

The History of JavaScript: Everything You Need to Know



T.J. DeGroat | 5 minute read | August 19, 2019



In this article

[The History of JavaScript](#)[What Is JavaScript?](#)[JavaScript Origins](#)[JavaScript vs. Java](#)[ECMAScript Is Born](#)[The Paper That Started a Movement](#)[Finding Common Ground](#)[JavaScript Today](#)

The History of JavaScript

JavaScript is everywhere, and for the seventh year in a row, it has been ranked the most commonly used programming language, with 67.8% of developers employing it in 2019. Its ascent to the world's most popular programming language is synonymous with the rise of the internet itself.

Created out of necessity, it is used to build 95.2% (1.52 billion) of websites today, including some of the world's largest, like Facebook and YouTube. Without it, we would not have popular and useful web apps such as Google Maps and eBay.

So, without further ado, let's take a look at what JavaScript is, how and why it was created, and what's next for the language.

What Is JavaScript?

JavaScript is a scripting language that is one of the three core languages used to develop websites. Whereas HTML and CSS give a website structure and style, JavaScript lets you add functionality and behaviors to your website, allowing your

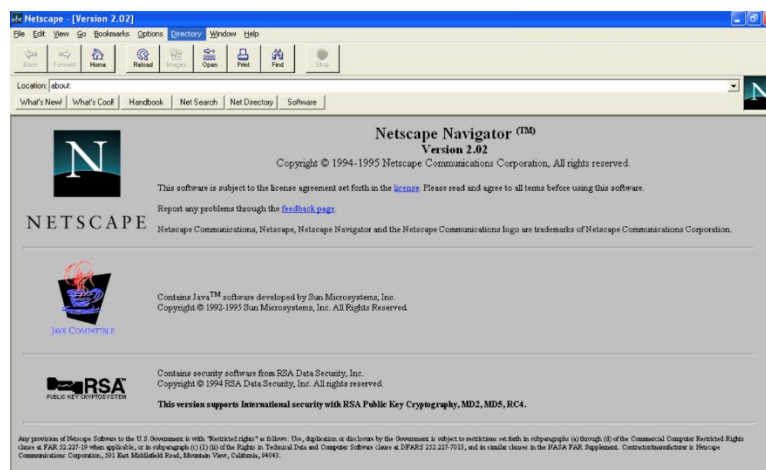


website's visitors to interact with content in many imaginative ways.

JavaScript is primarily a client-side language, meaning it runs on your computer within your browser. However, more recently the introduction of Node.js has allowed JavaScript to also execute code on servers.

Since its release, JavaScript has surpassed Java, Flash, and [other languages](#) because it is relatively easy to learn, has a free and open community, and, most importantly, is incredibly useful, allowing developers to quickly create apps with audiences in the millions.

JavaScript Origins



([Source.](#))

The early to mid-1990s was an important time for the internet. Key players like Netscape and Microsoft were in the midst of browser wars, with Netscape's Navigator and Microsoft's Internet Explorer going head to head.

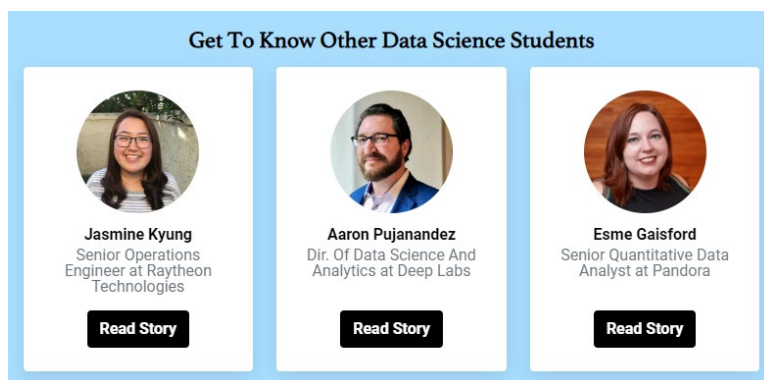
In September 1995, a Netscape [programmer](#) named [Brandon Eich](#) developed a new scripting language in just 10 days. It was originally named Mocha, but quickly became known as LiveScript and, later, JavaScript.

JavaScript vs. Java

There's often some confusion about the two, but JavaScript and Java (another popular programming language for data scientists after Python) have almost nothing in common. The name JavaScript came from Netscape's support of Java applets within its browser. Many say it was also a marketing tactic to divert some attention from Java, which was the most buzzed-about language at the time. To run Java programs, the code must be first compiled into an executable form. On the other hand, JavaScript was created to be interpreted at run time, making it

much more dynamic (these days the boundary of the two methods is a lot more blurred).

JavaScript didn't exactly get off to the best start. It didn't perform as well, and those developing in Java considered JavaScript more of a "UI glue" to be used mostly by designers and other non-engineers. But the reality is that having a "glue" language allowed the internet to really flourish. Programmers could react better to use events and compose interactive components. And due to that, JavaScript spread like wildfire and very quickly became the lingua franca of the web.

