

## Chapter 2: Browser Wars and an Odd Language

At 1:30 AM the morning of October 1, 1997, a group of Microsoft employees placed a giant metal Internet Explorer logo on the lawn of Netscape Communications in Mountain View, California.<sup>1</sup> According to a Reuters report of the incident, the Microsoft employees were returning from an Internet Explorer 4 release party in San Francisco; a card attached to the oversized prop read, “From the IE team.”<sup>2</sup>

“It seems awfully immature to resort to fraternity tactics to draw attention,” a Netscape spokeswoman said at the time. “We’re winning the battle. It’s something you’d expect from a startup, not the largest software company in the world.”[CITATION] Although this was the company’s official position, Netscape employees could not resist using similar tactics in response. By sunrise the logo was defaced with spray paint and Netscape’s mascot, a green foam creature named Mozilla, was standing proudly on the wreckage with a placard referring to recent market share percentages: “Netscape 72, Microsoft 18.” [CITATION] The Browser Wars, which would ultimately decide which company brought the Web to the masses, were in full swing.

Less than two years earlier, over the course of ten days in May 1995, a software engineer at Netscape named Brendan Eich wrote an HTML scripting language called Mocha. Eich is not quite sure which ten days it was. “From a calendar, I think it might have been May 6-15, 1995,” he writes in a 2013 answer to a question about Mocha on an online forum. The source files probably included a date—if anyone could find the originals—and also his office-mate at the time might remember, he adds.<sup>3</sup> “That’s the best I can do,” Eich apologizes.<sup>4</sup>

### *Browser Wars.*

Unfortunately for those interested in the origin story of Mocha, those ten days in May do not seem any different from any other ten days of Navigator development at Netscape in 1995. The mood at the time was one of urgency and excitement, perhaps bordering on mania. Netscape, founded in 1994, was riding the first wave of consumer Internet popularity with their hit product Navigator. Internet browsers at the time were a new phenomenon. Before the graphical interface, surfing the Web was a text-heavy activity, largely reserved for geeks and scientists. Netscape had been born out of a small research project at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign. The research project, Mosaic, with development led by 22-year-old Marc Andreessen, was a tool for viewing Web content through a graphical interface. Although not the first tool created for such a purpose, Mosaic was the best. Seeing the potential for the project’s commercial success, Andreessen left Illinois and took his expertise to Silicon Valley. There the young Andreessen met with Jim Clark, a Stanford professor who had made a fortune as a co-founder of Silicon Graphics, who could provide funding.[CITATION] Clark and Andreessen started Netscape together. Soon their venture exploded.

Less than a year after its founding, Netscape went public with 1200 employees and hundreds of millions of dollars of revenue.[CITATION] The company offered shares to the public for \$28 on August 9, 1995. By the end of that day, shares were trading for \$75.<sup>5</sup> It was certainly the fastest growing software company in history; many were calling it the fastest growing company ever. Clearly,

---

1 <http://home.snafu.de/tilman/mozilla/stomps.html>

2 <http://home.snafu.de/tilman/mozilla/stomps.html>

3 <http://www.quora.com/JavaScript/In-which-10-days-of-May-did-Brendan-Eich-write-JavaScript-Mocha-in-1995>

4 <http://www.quora.com/JavaScript/In-which-10-days-of-May-did-Brendan-Eich-write-JavaScript-Mocha-in-1995>

5 <http://www.npr.org/templates/story/story.php?storyId=4792365>

the graphical Web hit a nerve with the public. Andreessen sums up the feeling in a 2000 interview with Wired Magazine: “In 1995, from Q1 to Q4, Netscape's revenue went from \$5 million to \$10 million to \$20 million to \$40 million. No one had ever seen anything like that... Shit! That's it! We are a hit!”<sup>6</sup>

