

# The Complete Reference

Covers  
PHP 5.2

# PHP

- Design, debug, and deploy feature-rich Web applications
- Work with HTML, XML, database, and multimedia content
- Learn advanced AJAX techniques

For Sale in  
India, Pakistan  
Nepal, Bangladesh  
Sri Lanka and Bhutan  
only



TATA McGRAW-HILL  
EDITION

**Steven Holzner**



**Tata McGraw-Hill**

Information has been obtained by McGraw-Hill from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, McGraw-Hill, or others, McGraw-Hill does not guarantee the accuracy, adequacy, or completeness of any information and is not responsible for any errors or omissions or the results obtained from the use of such information.

## **PHP: The Complete Reference**

Copyright © 2008 by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of publisher.

## **Tata McGraw-Hill Edition 2008**

Second reprint 2008  
**RXYZCDRXRBQDD**

Reprinted in India by arrangement with The McGraw-Hill Companies, Inc.,  
New York

Sales territories: India, Pakistan, Nepal, Bangladesh, Sri Lanka and Bhutan .

## **Library of Congress Cataloging-in-Publication Data**

Holzner, Steven.

PHP: the complete reference / Steven Holzner.

p. cm.

ISBN 0-07-150854-6 (alk. paper)

I. PHP (Computer program language) I. Title.

QA76.73.P224H64 2007

005.2'762—dc22

2007044733

**ISBN-13: 978-0-07-022362-2**

**ISBN-10: 0-07-022362-9**

Published by Tata McGraw-Hill Publishing Company Limited,  
7 West Patel Nagar, New Delhi 110 008, and printed at  
Lalit Offset Printers, New Delhi 110 092

*The McGraw-Hill Companies*

---

# Contents at a Glance

<b>1</b>	<b>Essential PHP</b>	<b>1</b>
<b>2</b>	<b>Operators and Flow Control</b>	<b>41</b>
<b>3</b>	<b>Strings and Arrays</b>	<b>81</b>
<b>4</b>	<b>Creating Functions</b>	<b>123</b>
<b>5</b>	<b>Reading Data in Web Pages</b>	<b>161</b>
<b>6</b>	<b>PHP Browser-Handling Power</b>	<b>203</b>
<b>7</b>	<b>Object-Oriented Programming</b>	<b>245</b>
<b>8</b>	<b>Advanced Object-Oriented Programming</b>	<b>281</b>
<b>9</b>	<b>File Handling</b>	<b>319</b>
<b>10</b>	<b>Working with Databases</b>	<b>361</b>
<b>11</b>	<b>Sessions, Cookies, and FTP</b>	<b>395</b>
<b>12</b>	<b>Ajax</b>	<b>433</b>
<b>13</b>	<b>Advanced Ajax</b>	<b>467</b>
<b>14</b>	<b>Drawing Images on the Server</b>	<b>501</b>
<b>15</b>	<b>XML and RSS</b>	<b>537</b>
	<b>Index</b>	<b>575</b>

# Contents

<u>Introduction</u> .....	xvii
<b>1 Essential PHP</b> .....	1
Enter PHP .....	1
Getting PHP .....	3
PHP on the Internet .....	4
PHP on Your Local Machine .....	5
Creating Your Development Environment .....	6
Creating a First PHP Page .....	8
Running Your First PHP Page .....	9
Some Troubleshooting .....	9
Mixing HTML and PHP .....	10
Printing Some Text .....	14
Printing Some HTML .....	16
More Echo Power .....	17
Using PHP "Here" Documents .....	19
Command-Line PHP .....	20
Adding Comments to PHP Code .....	24
Working with Variables .....	26
Storing Data in Variables .....	27
Interpolating Strings .....	31
Creating Variable Variables .....	33
Creating Constants .....	35
Understanding PHP's Internal Data Types .....	37
<b>2 Operators and Flow Control</b> .....	41
PHP's Math Operators .....	41
Working with the Assignment Operators .....	46
Incrementing and Decrementing Values .....	48
The PHP String Operators .....	50
The Bitwise Operators .....	51
The Execution Operator .....	52
PHP Operator Precedence .....	53
Using the if Statement .....	55
The PHP Comparison Operators .....	59
The PHP Logical Operators .....	61

The else Statement .....	63
The elseif Statement .....	65
<b>The ternary Operator</b> .....	<b>66</b>
The switch Statement .....	67
Using for Loops .....	69
Using while Loops .....	72
Using do...while Loops .....	74
Using the foreach Loop .....	76
Terminating Loops Early .....	77
<b>Skiping Iterations</b> .....	<b>78</b>
PHP Alternate Syntax .....	80
<b>3 Strings and Arrays</b> .....	<b>81</b>
The String Functions .....	81
Converting to and from Strings .....	87
Formatting Text Strings .....	88
Building Yourself Some Arrays .....	92
Modifying the Data in Arrays .....	95
Deleting Array Elements .....	97
Handling Arrays with Loops .....	99
The for Loop .....	99
The print_r Function .....	100
The foreach Loop .....	101
The while Loop .....	103
The PHP Array Functions .....	104
<b>Converting Between Strings and Arrays Using implode and explode</b> .....	<b>106</b>
Extracting Data from Arrays .....	107
Sorting Arrays .....	109
<b>Using PHP's Array Operators</b> .....	<b>110</b>
Comparing Arrays to Each Other .....	112
Handling Multidimensional Arrays .....	112
<b>Using Multidimensional Arrays in Loops</b> .....	<b>114</b>
Moving Through Arrays .....	116
Splitting and Merging Arrays .....	117
Other Array Functions .....	119
<b>4 Creating Functions</b> .....	<b>123</b>
Creating Functions in PHP .....	123
Passing Functions Some Data .....	125
Passing Arrays to Functions .....	127
Passing by Reference .....	130
Using Default Arguments .....	132
Passing Variable Numbers of Arguments .....	133

