



# **PHP versus humanity**

**Aleh Talstoi**

# Table of Contents

<b>Table of Contents</b> .....	1
<b>Abstract</b> .....	2
<b>Introduction</b> .....	2
1.1 Background .....	2
1.2 Objectives.....	2
<b>Research Questions</b> .....	3
<b>Method</b> .....	3
<b>Analysis and Discussion</b> .....	3
4.1 Introduction .....	3
4.2 PHP Advantages.....	4
4.2.1 <i>Beginner friendly</i> .....	4
4.2.2 <i>Flexibility</i> .....	4
4.2.3 <i>Scalability</i> .....	5
4.2.4 <i>Excellent for Frameworks</i> .....	5
4.3 PHP Disadvantages.....	5
4.3.1 <i>Security</i> .....	5
4.3.2 <i>Not for huge apps</i> .....	5
4.3.3 <i>Weak type</i> .....	5
4.4 PHP compared with its competitors.....	5
4.4.1 <i>PHP vs Java</i> .....	6
4.4.2 <i>PHP vs ASP</i> .....	6
4.4.3 <i>PHP vs Ruby</i> .....	7
4.4.4 <i>PHP vs CGI/Perl</i> .....	7
<b>Results</b> .....	8
5.1 PHP vs Java.....	8
5.2 PHP vs ASP.....	8
5.3 PHP vs Ruby .....	8
5.4 PHP vs CGI/Perl.....	8
5.5 Summary.....	8
<b>Conclusion</b> .....	9
<b>Future work</b> .....	10
<b>References</b> .....	11

## Abstract

*The aim of this study is to research why PHP is getting so much of negative feedback from experienced programmers and what makes people to still stick with it after so many years of web-development. Many people do leave negative reviews of PHP and tells their opinions about other languages, telling us that those are much better alternatives.*

*Non less focus of this research was also to help new web-developers to make the right choice early in their carrier. Many people study PHP at universities and fall in love with it because it is pretty easy to use all over globe. Is that worth to keep studying it? Or is it best time to stop so early and focus on different language that many people rate as a better version?*

*To achieve this I was reading many articles and opinions about PHP as well as articles that explain good/bad sides of both PHP and other alternative languages such as Java, ASP, Ruby and CGI/Perl. Those researches gave me a clear vision of that every single language I compared PHP with and PHP itself have its own usages which means you can't decide if PHP is bad or good that easily. I understood at the end that every language I brought up could work best in specific project with a specific team of developers working on it.*

*In conclusion I can say that Programmers who started to learn PHP do not have to be afraid. PHP is a good beginner language which will not hurt if you learn it. Especially when it is so easy to learn and have many usages. For more advanced developers I can say that they have to try both PHP and the alternative they wonder about to decide themselves which language they are most comfortable with and which does fit better into their project and team. Yes, PHP has some deep problems of language construction but so does many other languages. And in my opinion PHP is worth learning.*

*Keywords: PHP, web-development, opinions, ASP, Java, Ruby, Perl, comparing, novice friendly.*

Faculty of Computing  
Blekinge Institute of Technology  
SE-371 79 Karlskrona Sweden

## 1 Introduction

### Background

It's important to not to waste time in the beginning of your carrier and focus on learning languages that is currently appreciated on market these days. It will help programmers to get an easy start and help them to find job more easily and get their lives started. But there is a small problem in education right now as I see it. It looks like most of the programs on universities worldwide does include a lot of PHP studies. Until today I thought that was nothing bad with it. PHP is easy to learn and use so what could go wrong? I started to dig into roots of PHP and the more I dug the more negativity I found. So I started thinking that there is something wrong with that and wanted to know why is that so? Why does PHP receive so much negativity even though it is so popular everywhere? Maybe my study could help novice web developers to make the right choice and stop focus on PHP studies and start to study something else. Maybe I will just refute all that negativity and prove that there is nothing wrong with PHP.

