Matt DaSilva

Dr. James Teresco

Programming Languages

December 4, 2023

Language Project: JavaScript

The JavaScript programming language is a highly versatile programming language used primarily in web design and web applications. Primarily used for web design and web development as a whole, JavaScript is considered one of the core technologies of the World Wide Web, which was introduced to the public in 1993. JavaScript itself on the other hand was first created in 1995, two years after the web became public. JavaScript is cross-platform and object oriented, primarily used to allow for further interactivity in web pages than standard HTML can provide at times. JavaScript also includes Node.js, which allows for more advanced functionality for websites, and allows one to activate a website on their browser, to observe what they may need to change before getting it distributed fully. JavaScript is one of the more flexible programming languages out there, allowing itself to be used with other languages, such as HTML and CSS, to fully form a website. In the case of HTML, JavaScript programs using HTML use a different file type. This file type is written as ".ejs" to signify it is a JavaScript file written in HTML.

While JavaScript has without a doubt become an absolute behemoth in the programming world since its inception. It was never intended to be this way initially. As a matter of fact, the development of JavaScript is quite short, taking only ten days to fully create in the mid-1990s. "When the Internet boomed with public usage in the 1990s, technology companies battled to

build the most powerful and efficient web browser on the market. This sparked the first ‘browser war,’ between Microsoft (Internet Explorer) and NetScape (NetScape Navigator) to gain dominance in the usage share of web browsers.” (Kurothe) The creation of JavaScript in part was the result of this competitive landscape, as both companies wanted to get ahead of the other and make a superior platform which would dominate this new and opportunity-filled market created by the introduction of the internet. Created by Brendan Eich, JavaScript was meant to be a small and lightweight scripting language for the company NetScape, an independent computer services company. The primary requests NetScape had for Eich were that the language much contain syntax similar to Java, and less like Scheme and other programming languages that were commonplace at the time of its creation. It even had a different name, being referred to as Mocha, prior to its publication and release in 1995. Upon its release, the programming language Mocha was changed to JavaScript, relating to its similarity to Java. While it did cause confusion, with many believing that it was directly related to Java itself rather than it simply being built with the Java syntax being a base for it. “However, except for syntactic resemblance, JavaScript has almost nothing to do with the Java programming language. They are both completely different languages.” (Dickson) JavaScript was not based on Java, nor was Java based on JavaScript. JavaScript had syntactic resemblance due to Java being heavily marketed and was the most relevant programming language in the world at the time. “Netscape thought it would be a good idea to capitalize on this success by creating the name ‘JavaScript.’” (Dickson) Java was the most-talked about language at the time, leading to Netscape changing the name of Mocha to JavaScript in the hopes of gaining more publicity for the language. However, after NetScape was acquired by AOL, they turned their code over to the Mozilla Foundation, and Eich renamed the language JavaScript. As a result, JavaScript grew as one of the primary programming languages

used by programmers worldwide over the span of over 25 years. The creation of JavaScript did not initially stop this browser war, as many other companies tried to implement their own programming languages for their own browsers. However, this quickly backfired. “This created a huge headache for developers, as code that worked fine on one browser was a total waste on another. This went on for a while till they all agreed to use the same language (JavaScript) in their browsers.” (Pierre) The primary frustration developers have had with web design in general still existed, and the problem was not solved. This led many software companies back into the same problem that they had been attempting to avoid to begin with. And much like how JavaScript escalated the browser wars of the 1990s, JavaScript also brought it to an end, with companies deciding to stick to JavaScript universally for all major browsers, to allow for better user interactivity and less headaches with learning entirely new languages for an ever-changing amount of browsers. As a result of its acceptance as a universal language for browsing platforms, JavaScript is now seen as an essential language that is supported by all major browsers. “As of November 2023, 98.8% of websites use JavaScript as a client-side programming language, which shows its increasing popularity.” (Raval) JavaScipt’s popularity has soared in the last nearly 30 decades, and continues to do so, reaching staggering heights of usage by the programming community. JavaScript is now a must-learn in the web developer world, and will continue to grow.

JavaScript has multiple capabilities, and as discussed before, it can do virtually anything you need it to. JavaScript can be used at any section of web development, be it front-end or back-end. You can even use Javascript for full stack development to simplify the process of writing your code and building your platform. JavaScript is also standardized, and frequently tweaked and updated with newer versions. This allows JavaScript to contain further uses in terms

