JavaScript is a programming language with a wide range of web based applications, including providing interactive and dynamic web content, web-based applications and even mobile app development. JavaScript has a growing popularity, and is increasingly used by websites across the internet.

Some of the most basic and common uses are for providing an interactive and dynamic user experience with websites. These uses include controlling drop-down menus, website pop-ups, slideshows, geolocation, and getting external API data to integrate within the website. For example, many company websites will ask you for permission to access your location when you are trying to find the closest brick and mortar location, this is done using JavaScript.

Another use for JavaScript is getting data from other sources to display on the webpage. My smart mirror uses this feature, requesting API data from Google traffic and DarkSky.net (a weather service) which returns live traffic and weather data. This information is then displayed on a web browser which is displayed on the mirrors monitor.

One easy website that uses JavaScript is Walmart.com. This site uses a large number of different scripts, and also uses a combination of script placements. Some scripts are contained in the <head> of the code, and some contained within the <body>. All of the scripts found on the site are links to external .js script files. Judging off of the names of the scripts, most within the <head> area of the code are focused on page functionality (scrollbars, framworks, responsiveness, navigation, etc), while the scripts within the <body> portion are for product viewing/placement (search results, ad displays, product display, pagination, etc).

JavaScript code placement can vary from site to site, and the optimal placement can vary depending on the specific needs of the page and the script itself. The benefits to placing script code with in the <head> section is typically faster load times, as the browser can cache the scripts before the rest of the content loads, and this also allows the browser to execute the code sooner, which can improve the overall performance. If the code is too complex or too large however, this can actually be a detriment to the page and cause load times to seem longer, and may block other resources from loading.

Placing scripts at the end of the <body> section will help with perceived load times, allowing the page content to be loaded first, however this method means that the browser will not execute the scripts until the page content has been loaded, which can cause a delay in script execution.

Because of the different advantages and disadvantages of each placement option, it is important for developers to consider the size, complexity, and need of each script to determine the optimal placement for each script used.