

# PHP

## Introduction

Nowadays not only e-commerce but also various kinds of industries and educational institutions seem to seek to build dynamic websites that can handle database and can be customized for each visitor. The technology that helps the implementation of such dynamic features has been making progress. CGI<sup>\*\*1</sup>, ASP, PHP, JSP, J2EE, and ColdFusion are considered to be the most typical technology for this purpose. Among them, I chose PHP as my research topic. The goal of my research is to provide useful information for people who need to choose Web application server, Web server, database, OS, and platform for their application system. Before designing websites, people (system architects, system engineers, and web designers whether or not they are professional) often face a situation in which they need to decide which software to use for the implementation of their requirements and specifications. This paper is mainly intended to provide such people with information about PHP and comparison between PHP and ColdFusion. I hope this will provide some criteria for their decision-making.

## Significance

There have been many Web application servers and script languages: CGI (Perl, C, etc.), ASP, PHP, ColdFusion, WebSphere, WebLogic, JRun, JSP, etc. Among them, PHP has been rapidly spreading since 1994 and has established its popularity. Although Perl/CGI (CGI: Common Gateway Interface<sup>\*1</sup>) programming is still popular and ColdFusion emerged with a powerful combination with DreamWeaver and Flash, PHP is still very popular among web application developers. One reason is that it is public domain software and free. However, there are some other features that support this popularity. Below, I would like to discuss these features of PHP's. And at the end, I would like to compare PHP with ColdFusion.

## Discussion

### I. What Is PHP?

PHP is one of the projects of the Apache Software Foundation. (Refer to <http://www.apache.org/>.) PHP (recursive acronym for "Hypertext Preprocessor") is free and offered under an open source license. (This means that you can use it as you wish. The "license" really is only to cover commercial use of the code. For your own use, you don't have to worry

about “license” at all, so you can download it and use it without any problem.) It is a widely-used general-purpose **scripting language** and **interpreter** that is especially used to create dynamic Web pages. It is commonly used to extract data out of a database and present it on the Web pages. It can be embedded into HTML. The statistics as of September, 2002 shows that PHP is being used in 9,458,364 Domains and 1,191,872 IP Addresses. (Statistics source: <http://www.php.net/usage.php>, Netcraft) PHP interpreter (or often called ‘parser’) generates HTML code within the Web server. Therefore, it does not consume CPU time or memory whereas, in CGI, another process is created and executed and it consumes a great deal of CPU time and memory. Before the page is sent to a user that has requested it, the Web server calls the PHP interpreter to parse the script and perform the operations called for in the PHP script.

An HTML page that includes a PHP script is typically given a file name suffix of ".php" ".php3," or ".phtml". In an HTML document, PHP script (similar syntax to that of Perl or C ) is enclosed within special PHP tags. Therefore, the author can jump in and out of PHP mode in an HTML file (similar to ASP and Cold Fusion) instead of having to rely on heavy amounts of code to output HTML. PHP-enabled web pages are treated just like regular HTML pages and you can create and edit them the same way you normally create regular HTML pages.\*<sup>1</sup> PHP does not require compilation. You can think of it as a normal HTML file which happens to have a set of special tags available to you that do a lot of interesting things. PHP can perform any task that any CGI program can do, but its strength lies in its compatibility with many types of databases. Also, PHP can talk across networks using IMAP, SNMP, NNTP, POP3, or HTTP. Originally, CGI mechanisms was used to invoke and execute a PHP script. But now, there are many Web servers that support PHP, that is, have PHP-specific modules. In those cases, the PHP script can be handled directly by the web server, and you will not need to execute it using the CGI mechanisms. It is commonly used to extract data out of a database and present it on the Web page. In order to avoid confusion, here I would rather repeat the following. By the word “PHP”, three things are referred to. One is PHP scripting language, which can be embedded in HTML code. The second one is PHP interpreter module within a Web server. The third one is PHP as a whole environment (including both PHP scripting language and PHP interpreter) which helps implement dynamic features on the Web.

