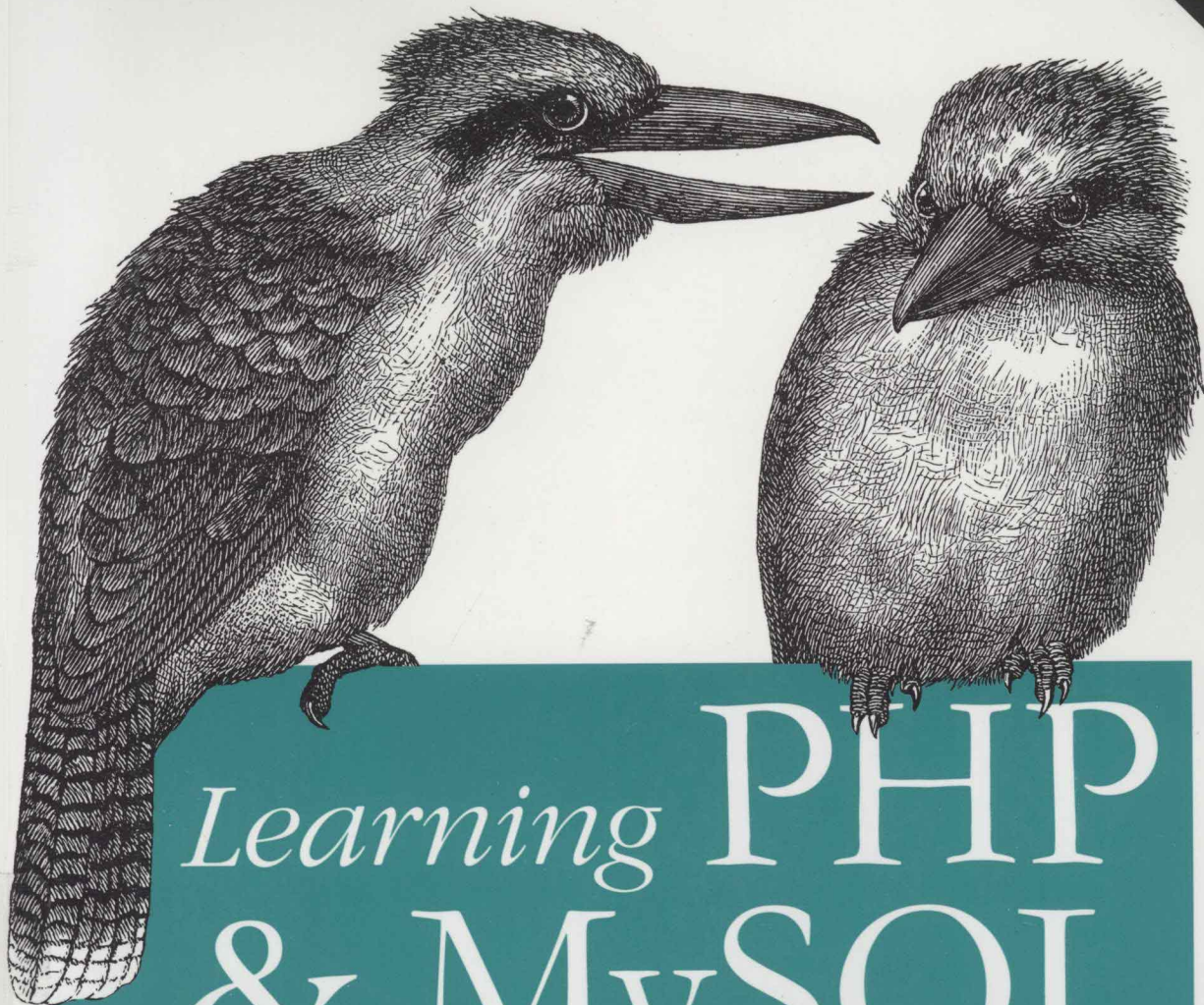


学习 PHP & MySQL (影印版)

第二版



Learning PHP
& MySQL

O'REILLY®

东南大学出版社

Michele E. Davis & Jon A. Phillips 著

第二版

学习 PHP & MySQL (影印版)

Learning PHP & MySQL



O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目 (CIP) 数据

学习 PHP & MySQL: 第 2 版: 英文 / (德) 戴维斯 (Davis, M.E.), (德) 菲利普斯 (Phillips, J.A.) 著. — 影印本. — 南京: 东南大学出版社, 2008.3

