



Ordering Information: [Internet & World Wide Web How to Program, 2/e](#) & [The Complete Internet & World Wide Web Training Course, 2/e](#)

- View the complete [Table of Contents](#)
- Read the [Preface](#)
- Download the [Code Examples](#)

To view all the Deitel products and services available, visit the Deitel Kiosk on InformIT at www.informIT.com/deitel.

To follow the Deitel publishing program, sign-up now for the *DEITEL™ BUZZ ONLINE* e-mail newsletter at www.deitel.com/newsletter/subscribeinformIT.html
To learn more about Deitel instructor-led corporate training delivered at your location, visit www.deitel.com/training or contact Christi Kelsey at (978) 461-5880 or e-mail: christi.kelsey@deitel.net.

Note from the Authors: This article is an excerpt from Chapter 29, Section 29.2 of *Internet & World Wide Web How to Program, 2/e*. This article introduces the reader to the PHP scripting language. Readers should be familiar with basic Web development in HTML or XHTML, basic programming and placing documentation in a Web server directory. The code examples included in this article show readers examples using the DEITEL™ signature *LIVE-CODE™ Approach*, which presents all concepts in the context of complete, working programs followed by the screen shots of the actual inputs and outputs.

Special Note: XHTML is a W3C technology designed to replace HTML. Readers comfortable with HTML will not have difficulty understanding the XHTML used in this article.

29.2 Introduction to PHP

PHP, or *PHP Hypertext Preprocessor*, is quickly becoming one of the most popular server-side scripting languages for creating dynamic Web pages. PHP was created in 1994 by Rasmus Lerdorf (who currently works for Linuxcare Inc. as a Senior Open-Source Researcher) to track users at his Web site. In 1995, Lerdorf released it as a package called the “Personal Home Page Tools.” PHP 2 featured built-in database support and form handling. In 1997, PHP 3 was released, featuring a rewritten parser, which substantially increased performance and led to an explosion in PHP use. It is estimated that over six million domains now use PHP. The release of PHP 4, which features the new *Zend Engine* and is much faster and more powerful than its predecessor, should further increase PHP’s popularity. More information about the Zend engine can be found at www.zend.com.

PHP is an *open-source* technology that is supported by a large community of users and developers. Open source software provides developers with access to the software’s source code and free redistribution rights. PHP is platform independent; implementations exist for all major UNIX, Linux and Windows operating systems. PHP also provides support for a large number of databases, including MySQL.

When the World Wide Web and Web browsers were introduced, the Internet began to achieve widespread popularity. This greatly increased the volume of requests for information from Web servers. The power of the Web resides not only in serving content to users, but also in responding to requests from users and generating Web pages with dynamic content. It became evident that the degree of interactivity between the user and the server would be crucial. While other languages can perform this function as well, PHP was written specifically for interacting with the Web.

PHP code is embedded directly into XHTML documents. This allows the document author to write XHTML in a clear, concise manner, without having to use multiple **print** statements, as is necessary with other Common Gateway Interface (CGI)-based languages. Figure 29.1 presents a simple PHP program that displays a welcome message.

PHP scripts usually end with **.php**, although a Web server can be configured to handle other file extensions. To run a PHP script, PHP must first be installed on your system. Visit

```
ftp://ftp.prenhall.com/pub/deitel/setup_instructions/  
iw32e-setup.html
```

for PHP installation and configuration instructions. Although PHP can be used from the command line, a Web server is necessary to take full advantage of the scripting language.

In PHP, code is inserted between the scripting delimiters **<?php** and **?>**. PHP code can be placed anywhere in XHTML markup, as long as the code is enclosed in these scripting delimiters. Line 8 declares variable **\$name** and assigns to it the string **"Paul"**. All variables are preceded by the *\$ special symbol* and are created the first time they are encountered by the PHP interpreter. PHP statements are terminated with a *semicolon* (**;**).

Common Programming Error 29.1



Failing to precede a variable name with a *\$* is a syntax error.

Common Programming Error 29.2



Variable names in PHP are case sensitive. Failure to use the proper mixture of case is a syntax error.

```

1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3
4  <!-- Fig. 29.1: first.php -->
5  <!-- Our first PHP script -->
6
7  <?php
8     $name = "Paul";    // declaration
9  ?>
10
11 <html xmlns = "http://www.w3.org/1999/xhtml">
12   <head>
13     <title>A simple PHP document</title>
14   </head>
15
16   <body style = "font-size: 2em">
17     <p>
18       <strong>
19
20         <!-- print variable name's value -->
21         Welcome to PHP, <?php print( "$name" ); ?>!
22       </strong>
23     </p>
24   </body>
25 </html>

```

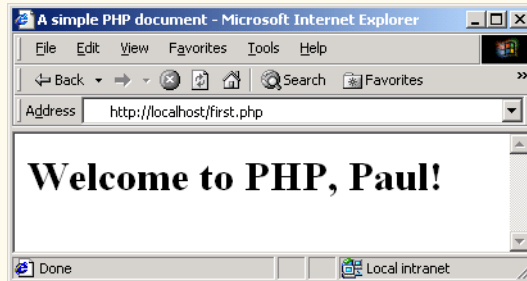


Fig. 29.1 Simple PHP program.



Common Programming Error 29.3

Forgetting to terminate a statement with a semicolon (;) is a syntax error.

Line 8 contains a *single-line comment*, which begins with two *forward slashes* (`//`). Text to the right of the slashes is ignored by the interpreter. Comments can also begin with the pound sign (`#`). Multiline comments begin with delimiter `/*` and end with delimiter `*/`.

Line 21 outputs the value of variable `$name` by calling function `print`. The actual value of `$name` is printed, instead of `"$name"`. When a variable is encountered inside a double-quoted (`" "`) string, PHP *interpolates* the variable. In other words, PHP inserts the variable's value where the variable name appears in the string. Thus, variable `$name` is

replaced by **Paul** for printing purposes. PHP variables are "*multitype*", meaning that they can contain different types of data (e.g., *integers*, *doubles* or *strings*) at different times. Figure 29.2 introduces these data types.

Data type	Description
Integer	Whole numbers (i.e., numbers without a decimal point).
Double	Real numbers (i.e., numbers containing a decimal point).
String	Text enclosed in either single (' ') or double (" ") quotes.
Boolean	True or false.
Array	Group of elements of the same type.
Object	Group of associated data and methods.
Resource	An external data source.
Null	No value.

Fig. 29.2 PHP data types.