

LEARNING MADE EASY



2nd Edition

PHP, MySQL[®], & JavaScript[®]

ALL-IN-ONE

for
dummies[®]
A Wiley Brand



7
Books
in one!

Richard Blum

Web programming and
design teacher



PHP, MySQL[®], & JavaScript[®]

ALL-IN-ONE

2nd Edition

by **Richard Blum**

for
dummies[®]
A Wiley Brand

PHP, MySQL®, & JavaScript® All-in-One For Dummies®, 2nd Edition

Published by: **John Wiley & Sons, Inc.**, 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Copyright © 2025 by John Wiley & Sons, Inc. All rights reserved, including rights for text and data mining and training of artificial technologies or similar technologies.

Media and software compilation copyright © 2025 by John Wiley & Sons, Inc. All rights reserved, including rights for text and data mining and training of artificial technologies or similar technologies.

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 the prior written permission of the Publisher. 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>.

The manufacturer's authorized representative according to the EU General Product Safety Regulation is Wiley-VCH GmbH, Boschstr. 12, 69469 Weinheim, Germany, e-mail: Product_Safety@wiley.com.

Trademarks: Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and may not be used without written permission. MySQL is a trademark or registered trademark of MySQL AB. JavaScript is a trademark or registered trademark of Oracle America, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, 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 <https://hub.wiley.com/community/support/dummies>.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2025938536

ISBN 978-1-394-34209-9 (pbk); ISBN 978-1-394-34211-2 (ebk); ISBN 978-1-394-34210-5 (ebk)

Contents at a Glance

Introduction	1
Book 1: Getting Started with Web Programming	5
CHAPTER 1: Examining the Pieces of Web Programming	7
CHAPTER 2: Using a Web Server	29
Book 2: HTML5 and CSS3	57
CHAPTER 1: The Basics of HTML5	59
CHAPTER 2: The Basics of CSS3	89
CHAPTER 3: HTML5 Forms	121
CHAPTER 4: Advanced CSS3	143
CHAPTER 5: HTML5 and Multimedia	163
Book 3: JavaScript	183
CHAPTER 1: Introducing JavaScript	185
CHAPTER 2: Advanced JavaScript Coding	213
CHAPTER 3: Using jQuery	233
CHAPTER 4: Reacting to Events with JavaScript and jQuery	253
CHAPTER 5: Troubleshooting JavaScript Programs	273
Book 4: PHP	291
CHAPTER 1: Understanding PHP Basics	293
CHAPTER 2: PHP Flow Control	315
CHAPTER 3: PHP Libraries	339
CHAPTER 4: Considering PHP Security	365
CHAPTER 5: Object-Oriented PHP Programming	385
CHAPTER 6: Sessions and Carts	411
Book 5: MySQL	435
CHAPTER 1: Introducing MySQL	437
CHAPTER 2: Administering MySQL	457
CHAPTER 3: Designing and Building a Database	483
CHAPTER 4: Using the Database	507
CHAPTER 5: Communicating with the Database from PHP Scripts	537

Book 6: Creating Object-Oriented Programs	557
CHAPTER 1: Designing an Object-Oriented Application.	559
CHAPTER 2: Implementing an Object-Oriented Application.	589
CHAPTER 3: Using AJAX	615
CHAPTER 4: Extending WordPress	647
Book 7: Using PHP Frameworks	673
CHAPTER 1: The MVC Method	675
CHAPTER 2: Selecting a Framework	689
CHAPTER 3: Creating an Application Using Frameworks.	707
Index	729

Table of Contents

INTRODUCTION	1
About This Book	1
Foolish Assumptions	2
Icons Used in This Book	3
Beyond the Book	3
Where to Go from Here	4
BOOK 1: GETTING STARTED WITH WEB PROGRAMMING	5
CHAPTER 1: Examining the Pieces of Web Programming	7
Creating a Simple Web Page	7
Kicking things off with the World Wide Web	8
Making sense of markup languages	9
Retrieving HTML documents	11
Styling your web content	14
Creating a Dynamic Web Page	18
Client-side programming	19
Server-side programming	21
Combining client-side and server-side programming	25
Storing Content	26
CHAPTER 2: Using a Web Server	29
Recognizing What's Required	29
The web server	30
The PHP server	31
The database server	32
Considering Your Server Options	33
Using a web-hosting company	34
Building your own server environment	35
Using premade servers	41
Tweaking the Servers	45
Customizing the Apache server	45
Customizing the MySQL server	48
Customizing the PHP server	51
Building a Development Environment	53
Program editors	53
Integrated development environments	54

