

# PHP & MySQL FOR DUMMIES®

## Learn to:

- Create well-formed PHP code that's compliant with PHP 4, 5, and 6
- Easily install and set up PHP and MySQL using XAMPP
- Choose a Web host and secure your files
- Build dynamic, database-delivered Web sites



Janet Valade

# PHP & MySQL® For Dummies®

## **Table of Contents**

### Introduction

[About This Book](#)

[Conventions Used in This Book](#)

[What You're Not To Read](#)

[Foolish Assumptions](#)

[How This Book Is Organized](#)

[Part I: Developing a Web Database Application Using PHP and MySQL](#)

[Part II: MySQL Database](#)

[Part III: PHP](#)

[Part IV: Applications](#)

[Part V: The Part of Tens](#)

[Part VI: Appendixes](#)

[Icons Used in This Book](#)

[Where to Go from Here](#)

### Part I: Developing a Web Database Application Using PHP and MySQL

#### Chapter 1: Introduction to PHP and MySQL

[What Is a Web Database Application?](#)

[The database: Storing data](#)

[The application: Moving data in and out of the database](#)

[MySQL, My Database](#)

[Advantages of MySQL](#)

[How MySQL works](#)

[Communicating with the MySQL server](#)

[PHP, a Data Mover](#)

[Advantages of PHP](#)

[How PHP works](#)

[MySQL and PHP, the Perfect Pair](#)

[Advantages of the relationship](#)

[How MySQL and PHP work together](#)

[Keeping Up with PHP and MySQL Changes](#)

[PHP versions](#)

[MySQL versions](#)

## [Chapter 2: Setting Up Your Work Environment](#)

[Anatomy of a Web Site](#)

[Building a Web Site](#)

[Deciding Where to Publish Your Web Site](#)

[Using a Web hosting company](#)

[Using a company Web site](#)

[Setting up your own server](#)

[Deciding Where to Develop Your Web Site](#)

[On your own computer](#)

[On another computer](#)

[Setting Up Your Web Site](#)

[With a Web hosting company](#)

[On a company Web site](#)

[Information you need](#)

[Setting Up Your Development Environment](#)

[Your own computer](#)

[Web hosting company](#)

[A company computer](#)

[Testing, Testing, 1,2,3](#)

[Understanding PHP/MySQL functions](#)

[Testing PHP](#)

[Testing your local PHP configuration file](#)

[Testing MySQL](#)

## [Chapter 3: Developing a Web Database Application](#)

[Planning Your Web Database Application](#)

[Identifying what you want from the application](#)

[Taking the user into consideration](#)

[Making the site easy to use](#)

[Leaving room for expansion](#)

[Writing it down](#)

## [Presenting the Two Running Examples in This Book](#)

[Stuff for Sale](#)

[Members Only](#)

## [Designing the Database](#)

[Choosing the data](#)

[Organizing the data](#)

## [Designing the Sample Databases](#)

[Pet Catalog design process](#)

[Members Only design process](#)

## [Types of Data](#)

[Character data](#)

[Numerical data](#)

[Date and time data](#)

[Enumeration data](#)

[MySQL data type names](#)

[Writing it down](#)

## [Taking a Look at the Sample Database Designs](#)

[Stuff for Sale database tables](#)

[Members Only database tables](#)

## [Developing the Application](#)

[Building the database](#)

[Writing the programs](#)

# [Part II: MySQL Database](#)

## [Chapter 4: Building the Database](#)

### [Communicating with MySQL](#)

[Building SQL queries](#)

[Sending SQL queries](#)

### [Building a Database](#)

[Creating a new database](#)

[Viewing the databases](#)

[Deleting a database](#)

[Adding tables to a database](#)

[Changing the database structure](#)

### [Moving Data Into and Out of the Database](#)

[Adding information](#)

[Viewing information](#)  
[Retrieving information](#)  
[Combining information from tables](#)  
[Updating information](#)  
[Removing information](#)

## [Chapter 5: Protecting Your Data](#)

[Controlling Access to Your Data](#)  
[Understanding account names and hostnames](#)  
[Finding out about passwords](#)  
[Taking a look at account privileges](#)  
[Setting Up MySQL Accounts](#)  
[Adding accounts](#)  
[Allowing access to a database](#)  
[Changing privileges](#)  
[Adding and changing passwords](#)  
[Removing accounts](#)  
[Backing Up Your Data](#)  
[Exporting your data with phpMyAdmin](#)  
[Viewing the Export file](#)  
[Restoring Your Data](#)

## [Part III: PHP](#)

### [Chapter 6: General PHP](#)

[Adding a PHP Section to an HTML Page](#)  
[Writing PHP Statements](#)  
[Using PHP Variables](#)  
[Naming a variable](#)  
[Creating and assigning values to variables](#)  
[Dealing with notices](#)  
[Using PHP Constants](#)  
[Working with Numbers](#)  
[Working with Character Strings](#)  
[Single-quoted strings versus double-quoted strings](#)  
[Joining strings](#)  
[Working with Dates and Times](#)  
[Setting local time](#)

[Formatting a date](#)

[Storing a timestamp in a variable](#)

[Using dates with MySQL](#)

### [Comparing Values](#)

[Making simple comparisons](#)

[Matching character strings to patterns](#)

[Joining Comparisons with and/or/xor](#)

[Adding Comments to Your Program](#)

## [Chapter 7: PHP Building Blocks for Programs](#)

### [Useful Simple Statements](#)

[Using echo statements](#)

[Using assignment statements](#)

[Using increment statements](#)

[Using exit](#)

[Using function calls](#)

### [Using PHP Arrays](#)

[Creating arrays](#)

[Viewing arrays](#)

[Removing values from arrays](#)

[Sorting arrays](#)

[Getting values from arrays](#)

[Walking through an array](#)

[Multidimensional arrays](#)

### [Useful Conditional Statements](#)

[Using if statements](#)

[Using switch statements](#)

### [Using Loops](#)

[Using for loops](#)

[Using while loops](#)

[Using do..while loops](#)

[Infinite loops](#)

[Breaking out of a loop](#)

### [Using Functions](#)

[Using variables in functions](#)

[Passing values between a function and the main program](#)

[Using built-in functions](#)

## [Chapter 8: Data In, Data Out](#)

[PHP and MySQL Functions](#)

[Making a Connection](#)

[Connecting to the MySQL server](#)

[Selecting the right database](#)

[Sending SQL queries](#)

[Getting Information from a Database](#)

[Sending a SELECT query](#)

[Getting and using the data](#)

[Using functions to get data](#)

[Getting Information from the User](#)

[Using HTML forms](#)

[Making forms dynamic](#)

[Using the information from the form](#)

[Checking the information](#)

[Giving users a choice with multiple submit buttons](#)

[Putting Information into a Database](#)

[Preparing the data](#)

[Adding new information](#)

[Updating existing information](#)

[Getting Information in Files](#)

[Using a form to upload the file](#)

[Processing the uploaded file](#)

[Putting it all together](#)

## [Chapter 9: Moving Information from One Web Page to the Next](#)

[Moving Your User from One Page to Another](#)

[Moving Information from Page to Page](#)

[Adding information to the URL](#)

[Storing information via cookies](#)

[Passing information with HTML forms](#)

[Using PHP Sessions](#)

[Understanding how PHP sessions work](#)