<b>Returning Data from Functions</b>	135
<b>Returning Arrays</b>	137
<b>Returning Lists</b>	139
<b>Returning References</b>	141
<b>Introducing Variable Scope in PHP</b>	143
<b>Accessing Global Data</b>	145
<b>Working with Static Variables</b>	147
<b>PHP Conditional Functions</b>	150
<b>PHP Variable Functions</b>	153
<b>Nesting Functions</b>	156
<b>Creating Include Files</b>	157
<b>Returning Errors from Functions</b>	158
<b>5 Reading Data in Web Pages</b>	161
<b>Setting Up Web Pages to Communicate with PHP</b>	161
<b>Handling Text Fields</b>	164
<b>Handling Text Areas</b>	167
<b>Handling Check Boxes</b>	170
<b>Handling Radio Buttons</b>	173
<b>Handling List Boxes</b>	175
<b>Handling Password Controls</b>	179
<b>Handling Hidden Controls</b>	182
<b>Handling Image Maps</b>	184
<b>Handling File Uploads</b>	187
<b>Handling Buttons</b>	191
<b>Making Button Data Persist</b>	192
<b>Using Submit Buttons as HTML Buttons</b>	195
<b>6 PHP Browser-Handling Power</b>	203
<b>Using PHP's Server Variables</b>	203
<b>Using HTTP Headers</b>	205
<b>Getting the User's Browser Type</b>	206
<b>Redirecting Browsers with HTTP Headers</b>	209
<b>Dumping a Form's Data All at Once</b>	212
<b>Handling Form Data with Custom Arrays</b>	215
<b>Putting It All in One Page</b>	218
<b>Performing Data Validation</b>	221
<b>Checking if the User Entered Required Data</b>	223
<b>Requiring Numbers</b>	227
<b>Requiring Text</b>	230
<b>Persisting User Data</b>	234
<b>Client-Side Data Validation</b>	237
<b>Handling HTML Tags in User Input</b>	241

<b>7 Object-Oriented Programming .....</b>	<b>245</b>
Creating Classes .....	246
<b>Creating Objects .....</b>	<b>250</b>
Setting Access to Properties and Methods .....	253
Public Access .....	253
Private Access .....	254
<b>Using Constructors to Initialize Objects .....</b>	<b>257</b>
<b>Using Destructors to Clean Up after Objects .....</b>	<b>260</b>
Basing One Class on Another with Inheritance .....	262
Protected Access .....	264
Constructors and Inheritance .....	266
Calling Base Class Methods .....	267
Overriding Methods .....	271
Overloading Methods .....	273
Autoloading Classes .....	277
<b>8 Advanced Object-Oriented Programming .....</b>	<b>281</b>
Creating Static Methods .....	281
<b>Creating a Static Method .....</b>	<b>283</b>
Passing Data to a Static Method .....	285
Using Properties in Static Methods .....	286
Static Members and Inheritance .....	291
Creating Abstract Classes .....	294
Creating Interfaces .....	297
Supporting Object Iteration .....	301
Comparing Objects .....	304
Creating Class Constants .....	306
Using the final Keyword .....	308
Cloning Objects .....	312
Reflection .....	315
<b>9 File Handling .....</b>	<b>319</b>
<b>Opening Files Using fopen .....</b>	<b>319</b>
Looping over a File's Contents with feof .....	322
Reading Text from a File Using fgets .....	322
Closing a File .....	323
<b>Reading from a File Character by Character with fgetc .....</b>	<b>325</b>
Reading a Whole File at Once with file_get_contents .....	328
<b>Reading a File into an Array with file .....</b>	<b>330</b>
Checking if a File Exists with file_exists .....	332
Getting File Size with filesize .....	334
Reading Binary Reads with fread .....	335

Parsing Files with <code>fscanf</code>	338
Parsing ini Files with <code>parse_ini_file</code>	339
Getting File Info with <code>stat</code>	341
<b>Setting the File Pointer's Location with <code>fseek</code></b>	<b>343</b>
<b>Copying Files with <code>copy</code></b>	<b>343</b>
Deleting Files with <code>unlink</code>	345
Writing to a File with <code>fwrite</code>	346
Reading and Writing Binary Files	348
Appending to Files with <code>fwrite</code>	352
Writing a File All at Once with <code>file_put_contents</code>	355
Locking Files	357
<b>10 Working with Databases</b>	<b>361</b>
What Is a Database?	362
Some Essential SQL	362
Creating a MySQL Database	364
Creating a New Table	367
Putting Data into the New Database	368
Accessing the Database in PHP	370
Connecting to the Database Server	371
Connecting to the Database	372
Reading the Table	372
Displaying the Table Data	374
Closing the Connection	376
Updating Databases	377
Inserting New Data Items into a Database	380
Deleting Records	383
Creating New Tables	385
Creating a New Database	389
Sorting Your Data	393
<b>11 Sessions, Cookies, and FTP</b>	<b>395</b>
Setting a Cookie	395
Reading a Cookie	397
Setting Cookies' Expiration	399
Deleting Cookies	400
Working with FTP	402
<b>    Downloading Files with FTP</b>	<b>406</b>
Uploading Files with FTP	408
Deleting a File with FTP	411
Creating and Removing Directories with FTP	414
<b>    Sending E-mail</b>	<b>416</b>