## **II. Why People Use PHP? - Advantages of Using PHP –**

1. The old fashioned approach to adding interactivity was to use CGI mechanism. The problem with CGI is that each new request requires the server to start a new process in the kernel, which uses both CPU time and memory, making CGI scripts much slower. This makes multiple concurrent CGI scripts run very slowly. PHP solves this problem by becoming a part of the Web server itself.
2. Another reason to use PHP is because it's free. PHP is an Open Source solution, freely available for a wide variety of platforms. It is a full-featured environment that has been available for years and is in widespread use on the Web. Any user can download both the source code and executables and install them on their computer for free. PHP is constantly being improved by many experienced programmers from all over the world. It is currently available for all major platforms. Therefore, if you're on a tight budget, PHP is a fine alternative for building dynamic data-driven Web sites.
3. PHP is easy. The language is a mixture of C and Perl, taking the best features from both.
4. One of the strongest and most significant feature of PHP is its native database support for a wide range of databases (MySQL, mSQL, Oracle, Sybase, Informix, Direct MS-SQL, PostgreSQL, Unix dbm, IBM DB2, dBase, InterBase, FrontBase, Ingres, Hyperwave, Adabas D, Solid, FilePro, Empress, Ovrimos, Velocis), which allows access to the databases directly through SQL statements.\*<sup>2</sup> Additionally PHP supports ODBC, the Open Database Connection standard, so you can connect to any other database supporting this world standard.
5. PHP can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the Web servers today. This includes Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet servers, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd, and many others. The majority of the Web servers have a PHP interpreter module. However, on the other servers supporting the CGI standard, PHP can work as a CGI processor. Anyway, with PHP, you have 'freedom of choice' regarding an operating system and a Web server.

### **III. What can PHP do?**

PHP is mainly focused on server-side scripting, so you can do anything that any other CGI program can do, such as collect form data, generate dynamic Web page content, or send and receive cookies. PHP is also good at database access, disk access, networking, and text manipulation.

1. Server-side scripting. This is the most traditional and main target field for PHP. You need three things to make this work. The PHP interpreter (as a server module or CGI), a Web server, and a Web browser. You need to run the Web server with PHP enabled.
2. Command line scripting. You can make a PHP script to run it without any server or browser. You only need the PHP interpreter to use it. This type of usage is ideal for scripts regularly executed using 'cron' (on UNIX or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.
3. Writing client-side GUI applications. PHP is probably not the very best language to write windowing applications, but if you know PHP very well and would like to use some advanced PHP features in your client-side applications, you can also use PHP-GTK (an extension to PHP; not available in the main distribution) to write such programs.
4. With PHP, you are not limited to output HTML. PHP's abilities include outputting images, PDF files, and even Flash movies. You can also output easily any text, such as XML file<sup>\*\*\*2</sup>.
5. PHP also has support for talking to other services using protocols such as LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (on Windows) and countless others.

#### **IV. How to Write PHP Scripts?**

The following is a basic example of PHP's features.

[Example]

```
1: <html>
2: <head>
3: <title>A Basic PHP page</title>
4: </head>
5: <body bgcolor="#ffffff" text="#000000">
6: <?php
7: $date = date( "l F d, Y" );
8: print( $date );
9: ?>
```

10: </body>

11: </html>

Line 6: Tells the PHP interpreter that the PHP code has begun.

Line 7: Assigns a value to the variable \$date. The value is a formatted date generated by PHP's built in 'date' function.

Line 8: Uses the built in 'print' function to output the variable \$date to the screen.

Line 9: Tell the PHP interpreter to stop interpreting PHP code.

Variables always start with a dollar-sign in PHP. Any form element in a form will automatically be available to your PHP scripts.

[Example]

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

When the user fills in this form and hits the submit button, the "action.php" page is called.

[Example of "action.php"]

```
Hi <?php echo $_POST["name"]; ?>.
You are <?php echo $_POST["age"]; ?> years old. Right?
```

A sample output of this script is:

```
Hi Joe.
You are 22 years old. Right?
```

The \$\_POST["name"] and \$\_POST["age"] variables are automatically set for you by PHP. Please note that you need to save the file with the '.php' extension when you have finished coding.

## V. Comparison with ColdFusion

ColdFusion is a web application server, and PHP is a web oriented scripting language. Both can be used to manage dynamic web content, but do so in different ways.

### A. Similarities:

- Both provide dynamic contents.
- Both use HTML-like syntax so are easy to learn and easy to do maintenance.

- Compared with CGI, both consume less CPU time and memory, so a user can get a quick response.
- Neither needs compilation.
- Both cover variables, conditional statements, loops, wide variety of functions, and function libraries.
- Both can be integrated with other resources such as XML<sup>\*\*\*2</sup>.

