A Brief Introduction to JavaScript

February 11, 2019

Contents

1 Introduction

JAVASCRIPT is a popular, high-level computer programming language most commonly used in the context of web development and design. It incorporates a dynamic typing style.

JavaScript was reportedly created in 10 days[4] and finished on December 4th, 1995 as 'Mocha' by Brendan Eich, at the time an employee of Netscape. Later iterations had used the name LiveScript. When its first official version was released in 1997, the name was settled as JavaScript—as it is called today. It was used in all of the company's proceeding web browsers beginning with Navigator 2.0.[1]

JavaScript and its other ECMAScript derivatives remain the sole, most-popular scripting paradigm for web development today. It is considered, among other things, as Brendan Eich once envisioned, the "glue" that holds the Net together.

2 Origins

Contrary to its name, JavaScript has close to nothing to do with **Java**. Standard, vanilla Java is a C-derivative language that was released in the early nineties—it had been created with a different end goal and was not anticipated for web-based development.[7]And, making things worse, is its *-Script* suffix. Scripting languages are usually much simpler and much less versatile than actual programming languages; they aren't compiled. JavaScript, however, has since evolved into what many may now consider a real, general-purpose programming language with the sheer amount of features it has garnered—it is, however, still called Java*Script* despite its gradual growth out of what would put it in the scripting category. It's name no longer truly, accurately represents its contents. The title doesn't quite match the pages.

To take a step back, Brendan Eich had originally joined the Netscape Foundation to implement the Scheme programming language, or at least a derivative, into the browser; likewise, a language similar to Scheme was then created. Higher-ups at Netscape, however, pushed for a language instead syntactically similar to Java (perhaps due to a long-standing partnership with Sun Microsystems).[8]

Then, leading to the creation of JavaScript, Brendan Eich said to have wanted to create a "simpler language [than Java]" that could easily be picked up by any beginner.[4] This is evident in its lack of necessity for semicolons (used in C and C-derivative programming languages to end statements or declarations); however, that does not mean semicolons cannot be used—in fact, their usage is *heavily encouraged for beginners* in order to develop good typing habits early-on. JavaScript, in fact, is often criticized for its "loose" or "weak" programming syntax.

But Eich's intentions seem to be holding up well today: many who are new to computer programming, besides learning simple markup languages, now unquestionably start with or are taught JavaScript (perhaps alongside Python) in their journey of learning computer logic.

2.1 ECMA-262

The JavaScript that Brendan Eich had created for Netscape later became ECMAScript, officially known as ECMA-262. The 'JavaScript' language is now considered a **dialect** of ECMAScript, whereas the original iteration Eich was creating went on to become ECMA-262. To reiterate, as the original JavaScript went on to become an international ECMA standard, what was left of it went on a path of its own while following that standard—today's JavaScript, then, is a derivative of what used to be itself.[11] As a quick example, modern browser vendors distribute the latter JavaScript—the derivative—with their browsers. While ECMA-262 can sometimes be used for such purposes as well, it cannot accommodate the sheer amount "custom" features held by JavaScript because it serves as a standard, not just a language.

JavaScript is the **language**, ECMAScript is the **standard** that one such language must follow. A scripting language must observe and follow the stated guidelines for it to be considered ECMA-262 compliant.

Of course, the guidelines stated in the ECMA-262 do not require a given language to conform to arbitrary, far-reaching, intricate rules—that would defeat the purpose of creating languages to conform to the ECMA-262 because it itself would then be a language. These guidelines, rather, define basic rules or restrictions that ensure a given scripting language is universally usable on all systems and is the same across different systems based on the standard.

Consider a hypothetical situation in which one popular web browser distributes scripting language 'ABC' with its own paradigm, its own syntax. Say, then, another language called 'XYZ' with an entirely different paradigm and syntax pops up with another, equally popular browser. Both would be in conflict and the internet would be split with whichever of the two languages a given website prefers to use.

In fact, exactly that happened in the early days of *Netscape Navigator* and Microsoft's *Internet Explorer*.

2.2 Browser Wars

In 1996 Netscape Navigator, at the time a paid web browser, was dominating the market. But there was a new browser in the industry, this one from Microsoft—it was freely integrated into Microsoft's Windows, the long-time-standing dominant operating system. It was later available as a free download. This and Windows' market share set up the newly-developed Internet Explorer for widespread success for years to come.

