
Web-Based Application Design and Development (ITIS 3135)

In-class Activity #8: Test Score Array

The goal of this lab is to work with an array and you'll add nodes to the DOM to display the Results and the Scores.

Part 0: Download files

1. Download **Activity8.html**, **Activity8.css** and **Activity8.js**.
2. Then, run the application to see the user interface shown below, although that interface won't do anything until you develop the JavaScript for it.
3. Verify there are no errors in the HTML and CSS files.
4. Rename the files to ***Lastname1Lastname2-Activity8.html***, ***Lastname1Lastname2-Activity8.css*** and ***Lastname1Lastname2-Activity8.js***.

Use a Test Score array

Name:

Score:

Results

Average score = 90
High score = Mike with a score of 99

Scores

Name	Score
Ben	88
Joel	98
Judy	77
Anne	88
Mike	99

Part 1: Make the following modifications

1. At the start of the JavaScript file, you'll see the declarations for two arrays: one for names and one for scores, and each array contains four elements. You'll also see the code for the \$ function as well as an onload event handler that attaches three functions named *addScore()*, *displayResults()*, and *displayScores()* to the click events of the buttons.

Web-Based Application Design and Development (ITIS 3135)

2. Write the `displayResults` function(). It should derive the average score and the highest score from the arrays and then display the results in the `div` element with “results” as its `id`, as shown above. To display the results, you need to add nodes to the DOM with the heading as an `h2` element and the average and highest scores as `<p>` elements. The easiest way to do that is to use the `innerHTML` property as shown in figure 6-13 in your Murach text.
3. Write the `displayScores()` function. It should get the names and scores from the arrays and display them as rows in the HTML table element with “scores_table” as its `id`, as shown above.
4. Write the `addScore()` function. It should add a name and score to the two arrays. To test whether this works, you can click the Display Scores button and see if the new name and score have been added to the table.
5. If you haven’t already done it, add data validation to `addScore()` function. The Name entry must not be empty and the Score entry must be a positive number from 0 through 100. If either entry is invalid, use the `alert()` method to display this error message: “You must enter a name and a valid score”.
6. Make sure that your application moves the cursor to the Name field when the application starts and after a name and score have been added to the array.
7. Verify there are no errors.

Part 3: Upload ALL files to you and your partner’s public_html folder and verify the URL.

Part 4: Turn in your activity

1. Each partner should upload the *.html*, *.css* and *.js* files to Canvas, and place in the Comments their URL and partner’s name.