

PHP: Introduction

Introduction

PHP is one of the most prevalent Web programming languages for creating dynamic Web content. Its similarity to C's syntax and open-source nature make PHP relatively easy to learn. This document provides a brief overview of PHP, several reasons for using PHP as a Web programming language, and a list of minimum requirements to begin working with PHP.

What is PHP?

PHP is a scripting language, created in 1994 by Rasmus Lerdorf, that is designed for producing dynamic Web content. PHP originally stood for "Personal Home Page," but as its functionality expanded, it became "PHP: Hypertext Processor" because it takes PHP code as input and produces HTML as output.

Although PHP is typically used for Web-based applications, it can also be used for command-line scripting – a task commonly handled by Perl – as well as client-side applications (possibly with a graphical user interface) – a task commonly handled by Python or Java. *NOTE:* This course only addresses PHP in the context of dynamic Web content creation.

PHP is...

- interpreted rather than compiled like Java or C.
- an embedded scripting language, meaning that it can exist *within* HTML code.
- a server-side technology; everything happens on the server as opposed to the Web browser's computer, the client.
- cross-platform, meaning it can be used on Linux, Windows, Macintosh, etc., making PHP very portable.

PHP does not...

- handle client-side tasks such as creating a new browser window, adding mouseovers, determining the screen size of the user's machine, etc. Such tasks are handled by JavaScript or Ajax (Asynchronous JavaScript and XML).
- do anything within the Web browser until a request from the client has been made (e.g., submitting a form, clicking a link, etc.).

Why use PHP?

PHP has been described as being "better, faster, and easier to learn than the alternatives." (Ullman, 2004) The alternatives being...

- HTML – static, no database/file interaction, cannot handle email, etc.
- CGI (Common Gateway Interface) scripts – Perl or C.
- Microsoft ASP (Active Server Pages), or ASP.NET – need to learn VBScript, C#, or another language and is limited to servers running Microsoft Windows Server.
- Macromedia ColdFusion – expensive to purchase.
- JSP (JavaServer Pages) – Java-based, so there is a higher learning curve plus you deal with the overhead of a compiled language. Personal experience also suggests that JavaServer sites are slower than other sites.

Additionally, PHP is free, works on multiple platforms, and is open source (i.e., user-driven development, not part of a corporation). These attributes combined with a large development community and the wealth of available PHP libraries make PHP an attractive choice for Web developers.

The Web content delivery process has the same beginning and end: A user requests a URL from the Web server, and HTML markup is sent to the user's browser. With the *static* HTML delivery process, the Web server sends the exact content of the requested page to the client without any additional processing. With *dynamic* Web content creation, PHP acts as a filter by processing PHP code – possibly embedded within HTML markup – to produce HTML markup that is ultimately sent to the user's browser.

PHP requirements

The following elements represent the minimum requirements to begin working with PHP. Concepts such as PHP installation and configuration are beyond the scope of this course.

1. PHP-enabled Web server. The EECS/Claxton Web server is running PHP 5.2.
2. If you are working on your own machine, you will need an SFTP client to transfer your PHP scripts to the Web server. Otherwise, you can edit your files directly on the file system via an SSH session with a text editor (e.g., vi, emacs).
3. Web browser (e.g., Mozilla Firefox, Apple Safari, Microsoft Windows Internet Explorer).