[Opening sessions](#)

[Using PHP session variables](#)

[Sessions without cookies](#)

[Making sessions private](#)

[Closing PHP sessions](#)

## [Part IV: Applications](#)

### [Chapter 10: Putting It All Together](#)

[Organizing the Application](#)

[Organizing at the application level](#)

[Organizing at the program level](#)

[Keeping It Private](#)

[Ensure the security of the computer](#)

[Don't let the Web server display filenames](#)

[Hide things](#)

[Don't trust information from users](#)

[Use a secure Web server](#)

[Completing Your Documentation](#)

### [Chapter 11: Building an Online Catalog](#)

[Designing the Application](#)

[Showing pets to the customers](#)

[Adding pets to the catalog](#)

[Building the Database](#)

[Building the Pet table](#)

[Building the PetType table](#)

[Building the Color table](#)

[Adding data to the database](#)

[Designing the Look and Feel](#)

[Showing pets to the customers](#)

[Adding pets to the catalog](#)

[Writing the Programs](#)

[Showing pets to the customers](#)

[Adding pets to the catalog](#)

### [Chapter 12: Building a Members Only Web Site](#)

[Designing the Application](#)

[Building the Database](#)

[Building the Member table](#)

[Building the Login table](#)

[Adding data to the database](#)

## [Designing the Look and Feel](#)

[Storefront page](#)

[Login page](#)

[New Member Welcome page](#)

[Members Only section](#)

## [Writing the Programs](#)

[Writing PetShopFront](#)

[Writing Login](#)

[Writing New\\_member](#)

[Writing the Members Only section](#)

## [Planning for Growth](#)

# [Part V: The Part of Tens](#)

## [Chapter 13: Ten Things You Might Want to Do Using PHP Functions](#)

[Communicate with MySQL](#)

[Send E-Mail](#)

[Use PHP Sessions](#)

[Stop Your Program](#)

[Handle Arrays](#)

[Check for Variables](#)

[Format Values](#)

[Compare Strings to Patterns](#)

[Find Out about Strings](#)

[Change the Case of Strings](#)

## [Chapter 14: Ten PHP Gotchas](#)

[Missing Semicolons](#)

[Not Enough Equal Signs](#)

[Misspelled Variable Names](#)

[Missing Dollar Signs](#)

[Troubling Quotes](#)

[Invisible Output](#)

[Numbered Arrays](#)

[Including PHP Statements](#)

[Missing Mates](#)

[Confusing Parentheses and Brackets](#)

## Part VI: Appendixes

[Appendix A: Installing PHP, MySQL, and Apache from XAMPP](#)

[Appendix B: Configuring PHP](#)

[In the general php.ini file](#)

[In a php.ini file on your Web site](#)

# ***PHP & MySQL® For Dummies®***

**by Janet Valade**



Wiley Publishing, Inc.

**PHP & MySQL® For Dummies®, 4th Edition**

Published by

**Wiley Publishing, Inc.**

111 River St.

Hoboken, NJ 07030-5774

[www.wiley.com](http://www.wiley.com)

Copyright ©2010 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600.

Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Trademarks:** Wiley, the Wiley Publishing logo, For Dummies, the Dummies Man logo, A Reference for the Rest of Us!, The Dummies Way, Dummies Daily, The Fun and Easy Way, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

**Limit of Liability/Disclaimer of Warranty:** The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Website is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Website may provide or recommendations it may make. Further, readers should be aware that Internet Websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002.

For technical support, please visit [www.wiley.com/techsupport](http://www.wiley.com/techsupport).

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2009940274

ISBN: 978-0-470-52758-0

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1



## About the Author

**Janet Valade** is the author of PHP & MySQL Web Development All-in-One Desk Reference For Dummies, PHP 5 For Dummies, PHP & MySQL Everyday Apps For Dummies, and PHP & MySQL: Your visual blueprint for creating dynamic, database-driven Web sites, as well as the first, second, and third editions of this book. In addition, Janet is the author of Spring into Linux and a coauthor of Mastering Visually Dreamweaver 8 and Flash 8.

Janet has twenty years of experience in the computing field. Most recently, she worked as a Web designer and programmer in an engineering firm for four years. Before that, Janet worked for thirteen years in a university environment, where she was a systems analyst. During her tenure, she supervised the installation and operation of computing resources, designed and developed a

data archive, supported faculty and students in their computer usage, wrote numerous technical papers, and developed and presented seminars on a variety of technology topics.

To keep in touch, see [janetvalade.com](http://janetvalade.com).

## Author's Acknowledgments

First, I want to express my appreciation to the entire open source community. Without those who give their time and talent, there would be no cool PHP and MySQL for me to write about. Furthermore, I never would have learned this software without the lists, where people generously spend their time answering foolish questions from beginners.

I want to thank my mother for passing on a writing gene, along with many other things. And my children always for everything. My thanks to my friends Art, Dick, and Marge for responding to my last-minute call for help. I particularly want to thank Sammy, Dude, Spike, Lucky, Upanishad, Sadie, and E.B. for their important contributions.

And, of course, I want to thank the professionals who make it all possible. Without the people at Wiley, this book would not exist. Because they all do their jobs so well, I can contribute my part to this joint project.

## Publisher's Acknowledgments

We're proud of this book; please send us your comments through our online registration form located at <http://dummies.custhelp.com>. For other comments, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002.

Some of the people who helped bring this book to market include the

following:

***Acquisitions, Editorial, and Media Development***

**Project Editor:** Rebecca Senninger (Previous Edition: Pat O'Brien)

**Acquisitions Editor:** Kyle Looper

**Copy Editor:** Virginia Sanders

**Technical Editor:** John Gosney

**Editorial Manager:** Leah Cameron

**Editorial Assistant:** Amanda Foxworth

**Sr. Editorial Assistant:** Cherie Case

**Cartoons:** Rich Tennant ([www.the5thwave.com](http://www.the5thwave.com))

***Composition Services***

**Project Coordinator:** Kristie Rees

**Layout and Graphics:** Melissa K. Jester, Christine Williams

**Proofreader:** Toni Settle

**Indexer:** BIM Indexing & Proofreading

***Publishing and Editorial for Technology Dummies***

**Richard Swadley**, Vice President and Executive Group Publisher

**Andy Cummings**, Vice President and Publisher

**Mary Bednarek**, Executive Acquisitions Director

**Mary C. Corder**, Editorial Director

## **Publishing for Consumer Dummies**

**Diane Graves Steele**, Vice President and Publisher

## **Composition Services**

**Debbie Stailey**, Director of Composition Services

# Introduction

---

Welcome to the exciting world of Web database applications. This book provides the basic techniques to build any Web database application, but I certainly recommend that you start with a simple one. In this book, I develop two sample applications, both chosen to represent two types of applications frequently encountered on the Web: product catalogs and customer- or member-only sites that require the user to register and log in with a password. The sample applications are complicated enough to require more than one program and to use a variety of data and data manipulation techniques, yet simple enough to be easily understood and adapted to a variety of Web sites. After you master the simple applications, you can expand the basic design to include all the functionality that you can think of.

## About This Book

Think of this book as your friendly guide to building a Web database application. This book is designed as a reference, not as a tutorial, so you don't have to read it from cover to cover. You can start reading at any point — in Chapter 1, Chapter 9, wherever. I divide the task of building a Web database application into manageable chunks of information, so check out the table of contents and locate the topic that you're interested in. If you need to know information from another chapter to understand the chapter you're reading, I reference that chapter.