Sending Advanced E-mail .....	418
Adding Attachments to E-mail .....	421
Storing Data in Sessions .....	425
Writing a Hit Counter Using Sessions .....	429
<b>12 Ajax .....</b>	<b>433</b>
<b>Getting Started with Ajax .....</b>	<b>433</b>
Writing Ajax .....	435
Creating the XMLHttpRequest Object .....	436
<b>Opening the XMLHttpRequest Object .....</b>	<b>440</b>
Handling Downloaded Data .....	441
Starting the Download .....	445
Creating XMLHttpRequest Objects .....	446
Ajax with Some PHP .....	448
Passing Data to the Server with GET .....	449
Passing Data to the Server with POST .....	453
Handling XML .....	456
Handling XML with PHP .....	464
<b>13 Advanced Ajax .....</b>	<b>467</b>
Handling Concurrent Ajax Requests	
with Multiple XMLHttpRequest Objects .....	467
Handling Concurrent Ajax Requests with an XMLHttpRequest Array .....	472
Handling Concurrent Ajax Requests with JavaScript Inner Functions .....	475
Downloading Images Using Ajax .....	479
Downloading JavaScript with Ajax .....	481
Connecting to Google Suggest .....	484
Connecting to Other Domains Using Ajax .....	494
Logging in with Ajax and PHP .....	495
Getting Data with Head Requests and Ajax .....	497
<b>14 Drawing Images on the Server .....</b>	<b>501</b>
Creating an Image .....	504
Displaying Images in HTML Pages .....	506
Drawing Lines .....	507
Setting Line Thickness .....	510
Drawing Rectangles .....	511
Drawing Ellipses .....	513
Drawing Arcs .....	514
Drawing Polygons .....	516
Filling in Figures .....	518
Drawing Individual Pixels .....	520
Drawing Text .....	522

Drawing Vertical Text .....	525
Working with Image Files .....	528
Tiling Images .....	531
Copying Images .....	535
<b>15 XML and RSS .....</b>	<b>537</b>
Creating XML .....	537
Creating RSS .....	540
Using the SimpleXML Functions .....	544
Extracting Attributes .....	550
Using XPath .....	552
Modifying XML Elements and Attributes .....	555
Adding New Elements and Attributes .....	557
Sending XML to the Browser .....	560
Interacting with Other PHP XML Packages .....	561
Parsing with the XML Parser Functions .....	563
<b>Index .....</b>	<b>575</b>

---

# Introduction

This book is your guide to PHP, and it was written to be as complete and comprehensive as possible. It puts the power of PHP to work for you, emphasizing seeing example after example. We push the PHP envelope here, in more than a hundred examples, ready to run.

PHP is a hot topic—it's become the most popular server-side language by far. A Google search for "PHP" results in a hefty 2,890,000,000 hits. That's two billion, eight hundred and ninety million hits, far more than any other server-side language can boast.

What's behind this incredible popularity? PHP is fast and easy to use, fast and easy to develop in. You can mingle it with HTML in your Web pages. You can write it easier than other languages—PHP has learned from their mistakes. You don't need to compile it, as you do with other languages, before running it. But more than all that, developing with PHP is just plain fun. It's just a terrific language that people who write server-side code really enjoy. And this book is all about bringing you that experience.

People who have Web sites are requiring more and more power these days, and increasingly, they're finding their answer in PHP. No longer content to be limited to working with JavaScript in the browser, they want to have the power of writing code to be executed on the server. Guest books, interactive calendars, databases, autoresponder e-mailers, blogs, chat rooms—the things you can do with PHP are unlimited. Using PHP, you have total control over your Web applications—and the good part is that they're not much harder to write than the typical Web page. You can do a lot with a little.

You're getting into PHP at the right time. Excitement is soaring, and PHP is flying high. This book tries to stay as true to the spirit of that excitement as possible, giving you the full PHP experience. You're going to find more PHP in this book than in any similar book as you get the complete details on the PHP story.

---

## This Book Is for You

This is your book if you want to develop all the power that PHP is capable of, and you want to see examples at every step along the way. You might, for example, want to start putting cookies on other people's computers rather than just accepting them on your computer. You might want to read the data the users enter into text fields, list boxes, check boxes, or radio buttons on your Web page. You might want to keep data for your online store in a database on the server. You might want to track users with sessions, gaining the capability to create multipage Web applications.

Whatever your online need, this book is for you.

And this book was written so that you don't need a lot of background to use it. In fact, the only thing you need to know before reading through and working with this book is a knowledge of HTML. You won't need to be an HTML wizard, but you'll need to know some HTML in this book. If you don't have any clue when it comes to HTML, now's the time to look up an online tutorial on the subject.

We use PHP 5.2 in this book—and you may already have it on your server. If not, you'll see where to get it for free in this book, and how to install it. In fact, you won't even need a Internet server that supports PHP to read this book—if you wish, you can develop and test your PHP pages all on the same computer. On the other hand, if you want to put your PHP code on the Internet, you'll need to use an ISP that supports PHP. Check with your ISP to find if they support PHP—more and more ISPs are doing so every day.

This book has been written to be as complete as possible, and to be at the top of its field. If you have questions or comments, please drop me a line—I'd love to hear from you.

---

## Where Can You Get the Code?

All the code for the examples in this book is available online, so you don't have to type it in yourself. You should be able to unzip those examples to your server; they're ready to run (with the exception of examples that need passwords for your database system or to connect to another ISP).

You can get the code for the examples in this book at [www.mhprofessional.com](http://www.mhprofessional.com). All you have to do is to download the Zip file and unzip it—everything's in there.

Alright, that gives us the start we need. PHP is your gateway to server-side power, and you're going to get a guided tour of PHP in this book. All the PHP that can fit into a single book has been packed in here. All that remains is to get started, by turning to Chapter 1.