BOOK 2: HTML5 AND CSS3	57
CHAPTER 1: The Basics of HTML5	59
Diving into Document Structure	59
Elements, tags, and attributes	59
Document type	61
Page definition	62
Page sections	64
Looking at the Basic HTML5 Elements	67
Headings	67
Text groupings	68
Breaks	70
Marking Your Text	72
Formatting text	72
Using hypertext	73
Working with Characters	77
Character sets	77
Special characters	78
Making a List (And Checking It Twice)	78
Unordered lists	79
Ordered lists	79
Description lists	82
Building Tables	83
Defining a table	83
Defining the table's rows and columns	84
Defining the table headings	86
CHAPTER 2: The Basics of CSS3	89
Understanding Styles	89
Defining the rules of CSS3	90
Applying style rules	96
Cascading style rules	97
Styling Text	98
Setting the font	98
Playing with color	102
Working with the Box Model	105
Styling Tables	107
Table borders	107
Table data	109
Positioning Elements	111
Putting elements in a specific place	113
Floating elements	116

CHAPTER 3: HTML5 Forms	121
Understanding HTML5 Forms	121
Defining a form	122
Working with form fields	123
Using Input Fields	124
Text boxes	124
Password entry	126
Check boxes	127
Radio buttons	128
Hidden fields	129
File upload	130
Buttons	131
Adding a Text Area	132
Using Drop-Down Lists	133
Enhancing HTML5 Forms	135
Data lists	135
Additional input fields	136
Using HTML5 Data Validation	139
Holding your place	140
Making certain data required	140
Validating data types	141
CHAPTER 4: Advanced CSS3	143
Rounding Your Corners	143
Using Border Images	145
Looking at the CSS3 Colors	148
Playing with Color Gradients	150
Linear gradients	150
Radial gradients	151
Adding Shadows	152
Text shadows	152
Box shadows	153
Creating Fonts	154
Focusing on font files	155
Working with web fonts	155
Handling Media Queries	157
Using the @media command	157
Dealing with CSS3 media queries	158
Applying multiple style sheets	161
CHAPTER 5: HTML5 and Multimedia	163
Working with Images	163
Placing images	164
Styling images	165
Linking images	168

Working with image maps	168
Using HTML5 image additions	170
Playing Audio	171
Embedded audio	172
Digital audio formats	173
Audio the HTML5 way	175
Watching Videos	177
Paying attention to video quality	177
Looking at digital video formats	178
Putting videos on your web page	179
Getting Help from Streamers	181
BOOK 3: JAVASCRIPT	183
CHAPTER 1: Introducing JavaScript	185
Learning about the Power of JavaScript	185
Changing web page content	186
Changing web page styles	186
Including JavaScript in Your HTML Code	187
Embedding JavaScript	187
Using external JavaScript files	191
The Basics of JavaScript	192
Working with data	192
Data types	193
Arrays of data	194
Operators	196
Controlling Program Flow	198
Conditional statements	198
Loops	204
Working with Functions	210
Creating a function	210
Using a function	211
CHAPTER 2: Advanced JavaScript Coding	213
The Document Object Model	213
The DOM tree	214
JavaScript and the DOM	215
Finding Your Elements	223
Getting to the point	224
Walking the tree	226
Working with DOM Form Data	228
Text boxes	228
Text areas	230
Check boxes	230
Radio buttons	231

