O'REILLY®

Covers PHP 7

Learning PHP

A GENTLE INTRODUCTION TO THE WEB'S MOST POPULAR LANGUAGE

David Sklar

O'REILLY®

Learning PHP

If you want to get started with PHP, this book is essential. Author David Sklar (*PHP Cookbook*) guides you through aspects of the language you need to build dynamic server-side websites. By exploring features of PHP 5.x and the exciting enhancements in the latest release, PHP 7, you'll learn how to work with web servers, browsers, databases, and web services. End-of-chapter exercises help you make the lessons stick.

Whether you're a hobbyist looking to build dynamic websites, a frontend developer ready to add server-side programs, or an experienced programmer who wants to get up to speed with this language, this gentle introduction also covers aspects of modern PHP, such as internationalization, using PHP from the command line, and package management.

- Learn how PHP interacts with browsers and servers
- Understand data types, variables, logic, looping, and other language basics
- Explore how to use arrays, functions, and objects
- Build and validate web forms
- Work with databases and session management
- Access APIs to interact with web services and other websites
- Jumpstart your project with popular PHP web application frameworks

"David Sklar brings his deep technical knowledge and crystal clear communication style to bear in *Learning PHP*. Highly recommended."

—Thomas David Baker

David Sklar works as a Staff Software Engineer at Google. Before that, he built platforms, APIs, and sandboxed PHP execution runtimes at Ning. He's the author of *Learning PHP 5*, *Essential PHP Tools*, and coauthor of *PHP Cookbook*.

WEB DEVELOPMENT

US $41.99 CAN $48.99

ISBN: 978-1-491-93357-2

Twitter: @oreillymedia
facebook.com/oreilly

Learning PHP

A Gentle Introduction to the Web's Most Popular Language

David Sklar

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Learning PHP
by David Sklar

Copyright © 2016 David Sklar. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Allyson MacDonald
Production Editors: Colleen Lobner and Nicole Shelby
Copyeditor: Gillian McGarvey
Proofreader: Rachel Head

Indexer: Ellen Troutman-Zaig
Interior Designer: David Futato
Cover Designer: Randy Comer
Illustrator: Rebecca Demarest

April 2016: First Edition

Revision History for the First Edition
2016-04-07: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781491933572* for release details.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Learning PHP*, the cover image of an eagle, and related trade dress are trademarks of O'Reilly Media, Inc.

While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-491-93357-2

[LSI]

To M and S: may you never stop learning.

Table of Contents

Preface xi

1. Orientation and First Steps 1
PHP's Place in the Web World 1
What's So Great About PHP? 4
PHP Is Free (as in Money) 4
PHP Is Free (as in Speech) 5
PHP Is Cross-Platform 5
PHP Is Widely Used 5
PHP Hides Its Complexity 5
PHP Is Built for Web Programming 6
PHP in Action 6
Basic Rules of PHP Programs 12
Start and End Tags 13
Whitespace and Case-Sensitivity 14
Comments 15
Chapter Summary 17

2. Data: Working with Text and Numbers 19
Text 19
Defining Text Strings 20
Manipulating Text 24
Numbers 29
Using Different Kinds of Numbers 30
Arithmetic Operators 30
Variables 31
Operating on Variables 33
Putting Variables Inside Strings 34

Chapter Summary 36
Exercises 37

3. Logic: Making Decisions and Repeating Yourself. 39
Understanding true and false 40
Making Decisions 41
Building Complicated Decisions 43
Repeating Yourself 51
Chapter Summary 54
Exercises 55

4. Groups of Data: Working with Arrays. 57
Array Basics 57
Creating an Array 58
Choosing a Good Array Name 60
Creating a Numeric Array 60
Finding the Size of an Array 61
Looping Through Arrays 62
Modifying Arrays 68
Sorting Arrays 70
Using Multidimensional Arrays 74
Chapter Summary 77
Exercises 78

5. Groups of Logic: Functions and Files. 81
Declaring and Calling Functions 82
Passing Arguments to Functions 83
Returning Values from Functions 87
Understanding Variable Scope 92
Enforcing Rules on Arguments and Return Values 96
Running Code in Another File 98
Chapter Summary 100
Exercises 100

