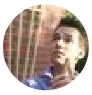
Why doesn't JavaScript have the bad reputation over PHP?

There has a been a lot of criticism of PHP over being a badly designed programming language. JavaScript seems quite a similar language with loose-typing and flexible array declarations/memory management, but is widely adopted by industry leaders such as Google. What makes JavaScript a better language?

19 Answers



Felix Gessert, PhD student in cloud data management, CEO@Baqend
9.7k Views • Upvoted by Sergiy Kyrylkov, Full Stack JavaScript Developer, Adam Hartline,Programming since age 8. Corporate programming since age 25.

JavaScript has features that appeal to advanced programmers: functional programming constructs, closures and prototyping. PHP has features that make every programmer scream in desperation and disbelief.
Both JavaScript and PHP have obvious design issues, here's a random collection of some that come to mind. The good parts of JavaScript are documented in the great book*JavaScript: The good Parts*, by Douglas Crockford [1]. The good parts of PHP have yet to be invented.

JavaScript fails:

- The `==` **equals comparison operator is broken**: it's not transitive. In JavaScript `"" == 0` is true, `0 == "0"` is true, but `"" == "0"` is false. That is bogus, since equality is an equivalence relation which demands transitivity. The `===` operator requires the types (e.g. `Number`) to be equivalent and thus has the desired effect.
- **Plus and Minus** sometimes work as arithmetic and sometimes as concatenation operators. Together with JavaScript's loose typing that can lead to strange cases, e.g. `'9' - '1'` being `8` but `'9' + '1'` being `'91'`.
- The existence of the **eval** function is just that: evil. It should never be used, not for parsing JSON and not for anything else.
- **IEEE 64 Bit Floating Point** Numbers are the only numeric datatype, there are no Integers.
- No **statical lexical scoping of blocks**. The body of an `if` or `while` statement doesn't open a new scope with it's own variables. Only nested Functions open a new scope. That leads to a strange pattern called immediately-invoked function expressions to get that block scoping in places you need it. ECMAScript 6 will have a `let` statement to achieve block scoping.
- The **this** reference is highly confusing. In a function it refers to the instance of the function the invoked function was called on (e.g. to `obj` in `obj.doStuff()`), or - if there is no such object - to the window object. The `this` reference can be set via the higher order functions `apply`, `call` and `bind` .
- **Semicolons** can be omitted, but it's very dangerous: the parser inserts semicolons, where it expects them to be. So if you write a `return` statement with a return value in the next line, the parser will add semicolon to the return line and the method won't return anything [2].
- **Arrays** aren't fundamental types. So `typeof [1]` is `'object'`, which is confusing.
- The **for...in** loop is strange: it iterates over all enumerable properties of an object (even for arrays). If you add a property (`arr.name = 'myArray'` to an array it will iterate over that too. The `Array.foreach` method which takes a callback is a better choice.

PHP fails:

- The **goto statement**. At first PHP didn't have one. In PHP 5 they introduced it. I don't have any words for that, except that the PHP creators probably never heard of Edgar Dijkstra.
- No **list/array** literals: you have to write `array( key => value, ...)`. Update: Fixed in PHP 5.4
- Horribly **inconsistent function naming**: e.g. `strip_tags` vs. `stripslashes`
- Parameters as **global variables**. PHP transforms HTTP GET and POST parameters into global variables of the current script. That's a tremendous security issue (SQL injections, XSS).
- **Error handling**: there are Exceptions, `trigger_error` calls and return codes, which are used very inconsistently.
- No distinction between **Arrays, Lists and Hashtables**. Makes them easy to use and very slow.
- Many different values are considered **false**. For instance `null, '0', 0, '', array()`. That's not a consequence of loose or dynamic typing, it's just bad design.
- Loose (`==`) and strict (`===`) **comparison**, similar to JavaScript but even more unclear [3].
- Variables with a **variable variable-name**: `$a = 'hello';  $$a = 'world';` creates the variables $a and $hello. Hello?! That's not what metaprogramming ist meant to be! You can even have variables like `${${"f$t$t"}}` [4]
- There is no **lexical scoping**. If you write `foreach ($array('a','b','c') as $l) { print "$l, "; }` you can access $l afterwards.
- No simple **extension** system, like Ruby's gems or Python's pip.
- **Four syntactic ways** to basically do the same: `MyClass::staticFunction`, `$myObj->method()`, `My\name\space`, `$foo['key']`
- No **declaration of variables**: variables are always created when they're first used. And now try to find an error.