CHAPTER 3: Using jQuery	233
What Is jQuery?	233
Loading the jQuery Library	234
Downloading to your server	235
Using a Content Delivery Network (CDN).....	236
jQuery Basics	237
Using the jQuery() function	237
Using jQuery selectors	237
Is the document ready?	239
Replacing Data	240
Working with text	241
Working with HTML	243
Working with attributes	243
Working with form values	244
Changing Styles	244
Playing with properties.....	244
Using CSS objects	247
Using CSS classes.....	247
Changing the DOM	249
Adding a node	249
Removing a node.....	250
Playing with Animation.....	251
CHAPTER 4: Reacting to Events with JavaScript and jQuery	253
What Are Events?	253
Event-driven programming	254
Watching the mouse	254
Listening for keystrokes	255
Your page is talking	256
JavaScript and Events	257
Saying hello and goodbye	257
Listening for mouse events	259
Listening for keystrokes	262
Event listeners	265
jQuery and Events	265
jQuery event functions	265
The jQuery event handler.....	269
CHAPTER 5: Troubleshooting JavaScript Programs	273
Identifying Errors	273
Working with Browser Developer Tools.....	275
The DOM Explorer	276
The Console	278
The Source debugger	280
Working Around Errors.....	286

BOOK 4: PHP	291
CHAPTER 1: Understanding PHP Basics	293
Why Use PHP?	293
Centralized programming language	294
Centralized data management	294
Using PHP	295
Embedding PHP code	295
Identifying PHP pages	296
Displaying output	297
Handling newline characters	300
Working with PHP Variables	301
Declaring variables	301
Data types	302
Array variables	305
Utilizing PHP Operators	307
Arithmetic operators	307
Arithmetic shortcuts	308
Boolean operators	309
String operators	310
Including Files	311
The include() function	311
The require() function	313
CHAPTER 2: PHP Flow Control	315
Using Logic Control	315
The if statement	316
The else statement	318
The elseif statement	318
The switch statement	320
Looping	321
The while family	321
The for statement	323
The foreach statement	324
Building Your Own Functions	326
Event-Driven PHP	329
Working with links	330
Processing form data	333
CHAPTER 3: PHP Libraries	339
Understanding How PHP Uses Libraries	339
Exploring PHP extensions	340
Examining the PHP core and bundled extensions	341
Including bundled extensions	342
Adding extensions	343

Tapping Into Text Functions	344
Altering string values	344
Splitting strings	346
Testing string values	348
Searching strings	350
Working with Math Functions	351
Arithmetic functions	351
Calculating logs and exponents	352
Working the angles	353
Hyperbolic functions	353
Tracking statistics	354
Dealing with Date and Time Functions	354
Generating dates	354
Using timestamps	357
Calculating dates	358
Accessing Image-Handling Functions	359
CHAPTER 4: Considering PHP Security	365
Exploring PHP Vulnerabilities	365
What is cross-site scripting?	366
Tracking data spoofing	369
Preventing invalid data	370
Blocking unauthorized file access	372
PHP Vulnerability Solutions	373
Sanitizing data	373
Validating data	378
CHAPTER 5: Object-Oriented PHP Programming	385
Understanding the Basics of Object-Oriented Programming	385
Defining a class	386
Creating an object instance	388
Using Magic Class Methods	391
Defining mutator magic methods	391
Defining accessor magic methods	393
Using the constructor	396
Using constructor promotion	397
Forcing data types	397
The destructor	398
Copying objects	399
Displaying objects	399
Loading Classes	400
Extending Classes	405

CHAPTER 6: Sessions and Carts	411
Storing Persistent Data	411
The purpose of HTTP cookies	412
Types of cookies	413
The anatomy of a cookie	414
Cookie rules	416
Interacting with Cookies in PHP	416
Setting cookies	416
Reading cookies	419
Modifying and deleting cookies	421
Working with Sessions in PHP	422
Starting a session	423
Storing and retrieving session data	423
Removing session data	427
Creating Online Shopping Carts	427
Creating a cart	428
Placing items in the cart	428
Retrieving items from a cart	428
Removing items from a cart	429
Putting it all together	429
 BOOK 5: MYSQL	 435
CHAPTER 1: Introducing MySQL	437
Understanding the Purpose of a Database	437
How databases work	438
Relational databases	441
Database data types	443
Data constraints	443
Structured Query Language (SQL)	444
Presenting MySQL	446
MySQL features	446
Storage engines	448
Data permissions	449
Advanced MySQL Features	450
Handling transactions	450
Making sure your database is ACID compliant	451
Examining the views	454
Working with stored procedures	455
Pulling triggers	455
Working with BLOBs	456