### Objectives

In this work I will try to understand why is PHP so underrated by digging into root of its problems. I will perform that by comparing it to most of its replacements which are popular on the market at the moment. Meanwhile I will research that I will try to decide which side do I belong to myself. Will I still be so interested in learning more about PHP or will I swap to one of those replacements, and why. I will be focusing on comparing PHP and its replacements that many people prefer over it such as ASP, Ruby, Java and Perl. Those four languages are in some way similar but in many ways different. Those differences makes so many people to prefer those listed languages over PHP, but why? I am about to find out. But first I need to see what is the main differences and if they are so important.

I will also take a look at some reviews/opinions of some people that worked with those languages and hopefully that will shed some light on all that negativity.

## 2 Research questions

In this work I will focus following areas and try to answer following questions:

- Why is so popular coding language as PHP is so underrated?
- What makes similar coding languages a better alternative to PHP?
- As web developer, is that a good idea still to learn PHP or not? And Why?

My plan is to give an explained answer to those questions during this work.

## 3 Method

To find information for this work I will mostly be using Google as source. But since it is a lot unverified information out there on internet I will be examining every source I find and determine if it is a relevant source and who have created it. For instance in my opinion, sources that has been created past 5 years are still relevant for this research because new coding languages does not releases that often. And after some of them do, they need a good period of time to be tested and compared to other ones. For instance if there would be an alternative to PHP that would be released back in 2013 it would by now gathered some feedback and different web developers would have builded an overview over it already that I could use in my work to answer my questions. I also might meet some blogs on the way with some developers's personal opinions and in that case I need to try to find out more information about that specific developer so I can be sure he does know what he is talking about. That could be tough since not all of the users do upload a lot of informations about themselves.

That[1] is a popular web page used by developers to hire assistants or find freelancers and work at some project together. There is also lots of blogs and material written by different developers about some topics. That one that I linked before in this chapter does bring up some of the information important to my research but I still have to be sure that it has been written by a qualified programmer and is true. As you can see, there is a tab at the bottom of the post that does describe poster, Yoshitaka Shiotsu in our case, and description of him as developer and writer. There is also a link to his profile that has even more info about him. That post has been posted at March 19, 2014 which is just about 3 years ago which is not that long ago. That source can be determined as a verified source and be used in my work.

## 4 Analysis and discussion

### Introduction

*“If you're currently a PHP developer, I mean no personal disrespect, but before you scroll down the to the comments to join the fray, I do sincerely and honestly encourage you to learn another way to make web pages.”*

- [Adam Bard, blogger, programmer.](#)

*“PHP is an embarrassment, a blight upon my craft. It's so broken, but so lauded by every empowered amateur who's yet to learn anything else, as to be maddening.”*

- [Eevee, blogger, programmer.](#)

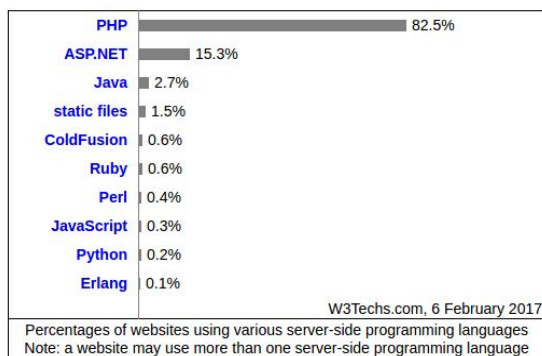
*“Yes, it's a horribly designed language (read [PHP: a fractal of bad design](#) if there is any doubt in your mind)”*

- [Mattias Petter Johansson, Developer at Spotify](#)

All quotes I mentioned above are just a few from a pile of such commentaries, opinions and blogs. You get many more hits on google while searching “Why PHP is bad?” rather than “Why PHP is

good?”. That is pretty strange, especially when PHP is used by circa 82% of all web pages as a server side programming language (pic 1 below). For you to better understand why such hate is strange I will briefly explain bright sides of PHP that at first made it so popular and maybe why people still stick to it. All of the people that are defending PHP does say that the strongest side with this programming languages that it is very Beginner Friendly.

