This was a really interesting challenge. I spent a lot of time looking at w3 Schools and talking to my dad about the best way to store the possible values of the difference table before I decided on trying it with arrays. Originally, I tried to create a function that would generate all the y values of a polynomial and generate the difference table from that. This unfortunately defeats the purpose of the difference engine, as it is effectively just a polynomial calculator. Instead, I reformatted my html to accept inputs for the first row of the difference table. When the user clicks the button, the javascript code is then executed. Here is a collection of the in-code comments that show my methodology:

//The rows variable will determine a few things for us. First, the number of spots inside our array, and second, the number of rows the program creates

//The variable is set to whatever is input into the textbox by the user.

//y is simply a variable we will use to generate the array.

//The leading variable will hold the leading coefficient of the polynomial.

//This will be used to inform the user of the leading coefficient as a way for the user to know if they input the desired values.

//This will also be used to set the derivatives after the constant derivative to 0, for example, y=x^3 y'3= 6 & y'4=0 so everything in the y'4 column will be 0.

//This array will be the basis for the Difference Engine. Its here that we will store all the values for our output.

//This fills the array with the number of rows that the user asks for.

//This sets the user inputs to the first row of our array. We are almost prepared to find the differences.

//This determines what the leading coefficient input was. It is used as a way for the user to confirm their entry. If the value output by the function is different from what the user intended, this will allow the user to know.
//This alert is what displays the leading coefficient

//This for loop makes sure that values that would return as undefined instead correctly become zeros.

//This code constructs calculates the differences. The way it works is simple: it is simply the reverse of how you calculate the first row by hand. Instead of subtracting [0][0] from [1][0] to find [0][1] you add [0][0] to [0][1] solve for [1][0], a simple application of algebra. Put another way, x - y = z --> x = y + z.

//This part of the code outputs the values into an html table. The X coordinates are displayed on the right hand side opposite from the Ys.

Unfortunately, I didn’t have time to fully debug the polynomial input part of the code. It was my intention to give the user the option to put in the first row of the table, or the polynomial they intended to use. The polynomial would be used to get the first row, and those values would be put into the first-row input. I had problems with proper array set up and some errors that made it too time consuming to do properly.