Consumers at the time who were in need of or ready to buy a computer were highly likely to buy a Windows computer. Moreover these consumers were, almost without a doubt, *not familiar with the internet* or any web browsing software; meaning they had nothing to compare Internet Explorer, their default browser, to. This scenario, compounded in the millions, very quickly gave Internet Explorer the majority market share.[16] This also led Microsoft to court in an anti-trust case, heavily accused of anti-competitive behaviour.[23]

But both Internet Explorer and Netscape Navigator *had conflicting methods of rendering certain parts of the web.* This made it common for websites to place banners, saying "Best Viewed With Internet Explorer," or "Best Viewed With Netscape" with corresponding links to download the browsers. Both were in conflict, and the Net's sites were divided among which browser they were best suited for.

Later, a 'web standards initiative' was put in place, advocating that browsers conform to certain standards set by the **World Wide Web Consortium** (W3C) in order to unify the Net. Banners saying "Best Viewed With Any Browser" were then popularized.[16]

3 Riding on Java

The reason behind JavaScript's naming was, as Eich himself remarks, no more than a "marketing scam" by Netscape.[12;13] Getting permission from **Sun Microsystems** to call this language *JavaScript* was a move to garner attention—therefore widespread usage and acceptance—and ride on the popularity of the then-famed and newly-released *Java* programming language. Making matters worse was Eich reportedly ordering the newly-named JavaScript to be made to "look more like Java."[19]

Was Java so popular, so anticipated to be used, so as to influence the naming of another language? Indeed it was. Java is a C-like, object-oriented, class-oriented general programming language. Among these rather ordinary attributes, however, was its famous promise of "write once, run anywhere."[7] *Sure,* one replies, *but what does this mean?*

"Write Once, Run Anywhere," written short as WORA, was a slogan popularized by Sun Microsystems to illuminate the vital cross-platform capabilities of Java.[14] Like C and C++, Java is compiled. Java, however, has a *virtual machine* or an abstract, simulated computer architecture called the JVM, short for Java Virtual Machine. This virtual machine takes Java Byte code, which top-level Java code is compiled into, and *interprets* it into localized machine code in order to let it run on the host computer or device. Ideally, Java would allow a developer to write software on one operating system and run it on another without issue—of course, the target operating system must have a version of the JVM developed for it as well.

Java was, and still is, an excellent language capable of both high and (relatively) low-level programming tasks—it is a versatile language quite on-par with all the rest of the general-purpose languages out there. In 1995, though, its prized-feature was its neat and sophisticated cross-platform capability.

"LiveScript," as it briefly had been, sought the permission to name the language "JavaScript" in order to catch the eyes of onlookers, under the effect of the Java hype, and potentially reap in enough usage and attention to become the dominant web-scripting language.

4 Syntactic Criticism

JavaScript is often praised for being both easy to understand and trivial to work with. But not everyone enjoys using it, some hate it, and it certainly isn't perfect.

Why? Some dislike it because they feel it extends too far beyond its purpose, like being used in back-end work as well as front-end work.[17]

Remember that JavaScript is a very high-level language. That means that its syntax can be much more 'loose' and can be more lenient than that of various lower-level languages such as C or, at its extreme, the **Assembly** language. Having a loose syntax (or loose 'type') may open up possibilities for programs produced by it to have vague syntactic errors—this situation can occur when, for example the language *implicitly converts* data types.

In JavaScript, one can declare a variable with an arbitrary name.

```
var a = 1;
console.log(a);
```

In the above example, a variable *a* is created and assigned a value of 1. We can proceed to print *a* and receive 1 in the console. The code, then, *implies* that a is an integer. But when we assign "Hello, World" to the variable,

```
a = "Hello, World";
console.log(a);
```

we duly receive "Hello, World" in the console *despite having initiated the variable with an integer*. This would, had we tried this in C, throw an error during compilation.

This means that variables in JavaScript determine their type by the value which they are assigned. The programmer doesn't need to declare its type; it is implicit and dynamic.

But that's just about declaring variables. The real issue begins when variables are *converted without warning*. This can happen anywhere. A common case is, when a user enters a value into a form, that value is taken as a string literal unless converted explicitly. Imagine a user had entered "5" into a form and that went in to a variable *x*.

```
var x = [form value];
console.log(x + 1);
```

[18]

Expecting 6? The actual result most likely would have been "51." Without an explicit conversion, the result is read as a string.

But none of this is utterly invariable; criticism, after all, is usually subjective. It is the job of the programmer to fit his planned work in to the syntax of the programming language with efficiency and functionality—perhaps organized as well—and it is the job of the programming language to be versatile, extensible, usable and reasonable.

5 Retrospect

Being a derivative of ECMA-262, JavaScript itself has now a countless number of derivatives and variations. Vanilla JavaScript, as its unmodified variant is called, is used alongside HTML in the creation of websites in order to make the pages more interactive and dynamic, able to more easily process and manipulate information. It can be used to add, remove or move page elements, take form input data, issue a page redirect, and more.