# CHAPTER

---

# Essential PHP

**T**ake a look at Figure 1-1. That's the Ohio State University home page. Pretty snazzy, eh? Now take a closer look at the URL in the address bar: <http://www.osu.edu/index.php>. That's a PHP page you're looking at, index.php.

Here's another page for you: <http://www1.umn.edu/twincities/index.php>, the home page for the University of Minnesota, which appears in Figure 1-2. Also a PHP page, as you can see from the URL. Not bad.

And here's another one: the Yahoo Maps page you see in Figure 1-3, <http://maps.yahoo.com/index.php>. Want driving directions? Just enter your start and end locations into that page and click Go. PHP will do the rest.

---

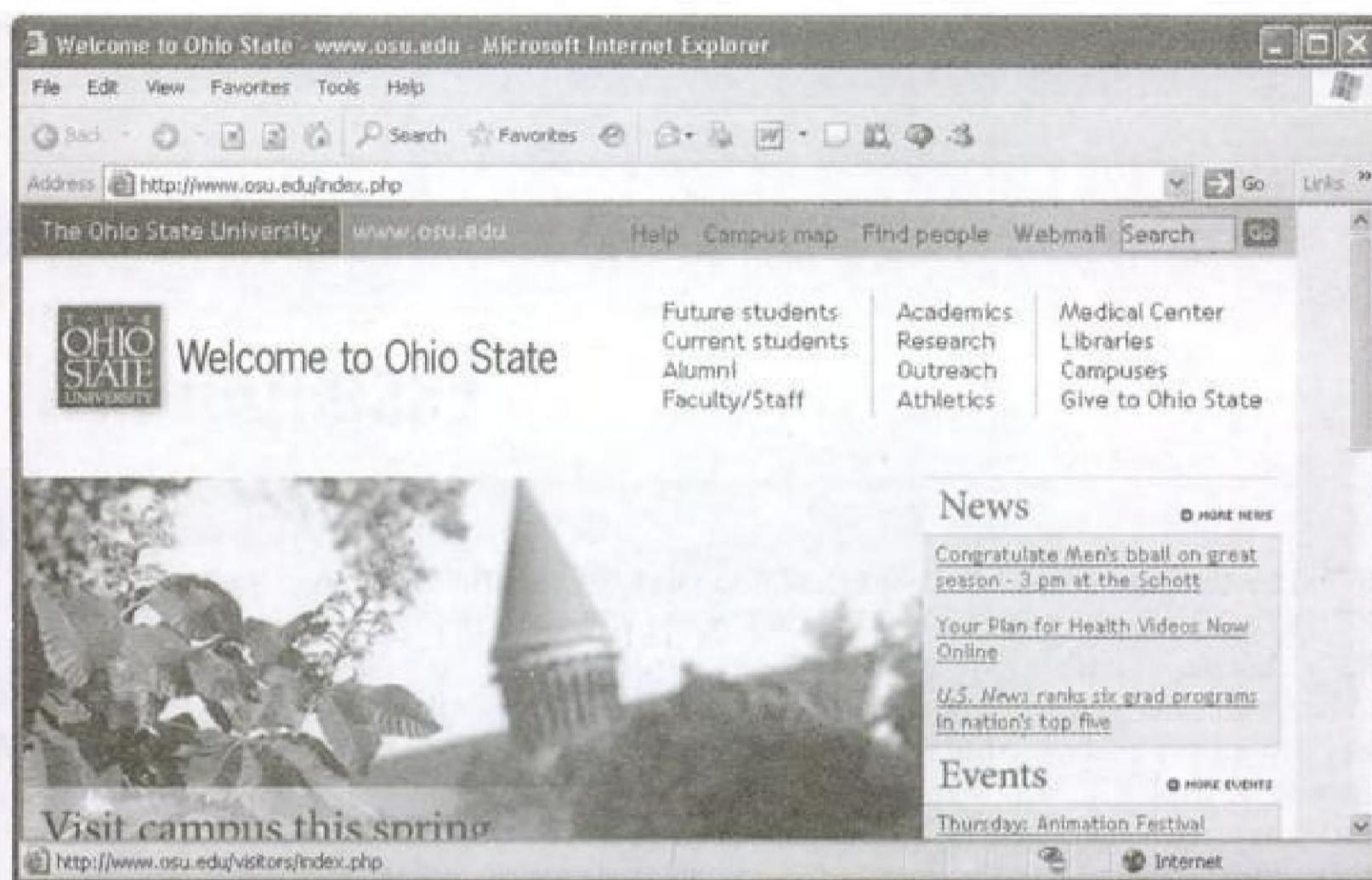
## Enter PHP

Welcome to the world of PHP. Officially, PHP stands for "PHP: Hypertext Preprocessor," but it's also still known around the world by its original name, Personal Home Page. It's the server-side programming language that's taken the Web world by storm—PHP is far and away the most popular programming language for use on Web servers. That's the idea behind PHP: being able to do some easy programming on the Web server, creating everything from online databases to guest books, from customer schedulers to chat rooms, from file uploading tools to shopping carts. It's all possible with PHP.

Where did PHP come from? PHP users are sometimes startled to learn that PHP has been around for quite some time; it was created by Rasmus Lerdorf in 1994 (Rasmus wanted a way of logging who was looking at his online résumé). PHP got such a good reputation that by 1995 it was available for use by other people, and the PHP revolution was underway.