Conclusion:
Both JavaScript and PHP have their issues. But JavaScript has a clear foundation (closures and prototyping), whereas PHP is just an unstructured conglomerate of half-understood and badly implemented features. In JavaScript you can easily escape the bad parts by following simple rules and learning its widely missunderstood philosophy. I

libraries, strange syntax and lack of concept.

[1] http://www.felixcrux.com/posts/d...
[2] http://eu.blizzard.com/de-de/gam...
[3] http://www.php.net/manual/en/typ...
[4] http://www.reddit.com/r/programm...

Updated Oct 2, 2015 • View Upvotes

Related Questions

More Answers Below

Why doesn't this JavaScript destructuring assignment work?

Why does PHP have a bad reputation?

Why doesn't the JavaScript drawImage() function work?

Why doesn't JavaScript have an integer numeric type?

Why doesn't JavaScript have default functions for cookies?

Carlos Ribeiro, Curious about languages since 1981.

45.2k Views • Upvoted by Mattias Petter Johansson, Developer at Spotify

Attributing Javascript success to it being "the only language for the job" is unfair. Javascript is hands down a much better language than PHP. At its core it's a fairly consistent language, which follows a few powerful principles. It's a truly dynamic language, prototype-based, that supports functional constructs, all with strong theorethical basis on older languages such as Self and Scheme. Javascript may have its problems, but at its core is a very coherent and elegant language.

PHP on the other hand was from the start much more hackish in nature. PHP was never intended to be a programming language, much less a object oriented one; it started as a hack to build dynamic pages, which solved this particular problem fairly well, and was easy enough for anybody to try. Most of its more advanced features were tacked over it as it evolved with little 'theoretical' consideration. This can be easily seen in the progression from PHP4 to PHP5 and the now dormant PHP6 project. As a result, the language does not have a coherent design.

The main problems with Javascript are in my opinion the name (which imply that the language was derived from Java, which is not true in any sense) and the DOM (which is in my humble opinion a terrible hack glued over HTML+CSS code). All the rest are conscious design decisions, which may be debatable as a matter of application or even taste, but have their own merits. The same can't be said about PHP, despite how popular or effective for the job it may be.

Written Mar 8, 2011 • View Upvotes

Vilan Natanzon, Internet Intellectualizer

3.2k Views • Upvoted by Kartik Ayyar, Google Engineer, Tesla Fanboy

JavaScript's reputation is bad enough. Google put a lot of support behind it because their business depends on the web and browsers. I remember an interview with one of the head V8 engineers and he showed no love for JavaScript. He felt it's the language they're stuck with.

Google's Web Toolkit compiles Java into JavaScript, which is stretching it because they're distant languages. They named their JavaScript framework "closure" because they wanted it to be the end of the multitude of half baked libraries strewn over the web. These are indicative of their attitude towards the language and the community, which were messy indeed.

What saves JavaScript is the original designers were people who cared about computer science, languages and programming. Rasmus, the original designer of PHP has a different, but perfectly understandable attitude about these matters.

From http://en.wikiquote.org/wiki/Ras...

> We have things like protected properties. We have abstract methods. We have all this stuff that your computer science teacher told you you should be using. I don't care about this crap at all.
>
> There are people who actually like programming. I don't understand why they like programming.

Such a developed sense of humor must inhibit programming ability.

With considerable effort, JavaScript can be turned into a presentable language. CoffeeScript http://jashkenas.github.com/coff... has gone a long way towards that, the programming experience is closer to Python and Ruby once the ugly syntax is cleaned up.

Even more effort was devoted to JavaScript's virtual machines. It's now faster than every "scripting" language except Lua. Node.js is designed for current needs like serving many simultaneous requests from one server and sites with constantly changing content. It happens to use JavaScript because there were so few JavaScript libraries for the server, it allowed them to work off a clean slate.

PHP doesn't get this kind of attention. Facebook made HipHop PHP, which speeds up PHP execution http://developers.facebook.com/b... but last I heard, they'll be rewriting large portions in C++.