That is both blessing and curse of PHP. It makes many novice web programmers to start to learn it because it is easy and decent. Bright side of it is that PHP community expands dramatically and since this language is old that means that community at present is enormous which could explain so huge usage of it. It is easy to find help that you need when there is millions of people working with/on it that are ready to help at any time and almost with any problems. But there is also downsides with that such as Beginner Friendliness lead to many new and unknowing programmers to use that Language which does lead to people complaining about language without its full understanding. By that I mean for example, a new programmer started with PHP because it seem to be an easy option. That programmer makes his first project but do get some errors and bugs because of misunderstanding of how to use it properly. Programmer getting tired of it and makes a post where he blames PHP for being an awful language and so on. That creates some of the hate around it. The thing is here is that if a programmer language is a bit more challenging, it sorts out unknowing people and only have left those who really try to learn all of its aspects before using it. But we cannot know what is bad with this programming language before we dig into its good sides. ( pic 1)



## Advantages

### Beginner friendly

Many people that do want to learn PHP are driven by thought of an easy job. The most popular framework known at the moments is Wordpress, because of, again, its simplicity and usage. Many companies that want to get an easy and fancy web-page do often search for programmers that could give them that with Wordpress. It will not take long time for a hired person to create such website with Wordpress and therefor they would not need to pay him a lot for that. PHP is, as I mentioned above, indeed very popular since it is Nr3 Largest StackOverflow Community. It also is Nr5 Largest Meetup Community. For instance it is over 500k members spread across groups on [Meetup.com](https://www.meetup.com/). PHP is also 5th Most Popular Language at [GitHub](https://github.com/). That basicly means that the more projects it is on GitHub the bigger chance it that there is some helpful functions that will speed up your development which is another benefit of using a popular language. But what are those top 3 languages at [GitHub](https://github.com/)? It is Javascript, Java and Python. Those are even more popular over past years since PHP started to getting more and more problems in late term because of bad design.

### Flixibility

PHP is considered to be one of the most flexible web programming languages[7]. Mostly because developers can do different changes and additions to futures at any time anywhere in the code. Developers are not required to write all code back from scratch as they are required in other similar languages. They just simply need to adjust some of the functions to fit their new program. PHP developers are, for example, allowed to even use external programs from within PHP and later inject results back into a PHP application! That is all incredibly easy with PHP. There is many uses for flexibility for a web programming languages so that is definitely a benefit.

## Scalability

Scalability is an important aspect of web programming language. Most of the web pages will grow and will have more and more traffic coming through. To keep web page running fast and having great uptimes are main things you want from a web language. Most people who want a small simple web page for themselves might easily use PHP code embedded into HTML. But what should you do if you want to use it to larger pages? Well, PHP itself is not that good at handling high traffic. But there is plenty of frameworks build in PHP that is capable of creating large web applications within a short time that could handle that high traffic in the future. [Facebook](#) is a great example of a high traffic web page. While being used by around 2 billions people around the globe at June 2017 you want to use a Language capable of handling so much data. Guess what, Facebook uses a [modified version of PHP](#) which they extended and adapted to high traffic. That proves that PHP is indeed scalable to large scales if needed.

### Excellent for Frameworks

As I mentioned just above, PHP does have many frameworks that help people to develop much faster. What exactly makes it easier is for example that frameworks provide you with well-organized, reusable and maintainable code as well as it takes care of most of low-level security of your web page so you don't have to worry about it. There is plenty of good frameworks out there but here is top 10 of the most popular and worth checking out:

1. [Laravel](#)
2. [Symfony](#)
3. [CodeIgniter](#)
4. [Yii 2](#)
5. [Phalcon](#)
6. [CakePHP](#)
7. [Zend](#)
8. [Slim](#)
9. [FuelPHP](#)
10. [PHPixie](#)

Faculty of Computing  
Blekinge Institute of Technology  
SE-371 79 Karlskrona Sweden

## Disadvantages

### Security

Frameworks do the great job to compensate those PHP weaknesses. But we are here to discuss what problems do PHP itself have and if it worth learning at all so It is right about the time to discuss what are the main weaknesses of such open sourced language could it have. So PHP as an open source language which gives possibility to absolutely anyone to explore its core and its bugs which lead us to huge security problem. Any well educated hacker could simply open up source code for php and find a couple of bugs as an advantage trying to hack some sites built in PHP. Not that assuring huh?

