
Understanding PHP

PHP is an open source server-side scripting language, which can be embedded inside HTML as a clever means of providing dynamic Web pages. C and Perl developers will find it particularly intuitive to learn.

**By Paul Oldham
Technical Consultant**

PHP, the PHP Hypertext Processor, is an open source server-side scripting language for Web servers, which provides a real alternative to ASP, ColdFusion, Mod_Perl or JSP if your aim is to provide dynamic Web pages. Dynamic Web pages are pages which interact with the user, so that each user visiting the page sees customized information - which may vary each time and which may be based on a form they've just filled in, or on information extracted from a database or some other external source. Typical applications include e-commerce, online newspapers, visitors' books, ticketing systems, project management, and other groupware projects. The traditional way to produce this type of dynamic page is via CGI scripts, but these are separate programs which must be executed as a new process for each page hit, so they scale badly and rapidly become memory and processor hogs as server load increases.

PHP solves this problem by becoming a part of the Web server, essentially extending the functionality of the server itself, so that the server can do the processing without having to spawn extra processes. It's not alone in doing this, but unlike most other scripting languages for Web page development PHP also offers excellent connectivity to most of the databases in use today. Perhaps the greatest advantage of PHP, when compared to other scripting languages such as ASP or ColdFusion, is that it is open source and cross-platform. PHP's natural home is on Linux servers running Apache server software, but it runs equally well on any other Unix or Windows platform, and can be used with other Web servers.

PHP started life as a Perl program written by Rasmus Lerdorf to track visitors to his online résumé. It was then rewritten in C and was extended to include support for database access. From these simple beginnings the open source community has expanded and developed PHP into a powerful server-side scripting language.

Which Version?

An earlier version of PHP that was released publicly was called PHP/FI 2.0, and you may still find this on some ISPs' servers and Linux distributions, but it wasn't until PHP3 that the language really came into its own. PHP3 has been around since June 1998 and is still the most widely used release. However, we are currently in a transition period, as PHP4 came out of beta in May this year. PHP4 uses a new engine called Zend and offers greatly improved performance and increased functionality, so we can expect developers and ISPs to start migrating to that release soon. Code written in PHP3 is compatible with the PHP4 engine with a few minor exceptions, and in this article we'll consider both releases, highlighting any differences as we go.

In July 2000 Netcraft's ongoing survey of Web servers reported that 3,121,918 domains at 598,213 IP addresses had PHP available on their Web server. Of course, such wide availability doesn't mean it has actually been used on all those servers, but organisations such as Mitsubishi, Redhat, livebid.amazon.com, Der Spiegel, MP3-Lycos, Ericsson, Sprint Canada, xoom.com and NASA are all actively using PHP today.

Writing PHP

What sets PHP apart from its origins in Perl is the ability to embed PHP code inside HTML, allowing you to mix HTML and PHP in one source file rather than having to crank out the HTML from within Perl. For example, a simple PHP script to show today's date is shown in Figure 1. If you're familiar with HTML then the top and bottom of this code will look very familiar - it's just standard HTML. The

PHP code is embedded within the HTML by bracketing it with “<?php” and “?>”. The code consists of one line, a call to the PHP function “date”, which returns a date in the format we’ve specified as a string, which is then output as part of the HTML using the “echo” function. This ability to embed PHP inside HTML means that you can continue to write your Web pages as you do now, and then embed PHP into them to provide the dynamic elements.

This is perhaps a good point at which to highlight what is meant by a server-side language. The PHP code is executed on the server, not the browser. The source of the page the browser receives back from the server is shown in Figure 2. You can see that the PHP has been processed by the server, so no special software is needed on the client to view pages with PHP elements - the intelligence lies at the server end. This means you can use PHP with any browser - even simple, text-only ones like Lynx and w3m, or browsers in PDAs or set-top boxes. So long as the browser understands HTML it can display pages written in PHP.

PHP Syntax

In view of where it started, it’s not surprising that PHP shares syntactical ideas with Perl and C, and if you’ve programmed in either of these or you’ve done any Unix shell programming, then PHP should be easy to pick up. PHP uses loosely typed variables which don’t need to be declared before first use, and evaluates them according to context, so you can add a numeric string to a numeric to give a numeric, and then manipulate the result as a string. The PHP online manual gives a nice example of how a variable’s type can change due to context. This is shown in Figure 3.

