JavaScript is the English of programming languages

published on February 22, 2023

I code primarily in JavaScript,[1] a bad programming language. It was created in 10 days back in 1995,[2] and is unsurprisingly littered with flaws.

JavaScript's strange behavior is a common joke among programmers. Most infamously, it has three different ways to check for equality (==, ===, and `Object.is`),[3] two incompatible ways to represent an empty value (`undefined` and `null`),[4] and did not have a way to define a block-scoped variable for the first 20 years of its existence.[5] And those quirks are just the start! There are so many questionable "features" that any valid JS code has an equivalent expressed with only `!`, `[`, `]`, `(`, `)`, and `+`.[6]

Despite all this nonsense, though, JavaScript is immensely popular. It has vied back and forth with Python over the last few years for the mantle of *most-used* programming language![7]

I also write in English, a bad human language. "Writing in English is like throwing mud at a wall"—everyone who's ever tried to do so will agree.

There is little rhyme or reason to this chaotic tongue. After a thousand years of stealing every useful word or rule that they encountered, our linguistic predecessors passed down to us a patchwork abomination that's almost impossible to "fluently" learn unless you grow up with it.

Typically illustrative is the common rhyme: "I before E, except after C." Guess what? This rule is *exactly* incorrect: out of 350,000 common English words, a researcher found that 75% break this rule.[8] You're three times as likely to be right if you *invert* our most famous grammatical mnemonic than if you follow it!

And yet… English, too, is dominant. More people speak it than any other language,[9] and this understates its triumph: it is the common tongue of educated humans, the world's default language. When Chinese and Japanese businessmen need to communicate, they do so in English. The European Parliament, in which no country primarily speaks English, nonetheless uses it to debate and pass continental laws.[10] Even the syntaxes of almost all global *programming languages*—including JavaScript!—are littered with English words and therefore almost unusable to non-speakers.[11]

What is to explain these similarities of my main two languages? They are both so ugly, but universal. So messy, yet omnipresent. Why haven't better languages taken their place? And why didn't nicer ones have it initially?

Well, because they couldn't have. The imperfections of JavaScript and English are *inseparable* from their preeminence. They are similar in both of these aspects, because they have the same three-part story:

1. They did not gain prominence because of any linguistic merits, but were boosted to the top by attachment at the hip to a conquering force.
2. Their popularity extends even beyond their sponsors' territory because they are Schelling points: everyone uses them because everyone uses them.
3. Their massive user bases make many variations, which enable evolution; both stay competitive by adapting smoothly to users' changing needs.

Allow me to elaborate.

1. Initial spread

The British had the largest colonial empire the world has ever seen,[12] and English was their language. As Britain declined, its daughter America took the global reins,[13] still speaking English. For the last two centuries, whenever people want to interface with global power—in business, science, military, or culture—they have learned English.

English did not spread because it was aesthetically pleasant, and has not replaced Latin and French as the global *lingua franca* because of its (lack of) excellent grammar. No, it rules because its *speakers* have ruled the world for the last 200 years. The need to learn it to work with England and America has, alone, attracted (or compelled) multitudes.

In a similar way, JavaScript was spread by the internet's growth: it is the only language that all web browsers can consistently execute. If anyone wants to make a website, they have to get used to its quirks. Websites are valuable, so people suck it up and learn. Web browsers and the Internet have become humanity's main interface to computer applications; inevitably, as a consequence, JavaScript has been adopted at scale.

2. Widespread popularity as a common tongue

Once a language is among the most popular, it is increasingly chosen on that basis alone, even in situations unrelated to its origin. This network effect has already been established for English, and is gradually solidifying for JavaScript in turn.

English is the primary language of business, and of science, and of the Internet. It's no coincidence, then, that it is the most common second language in the world:[14] if you're going to learn a new language, of course you choose the one that opens up the most opportunities!

So artists and writers and meme-makers and businesspeople operate primarily in English… because of the massive audiences sitting ready to receive and pay them for their content. And two people in strange countries who need to communicate are each more likely to speak English well than the other's language—so they do. This only reinforces its status.

Developers learn JavaScript because, being popular, it has the most resources for beginners. They build web apps in it because they're trivial to distribute them online. They write open-source code for the JavaScript ecosystem because it has the most people to download and improve their packages.

Companies use JavaScript for their codebase because of these factors: JS developers are the easiest to hire, and pre-written JS code—which helps avoid wheel-reinvention—is the easiest to find and bring into projects. Because companies use it, there are tons of jobs in JavaScript, and its packages only get better as well (as people are paid to maintain them).