of functionality and be more versatile than previous versions to the programmer. JavaScript primarily works with the document object model to respond to any and all user interactions with software. “Client-side JavaScript extends the core language by supplying objects to control a browser and its Document Object Model… client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.” (Kurothe) This model is meant to primarily help with client-side applications. This would mean it’d need to work primarily with HTML and link HTML files and programs together in a way that cannot be done with exclusively HTML. This can allow for a lot of the increased interactivity JavaScript has become known for, especially in regards to web platforms and web development as a whole. The model itself provides the structure in the browser that the user uses to access the web pages that the program you built is meant to display. In terms of web design, Java Script can provide a large array of interactions. These include scroll transitions and movement of objects. JavaScript can also, unlike HTML on its own, be used on mobile platforms. HTML cannot regularly be implemented on mobile phones due to the fact that mobile browsers are designed in a fundamentally different way due to the use of touchscreens and a smaller screen size. As a result, their infrastructure is built in a way where HTML may not lead to the best result. In some browsers, certain media types in HTML work differently or simply do not work at all, and many older phone models were restricted to a certain form of HTML. Using a hybrid app using JavaScript, it is much easier and more convenient to package an app and allow it to be installed on a device. This tends to be the much more favored approach with developers nowadays, as it doesn’t need to be inherently complex, and can save programmers a lot of time and stress. There is also a server-side focus in JavaScript as well, primarily focusing on extending the core language, allowing for further uses and far more possibilities for

JavaScript programmers. “Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server. For example, server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.” (Kurothe) JavaScript’s ability to allow communication with databases, manipulate files, or share information from place to place in its programming can allow for further interactivity as well as allowing for more seamless interactions to go along with this further intractability enhancements. This also allows for JavaScript to provide a far more unified ecosystem, as well as making the language more unified in it of itself. In addition to all of the versatility JavaScript already has, JavaScript also has tons of frameworks and libraries that can allow developers to make more complex applications and platforms with less written code and, as a result, in less time than it would have taken the programmer otherwise. Programmers can easily import libraries and frameworks in a manner similar to how they would in Java and C-based programming languages. “JavaScript is an incredibly powerful scripting language, designed specifically for powering immersive web applications. With its lightweight, browser-side execution capabilities and tailored libraries aimed at providing the best possible experience in a variety of scenarios.” (Chauhan) These libraries are essential to JavaScript’s popularity, as they are one of the primary reasons for its versatility, as they are specifically designed to bring the best possible experience for whatever scenario the programmer may require in a language. Its browser execution abilities also provide an efficient method of forming the backend infrastructure for a website, primarily due to its light nature and easy to grasp structure as a whole.

Much of JavaScript’s versatility comes from the vast array of features JavaScript has to offer. JavaScript is platform independent, allowing developers to not have to be concerned about

compatibility with certain browsers. Due to browsers being made by different companies and having differing specifications, even back in the late 90s, the specifications of web browsers could make developing a website quite time consuming and stressful. However, with JavaScript, a problem such as this is nearly nonexistent in the world of the modern web developer. It is also effortlessly easy to embed a program in HTML into any JavaScript program to further enable the capabilities of any web platform. "HTML sets up the structure of your website but JavaScript is what adds functionality to your website… you can create forms, buttons, etc with HTML but it won't function without you applying some JavaScript." (Pierre) JavaScript is, in a sense, what allows HTML to reach its full potential. Analogically, HTML is the circuit board of a website, JavaScript is the wiring and soldering that allows everything to work seamlessly. JavaScript can allow anything HTML can do in an HTML file to be utilized in a JavaScript file. Such instances include the ability to include links, images, tables, linking with a CSS stylization sheet, etc. JavaScript can also allow for template literals, which allow the programmer to save certain variables directly into strings, for example, which can prove useful as it can allow the programmer to focus on more relevant aspects of the code and of the software in general. Arrow functions are also a prevalent feature in JavaScript programming, as they allow for further simplicity in computer code. This is simply due to the fact that they are an, at times, desirable alternative to a traditional function expression. They allow for the shortening of the code, and thus better readability of said code. "Since these are light-weight in syntax, they can be very easily used in anonymous functions in JavaScript." (Kurothe) JavaScript programmers can frequently use arrow functions to simplify their web application programs and allow for further simplification in the syntax of their web code. JavaScript also features dynamic typing, which can help greatly enhance and simplify the code. Unlike other languages, JavaScript's dynamic

typing can allow the programmer to reference variables which aren't explicitly stated in the file itself. JavaScript treats objects as first class citizens. "The term first-class citizen means 'being able to do what everyone else can do'. In JavaScript Objects prototype is the base prototype of all. They can be passed as reference, returned in a function, and assigned to variables for manipulation." (InterviewBit) Essentially, objects can be whatever you want them to be at any given opportunity in a JavaScript program. The interpreter assigns the variables at runtime based on the variable's initial value when assigned, whether it be an int, a string, or something else entirely. This can allow for far more versatile coding, as JavaScript is well known for. Its interpreter is also quite different from other languages. "JavaScript offers unique advantages over traditional compiled languages like Java and C++. By allowing a browser to interpret its code, line by line in real-time, JavaScript reduces the complexity of development without sacrificing speed or power." (Chauhan) JavaScript's simplicity in in its development and runtime allows for a programmer to have both an efficient language and a powerful language. With JavaScript, you can structure your code however you like, in a way that suits what you want it to do the best.

