**What is PHP?**

PHP is an open-source, server-side, HTML-embedded **scripting language** that is used to develop **Static or Dynamic websites** or Web applications.

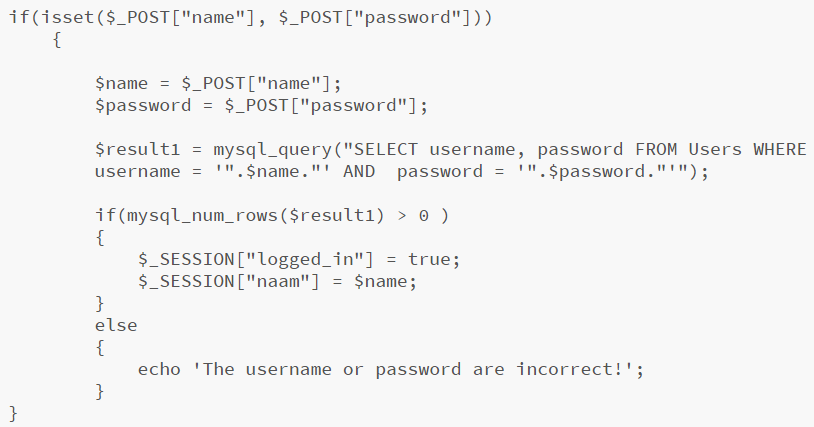
PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages.

PHP scripts can only be interpreted on a server that has PHP installed.

The client computers accessing the PHP scripts require a web browser only.

A PHP file contains PHP tags and ends with the extension "php".

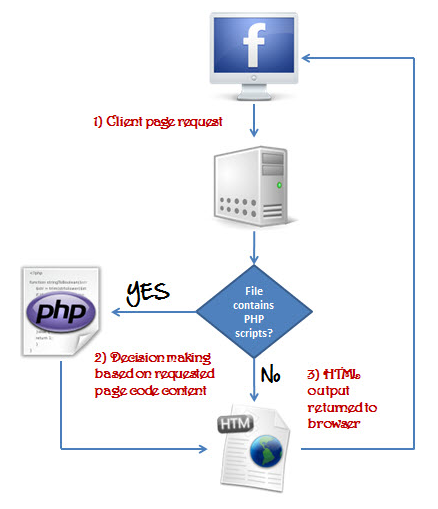
**Syntactical example:**

****

A PHP file can also contain tags such as HTML and client side scripts such as JavaScript.

* **HTML** **is an added advantage** when learning PHP Language. You can even learn PHP without knowing HTML but it’s recommended you at least know the basics of HTML.
* **Database management systems** DBMS for database powered applications.
* For more advanced topics such as interactive applications and web services, you will need **JavaScript and XML**.

The flowchart diagram shown below illustrates the basic architecture of a PHP web application and how the server handles the requests.



Also notice how this is different from a CGI script written in other languages like Perl or C -- instead of writing a program with lots of commands to output HTML, you write an HTML script with a some embedded code to do something (in this case, output some text). The PHP code is enclosed in special [start and end tags](https://ifj.edu.pl/private/krawczyk/php/language.basic-syntax.html#language.basic-syntax.phpmode) that allow you to jump into and out of PHP mode.

What distinguishes PHP from something like client-side Javascript is that the code is executed on the server. If you were to have a script similar to the above on your server, the client would receive the results of running that script, with no way of determining what the underlying code may be. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

## So what is a Scripting Language?

A script is a set of programming instructions that is interpreted at runtime.

A scripting language is a language that interprets scripts at runtime. Scripts are usually embedded into other software environments.

The purpose of the scripts is usually to enhance the performance or perform routine tasks for an application.

Server side scripts are interpreted on the server while client side scripts are interpreted by the client application.

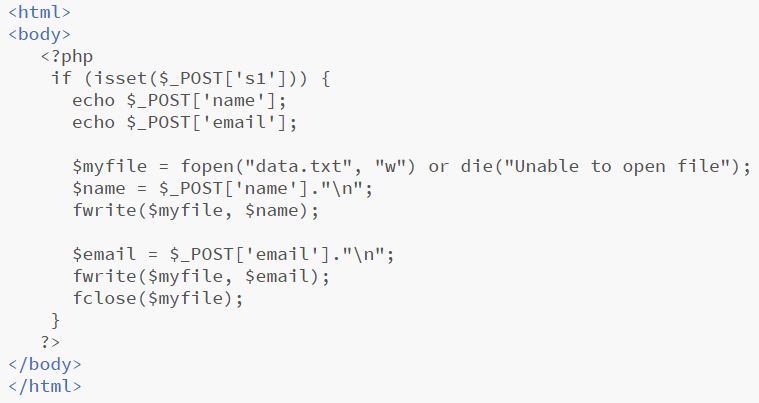
PHP is a server side script that is interpreted on the server while[JavaScript](https://www.guru99.com/interactive-javascript-tutorials.html)is an example of a client side script that is interpreted by the client browser. Both PHP and JavaScript can be embedded into HTML pages.

