



e-mail it!

Introduction to PHP

Finally, the perfect language for dynamic content and database interaction

Joe "Zonker" Brockmeier

Senior Editor, User Friendly Media

December 2000

Joe Brockmeier presents a brief introduction to the PHP scripting language with a discussion of PHP's origins, capabilities, and the platforms it's available on. A simple PHP script example highlights basic syntax and usage.

If you work with Web-based development, you've probably heard about PHP. You might not know exactly what it is, how it works, or why it's so hot, but you do know it's time to find out more about it. So here's a quick intro to the basic concepts that underlie PHP.

A bit of background

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994. It has been picking up steam ever since and is now at version 4.0.3 with numerous improvements and refinements over the original release.

PHP is a scripting language that is embedded in HTML and interpreted by the server. It can be used to manage dynamic content, work with databases, handle session tracking, and even build entire e-commerce sites. It works well with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

Dynamic vs. static content

What's so hot about dynamic content? Let's say you're managing an e-commerce site with 10 products. It's not that difficult to hand-code ten static product pages with all the requisite information, forms and such, provided your products don't change often and you don't anticipate much growth. However, let's say you add ten more products this month, and then more next month, and occasionally prices change or you want to change the look and feel of your site. Then you're stuck re-coding dozens, maybe hundreds, of static pages by hand.

On the other hand, let's say you start by creating one page that is called product.php. Instead of holding static information, it's coded to pull information out of your product database and build a page dynamically. You then have one meta page that can serve up one or one hundred or even a hundred thousand unique pages based on information stored in a database. Rather than requiring a Web master to spend an entire day doing nothing but monkey-work updating static Web pages, the information can now be updated at the same time the information is changed in the company database. You eliminate the headache-inducing lag between the time information is changed in the database and the time it

Contents:

[Background](#)[Dynamic vs. static content](#)[Platforms](#)[Licensing and use](#)[Using PHP](#)[Summary](#)[Resources](#)[About the author](#)

makes its way onto the Web site.

At the risk of generalizing, for work on the Web, PHP is a great way to go. It's not the only way to go; Perl, Java, JavaScript, ASP, Python, Tcl, CGIs, and probably dozens of other ways are available for generating dynamic content. However, PHP has the benefit of being designed just for Web-based problems and of being an open source project.

If you're looking for a programming language for a word processor or 3D game, then PHP probably isn't the way to go. If you need to run a Web site with dynamic content, database interaction, and e-commerce, read on, because PHP is going to be very helpful indeed.

Platforms available for PHP

The most common PHP installation is probably the PHP module running with Apache on Linux or a UNIX-variant. But if you're using something else, don't worry. PHP works on Windows NT and 9x, as well as with a number of other Web servers. You'll find more documentation floating around on the Web that's specific to the Apache/Linux/PHP combo, but it isn't by any means the only platform that PHP is supported on.

Licensing and use

How much might you pay for a full-featured embedded Web scripting language? How about nothing? PHP is an open source project, so there are no licensing fees or restrictions on use. You could run a small, non-profit site, or you could run a billion-dollar e-commerce site with PHP, and the cost is the same: zero. Not only that, but the PHP code is available in case you want or need to tweak it.

PHP is not licensed under the GPL, but its own license permits redistribution of code and/or binaries.

Using PHP

Okay, so now you're convinced that you want to actually try PHP out. We'll walk through a few simple examples so you can get your feet wet. Keep in mind that this is by no means an exhaustive look at PHP, just a quick starter.

"Hello, World!"

To get a feel for PHP, let's look at some very simple PHP scripts. Since "Hello, World!" is an obligatory example, we'll produce a friendly little "Hello, World!" script.

As mentioned earlier, PHP is embedded in HTML. (You could have a file that contains almost no HTML, but usually it's a mixture.) That means that in amongst your normal HTML (or XHTML if you're cutting-edge) you'll have PHP statements like this:

```
<body bgcolor="white">
    <strong>How to say "Hello, World!"</strong>
        <?php echo "Hello, World!";?>
    <br>
    Simple, huh?
</body>
```

Simple, right? Just an "echo" statement, and that's it. Of course, that on its own isn't terribly useful. But it does teach us something about the language. (By the way, if you examine the HTML output, you'll notice that the PHP code is not present in the file sent from the server to your Web browser. All of the PHP present in the Web page is processed and stripped from the page; the only thing returned to the

client from the Web server is pure HTML output.)