6. Data and Logic Together: Working with Objects. 103
Object Basics 104
Constructors 107
Indicating a Problem with Exceptions 108
Extending an Object 110
Property and Method Visibility 113
Namespaces 114
Chapter Summary 116

Exercises 117

7. Exchanging Information with Users: Making Web Forms. 119
Useful Server Variables 123
Accessing Form Parameters 124
Form Processing with Functions 127
Validating Data 129
Required Elements 131
Numeric or String Elements 131
Number Ranges 134
Email Addresses 135
<select> Menus 136
HTML and JavaScript 138
Beyond Syntax 141
Displaying Default Values 142
Putting It All Together 144
Chapter Summary 153
Exercises 153

8. Remembering Information: Databases. 155
Organizing Data in a Database 156
Connecting to a Database Program 158
Creating a Table 160
Putting Data into the Database 162
Inserting Form Data Safely 168
A Complete Data Insertion Form 170
Retrieving Data from the Database 173
Changing the Format of Retrieved Rows 178
Retrieving Form Data Safely 179
A Complete Data Retrieval Form 182
Chapter Summary 186
Exercises 187

9. Working with Files. 189
Understanding File Permissions 189
Reading and Writing Entire Files 190
Reading a File 190
Writing a File 192
Reading and Writing Parts of Files 192
Working with CSV Files 195
Inspecting File Permissions 198
Checking for Errors 199

Sanitizing Externally Supplied Filenames 202
Chapter Summary 204
Exercises 204

10. Remembering Users: Cookies and Sessions. 207
Working with Cookies 208
Activating Sessions 213
Storing and Retrieving Information 214
Configuring Sessions 218
Login and User Identification 219
Why setcookie() and session_start() Want to Be at the Top of the Page 226
Chapter Summary 228
Exercises 228

11. Talking to Other Websites and Services. 231
Simple URL Access with File Functions 231
Comprehensive URL Access with cURL 236
Retrieving URLs via GET 236
Retrieving URLs via POST 239
Using Cookies 240
Retrieving HTTPS URLs 243
Serving API Requests 244
Chapter Summary 247
Exercises 248

12. Debugging. 249
Controlling Where Errors Appear 249
Fixing Parse Errors 251
Inspecting Program Data 254
Adding Debug Output 255
Using a Debugger 258
Handling Uncaught Exceptions 261
Chapter Summary 263
Exercises 263

13. Testing: Ensuring Your Program Does the Right Thing. 265
Installing PHPUnit 266
Writing a Test 266
Isolating What You Test 270
Test-Driven Development 272
More Information About Testing 275
Chapter Summary 275

Exercises 276

14. Software Engineering Practices You Should Be Aware Of. 279
Source Control 280
Issue Tracking 281
Environments and Deployment 282
Scaling Eventually 283
Chapter Summary 284

15. Handling Dates and Times. 285
Displaying the Date or Time 285
Parsing a Date or Time 288
Calculating Dates and Times 290
Working with Timezones 291
Chapter Summary 292

16. Package Management. 293
Installing Composer 293
Adding a Package to Your Program 294
Finding Packages 295
Getting More Information on Composer 296
Chapter Summary 298

17. Sending Email. 299
Swift Mailer 299
Chapter Summary 301

18. Frameworks. 303
Laravel 304
Symfony 305
Zend Framework 307
Chapter Summary 309

19. Command-Line PHP. 311
Writing Command-Line PHP Programs 312
Using PHP's Built-in Web Server 313
Running a PHP REPL 314
Chapter Summary 316

20. Internationalization and Localization. 317
Manipulating Text 318
Sorting and Comparing 320

Localizing Output 321
Chapter Summary 323

A. Installing and Configuring the PHP Engine. 325

B. Answers to Exercises. 335

Index. 381

Preface

Boring websites are *static*. Interesting websites are *dynamic*—that is, their content changes. A giant static HTML page listing the names, pictures, descriptions, and prices of all 1,000 products a company has for sale is hard to use and takes forever to load. A dynamic web product catalog that lets you search and filter those products so you see only the six items that meet your price and category criteria is more useful, faster, and much more likely to close a sale.

The PHP programming language makes it easy to build dynamic websites. Whatever interactive excitement you want to create—whether it be as a product catalog, a blog, a photo album, or an event calendar—PHP is up to the task. And after reading this book, you'll be up to the task of building that dynamic website, too.