CHAPTER 2: Administering MySQL	457
MySQL Administration Tools	457
Working from the command line	458
Using MySQL Workbench.....	463
Using the phpMyAdmin tool	467
Managing User Accounts	469
Creating a user account	470
Managing user privileges	474
CHAPTER 3: Designing and Building a Database	483
Managing Your Data	483
The first normal form	484
The second normal form	485
The third normal form	485
Creating Databases.....	486
Using the MySQL command line.....	486
Using MySQL Workbench.....	489
Using phpMyAdmin.....	492
Building Tables.....	494
Working with tables using the command-line interface	494
Working with tables using Workbench.....	499
Working with tables in phpMyAdmin	503
CHAPTER 4: Using the Database	507
Working with Data.....	507
The MySQL command-line interface	508
The MySQL Workbench tool	513
The phpMyAdmin tool	516
Searching for Data.....	519
The basic SELECT format	519
More advanced queries	521
Playing It Safe with Data.....	526
Performing data backups.....	526
Restoring your data.....	533
CHAPTER 5: Communicating with the Database from PHP Scripts	537
Database Support in PHP.....	537
Using the mysqli Library.....	539
Connecting to the database.....	540
Closing the connection.....	541
Submitting queries	542
Retrieving data.....	543

	Being prepared	545
	Checking for errors	547
	Using miscellaneous functions	549
	Putting It All Together	550
BOOK 6: CREATING OBJECT-ORIENTED PROGRAMS.....		557
CHAPTER 1:	Designing an Object-Oriented Application.....	559
	Determining Application Requirements.....	560
	Creating the Application Database	561
	Designing the database	561
	Creating the database.....	564
	Designing the Application Objects	567
	Designing objects	567
	Coding the objects in PHP	569
	Designing the Application Layout	575
	Designing web page layout	575
	Designing the AuctionHelper page layout.....	577
	Coding the Website Layout	578
	Creating the web page template.....	578
	Creating the support files.....	583
CHAPTER 2:	Implementing an Object-Oriented Application ...	589
	Working with Events	589
	Bidder Object Events.....	591
	Listing bidders	591
	Adding a new bidder.....	599
	Searching for a bidder	602
	Responding to Item Object Events	602
	Listing items	602
	Adding a new item.....	608
	Searching for an item	611
	Logging Out of a Web Application.....	611
	Testing Web Applications.....	613
CHAPTER 3:	Using AJAX	615
	Getting to Know AJAX	615
	Communicating Using JavaScript	617
	Considering XMLHttpRequest class methods.....	618
	Focusing on XMLHttpRequest class properties.....	619
	Trying out AJAX.....	621
	Using the jQuery AJAX Library	625
	The jQuery \$.ajax() function.....	625
	The jQuery \$.get() function.....	629
	Transferring Data in AJAX.....	631

	Looking at the XML standard	632
	Using XML in PHP	634
	Using XML in JavaScript.	637
	Modifying the AuctionHelper Application	640
CHAPTER 4:	Extending WordPress	647
	Getting Acquainted with WordPress.	647
	What WordPress can do for you	648
	How to run WordPress	649
	Parts of a WordPress website	650
	Installing WordPress.	651
	Downloading the WordPress software.	652
	Creating the database objects.	653
	Configuring WordPress.	654
	Examining the Dashboard	658
	Customizing Your WordPress Website.	660
	Exploring the World of Plugins	662
	WordPress APIs	662
	Working with plugins and widgets	664
	Creating Your Own Widget.	667
	Coding the widget	667
	Activating the widget plugin.	669
	Adding the widget	670
	BOOK 7: USING PHP FRAMEWORKS	673
CHAPTER 1:	The MVC Method	675
	Getting Acquainted with MVC	675
	Exploring the MVC method	676
	Digging into the MVC components.	678
	Communicating in MVC	682
	MVC and microservices	683
	Comparing MVC to Other Web Models	685
	The MVP method.	685
	The MVVM method	686
	Seeing How MVC Fits into N-Tier Theory	687
	Implementing MVC	688
CHAPTER 2:	Selecting a Framework	689
	Getting to Know PHP Frameworks	689
	Convention over configuration	690
	Scaffolding	692
	Routing	693
	Helper methods.	694
	Form validation	694