书名原文: Learning PHP & MySQL, Second Edition

ISBN 978-7-5641-1142-7

I . 学… II . ①戴… ②菲… III . ① PHP 语言—程序设计—英文 ②关系数据库—数据库管理系统, MySQL—英文 IV . TP312 TP311.138

中国版本图书馆 CIP 数据核字 (2008) 第 024163 号

江苏省版权局著作权合同登记

图字: 10-2007-224 号

©2007 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2008. Authorized reprint of the original English edition, 2007 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2007。

英文影印版由东南大学出版社出版 2008。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式复制。

学习 PHP & MySQL 第二版

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出版人: 江汉

网 址: <http://press.seu.edu.cn>

电子邮件: press@seu.edu.cn

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 27 印张

字 数: 454 千字

版 次: 2008 年 3 月第 1 版

印 次: 2008 年 3 月第 1 次印刷

书 号: ISBN 978-7-5641-1142-7/TP · 186

印 数: 1~2500 册

定 价: 66.00 元 (册)

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

O'Reilly Media, Inc. 介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权东南大学出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc. 一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以 O'Reilly Media, Inc. 知道市场上真正需要什么图书。

出版说明

随着计算机技术的成熟和广泛应用,人类正在步入一个技术迅猛发展的新时期。计算机技术的发展给人们的工业生产、商业活动和日常生活都带来了巨大的影响。然而,计算机领域的技术更新速度之快也是众所周知的,为了帮助国内技术人员在第一时间了解国外最新的技术,东南大学出版社和美国 O'Reilly Meida, Inc.达成协议,将陆续引进该公司的代表前沿技术或者在某专项领域享有盛名的著作,以影印版或者简体中文版的形式呈献给读者。其中,影印版书籍力求与国外图书“同步”出版,并且“原汁原味”展现给读者。

我们真诚地希望,所引进的书籍能对国内相关行业的技术人员、科研机构的研究人员和高校师生的学习和工作有所帮助,对国内计算机技术的发展有所促进。也衷心期望读者提出宝贵的意见和建议。

最新出版的影印版图书,包括:

- 深入浅出 SQL (影印版)
- JavaScript & DHTML Cookbook 第二版 (影印版)
- 集体智慧编程 (影印版)
- SOA 实践 (影印版)
- 学习 PHP & MySQL 第二版 (影印版)
- Linux 系统编程 (影印版)
- Beautiful Code (影印版)
- Mac OS X: The Missing Manual, Leopard Editon (影印版)
- 高性能网站 (影印版)
- WPF 编程 第二版 (影印版)
- 敏捷开发艺术 (影印版)
- 学习 Python 第三版 (影印版)
- Ruby 程序设计语言 (影印版)

Preface

PHP and MySQL are a powerful combination that makes it easy to create web applications. If you've been creating web pages but want to build more sophisticated sites that can grow and interact with users, PHP and MySQL let you get started easily and then build complex applications on those foundations.

Our goal is to help you learn the ins and outs of PHP and MySQL and to save you some of the "Why doesn't that work?" moments that we've already been through. We'll show you what to watch for and how to fix these issues without pulling out your hair.

Audience

This book is for people who want to know how to create dynamic web sites. That could include graphic designers who are already working in an IT or advertising firm creating static web sites, and who may need to move forward with coding database-driven web sites. It might also include people who already know, say, Flash development and HTML markup, but need to expand their repertoire of skills to databases and programming.

Assumptions This Book Makes

This book assumes you understand how web browsers work and have a basic understanding of HTML. Some understanding of JavaScript may be useful (for Chapter 16) but isn't generally required.

You might also be overqualified. If you already know how to create pages using MySQL and PHP, then you'd probably be better off with a book that is more a reference than a learning book, such as Paul Hudson's *PHP in a Nutshell*, or Russell Dyer's *MySQL in a Nutshell*, both from O'Reilly.

Organization of This Book

This book starts out with an overview of how all of the pieces you'll be working with fit together. Because there are multiple languages and technologies that interact to form dynamic web pages, it's best to start with a solid understanding of how the pieces work together. The PHP that you'll learn works as an integration package for dynamic web sites.

Next, we'll walk through installing the core software packages on your local computer. This book focuses on PHP and MySQL, but making this work also usually requires the Apache web server. The PHP interpreter works with the web server when processing dynamic content. Finally, you'll install the MySQL database. Installation is covered for PC, Mac, and Linux systems. You can also use a hosted Internet service provider (ISP) account to develop your pages, if you don't want to install everything locally.

Since PHP plays an important role in pulling everything together, we next explain the basics of working with the PHP language. This includes language essentials such as data types, program flow logic, and variables. Functions, arrays, and forms each get their own chapter to fully explore them.