Who This Book Is For

This book will be useful for many different kinds of people:

- A hobbyist who wants to create an interactive website for himself, his family, or a nonprofit organization
- A website builder who wants to use the PHP setup provided by an ISP or hosting provider
- A developer or designer who needs to write a plugin or extension for a popular piece of software written in PHP, such as Drupal, WordPress, or MediaWiki
- A page designer who wants to communicate better with her developer coworkers
- A JavaScript whiz who wants to build server-side programs that complement her client-side code
- A Perl, Python, or Ruby programmer who wants to get up to speed with PHP
- Anybody who wants a straightforward, jargon-free introduction to one of the most popular programming languages for building interactive websites

PHP's gentle learning curve and approachable syntax make it an ideal "gateway" language for the nontechnical web professional. *Learning PHP* is aimed at this interested, intelligent, but not necessarily technical individual as well as at programmers familiar with another language who want to learn PHP.

If you are completely new to programming and embarking on your first interactive website, you've got the right book in your hands. The beginning chapters will give you a gentle introduction to the syntax of the PHP language and basic computer programming topics as they apply to PHP. Start at the beginning of the book and work your way forward.

If you are familiar with programming in another language but starting your first PHP project, you may want to start with the second section of the book and dip back into the first set of chapters when you have a specific question about syntax or how something basic is done in PHP.

Aside from basic computer literacy (knowing how to type, moving files around, surfing the Web), the only assumption that this book makes about you is that you're acquainted with HTML. You don't need to be an HTML wizard, but you should be comfortable with the HTML tags that populate a basic web page, such as `<html>`, `<head>`, `<body>`, `<p>`, `<a>`, and `<br>`. If you're not familiar with HTML, read *Head First HTML and CSS* by Elisabeth Robson and Eric Freeman (O'Reilly).

Contents of This Book

This book is designed so that you start at the beginning and work through the chapters in order. For the most part, each chapter depends on material in the previous chapters. Chapters 2 through 13 each end with exercises that test your understanding of the chapter's content.

Chapter 1 provides some general background on PHP and how it interacts with your web browser and a web server. It also shows some PHP programs and what they do, to give you an idea of what PHP programs look like. Especially if you're new to programming or building dynamic websites, it is important to read Chapter 1.

The next five chapters give you a grounding in the fundamentals of PHP. Before you can write great literature, you need to learn a little grammar and some vocabulary. That's what these chapters are for. (Don't worry—you'll learn enough PHP grammar and vocabulary right away to start writing some short programs, if not great literature.)

Chapter 2 shows you how to work with different kinds of data, such as pieces of text and numbers. This is important because the web pages that your PHP programs generate are just big pieces of text.

Chapter 3 describes the PHP commands that your programs can use to make decisions. These decisions are at the heart of the "dynamic" in *dynamic website*. The concepts in Chapter 3 are what you use, for example, to display only those items in a product catalog that fall between two prices a user enters in a web form.

Chapter 4 introduces *arrays*, which are collections of a bunch of individual numbers or pieces of text. Many frequent activities in PHP programs, such as processing submitted web form parameters or examining information pulled out of a database, involve using arrays.

As you write more complicated programs, you'll find yourself wanting to repeat similar tasks. *Functions*, discussed in Chapter 5, help you reuse pieces of your programs.

Chapter 6 shows how data and logic together are combined into *objects*. Objects are reusable bundles of code that help you structure your programs. Objects also allow you to integrate existing PHP add-ons and libraries into your code.

The next five chapters cover essential tasks in building a dynamic website: interacting with users, saving information, and interacting with other websites.

Chapter 7 supplies details on working with web forms, which are the primary way that users interact with your website.

Chapter 8 discusses databases. A database holds the information that your website displays, such as a product catalog or event calendar. This chapter shows you how to make your PHP programs talk to a database. With the techniques in Chapter 8, your website can do user-specific things such as display sensitive information only to authorized people or tell someone how many new message board posts have been created since she last logged in.

In addition to a database, you might also need to work with data stored in files. Chapter 9 explains to how read and write files from a PHP program.

Next, Chapter 10 details how to keep track of your users. This includes using cookies for transient data, but also users logging in to accounts and tracking session data such as a shopping cart of products.