PHP at that time was called Personal Home Page, or Personal Home Page Tools. At that time, as you might expect, PHP was very simple, and could be used to create Web page hit counters, guest books, and the like. The 1995 version of PHP was called PHP/FI Version 2 (FI was an HTML form reader package, also written by Rasmus).

In time, Rasmus added support for interfacing with Mini SQL (mSQL), and PHP/FI started growing at an astonishing rate as more people contributed code to it. There was a real need for an easy Web server programming language then, and the number of PHP pages just kept growing. In 1996, PHP/FI was already being used by about 15,000 Web pages. In 1997, that number grew to more than 50,000.



**FIGURE 1-1** The Ohio State Home page

Things started happening fast at that point. In 1997, PHP/FI became just PHP, and more people got involved as PHP teams started appearing. Much core work was redone by Zeev Suraski and Andi Gutmans, and PHP Version 3 appeared—much of it totally rewritten.

Today, PHP is everywhere you look on the Web, with an estimated 100 million PHP pages (it's hard to get accurate statistics—if you do a search on Google for PHP, for example, you get an astounding estimated 2,740,000,000 hits). PHP is still true to its original name: Personal Home Page, because it gives you the easiest way to make your Web pages come alive on the server. But PHP has also become a very professional language, suitable for top-notch sites.

You're going to see it all in this book. This is where your Web pages and applications come alive.

The lid is off the box when you start to work with PHP now. Just about anything you can do on the Web, you can do with PHP. No longer do Web pages have to be static, unchanging things—you're going to be able to interact with users in a safe and secure way, sending them back Web pages tailored to their input. And it all happens in real time.

You can handle button clicks, radio button selections, and list box choices with ease using PHP. You can code simple Web applications such as guest books, or do anything advanced that's possible on the Web: create database applications, client/server applications, and multilayer data processors; create graphics interactively on the server and send them back to the browser; register students for your online school; create Web-based classrooms; and more.

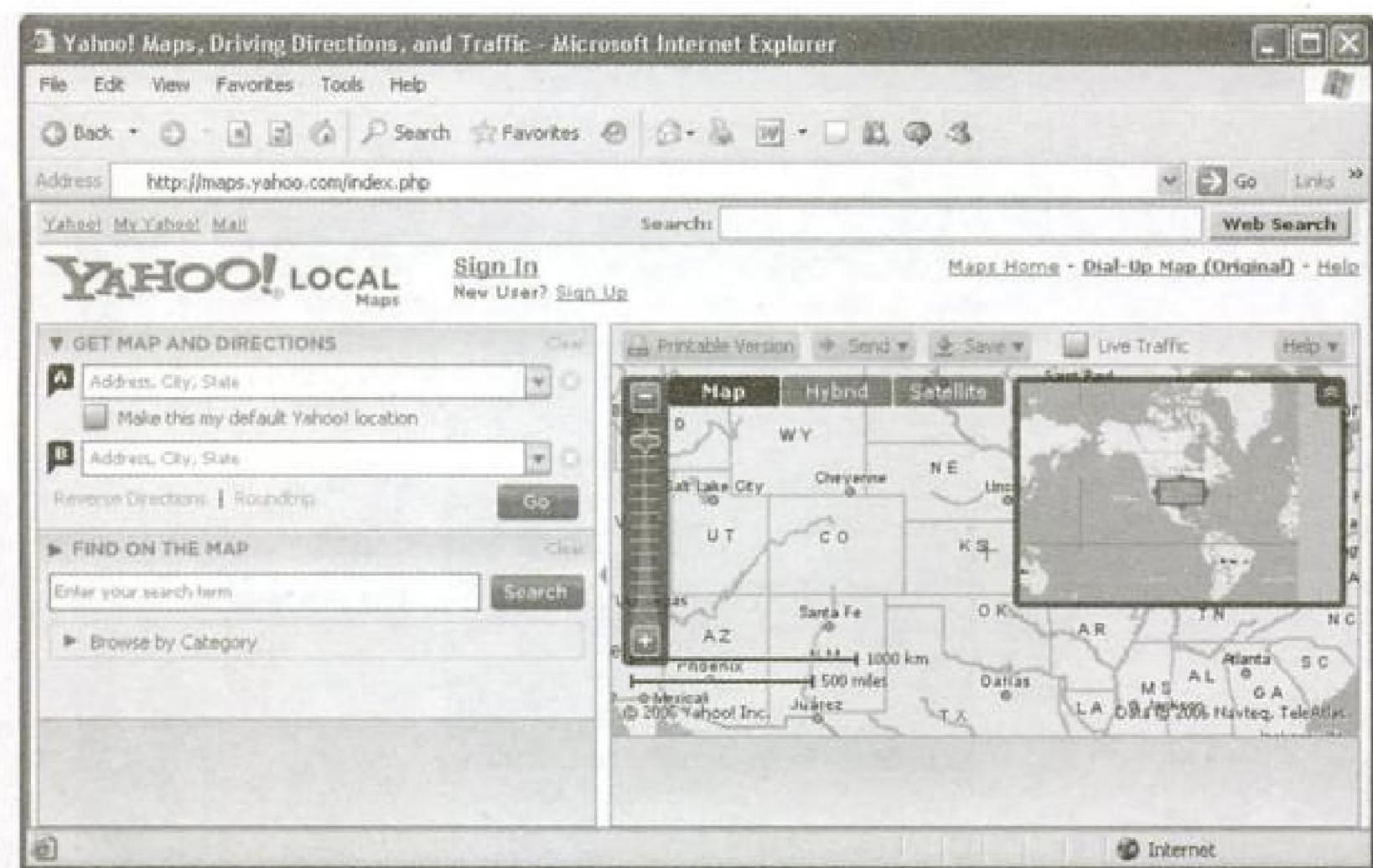


**FIGURE 1-2** The University of Minnesota Home page

Static Web pages are, well, static. They can display data just fine, as well as text and images. But there's nothing really going on there—the user can't interact with anything. PHP changes all that by making those Web pages come alive—things start happening on the server side. Unlike languages like JavaScript, which work in the browser and don't create any lasting effects (JavaScript can't write files, and it can't work with data on the server), languages that execute on the server can be used as the basis of true Web applications. Users will be able to open your pages, seeing everything they'd expect from a full-fledged application, including everything from text fields they can enter text in to tables full of data you create on the fly, from retrieving data from databases to fluidly creating graphics—everything you might see on the most professional interactive Web application is now within your grasp. That's the name of the PHP game—being able to respond to the user dynamically, on the fly.