Support for mobile devices	694
Templates	695
Unit testing	695
Knowing Why You Should Use a Framework	696
Focusing on Popular PHP Frameworks	698
CakePHP	698
CodeIgniter	700
Laravel	701
Symfony	702
Laminas Framework	703
Looking At Micro-Frameworks	705
Slim	705
Yii	706
CHAPTER 3: Creating an Application Using Frameworks	707
Building the Template	707
Initializing the application	708
Exploring the files and folders	710
Defining the database environment	711
Creating an Application Scaffold	713
Installing the scaffolding	714
Exploring the scaffolding code	717
Modifying the Application Scaffold	718
Adding a new feature link	719
Creating the controller code	722
Modifying the model code	723
Painting a view	725
INDEX	729

Introduction

The Internet has become an amazing place to shop, do your banking, look up homework assignments, and even keep track of your bowling league scores. Behind all those great applications are a bunch of different web technologies that must all work together to create the web experience you come to expect.

You may think that creating web applications is best left for the professionals, but you'd be surprised by just how well you can do with just a little knowledge and experience! That's the point of this book.

About This Book

Think of this book as a reference book. Like the dictionary or an encyclopedia (remember those?), you don't have to read it from beginning to end. Instead, you can dip into the book to find the information you need and return to it again when you need more. That said, you won't be disappointed if you work through the book from beginning to end, and you may find it easier to follow along with some of the examples.

In this book, I walk you through all the different technologies involved with creating dynamic web applications that can track data and present it in an orderly and pleasing manner. I cover several key topics that you'll need to know to create a full-featured, dynamic web application:

- » **Creating the basic layout of a web page:** In this book, you see the program code behind placing content on a web page and reacting to your website visitors' mouse clicks.
- » **Styling the web page:** Just placing data on a web page is boring. In this book, you learn how to use CSS to help use color, images, and placement to help liven up your web applications.
- » **Adding dynamic features:** These days, having a static web page that just sits there doesn't get you many followers. This book shows you how to incorporate JavaScript to animate your web pages and provide dynamic features.

- » **Leveraging the power of the server:** The PHP programming language allows you to harness the power behind the web server to dynamically generate web pages “on the fly” as your website visitors make choices.
- » **Storing data for the future:** Just about every dynamic web application needs to store data, and in this book you learn exactly how to do that using the MySQL server, which is commonly available in just about every web platform.
- » **Creating full applications:** Many books throw a bunch of technology at you and expect you to put the pieces together yourself. This book not only shows you the technology, but also demonstrates how all the parts fit together to create a dynamic web application.
- » **Using helper programs:** No one is an island; everyone needs some help putting together those fancy web applications. There are plenty of tools to help you get the job done, and with this book you find out which tools will help you with which features of your application.

Throughout this book you see sidebars (text in gray boxes) and material marked with the Technical Stuff icon. All of these things are skippable. If you have time and are interested, by all means read them, but if you don’t or aren’t, don’t.

Finally, within this book, you may note that some web addresses break across two lines of text. If you’re reading this book in print and want to visit one of these web pages, simply key in the web address exactly as it’s noted in the text, pretending as though the line break doesn’t exist. If you’re reading this as an e-book, you’ve got it easy — just click the web address to be taken directly to the web page.

Foolish Assumptions

You don’t need any level of programming experience to enjoy this book and start creating your own web applications. Each chapter walks through all the basics you need to know and doesn’t assume you’ve ever coded before. As long as you’re reasonably comfortable navigating your way around a standard desktop computer, you have all the experience you need!

That said, if you’ve already tried your hand at web programming and you just want to fill in a few holes, this book will work well for you, too!