Here's a sample of the topics I discuss:

- Building and using a MySQL database

- Adding PHP to HTML files

- Using the features of the PHP language

- Using HTML forms to collect information from users

Showing information from a database in a Web page

Storing information in a database

# Conventions Used in This Book

This book includes many examples of PHP programming statements, MySQL statements, and HTML. Such statements are shown in a different typeface, which looks like the following line:

A PHP program statement

In addition, snippets or key terms of PHP, MySQL, and HTML are sometimes shown in the text of a paragraph. When they are, the special text in the paragraph is also shown in the example typeface, different than the paragraph typeface. For instance, this text is an example of a PHP statement within the paragraph text.

In examples, you will often see some words in italic. Italicized words are general types that need to be replaced with the specific name appropriate for your data. For instance, when you see an example like the following:

```
SELECT field1,field2 FROM tablename
```

field1, field2, and tablename need to be replaced with real names because they are in italic. When you use this statement in your program, you might use it in the following form:

```
SELECT name,age FROM Customer
```

In addition, you might see three dots (...) following a list in an example line. You don't type the three dots. They just mean that you can have as many items in the list as you want. For instance, when you see

```
SELECT field1,field2,... FROM tablename
```

the three dots just mean that your list of fields can be longer than two. It means you can go on with field3, field4, and so forth. For example, your statement might be

```
SELECT name, age, height, shoesize FROM Customer
```

From time to time, you'll also see something in bold. Pay attention to these; they indicate something I want you to see or something you need to type.

## What You're Not To Read

Some information in this book is flagged as *Technical Stuff* with an icon off to the left. Sometimes you'll see this technical stuff in a sidebar: Consider it information that you don't need to read to create a Web database application. This extra information might contain a further look under the hood or describe a technique that requires more technical knowledge to execute. Some readers may be interested in the extra technical information or techniques, but feel free to ignore them if you don't find them interesting or useful.

## Foolish Assumptions

To write a focused book rather than an encyclopedia, I needed to assume some background for you, the reader. I assumed that you know HTML and CSS and have created Web sites with HTML and CSS. Consequently, although I use HTML/CSS in many examples, I do not explain the HTML/CSS. If you don't have an HTML background, this book will be more difficult to use. I suggest that you read an HTML book — such as *HTML, XHTML & CSS For Dummies* by Ed Tittel and Jeff Noble (Wiley) — and build some practice Web pages before you start this book. In particular, some background in HTML forms and tables is useful. However, if you're the impatient type, I won't tell you it's impossible to proceed without knowing HTML and CSS. You may be able to glean enough HTML and CSS from this book to build your particular Web site. If you choose to proceed without knowing HTML, I suggest that you have

an HTML book by your side to assist you.

If you're proceeding without any experience with Web pages, you might not know some required basics. You must know how to create and save plain text files with an editor such as Notepad or save the file as plain text from your word processor (not in the word processor format). You also must know where to put the text files containing the code (HTML or PHP) for your Web pages so that the pages are available to all users with access to your Web site, and you must know how to move the files to the appropriate location.

You do *not* need to know how to design or create databases or how to program. All the information that you need to know about databases and programming is included in this book.

## How This Book Is Organized

This book is divided into six parts, with several chapters in each part. The content ranges from an introduction to PHP and MySQL to installing to creating and using databases to writing PHP programs.

### Part I: Developing a Web Database Application Using PHP and MySQL

Part I provides an overview of using PHP and MySQL to create a Web database application. It describes and gives the advantages of PHP, of MySQL, and of their use together. You find out how to get started, including what you need, how to get access to PHP and MySQL, and how to test your software. You then find out about the process of developing the application.

### Part II: MySQL Database

In Part II you find out the details of working with MySQL databases. You create a database, change a database, and move data into and out of a database.

## **Part III: PHP**

Part III provides the details of writing PHP programs that enable your Web pages to insert new information, update existing information, or remove information from a MySQL database. You find out how to use the PHP features that are used for database interaction and forms processing.

## **Part IV: Applications**

Part IV describes the Web database application as a whole. You find out how to organize the PHP programs into a functioning application that interacts with the database. Two complete sample applications are provided, described, and explained.

## **Part V: The Part of Tens**

Part V provides some useful lists of important things to do and not to do when developing a Web database application.

## **Part VI: Appendixes**

The final part, Part VI, provides instructions for installing PHP and MySQL for those who need to install the software themselves. Appendix B discusses how to configure PHP.

# Icons Used in This Book



This icon is a sticky note of sorts, highlighting information that's worth committing to memory.



This icon flags information and techniques that are more technical than other sections of the book. The information here can be interesting and helpful, but you don't need to understand it to use the information in the book.



Tips provide extra information for a specific purpose. Tips can save you time and effort, so they're worth checking out.



You should always read warnings. Warnings emphasize actions that you must take or must avoid to prevent dire consequences.

## Where to Go from Here

This book is organized in the order in which things need to be done. If you're a newbie, you probably need to start with Part I, which describes how to get started, including how to design the pieces of your application and how the pieces will interact. When implementing your application, you need to create the MySQL database first, so I discuss MySQL before PHP. After you understand the details of MySQL and PHP, you need to put them together into a complete application, which I describe in Part IV. If you're already familiar

with any part of the book, you can go directly to the part that you need. For instance, if you're familiar with database design, you can go directly to Part II, which describes how to implement the design in MySQL. Or if you know MySQL, you can just read about PHP in Part III.