The last chapter in this section, Chapter 11, delves into how your PHP program can interact with other websites and web services. You can retrieve the contents of other web pages or web APIs to use in your programs. Similarly, you can use PHP to serve up not just regular web pages but API responses to other clients.

Instead of new features you could incorporate into your programs, the next three chapters discuss things that help you be a better programmer.

Chapter 12 explains debugging: finding and fixing errors in your programs.

Chapter 13 shows how to write tests that exercise different parts of your program. These tests provide a way to make sure that your program does what you expect it to do.

Lastly, Chapter 14 talks about some aspects of software engineering that are not specific to PHP but that you should be familiar with as you work on projects with other developers.

The final section of the book is a collection of short explorations into a few common tasks and topics. These are not as fundamental as the material on the basic structure of PHP, or how to store information, but are still things that you're likely to run into as you spend time with PHP. These chapters give you the basics.

Chapter 15 shows PHP's powerful and comprehensive set of capabilities for working with dates and times. Chapter 16 discusses *package management*, with which you have a drop-dead simple way of incorporating useful libraries written by others into your code. Chapter 17 explains how to send email messages from your PHP program. Chapter 18 examines three popular PHP web application frameworks, which can jumpstart your project by eliminating a lot of common boilerplate code. Chapter 19 delves into using PHP from the command line (rather than from a web server), which can be a handy way to write simple utilities or test short programs. Finally, Chapter 20 lays out some techniques for successfully writing PHP programs that flawlessly handle text in different languages and character sets.

The two appendixes provide supplementary material. To run PHP programs, you need to have a copy of the PHP engine installed on your computer (or have an account with a web-hosting provider that supports PHP). Appendix A helps you get up and running, whether you are using Windows, OS X, or Linux.

Appendix B contains the answers to all the exercises in the book. No peeking until you've tried the exercises!

What's Not in This Book

This book is of finite length, so unfortunately it can't include everything there is to know about PHP. The primary goal of the book is to provide an introduction to PHP and to some of the basics of computer programming.

If you're already a PHP programmer and are primarily interested in what's new in PHP 7, *Upgrading to PHP 7* by Davey Shafik (O'Reilly) is a great place to look for all the details on what's new and different in this latest version of PHP. Bruno Skvorc's compilation of links and references at SitePoint also has a lot of great detail.

Other Resources

The online annotated PHP Manual is a great resource for exploring PHP's extensive function library. Plenty of user-contributed comments offer helpful advice and sample code, too. Additionally, there are many PHP mailing lists covering installation, programming, extending PHP, and various other topics. You can learn about and subscribe to these mailing lists at *php.net*. Also worth exploring is the PHP Presentation System archive. This is a collection of presentations about PHP that have been delivered at various conferences.

PHP The Right Way is also a splendid resource for getting to know PHP, especially if you're familiar with another programming language.

After you're comfortable with the material in this book, the following books about PHP are good next steps:

- *Programming PHP* by Rasmus Lerdorf, Peter MacIntyre, and Kevin Tatroe (O'Reilly). A more detailed and technical look at how to write PHP programs. Includes information on security, XML, and generating graphics.
- *PHP Cookbook* by David Sklar and Adam Trachtenberg (O'Reilly). A comprehensive collection of common PHP programming problems and their solutions.
- *Modern PHP* by Josh Lockhart (O'Reilly). This book is not about syntax and specific PHP tasks. Instead, it helps you write PHP with consistent, high-quality style and understand good practices for software engineering with PHP: it covers issues such as code deployment, testing, and profiling.

These books are helpful for learning about databases, SQL, and MySQL:

- *Learning PHP, MySQL & JavaScript* by Robin Nixon (O'Reilly). Explains how to make PHP, MySQL and JavaScript sing in harmony to make a robust dynamic website.
- *SQL in a Nutshell* by Kevin E. Kline, Daniel Kline, and Brand Hunt (O'Reilly). Covers the essentials you need to know to write SQL queries, and covers the SQL dialects used by Microsoft SQL Server, MySQL, Oracle, and PostgreSQL.
- *MySQL Cookbook* by Paul DuBois (O'Reilly). A comprehensive collection of common MySQL tasks.
- *MySQL Reference Manual*. The ultimate source for information about MySQL's features and SQL dialect.

