InnerHTML is a JavaScript output that access a specific HTML element and writes directly into that element. This can be useful when getting data from other sources that you want to insert into an HTML document, or where the data displayed within that element can change, such as the number of items in a shopping cart, or the total cost of a shopping cart, which the data changes with each item added or removed.

Document.write() is a JavaScript output that displays a script output, but will erase all existing HTML being displayed. It is important to note this should only be used for testing purposes. Using document.write() will clear all of the HTML that has been loaded, replacing it with the output that has been provided to the write() function once it is called.

Window.alert() is a useful JavaScript tool that allows the website to display its own alert pop-up to notify users of something, or even get permission to perform a certain action. A common example is a website asking for permission to use your location, which uses an alert pop-up window, typically with yes/no options, giving the other scripts on the page permission to access your location in order to find a store location geographically close to yours.

Console.log() is a JavaScript output function that writes output to a console, which is helpful for testing and debugging. The output from this option is what we have grown accustomed to with many of out Java assignments, using system.out.println to print output into the console/terminal, excluding our use of JavaFX so far.

The most common JavaScript display we will use are the InnerHTML, used to write external data to a specific place on a webpage. Somewhat common, but less frequently used will be the Window.Alert() to alert a user or get user permissions. This will be still be common but used less frequent because it is typically used once per site vs the InnerHTML that can be used multiple times on a single page.

The advantages of the InnerHTML is that a single webpage could be used as a template, where different data is fed into the page based on various criteria. An example of this would be our blackboard gradebooks. This page is the same for each of us students, but the data is different based on our personal accounts. It would be a waste of resources for blackboard to have a different html page for each user, so they use a template page that is populated with data from the database based on the logged in user.

The Document.write()’s disadvantage is that is erases all of the HTML that is already loaded, making it unrealistic in a production environment. This display option could be used for testing the output of a script during development, but in a real world production/deployment it would not be used.

Console.log() could be used on the back end of a deployed site for testing and debugging, where the developers were able to view the log/console to see what was happening when an issue arose. The advantage here would be to provide backend information to developers without having to display detailed information to the end user.