Written Mar 9, 2011 • View Upvotes

Santosh Rajan, Entrepreneur. Crazy Indian Software Engineer
2.6k Views

I started programming in the early eighties with z80 Assembly language and then went through basic, cobol, dbase, c, c++, lisp, scheme, postscript (yes it is a language), perl, javascript, php, python, clojure and god knows what other language (can't remember off hand now). Let me tell you something.

I found only three languages out of them, with a certain kind of elegance about them.

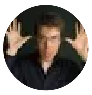
First is of course Lisp (and any of its dialects). The elegance of Lisp comes from the fact that it is not a language that follows the machine language paradigm, like all other languages. Lisp as its name suggests follows the List Processing model.

The second is Python, whose elegance comes from its clean semantics. Its lack of curly braces and other such constructs makes it very easy on the eye.

And the third is Javascript. Yes! After all the Javascript bashing that has gone on here, let me tell you that it is one of the most elegant languages out there. The Javascript Ninja's will swear by it. It has closures, is functional, and prototype based, which allows you to do some very nifty stuff. You want to do OOP? No problem. You want to do functional programming? Be my guest.

The reason Javascript got a bad name is because it is the default and only language of the browser. Since it is the only language available on the browser, all kinds of folks including non, novice, average and expert programmers program in Javascript. Javascript has this burden to bear. Thanks to the Javascript Ninja's, who have kept this great language from sinking with the burden of the novices and non programmers.

Written Mar 18, 2011 • View Upvotes



Kevin Turner
1.6k Views

Douglas Crockford has spoken (and written) a lot about JavaScript's history, design, and pitfalls. See http://javascript.crockford.com/ and his lectures presented by YUI theater, including http://yuiblog.com/crockford/

There's a lot of material there, but I've found it an excellent resource on the strengths and weakness of the language.

(Personally, I wouldn't say JavaScript is a *better* language than PHP; for example, they both struggled for a long time with lack of any coherent namespace mechanism. But they are different. While modern javascript libraries and development environments has made it a tolerable medium to work in, I still wouldn't choose either PHP or JS in any environment where Python or Ruby was an option. (Or C♯ or Scala or, well ... yeah, I'd probably even take Perl over JS. Which is to say, pretty much anything, really.)

Written Mar 10, 2011 • View Upvotes

Max Roeleveld, code ninja
1.5k Views

Javascript is better designed than PHP is, simple as that. The PHP team and community are working towards improving the language, and it really has improved in the last couple of years, but it has a lot of baggage that can't be gotten rid of without massive backwards incompatibilities. You could argue for or against that being a reason, but there you go.

I work in PHP every day, have done so for the last 10 years or so. Are there languages that are better designed? Sure there are. Javascript has a few things I wouldn't spit on having in PHP, Python has a syntax that approaches art, and so on. But at the end of the day, PHP *is* perfectly usable to write code and get stuff done in. I think most of the hate comes from not really knowing the language (or maybe having run into one of its rough edges too often...)

Written Aug 31, 2014 • View Upvotes

Manuel Lemos, Developer of the PHPClasses.org site
1.7k Views • Upvoted by Mattias Petter Johansson, Developer at Spotify

The fact is that PHP is extremely popular. It attracted many people that did not have a background in programming to develop sites. For many of those people developing in other language would not be as easy as in PHP. If you think about it, the basic PHP hello world page can be just HTML with no PHP code at all.

What really has bad reputation is not so much the PHP language itself, but rather the people that it attracted which do not have any background in programming.

Also the lack of qualification of many PHP developers also pushes down the average cost of development of PHP sites, making PHP more attractive to companies that want to lower their budgets.

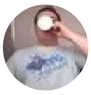
Of course this is not to say that the majority of PHP developers is underqualified. There are PHP developers of all levels of qualification. Obviously qualified PHP developers which are available for hire, are really hard to find because most of them are already working for companies that pay them well.

These are facts that make PHP a very popular language. Fans of other languages that do not understand why PHP is this popular, often engage is anti-PHP campaigns in the hope to attract more developers to the languages of their preference.

Obviously that is a strategy that will not work because most developers work on markets that pay them better, and more than often PHP is that well-paying market that keeps them around PHP.

As for JavaScript, every Web developer uses JavaScript, regardless of the server side language that they prefer. That is why you do not see any anti-JavaScript campaigns, like you see against PHP.