This book doesn’t expect you to run out and buy any expensive software packages to start your web development career. All the tools that are used in the book are

freely available open-source software. I walk you through how to set up a complete development environment, whether you're working in Microsoft Windows, Apple macOS, or Linux.

Icons Used in This Book

I use some icons throughout the book to help you identify useful information. Here are the icons and what I use them for:



TIP

Anything marked with the Tip icon provides some additional information about a topic to help you better understand what's going on behind the scenes or how to better use the feature discussed in the text.



REMEMBER

You don't have to commit this book to memory — there won't be a test. But every once in a while, I tell you something so important that you should remember it. When I do, I mark it with the Remember icon.



WARNING

The Warning icon is there to point out potential pitfalls that can cause problems. If you want to save yourself a lot of time or trouble, heed these warnings.



TECHNICAL
STUFF

This icon explains technical details about the concept being explained. The details might be informative or interesting, but are not essential to your understanding of the concept at this stage.

Beyond the Book

So that you don't have to spend your precious time typing in long blocks of code, I have provided much of the code you find in this book for you online. You can download this code for each book and chapter by going to www.dummies.com/go/phpmysqlandjavascriptaio2e and following the instructions there.

You also get access to a free online Cheat Sheet filled with more tips and tricks on building a web application, including accessing any database from your PHP programs, filtering data your program receives from web forms to block unwanted or potentially dangerous data, quickly finding data in a MySQL database, and triggering JavaScript events at predetermined times in a browser. To access this resource go to www.dummies.com and enter **PHP, MySQL & JavaScript All-in-One For Dummies Cheat Sheet** in the search box.

Where to Go from Here

This book doesn't have to be read from beginning to end, so you can dive in wherever you want! Use the Table of Contents and Index to find subjects that interest you. If you already know PHP and JavaScript and you're just interested in learning how to create a dynamic web application from scratch, start out with Book 6, Chapter 1. If you're interested in learning how to use one of the framework packages available for PHP, check out Book 7, Chapter 1. Or, if you're interested in everything, start with Book 1, Chapter 1, and read until the very end.

With the information in this book, you'll be ready to start creating your own dynamic web applications. Web programming is one of those skills that takes time and practice to get good at, so the more coding you can do, the better you'll get at it. To get some practice, you may want to offer your services for free at first, to build up a reputation. Find a needy nonprofit organization that you're interested in supporting and offer to work on its website. They'll get a great website, and you'll get a project to add to your résumé!

Don't stop learning! There are always new things coming out in the web world, even if you just stick to using the same software packages to develop your web applications. Stay plugged in to the PHP world by visiting the official PHP website at www.php.net or by visiting (and even participating in) one or more of the many PHP forums. Just do some Googling to find them.

Enjoy your newfound skills in developing dynamic web applications!

1

Getting Started with Web Programming

Contents at a Glance

CHAPTER 1: Examining the Pieces of Web Programming	7
Creating a Simple Web Page	7
Creating a Dynamic Web Page	18
Storing Content	26
CHAPTER 2: Using a Web Server	29
Recognizing What's Required.	29
Considering Your Server Options	33
Tweaking the Servers	45
Building a Development Environment.....	53

IN THIS CHAPTER

- » Understanding how simple web pages work
- » Incorporating programming into your web page
- » Storing content in a database

Chapter **1**

Examining the Pieces of Web Programming

At first, diving into web programming can be somewhat overwhelming. You need to know all kinds of things in order to build a web application that not only looks enticing but also works correctly. The trick to learning web programming is to pull the individual pieces apart and tackle them one at a time.

This chapter gets you started on your web design journey by examining the different pieces involved in creating a simple web page. Then it kicks things up a notch and walks you through dynamic web pages. And finally, the chapter ends by explaining how to store your content for use on the web.

Creating a Simple Web Page

Before you can run a marathon, you need to learn how to walk. Likewise, before you can create a fancy website, you need to know the basics of how web pages work.