Conventions Used in This Book

The following programming and typesetting conventions are used in this book.

Programming Conventions

The code examples in this book are designed to work with PHP 7.0.0. They were tested with PHP 7.0.5, which was the most up-to-date version of PHP 7 available at the time of publication. Where the book references or uses features added in PHP 5.4.0 or later, there is generally a mention of which version the feature was added in.

Typographical Conventions

The following typographical conventions are used in this book:

Italic
: Indicates new terms, example URLs, example email addresses, filenames, file extensions, pathnames, and directories.

`Constant width`
: Indicates commands, options, switches, variables, attributes, keys, functions, types, classes, namespaces, methods, modules, properties, parameters, values, objects, events, event handlers, XML tags, HTML tags, macros, the contents of files, or the output from commands.

`Constant width italic`
: Shows text that should be replaced with user-supplied values.

This icon signifies a tip, suggestion, or general note.

This icon indicates a warning or caution.

Using Code Examples

Typing some of the example programs in the book yourself is instructive when you are getting started. However, if your fingers get weary, you can download all of the code examples from *https://github.com/oreillymedia/Learning_PHP*.

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact the publisher for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: *Learning PHP* by David Sklar. Copyright 2016 David Sklar, 978-149-193357-2." If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact the publisher at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of plans and pricing for enterprise, government, education, and individuals.

Members have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and hundreds more. For more information about Safari Books Online, please visit us online.

Comments and Questions

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway
North Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international or local)
(707) 829-0104 (fax)

There is a web page for this book, where we list errata, examples, and any additional information. You can access this page at *http://bit.ly/learning_php*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

Or you can contact the author directly via his website, *http://www.sklar.com*.

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our website at *http://www.oreilly.com*.

Acknowledgments

This book is the end result of the hard work of many people. Thank you to:

- The many programmers, testers, documentation writers, bug fixers, and other folks whose time, talent, and devotion have made PHP the first-class development platform that it is today. Without them, I'd have nothing to write about.
- My diligent reviewers: Thomas David Baker and Phil McCluskey. They caught plenty of mistakes, turned confusing explanations into clear ones, and otherwise made this book far better than it would have been without them.
- My diligent editor: Ally MacDonald. The author is just one of the many pieces it takes to make a book and Ally made sure everything that needed to happen with all of those pieces actually happened!

For a better fate than wisdom, thank you also to Susannah, with whom I continue to enjoy ignoring the syntax of things.

CHAPTER 1

Orientation and First Steps

There are lots of great reasons to write computer programs in PHP. Maybe you want to learn PHP because you need to put together a small website that has some interactive elements. Perhaps PHP is being used where you work and you have to get up to speed. This chapter provides context for how PHP fits into the puzzle of website construction: what it can do and why it's so good at what it does. You'll also get your first look at the PHP language and see it in action.

PHP's Place in the Web World

PHP is a programming language that's used mostly for building websites. Instead of a PHP program running on a desktop computer for the use of one person, it typically runs on a web server and is accessed by lots of people using web browsers on their own computers. This section explains how PHP fits into the interaction between a web browser and a web server.

When you sit down at your computer and pull up a web page using a browser such as Safari or Firefox, you cause a little conversation to happen over the Internet between your computer and another computer. This conversation, and how it makes a web page appear on your screen, is illustrated in Figure 1-1.

Here's what's happening in the numbered steps of the diagram:

1. You type *www.example.com/catalog.html* into your browser's location bar.
2. The browser sends a message over the Internet to the computer named `www.example.com` asking for the */catalog.html* page.
3. Apache HTTP Server, a program running on the `www.example.com` computer, gets the message and reads the *catalog.html* file from its disk drive.

4. Apache sends the contents of the file back to your computer over the Internet as a response to the browser's request.
5. Your browser displays the page on your screen, following the instructions of the HTML tags in the page.