Celebration would have to be short-lived. Competition for Netscape was heating up. Seeing the potential of Andreessen's innovation, Microsoft had offered to buy Netscape in 1994.<sup>7</sup> Netscape executives had scoffed at the deal, which they thought was far too low a sum. At the time, the Redmond software giant, with Bill Gates at the helm, was the gorilla in the room. The company had popularized personal computing and they were on track to release Windows 95 that year. But the explosive growth at Netscape was worrisome for Microsoft. Netscape was calling their product, perhaps prematurely, a “distributed OS,” a claim that if true would threaten Microsoft's core business.<sup>8</sup> So in early 1995 Microsoft licensed Mosaic, the same research product that Andreessen had worked on, from Spyglass Inc, an offshoot of University of Illinois that was created to commercialize the work of NCSA.<sup>9</sup> By the summer of 1995, Microsoft was only months away from releasing what amounted to a thin varnish over the Mosaic codebase that they were calling Internet Explorer 1.[CITATION] Although late to the browser game, Microsoft was serious about their new venture. Due to the company's extraordinary success in the personal computing space, Microsoft had amassed an elite team of software engineers and had nearly bottomless resources to throw at the project. Netscape's competition was about to get supercharged, and everyone knew it.

### *Proving Mocha.*

Mocha, Eich's ten-day language, was one of an array of strategic plays that Netscape was making to stay ahead of Microsoft with their Navigator 2.0 release. Given that Web content in 1995 was completely static, the need for a HTML scripting language was not necessarily obvious. But there were visionaries at Netscape and elsewhere who saw the need for such a technology and understood this was a place for a big strategic win. The development of Mocha was mandated from the top. “The impetus was the belief on the part of at least Marc Andreessen and myself, along with Bill Joy of Sun, that HTML needed a 'scripting language,’” said Brendan Eich in a 2008 interview with ComputerWorld on the topic.<sup>10</sup> “We aimed to provide a 'glue language' for the Web designers and part time programmers.”<sup>11</sup> But even with support from Andreessen, not everyone at Netscape was convinced that a glue language was a priority. Even if a browser language was important, why create something new? At the time, the Java programming language was gaining popularity and many thought it was a logical language for the browser (given its hardware agnostic nature). In fact, Netscape was in the midst of crafting a deal with Sun Microsystems to support Java in Navigator. So why was Mocha worth the effort? The big debate inside Netscape became “why not just Java?” [CITATION, CONTEXT]

This was a reasonable question to ask. But Eich (and Andreessen) thought that the two languages were different enough that both were needed. The professional software engineers, who would also be coding the backend servers powering websites, deserved a powerful, robust, and extendable language. On the other hand, the front-end amateurs needed something expressive, quick and easy. In order to appeal to both audiences, a move that could cement Netscape's position, Navigator could support both. But in order to prove this idea to the company, Eich had to move quickly. Political

---

6 <http://www.wired.com/wired/archive/8.08/loudcloud.html?pg=4>

7 <http://www.youtube.com/watch?v=Rj49rmc0IHs>

8 <http://www.computer.org/csdl/mags/co/2012/02/mco2012020007.html>

9 [http://www.ericssink.com/Browser\\_Wars.html](http://www.ericssink.com/Browser_Wars.html)

10 [http://www.computerworld.com.au/article/255293/a-z\\_programming\\_languages\\_javascript/](http://www.computerworld.com.au/article/255293/a-z_programming_languages_javascript/)

11 [http://www.computerworld.com.au/article/255293/a-z\\_programming\\_languages\\_javascript/](http://www.computerworld.com.au/article/255293/a-z_programming_languages_javascript/)

forces at Netscape, with Microsoft squarely on their minds, demanded a proof of concept. “Doing this work so fast was important,”<sup>12</sup> Eich recalls in a talk at a conference on JavaScript in 2012. “We knew Microsoft was coming after us.” Eich buckled down and wrote the Mocha interpreter in ten days, largely because the fate of Mocha was not sealed. Until Eich could point at something real, Andreessen's arguments for the language had no ammo. “I had to [write the interpreter] for an internal demo, because otherwise people would doubt that it was either real or necessary,” Eich recalls at the 2012 conference.<sup>13</sup> Apparently his demo was good: Netscape shipped Mocha, by then renamed JavaScript, with Navigator 2.0 in December 1995.

### *Sun and Netscape.*

By the time Navigator 2.0 shipped, Eich's language had been renamed twice. First to “LiveScript” in September 1995, to jell with the rest of Netscape's products (which all began with a “Live” prefix). Then subsequently in December to “JavaScript,” in a licensing deal with Sun, that was aggressively marketing its Java platform at the time. Sun and Netscape had been working together on a major deal to put Java into Navigator. But Sun did not like the idea of the scripting language. In their minds, Java was going to be the browser language, end of story. Why was Netscape trying to put two language into Navigator? Andreessen believed strongly in LiveScript and was not willing to compromise by leaving it out of Navigator. So in order to satisfy Sun, Netscape agreed to rename the language JavaScript as part of the deal.