## Getting PHP

This book will use PHP 5.2.0, whose official Web site is [www.php.net](http://www.php.net) (PHP missed its chance for [php.com](http://php.com)—that's now the Parents Helping Parents Web site). You're going to need PHP to work with this book, and that's going to mean getting access to a Web server that runs it.



**FIGURE 1-3** The Yahoo Maps page

As of this writing, PHP 6 is in the works. There are many minor changes coming up in PHP 6—none of which should stop your PHP 5 code from running—and a major change: support for Unicode. Unicode ([www.unicode.org](http://www.unicode.org)) is a character set that's designed to encompass many of the world's languages, unlike PHP today. As I'm writing this, support for Unicode in PHP 6 is transparent—that is, your PHP 5 code will run fine. At most, you'll be required to place a directive at the beginning of your scripts indicating whether you want Unicode support turned on or off. And then you'll have the full Unicode character set—everything from Arabic to Cherokee to Tibetan.

## PHP on the Internet

In fact, your Internet service provider (ISP) quite probably already supports PHP—you can ask your support staff, or you can try to upload and run a PHP file of the kind we'll be developing shortly.

You can also open a command prompt for your server and check on PHP that way. You can open a command-prompt window connected to your server using various utilities—Telnet, SSH, or SSH2 (you won't need these applications in this book, so don't worry if you don't have them). Windows, for example, comes with a Telnet utility built in—just enter C:\>telnet followed by your ISP's name (e.g., [phpbigserver.com](http://phpbigserver.com)), and press ENTER.

Once you have a command prompt open for your server, you can check if PHP is installed with the -v option, which gives the version of PHP if it can be reached (note that

I'm going to use % as a generic command prompt in this book, standing for the command prompt in Windows, Linux, and so forth):

```
*php -v
```

If PHP is installed and accessible, you'll see the PHP version and date displayed like this:

```
*php -v
PHP 5.2.0 (cli) (built: Nov 2 2006 11:57:36)
Copyright (c) 1997-2006 The PHP Group
Zend Engine v2.2.0, Copyright (c) 1998-2006 Zend Technologies
```

The other way to test if you have PHP installed is to upload a PHP script and see if it runs. To do that, check out the sample scripts you'll see in this chapter. Upload such a test script to your ISP, and see if you can access it in your browser.

Note that on Unix-based systems, you have to first explicitly set the permission of the script to *executable*, and most FTP utilities (File Transfer Protocol utilities, including the one built into Windows and accessible by typing **ftp** at a DOS command prompt) will let you set that permission. If you have a PHP script on a Unix-based server, you should set its permissions to Owner: Execute, Read, and Write; Group: Execute and Read; and Public: Execute and Read. Numerically, that works out to a permission setting of 755 for PHP scripts on Unix-based servers.

---

**Tip** If you want a list of ISPs that already run PHP, take a look at [www.php.net/links.php#hosts](http://www.php.net/links.php#hosts).

## PHP on Your Local Machine

It's a good idea to install PHP on your local machine if you want to do any substantial PHP development (and who doesn't?). That way, you can examine your PHP pages in your Web browser on your machine immediately after you edit them locally. That speeds things up tremendously, and it cuts the development cycle in half. But to do things this way, you'll need PHP installed locally.

Some operating systems, such as Linux, and some versions of Unix, come with PHP already installed. You can test that out with the **php -v** command, as described in the preceding section. If you get a response showing PHP's creation date, you're all set. In Windows, however, you're going to have to install PHP yourself.

There are prebuilt "binary" versions ready for download and immediate installation for a number of operating systems: Windows, Mac OS X, Novell NetWare, OS/2, RISC OS, SGI IRIX 6.5.x, AS/400. You can find the binary installation package for Windows at [www.php.net/downloads.php](http://www.php.net/downloads.php), along with links to the binaries for the other operating systems mentioned.

You should be able to install PHP on your machine simply by using the appropriate binary file. For example, in Windows, you'll download a Windows installer (.msi) file, and double-click it. Answer the questions it asks, and you're in.

Before installation, you'll have to decide on the Web server you want to use. The installer will currently set up Microsoft Internet Information Server (IIS), Apache, Xitami, and Sambar Server; if you are using a different Web server you'll need to configure it manually using the directions you can find at [www.php.net/download-docs.php](http://www.php.net/download-docs.php) (download the whole PHP documentation—it includes a section on installation).



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

If you're using a local Web server, such as Windows IIS, that means placing `phpinfo.php` in `C:\Inetpub\wwwroot`, or a subdirectory of `wwwroot`, and then opening your browser using an URL like `http://localhost/phpinfo.php` or `http://localhost/subdirname/phpinfo.php` if you've placed `phpinfo.php` in a subdirectory of `wwwroot`.

The code that accompanies this book is stored in the folders `ch01` for Chapter 1, `ch02` for Chapter 2, and so on. So you'll find `phpinfo.php` in the `ch01` folder.