Written Mar 8, 2011 • View Upvotes



Zac Bowling, Software Engineer
1.1k Views

Javascript and PHP both have received some bad stigmas. You really have to look at it deeper though. Everything has a place and every language has some kind of problem.

PHP has a bad wrap because of weird security issues in the past with the basic design, inconsistent style with some of the modules, and lack of concurrency and threading support. Also like BASIC, it attracts a lot of hobbyist and self taught programmers that have a tendency to push out some less than great code which has added to the negative stigma around PHP.

I don't really hate PHP. I often prototype in it. Usually though I know I will need to deal with doing background/batch work, so I like to use a language that has threading or some kind of concurrency support. I know I can work around it in PHP but it's nice not having too. Also the frameworks and libraries that exist today in other languages these days offer some nice benefits. It usually sways me away from PHP each time when evaluating what technology I should use for the job.

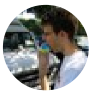
When it comes to Javascript, I still have few reservations of my own that make cringe, like the lack of big integer support, prototypal inheritance, and no built in clean way of handling binary data (but then python < 3.0 has this problem by storing byte arrays and strings in the same construct).

However, bridging the client language and the server language does have a few benefits. Node.js is a unique monster. The evented server model lends itself to doing websockets and http long polling in a very nice way. Being purely asynchronous and not bogged down by existing libraries that are purely synchronous makes it compelling for me. The fact that I can write a http long poll chat server in 30 minutes is incredible. The fact that share the same code for form validation and templates building on both the client and server side has some really great benefits with reducing complexity.

Javascript isn't lost though to it's legacy issues. It has room to improve and innovate within it's current design without breaking compatibly. Take a look at at coffee-script. It's cleans up a bunch of the hard edges and gives javascript a ruby/python feel and it runs on top and can compile directly to javascript. Then you have CommonJS which adds modularizing and namespacing.

After that you are left you are pretty much left with the same arguments around using ruby and python and having a dynamically typed language vs a statically typed one.

Updated May 21, 2011 • View Upvotes



Scott Schlegel, Web developer
1.3k Views

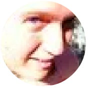
This is more about options. When the choice comes to use or not use php, you have many strong options for server side languages. With Javascript you are pretty much stuck with it. Flash is the only thing that comes close and it has its own problems.

Written Mar 9, 2011 • View Upvotes

Tiago Celestino
815 Views

Today Javascript is better (or no!), but last years the company and communites developers don't works for upgrade language.Today we seen Google Chrome Dev Team, Mozilla Team, Opera and so Microsoft works for upgrade javascript on your browsers.

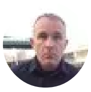
Written Feb 19, 2013



Sean Canton, Self-Taught. Coding professionally since 1999.
820 Views

I would simply say that Javascript core has not been overloaded with loads of libraries with conflicting syntax. Being prototypical has made modules part of JavaScript developer culture, and this allows us to have a minimally functional core, and extend the language as needed for a given project.

Written Aug 28, 2014 • View Upvotes



Michael Stearne, PHP since PHP/FI
234 Views

Languages/frameworks, etc. gain momentum by the users of it and the creators many times.

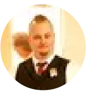
JavaScript was embedded into every browser for the last 15 years so many casual programmers learned its name and something about JavaScript. Some of these developers turned it into a (for the second time) server side language. You suddenly have one language to learn. Then with increased adoption by Google, Yahoo, Twitter, etc. JavaScript grows.

Syntactically neither JavaScript or PHP are probably better than C# but JavaScript has gotten blessed by the right people at the right time.

We'll see how it grows or declines but PHP will remain for me and I'll increase my use of JavaScript.

Written Jan 16 • View Upvotes

Andy Boyd



2.2k Views

In short, because PHP has been in use for 15 years in environments that include bottom-end shared webservers. It grew organically as the internet became mainstream, so it has stumbled through a lot of ugly phases and has been around long enough to be complained about. Javascript, until the last few years, had been mainly used for browser scripting - it didn't have to bumble through the early days of the public internet.

I would not necessarily call Javascript a 'better language'. It is basically the only language in common use that is prototype based/classless, which means developers with backgrounds in other languages have to learn a completely different philosophy of object orientation in order to write complex Javascript. The up side is that it is probably more easily adopted by front-end developers who don't want to learn a new language to code at the server.