Nowadays, sharing documents on the Internet is easy, but it wasn't always that way. Back in the early days of the Internet, documents were often created using proprietary word-processing packages and had to be downloaded using the

cumbersome File Transfer Protocol (FTP). To retrieve a document, you had to know exactly what server contained the document, you had to know where it was stored on the server, and you had to be able to log into the server. After all that, you *still* needed to have the correct word-processing software on your computer to view the document. As you can imagine, it wasn't long before a new way of sharing content was required.

To get to where we are today, several different technologies had to be developed:

- » A method for linking related documents together
- » A way for the document reader to display formatted text the same way in any type of device
- » An Internet standard allowing clients to easily retrieve documents from any server
- » A standard method of styling and positioning content in documents

This section describes the technology that made viewing documents on the Internet work the way it does today.

Kicking things off with the World Wide Web

In 1989, Tim Berners-Lee developed a method of interconnecting documents to make sharing research information on the Internet easier. His creation, the *World Wide Web*, defined a method for linking documents together in a web structure, so that a researcher could follow the path between related documents, no matter where they were located in the world. Clicking text in one document took you to another document automatically, without your having to manually find and download the related document.

The method Berners-Lee developed for linking documents is called *hypertext*. Hypertext embeds links that are hidden from view in the document and directs the software being used to view the document (known as the *web browser*) to retrieve the referenced document. With hypertext, you just click the link, and the software (the web browser) does all the work of finding and retrieving the related document for you.

Because the document-viewing software does all the hard work, a new type of software had to be developed that was more than just a document viewer. That's where web browsers came into existence. Web browsers display a document on a computer screen and respond to the reader clicking hypertext links to retrieve other specified documents.

To implement hypertext in documents, Berners-Lee had to utilize a text-based document-formatting system. Fortunately for him, a lot of work had already been done on that.

Making sense of markup languages

Markup languages were developed to replace proprietary word-processing packages with a standard way of formatting documents so that they could be read by any type of document viewer on any type of device. This goal is accomplished by embedding *tags* in the text. Each tag indicates a formatting feature, such as headings, bold or italic text, or special margins. What made markup languages different from word-processing packages is that these tags were common text codes instead of proprietary codes, making it generic enough that any device could read and process them.

The first popular markup language was the Generalized Markup Language (GML), developed by IBM in the 1960s. The International Organization for Standardization (ISO) took up the challenge of creating markup languages and produced the Standard Generalized Markup Language (SGML), mainly based on GML, in the 1980s. However, because SGML was developed to cover all types of document formatting on all types of devices, it's extremely complex and it wasn't readily adapted.

Berners-Lee used the ideas developed in SGML to create a simplified markup language that could support his hypertext idea. He called it *Hypertext Markup Language* (HTML). HTML uses the same concept of tags that SGML uses, but it defines fewer of them, making it easier to implement in software.

An example of an HTML tag is `<h1>`. You use this tag to define text that's used as a page heading. Just surround the text with an opening `<h1>` tag, and a corresponding closing `</h1>` tag, like this:

```
<h1>This is my heading</h1>
```

When the browser gets to the `<h1>` tag, it knows to format the text embedded in the opening and closing tags using a different style of formatting, such as a larger font or a bold typeface.

To define a hypertext link to another document, you use the `<a>` tag:

```
<a href="anotherdoc.html">Click here for more info</a>
```

When the reader clicks the *Click here for more info* text, the browser automatically tries to retrieve the document specified in the `<a>` tag. That document can be on the same server or on another server anywhere on the Internet.

HTML development has seen quite a few changes since Berners-Lee created it and turned it over to the World Wide Web Consortium (W3C) to maintain. After many years of faithfully maintaining the HTML standard, unfortunately, it had met with some controversy, as a competing standard, maintained by the Web Hypertext Application Technology Working Group (WHATWG), a consortium of several vendors, emerged. Table 1-1 shows the path the HTML standard has taken.