Over the years, the marketing decision behind the “JavaScript” name has led to a lot of confusion. Because Java was so well regarded at the time, the confusion was probably a good thing for Netscape in JavaScript's early years. But even today software developers and laypersons alike have been known to assume that JavaScript is an offshoot of Java. Oracle, the company that current owns and develops the Java platform, even includes a webpage dedicated to clearing up the misunderstanding.<sup>14</sup> In reality the languages are fundamentally different. Although they share some superficial similarities, the cores of each language have almost nothing to do with each other. That said, the politics of Java has direct impacted JavaScript, before, during and after JavaScript's development.

Java started its life in 1991 as a language called Oak, written for a research project developed by a small team at Sun that was very removed from the rest of the company. According to a Sun document recounting the history of Java written on the third anniversary of Java's public release, Sun's secret “Green Team,” with 13 total members, was formed to anticipate and plan for the next big advances in computing. “Their initial conclusion was that at least one significant trend would be the convergence of digitally controlled consumer devices and computers.”<sup>15</sup> With this thesis in mind, the Green Team set out to build technology for consumer devices, mainly televisions. Almost immediately, they realized that such a project demanded that they had to work with many different processor architectures and that C++ was not the best language for the job. Out of frustration, James Gosling, a software engineer on the Green Team wrote Oak.<sup>16</sup>

Gosling's Oak was compiled into a processor-agnostic set of instructions, which then could be run on a “virtual machine” on a number of different architectures.<sup>17</sup> As such, Oak allowed Green Team engineers to write a program once and then run that program on many different devices without extra

---

12 <http://www.youtube.com/watch?v=Rj49rmc0IHs>

13 <http://www.youtube.com/watch?v=Rj49rmc0IHs>

14 [http://www.java.com/en/download/faq/java\\_javascript.xml](http://www.java.com/en/download/faq/java_javascript.xml)

15 <http://web.archive.org/web/20050420081440/http://java.sun.com/features/1998/05/birthday.html>

16 <http://www.cs.gmu.edu/cne/itcore/virtualmachine/history.htm>

17 <http://web.archive.org/web/20050420081440/http://java.sun.com/features/1998/05/birthday.html>

effort. Looking back, this was a major step forward for programming abstraction, but at the time Oak did not seem critical for Sun. For many years, it existed only in a strange back corner of the company, where the Green Team, eventually spun off as a subsidiary called First Person, continued their work with media technology.[CITATION] It was not clear whether they were making a ton of progress, or that Oak would ever see the light of day.

But in 1995, the world was changing rapidly. Suddenly it became clear to Sun executives that Oak, which was subsequently renamed Java, had some characteristics that made it perfect for programming in the browser. James Gosling explains their realization in an interview for the 1998 historical account at Sun: “Even though the Web had been around for 20 years or so, with FTP and telnet, it was difficult to use. Then Mosaic came out in 1993 as an easy-to-use front end to the Web, and that revolutionized people's perceptions. The Internet was being transformed into exactly the network that we had been trying to convince the cable companies they ought to be building.”<sup>18</sup> The processor independent nature of Java, which was not easy to replicate in C++ or another serious programming language at the time, was perfect for browser programs (which had to run on every computer). “It was just an incredible accident. And it was patently obvious that the Internet and Java were a match made in heaven. So that's what we did.”<sup>19</sup>

On May 23rd, 1995, John Gage, director of the Science Office at Sun, and Marc Andreessen stepped on the stage at the SunWorld conference to announce simultaneously the public release of Java and the incorporation of Java into Navigator.<sup>20</sup> Although Java had been developed separately from concerns associated with the Web, it appeared to the world that Java was built for the Web. This is exactly what Sun and Netscape executives wanted. The partnership was a strategic play that they knew could forward the momentum of both Sun and Netscape. Both technologies, Java and Navigator, were receiving a lot of press and enthusiasm. Together, could they challenge Microsoft? More pessimistically, if they did not act together, did they stand a chance? As Douglas Crawford, a major figure in the JavaScript development community, said of the deal in 2011, “At the time there was a lot of excitement about Java and the Netscape browser. Sun and Netscape decided they needed to work together against Microsoft, because if they didn't join forces, Microsoft would play them off against each other and they'd both lose.”<sup>21</sup>