### Not for huge apps

As I mentioned before, Frameworks do it good to maintain high traffic when PHP itself just simply can't. In a way how PHP was created, as a not highly modular language, it will be hard to maintain huge applications. Without using PHP frameworks it is highly not recommended to attempt using it in any large websites or applications. It will do much better in smaller ones.

### Weak type

Another weakness worth mentioning is that even if PHP is easy to learn it still takes some time to master it and its behaviour. For instance "Implicit conversion" might be a surprise for a novice programmer. As example strings "1e3" and "1000" would show as equal in PHP even though they are completely different. That happens because those strings are implicitly cast to floating point numbers. Such flaws could lead to some trouble while building larger sites in future.

## PHP compared with its competitors

As we already know about both Advantages and Disadvantages of PHP it is time to see why is other programming languages is more popular to use instead of PHP. We will start with comparing it with Java.

### **PHP vs Java**

Both of those languages have their usages but Java is still a better choice long term. Compared to PHP, Java isn't an open source language which automatically removes all of those security risks I was mentioning before. People making small websites for themselves might not care about security that much as people at huge companies do. Security issues in their web pages might cost them millions so they would definitely prefer to use Java over PHP so 1:0 to Java so far. Java also provides developers with such useful feature as platform independence. That might not sound like much but it does speed up development dramatically. It gives developers a possibility to easily move their work between different machines and systems which increases flexibility in development. So Java claims another point here. Java is also, compared to PHP, an object oriented language which makes it a great choice while building large applications since it will be much easier to maintain such applications. Java leads 3:0 against PHP does it mean that Java is a better alternative? No matter how good of alternative Java might sound, it has some issues. For example development speed. Developers do choose PHP over Java mainly because PHP tends to have much less issues when using shared hosting meanwhile for Java it would be best to have a dedicated server for hosting. One of disadvantages of Java could also be its availability. The thing is that Java is not free to use compared to PHP which means that PHP is more available for novice programmers than Java. Because of that you, as a Java student, could though earn a certification, while PHP doesn't have any type of qualification level.

### **PHP vs ASP**

Compared to Java, ASP or Active Server Pages is much more similar to PHP and does share many advantages and disadvantages. Both of those languages are very popular to make Dynamic Websites. Mostly because there is many similarities in those two programming languages. For instance, both does interact with databases and help with exchanging of information. Both languages are equally equipped with tools to access file systems, find images, and display pages on a web server and those do work almost as fast as each other. Speed can vary dependant on database server, bandwidth and user's computer. As we know, one of PHP's benefits is that it is a very scalable language. Well, ASP does also share this benefit. But those similar languages does also have many differences. One point that PHP wins almost always is that it is free and open source and plus it does run on Mac, Windows or Linux while ASP.net runs on Windows only (there is a possibility to run it on Linux with ASP Apache installed on your server) and also ASP is not free and neither open source. ASP is a microsoft product which provides developers with a high level security in cost of availability. That means once again that PHP has larger pool of developers and available help which is definitely a point to PHP. Another big difference between PHP and ASP is Base Language. PHP does use C++ as base, while ASP uses Visual Basic syntax that often comes in many Microsoft products. C++ is considered to be one of the most popular programming languages by almost all programmers and since PHP is very similar to C++ it is loved more than ASP.net. In general PHP is a better choice here when it comes to make a better web page, when much better application can be made in ASP.