Because you may be new to databases in general, we ease into MySQL by first explaining concepts that apply to designing and using any relational database. Then we give specific examples of using MySQL to interact with your data. Once you can get data in and out of the database, you'll need to work with PHP to integrate that data into your dynamic content.

Security and access control get their own chapters. While security may sound like a dull subject, it's still a huge issue if you store any private information on your web page. We'll guide you around several common security pitfalls.

We also touch on how XHTML, the next generation of HTML, works with PHP and your web sites.

Finally, we close with sample applications that demonstrate how the technologies work together to rapidly build workable, fast web sites. You'll also be provided with web sites and forums to gain additional information on the topics covered in the book.

Supporting Books

Even if you feel you are ready for this book, you may want to explore some of the technologies in greater depth than is possible here. The following list offers some good places to start:

- *Run Your Own Web Server Using Linux & Apache*, by Tony Steidler-Dennison (SitePoint).
- *PHP in a Nutshell*, First Edition, by Paul Hudson (O'Reilly).

- *MySQL in a Nutshell*, First Edition, by Russell Dyer (O'Reilly).
- *CSS Cookbook*, Second Edition, by Christopher Schmitt (O'Reilly).

There are also several good online resources for dynamic web development, including <http://onlamp.com>, part of the O'Reilly Network. LAMP stands for Linux, Apache, MySQL, PHP. LAMP is the de facto standard for serving dynamic web pages.

Conventions Used in This Book

The following font conventions are used in this book:

Italic

Indicates pathnames, filenames, and program names; Internet addresses, such as domain names and URLs; and new items where they are defined.

Constant width

Indicates command lines; names and keywords in programs, including method names, variable names, and class names; HTML element tags; values; and database engines.

Constant width italic

Indicates text that should be replaced with user-supplied values.

Constant width bold

Indicates emphasis in program code lines and user input options that should be typed verbatim.



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



This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you can use the code in this book in your programs and documentation. You do not need to contact O'Reilly 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 and MySQL*, Second Edition, by Michele E. Davis and Jon A. Phillips. Copyright 2007 Michele E. Davis and Jon A. Phillips, 978-0-596-51401-3.”

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.

How to Contact Us

We have tested and verified the information in this book to the best of our ability, but mistakes and oversights do occur. Please let us know about any errors you find, as well as your suggestions for future editions, by writing to:

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)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://www.oreilly.com/catalog/9780596514013>

There is also a blog for this book located at:

<http://www.krautgrrl.com/learningphp/>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O’Reilly Network, see our web site at:

<http://www.oreilly.com>

Safari® Books Online



When you see a Safari® Books Online icon on the cover of your favorite technology book, that means the book is available online through the O’Reilly Network Safari Bookshelf.

Safari offers a solution that’s better than e-books. It’s a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at <http://safari.oreilly.com>.

Acknowledgments

We are happy to have this newly improved and expanded Second Edition out for our audience. We'd like to thank our wonderful agent, Matt Wagner of Fresh Books, along with Simon St. Laurent at O'Reilly for getting this Second Edition rolling; without them, this book wouldn't be in your hands.

Second, profuse thanks to our technical editors, especially Jereme Allen, Charlie Maguire, and Peter MacIntyre for their fantastic edits to our book. We'd also like to thank our local Minneapolis/St. Paul PHP community: <http://www.tcphp.org>, which sparked our interest in PHP and MySQL years ago. Lastly, thanks to Simon, Mimi, and Zack for being patient while their parents reworked a very important book.

Table of Contents

Preface	xi
1. Dynamic Content and the Web	1
HTTP and the Internet	1
PHP and MySQL's Place in Web Development	2
The Components of a PHP Application	4
Integrating Many Sources of Information	7
Requesting Data from a Web Page	11
2. Installation	15
Developing Locally	15
Working Remotely	35
3. Exploring PHP	39
PHP and HTML Text	39
Coding Building Blocks	43
4. PHP Decision-Making	62
Expressions	62
Operator Concepts	64
Conditionals	71
Looping	77
5. Functions	85
Calling Functions	87
Defining Functions	89
Object-Oriented Programming	96