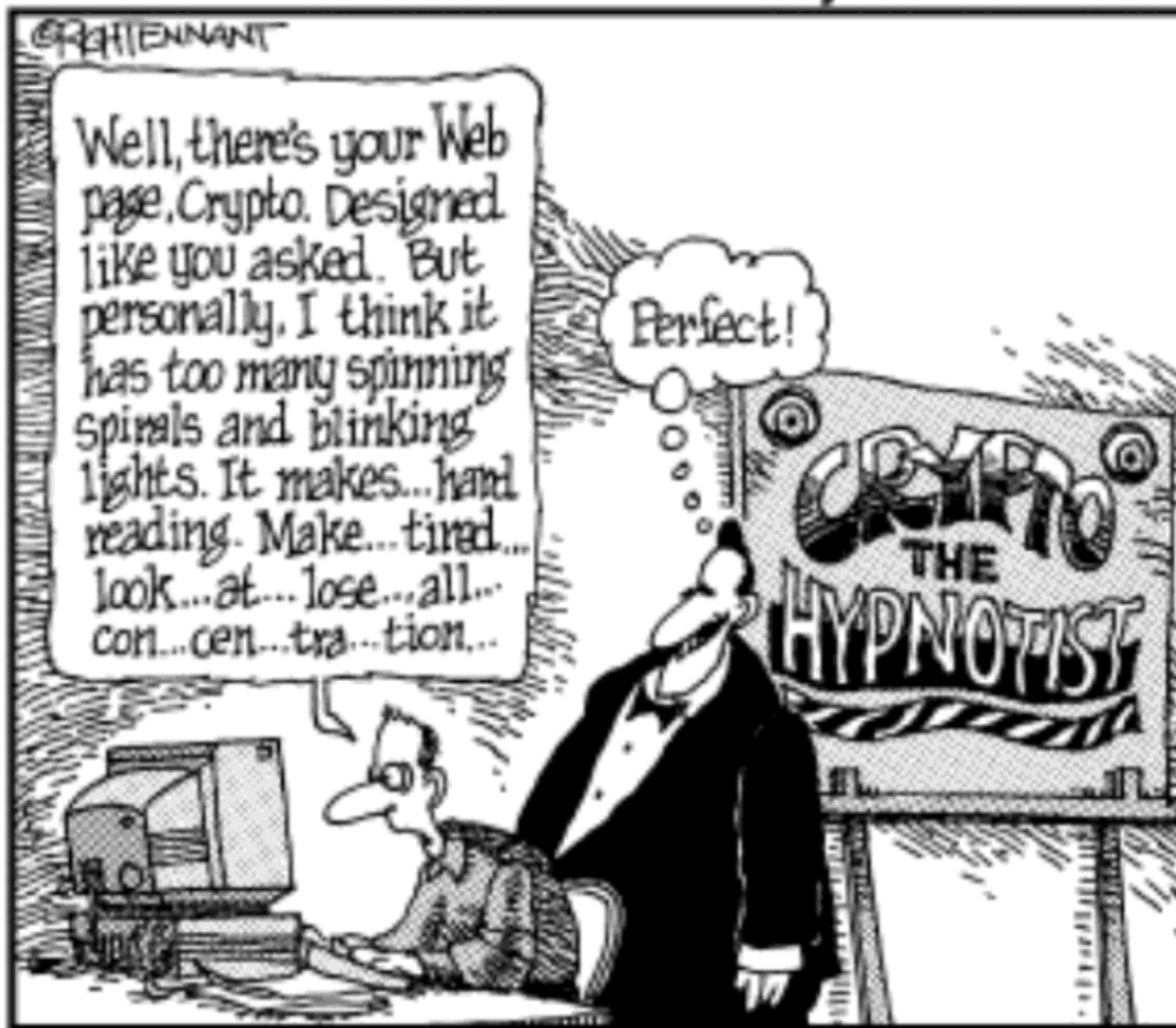
And if you want even *more* information, check out the cheat sheet at [www.dummies.com/cheatsheet/phpmysql](http://www.dummies.com/cheatsheet/phpmysql).

## Part I

# Developing a Web Database Application Using PHP and MySQL

The 5<sup>th</sup> Wave

By Rich Tennant



## In this part . . .

In this part, I provide an overview. I describe PHP and MySQL, how each one

works, and how they work together to make your Web database application possible. After describing your tools, I show you how to set up your working environment. I present your options for accessing PHP and MySQL and point out what to look for in each environment.

After describing your tools and your options for your development environment, I provide an overview of the development process. I discuss planning, design, and building your application.

# Chapter 1

# Introduction to PHP and MySQL

---

## ***In This Chapter***

Finding out what a Web database application is

Discovering how MySQL works

Taking a look at PHP

Finding out how PHP and MySQL work together

---

So you need to develop an interactive Web site. Perhaps your boss just put you in charge of the company's online product catalog. Or you want to develop your own Web business. Or your sister wants to sell her paintings online. Or you volunteered to put up a Web site open only to members of your circus acrobats' association. Whatever your motivation might be, you can see that the application needs to store information (such as information about products or member passwords), thus requiring a database. You can see also that the application needs to interact *dynamically* with the user; for instance, the user selects a product to view or enters membership information. This type of Web site is a *Web database application*.

I assume that you've created static Web pages before, using HTML (HyperText Markup Language), but creating an interactive Web site is a new challenge, as is designing a database. You asked three computer gurus you know what you should do. They said a lot of things you didn't understand, but among the technical jargon, you heard "quick" and "easy," and "free" mentioned in the same sentence as PHP and MySQL. Now you want to know more about using PHP and MySQL to develop the Web site that you need.

PHP and MySQL work together very well; it's a dynamic partnership. In this chapter, you find out the advantages of each, how each one works, and how they work together to produce a dynamic Web database application.

# What Is a Web Database Application?

An *application* is a program or a group of programs designed for use by an *end user* (for example, customers, members, or circus acrobats). If the end user interacts with the application via a Web browser, the application is a *Web based* or *Web application*. If the Web application requires the long-term storage of information using a database, it's a *Web database application*. This book provides you with the information that you need to develop a Web database application that can be accessed with Web browsers such as Internet Explorer and Firefox.

A Web database application is designed to help a user accomplish a task. It can be a simple application that displays information in a browser window (for example, current job openings when the user selects a job title) or a complicated program with extended functionality (for example, the book-ordering application at Amazon.com or the bidding application at eBay).

A Web database application consists of just two pieces:

**Database:** The *database* is the long-term memory of your Web database application. The application can't fulfill its purpose without the database. However, the database alone is not enough.

**Application:** The *application* piece is the program or group of programs that performs the tasks. Programs create the display that the user sees in the browser window; they make your application interactive by accepting and processing information that the user types in the browser window; and they store information in the database and get information out of the database. (The database is useless unless you can

move data in and out.)

The Web pages that you've previously created with HTML alone are *static*, meaning the user can't interact with the Web page. All users see the same Web page. *Dynamic* Web pages, on the other hand, allow the user to interact with the Web page. Different users might see different Web pages. For instance, one user looking at a furniture store's online product catalog might choose to view information about the sofas, whereas another user might choose to view information about coffee tables. To create dynamic Web pages, you must use another language in addition to HTML.

One language widely used to make Web pages dynamic is JavaScript. JavaScript is useful for several purposes, such as *mouse-overs* (for example, to highlight a navigation button when the user moves the mouse pointer over it) or accepting and validating information that users type into a Web form. However, it's not useful for interacting with a database. You wouldn't use JavaScript to move the information from the Web form into a database. PHP, however, is a language particularly well suited to interacting with databases. PHP can accept and validate the information that users type into a Web form and can also move the information into a database. The programs in this book are written with PHP.

## The database: Storing data

The core of a Web database application is the *database*, which is the long-term memory (I hope more efficient than my long-term memory) that stores information for the application. A database is an electronic file cabinet that stores information in an organized manner so that you can find it when you need it. After all, storing information is pointless if you can't find it. A database can be small, with a simple structure — for example, a database containing the titles and authors' names of all the books that you own. Or a database can be huge, with an extremely complex structure — such as the database that Amazon.com has to hold all its information.

The information that you store in the database comes in many varieties. A company's online catalog requires a database to store information about all the

company's products. A membership Web site requires a database to store information about members. An employment Web site requires a database (or perhaps two databases) to store information about job openings and information from résumés. The information that you plan to store could be similar to information that's stored by Web sites all over the Internet — or information that's unique to your application.

The term *database* refers to the file or group of files that holds the actual data. The data is accessed by using a set of programs called a DBMS (Database Management System). Almost all DBMSs these days are RDBMSs (Relational Database Management Systems), in which data is organized and stored in a set of related tables.

In this book, MySQL is the RDBMS used because it's particularly well suited for Web sites. MySQL and its advantages are discussed in the section, "MySQL, My Database," later in this chapter. You can find out how to organize and design a MySQL database in Chapter 3.