## B. Differences:

There are many differences between PHP and ColdFusion. Below I am putting some of them:

		PHP	ColdFusion <sup>*3</sup>
Basic Architecture		Within Web server	Own ColdFusion server
Control		More control	Less control
Language	Tag format	<?php statement1; statement2; ...?> (Inside the tag, syntax resembles a mixture of Perl and C.)	<cfxxxx param1= value1 param2=value2 ...> </cfxxxx>
	Form input value	[Example] \$_POST["productid"] (* ) One of the most powerful features of PHP is the way it handles HTML forms.	[Example] form.productid
	Variable	\$xxxxx	xxxxx (However, for example, for output and SQL, the variable must be surrounded by '#' symbols (#XXXX#).)
Editor		Many	ColdFusion Studio (& others)
Development & Test Tools		Less convenient for a novice	Convenient tools (e.g. Integration with DreamWeaver, Flash)
Development period		Longer (However, it depends on the person's skill.)	Shorter (However, it depends on the person's skill, too.)
Cost		Free	Expensive

## Summary

PHP is a free, open source scripting language. It is easy to develop and provides powerful control. It provides strong support for a wide range of databases, too. PHP is also available for a wide variety of platforms. It can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, etc. It has also support for most of the Web servers today. This includes Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet servers, O'Reilly Website Pro server, and many others. With PHP, you will have 'freedom of choice' regarding a platform, an operating system, a Web server, and a database. It is intriguing to think that we can easily build up a database and a dynamic Web site by using these kinds of free, public domain software. The Internet liberated knowledge and information from the monopoly by a small group of people to the mass. Now knowledge and information are open to everyone. People can obtain knowledge and information much more easily than before and can organize them in a way they like by utilizing new technology that costs you less expensive or even free.

## Notes:

\*\*\*1 **CGI** CGI is the name of specifications that defines the method (e.g. regarding the names and values of environment variables) used when a server creates (invokes) a process (program). A program made based on the specifications is called "CGI program". Among "CGI programs", those that are written in script language that does not require compilation such as 'Perl', 'awk', 'sh' are called "CGI Script". Although Perl is most popular, any language that provides a means to reference to 'environment variables' and 'standard input' and to write the results of the processing to 'standard output' can be used for CGI programming. For example, C, C++, any Unix shell, and VisualBasic can be used for CGI programming. It is not that there is a separate software named CGI. CGI is just a mechanism by which a Web server can execute an external process or script. The external script could be a Perl script, C, shell script, or any executable program. CGI is itself, though, not an external process or a script. All major Web servers (Apache, IIS, Netscape) have mechanisms to support CGI.

\*\*\*2 **XML** Short for **Extensible Markup Language**, mainly designed for Web documents. It allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

## References Used

- \*1 <http://www.php.net/manual/en/tutorial.firstpage.php>
- \*2 <http://www.webreference.com/new/991028.html#feature>
- \*3 <http://www.allaire.com/products/coldfusion>  
[http://www.macromedia.com/jp/software/coldfusion/resources/get\\_started/](http://www.macromedia.com/jp/software/coldfusion/resources/get_started/)  
<http://www.macromedia.com/jp/software/coldfusion/productinfo/newfeatures/>

## Related Links on the Web

<http://www.php.net/>

<http://www.php.net/usage.php>

<http://www.php.net/manual/en/intro-whatcando.php>

[http://www.php.net/get\\_download.php?df=php-4.2.3-installer.exe](http://www.php.net/get_download.php?df=php-4.2.3-installer.exe)

<http://webopedia.internet.com/TERM/P/PHP.html>

<http://www.techweb.com/encyclopedia/defineterm?term=PHP>

<http://phpwebsite.appstate.edu/mod.php?mod=faq&menu=0&PHPSESSID=8f99e8e4244511b946ab7fea96f94a5c>

<http://www.apache.org/>

<http://www.fokus.gmd.de/linux/ht tpd-help/php/manual/manual.html>

<http://www.phpbuilder.com/>

<http://www.devshed.com/>

<http://www.hotwired.com/webmonkey/99/21/index2a.html>

<http://webreference.com/perl/xhoo/php1/>

<http://www.webreference.com/new/991028.html#feature>

<http://www.web2010.com/tutorial/>

<http://www.phpwizard.net>

<http://www.netcraft.com/Survey/>

[http://www.samag.com/documents/s=1155/sam0101d/0101d\\_s1.htm](http://www.samag.com/documents/s=1155/sam0101d/0101d_s1.htm)

<http://www.allaire.com/products/coldfusion>

[http://searchdatabase.techtarget.com/sDefinition/0,,sid13\\_gci334246,00.html](http://searchdatabase.techtarget.com/sDefinition/0,,sid13_gci334246,00.html)

<http://www.itworks.demon.co.uk/phpeditors.htm>

[http://searchwebservices.techtarget.com/ateQuestionNResponse/0,289625,sid26\\_cid460567\\_tax288842,00.html](http://searchwebservices.techtarget.com/ateQuestionNResponse/0,289625,sid26_cid460567_tax288842,00.html)

This paper is written by Yuko Kashiwagi for the course  
EDC385G Interactive Multimedia Design & Production  
at the University of Texas - Austin