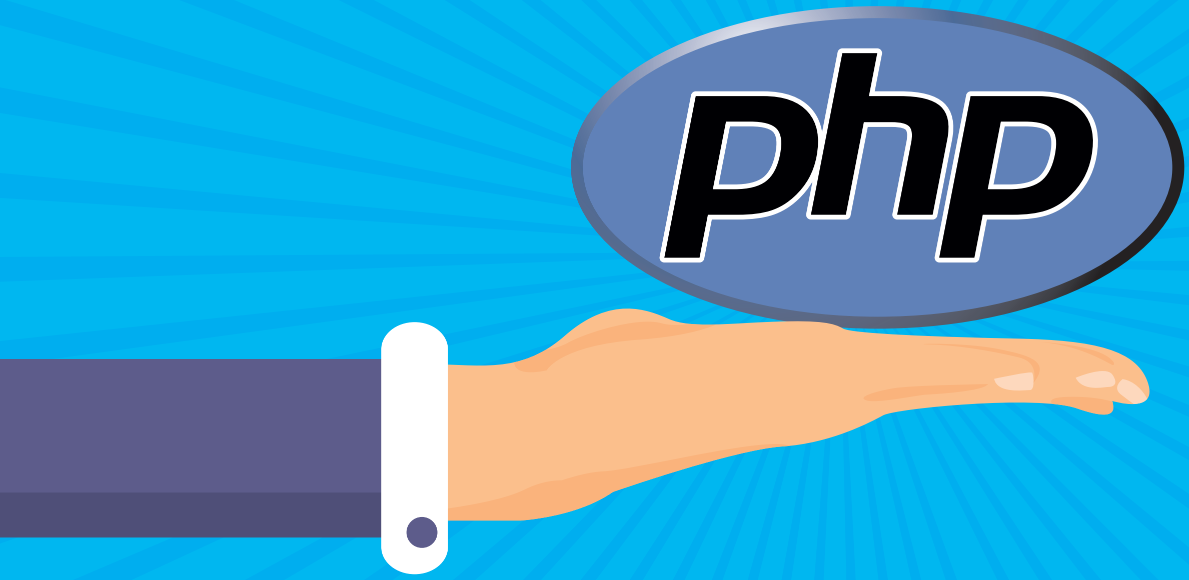


WHEN AND WHY YOU SHOULD USE PHP

The layman's guide, and the businessman's tool, to decipher PHP.

» UNNI P N «

Although PHP has been primarily used to program the front-end side of Web-based Applications, like all good Open Source initiatives, the language has evolved beyond recognition from the 1995 original. It now handles complex tasks, and is used by the world's biggest and most futuristic enterprises.

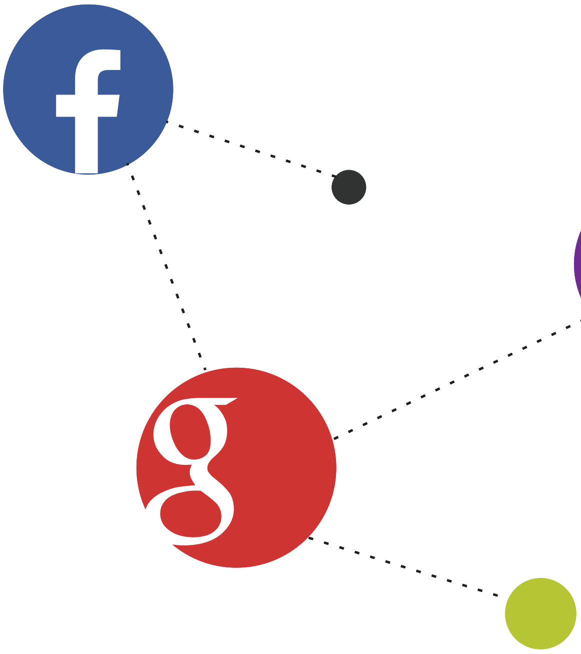
Introduction

PHP has been around from 1995. Since its advent, it has slowly and steadily grown into the major open source web-development language that is out there. The language has received mixed reviews over the years, and has had many overhauls.