## Running Your First PHP Page

Make sure the Web server is running (if you're using IIS in Windows, it's always running) and navigate to `phpinfo.php` in your browser, as shown in Figure 1-5. Congratulations, you're running your first PHP page.

You can see the results in that figure—the call to `phpinfo()` returned an HTML table containing information about your PHP installation. What actually happened is that the `phpinfo()` function returned the text holding the HTML table, and that table is inserted into the resulting page, which you see in Figure 1-5.

### Some Troubleshooting

What if it doesn't work? What if you don't see the display in Figure 1-5? Unfortunately, there are many things that could go wrong, especially with a local installation of PHP. Don't panic—the problem can be fixed. It'll just take a little time.

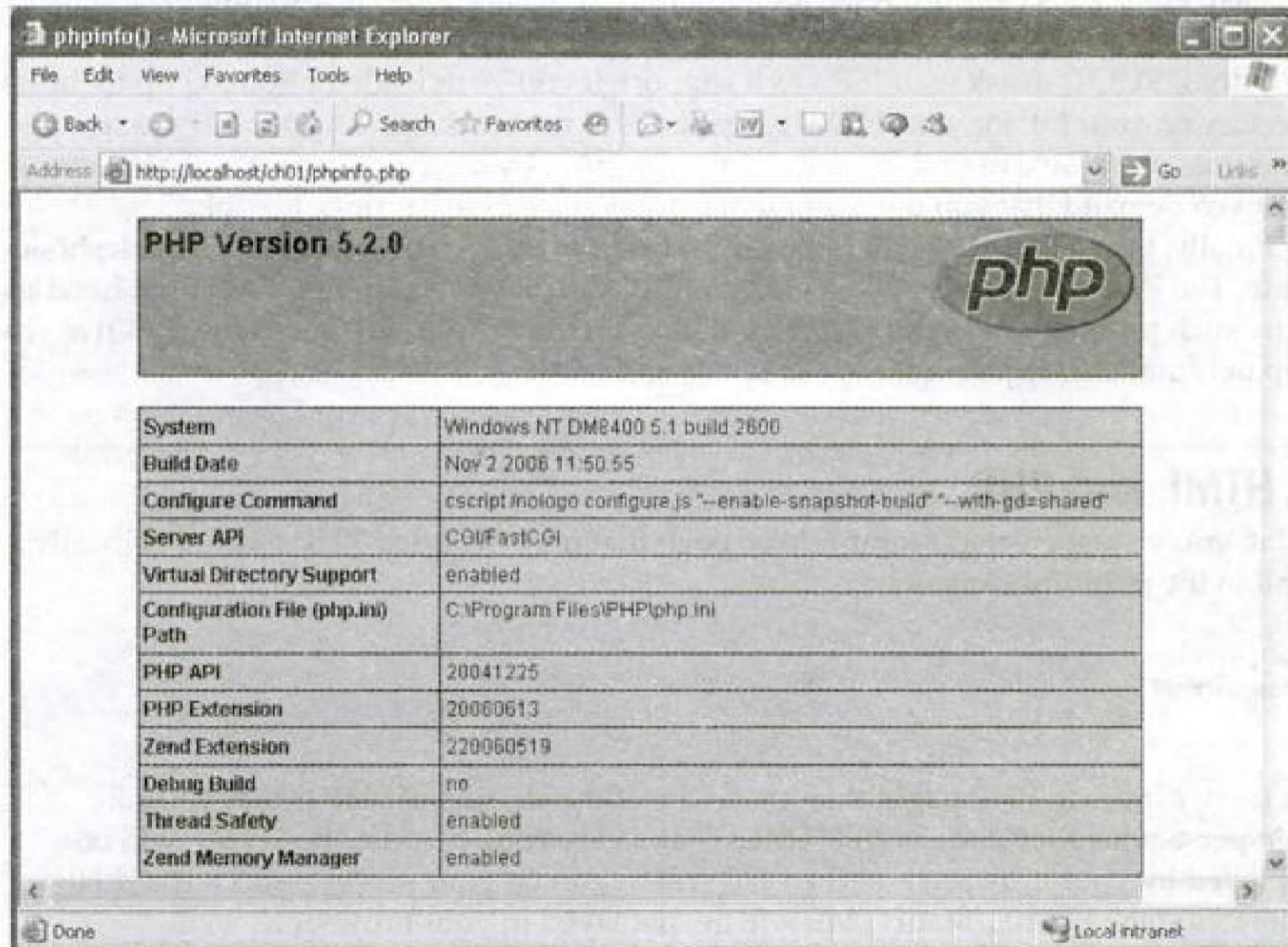


FIGURE 1-5 Calling `phpinfo()`



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

the image `php-med-trans-light.gif` in a page named `phpimage.php` (put the image file in the same directory as `phpimage.php` on your server):

```
<html>
  <head>
    <title>
      Using PHP and HTML together
    </title>
  </head>

  <body>
    <h1>
      Using PHP and HTML together
    </h1>
    Here is your PHP info:
    <br>
    <br>
    <?php
      phpinfo();
    ?>
    <img src='php-med-trans-light.gif'>
  </body>
</html>
```

And you can see the results—the logo is at the lower right—in Figure 1-7.