## PHP vs Ruby

I found out that Ruby and PHP does have some similarities and differences as well as when comparing PHP to other languages. Both are dynamically typed, which means that you can access variables after or before declaring them. Both do have external libraries to call on if needed as well as both languages have exceptions and classes. Differences though are not that big. As I noticed, main difference that PHP has with every single language I've compared it to is that PHP is much easier to learn and still is widely used and asked for. In that case, compared to Ruby Rails, PHP's easy learning curve might be seen by an easy example. If you want to send a header to the browser using PHP you just use its build-in function `header()`. Unlike PHP, Ruby must first load in required libraries then use modules or namespaces to get to the same result. As you can see both languages do offer same result at the end, but first one is at first must easier to find thanks to PHP's large support community. But to get equivalent result in Ruby you need to learn a bit more about Ruby itself which requires a bit more time. But if you manage to put some time into learning Ruby Rails you will get access to a very powerful and flexible language with an excellent web framework in Rails. But as I said before you will need to put some time into it as well as it will require a high technical cap of you. With other words Ruby is not a language for a novice programmer but it does its job for an experienced programmer. PHP compared to it again, just an easy option.

## PHP vs CGI/Perl

So, Perl. In fact second free alternative to PHP after Ruby. Also a very-very old programming language that has been developed back in 1987, even before the Web itself. You may wonder how a Web how a server-side Web development language could exist even before Web? The answer is simple. Perl, as well as PHP, was not developed to do what it does nowadays. When created, Perl was intended to be a multi purpose scripting language that would be amazing at getting text from different sources and combine them into complete document. Sound familiar? That's because this is exactly how server-side Web development works. So developers of Perl modernized it and at the end it became what it is today. PHP has almost the same backstory to it. PHP was created by Rasmus Lerdorf as a problem solving tool and not as web programming language for huge projects. With time PHP has grown with different helpfull functions. Compared to Perl, PHP was kind of made by accident(I will explain this later) when Perl has been developed into exactly what is it today over longer than a decade. That means it is much more well designed and does have much less bugs in it. But what about learning curve and support? Well, even here perl does not give up. Since it is even older than PHP it does have a lot of support that you can get from experienced developers that have been using it for years. Learning of Perl could be a challenge because it is notorious for its different ways to do the same simple thing. That could make it a bit slower to learn and understand but should not be a big of a problem though. Another difference, not a downside, that Perl has been developed in Unix platform so it won't work as good on for example Windows platform. PHP however, is made on Windows which would be a better choice if you are planning to develop on that platform.



## 5 Results

### PHP vs Java

Those two languages aren't really same which means they have their own uses and benefits. If you are searching for an available programming language for some personal pages - PHP is your choice. IT is free to use, does have easy learning curve, additionally development will go much faster due amount of available support that you can find to help you with your project. But if you are a serious company searching for a language that will provide you with high level of security, that will also be able to handle high traffic in larger projects. Java is overall a better option versus PHP in large project due its all benefits. But let's move on[11].

### PHP vs ASP

Once again PHP's only big benefit is that is open sourced and free programming language. As I mentioned several times before it grants a PHP programmer huge support and wide choice of solutions to every problem. From other side that is really it for benefits while comparing to ASP.net. ASP is 2nd most popular programming language after PHP. If you manage to learn ASP it will serve you well in your projects and application making. Mostly because ASP is a framework which I described before is much better in larger scales[12,13].

### PHP vs Ruby

When comparing PHP and Ruby I once again reassured myself that PHP is not bad at all but vice versa, it is a very good language to begin with. It does provide user with a lot of potential and easy-to-remember functions. That is the biggest difference I could find between those two languages. But as I mentioned in comparison above, Ruby is not worse that PHP but takes longer to learn syntax and how it really works. If you manage to do that you can get even better results with Ruby on Rails since it is overall is better organized language.[14]

### PHP vs CGI/Perl

I was quite surprised by how similar those two languages are from backstory, to syntax itself. There is some differences especially in, once again, learning curve that is much lower for PHP rather than Perl(surprise-surprise). Difference in learning curves isn't that big as for instance between Ruby and PHP but PHP still does require a lot less time to master it compared to Perl. By researching about Perl I can say that it is more reassuring than PHP is. Both are open-source programming languages which that they are equally risking security for community. But it is not so in that case at all. Developers working on Perl put much more time and will into it than Rasmus did put in PHP.