### *Java and JavaScript*

The close relationship between Sun and Netscape in 1995 cast a long shadow over the development of JavaScript. According to Eich, he was hired at Netscape to embed a functional language in Navigator. Eich writes in a 2008 blog post: “As I’ve often said, and as others at Netscape can confirm, I was recruited to Netscape with the promise of 'doing Scheme' in the browser... Whether that language should be Scheme was an open question, but Scheme was the bait I went for in joining Netscape.”<sup>22</sup> But when he arrived at Netscape and the development of the HTML scripting language was about to begin, Eich recalls, management had new requirements. “The diktat from upper engineering management was that the language must 'look like Java.’”<sup>23</sup> There was a strong desire by marketers at Netscape to associate their browser with Java. Many people were predicting that Java was going to take over the world with its “Write Once, Run Everywhere” philosophy, so perhaps that desire

---

18 <http://web.archive.org/web/20050420081440/http://java.sun.com/features/1998/05/birthday.html>

19 <http://web.archive.org/web/20050420081440/http://java.sun.com/features/1998/05/birthday.html>

20 <http://web.archive.org/web/20050420081440/http://java.sun.com/features/1998/05/birthday.html>

21 [http://www.youtube.com/watch?v=t7\\_5-XYrkqg](http://www.youtube.com/watch?v=t7_5-XYrkqg)

22 <https://brendaneich.com/2008/04/popularity/>

23 <https://brendaneich.com/2008/04/popularity/>

was well-founded. Regardless of its marketing potential, asking that the language look like Java, Eich notes, “ruled out Perl, Python, and Tcl, along with Scheme,” which otherwise would have been reasonable choices to adapt for Navigator.<sup>24</sup> In fact, there was only a certain class of languages that looked like Java: languages in the C-family.

Java's syntax was designed to look like C++, which was Java's main inspiration and competition. The C-family of languages has a very particular brace syntax that was very familiar to essentially all programmers. However this style was designed for procedural programming and none of the popular scripting languages (which largely took inspiration from the LISP-family of languages) had this syntax. Of course, Eich could not simply use Java. Java's rigid class system, static typing and lack of functional capabilities made it attractive for large-scale application development, but very unattractive for writing short programs to write together simple tasks. In order to satisfy these conflicting goals, Eich and his superiors knew that a new language had to be written. And once that became clear, there was pressure to not to copy Java. Eich recounts, “If I had done classes in JavaScript back in May 1995, I would have been told that it was too much like Java or that JavaScript was competing with Java ... I was under marketing orders to make it look like Java but not make it too big for its britches ... [it] needed to be a silly little brother language.”<sup>25</sup>

### *The Good, the Bad and the Ugly*

Given all that surrounded the origins of JavaScript, and the requirements that political forces put on its development, the fact that Eich got anything working might be a miracle. JavaScript had to simultaneously look like Java, avoid copying Java and act very differently than Java. It had to play nice with HTML and Navigator's internals and be interpreted on any machine architecture. Lastly, it had to be designed and written on an extremely short timeline. Although the language was not released until December, Eich, who was the only developer on the project until 1996, had little time to devote to the language after the infamous ten days in May.<sup>26</sup> Mostly this was because he had to make sure the language would work in Navigator, which ended up being the bulk of the work: “I spent the rest of 1995 embedding [JavaScript] in the Netscape browser and creating what has become known as the 'DOM': APIs from JS to control windows, documents, forms, links, images, etc., and to respond to events and run code from timers,” said Eich in a 2008 ComputerWorld interview.<sup>27</sup> Without the DOM interface, JavaScript was of little interest to anyone at Netscape. So the design of the language itself had to be accelerated.

The language that Eich produced had C-like brace syntax and took inspiration from many parts of Java's standard library. But that is largely where the similarities with Java stop. JavaScript functional concepts are taken from Scheme, and the language's object-oriented concepts are inspired by Self and Smalltalk. JavaScript is an impressively coherent language given its origins, but Eich is not a superhero: the stresses of the development period clearly took a toll on the language.