Kevin Schroeder, who was earlier involved with ZEND, and is now associated with Magento (among many other interests), has listed out ten very compelling reasons why he uses PHP in this well-written article. You can find a lot of similar evaluations on the Internet about what is wrong or right about PHP. But, search engines being what they are, most of the content that turns up on a cursory search is outdated, and can lead to misconceptions about the state and capabilities of the language.

Although PHP has been primarily used to program the front-end side of web-based applications, like all good Open Source initiatives, the language has evolved. In recent years PHP and the tool-chain eco system around PHP has evolved to a level of maturity that PHP is now being used to program complex business logic and develop enterprise applications.

In an article for O'Rielly Media, Josh Lockhart (a.k.a @codeguy), the creator of the Slim Framework, has written about the current state of PHP and the changes in its core, and why it is now ranked among the best programming languages out there.



This is not your parents' PHP. The new PHP is a more mature language with community standards, a growing affinity for interoperable components, and a passionate movement to improve performance. If you have bypassed PHP for alternative languages, or if you are a PHP veteran unaware of recent changes, you owe it to yourself to give PHP a second look.

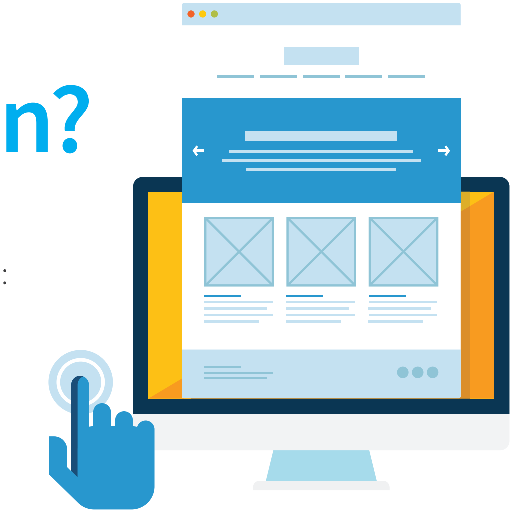
Most of you will recognize the three below:

There is quite a list of enterprises that use PHP in production.

- **FACEBOOK**
- **GOOGLE**
- **YAHOO**

Facebook is probably the best example to illustrate what PHP can be used to achieve, in a very short span of time. Most of the core functionalities of Facebook are written in PHP, simply because it is easier to get more developers on board with PHP, as well as the fact that it offered the best turnaround time when Zuckerberg and Co. made the decision. Facebook relies so heavily on PHP, that it has released a new PHP engine called the Hip-Hop Virtual Machine to optimize its performance, but more about that later.

When is PHP your best option?



PHP can be made to do many things well, like:

- **eCommerce**
- **Content Management Systems**
- **Project Management Systems**
- **Online Communities**
- **FaceBook Applications**

Having said that, there is NO one glove that fits all hands. There are some things that PHP does better than other languages, and vice-versa. What needs to be figured out is whether PHP can solve your problem. Most of the best enterprises out there employ a complex tool-chain utilizing certain languages for what they do best. For example, Twitter was initially using Ruby on Rails for all of its functioning. But they found that Ruby itself was so slow in managing their message queues, that they re-implemented their message queue system using Scala, which has its own set of pros and cons. The Twitter eco system has now been reworked quite heavily, to add performance as they grow. But let's face it, they can afford to do so. The question is can you?

The choice of technology for an application is based most of the time, sadly, on the promoted opinions about certain tool chains or languages, and not on an objective evaluation of the capabilities of the tools themselves.

When choosing the technology that will build your dream application, the first question to answer is what you/your application needs. Ask if your idea/application checks true to one or more of the points below. If so, you might want to give PHP some serious consideration.

Is your application going on the web?

Is your application available over the Internet? Web applications are primarily hosted out on a server so that your users can use it on a standard web browser. The primary goal that drove the creation of PHP was to provide the ability to develop dynamic web applications easily. Though PHP now has the capability to drive GUI based stand-alone applications, it is still best suited for the Web. Being natively developed for the Web, PHP comes with an API that has been sharpened over the years, to hit the target of making development for the Web as easy as writing the alphabet.