One of the joys of PHP is that some variables are already set for you in any PHP script. For example, the variable \$HTTP_USER_AGENT will contain a string identifying the browser which requested the page. So, for example, if you’re using Netscape 4.7 under NT4 to request the page, \$HTTP_USER_AGENT will contain the string “Mozilla/4.7 [en] (WinNT; I)”. So if you need to generate HTML which exploits browser-specific features it’s easy to do. Similarly, any cookies you have set will automatically appear as variables in your script without you having to write any code to extract them. As well as simple variables PHP also lets you create arrays which act both like hash tables (associative arrays) and indexed arrays (vectors), and these arrays can be multi-dimensional.

PHP has a full range of comparative, logical and bitwise operators, auto increment and decrement, and assignment operators. It has all the control structures you might expect from a procedural language, including if/else, switch/case, for, while (test before) and do (test after). You can write your own functions or use the extensive range of functions provided as standard. Arguments can be parsed by value or reference, and functions can return a value - or multiple values if you use an array as the return variable. Variables are local to a function unless declared global.

PHP also supports include files, both static and dynamic, allowing common code such as custom function libraries and HTML fragments to be embedded in your code. This is all very good news from the developer’s viewpoint, allowing code re-use wherever possible. It also means you can separate the PHP code from the HTML if you employ different people for Web design and Web programming. PHP3 includes support for object oriented programming, but there is a problem with polymorphism: once a parent’s function is overridden by a child it can no longer be used. This is resolved in PHP4, which provides expanded functionality and new features for object oriented programming, and for building classes and objects.

Forms

One of the most powerful features of PHP is the ease with which you can handle forms. If you make your PHP script the action of the form then all the fields in the form appear as variables in your script automatically, ready for you to use. To take a simple example, suppose you have a page containing a form such as the one at the top of Figure 4. When the user clicks on the submit button the page action.php3 will be invoked to process that form. In action.php3 you might have the line of code at the bottom of Figure 4. As you can see, the input fields “age”

```
<html> <head> <body>
<p>Today is
  <?php      echo date
    ("F dS, Y"); ?>
</body> </html>
```

Figure 1 - A simple PHP script to show today’s date.

```
<html> <head> <body>
<p>Today is
  August 30th 2000
</body> </html>
```

Figure 2 - The source of the page the browser receives back from the server.

```

$foo = "0"; // $foo is string (ASCII 48)
$foo++;    // $foo is now the string "1" (ASCII 49)
$foo += 1; // $foo is now an integer (2)
$foo = $foo + 1.3; // $foo is now a double (3.3)
$foo = 5 + "10 Little Piggies"; // $foo is an integer (15)

```

Figure 3 - A variable's type can change depending on context.

and "name" have automatically become variables of the script action.php3, with names "\$name" and "\$age" ready for you to do what you like with them (in this case echoing them as part of the line of HTML).

Functions

The real strength of PHP lies in the extensive range of built-in functions. Those available include functions to:

- Access a wide variety of databases (see below).
- Interrogate the Web server on which the code is running.
- Manipulate arrays.
- Access a full range of mathematical functions.
- Perform arbitrary precision mathematics (log, trig etc).
- Get the date and time and display them in a variety of formats.
- Browse directory trees.
- Execute external programs on the server and receive the results back.
- Perform a wide variety of operations on the server file system.
- Set cookies.
- Send mail.
- Perform network functions such as "Get the Internet host name corresponding to a given IP address".
- Use Perl-compatible or POSIX-extended regular expressions.
- Use System V semaphores and shared memory o Manipulate strings.
- Test a variable's current type.
- Use WDDX to easily exchange data with other applications over the Web.
- Gzip and gunzip files.

Creating powerful Web applications inevitably means storing and manipulating a lot of data, so the ability to access databases is vital. PHP includes support, via function calls, for a wide variety of popular databases, including Adabase D, dBase, Unix dbm, filePro, Interbase, Informix, Microsoft SQL Server, mSQL, MySQL, Oracle, Sybase, PostgreSQL, and Solid. In addition, any database which supports ODBC, such as Microsoft Access, can be used. If you're trying to create an application which isn't database-specific then there are a number of open source abstraction layers available, including Metabase and phpDB, which will let you do this painlessly.

Optional Extras

In addition to the standard functions and the database access functions, PHP has a number of additional functions which only work if you have the appropriate library loaded on your server.

```

<form action="action.-
php3" method="POST">
Your name: <input
type=text name=name>
Your age: <input
type=text name=age>
<input type=submit>
</form>

action.php3:
<p>Hello <?php echo
$name?>. You are <?php

```

Figure 4 - Forms in PHP - see main text.

