Mason Daniel
Southwestern University
CSC54-394 01

# PHP: Hypertext Preprocessor

PHP is a web-based programming language that was created around the time that the world wide web came into the world of technology. It was initially conceived by a programmer of the name Rasmus Lerdorf in 1994. The first public release came about in 1995 under the name Personal Homepage Tools version 1.0. Lerdorf originally intended the language to be a simple templating system that would take back-end code from C and then provide the front-end display code. Essentially, it was only meant to format how a web page looks; the heavy coding would be done in C, and the language would format how that code looks as a final product on the web browser. Although this was his vision, he received feedback from many other programmers at the time all asking for more features in PHP itself because the language was easier to code in than C. Lerdorf complied to these wishes, and updated the language to give it more functionality [1]. As the web became more and more prevalent and grew rapidly, so did PHP. The second release, PHP 2.0, was announced in 1996. Lerdorf described the language as a “Server-side HTML-embedded scripting language” in his release announcement, and this was actually the first time the term ‘scripting language’ was used [2]. At this point, the project was still a one-man team, with minor contributions from others. With the language’s user base expanding, two other developers, Zeev Suraski and Andi Gutmans volunteered to rewrite the original code for a third release, PHP 3.0. The new edition included compatibility with popular operating systems like Windows, Mac, and Unix, popular web servers such as Apache, and databases like Oracle and PostgreSQL [2]. These new features gave PHP greater flexibility, and with that came a huge increase of users. At this point, the project had already evolved from a one-man effort to a worldwide, open-source project; and that was only the beginning.

The language is still very prevalent in 2018. The current version at the time of writing is PHP 7. It is a language that is still used universally, even though it has been around for two decades. It is so deeply rooted in the web that it would be difficult for it to become obsolete. In fact, according to statistics recorded by software consultant

company Q Success, “PHP is used by 78.9% of all the websites whose server-side programming language we know.” [3]. (This study was taken from records of the top 1 million websites based on traffic). PHP grew alongside the web itself, and became an essential tool in web development. A combination of the fact that it was introduced during the early stages of the Internet and that it was so flexible even from the beginning allowed the language to become so universal and longstanding. So what exactly makes the language so universal and flexible? How exactly does it work?

Two big reasons that PHP is so appealing is its ability to interact in conjunction with databases and its ability to create dynamic web pages. Additionally, it can be embedded into HTML. Before explaining how these features are benefits, though, and what they even mean, it is important to first discuss how PHP is implemented in general. To start, an simple overview of how users access a website is in store—i.e. how the Internet works. First, a user will open a browser, such as Google Chrome. When a user types in a URL, such as ‘google.com’, that URL is translated to a string of numbers, which represents an ip address. This ip address represents another machine—a server—that is connected to the internet, which contains the files that ‘google.com’ is associated with (e.g. the Google homepage). So, the user’s computer has now found another computer of sorts that contains the web pages they are looking for. The user machine’s browser will then request the server to send any necessary files over that the user has asked for—in this example, the Google homepage. This request is fulfilled, and the user machine’s browser has now received the Google homepage from the server, and it is displayed on the user’s screen. So where does PHP come in?

Like other languages, PHP is coded in a text editor and saved into a file with a unique extension; in this case, it is .php. A user will use the language’s syntax/lexical structure to write a program, and then save that program. The user will then use the downloaded PHP software, as well as another software that will act as the webpage’s server. An example of one of these server softwares is called Apache, which is one of the most widely used web server softwares. While a server is primarily used to contain all the files that will be requested by a user machine when trying to access a webpage,

an Apache server contains the necessary software to process many different kinds of files, such as .html, .htm, and in our case, .php files [2]. As stated before, the user will request a webpage from the server, and then the server will send that webpage to the user's browser. With this knowledge, the idea that PHP is a server-side scripting language can now be explained to show where it comes in during the process of web development. If a webpage that is being requested was created using PHP (and thus has a .php extension), then the apache server will recognize this. That .php file is then processed by the Apache server, and will output any .php code into HTML format. Once the file is converted to HTML format, the server finally fulfills the user's request and sends the webpage back to the browser. Then, the browser recognizes that the file is HTML, and displays it accordingly [2]. Since all the processing of .php files is done by the server, the PHP code that was written is hidden from visitors of the webpage, because they can only see the HTML that was outputted by the server.

PHP webpages are useful because all work of processing them is done by the server, and not the user's machine from which they are requesting to see those webpages. Additionally, the language is so popular because it allows developers to create dynamic webpages. A dynamic webpage is simply a webpage that can be changed. For example, you could create a .php file to store a website's homepage, and then create other .php files that correspond with popular links found on a website's homepage, such as 'About' and 'Contact Us'. You can then include code in the homepage file to refer to the other .php files that represent different pages. These .php files can all be stored under one directory and even if there are many of them, they can still be considered as a single entity if placed in one folder ('entity' here is not used as a technical term, it is simply used to explain that one can create many different .php files in the same folder that can be used to display a single web page). This makes the code much more readable, as certain parts of the webpage are stored one time and then accessed, rather than loaded each time a new link is clicked. Additionally, PHP files allow interaction with databases. In the context of the browser-server relationship that happens when a web page is requested, access to the database will happen on the

server while the .php code is being processed. PHP includes MySQL commands in its dictionary, so databases can be queried through PHP code [4]. This is especially useful for websites that require databases to store large amounts of data, which is extremely common as of now.

