Hands-On #2

Enhancing the MPG Application

In this hands-on session we shall be enhancing the MPG application.

1. Using your text editor open the file “mpg\index.html”
   Then review the code

2. Run and test this application with valid data to see how it works. When you click the **Calculate MPG** button, the correct results should be displayed

3. Now test the data validation routine. Note that one error message is displayed no matter which entry is invalid

4. Your task now is to **Enhance the Data Validation** of this application.

   Enhance the selection statement so that it provides a different error message for each text box and for each type of error:

   - Miles must be numeric
   - Miles must be greater than zero
   - Gallons must be numeric
   - Gallons must be greater than zero

   The selection statement should include one if clause for the first condition, followed by three if else clauses that test for the next three conditions.

   Test these changes. If necessary fix the errors and test again.

5. Your next task is to add a **Clear Entries** button below the **Calcualte MPG** button.

   To do that copy the HTML for the label and the input elements for the Calculate button, and paste it after the input element. Then modify HTML for Clear Entries button so it has a unique id and an appropriate value attribute

6. Add the JavaScript code for an anonymous function that's stored in a variable clear. This function should clear the text boxes by using the $ function to get a Textbox object for each text box and then set the value property of the text box to an empty string.

   Later add a statement in the onload event handler that attaches the clear function to the click event of the Clear Entries button.

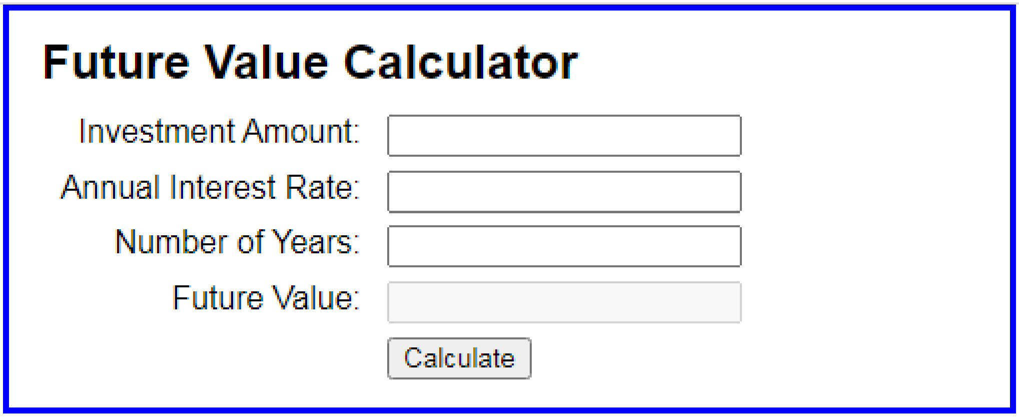
   Now test this enhancement.

7. Add a statement to the onload event handler that attaches the clear function to the double-click event of the miles text box.

   Test this change.

Develop a Future Value Application

This part of the hands-on session will guide you through the development of a new application call the **Future Value Application**. The interface for this application is as follows:



Be sure to test this application after each of the following steps.

8. Using your text editor open the HTML and JavaScript file from the **02\future_value** folder

   Run the HMTL file to see that it provides all the code for the above shown user interface.
   Note that the JavaScript file only has the $ function that's commonly used to get the objects for elements with specific Ids

9. Add a **calculateClick** function that will be used as the event handler for the click event of the Calculate button. Within this function, code three variables to get the values from the first three text boxes by using the $ function. Convert the first two values to decimal and the third value to an integer.

   Code the for loop that calculates the future value of an investment. When the loop ends use the **toFixed** method to remove the decimal places from the future value and put the value in the Future Value text box

10. Add a selection statement that tests just the first entry is a valid number greater than zero. If the entry isn't valid, use the alert method to display appropriate message. Finish the

selection statement with the else clause that contains the code that you used for calculating and displaying the future value.

11. Enhance the selection statement so that it checks the other two entries for validity. Here again, each entry should be a number greater than zero and the alert method should be used to display appropriate message

12. The Future Value Application is now a working application that you developed by coding JavaScript and testing a limited number of statements in each step.

See how you can have a better interface for the Future Value Application. This task is left up to you.