <http://foundationphp.com/phpsolutions>.

Some people go to great lengths to find old versions of PHP or MySQL so that they can install the same setup as their hosting company. This is totally unnecessary. If anything, you should be pressuring your hosting company to upgrade to the latest versions. Not only do

they have more features, but also they are usually safer. Nevertheless, this book has been written with both backward and forward compatibility in mind. Except where noted, all the code in this book should run on PHP 4.3.1 and MySQL 3.23.32 or later. I have also deliberately avoided using any code that is likely to break in PHP 6.

So, let's get on with it . . .

This chapter has provided only a brief overview of what PHP can do to add dynamic features to your websites and what you can expect from the rest of this book. The first stage in working with PHP is to set up a testing environment. The next chapter covers the process in detail for both Windows and Mac OS X.

2 GETTING READY TO WORK WITH PHP



What this chapter covers:

- Determining what you need
- Deciding whether to create a local testing setup
- Using a ready-made package
- Doing it yourself—setting up Apache and PHP on Windows and Mac OS X
- Getting PHP to work with IIS on Windows
- Making sure PHP has the right settings

Now that you've decided to use PHP to enrich your web pages, you need to make sure that you have everything you need to get on with the rest of this book. Although you can test everything on your remote server, it's usually more convenient to test PHP pages on your local computer. Everything you need to install is free. In this chapter, I'll explain the various options and give instructions for both Windows and Mac OS X.

What you need to write and test PHP pages

PHP is written in plain text, so you don't need any special authoring software. However, your life will be a lot easier if you choose a good script editor. I'll offer some advice on what to look for. The other thing you need is a web server capable of understanding PHP.

Checking whether your website supports PHP

The easiest way to find out whether your website supports PHP is to ask your hosting company. The other way to find out is to upload a PHP page to your website and see if it works. Even if you know that your site supports PHP, do the following test to confirm which version is running.

Checking the PHP version on your server

1. Open Notepad or TextEdit and type the following code into a blank page:

```
<?php echo phpversion(); ?>
```
2. Save the file as `phptest.php`. It's important to make sure that your operating system doesn't add a `.txt` filename extension after the `.php`. Mac users should also make sure that TextEdit doesn't save the file in Rich Text Format (RTF). If you're at all unsure, use `phptest.php` from the download files for this chapter.
3. Upload `phptest.php` to your website in the same way you would an HTML page, and then type the URL into a browser. If you see a three-part number like 5.2.0 displayed onscreen, you're in business: PHP is enabled. The number tells you which