Javascript Controversy

Javascript is in an odd spot within the programming world. It’s an extremely unique programming language that’s become ubiquitous with web applications to the point that people often forget it can be used outside the web. Despite javascripts inadequacy when compared to almost any other language it’s the only option when it comes to adding functionality to a website. This fact alone has made javascript into the highly popular language it is.

Most complaints against javascript come in the form of security issues and pure disdain for the inner working of the language. As David Arno points out in his article “Why JavaScript is a toy language” javascript lacks several key components of your average language. Without classes javascript fails to meet the basic requirements of an object oriented language. The inability to compile leaves all your javascript code open for anyone to use. Not only is this a security risk it makes it impossible to keep your code your own (Arno David).

It’s been debated whether javascript should or should not be a part of web development. Javascript is the only option if you want to add features not possible within HTML and CSS, but does a site ever actually need these extra features? Javascript only serves to make sites more bloated and open security risks with potentially exposing data. On the other hand, why not use it? Assuming your not handling sensitive information or demanding optimization is there really any reason you should care? Despite its flaws it’s still a fully functional language that can add some spice to your site.

Die hard programmers tend to argue against the use of Javascript. Instead pushing for literally any other language to replace it. These are the people that are so far into the programming world that they find it harder to work with simplified versions of what they’re used to. Despite this the language does exactly what is says and provides exactly what is needed. Why would your average programmer or user care about better options when they already get what they want. To anyone outside the inner programming circle, there is no controversy over javascript.

Though the use of Javascript might be affecting them whether they know it or not. Javascript is riddled with security issues. Along with the aforementioned inability to hide your javascript code, this also makes it impossible to hide data within the javascript. Because of this reason you often find sensitive information being handled with PHP instead. But privacy isn't the only risk you face with Javascript. Javascript can be used to run malicious code on a users computer. Like running a script on page load that attempts to download a virus. Any site that uses javascript has the ability to do this. Even an innocent site could be hacked to force a malicious script.

While PHP is your best bet in solving these security issues, Ajax can be used to add a layer of obscurity to your data. Ajax is a technique of calling data from a server and presenting it on a page using Javascript. This removes the data from the pages source and hides it off on a server. That being said the data is still sitting there on the server for anyone to access if they put in the effort to dig it up. To truly secure your data it needs to be encrypted, this is candone through the use of HTTPS or manually done via custom code.

Ajax can be used to store data in various formats. Json and XML are two of the most common formats. Both serve the same purposes with different levels of complexity. XML allows for unique tags to identify data and organize it in an easy to read structure that defines parts in human language. Json uses the bare minimum, storing data in an extremely basic format that relies on formatting more than naming and tags.

On the less important side of things Javascript also has the issue of an open source code library. While languages like C# use the .net framework to provide a filtered set of useful class libraries, Javascript relies on the community to provide useful bits of code. As seen in the article “How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript” by Chris Williams this can cause some problems. When a library of code is suddenly removed, every site relying on it will most likely break completely. This was the case when Koçulu shut down his code for left padding text (Williams Chris).

Though this open code library also provides an expansive set of tools to coders looking for more options. Javascripts simplicities often make more complicated actions far harder than they need to be. Luckily the magic of open source means almost anything your trying to do has already been done by someone else and is ready for you to use. Jquery stands out as one of the most popular and complete libraries of code that you will find used all over the internet.

Personally I find it hard to find a situation where javascript genuinely improved the experience of a website. On the other hand it’s still fun to mess around with such a simple language and make easy to use games that can be pushed strait to my website. I recognize the limitations and understand the complaints. If I want to make an application outside of my browser javascript would be my last choice. But javascript is far from useless and definitely makes a good learning tool.

I don’t see javascript moving far from where it is now. It’s universally known as a third of the web development trifecta. While a minority of programmers take issue with it and seek out alternatives, I can’t imagine a sudden shift to a completely new language that would require a complete rework of many sites currently live on the web. At most I could see some alternative eventually reaching javascripts popularity and competing with it.

Works Cited

Arno, David. “Why JavaScript Is a Toy Language.” *The Random Utterances of David Arno*, 18 May 2010, www.davidarno.org/2010/05/18/why-javascript-is-a-toy-language/.

Williams, Chris. “How One Developer Just Broke Node, Babel and Thousands of Projects in 11 Lines of JavaScript.” *The Register*, 23 Mar. 2016, www.theregister.co.uk/2016/03/23/npm\_left\_pad\_chaos/.