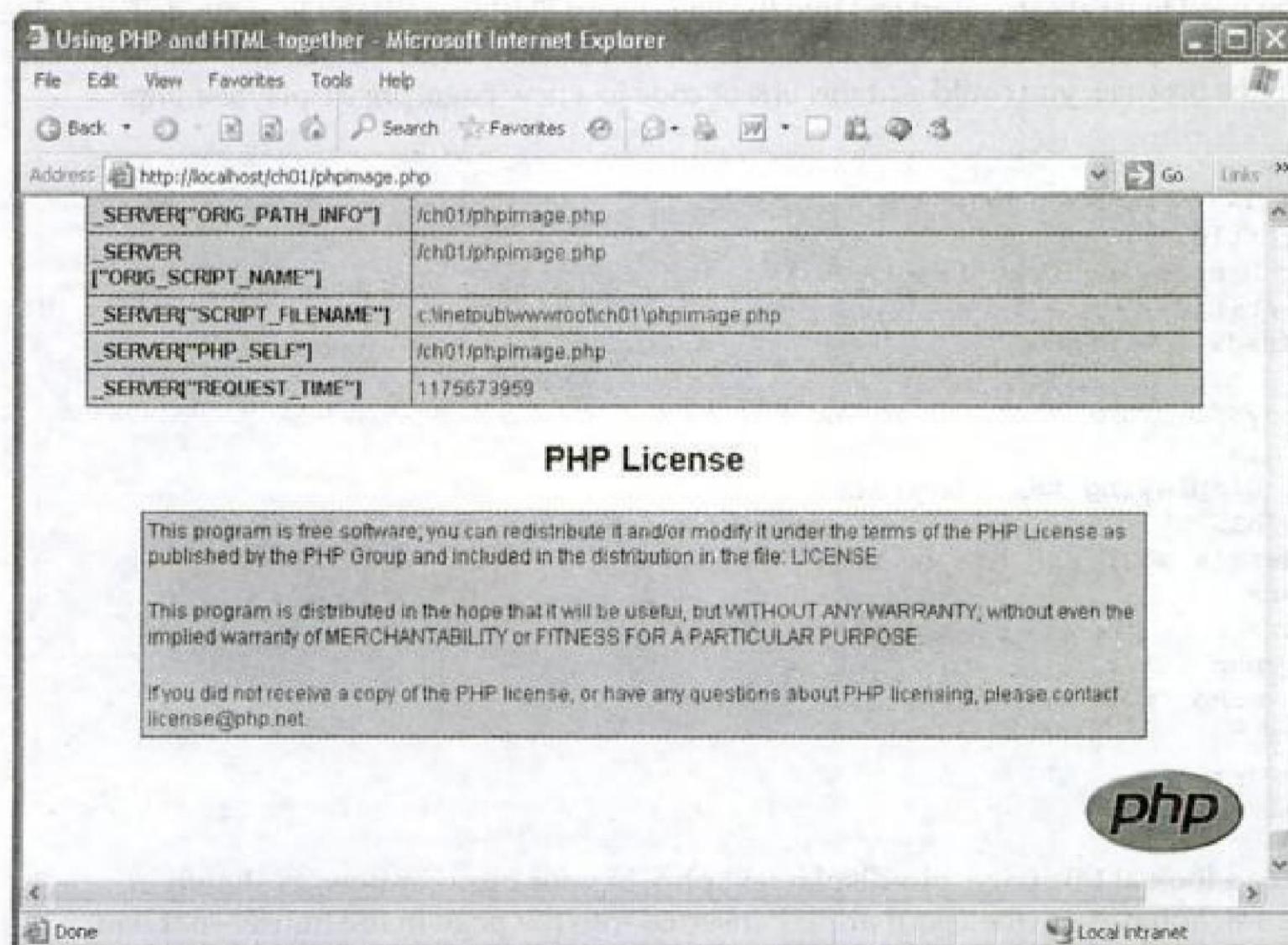


FIGURE 1-7 Using PHP logos



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

So if you want to skip to the next line in your displayed output, you have to insert the correct HTML, <br>, into your displayed text.

## More Echo Power

You can also run PHP from the command line, in fact, simply by using the `php` command. Here's what you'd see if you ran `phpdisplayhtml.php` from the command line (remember that % stands for a generic command-line prompt in this book):

```
%php phpdisplayhtml.php
<html>
  <head>
    <title>
      Displaying text from PHP
    </title>
  </head>

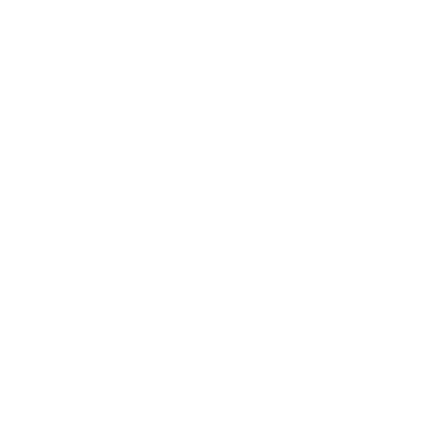
  <body>
    <h1>
      Displaying text from PHP
    </h1>
    Here's what PHP has to say:
    <br>
    <br>
    <i>Welcome</i><br><u>to</u><br><b>PHP</b>.  </body>
</html>
%
```

Note that the HTML wasn't interpreted as HTML here; it was simply printed out as plain text. If you want to skip to the next line in this case, you should use the `\n` control character, which PHP will interpret as a newline character—this will display all three words on their own lines:

```
echo "Welcome\n";
echo "to\n";
echo "PHP.";
```

Here is a sampling of the control characters available in PHP:

- `\n` Newline character
- `\r` Carriage return
- `\t` Tab
- `\\"` Displays a \
- `\$` Displays a \$
- `\"` Displays a "
- `\0 to \777` Displays a character corresponding to a hexadecimal (base 8) code
- `\x0 to \xFF` Displays a character corresponding to a hexadecimal (base 16) code



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.