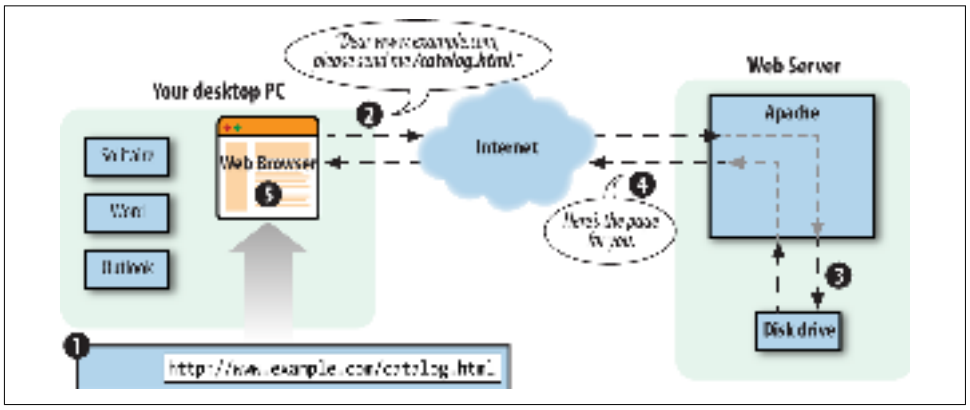

Figure 1-1. Client and server communication without PHP

Every time a browser asks for *http://www.example.com/catalog.html*, the web server sends back the contents of the same *catalog.html* file. The only time the response from the web server changes is if someone edits the file on the server.

When PHP is involved, however, the server does more work for its half of the conversation. Figure 1-2 shows what happens when a web browser asks for a page that is generated by PHP.

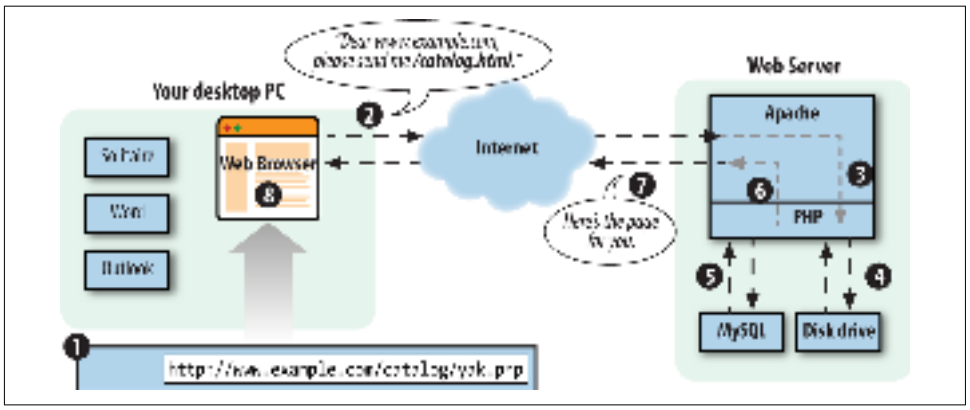

Figure 1-2. Client and server communication with PHP

Here's what's happening in the numbered steps of the PHP-enabled conversation:

1. You type *www.example.com/catalog/yak.php* into your browser's location bar.

2. Your browser sends a message over the Internet to the computer named `www.example.com` asking for the */catalog/yak.php* page.
3. Apache HTTP Server, a program running on the `www.example.com` computer, gets the message and asks the PHP engine, another program running on the `www.example.com` computer, "What does */catalog/yak.php* look like?"
4. The PHP engine reads the file *yak.php* from the disk drive.
5. The PHP engine runs the commands in *yak.php*, possibly exchanging data with a database program such as MySQL.
6. The PHP engine takes the *yak.php* program output and sends it back to Apache HTTP Server as an answer to "What does */catalog/yak.php* look like?"
7. Apache HTTP Server sends the page contents it got from the PHP engine back to your computer over the Internet in response to your browser's request.
8. Your browser displays the page on your screen, following the instructions of the HTML tags in the page.

PHP is a programming language. Something in the web server computer reads your PHP programs, which are instructions written in this programming language, and figures out what to do. The *PHP engine* follows your instructions. Programmers often say "PHP" when they mean either the programming language or the engine. In this book, just "PHP" means the programming language. "PHP engine" means the thing that follows the commands in the PHP programs you write and that generates web pages.

If PHP (the programming language) is like English (the human language), then the PHP engine is like an English-speaking person. The English language defines various words and combinations that, when read or heard by an English-speaking person, translate into various meanings that cause the person to do things such as feel embarrassed, go to the store to buy some milk, or put on pants. The programs you write in PHP (the programming language) cause the PHP engine to do things such as talk to a database, generate a personalized web page, or display an image.