Now the dominant scripting language for the web, JavaScript is used in **over 95.1%** of all websites on the Net as opposed to Java, Flash and Microsoft's Silverlight.[20] If that percentage means nothing to you, consider the total amount of websites on the Net. A hundred-thousand? A million? 100 million? No. According to a January 2018 survey by Netcraft, the estimate for the total number of hostnames on the Net end up at **over 1.8 billion**.[22]

But that's just an estimate derived from a survey—which probably means that there are even more. This means (though somewhat of a stretch) that *ninety-five percent of an estimated total of 1.8 billion websites use JavaScript in one way or another.*

That's a lot.

***

Quotes from the ’90s:

> *“JavaScript and Java represent important steps in the evolution of the Internet and Intranets for business computing.”*

— Nancy Li, Executive Vice President and CT at Computer Associates.

> *“JavaScript is an exciting technology because it represents the next generation of software designed specifically for the Internet.”*

— Jan Silverman, Director at Hewlett-Packard.

> *“The Java and JavaScript languages will serve an important role in allowing Internet applications to take advantage of enterprise client/server computing.”*

— Mitchell Kertzman, Executive Vice President and CEO at Sybase’s Powersoft Division.

> *“The creation of a general, standard scripting language for Java development will accelerate adoption of this new, exciting technology for delivering dynamic, live content to the consumer.”*

— Greg Galanos, President and CEO at Metrowerks, Inc.[21]

You may use and/or distribute this article in part or full for educational purposes. Written by the author of https://elaboraet.neocities.org/.

References

[1] Wikipedia, JavaScript; https://en.wikipedia.org/wiki/JavaScript

[2] Wikipedia, ECMAScript; https://en.wikipedia.org/wiki/ECMAScript

[3] ECMAScript Specifications Document;
https://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf

[4] Stack Overflow; Difference Between JavaScript & ECMAScript;
https://stackoverflow.com/questions/912479/what-is-the-difference-between-javascript-and-ecmascript

[5] W3CSchools, JavaScript Starter Page; https://www.w3schools.com/js/default.asp

[6] "History of ECMAScript,"
http://www.benmvp.com/learning-es6-history-of-ecmascript/

[7] Wikipedia, Java; https://en.wikipedia.org/wiki/Java_(programming_language)

[8] "JavaScript: How It All Began,"
http://2ality.com/2011/03/javascript-how-it-all-began.html

[9] Brendan Eich, Wikipedia;
https://en.wikipedia.org/wiki/Brendan_Eich#Netscape_and_JavaScript

[10] Medium; Difference Between JavaScript & ECMAScript,
https://medium.freecodecamp.org/whats-the-difference-between-javascript-and-ecmascript-cba48c73a2b5/

[11] Stack Overflow, "What is ECMAScript?"
https://stackoverflow.com/questions/4269150/what-is-ecmascript/

[12] "Brendan Eich – CEO of Brave,"
https://www.youtube.com/watch?v=XOmhtfTrRxc&t=2m5s

[13] "What's the Difference Between Java & JavaScript?"
https://www.upwork.com/hiring/development/java-vs-javascript/

[14] Wikipedia, "Write Once, Run Anywhere,"
https://en.wikipedia.org/wiki/Write_once%2C_run_anywhere

[15] Scripting Languages & Programming Languages, https://stackoverflow.com/questions/17253545/scripting-language-vs-programming-language

[16] Wikipedia, “Browser Wars,” https://en.wikipedia.org/wiki/Browser_wars

[17] HackerNoon, “The JavaScript Phenomenon is A Mass Psychosis,” https://hackernoon.com/the-javascript-phenomenon-is-a-mass-psychosis-57adebb09359

[18] “JavaScript quirk 1: Implicit Conversion,” http://2ality.com/2013/04/quirk-implicit-conversion.html

[19] ‘QuirksMode,’ “JavaScript – General Introduction,” https://www.quirksmode.org/js/intro.html

[20] JavaScript Usage Statistics, https://w3techs.com/technologies/details/cp-javascript/all/all

[21] “JavaScript’s Launch in 1995,” https://medium.com/@briantunison/the-year-of-eich-adumbrating-quotes-from-javascripts-launch-in-1995-430163a05551

[22] “Jan. 2018 Web Server Survey,” https://news.netcraft.com/archives/2018/01/19/january-2018-web-server-survey.html

[23] Microsoft v. United States, Anti-Trust, https://www.nytimes.com/2000/04/04/business/us-vs-microsoft-overview-us-judge-says-microsoft-violated-antitrust-laws-with.html