There are **multiple frameworks of all sizes and shapes** built with PHP, combined with a lot of **documentation/community support** that make developing your application for the web a cakewalk, be it creating your own Social Networking Giant, or building the fastest API for your Mobile Application. Since PHP has been in the domain of creating web applications for nearly two decades, the biggest advantage that PHP offers for Web-based applications is that almost **all cloud/web hosting service providers support PHP out of the box.**

Is your application going to be content heavy?

Content management has been a major pain point in web development. On the Web, your application/portal is only as good as its content. Is it relevant and updated? Most applications require powerful content management features that allow non-tech administrators to maintain and update the content. It is quite expensive to build something like that from scratch to suit your needs, whatever the technology.

But, this is now a solved problem. There are multiple content management systems (CMS) out there, which provide an extremely robust set of features to satisfy most of your content management needs. All CMSs are designed to be extended, so that any custom business logic or workflow can be built on top of its feature sets. The top CMSs out there, like Wordpress, Drupal or Joomla, are built with PHP. Together, the PHP based CMSs power the majority of the content heavy applications across the Web. The ease with which each of them can be customized will also ensure that development costs are minimal.

The biggest advantages of choosing a PHP-powered open source language are:

- They are all tested and proven
- The availability of add-ons and plugins are enormous. This keeps customization costs to the minimum, making your applications extremely extendable even without expert developers
- The CMS itself is FREE, meaning you only have to pay a developer to tweak it for you.

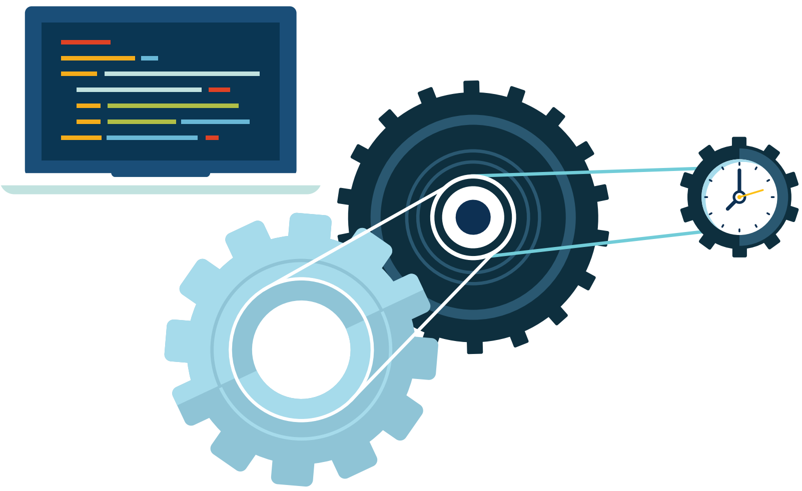
Is your application workflow oriented?

Is your application going to be about workflows, where the user performs actions from a predefined set, and the server returns results mainly based on database interactions and some business logic? Yes? Then, PHP is best suited for such applications. PHP has a shared nothing architecture, making sure that there are no conflicts or contention for resources between individual requests made to the server. PHP also provides inherent support for a large range of databases. For applications that use databases heavily, or require advanced text and XML processing capabilities, PHP is a great choice. One of the most used as well as robust eCommerce frameworks, Magento, is also written in PHP.

Do you require a short turnaround time?

Last but not the least; this is the most important question to answer. The amount of time that you are willing to wait for your application, can have varied effects on the amount of money you will have to invest to get it done. There are a plethora of scripts, libraries, frameworks and support available, that are well written to do most of the generic tasks in the domain.