## The application: Moving data in and out of the database

For a database to be useful, you need to be able to move data into and out of it. Programs are your tools for this because they interact with the database to store and retrieve data. A program connects to the database and makes a request: "Take this data and store it in the specified location." Another program makes the request: "Find the specified data and give it to me." The application programs that interact with the database run when the user interacts with the Web page. For instance, when the user clicks the submit button after filling in a Web form, a program processes the information in the form and stores it in a database.

### E-mail discussion lists

Good technical support is available from *e-mail discussion lists*, which are groups of people discussing specific topics through e-mail. E-mail

lists are available for pretty much any subject you can think of: Powerball, ancient philosophy, cooking, The Beatles, Scottish terriers, politics, and so on. The *list manager* maintains a distribution list of e-mail addresses for anyone who wants to join the discussion. When you send a message to the discussion list, your message is sent to the entire list so that everyone can see it. Thus, the discussion is a group effort, and anyone can respond to any message that interests him or her.

E-mail discussion lists are supported by various sponsors. Any individual or organization can run a list. Most software vendors run one or more lists devoted to their software. Universities run many lists for educational subjects. In addition, some Web sites manage discussion lists, such as Yahoo! Groups and Topica. Users can create a new list or join an existing list through the Web application.

Software-related e-mail lists are a treasure trove of technical support. Anywhere from a hundred to several thousand users of the software subscribe to the list. Often the developers, programmers, and technical support staff for the software vendor are on the list. You're unlikely to be the first person to ever experience your problem. Whatever your question or problem, someone on the list probably knows the answer or the solution. When you post a question to an e-mail list, the answer usually appears in your inbox within minutes. In addition, most lists maintain an archive of previous discussions so that you can search for answers. When you're new to any software, you can find out a great deal simply by joining the discussion list and reading the messages for a few days.

PHP and MySQL have e-mail discussion lists. Actually, each has several discussion lists for special topics, such as *databases and PHP*. You can find the names of the mailing lists and instructions for joining them on the PHP ([www.php.net](http://www.php.net)) and MySQL ([www.mysql.com](http://www.mysql.com)) Web sites.

# MySQL, My Database

MySQL is a fast, easy-to-use RDBMS used on many Web sites. Speed was the developers' main focus from the beginning. In the interest of speed, they made the decision to offer fewer features than their major competitors (such as Oracle and Sybase). However, even though MySQL is less full-featured than its commercial competitors, it has all the features needed by the majority of database developers. It's easier to install and use than its commercial competitors, and the difference in price is strongly in favor of MySQL.

MySQL was developed originally by a Swedish company but is now developed, marketed, and supported by Sun Microsystems. The company licenses it in two ways:

**MySQL Community Server:** A freely downloadable, open source edition of MySQL, released early and often with the most advanced features. Anyone who can meet the requirements of the GPL can use the software for free. If you're using MySQL as a database on a Web site (the subject of this book), you can use MySQL for free, even if you're making money with your Web site.

**MySQL Enterprise Subscription:** A comprehensive offering of production support, monitoring tools, and MySQL database software. For a subscription fee paid per year per server, monthly software updates, consulting services, technical support, and other services are available. You can choose the level of services you want for the fee that you want to pay.



Finding technical support for MySQL Community Server is not a problem. You can join one of several e-mail discussion lists offered on the MySQL Web site at [www.mysql.com](http://www.mysql.com). You can even search the e-mail list archives, which contain a large archive of MySQL questions and answers.

## Advantages of MySQL

MySQL is a popular database with Web developers. Its speed and small size make it ideal for a Web site. Add to that the fact that it's open source, which means free, and you have the foundation of its popularity. Here's a rundown of some of its advantages:

**It's fast.** The main goal of the folks who developed MySQL was speed. Thus, the software was designed from the beginning with speed in mind.

**It's inexpensive.** MySQL is free under the open source GPL license, and the fee for a commercial license is reasonable.

**It's easy to use.** You can build and interact with a MySQL database by using a few simple statements in the SQL language, which is the standard language for communicating with RDBMSs. Check out Chapter 4 for the lowdown on the SQL language.

**It can run on many operating systems.** MySQL runs on many operating systems — Windows, Linux, Mac OS, most varieties of Unix (including Solaris and AIX), FreeBSD, OS/2, Irix, and others.

**It's available on almost all Web hosts.** If you're going to run your Web site on a Web hosting company, MySQL is widely available without extra cost. Using MySQL on a Web host is discussed in more detail in Chapter 2.

**Technical support is widely available.** A large base of users provides free support through mailing lists. The MySQL developers also participate in the e-mail lists.

**It's secure.** MySQL's flexible system of authorization allows some or all database privileges (such as the privilege to create a database or delete data) to specific users or groups of users. Passwords are encrypted.

**It supports large databases.** MySQL handles databases up to 50 million rows or more. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

**It's customizable.** The open source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

# How MySQL works

The MySQL software consists of the MySQL server, several utility programs that assist in the administration of MySQL databases, and some supporting software that the MySQL server needs (but you don't need to know about). The heart of the system is the MySQL server.

The MySQL server is the manager of the database system. It handles all your database instructions. For instance, if you want to create a new database, you send a message to the MySQL server that says “create a new database and call it newdata.” The MySQL server then creates a subdirectory in its data directory, names the new subdirectory newdata, and puts the necessary files with the required format into the newdata subdirectory. In the same manner, to add data to that database, you send a message to the MySQL server, giving it the data and telling it where you want the data to be added. You find out how to write and send messages to MySQL in Part II.

Before you can pass instructions to the MySQL server, it must be running and waiting for requests. The MySQL server is usually set up so that it starts when the computer starts and continues running all the time. This is the usual setup for a Web site. However, it's not necessary to set it up to start when the computer starts. If you need to, you can start it manually whenever you want to access a database. When it's running, the MySQL server listens continuously for messages that are directed to it.

## Communicating with the MySQL server

All your interaction with the database is accomplished by passing messages to the MySQL server. You can send messages to the MySQL server several ways, but this book focuses on sending messages using PHP. The PHP software has specific statements that you use to send instructions to the

MySQL server.

The MySQL server must be able to understand the instructions that you send it. You communicate by using *SQL* (*Structured Query Language*), which is a standard language understood by many RDBMSs. The MySQL server understands SQL. PHP doesn't understand SQL, but it doesn't need to: PHP just establishes a connection with the MySQL server and sends the SQL message over the connection. The MySQL server interprets the SQL message and follows the instructions. The MySQL server sends a return message, stating its status and what it did (or reporting an error if it was unable to understand or follow the instructions).

Software designed specifically to interact with MySQL database is also discussed in this book. You can use this software, called phpMyAdmin, on your own computer to communicate with your MySQL databases. PhpMyAdmin is also available on almost all Web hosts.

For the lowdown on how to write and send SQL messages to MySQL, check out Part II.

## PHP, a Data Mover

*PHP*, a scripting language designed specifically for use on the Web, is your tool for creating dynamic Web pages. Rich in features that make Web design and programming easier, PHP is in use on more than 20 million domains (according to the Netcraft survey at [www.php.net/usage.php](http://www.php.net/usage.php)). Its popularity continues to grow, so it must be fulfilling its function pretty well.