As JavaScript and English gain popularity, their attractiveness only grows.

3. Enduring evolution

We all invent improvements to our languages, and sometimes these ideas are good and other people use them too. Because English and JavaScript have the most users, they have the most such variation; and therefore the most good new variations; and therefore the most rapid improvement.

On the English side, social media fuels the constant creation and proliferation of new words (or meanings of old words) like "based", "rizz", and "yeet." (They make it all the way to the dictionaries, too— as of September 2022, "yeet" is in Merriam Webster,[15] and Apple Notes no longer yells at me for typing it!) This is memetic evolution: it's just variation and selection. New words spread if they are useful, and old words die if they are not. The speed of this change is a function of the variation, which is a function of the population size: English has the most people, so English is the most flexible and modern language, suitable for talking about the world in real time as it changes.

JavaScript started out horribly, and has gradually gotten better as people fix their frustrations and share their solutions. People got tired of sending huge files over the internet, so in 2012 "Webpack" made them smaller.[16] JavaScript didn't have types; but coders at Microsoft got tired of that and built "TypeScript," a layer that anyone can use that adds them, and released it publicly later in 2012.[17] It's a royal pain to repeatedly put data on a webpage in a good-looking way, so in 2013 Facebook made and released “React,” a better system for rendering.[18] Old browsers kept breaking or hampering developers because of new JS features not working properly, so in 2015 “Babel” started automatically turning nice compact modern JavaScript—that humans could write—into uglier code that still works everywhere.[19] *All* of these tools are available for everyone who builds apps in JavaScript today, and almost everyone uses them. If JavaScript did not have the massive network effect, from its boost by the browser, that got big companies to invest in these open-source solutions to make their coders more efficient, it would not be nearly as good as it is today.

English and JavaScript both started out as crappy languages that you *had* to use to get some important things done. Then people kept using them, because so many others used them. And now they evolve quickly, and are frequently the best tools for expression!

1. Technically, I program in TypeScript, a variant of JavaScript. But it’s all the same family, and generally referred to interchangeably. ↩
2. https://www.computer.org/csdl/magazine/co/2012/02/mco2012020007/13rRUy08MzA ↩
3. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Equality_comparisons_and_sameness ↩
4. https://www.geeksforgeeks.org/undefined-vs-null-in-javascript/ ↩
5. https://www.geeksforgeeks.org/what-are-block-scoped-variables-and-functions-in-es6/ ↩
6. See an explanation of this 6-character JS “dialect”, which has a profane name, here. ↩
7. For this comparison, I lump TypeScript in with JavaScript. My source is GitHub’s blog: https://octoverse.github.com/2022/top-programming-languages ↩
8. https://www.rd.com/article/i-before-e-rule/ ↩
9. https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide ↩

[10] Technically, English is one of three main “procedural” languages that the European Commission uses, alongside French and German: https://ec.europa.eu/commission/presscorner/detail/en/MEMO_13_825 ↩

[11] The dominance of English-based programming languages is not widely known, but almost total. An illuminating example: the only Chinese-native programming language I could find hasn’t had a new release for almost three years, and is almost wholly unused. Its codebase and documentation are written in TypeScript and English, respectively. ↩

[12] In 1938, the British empire controlled 33.6 million km^2 of land (just over 20% of total land area); the closest rival colonial powers were Spain (peaking at 12.3 million km^2 in 1760) and France (peak: 12.1 million km^2 in 1938). Population numbers tell a similar but even more dramatic story: Britain was sovereign over 23% of all humans at its 1938 peak, while the nearest colonial power is France with a paltry 8% way back in 1812. (These data are from Walter Scheidel’s “The Scale of Empire: Territory, Population, Distribution” (2021).) ↩

[13] America has controlled the world’s economy for at least the last 60 years, arguably even more than Britain ever did to the world’s land or people. Despite only having around 5% of the world’s population, in 1960 our GDP accounted for 39% of the world total; in 2020 this figure has declined, but is still an astronomical 24%. ↩

[14] https://www.movehub.com/blog/global-second-languages/ ↩

[15] https://www.merriam-webster.com/words-at-play/yeet-meaning-and-history ↩

[16] first version published 2012-03-11 ↩

[17] first published on 2012-10-01 ↩

[18] the “react” package was taken over by FB with version 0.8.0 on 2013-12-19; there are older package versions, but previously it was some other guy’s project. Before this, the Facebook project was at “react-tools”, first published as 0.3.0 on 2013-05-29 ↩

[19] “6to5” was first published on 2014-09-30 , and renamed to “babel” on 2015-02-15 ↩