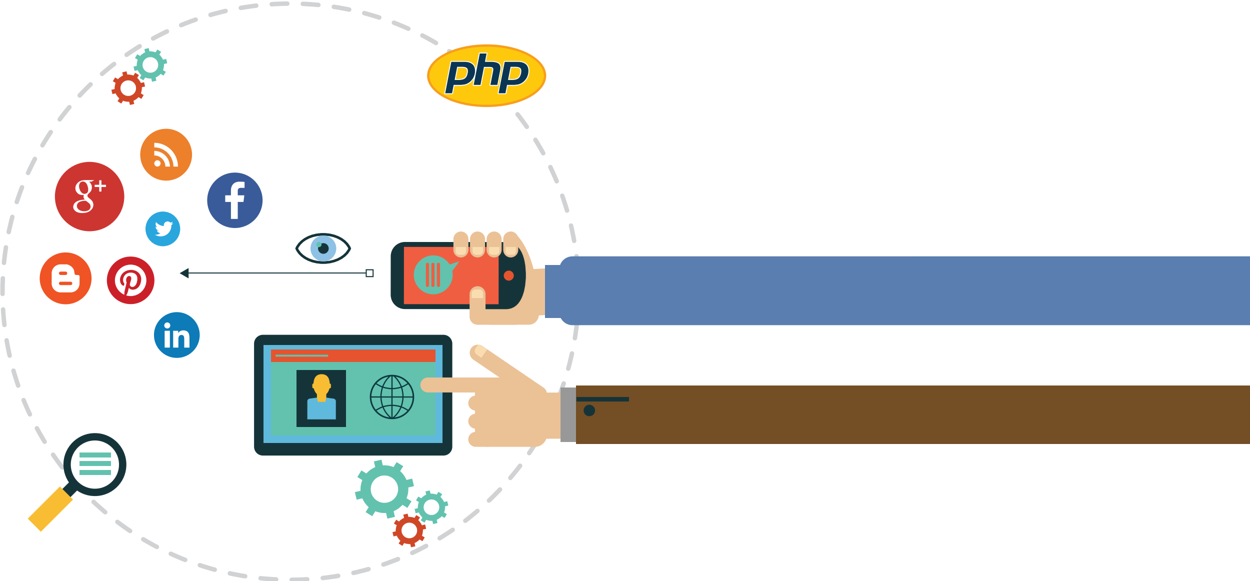
If your requirement falls into one of the above categories, PHP can deliver the best quality in the minimum time with minimum effort, simply because it was created to do just that!



3 unbeatable reasons why PHP stands at the top of the heap

Now that you have seen the best scenarios for, and the immense possibilities of using PHP, there are three simple, reasons why you *should* use PHP:

1. PHP and its related Stack is completely **free**
2. PHP is known to have the **least learning curve** among all interpreted web development languages anywhere. This would mean that you can **easily find talent** in PHP and plug in developers much quicker to projects. Any developer can pick up PHP. What's more, a lower learning curve and higher resource availability translates to **lower development costs**
3. The availability **ready-to-use frameworks, libraries and scripts** are higher in PHP than other languages. PHP also allows you to **bootstrap projects quickly,** as it is **supported out-of-box** on all platforms.



Is the Future of your Application Secure with PHP?

It would seem so!!

Many online giants, like Facebook, have put their money behind the language. They have recently released the HipHop Virtual Machine (HHVM) to run Hack, a language that is in effect PHP, but much better under the hood. Translated for PHP programmers and application users, this would mean that developing/ using PHP applications will be like driving your favorite car, only with a much better engine. What's so great about that?

The stated goal of the HHVM is:
To run all existing PHP code, existing out in the wild.

This means all your existing PHP code can also run on this high-performance engine. Knowing Facebook and how openly-governed projects have evolved, this should imply that **PHP is here to stay,** and that it will soon power systems that are on scale with Facebook or Google.

> “PHP solves problems faster, easier and with a lower cost of ownership than almost anything out there.
>
> ..I’m putting my money where my mouth is, and my mouth is saying that, if you’re doing the web, PHP is where it’s at for at least the next 5 years, if not longer. Beyond the 5-10 year range we don’t really know what programming languages will look like. People will guess and 73.5% of them will get it wrong. So for the foreseeable future, PHP is where it’s at on the web.”

Kevin Schroeder

PHP Developer, Author, Musician, Global Domination Theoretician

Suyati
dream. dare. do

Unni P N, *Team Architect & FOSS Protagonist*

Suyati Technologies

At the core of Suyati's corporate culture, is the spirit of collective good, much like the Open Source movement itself. Our **expertise in FOSS technologies** covers application development, web application development, and mobile app development.

To know more about how we can help you, get in touch with us:
services@suyati.com

www.suyati.com