**PHP File Extensions**

*File extension and Tags* In order for the **server** to **identify** our **PHP** **files** and **scripts**, we must **save** the **file** with the **“.php” extension**. Older PHP file extensions include

* .phtml
* .php3
* .php4
* .php5
* .phps

PHP was designed to work with HTML, and as such, it can be embedded into the HTML code, for example:



You can create PHP files without any html tags and that is called a **Pure PHP** file.

The server interprets the PHP code and outputs the results as HTML code to the web browsers.

In order for the server to identify the PHP code from the HTML code, we must always enclose the PHP code in PHP tags.

A PHP tag starts with the less than symbol followed by the question mark and then the words “php”.

PHP is a case sensitive language, “VAR” is not the same as “var”.

The PHP tags themselves are not case-sensitive, but it is strongly recommended that we use lower case letter. The code below illustrates the above point.



We refer to the PHP lines of code as **statements**. PHP statements end with a semi colon (;). If you only have one statement, you can omit the semi colon. If you have more than one statement, then you must end each line with a semi colon. For the sake of consistency, it is recommended that you always end your statement(s) with a semi colon.  PHP scripts are executed on the server. The output is returned in form of HTML.

**How PHP came into being:**

PHP was conceived sometime in the fall of 1994 by [Rasmus Lerdorf](mailto:rasmus@php.net" \t "_top). Early non-released versions were used on his home page to keep track of who was looking at his online resume.

The first version used by others was available sometime in early 1995 and was known as the Personal Home Page Tools. It consisted of a very simplistic parser engine that only understood a few special macros and a number of utilities that were in common use on home pages back then.

A guestbook, a counter and some other stuff. The parser was rewritten in mid-1995 and named PHP/FI Version 2. The FI came from another package Rasmus had written which interpreted html form data. He combined the Personal Home Page tools scripts with the Form Interpreter and added mSQL support and PHP/FI was born. PHP/FI grew at an amazing pace and people started contributing code to it.

It is difficult to give any hard statistics, but it is estimated that by late 1996 PHP/FI was in use on at least 15,000 web sites around the world. By mid-1997 this number had grown to over 50,000. Mid-1997 also saw a change in the development of PHP. It changed from being Rasmus' own pet project that a handful of people had contributed to, to being a much more organized team effort.

The parser was rewritten from scratch by Zeev Suraski and Andi Gutmans and this new parser formed the basis for PHP Version 3. A lot of the utility code from PHP/FI was ported over to PHP3 and a lot of it was completely rewritten.

Today (end-1999) either PHP/FI or PHP3 ships with a number of commercial products such as C2's StrongHold web server and RedHat Linux. A conservative estimate based on an extrapolation from numbers provided by [NetCraft](http://www.netcraft.com/" \t "_top) (see also [Netcraft Web Server Survey](http://www.netcraft.com/survey/" \t "_top)) would be that PHP is in use on over 1,000,000 sites around the world. To put that in perspective, that is more sites than run Netscape's flagship Enterprise server on the Internet.

Also as of this writing, work is underway on the next generation of PHP, which will utilize the powerful [Zend](http://www.zend.com/" \t "_top) scripting engine to deliver higher performance, and will also support running under webservers other than Apache as a native server module.

**What does PHP offer, and why not use another language?**

* PHP is **open source and free.**
* Short learning curve compared to other languages such as JSP, ASP etc.
* Large community document
* Most web hosting servers support PHP by default unlike other languages such as ASP that need IIS. This makes PHP a cost effective choice.
* PHP is regular updated to keep abreast with the latest technology trends.
* Other benefit that you get with PHP is that it’s a **server side scripting language**; this means you only need to install it on the server and client computers requesting for resources from the server do not need to have PHP installed; only a web browser would be enough.
* PHP has **in built support for working hand in hand with MySQL**; this doesn’t mean you can’t use PHP with other database management systems. You can still use PHP with
  + Postgres
  + Oracle
  + MS[SQL](https://www.guru99.com/sql.html)Server
  + ODBC etc.
* PHP is **cross platform;** this means you can deploy your application on a number of different operating systems such as windows, Linux, Mac OS etc.

## Market share:

In terms of market share, there are over 20 million websites and application on the internet developed using PHP scripting language, this may be attributed to the points raised above.