### Summary

After researching a lot about PHP and a lot of its alternatives[18] and also looking at several people's thoughts about it overall I could wrap up a single impression about it. Compared to almost every single language PHP has a very simple learning curve. It has great support from all around the globe. Be careful though. Since it is easy to learn, you might get some help from people that does not know PHP so well as they should. PHP is also open-sourced free language that offers large accessibility, which could lead to a huge security risk compared to most of the languages. Any hacker has a chance to take a look at source code, study weaknesses for language and later us it to hack into your web page. This is why this language is not recommended being used in larger projects. Also a reason why it is used more often in smaller projects is because of PHP's lack of ability to handle high traffic. So overall PHP is not a bad language, it is just not as well designed as other alternatives, but if you are out and searching for a language that could provide you with a site fast and quickly PHP is a way to go. But if you want yourself a high quality website that will expand in future and that you want to be able to maintain that whole time, then the best you can do is to spend some time learning another language that will do much better.

## 6 Conclusion

Now, back to the questions that I put in the very beginning of this work.

- Why is so popular coding language as PHP is so underrated?

PHP isn't really underrated as I found out. I got idea of my work from just simply finding negative feedback about PHP and commentaries from different people all over internet that was complaining about it. I thought that PHP might be that bad. I wanted to know why people do say so and do a reasearch around it. As I later found out people mostly was complaining about how language is designed and that is PHP has a many issues when it comes to making large projects. Well, to understand where does all this hate comes we need to take a look at the PHP's origins. PHP was created by Rasmus Lerdorf back in 1994. And originally PHP wasn't even planned to be what it is now. As Rasmus said himself in an Interview[17]:

*"Well, I didn't plan PHP. I think in terms of solving problems, not in terms of software projects. I actually hate programming, but I love solving problems."*

Basicly PHP was not even designed to be a server sided-sided web development language but instead a kind of "Toolbox". But later on people liked it because of its simplicity and variety of tools available at that moment. So instead of starting thinking about evolving his "Toolbox" into a dedicated language, he just kept adding functions and tools until it almost accidentally became a language. Other languages such as C# or Clojure spent several years with a team with great experience to design those languages. That is the main issue with PHP when it comes to poor design and some issues. But PHP is not useless because of that. No, it's instead a very powerful language for smaller web pages. It might still be useful in some projects for students or just for personal web pages but it is not really recommended to use it at some large and quick-growing web pages. PHP can risk sites security as well as its

performance since it is an open-sourced language as I mentioned before as well as it cannot handle huge traffics.

- What makes similar coding languages a better alternative to PHP?

So when we think of how bad is PHP in large projects we come to a question number to about alternatives. Answer is pretty obvious as well. Alternatives that people prefer such as Ruby, ASP, Java and other frameworks that I didn't brought up in this work, are all better in handling high traffic that may occur in popular pages such as youtube or facebook. They are also much more secure because most of them are not open-sourced and therefore much harder to hack through. So what do we have? Programmers that just begun their journey in development do start with some simpler option such as PHP, since it is the most easy-to-learn language, they continue working with it until they come to bigger projects and then they start to see that PHP is giving up and doesn't work so well. After they go to forums and tell everyone that PHP is bad and that other people should use its alternatives. It is completely wrong. Do you see what I see? PHP has done its job helping novice programmers to start up. Later when they grow bigger they do have more needs and need just a different language for that. As we found out already, those alternatives are better in that case. But it also leads to my question to be a bit incorrect. Those languages I mentioned and compared PHP with are not really alternatives. Those are just some kind of next step in web development. Something that people moves on to after they mastered "the beginning" of their journey in web development.

- As web developer, is that a good idea still to learn PHP or not? And Why?

I think I pretty much answered this question by everything I said before. Yes, they should still learn it. And why is a bit more tricky question. In my

opinion PHP is a good language to start with as web developer. Later on if you are planning to expand and start work in large companies and build huge web sites for extremely many people to use, you need to learn additional languages that could help you with that. But if you are planning to just do small private web-pages and don't care about its security, you could stick to PHP like some of the developers still does. There is a very good example that explains that situation with PHP: *Think of you having a broken table. If you want to repair it, you could go and by some tools, take your time to learn how to repair tables and do that. Or you just can take duktape and fix that table in several minutes because guests are about to come.*

If you don't get it, that means that to make a webpage fast you could use a tool that is free and that would do its job or pay for some more advanced tools, spend some time learning new, advanced techniques and make a high quality web page.

