The standards and processes of web development has rapidly changed when Tim Berners-Lee, the inventor of the World Wide Web, created the first and discontinued web browser called WorldWideWeb in 1990 at CERN (Berners-Lee, n.d.). The modern web development has been divided into two layers: front-end and back-end. Front-end, or client-side, development concerns about what the user sees. It is also called client-side because the source code is executed locally on the user’s machine or browser. The modern core technologies of client-side web development are HTML (Hypertext Markup Language), CSS (Cascading Style Sheet), and JS (JavaScript). JS has dramatically changed how users can interact with websites. Users are referred to those that view or interact with a website in any way, shape, or form. However, in the past, developing JS has been cumbersome and thus many different JS libraries were created to combat the pain points of programming in JS. As of now, JS has been updated to the point where developers have begun arguing about the modern value of these JS libraries such as jQuery. The focus of this paper will be on the client-side of web development and common JS libraries.

HTML, CSS, and JS are the “basis of web technologies” (Pastore, 2012, p. 310). These three core web technologies work together to shape a website. HTML, developed since 1993 by Web Hypertext Application Technology Working Group (WHATWG), can be thought of as the structural layer of the website. Contents such as the header containing the logo or title of the website, the images, the footer containing the copyright, and all of the text on the website can all be described using HTML. The presentation of an HTML-only website is decent but can be proven to be boring or bland according to modern standards. HTML-only website will just use the default presentation provided by the browser such as black text and white background. Developers also used hacks or unorthodox approaches to make a website look better such as using HTML tables to structure the layout. The World Wide Web Consortium (W3C) decided that there should be a separate technology to handle the presentation of the website and thus created CSS in 1996. By adding in CSS to an HTML website, the website can then better retain or even bring in more users. CSS improves the presentation of the website by adding different colors or changing the sizes or fonts of the text. The developer can also use CSS to make animations on the website such as a link changing its background color when hovered over by the user. Therefore, CSS can be thought of as the presentation layer of the website. A website that only uses HTML and CSS can be considered good enough to present information. However, these two technologies can only do so much with regards to interactivity.

Suppose that an ecommerce business website has a form that a user would have to fill out when purchasing an item from the business. The form consists of input fields for name, mailing address, phone number, email address, and the preferred method of payment. The user can be unpredictable in how they would fill out this form. For example, the user may forget an @ symbol when inputting their email address or inputting the incorrect format of their phone number. As it stands, HTML and CSS website can only put a warning message or description label on how the user should enter the inputs. However, HTML and CSS website cannot verify or validate the user inputs before the user submits the form. The user can only see their invalid inputs after submitting and they have to either refresh the page or backtrack to the page containing the form. However, the form will be unfilled, and the user will have to re-enter their inputs again. This can be a frustrating experience for the user especially if the form is long. In 1995, Brendan Eich created JS during his employment at Netscape. JS is what makes interactivity possible on a website. By adding JS into the previously mentioned ecommerce business website, the form can create a popup alert or warning message if there is an invalid input before the user submits. JS can change the background color of a website when a button is clicked. JS can also be used to create animations such as a slide-down popup or a rotating image. JS is the interactive layer of the website. As of today, HTML, CSS, and JS are the most common client-side web development languages. They all help build the websites, but each has their own roles, functions, and syntax in building the website. A website can be divided into three layers: HTML is the structural layer, CSS is the presentation layer, and JS is the interactive layer. A little bit more discussion on JS should be made since there has been many different libraries created to make developing in JS easier.

As websites are becoming more dynamic, developers develop pain points in writing in JS. A simple interaction of sliding down a popup alert when the user presses a button can contain more lines of code when written from scratch. Developers had started to save pre-written JS. The accumulation of these pre-written JS led to the creation of JS libraries. If one were to search up ‘JavaScript libraries’ on Google, there would be a lot of options. Q-Success listed 29 JS existing libraries, but this can be argued. There are many different JS libraries in development to help with cryptography to data visualization. Therefore, there is not a definite number of JS libraries. A well-known JS library is jQuery. jQuery makes it easier to interact with the Document Object Model (DOM), the programming interface for a HTML document. For example, jQuery can allow the developer to use a single line of code to make a slide-down popup alert message compared to the multiple lines written without using jQuery, thus their motto: “write less, do more”. According to Q-Success, about 76.5% of all websites use jQuery.

In conclusion, HTML, CSS, and JS are the core technologies of web development. JS libraries, such as jQuery, are created to ease in writing JS and the number of JS libraries are indefinite as many more will continue to be created to meet the needs of the developers.

References

Berners-Lee, T. (n.d.). The WorldWideWeb browser. Retrieved October 04, 2020, from https://www.w3.org/People/Berners-Lee/WorldWideWeb.html

Pastore, S. (2012). Website development and web standards in the ubiquitous world: Where are we going? *WSEAS TRANSACTIONS on COMPUTERS,* *11*(9), 309-318.

JS Foundation. (n.d.). jQuery. Retrieved October 04, 2020, from https://jquery.com/

Q-Success. (n.d.). Usage statistics of JavaScript libraries for websites. Retrieved October 04, 2020, from https://w3techs.com/technologies/overview/javascript\_library