PHP stands for *PHP: HyperText Preprocessor*. In its early development by a guy named Rasmus Lerdorf, it was called *Personal Home Page tools*. When it developed into a full-blown language, the name was changed to be more in line with its expanded functionality.

needed to create dynamic Web pages. The PHP language is designed to be included easily in an HTML file.

**It can run on many operating systems.** It runs on a variety of operating systems — Windows, Linux, Mac OS, and most varieties of Unix.

**It's available on almost all Web hosts.** If you are going to publish your Web site on a Web host, you will find PHP installed on almost all Web hosts for free.

**Technical support is widely available.** A large base of users provides free support through e-mail discussion lists.

**It's secure.** The user does not see the PHP code.

**It's designed to support databases.** PHP includes functionality designed to interact with specific databases. It relieves you of the need to know the technical details required to communicate with a database.

**It's customizable.** The open source license allows programmers to modify the PHP software, adding or modifying features as needed to fit their own specific environments.

## How PHP works

PHP is an embedded scripting language when used in Web pages. This means that PHP code is embedded in HTML code. You use HTML tags to enclose the PHP language that you embed in your HTML file — the same way that you would use other HTML tags. You create and edit Web pages containing PHP the same way that you create and edit regular HTML pages.

The PHP software works with the Web server. The Web server is the software that delivers Web pages to the world. When you type a URL into your Web browser, you're sending a message to the Web server at that URL, asking it to send you an HTML file. The Web server responds by sending the requested file. Your browser reads the HTML file and displays the Web page. You also request the Web server to send you a file when you click a link in a Web page. In addition, the Web server processes a file when you click a Web page button

that submits a form.

When PHP is installed, the Web server is configured to expect certain file extensions to contain PHP language statements. Often the extension is .php or .phtml, but any extension can be used. When the Web server gets a request for a file with the designated extension, it sends the HTML statements as is, but PHP statements are processed by the PHP software before they're sent to the requester.

When PHP language statements are processed, only the output is sent by the Web server to the Web browser. The PHP language statements are not included in the output sent to the browser, so the PHP code is secure and transparent to the user. For instance, in this simple PHP statement:

```
<?php echo "<p>Hello World</p>"; ?>
```

<?php is the PHP opening tag, and ?> is the closing tag. echo is a PHP instruction that tells PHP to output the upcoming text. The PHP software processes the PHP statement and outputs this:

```
<p>Hello World</p>
```

which is a regular HTML statement. This HTML statement is delivered to the user's browser. The browser interprets the statement as HTML code and displays a Web page with one paragraph — Hello World. The PHP statement is not delivered to the browser, so the user never sees any PHP statements. PHP and the Web server must work closely together.

PHP is not integrated with all Web servers but does work with many of the popular Web servers. PHP is developed as a project of the Apache Software Foundation — thus, it works best with Apache. PHP also works with Microsoft IIS/PWS, iPlanet (formerly Netscape Enterprise Server), and others.



Although PHP works with several Web servers, it works best with Apache. If you can select or influence the selection of the Web server used in your organization, select Apache. By itself, Apache is a good

choice. It's free, open source, stable, and popular. It currently powers more than 60 percent of all Web sites, according to the Web server survey at [www.netcraft.com](http://www.netcraft.com). It runs on Windows, Linux, Mac OS, and most flavors of Unix.

# MySQL and PHP, the Perfect Pair

MySQL and PHP are frequently used together. They're often called the *dynamic duo*. MySQL provides the database part, and PHP provides the application part of your Web database application.

## Advantages of the relationship

MySQL and PHP as a pair have several advantages:

**They're free.** It's hard to beat free for cost-effectiveness.

**They're Web oriented.** Both were designed specifically for use on Web sites. Both have a set of features focused on building dynamic Web sites.

**They're easy to use.** Both were designed to get a Web site up quickly.

**They're fast.** Both were designed with speed as a major goal. Together they provide one of the fastest ways to deliver dynamic Web pages to users.

**They communicate well with one another.** PHP has built-in features for communicating with MySQL. You don't need to know the technical details; just leave it to PHP.

# How MySQL and PHP work together

PHP provides the application part, and MySQL provides the database part of a Web database application. You use the PHP language to write the programs that perform the application tasks. PHP can be used for simple tasks (such as displaying a Web page) or for complicated tasks (such as accepting and verifying data that a user typed into an HTML form). One of the tasks that your application must do is move data into and out of the database — and PHP has built-in features to use when writing programs that move data into and out of a MySQL database.

PHP statements are embedded in your HTML files with PHP tags. When the task to be performed by the application requires storing or retrieving data, you use specific PHP statements designed to interact with a MySQL database. You use one PHP statement to connect to the correct database, telling PHP where the database is located, its name, and the password needed to connect to it. The database doesn't need to be on the same machine as your Web site; PHP can communicate with a database across a network. You use another PHP statement to send an SQL message to MySQL, giving MySQL instructions for the task you want to accomplish. MySQL returns a status message that shows whether it successfully performed the task. If a problem came up, it returns an error message. If your SQL message asked to retrieve some data, MySQL sends the data that you asked for, and PHP stores it in a temporary location where it's available to you.

You then use one or more PHP statements to complete the application task. For instance, you can use PHP statements to display data that you retrieved. Or you might use PHP statements to display a status message in the browser, informing the user that the data was saved.

As an RDBMS, MySQL can store complex information. As a scripting language, PHP can perform complicated manipulations of data, on either data that you need to modify before saving it in the database or data that you retrieved from the database and need to modify before displaying or using it for another task. Together, PHP and MySQL can be used to build a

sophisticated and complicated Web database application.

# Keeping Up with PHP and MySQL Changes

PHP and MySQL are open source software. If you've used only software from major software publishers — such as Microsoft, Macromedia, or Adobe — you'll find that open source software is an entirely different species. It's developed by a group of programmers who write the code in their spare time, for fun and for free. There's no corporate office.

Open source software changes frequently, rather than once every year or two like commercial software does. It changes when the developers feel that it's ready. It also changes quickly in response to problems. When a serious problem is found — such as a security hole — a new version that fixes the problem can be released in days. You don't receive glossy brochures or see splashy magazine ads for a year before a new version is released. Thus, if you don't make the effort to stay informed, you could miss the release of a new version or be unaware of a serious problem with your current version.



Visit the PHP and MySQL Web sites often. You need to know the information that's published there. Join the mailing lists, which often are high in traffic. When you first get acquainted with PHP and MySQL, the large number of mail messages on the discussion lists brings valuable information into your e-mail inbox; you can pick up a lot by reading those messages. And soon, you might be able to help others based on your own experience. At the very least, subscribe to the announcement mailing list, which delivers e-mail only occasionally. Any important problems or new versions are announced here. The e-mail that you receive from the announcement list contains information you need to know. So, right now, before you forget, hop over to the PHP and MySQL Web sites and sign up for a list or two at

[www.php.net/mailing-lists.php](http://www.php.net/mailing-lists.php) and [lists.mysql.com](http://lists.mysql.com).

## PHP versions

The current version of PHP is PHP 5. Some existing applications still run PHP 4, but because you're building your first PHP application, you should be using PHP 5.

PHP 6 is due to be released soon. Perhaps it has already been released by the time you're reading this book. When PHP changes from version 5 to version 6, the following important changes will occur:

The setting for `register_globals` will no longer exist.

The setting for magic quotes will no longer exist.