This book is concerned with the details of writing those programs—i.e., what happens in step 5 of Figure 1-2 (although Appendix A contains details on configuring and installing the PHP engine on your own web server).

PHP is called a *server-side* language because, as Figure 1-2 illustrates, it runs on a web server. A language such as JavaScript can be used as a *client-side* language because, embedded in a web browser, it can cause that browser, while running on your desktop PC, to do something such as pop up a new window. Once the web server has sent the generated web page to the client (step 7 in Figure 1-2), PHP is out of the picture. If the page content contains some JavaScript, then that JavaScript runs on the client, but it is totally disconnected from the PHP program that generated the page.

A plain HTML web page is like the "sorry you found a cockroach in your soup" form letter you might get after dispatching an angry complaint to a bug-infested airline.

When your letter arrives at the airline's headquarters, the overburdened secretary in the customer service department pulls the "cockroach reply letter" out of the filing cabinet, makes a copy, and puts the copy in the mail back to you. Every similar request gets the exact same response.

In contrast, a dynamic page that PHP generates is like a postal letter you write to a friend across the globe. You can put whatever you like down on the page—doodles, diagrams, haikus, and tender stories of how unbearably cute your new baby is when she spatters mashed carrots all over the kitchen. The content of your letter is tailored to the specific person to whom it's being sent. Once you put that letter in the mailbox, however, you can't change it any more. It wings its way across the globe and is read by your friend. You don't have any way to modify the letter as your friend is reading it.

Now imagine you're writing a letter to an arts-and-crafts-inspired friend. Along with the doodles and stories you include instructions such as "Cut out the little picture of the frog at the top of the page and paste it over the tiny rabbit at the bottom of the page," and "Read the last paragraph on the page before any other paragraph." As your friend reads the letter, she also performs actions the letter instructs her to take. These actions are like JavaScript in a web page. They're set down when the letter is written and don't change after that. But when the reader of the letter follows the instructions, the letter itself can change. Similarly, a web browser obeys any JavaScript commands in a page and pops up windows, changes form menu options, or refreshes the page to a new URL.

What's So Great About PHP?

You may be attracted to PHP because it's free, because it's easy to learn, or because your boss told you that you need to start working on a PHP project next week. Since you're going to use PHP, you need to know a little bit about what makes it special. The next time someone asks you "What's so great about PHP?" use this section as the basis for your answer.

PHP Is Free (as in Money)

You don't have to pay anyone to use PHP. Whether you run the PHP engine on a beat-up 10-year-old PC in your basement or in a room full of million-dollar "enterprise-class" servers, there are no licensing fees, support fees, maintenance fees, upgrade fees, or any other kind of charge.

OS X and most Linux distributions come with PHP already installed. If yours doesn't, or you are using another operating system such as Windows, you can download PHP from *http://www.php.net*. Appendix A has detailed instructions on how to install PHP.

sample content of Learning PHP: A Gentle Introduction to the Web's Most Popular Language

- Sugar Street: The Cairo Trilogy, Volume 3 pdf, azw (kindle), epub, doc, mobi
- *click Dialectics and Deconstruction in Political Economy*
- download Beyond Hate: White Power and Popular Culture (The Cultural Politics of Media and Popular Culture) book
- **click The Ghosts of Athens (Aelric, Book 5) pdf, azw (kindle)**
- read online Vegan Italiano: Meat-free, Egg-free, Dairy-free Dishes from Sun-Drenched Italy pdf, azw (kindle)

- http://diy-chirol.com/lib/I-Love-You--Asshole---Green-s-Hill--Book-2-.pdf
- http://www.mmastyles.com/books/An-Anatomy-of-Addiction--Sigmund-Freud--William-Halsted--and-the-Miracle-Drug--Cocaine.pdf
- http://serazard.com/lib/The-Piracy-Crusade--How-the-Music-Industry-s-War-on-Sharing-Destroys-Markets-and-Erodes-Civil-Liberties.pdf
- http://diy-chirol.com/lib/The-Lords-of-Salem.pdf
- http://twilightblogs.com/library/Vegan-Italiano--Meat-free--Egg-free--Dairy-free-Dishes-from-Sun-Drenched-Italy.pdf