I hope that helps to make a decision to every novice web developer out there who does love PHP and is not sure if he should continue working with it or not.

## **7 Future work**

I did answer all of my questions that I had in the beginning of this work, but I got even more meanwhile. Or more like I want to know more about it. For instance dive deeply into code and study it more detailed to know exactly what PHP problems are or how does other languages manage to handle high traffic much more efficient than PHP. There is a lot more to study around it that I could explore by diving more in-deep into the subject as a whole.

## References

- [1] <https://www.upwork.com/blog/2014/03/web-development-101-top-web-development-languages-2014/> - Top Web Development Languages in 2014 (Yoshitaka Shiotsu, 2014-03-19)
- [2] <https://adambard.com/blog/you-write-php-because-you-dont-know-better/> - “You use PHP because you don’t know better.” (Adam Bard, 2013-06-19)
- [3] <https://eev.ee/blog/2012/04/09/php-a-fractal-of-bad-design/> - “PHP: a fractal of bad design” (Eevee, 2012-04-09)
- [4] <https://www.sitepoint.com/why-choose-php/> - “Why Choose PHP Over Alternatives?” (Bruno Skvorc, 2014-02-21)
- [5] <https://medium.com/@mindfiresolutions.usa/how-relevant-is-php-for-web-developers-in-2017-66d8fca75f9f> - “How Relevant is PHP for Web Developers” ([Mindfire Solutions](http://Mindfire Solutions), 2017-07-11)
- [6] <http://github.info/> - GitHub’s info tab that shows popularity of languages.
- [7] <http://www.techrepublic.com/article/take-advantage-of-built-in-php-flexibility-to-execute-external-programs/> - “Take advantage of built-in PHP flexibility to execute external programs.”
- [8] <http://www.cuelogic.com/blog/how-facebook-is-not-using-php-by-still-using-php/> - “How Facebook is not using PHP by still using PHP?” ([Cuelogic](http://Cuelogic), 2015-03-16)
- [9] <https://www.hongkiat.com/blog/best-php-frameworks/> - “Why Use A PHP Framework” ([Anna Monus](http://Anna Monus))
- [10] <https://bigcheaphosting.com/advantages-and-disadvantages-of-php/> - “Advantages and Disadvantages of PHP” ([Davis Paul](http://Davis Paul), 2016-12-30)
- [11] <https://itxdesign.com/php-vs-java/> - “PHP vs Java” (<https://itxdesign.com>)
- [12] <https://itxdesign.com/php-vs-asp/> - “PHP vs ASP.net” (<https://itxdesign.com>)
- [13] <https://www.pixelcrayons.com/blog/web/php-vs-asp-net-how-to-choose-the-right-one/> - “PHP Vs ASP.NET: How to Choose the Right One?” (<https://www.pixelcrayons.com>, 2017-09-08)
- [14] <https://www.upguard.com/articles/blog/ruby-vs-php> - “Ruby vs PHP” (2017-07-04)

## References

[15] [Reference\(long link\)](#) -

“Is PHP a badly designed programming language?” (Mattias Petter Johansson, 2016-02-04)

[16] <https://www.quora.com/What-are-the-alternatives-of-PHP-if-any> -

(Jabali Searcot, 2015-03-06)

[17] <https://www.sitepoint.com/phps-creator-rasmus-lerdorf/> -

(Interview with Rasmus Lerdorf, creator of PHP, 2002-05-22)

[18]

	<b>PHP</b>	<b>Java</b>	<b>ASP.NET</b>	<b>Ruby</b>	<b>CGI/Perl</b>
<b>Learning Curve</b>	Very Low	Normal	Normal	High	Low
<b>Open Source</b>	Yes	No	Yes	Yes	Yes
<b>Scaleable</b>	No	Yes	Yes	Yes	Yes
<b>Security</b>	Very Low	High	High	High	Normal
<b>Available Support</b>	Very Much	Much	Not Much	Not Much	Normal