The long arrays, such as `HTTP_POST_VARS`, will no longer exist. These arrays were commonly used in PHP 4.



If you're ever converting scripts that ran under PHP 4 or 5 to run under PHP 6, you may need to make changes to the scripts, based on the preceding changes, before the scripts will run correctly under PHP 6. I explain these changes throughout the book where they apply to the techniques and procedures.

## MySQL versions

MySQL 5.1 is the current version, as of this writing. MySQL 5.0 is also available. The examples and scripts in this book run equally well under either version. Some of the more advanced features of 5.1 may not be available on sites running 5.0, but none of those advanced features are discussed in this book.

MySQL 6 is also available for download on the MySQL Web site. However, at

view the Web page at that address. Software at the Web site, called a Web server, receives the request and responds by sending the requested Web page. The browser receives the Web page file and displays the Web page in the browser window.

To make your Web site available to the public, you place the text files containing HTML code on the Web site where users can access them. A Web database application is similar. The difference is that the files contain PHP code, as well as HTML code.

To provide the dynamic Web database applications discussed in this book, your Web site must have the following software:

**A Web server:** The software that delivers your Web pages to the world

**MySQL:** The RDBMS (Relational Database Management System) that will store information for your Web database application

**PHP:** The scripting language that you'll use to write the programs that provide the dynamic functionality for your Web site

I describe these three tools in detail in Chapter 1.

# Building a Web Site

As discussed in the previous section, a Web site is a collection of text files placed on a computer in a location where users can access them. Placing the Web site files where they can be accessed by the public is called *publishing* the Web site. However, this is the final step of building the Web site, not the first step. You don't want to publish the Web site until it's finished — a perfect Web site ready for public viewing.

To prevent the public from seeing your half-finished Web site, warts and all, you need to develop your Web site in a location that isn't available to the public. While developing your Web site, you'll be testing things and troubleshooting problems. You need to do this work in private.

Because you need to build your Web site in private and hold off on making it public until it's finished and perfect, your work environment needs two sites:

**Your Web site:** The site where your published Web site is located. The location where the public views your Web site.

**Your development site:** The location where you develop your Web pages. When your pages are complete, you then move them to your Web site.

Your Web site publishes your Web pages to the world. Your development site shouldn't be available for the world to see your errors and half-done Web pages. Your development site needs to be hidden from the world. Never publish your Web pages until they are complete and perfect.

You need to decide where you're going to publish your Web site and where you're going to develop it. The information you need to make these decisions is provided in the next few sections of this chapter.

# Deciding Where to Publish Your Web Site

One of your first decisions is where to publish your Web site. You need to publish it on a computer that's connected to the World Wide Web. The computer should also provide the tools you need, as discussed earlier: a Web server, PHP, and MySQL. The most common locations for publishing your Web site are

**A Web site hosted by a Web hosting company:** The Web site is located on the Web hosting company's computer. The Web hosting company installs and maintains the Web site software and provides space on its computer where you can install the files for the Web site.

**A Web site put up by a company on its own computer:** The company — usually the company's IT (Information Technology) department —

installs and administers the Web site software. Your job, for the purposes of this book, is to program the Web site, either as an employee of the company or as a contractor.

**A Web site that you set up yourself:** You plan to install and maintain the Web site software yourself. It could be a Web site of your own that you're building on your own computer, or it might be a Web site that you're installing for a client on the client's computer.

You'll most likely publish your Web site on one of the first two options. For these options, you don't need to know much about the administration and operation of the Web site software. The Web server, PHP, and MySQL are already installed, and the information you need to access them is provided by the company responsible for the Web site.

The third option requires that you install, set up, administer, and maintain the Web site software yourself. This option requires much more technical knowledge of computer software than the first two options, where others provide the software for you. However, the advantage of this option is that you have more control. You can set up the Web site software with the settings that you prefer.

In the next three sections, I describe the publishing options in more detail and provide the information you need to decide where to publish your Web site.

## Using a Web hosting company

A *Web hosting company* provides everything that you need to put up a Web site, including the computer space and all the Web site software. You just create the files for your Web pages and move them to a location specified by the Web hosting company. Most small-to-medium-sized Web sites are hosted by Web hosting companies.

About a gazillion companies offer Web hosting services. Most charge a monthly fee (often quite small), and some are even free. (Most, but not all, of the free ones require you to display advertising.) Usually, the monthly fee varies depending on the resources provided for your Web site. For instance, a

Web site with 2MB of disk space for your Web page files costs less than a Web site with 10MB of disk space.

When looking for a place to host your Web site, make sure that the Web hosting company offers the following:

**PHP and MySQL:** Not all companies provide these tools. You might have to pay more for a site with access to PHP and MySQL; sometimes you have to pay an additional fee for MySQL databases.

**A recent version of PHP:** Sometimes the PHP versions offered aren't the most recent versions. As of this writing, PHP 6 is close to being released.

Until PHP 6 is released, two versions of PHP are generally available — PHP 4 and PHP 5. Even though PHP 5 has been out for several years, many Web sites still run PHP 4. PHP 4 is still supported because existing PHP 4 code does not always run perfectly under PHP 5. Many developers have not yet converted their code to run under PHP 5. However, the demise of PHP 4 is looming. Support for PHP 4 stopped at the end of 2007. There will be no more releases of PHP 4, and critical security fixes ended in late 2008. There is no reason for anyone developing new code to use PHP 4.



Look for a Web hosting company that provides PHP 5. Some Web hosts provide both PHP 4 and PHP 5, but they use PHP 4 as the default. You may need to talk to technical support at the Web hosting company to find out how to get PHP 5 on your Web site, rather than PHP 4.

**A recent version of MySQL:** The current preferred version of MySQL is MySQL 5.1. However, using an older version of MySQL is not as much of a problem as using older versions of PHP. The techniques in this book work with older versions of MySQL. In the future, you may learn more advanced MySQL features and may need a newer version of MySQL. However, even older versions provide a feature set that allows quite sophisticated dynamic Web sites.

**Ability to change PHP settings:** Changing PHP settings can affect some of PHP's behavior. Web hosts vary in the amount of access to PHP settings that you, as their customer, are given. More access to PHP settings gives you more control over your Web site functionality.

A text file named `php.ini` contains the PHP settings. Your Web host will not give you access to the general `php.ini` file for the host's system, but some hosts allow you to use a local `php.ini` file that affects only your Web site. This is a useful feature to look for because it's an easy way to change the settings.

Another way to change PHP settings is using an `.htaccess` file. This is a file that the Apache Web server reads that can contain some PHP settings. Many Web hosts allow you to store an `.htaccess` file on your Web site, which changes settings for your Web site only.

When you select a Web host, be sure the hosting company allows you to use either a local `php.ini` file or an `.htaccess` file. It's important that you be able to change the PHP settings for your Web site.

**PhpMyAdmin:** To create and use MySQL databases, you need specific software. Any Web host that provides MySQL needs to provide software to communicate with MySQL databases. Most Web hosts provide phpMyAdmin, a Web application written in PHP and designed specifically for managing MySQL databases. Other software also works, but this book assumes you have access to phpMyAdmin.

Other considerations when choosing a Web hosting company are

**Reliability:** You need a Web hosting company that you can depend on — one that won't go broke and disappear tomorrow, and one that isn't running on old computers, held together by chewing gum and baling wire, with more downtime than uptime.