JavaScript has a number of fundamental problems that have been acknowledged by nearly everyone, including Eich, since the language was released. At certain points these flaws have been considered insurmountable. The language, and Eich as its author, have been lambasted by those who have called JavaScript nothing but a collection of mistakes. Even Robert Cailliau, a well respected Internet pioneer who worked with Tim Berners-Lee on the HTML format and the first browser, has said

---

24 <https://brendaneich.com/2008/04/popularity/>

25 <http://www.computer.org/csdl/mags/co/2012/02/mco2012020007.html>

26 [http://www.computerworld.com.au/article/255293/a-z\\_programming\\_languages\\_javascript/](http://www.computerworld.com.au/article/255293/a-z_programming_languages_javascript/)

27 [http://www.computerworld.com.au/article/255293/a-z\\_programming\\_languages\\_javascript/](http://www.computerworld.com.au/article/255293/a-z_programming_languages_javascript/)

that JavaScript is “the most horrible kluge in the history of computing.”<sup>28</sup> These critiques are probably too harsh, but no one denies that JavaScript has its warts.

JavaScript can be written with a functional style (closures are supported and functions are first class objects in the language), but most would consider JavaScript to be an object-oriented. However, unlike every other popular object-oriented language, JavaScript does not use classes. In Java, a class based object-oriented language, classes representing data and functions for operating on that data are written by the programmer. Then instances of these classes, objects, are created and used when the program is run. On the other hand, in JavaScript there are no classes. JavaScript objects can be polymorphic, and a programmer can encapsulate data within a JavaScript object, but inheritance is nothing like Java's clean implementation. To create a new object (say a Racecar object) by extending an object that already exists (say a Car object), a programmer must literally copy the object and add new functionality. This copying happens at runtime, rather than at compile time and means that clean data encapsulation inside the copied object is sometimes not possible.

JavaScript's type system often gets a hefty portion of the blame for JavaScript's problems as well. First of all, JavaScript has no static typing. This makes it difficult to reason about JavaScript programs formally: the output of a JavaScript program is only defined when the program is run, unlike statically typed languages like C++ which allow for a compiler to optimize programs. This would not be a big deal, considering that many languages lack static types, but JavaScript's type system has a multitude of other design problems. There is only one “Number” type in JavaScript. Unlike other modern languages, in which numbers can be represented as integers or floating point numbers for different levels of precision, all numbers in JavaScript are implemented as floating point numbers under the IEEE standard. This leads to unfortunate consequences. In JavaScript, 0.1 plus 0.2 equals 0.30000000000000004. Dealing with very large or very small numbers (numbers that lack reasonable precision in the IEEE standard) is also difficult. Could a bank use JavaScript to keep track of interest payments? Or NASA use it to predict a rocket's path? Probably not.

Type conversion and comparison in JavaScript is also a common pain-point. There are two comparison operators in JavaScript “==” and “===”, with different but almost identical meanings. “==” includes automatic type conversion, whereas “===” does not. JavaScript's strange auto-type conversion rules mean that some odd things are true within the language. For instance, the string “0” equals 0, which also equals the empty string “”, which also equals the boolean false. JavaScript has the values null, undefined, NaN (“not a number”) and false, which are all used in different contexts to signify that something is a “bad” value that cannot be operated on in a meaningful way. Each of these has slightly different meaning, but for beginners as well as seasoned programmers, those meanings are not always obvious.

There is a laundry list of complaints about JavaScript. Many would argue that variable scope in JavaScript could have been designed in a much better way, for instance. But it is important to note that these complaints mostly emerged *after* the language was already in widespread use. Until people started using JavaScript to write more than “toy” programs, the design issues of JavaScript did not show themselves. For instance, JavaScript does not have a built in module system; the language was designed to rely on HTML's “<script>” tag in order to have access to other JavaScript programs. But this is only a problem worth complaining about if a programmer wants to import models into another program. And that desire only manifests itself when a programmer is writing a reasonably large program or wants to build on other people's code. The common complaints about JavaScript only make sense if you are comparing the language to more serious popular languages, such as Java, Python or C++. Only the authors of serious pieces of software in JavaScript would complain about its dynamic

---

28 [http://en.wikinews.org/wiki/Wikinews\\_interviews\\_World\\_Wide\\_Web\\_co-inventor\\_Robert\\_Cailliau](http://en.wikinews.org/wiki/Wikinews_interviews_World_Wide_Web_co-inventor_Robert_Cailliau)

types. Who needs to reason formally about a 20 line program that turns a webpage green when a button is clicked? JavaScript is an odd language with a strange history and a number of flaws, but it has come a long way. Complaints about JavaScript, in a way, prove that it being taken more seriously.