6. Arrays	107
Array Fundamentals	107
7. Working with MySQL	122
MySQL Database	122
Managing the Database	125
Using phpMyAdmin	126
Database Concepts	131
Structured Query Language	132
8. Database Best Practices	146
Database Design	146
Backing Up and Restoring Data	155
Advanced SQL	159
9. Getting PHP to Talk to MySQL	179
The Process	180
Querying the Database with PHP Functions	180
Using PEAR	190
10. Working with Forms	199
Building a Form	199
Templates	218
11. Practical PHP	223
String Functions	223
Date and Time Functions	233
File Manipulation	238
Calling System Calls	249
12. XHTML	251
Why XHTML?	253
XHTML and XML Namespaces	254
XHTML Versions	254
Generating XHTML with PHP	261
13. Modifying MySQL Objects and PHP Data	263
Changing Database Objects from PHP	263
Manipulating Table Data	266
Displaying Results with Embedded Links	267

Presenting a Form to Add and Process in One File	270
Updating Data	276
Deleting Data	277
Performing a Subquery	282
14. Cookies, Sessions, and Access Control	285
Cookies	285
PHP and HTTP Authentication	288
Sessions	294
Using Auth_HTTP to Authenticate	301
15. Security	307
Session Security	316
16. Validation and Error Handling	325
Validating User Input with JavaScript	325
Pattern Matching	329
Redisplaying a Form After PHP Validation Fails	333
17. Sample Application	339
Configuration File	340
Page Framework	340
Database	343
Displaying a Postings Summary	346
Displaying a Posting and Its Comments	349
Adding and Changing Posts	352
Adding and Changing Comments	358
18. Finishing Your Journey	366
PHP Coding Standards	366
PEAR	371
Frameworks	372
Ajax	373
Wikis	373
Finding Help on the Web	373
Appendix. Solutions to Chapter Questions	377
Index	391

Dynamic Content and the Web

To the average user, a web page is a web page. It opens in the browser and provides information. Looking closer, though, some pages stay mostly the same, while other pages change regularly. Pages that don't change—*static* pages—are relatively simple to create. Someone has to create an HTML document, by hand or with tools, and upload it to a site where web browsers can visit. One of the most common tools to create HTML documents is Adobe Dreamweaver. When changes are needed, you just replace the old file with a new one. *Dynamic* pages are also built with HTML, but instead of a simple build-and-post approach, the pages are updated regularly, sometimes every time that they are requested.

Static sites provide hyperlinked text and perhaps a login screen, but beyond that, they don't offer much interaction. By contrast, Amazon.com (<http://www.amazon.com>) demonstrates much of what a dynamic web site can do: your ordering data is logged, and Amazon offers recommendations based on your purchasing history when you access their page. In other words, dynamic means that the user interacts with the web site beyond just reading pages, and the web site responds accordingly. Every page is a personalized experience.

Creating dynamic web pages—even a few years ago—meant writing a lot of code in the C or Perl languages, and then calling and executing those programs through a process called a Common Gateway Interface (CGI). Having to create executable files wasn't much fun, and neither was learning a whole new complicated language. Thankfully, PHP and MySQL make creating dynamic web sites easier and faster.

HTTP and the Internet

Some basic understanding of how the Internet works may be useful if you haven't programmed for the Web before. The HyperText Transfer Protocol (HTTP) defines how web pages are transferred across the Internet. HTTP is the method used to transfer or convey information on the World Wide Web. Its original purpose was to provide a way to publish and retrieve HTML pages.

The World Wide Web Consortium (W3C) and the Internet Engineering Task Force coordinated the development of HTTP, which is a request-and-response protocol that connects clients and servers. The originating client, usually a web browser, is referred to as the *user agent*. The destination server, which stores or creates resources and can contain HTML files and images, is called the *origin server*. Between the user agent and origin server, there may be several intermediaries, such as proxies.

An HTTP client initiates a request by establishing a Transmission Control Protocol (TCP) connection to a particular port on a remote host (port 80 is the default). An HTTP server listening on that port waits for the client to send a request message. Upon receiving the request, the server sends back a status line, like “HTTP/1.1 200 OK,” and its own response. Depending on the status, this response could be the requested file, an error message, or some other information.

HTTP is built on top of TCP, which is itself layered on top of Internet Protocol (IP). The two are often referred to together as TCP/IP. Applications on networked hosts can use TCP to create connections to one another, and then exchange streams of data. The protocol guarantees reliable delivery of data from sender to receiver. TCP supports many of the Internet’s most popular application protocols and applications, including the Web, email, and Secure Shell (SSH).

PHP and MySQL’s Place in Web Development

PHP is a programming language designed to generate web pages interactively on the computer serving them, which is called a *web server*. Unlike HTML, where the web browser uses tags and markup to generate a page, PHP code runs between the requested page and the web server, adding to and changing the basic HTML output.

