



An Overview of PHP

[Like](#)

5 people like this. Be the first of your friends.

0

[Previous](#)[Table of Contents](#)[The History of PHP](#)**From Techotopia**[+](#) [SHARE](#) [f](#) [t](#) [e](#) ...[Next](#)[Creating a Simple PHP Script](#)

Purchase and download the full PDF and ePub versions of this eBook for only \$8.99

[Buy Now](#)

Having covered The History of PHP in the previous chapter it is now time to provide some detail as to what PHP actually is. In this chapter we will take a high level look at PHP and provide a basic understanding of what it is, what it does and how it does it.

Contents

- 1 What Exactly is PHP?
- 2 How Does PHP Work?
- 3 Why is PHP so Useful?
- 4 Summary

What Exactly is PHP?

PHP is an intuitive, server side scripting language. Like any other scripting language it allows developers to build logic into the creation of web page content and handle data returned from a web browser. PHP also contains a number of extensions that make it easy to interact with databases, extracting data to be displayed on a web page and storing information entered by a web site visitor back into the database.

PHP consists of a scripting language and an interpreter. Like other scripting languages, PHP enables web developers to define the behavior and logic they need in a web page. These scripts are embedded into the HTML documents that are

served by the web server. The interpreter takes the form of a module that integrates into the web server, converting the scripts into commands the computer then executes to achieve the results defined in the script by the web developer.

How Does PHP Work?

To develop an understanding of how PHP works it is helpful to first explore what happens when a web page is served to a user's browser.

When a user visits a web site or clicks on a link on a page the browser sends a request to the web server hosting the site asking for a copy of the web page. The web server receives the request, finds the corresponding web page file on the file system and sends it back, over the internet, to the user's browser.

Typically the web server doesn't pay any attention to the content of the file it has just transmitted to the web browser. As far as the web server is concerned the web browser understands the content of the web page file and knows how to interpret and render it so that it appears as the web designer intended.

Now let's consider what kind of web page content a web browser understands. These days a web page is likely to consist of HTML, XHTML and JavaScript. The web browser contains code that tells it what to do with these types of content. For example, it understands the structure HTML in terms of rendering the page and it has a JavaScript interpreter built in that knows how to execute the instructions in a JavaScript script. A web browser, however, knows absolutely nothing about any PHP script that may be embedded in an HTML document. If a browser was served a web page containing PHP it would not know how to interpret that code.

Given that a web browser knows nothing about PHP in a web page, then clearly something has to be done with any PHP script in the page before it reaches the browser. This is where the PHP pre-processing module comes in. The PHP module is, as mentioned previously, integrated into the web server. The module tells the web server that when a page is to be served which contains PHP script (identified by special markers) that it is to pass that script to the PHP pre-processing module and wait for the PHP module to send it some content to replace that script fragment. The PHP processing module understands PHP, executes the PHP script written by the web developer and, based on the script instructions, creates output that the browser will understand. The web server substitutes the content provided by the PHP pre-processor module in place of the PHP script in the web page and sends it to the browser where it is rendered for the user to view.

To help understand this concept let's take a quick look at a before and after scenario. The following HTML contains some PHP script that is designed to output an HTML paragraph tag:

```
<html>
<head>
<title>A PHP Example</title>
</head>
<body>
<?php
    echo '<p>This line of HTML was generated by a PHP script embedded into an HTML document</p>';
?>
</body>
</html>
```

The above example looks very much like standard HTML until you reach the part surrounded by `<?php` and `?>`. These are markers that designate where the embedded PHP script begins and ends. When the web server finds this it sends it to the PHP module. The PHP module interprets it, converts it to HTML and sends it back to the web server. The web server, in turn, sends the following to the browser:

```
<html>
<head>
```

```
<title>A PHP Example</title>
</head>
<body>
<p>This line of HTML was generated by a PHP script embedded into an HTML document</p>
</body>
</html>
```

Once loaded into the browser, it is rendered just like any other web page. The fact that the web page originally contained PHP is completely transparent to the web browser.

The above example is certainly an oversimplification of the power of PHP. Some may question why one would use PHP to output some static text that could have been achieved more easily using an HTML tag. The fact is, however, HTML only makes sense if you know beforehand exactly what needs to be displayed in the web page. Imagine instead, that you are developing an online banking application. One of the pages you need to display must contain the customer's bank account number combined with the current balance. Obviously this information is going to be different for each customer. In this scenario you would develop an HTML page that essentially serves as a template for the page, and then embed PHP into the page to extract the account and balance information from a database. Once processed by the PHP module integrated into the web server, this customer specific content will then appear in the HTML page in place of the PHP script when the page is loaded into the browser.

Why is PHP so Useful?

In terms of web page content we have two extremes. At one extreme we have HTML which is completely static. There is very little that can be done with HTML to create dynamic content in a web page. At the other extreme we have scripting languages like JavaScript. JavaScript provides a powerful mechanism for creating interactive and dynamic web pages.

When talking about JavaScript it is important to understand that it is, by design, a client side scripting language. By this we mean that the script gets executed inside the user's browser and not on the web server on which the web page originated. Whilst this is fine for many situations it is often the case that by the time a script reaches the browser it is then either too late, or inefficient, to do what is needed. A prime example of this involves displaying a web page which contains some data from a database table. Since the database resides on a server (either the same physical server which runs the web server or on the same network as the web server connected by a high speed fiber network connection) it makes sense for any script that needs to extract data from the database to be executed on the server, rather than waiting until it reaches the browser. It is for this kind of task that PHP is perfectly suited. It is also fast and efficient (because the script is executed on the server it gets to take advantage of multi-processing, large scale memory and other such enterprise level hardware features).

In addition to the advantages of being a server side scripting language PHP is easy to learn and use. The fact that PHP works seamlessly with HTML makes it accessible to a broad community of web designers.

Perhaps one of the most significant advantages of PHP to some is the ease with which it interacts with the MySQL database to retrieve and store data.

Summary

In summary, PHP has many advantages, and those listed here are just some of the reasons for the success of PHP. Many people will offer their own reasons for using PHP - and this fact alone is testament to the power and flexibility of PHP.

Now that we know a little about The History of PHP and have an overview of how it works we can start looking at how to develop PHP based web sites, beginning with Creating a Simple PHP Script.



Purchase and download the full PDF and ePub versions of this PHP eBook for only \$8.99

Buy Now

Retrieved from "http://www.techotopia.com/index.php/An_Overview_of_PHP"

- This page was last modified 19:22, 13 May 2009.
- Copyright 2011 Techotopia.com. All Rights Reserved.

This ad is supporting your extension *Bookmark Sentry*. [More info](#) | [Privacy Policy](#) | [Hide on this page](#)