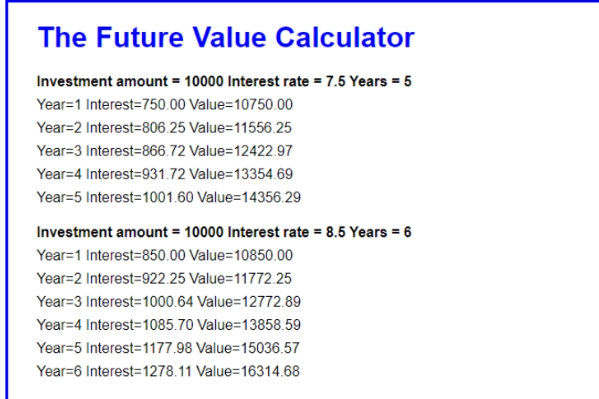
**Activity 01: Enhance the Future Value application**

This exercise will give you a chance to use if statements, do loops, and for loops as you enhance the Future Value application. The eventual output of this application will be displayed in the browser and should look something like this:



Test the Future Value application

1. Open your text editor or IDE. Then, open the application in the folder attached that is future value.

2. Run the application by opening the index.html file in a browser, and test it with valid entries. Then, test it with invalid entries.

**Add a do-while statement for continuing the entries**

3. Open the future\_value.js file. Then, add a do-while statement to the application so the user can repeat the calculation for another series of entries.

**Improve the validation for the investment, rate, and year entries**

4. Update the condition for the do-while statement for each user entry so the entry must also be a positive number.

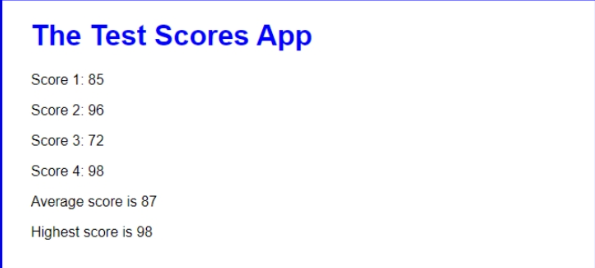
5. For the interest rate entry, add code that makes sure the user enters a value that’s greater than zero and less than 15.

**Enhance the display of the results as shown above**

6. Modify the for loop that calculates the future value so it displays the interest and future value for each year, as shown above. Use h4 elements for the user entries and <p> elements for the results. Note that you’ll have to change the location of the existing write() methods to get the display the way it is above.

**Activity 02: Enhance the Test Scores application**

In this activity, you’ll make an enhancement to the Test Scores application that uses an array. What you’ll do is add a for loop that gets the highest score in the array and displays it below the average score in the browser:



1. Open your text editor or IDE. Then, open the application in this folder attached that scores\_array

2. Run the application by opening the index.html file in a browser, and test it with valid entries. Then, test it with invalid entries.

3. Declare a variable named highestScore at the start of the script that will be used to store the highest score. Its starting value should be zero.

4. Add a for-of loop at the end of the script that’s executed once for each score in the array. Within the loop, an if statement should replace the value in the highestScore variable with the current score if that score is greater than the value in highestScore. That way, the highestScore variable will eventually store the highest score in the array.

5. Add a write() method after the for-of loop that displays the highest score as shown above.

6. When you’ve got that working, comment out the for-of loop that you just coded. Then, modify the other for loop so it not only sums the scores but also puts the highest score in the highestScore variable. After this change, the application should work the same as it did after step 5.