Printing date and time in a Web page

Now we'll do something a little more useful. This example will print out the date and time in a Web page.

```
<body bgcolor="white">
    <strong>An Example of PHP in Action</strong>

<?php echo "The Current Date and Time is: <br>";
    echo date("g:i A l, F j Y.");?>

// g = the hour, in 12-hour format
// i = minutes
// A = print AM or PM, depending...
// l = print the day of the week
// F = print the month
// j = print the day of the month
// Y = print the year - all four digits
```

This code produces the output:

```
The Current Date and Time is:
11:00 AM Friday, October 20 2000.
```

Notice the blend of PHP and HTML here. I'll assume that you can already read HTML, so I'll only explain the PHP code. You can find a complete PHP reference online at PHP.net (see [Resources](#)).

The PHP code begins with the tag `<?php` and ends with `?>`. This tells the server that everything between `<?php` and `?>` needs to be parsed for PHP instructions, and that if they're found, they need to be executed. Note that when your document is processed and served, it will be received by the client as plain HTML. Someone browsing your site will not see any of your PHP instructions, unless you've made an error and the server spits them out as-is instead of processing them first.

Regular HTML tags within the `<?php` and `?>` will be processed normally. Note that the simple script above contains a `
` line-break tag. PHP wouldn't be very useful if you couldn't include HTML formatting as well.

If you're going to be working with others, or if you're just plain forgetful like me, you'll want to comment your code as well. The `//` characters indicate a comment, which the server will not process or pass on to the client, unlike comments in HTML. If you include a standard `<!-- comment -->` within the `<?php` and `?>` tags, it is likely to cause an error when parsed by the server. Obviously, you probably wouldn't comment your code quite so much as I have above for such a basic function, but it makes a good example.

Finally, note that each PHP function is enclosed in parentheses and ends with a semicolon, which will seem familiar to fans of C or Perl. It's not uncommon to forget a closing parenthesis or semicolon and

have a number of parse errors just because of a simple typo, so be sure to check them. It's helpful to edit PHP in an editor like Vim or Emacs that is capable of syntax highlighting. This allows you to catch your errors right away.

The date function is just one of the built-in PHP functions. PHP also comes with functions for database connectivity, creating PDF, Shockwave, JPG, GIF, PNG, and other graphics files, sending e-mail, reading and writing files, parsing XML, session handling, talking directly to the browser via HTTP, and many other functions.

PHP also allows the user to define their own functions. This makes PHP a language capable of providing a huge number of solutions via the Web. Rather than just writing everything from scratch, however, be sure to check sites like Zend.com, PHP Wizards, and, of course, Freshmeat to see if what you're trying to do has been done already (see [Resources](#)).

There are a lot of open sourced PHP solutions for serving banners, automating news sites, Web-based e-mail clients, database management, and much more. There's no sense re-inventing the wheel. Instead, start from the foundation that has already been built and customize it into your own solution. If you're just poking around with PHP to learn it and don't have a specific project in mind, these projects are still great examples of what you can do with PHP and serve as great learning resources.

Summary

This brief overview of the PHP scripting language should give you an idea of what PHP is capable of and how PHP is being used. In a future article, I'll walk through accessing a MySQL database to create a dynamic Web product page. Until then, check out the resources on PHP below for further information.

Resources

- [PHP.net](#) is the main PHP Web site.
- [PHP Wizards](#) contains documentation and several popular PHP projects.
- [Thickbook.com](#), by Julie C. Meloni, has a number of very helpful tutorials. Julie is the author of *PHP Essentials*, one of the best PHP books available.
- [PHP Builder](#) is another popular PHP site with tutorials and code libraries.
- [Zend.com](#) contains plenty of useful links and information about PHP. Zend is the PHP Optimizer.
- Check out [Freshmeat](#) to get updates on PHP development, among other things.
- Go to [Perl.com](#) to find out more about another popular scripting language.
- Check out the [MySQL](#) official home page.
- Go to the [PHPMyAdmin](#) site to see PHP and MySQL handled.
- The official [Python](#) site is always useful.
- The [Java Lobby](#) has lots of useful Java info.

About the author

Joe "Zonker" Brockmeier is a contributing editor for *Linux Magazine* and has written *Install, Configure and Customize Slackware Linux* for Prima Publishing. His second book, *DocBook Publishing*, will be published by Prima in early 2001. He can be reached at jbrockmeier@earthlink.net.

What do you think of this article?

Killer! (5)

Good stuff (4)

So-so; not bad (3)

Needs work (2)

Lame! (1)

Comments?

[Privacy](#) [Legal](#) [Contact](#)