# Lab 10.1 Write the JavaScript for the Test Scores application

This exercise gives you a chance to work with arrays as you write the JavaScript for the Test Scores application:



**Open, test, and review this application**

1. Open the files in the test\_scores folder:

2. Test the application in Chrome. Notice that the Add Score button does not do anything yet.

3. Review the JavaScript file for this application to see that the ‘DOMContentLoaded’ event handler contains a click event handler for the Add Score button. Also notice that there is a global array called scores which should be used to store all the scores. The $ function is also coded.

**Code the Add Score event handler**

4. Add code to the addScore array to read in the value of a score entered by the user. Add error checking to ensure that the score is between 0 and 100 inclusive, and that it is a number. If any of these constraints fail, add the following text to the span immediately after the button: 'Score must be between 0 and 100.'

5. If the score passes the validation, add the new score to the scores array, and display all scores in the label with id=”all”.

6. Work out the average of all the scores - use the reduce method to total the scores and then divide by the length to get the average. Display the average score in the label with id=”avg”.

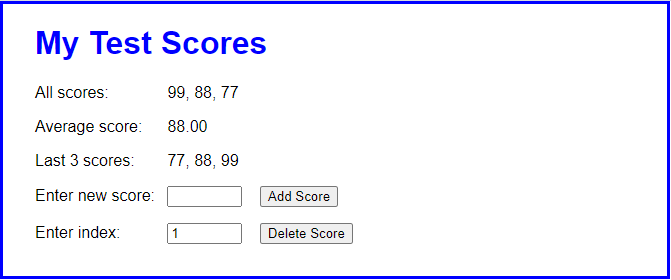
7. Display the last three scores. To do this, use the slice method to get the last 3 elements of the array; use the reverse method to reverse these elements; then use the join method to put a string version of the array in the label with id=”last”.

8. Get the input textbox ready for the next entry by clearing the value and putting the focus in it. Ensure that the textbox also has the focus when the application starts.

9. Make any other adjustments you see necessary.

# Lab 10.2 Add a Delete Score button to the Test Scores application

This exercise builds on part 1 to add a Delete Score button.



1. Open the files in the /test\_scores\_delete/ folder. Notice that the html file has a new label, textbox and button to delete a score.

2. You will have to copy the code you completed in part 1 to the JavaScript file and test to make sure the Add Score functionality is working.

3. In the ‘DOMContentLoaded’ event handler, add an event handler called deleteScore to the click event of the Delete Score button.

4. Code the deleteScore event handler. It should start by getting the index entry as an integer. Then, it should use the splice() method to delete the element and redisplay the list of scores. For now, don’t worry about updating the average score or the list of the last three scores.

5. Run the application, add one or more scores, and test the Delete Score button.

6. Add data validation so that the user’s entry must be a number and has to be an index value that’s in the array. If either data validation fails, display a message to the user in the span element that follows the Delete Score button. Then, test this change.

7. Move the code that calculates and updates the DOM with all scores, the average score, and the last 3 scores to a helper function. Then call this function from the addScore() and deleteScore() functions to display the results.

8. Add code to the Delete Score event handler that clears the Index text box and moves the focus to the Score text box. Then, test the application one more time to be sure everything is working properly.

# Lab 10.3 Modify the Test Scores application to use LocalStorage

In this exercise you will build on the previous application to save the scores in Local Storage. i.e. if the user restarts the browser they will not lose the data they’ve already entered.

1. Make a copy of the /test\_scores\_delete/ folder and call it /test\_scores\_storage/

2. In the addScore function, just after you add a score to the array, add a new line that saves the scores array in a variable called ‘scores’ in localStorage. To do this, use the array join() method to convert the array to a string with a comma (,) between each element.

3. In the deleteScore function just after you delete a score from the array, add a new line that saves the scores array in a variable called ‘scores’ in localStorage, same as in step 2.

4. We now need to ensure that the data in localStorage is copied into the array when the application starts. To do this, add code to the start of the processScores function that reads the scores string from localStorage. You should check if the scores array length = 0, which means that we may have just started the application, so we should read the values from localStorage and populate the array. Add the code to do this.