**Speed:** Web pages that download slowly are a problem because users will get impatient and go elsewhere. Slow pages could be a result of a Web hosting company that started its business on a shoestring and has a shortage of good equipment — or the Web hosting company might be so successful that its equipment is overwhelmed by new customers. Either way, Web hosting companies that deliver Web pages too slowly

are unacceptable.

**Technical support:** Some Web hosting companies have no one available to answer questions or troubleshoot problems. Technical support is often provided only through e-mail, which can be very good if the response time is short. Sometimes you can test the quality of the company's support by calling the tech support number, or test the e-mail response time by sending an e-mail.

**The domain name:** Each Web site has a domain name that Web browsers use to find the site on the Web. Each domain name is registered for a small yearly fee so that only one Web site can use it. Some Web hosting companies allow you to use a domain name that you have registered independently of the Web hosting company, some assist you in registering and using a new domain name, and some require that you use their domain name. For instance, suppose that your name is Lola Designer and you want your Web site to be named LolaDesigner. Some Web hosting companies allow your Web site to be [LolaDesigner.com](http://LolaDesigner.com), but some require that your Web site be named LolaDesigner.webhostingcompanyname.com, or webhostingcompanyname.com/~LolaDesigner, or something similar. In general, your Web site looks more professional if you use your own domain name.

**Backups:** *Backups* are copies of your Web page files and your database that are stored in case your files or database are lost or damaged. You want to be sure that the company makes regular, frequent backup copies of your application. You also want to know how long it would take for backups to be put in place to restore your Web site to working order after a problem.

**Features:** Select features based on the purpose of your Web site. Usually a hosting company bundles features together into plans — more features equal a higher cost. Some features to consider are

- *Disk space:* How many MB or GB of disk space will your Web site require? Media files, such as graphics or music files, can be quite large.
- *Data transfer:* Some hosting companies charge you for sending

Each domain name must be unique in order to serve as an address. Consequently, a system of registering domain names ensures that no two locations use the same domain name. Anyone can register any domain name as long as the name isn't already taken. You can register a domain name on the Web. First, you test your potential domain name to find out whether it's available. If it's available, you register it in your name or a company name and pay the fee. The name is then yours to use, and no one else can use it. The standard fee for domain name registration is \$35 per year. You should never pay more, but bargains are often available.

Many Web sites provide the ability to register a domain name, including many Web hosting companies. A search at Google ([www.google.com](http://www.google.com)) for *register domain name* results in more than 85 million hits. Shop around to be sure that you find the lowest price. Also, many Web sites allow you to enter a domain name and see whom it is registered to. These Web sites do a domain name database search using a tool called *whois*. A search at Google for *domain name whois* results in more than 17 million hits. A couple of places where you can do a whois search are Allwhois.com ([www.allwhois.com](http://www.allwhois.com)) and BetterWhois.com ([www.betterwhois.com](http://www.betterwhois.com)).

Researching Web hosting companies from a standing start is pretty difficult — a search at Google.com for “*Web hosting*” results in almost 400 million hits. The best way to research Web hosting companies is to ask for recommendations from people who have experience with those companies. People who have used a hosting company can warn you if the service is slow or the computers are down often. After you gather a few names of Web hosting companies from satisfied customers, you can narrow the list to the one that's best suited to your purposes and the most cost effective.

The following is a list of Web hosts that offer the tools needed, including PHP 5, MySQL, phpMyAdmin, .htaccess files, and good technical support:

Host Gator, [www.hostgator.com](http://www.hostgator.com)

HostMonster, [www.hostmonster.com](http://www.hostmonster.com)

WebHostingBuzz (WHB), [www.webhostingbuzz.com](http://www.webhostingbuzz.com)  
midPhase, [www.midphase.com](http://www.midphase.com)  
BlueHost, [www.bluehost.com](http://www.bluehost.com)

Please bear in mind that this list is based solely on the Web site of the hosting company and reviews found on Web sites that review Web hosts — not on personal experience.

## Using a company Web site

When the Web site is run by the company, you don't need to understand the installation and administration of the Web site software at all. The company is responsible for the operation of the Web site, so that burden is off your shoulders. In most cases, the Web site already exists, and your job is to add to, modify, or redesign the existing Web site. In a few cases, the company might be installing its first Web site, and your job is to design the Web site. In either case, your responsibility is to write and install the HTML files for the Web site.

You access the Web site software through the company's IT department. The name of this department can vary in different companies, but its function is the same: It keeps the company's computers running and up-to-date.

If PHP or MySQL or both aren't available on the company's Web site, IT needs to install them and make them available to you. PHP and MySQL have many options, but IT might not understand the best options — and might have options set in ways that aren't well suited for your purposes. If you need PHP or MySQL options changed, you need to request that IT make the change; you won't be able to make the change yourself. For instance, PHP must be installed with MySQL support enabled, so if PHP isn't communicating correctly with MySQL, IT might have to reinstall PHP with MySQL support enabled.



You will interact with the IT folks frequently as needs arise. For example, you might need options changed, you might need information to help you interpret an error message, or you might need to report a problem with the Web site software. So a good relationship with the IT folks will make your life much easier. Bring them tasty cookies and doughnuts often.

## Setting up your own server

If you're setting up your own Web server from scratch, to publish your own Web site, you need to understand the Web site software fairly well. You have to make several decisions regarding hardware and software. You have to install a Web server, PHP, and MySQL — as well as maintain, administer, and update the system yourself. Taking this route, rather than using a Web site provided by others, requires more work and more knowledge. Don't attempt this unless you are pretty knowledgeable about the Web, legal restrictions, security concerns, and other relevant issues. The advantage is that you have total control over the Web development environment.

Here are the general steps that lead to your dynamic Web site:

### **1. Set up the computer.**

While you can set up your existing computer that you use for all your other computer work to be the Web server that delivers a Web site, it's not wise. If your Web site receives much traffic, it may not have enough resources and may bog down. It's much better to set up a new machine to be your Web server.

### **2. Install the Web server.**

After you set up the computer, you need to install Web server software. In most cases, you want to install Apache. It's free, popular, reliable, secure, and runs on most operating systems. Apache is automatically installed with the operating system on Macs and Linux. Currently, Apache powers about 60 percent of Web sites. You can find information

about installing Apache at <http://httpd.apache.org>.

### **3. Install MySQL.**

To run your Web database application, you need to install MySQL. Many Mac and Linux computers arrive with MySQL already installed, although they still may need to be upgraded to the most recent version. You can download and install MySQL from [www.mysql.com](http://www.mysql.com).

### **4. Install PHP.**

After you install MySQL and Apache, you're ready to install PHP. Some versions of Mac and Linux arrive with PHP already installed. You can find software to download and install, as well as thorough documentation, at [www.php.net](http://www.php.net).

## **Deciding Where to Develop Your Web Site**

As discussed previously, you need to develop your Web site in a different location from where you publish your Web site. You need a location where you can write the Web page files free from public view. You don't want your experiments and error-filled first-tries to be public.

The most common place to develop your Web site is on your local computer. If you can't develop on your computer, you have to have a private area of someone else's computer, such as your Web host's computer, where you can develop your Web site.

### **On your own computer**

You can develop your Web site on your local computer and upload the files to your Web site when your Web site is finished and ready for the world to see. In most cases, this work process is the best solution.