While these features make PHP appealing, they result in the language having characteristics that web developers find satisfying. Firstly, it has great readability and writability. It was intended to be easy on the eye at its conception, as mentioned by Lerdorf. Much of its syntax is self-explanatory, and keywords like ‘echo’ or ‘print’ are not ambiguous to their function. Additionally, the language uses semicolons at the end of statements, and brackets to outline control structures (see Figure 1).

```php
<?php
$value = 'yes';
echo "<input type='checkbox' name='subscribe' value='yes'/> Subscribe?";

if (isset($_POST['subscribe'])) {
    if ($_POST['subscribe'] == $value) {
        $subscribed = true;
    } else {
        $subscribed = false;
        print 'Invalid checkbox value submitted.';
    }
} else {
    $subscribed = false;
}

if ($subscribed) {
    print 'You are subscribed.';
} else {
    print 'You are not subscribed';
}
```

Figure 1: This PHP code excerpt is an implementation of a validation of a single checkbox. In this example, a checkbox is displayed on the web page, and if it is clicked, the user is ‘subscribed’; if it is not clicked, they are ‘not subscribed’ [5].

In comparison to other languages, it has many similarities to Perl and C. PHP 3 brought the introduction of object-oriented programming to the language, and it was later

improved upon in PHP 4. Additionally, PHP 5 introduced many other structures such as private methods and variables, abstract classes, constructors, destructors, and a standard exception handling model [4]. These kinds of structures are commonly found in modern versions of object oriented programming languages such as Java and Python, so it has kept up with current programming trends. In addition to PHP being a flexible language with many modern programming implementations, it is also very accessible. The PHP software is completely free to the public, and can be downloaded from their website. The language can be used in conjunction with other softwares too, as discussed before. Still, these softwares (such as Apache for web servers or MySQL for database querying) are free as well, which allows anybody with access to the internet a costless way to get involved in web development. PHP is compatible with many operating systems as well, like Windows, Mac, or Linux (Linux being a free operating system), opening the doors for many computer users.

Despite PHP being a flexible, accessible, readable and writable language, all of these advantages do not come without drawbacks. One of the main disadvantages has also been discussed as an advantage—server-side scripting. While this feature means that all the PHP processing is done on the server, and thus not on the client's machine where it could be viewed easily, this could end up being a disadvantage. If many requests are being sent in to one server, and they all require .php files to be processed, this can cause the server to slow down. Speed is a critical factor when it comes to web development, as web pages are expected to be displayed instantaneously. If creating a web page in PHP could result in slower response times, developers may be turned off from the language. With this said, a web server slowing down due to high traffic is unavoidable in any scripting language. The solution is not in which language is used, but rather in the amount of servers that a web host uses—more servers must be used if traffic becomes too congested. Another characteristic of PHP that could be seen as a disadvantage is the fact that it is weakly typed. This means that variable types do not have to be specified when the variable is declared. While some programmers consider this a plus, it requires more manual organization than a strongly typed language.

Mason Daniel
Southwestern University
CSC54-394 01

Weakly typed languages allow greater room for error, as types can be mismatched in the code, but not recognized as errors until an incorrect output is produced. This trait being a positive or negative one comes down to whether or not the developer prefers flexibility or structure and organization. Nevertheless, it should be considered a drawback because the increased chance of error must be accounted for.

Overall, PHP is a relatively simple language to learn. It has many similarities to languages that many programmers know and use today. It is compatible with other softwares, like MySQL or Apache, and with commonly used operating systems, free of cost. It is a scripting language that is processed on a web server rather than a client, and its syntax, control structures, object-oriented nature and other language paradigms are familiar to nearly every programmer with previous experience in coding. Even to a beginner, PHP would be a good starting point for learning how to code, due its unambiguous, readable and writable nature. Although it is a scripting language usually used for web development, it is quite versatile because of its features. On a greater level, PHP represents the efforts of many individuals who were—and still are—passionate about programming. It began in the hands of a single man, Rasmus Lerdorf, and evolved into a worldwide, open-source project that is still functioning today.

Mason Daniel
Southwestern University
CSC54-394 01

# References

[1] Rasmus Lerdorf. 2017. PHP in 2017 - Rasmus Lerdorf @ WeAreDevelopers Conference 2017. Video. (30 May 2017). Retrieved October 11, 2018 from https://www.youtube.com/watch?v=fYTKm2oUzAg

[2] Kevin Tatroe, Peter MacIntyre, Rasmus Lerdorf. 2013. *Programming PHP* (3rd. ed.). Sebastopol, California, NY.

[3] W3Techs. 2018. Usage statistics and market share of PHP for websites. (October 2018). Retrieved October 13, 2018 from https://w3techs.com/technologies/details/pl-php/all/all

[4] The PHP Group. History of PHP - Manual. Retrieved October 10, 2018 from http://php.net/manual/en/history.php.php

[5] Demo Source and Support. 2015. Validating a single checkbox : Form Checkbox « Form « PHP. (2015). Retrieved October 5, 2018 from http://www.java2s.com/Code/Php/Form/Validatingasinglecheckbox.htm