Obtaining PHP

The home of PHP on the Web is www.php.net, and Zend Technologies have their own site at www.zend.com. These two sites between them contain links to an ever increasing list of other sites devoted to PHP. PHP is open source, so you can simply download PHP from any one of the many mirror sites. PHP also comes as a package in some Linux distributions.

If the target audience for your PHP-generated pages is the Web rather than your company intranet, then a growing number of ISPs can provide you with Web hosting on servers which support PHP, and an SQL database back-end, usually MySQL.

The gd library allows you to create images on the fly. The format of images you are able to manipulate depends on the version of gd you install, and any other libraries gd might need to access those image formats. Older versions of gd support GIF format images, newer ones PNG. It is also possible to generate images in JPEG format. This ability to generate images means you can create entities such as labelled buttons from within PHP, rather than having to rely on a pre-defined set of images. The ClibPDF library will let you create PDF files from within PHP, as will Thomas Merz's PDF library. The former is faster and creates smaller documents, but although it is free for private, non-profit use, a commercial licence is required for "for-profit" applications.

James Clark's expat toolkit lets you parse XML documents, create XML parsers and then define handlers for different XML events. Other libraries allow you to encrypt and decrypt data in a wide variety of block algorithms, use hash algorithms such as MD5, SHA1 and GOST, access mail on an IMAP mail server, do spell checks on words in any string, access LDAP servers, and gain access to SNMP (available as standard under Windows NT).

New Features In PHP4

PHP4 is significantly faster than PHP3. This improvement in performance is particularly noticeable with larger and more complex scripts, and is the result of the PHP engine having been totally rewritten by Andi Gutmans and Zeev Suraski. The new Zend engine uses a much more efficient "compile-then-execute" method, instead of the "execute-while-parsing" model used by PHP3. The authors claim the resulting engine is up to fifty times faster than the PHP3 engine. New features in PHP4 include:

- Extended API module.
- Generalized build process under Unix.
- Generic Web server interface that also supports multi-threaded Web servers.
- Improved syntax highlighter.
- Native http session support.
- Output buffering support.
- A more powerful configuration system.
- Reference counting.
- ftp support.

PHP4 also provides COM/DCOM support (on Windows only), allowing you to seamlessly access COM objects and instantiate them. PHP4 also has a number of additional optional functions which, with the right libraries, extend PHP to allow you to process credit cards and other financial transactions using Verisign Payment Services, and to handle Cybercash payments.

Further Reading

PHP comes with a well-written online manual, and if you're a experienced programmer who's written in Perl or C before you may find that sufficient for your needs. If you want a more gentle introduction then a good primer is *PHP3 Programming Browser-Based Applications* by David Medinets, published by McGraw Hill, ISBN 0071353429.

For PHP4 the definitive book is the recently published *Web Application Development With PHP* by Tobias Ratschiller and Till Gerken, published by New Riders Publishing, ISBN 0735709971.

Future Developments

The developers of the Zend engine have now set up their own company, Zend Technologies, as a vehicle for producing more PHP-related products. They have two products in the pipeline, both due to be released shortly:

- The Zend Cache saves any PHP script it has compiled in a cache, so that subsequent hits on the same script don't need compiling before running. This reduces server disk accesses and overall execution time.
- The Zend Compiler allows you to compile your PHP scripts before distributing them. By saving the code in a closed Zend Intermediate Code format, the compiler allows you to protect source code from copyright infringement.

PCNA

Copyright ITP, 2000

Recent Reviews from [Tech Support Alert](#)

[Reviews of the Best Windows Backup Software](#)

In this detailed comparative review, we checked out eighteen backup software utilities designed for home or SOHO use. Many of the products reviewed were disappointing. However 6 products passed our tests with flying colors and 2 of these were so impressive, they were awarded our "Editor's Choice."

[Suppliers of Cheap Inkjet Printer Cartridges Reviewed and Rated](#)

With hundreds of companies all claiming to have the "*cheapest and best inkjet printer cartridges*," our editors decided to put their claims to the test. Not unexpectedly, many suppliers flunked but we did manage to come up with a number of web sites that sell good quality inkjet printer cartridges at heavily discounted prices.

[The Best Anti Trojan Software](#)

Our editors took a close look at the 6 leading anti-trojan/trojan remover software utilities. Unfortunately, they found only 2 products that were effective in their ability to detect and remove dangerous modern polymorphic and process injecting trojans.

[The 46 Best Ever Freeware Utilities](#)

This is our Editor, Ian "Gizmo" Richards, personal selection of the best freeware utilities. He's hunted down some real gems, many of which perform better than expensive commercial products.