Written Sep 28, 2015

Nicolas Sebban, Analytics at Orange, innovative side projects, passive income afficionado, co...
482 Views

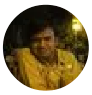
Javascript is very much liked lately, and it seems to be it's because anyone with a tiny background in IT can do "nice" things, by integrating one of the many available libraries. Exactly just like PHP used to be.

But it has flaws that lead people to write bad code, and the lack of tools makes it harder to fix, let alone optimize. Exactly just like PHP used to be.

I'm pretty sure JS will someday be judged just as bad as PHP. Or maybe far worse. But by then, it will be almost universally used. Exactly like PHP is right now.

Note : PHP and JS are the two main languages I use.

Written Dec 23, 2014



Saurabh Gayali
412 Views

I am a bioinformatics, biotech and graphics guy.
just started coding for servers.
I tried Javascript learning and Its very tough as compared to PHP.
So from beginners point of view PHP is good.
While I came through one limitation of PHP that is overcome in Javascript. PHP effects cant be realtime shown on page while JS effects are dyanamically loaded on the current page.
That awesome but again its hard to code.
And most of the times the blurred line between pure JS and JQuery and Angular JS confuses as there are many answers to same thing in JS community.
PHP has a defined path of tutorials at least. But again they are limited and most of the time not useful.
But its easy to try in PHP.

And most of all, For pur JS you dont need to install a thing. Mere JSFiddle would do. though for PHP ther eis PHPfiddle but very limited functionality. Thats because of PHP security issued the server has a limited functionality on purpose.

Written Aug 24, 2015

Bryan McGuire
393 Views

I think the bad reputation of php stems from its early history.

1. javascript never had clumsy crap built into it like "array()"
2. server-side javascript never existed in a form where every single variable was global in scope, not even when it was a part of Netscape Enterprise Server.
3. server-side javascript never allowed web clients to overwrite arbitrary variables using parameters in the query string, not even when it was a part of Netscape Enterprise Server.
4. javascript was made by professional programmers working at a large company (Netscape), and not by a hobbyist.
5. server-side javascript was always able to handle sessions, unlike php which needed 3rd-party hacks like phplib.
6. javascript always had the object oriented concept. This didn't get into php until much later.
7. Database support in early php didn't have the concept of bind variables, but relied on the programmer to use functions like "addslashes()" for quoting input.

Basically, early adopters of php were pretty much rolling out the welcome mat for people to rootkit their servers. I mean, it was so bad that it made Perl look professional by contrast. It's gotten better. It's gotten much better, but even after 15 years some of that early stigma still remains.

Written Sep 29, 2015 • View Upvotes

Cynthia Avishegnath
208 Views

You actually asked three different questions.

1. What makes JavaScript a better language?
2. Why doesn't JavaScript have the bad reputation over PHP?
3. PHP being badly designed. JavaScript seems quite a similar but is widely adopted by industry leaders such as Google.

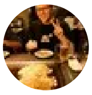
Answers:

1. JavaScript is NOT a better language than PHP. Both are crappy languages. Both were shortsightedly invented of/for the moment, and bandaided languages.
2. JavaScript does have a very, very, very .... very bad reputation.
3. There is no choice. There is only one real browser language. It is a blimey monopoly. If JavaScript were a corporation, it would have been hauled into a US federal court for non-competitive activities.

JavaScript is like Daesh/ISIS - everyone hopes it will die away and disappear, but it is not in any one of the computing super powers to remove it. The weaponization invested behind its weaknesses having translated into competitive advantages. JavaScript is a computing terrorist, a nightmare.

Like those adversarial politics behind non-committal of Daesh to a demise, those superpowers behind non-committal of JavaScript to its demise, whether unwillingly or otherwise are acting irresponsibly and wasting a lot of valuable computing skill-hours - which on the flip-side increase our collective employability.

Written Mar 15 • View Upvotes



Eric Florenzano, CTO of Convore
1.4k Views

I truly suspect in 5 years it will have just as bad a reputation once more mediocre developers jump onto the node.js bandwagon and write unmaintainable spaghetti code.

Written Mar 8, 2011 • View Upvotes

Dustin Davis, Web Developer, Sports fan, Husband & father of 4
703 Views

Stockholm Syndrome.