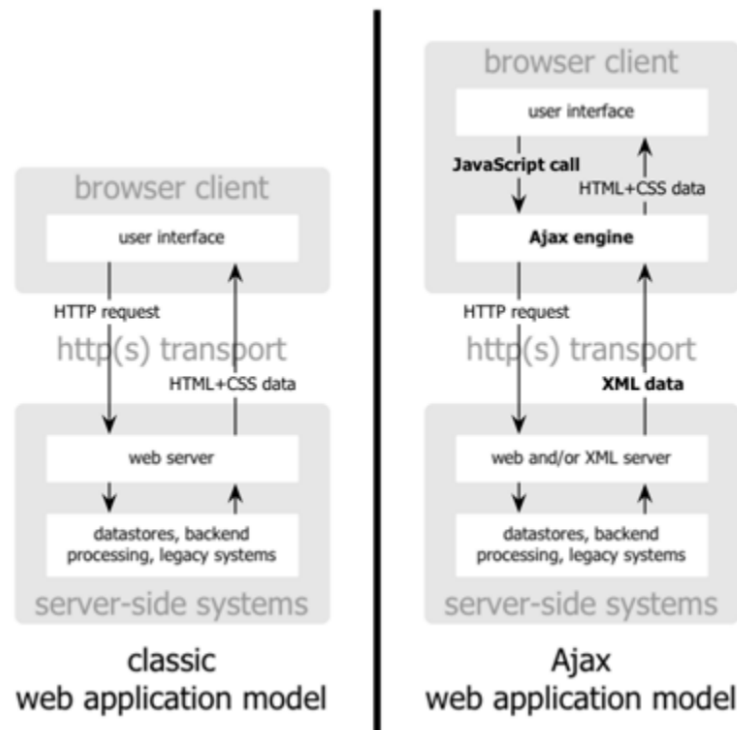


ECMAScript Is Born

In 1997, due to JavaScript's rapid growth, it became clear that the language would need to be properly maintained and managed. Therefore, Netscape handed the job of creating a language specification to the European Computer Manufacturers Association (ECMA), a body founded with the goal of standardizing computing. The ECMA specifications were labeled ECMA-262 and ECMAScript languages included JavaScript, JScript, and ActionScript.

Between 1997 and 1999, ECMA-262 had three revisions, but nearly 10 years later, version 4 was abandoned due to differing opinions on the direction of the language and its proposed features. Interestingly, many of these controversial features, such as generators, iterators, and destructuring assignments, have been included in more recent ECMAScript specifications.

The Paper That Started a Movement



([Source.](#))

The year 2005 proved to be a big one for JavaScript. A paper released by Jesse James Garrett [introduced Ajax](#), a revolutionary suite of technologies that included JavaScript. Ajax vastly improved user experience by allowing web pages to feel more like native desktop apps. This really pushed JavaScript into the spotlight as a professional programming language.

This paper is considered to be one of the founding backbones of the JavaScript community. At the time, JavaScript had many challenges, including its verbose nature when doing simple things and the incompatibility issues between browsers. The community responded with large and popular JavaScript frameworks and libraries, such as Dojo and Mootools and jQuery.

Finding Common Ground

Following a 2008 event in Oslo, the ECMAScript 4 proposals were scaled back by many organizations and parties involved with JavaScript, including Yahoo, Google, and Microsoft. The project was codenamed Harmony and it came to fruition in 2015, when ECMAScript 6 was released.

In 2009, the CommonJS project set out to define and promote JavaScript development outside the browser by using modules to package useful code and functionality. This paved the way for

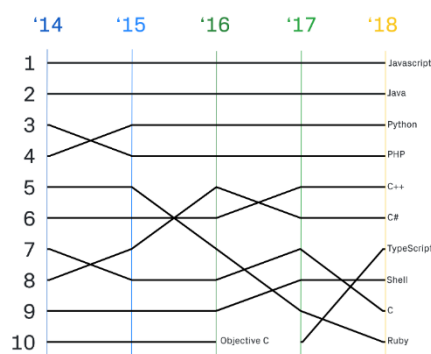
Node.js as an environment to run browserless JavaScript. Now the language that ran the frontend of the internet was able to tackle the servers behind the scenes.

JavaScript Today

Top languages over time

You're coding on GitHub in hundreds of programming languages, but JavaScript still has the most contributors in public and private repositories, organizations of all sizes, and every region of the world.

This year, TypeScript shot up to #7 among top languages used on the platform overall, after making its way in the top 10 for the first time last year. TypeScript is now in the top 10 most used languages across all regions. GitHub contributors come from—and across private, public, and open source repositories. *



([Source.](#))

From its slightly rocky start, JavaScript has risen to be the most [popular programming language](#) in the world. According to GitHub's 2018 [Octoverse report](#), there are more JavaScript code repositories than any other language—and that number is steadily on the rise.

A series of JavaScript frameworks and libraries, such as Ember, Angular, React, and Vue, have been developed to allow powerful and complicated web applications to be written using small teams within short time spans. Alongside client and server software, it is now even possible to write native mobile apps using JavaScript. Unsurprisingly, this is becoming increasingly popular due to the ability to share code between the worlds of mobile and web.

With all this choice, it's somewhat understandable that there has also been a movement toward a more grassroots, "vanilla" implementation of JavaScript. Web components, small reusable custom browser elements, are the latest challenger aiming to be the next breakthrough in the JavaScript world. Recently, many libraries related to [data science](#) are being introduced to support the various domains of artificial intelligence. Tensorflow, for example, is a popular Javascript library for [data scientists](#) that allows them to create dataflow graphs. Whatever the next big thing is, it's clear that JavaScript is going to be with us for many years to come.

Since you're here...

Thinking about a career in data science? Enroll in our [Data Science Bootcamp](#), and we'll get you hired in 6 months. If you're just getting started, take a peek at our foundational [Data Science Course](#), and don't forget to peep our [student reviews](#). The data's on our side.



About T.J. DeGroat

T.J. is a writer and editor waging war against unnecessary capitalization. You can follow him on Twitter [@tjdegroat](#).

Download our guide to becoming a data scientist in six months

Learn how to land your dream data science job in just six months with in this comprehensive guide.

Download now