## PHP vs Asp.Net VS JSP VS CFML

## [ASP](https://www.guru99.com/asp-net-tutorial.html) – Active Server Pages, [JSP](https://www.guru99.com/jsp-tutorial.html)– Java Server Pages, CFML – Cold Fusion Markup language The table below compares the various server side scripting languages with PHP

## 

**Dynamic vs Static websites:**

Websites are separated into two different types: static and dynamic. Static websites are ones that are fixed and display the same content for every user, usually written exclusively in HTML. A dynamic website, on the other hand, is one that can display different content and provide user interaction, by making use of advanced programming and databases in addition to HTML. Static websites are easier to create, while dynamic websites require more work.

Before getting into details about each website type, understand how the Internet goes about serving up websites in the first place. Internet communication involves a server and a web browser.

To establish a connection between the two, a set of rules called Hypertext Transfer Protocol (HTTP) is used. Simply put, the web browser transmits an HTTP request to the server, and the server then replies with an HTTP response along with the requested webpage in HTML.

### What is a static website?

Static [websites usually come with a fixed number of pages that have a specific layout.](https://wpamelia.com/website-layouts/) When the page runs on a browser, the content is literally static and doesn’t change in response to user actions. A static website is usually created with HTML and CSS in simple text editors like Notepad.

If you need a [website smaller than three pages,](https://wpamelia.com/single-page-website/) opting for a static website is the proper choice. Building it doesn’t take as much time or effort as in the case of dynamic websites. If the pages of your website must look different, the HTML code can easily be duplicated on each of these pages, containing the necessary changes.

Even though the website will display the same thing with no intricate navigation details, static websites don’t need to feature just plain text. In fact, you can use various multimedia elements and videos. An HTML website can look beautiful, but the page’s [source code](https://wpamelia.com/javascript-ide/) won’t change, no matter what actions a user takes on it.

### What is a Dynamic website?

Compared to static websites, which are purely informational, a dynamic website is more functional. It allows users to interact with the information that is listed on the page. Of course, that requires utilizing more than just HTML code.

Static websites use only client-side HTML and CSS code while dynamic websites rely on both client-side and server-side scripting languages such as JavaScript, PHP, or ASP. When a user accesses a dynamic website, the site can be changed through code that is run in the browser and/or on the server. The end result is the same as that on a static website: an HTML page displayed on the web browser.

To generate dynamic content, such websites use a combination of server-side and client-side scripting. Client-side scripting refers to code that is executed by the browser, usually with JavaScript. Meanwhile, server-side scripting refers to code that is executed by the server (before the content is sent to the user’s browser).

### Static vs Dynamic:

* **Meaning**

The word static refers to something that is fixed, that doesn’t move or change in any way. This is enough to understand what a static page is all about. No elements on this page are changed when accessing it.

Static websites are basic pages that require simple code and design elements to create. “Static” also refers to the website being fixed in terms of page numbers. A fixed number of pages are delivered just the way it is designed and stored.

Conversely, the word dynamic refers to elements that are continuously changing, interactive, and functional. Instead of being simply informational, dynamic websites include aspects that are characterized by interactivity and functionality. They are more complex in terms of building and design, but they are also more versatile.

* **Thechnicality**

When discussing static vs. dynamic websites from a technical point of view, the differences between the two types of websites become even clearer. Because static websites only contain fixed content, building them can easily be done in plain HTML. The only way that a user can interact with a static page is by clicking hyperlinks and filling in forms (such as a contact form).

Dynamic websites are ultimately based on HTML and CSS as well, but server-side scripting is required to make them functional. HTML coding is used to create the basic design elements, while server-side languages are used to manage events and control actions that may occur on the dynamic page. For example, a WordPress website built with a theme like [Ombrello](https://templatic.com/wordpress-corporate-business-themes/agency-theme/) is a dynamic website.

* **Coding**

To create a static website, the user doesn’t need to use complex software programs. Some knowledge in HTML and CSS along with Notepad should be enough to build a simple static website. Static pages include elements such as text and multimedia elements. They are not as technical as a dynamic website, but they are not as effective either. Users will see the same design and content each time they visit the website unless you change the source code manually.

A dynamic website generates the content and displays it based on what actions the users make on the page. The preferences of the user alter what is displayed to them, which can be an intricate process based on the sophistication of the website. A special editor, such as an IDE (Integrated Development Environment), is required to build dynamic websites, along with strong technical skills in server-side language programming.

**References:**

* <https://www.php.net/manual/en/intro-whatcando.php>
* <https://ifj.edu.pl/private/krawczyk/php/introduction.html>
* <https://ifj.edu.pl/private/krawczyk/php/intro-history.html>
* <https://www.guru99.com/what-is-php-first-php-program.html>
* <https://wpamelia.com/static-vs-dynamic-website/>