It is for these reasons that I had selected JavaScript as the language I would not only do my research on, but also form a small program using. I had decided the best way to display the capabilities of JavaScript was with a small website. This website will include 3 small pages, CSS being optional, as it is not the main focus of my project, the focus itself is on JavaScript. My program itself is not too complex, as this was not the goal. The program first leads the user to a list of accounts, with a hyperlink which requests that they create an account. In the account creation page, where they will have to fill out the required information in the text boxes. This required information includes your full name, date of birth in the "MM/DD/YYYY" format, as well as the user's hometown. In addition to this general information, they are also asked 2

questions based on personal preference. One of these questions requires the user to write down one of the user's favorite movies, as well as the user's favorite hobbies. Due to the lack of a limitation on how many characters the user can type, the user is allowed to list as many hobbies as they would like into the "Hobbies" category. Using the JavaScript scripting I created, the program will put their profile in a list upon clicking submit. After submitting their information, their profile is added to a list of Profile classes where they can later find it among all of the other profiles on a separate web page. The user can then manually return to the home page, which will list all of their information that they had put in. Because a password feature was not implemented, the account cannot be truly stolen, and the information is thankfully, not relevant to one's identity. In addition to viewing the user's own profile, the user can click on a button connected to a link, which will open to a different web page, which will allow the user to be able to access a page listing other profiles. The profiles will list out all of the information the user had written down for the website. Due to my desire for simplicity over all else, and the primary focus being the storage of data on lists in JavaScript, you will be able to access other profiles by editing the link to an index in the list of profiles. Once the link is properly, the profile that link belongs to will emerge, and display information in the same template as the user's own profile. They are also allowed to go back to the list of profiles and look if they so choose. All profiles in the site will be using the same general template, as this will allow for simplified coding, and allow for better implementation of JavaScripts's list structure. The user can input as many profiles as they would like, and expand the list as far as they would like. The primary purpose of the website is primarily to demonstrate JavaScript's effectiveness in accessing user-put information on a website, and form that data into a cohesive profile, which is then additionally put into a list of profiles. Every profile will follow the same template as is shown in the

Registration page. The user can even observe their own profiles, should they remember the index that it was on. There is even the option to look at the first profile in the list, by simply not putting any index in the link. This will automatically take it to the index of 0, which is the first index in the list, leading the user to the very first profile that the website had created and continues to carry.

Learning about JavaScript has given me an entirely new perspective on what it means to be a programmer in the modern world. With all of its versatility, features, and capabilities which unfortunately I cannot entirely explore with just one project in this timespan, JavaScript is a programming language which I am growing to admire. I had only primarily worked with Python and Java for the majority of my time as a programmer. Naturally, I had to start somewhere, but I always wanted to explore more programming languages commonly used in modern day programming. This is why I decided on JavaScript as the language I would use in my project for Programming Languages. This came with a learning curve, as while it was similar to some languages I have used in the past, simultaneously it was unlike any other language I've coded in in certain aspects. However, I predict that as I learn more and more about JavaScript and code in it more often, I will be more skilled in the language than I am currently, as this is simply a look at a couple of core aspects, and I wish to explore the language fully. As a result of learning more and more about JavaScript, with my eyes opened to the potential of JavaScript. JavaScript is one of the most versatile and useful languages a software developer can use in this day and age, and it shows with its staggering growth in popularity since its introduction to the public in the 1990s up to today in 2023. JavaScript is a wonderful language to learn as a programmer and an essential one at that. In web development, it is one of the most commonly used programming languages due to its versatility and compatibility with commonly used browsers. It is also handy

for increased interactivity where HTML and CSS could previously not satisfy the requirements or demands of the programmer or the client. JavaScripts many features, libraries, and frameworks allow for efficient, quick, reliable use. It also has various shortcuts that can greatly enhance readability and maintain a sense of writability when used properly and in moments where it is most required. Overall, JavaScript is not just an interesting programming language, it is one of the most advanced languages we currently have as programmers. It also continues to grow even after its inception. JavaScript has essentially changed the internet, and programming as a whole, in a way that is unlikely to be replicated by any other language that came or will come after it. I have learned quite a lot with JavaScript and I hope to learn much more about it in the future.

Sources:

Chauhan, Ankur. “Top Features of Javascript: Characteristics of Javascript - DS.” *Devstringx Technologies*, 31 Oct. 2023, www.devstringx.com/features-of-javascript.

Dickson, Boateng. “A Brief History of JavaScript.” *DEV Community*, DEV Community, 22 May 2022, dev.to/dboatengx/history-of-javascript-how-it-all-began-92a.

Kurothe, Divya. “Everything about JavaScript.” *Medium*, FAUN - Developer Community, 3 Feb. 2023, faun.pub/everything-about-javascript-e4e5f7ad7fdb.

Pierre, Gedler Jean. “HTML & Javascript: A Relationship Made in Heaven.” *Medium*, Medium, 20 Oct. 2021, gedler-jeanpierre.medium.com/html-javascript-a-relationship-made-in-heaven-c532eff25734.