### *JavaScript and Microsoft*

On August 13, 1996, Microsoft released Internet Explorer 3. The application represented increased efforts by the company to develop the product internally, and no longer relied on the Mosaic codebase. In a press release for the beta version of product, Brad Chase, a Microsoft executive claims that “Microsoft Internet Explorer 3.0 is a step ahead of the pack.”<sup>29</sup> Although this is marketing speak, no one could argue that IE3 was not a good browser. With the release of the 3rd generation of their browser product, Microsoft finally achieved feature parity with Navigator. For instance, the browser included the first market implementation of Cascading Style Sheets (CSS), allowing HTML to be styled with different fonts, colors and layouts [CITATION]. Before it was superseded by Internet Explorer 4, IE3 took significant market share from Navigator. In 1996, Microsoft's market share of the space was in the single digits. By the end of 1997, it was closer to 30%.<sup>30</sup>

Among a multitude of improvements that the product brought over its predecessors, Internet Explorer 3 shipped with JavaScript support. Microsoft had reverse engineered Navigator's JavaScript with an implementation they called JScript (for copyright reasons).<sup>31</sup> JScript was essentially a direct copy of JavaScript. There were small differences, but for the most part, a program written in JavaScript could be run in both browsers. This did not seem like a big deal at the time, except for Netscape, which was obviously not very happy about it, but it was this development that took JavaScript from being a proprietary tool to something of a standard. As of 1997, the two largest browsers had committed to supporting the language.

Internet Explorer 3 was replaced by Internet Explorer 4 in late 1997, which was replaced not too much later by IE5, and then IE6. With each iteration, Microsoft took more and more market share from NetScape. Microsoft had a number of market advantages over Netscape that made their progress much easier. The most conspicuous was financial resources. In the mid-90s, Microsoft had more cash on hand than Netscape was making in revenue by a wide margin.[CITATION] Enormous profits from its other businesses, \$559 million in the second quarter of 1996, allowed Microsoft to give away Internet Explorer for free and market it heavily.<sup>32</sup> NetScape could not afford to compete in this way. Microsoft's second big advantage was its domination of the operating system space—Windows faced no real competition. Taking advantage of this, Microsoft began bundling Internet Explorer with its operating system. Most copies of Windows at the time were being sold to people who had never owned a personal computer before, let alone used a browser. So users had nothing to compare Microsoft's browser against; it is possible they did not even know they *could* download a competing product. As the market for web browsers grew rapidly, and spread to encompass millions of less technical people, Microsoft's bundling strategy made Internet Explorer the default for new users. Soon, the default became the standard. By 1998, Internet Explorer owned the market. In that year Microsoft was sued for anti-competitive behavior (partially because of its software bundling strategy), but the war was over. Netscape was acquired by AOL. By 2002, Microsoft had 96% market share.[CITATION]

Although its original implementation did not become its dominant form, JavaScript lived on. NetScape had submitted JavaScript to a standards board (ECMA) for a language standardization in

---

29 <http://www.microsoft.com/en-us/news/press/1996/may96/ie3btapr.aspx>

30 <http://www.cnn.com/TECH/computing/9810/08/browser.idg/>

31 <http://www.microsoft.com/en-us/news/press/1996/may96/ie3btapr.aspx>

32 [http://articles.latimes.com/1996-07-23/business/fi-27151\\_1\\_fourth-quarter-profit](http://articles.latimes.com/1996-07-23/business/fi-27151_1_fourth-quarter-profit)

1996, and more practically Internet Explorer implemented it. When other browsers came along later and took market share away from Internet Explorer, implementing JavaScript for these browsers was a no-brainer. Mozilla (a non-profit born out of the rubble of Netscape) introduced Firefox, Apple introduced Safari and Google introduced Chrome. All three of these browsers supported JavaScript from version 1. The fierce competition between Netscape and Microsoft, followed by Internet Explorer's rise to prominence, propelled Eich's language to the status of a standard. What language to implement in a browser is no longer up for debate.