TABLE 1-1

HTML Versions

Version	Description
HTML 1.0	Formally released in 1989 as the first public version of HTML
HTML 2.0	Released in 1995 to add interactive elements
HTML 3.0	Released in 1996 but never widely adopted
HTML 3.2	Released in 1997, adding support for tables
HTML 4.01	Released in 1999, widely adopted, and remains an often-used standard
XHTML 1.0	Released in 2001, standardizing HTML around the XML document format
XHTML 1.1	Released in 2002, making updates and corrections to XHTML 1.1
HTML 5.0	Released in 2014, adding multimedia features
HTML 5.1	Released in mid-2017, adding form validation and context menus
HTML 5.2	Released in late-2017, adding additional styling features
HTML 5.3	Also released in late-2017, this was the final version released by the W3C

In 2019, the W3C stopped as the sole maintainer of the official HTML standard and joined with the WHATWG consortium to produce a single HTML standard, called the *HTML Living Standard*. This is now considered the official HTML standard, and the standard that this book focuses on. The Living Standard doesn't have specific release versions, but instead, incorporates changes "on the fly" to the HTML specifications once they are approved by their board. You can find the latest HTML features described at the WHATWG website, html.spec.whatwg.org/multipage/. The WHATWG documentation refers to the term HTML5 as a buzzword, often used to describe the modern HTML standard.

Retrieving HTML documents

Besides a document-formatting standard, Berners-Lee also developed a method of easily retrieving the HTML documents in a client-server environment. A *web server* software package runs in the background on a server, listening for connection requests from *web clients* (the browser). The browser sends requests to retrieve HTML documents from the server. The request can be sent anonymously (without using a login username), or the browser can send a username and password or certificate to identify the requestor.

These requests and responses are defined in the *Hypertext Transfer Protocol* (HTTP) standard. HTTP defines a set of requests the client can send to the server and a set of responses the server uses to reply back to the client.

This section walks you through the basics of how web servers and web clients use HTTP to interact with each other to move web pages across the Internet.

Web clients

The web client sends requests to the web server on a standard network communication channel. The requests use a common Internet protocol known as Transmission Control Protocol (TCP). Each TCP connection uses a defined port number to communicate between the server application and clients. HTTP uses TCP port 80 as the standard for communications between servers and clients. HTTP uses standard text requests sent to the server, either requesting information from the server or sending information to the server. Table 1-2 shows the basic HTTP client requests available.

TABLE 1-2

HTTP Client Requests

Request	Description
CONNECT	Converts the connection into a secure tunnel for sending data
DELETE	Deletes the specified resource
GET	Requests the specified resource
HEAD	Requests the title of the specified resource
OPTIONS	Retrieves the HTTP requests that the server supports
PATCH	Applies a modification to a resource
POST	Sends specified data to the server for processing
PUT	Stores specified data at a specified location
TRACE	Sends the received request back to the client

As shown in Table 1-2, when you ask to view a web page from your client browser, the browser sends the HTTP GET request to the server, specifying the filename of the web page. The server then responds with a response code along with the requested data. If the client doesn't specify a filename in the GET request, most servers have a default file with which to respond.

Web servers

With HTTP, the web server must respond to each client request received. If the client sends a request that the server can't process, the server must send some type of error code back to the client indicating that something went wrong.

The first part of the server response is a status code and text that the client uses to determine whether the submitted request was successful. The format of the HTTP response uses a three-digit status code, followed by an optional text message that the browser can display. The three-digit codes are broken down into five categories:

- » **1xx:** Informational messages
- » **2xx:** Success
- » **3xx:** Redirection
- » **4xx:** Client error
- » **5xx:** Server error

The three-digit status code is crucial to knowing what happened with the response. Many status codes are defined in the HTTP standards, providing some basic information on the status of client requests. Table 1-3 shows just a few of the standard HTTP response codes that you may run into.

As you can see from Table 1-3, a web server can return many possible responses. It's the client's job to parse the response and determine the next action to take.

If the response indicates the request was successful, the server will follow the response code with the data related to the request, such as the contents of an HTML file. The client must then read the returned data and decide what to do with it. For HTML files, the browser will display the requested file, applying the HTML formatting tags to the data.



TIP

Don't worry about trying to memorize all the HTTP status codes. Most of them you'll never run into in your web-programming career. Before long, you'll start to remember a few of the more common ones, and you can always look up any others you run into.