PHP makes web development easy because all the code you need is contained within the PHP framework. This means that there’s no reason for you to reinvent the wheel each time you sit down to develop a PHP program; it comes with web functionality built-in.

While PHP is great for web application development, it doesn’t store information by itself. For that, you need a database. The database of choice for PHP developers is MySQL, which acts like a filing clerk for PHP-processed user information. MySQL automates the most common tasks related to storing and retrieving specific user information based on your supplied criteria.



Consider the Amazon.com example: the recommendations Amazon offers are based on a database that records your prior order information.

MySQL is easily accessed from PHP, and they work well together. An added benefit is that PHP and MySQL run on various computer types and operating systems, including Mac OS X, Windows-based PCs, and Linux.

Advantages of Using PHP with MySQL

There are several factors that make using PHP and MySQL together a natural choice:

PHP and MySQL work well together

PHP and MySQL have been developed with each other in mind, so they are easy to use together. The programming interfaces between them are logically paired up. Working together wasn't an afterthought when the developers created the PHP and MySQL interfaces.

PHP and MySQL have open source power

As they are both open source projects, PHP and MySQL can both be used for free. MySQL client libraries are no longer bundled with PHP. Advanced users have the ability to make changes to the source code, and therefore change the way the language and programs work.

PHP and MySQL have community support

Both tools active communities on the Web in which you can participate, and the participants will help you answer your questions. You can also purchase professional support for MySQL if you need it.

PHP and MySQL are fast

Their simple and efficient designs enable faster processing.

PHP and MySQL don't bog you down with unnecessary details

You don't need to know all of the low-level details of how the PHP language interfaces with the MySQL database, as there is a standard interface for calling MySQL procedures from PHP. Online application programming interfaces (APIs) at <http://www.php.net> offer unlimited resources.

The Value of Open Source

As we mentioned above, both PHP and MySQL are open source projects, so you don't need to worry about buying user licenses for every computer in your office or home. When using open source projects and technologies, programmers have access to the source code. This enables individual or group analysis to identify potentially problematic code, test, debug, and offer changes as well as additions to that code. For example, Unix—the forerunner in the open source software community—was freely shared with university software researchers. Linux, the free alternative to Unix, is a direct result of their efforts and the open source-licensing paradigm. Most open source licenses include the right to distribute modified code with some restrictions. For example, some licenses require that derivative code must also be released under the same license, or there may be a restriction that others can't use your code.

As Tim O'Reilly puts it, “Open source licensing began as an attempt to preserve a culture of sharing, and only later led to an expanded awareness of the value of that sharing.” Today, open source programmers share their code changes on the Web via <http://www.php.net>, listservs, and web sites. If you're caught in a coding nightmare and can't wake up, the resources mentioned previously can and will help you.

We'll arm you with open source user forums later in this book so you can check them out yourself. We'll include listservs and web sites so that you have numerous resources if you run into a snafu.

The Components of a PHP Application

In order to process and develop dynamic web pages, you'll need to use and understand several technologies. There are three main components of creating dynamic web pages: a web server, a server-side programming language, and a database. It's a good idea to have an understanding of these three basic components for web development using PHP. We'll start with some rudimentary understanding of the history and purpose of Apache (your web server), PHP (your server-side programming language), and MySQL (your database). This can help you to understand how they fit into the web development picture.

Remember that dynamic web pages pull information from several sources simultaneously, including Apache, PHP, MySQL, and Cascading Style Sheets (CSS), which we'll talk about later.

PHP

PHP grew out of a need for people to develop and maintain web sites containing dynamic client-server functionality. In 1994, Rasmus Lerdorf created a collection of open source Perl scripts for his personal use, and these eventually were rewritten in C and turned into what PHP is today. By 1998, PHP was released in its third version, turning it into a web development tool that could compete with similar products such as Microsoft's Active Server Pages (ASP) and Sun's Java Server Pages (JSP). PHP also is an interpreted language, rather than a compiled one. The real beauty of PHP is simplicity coupled with power.



Compiled languages create a binary file such as an `.exe`, while interpreted languages work directly with the source code when executing, as opposed to creating a standalone file.

PHP is ubiquitous and compatible with all major operating systems. It is also easy to learn, making it an ideal tool for web programming beginners. Additionally, you get to take advantage of a community's effort to make web development easier for everyone. The creators of PHP developed an infrastructure that allows experienced C programmers to extend PHP's abilities. As a result, PHP now integrates with advanced technologies like XML, XSL, and Microsoft's Component Object Model Technologies (COM).