
Web-Based Application Design and Development**Activity #8: Height array**

The goal of this activity is to work with an array and also add nodes to the DOM to display the Results and the Heights.

Part 0: Download files

1. Download **Activity8.zip** folder and unzip the folder to get the **Activity8.html**, **Activity8.css** and **Activity8.js** files.
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 ***Lastname-Activity8.html***, ***Lastname-Activity8.css*** and ***Lastname-Activity8.js***.

Heights of Students

Use of Array with Heights

Name:

Height:

Results

Average Height = 69
Highest height = Mike with a height of 80

Heights

Name	Heights
Tom	65
Lily	60
Jose	72
Sarah	68
Mike	80

[Validate HTML](#)
[Validate CSS](#)



Web-Based Application Design and Development

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 heights, and each array contains four elements. You'll also see the code for the \$ function as well as the code for registering three functions named `addHeight()`, `showResults()`, and `displayHeight()` to the click events of the buttons.
2. Write the **showResults ()** function. It should derive the average height and the highest height from the arrays and then display the results in the div element with "result" as its id, as shown above. To show the results, you need to add nodes to the DOM with the heading "Results" as an h2 element and the average and highest height as <p> elements. The easiest way to do that is to use the `innerHTML` property as shown in zyBooks Chapter 6 (6.2).
3. Write the **displayHeight()** function. It should get the names and height from the arrays and display them as rows in the HTML table element with "height_table" as its id, as shown above.
4. Write the **addHeight()** function. It should add a name and height to the two arrays. To test whether this works, you can click the Display Heights button and see if the new name and height have been added to the table.
5. If you haven't already done it, add **data validation** to **addHeight()** function. The Name entry must not be empty and the Height 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 height".
6. Make sure that your application **moves the cursor to the Name field** when the application starts and after a name and height have been added to the array (or "Add heights to Array" button is clicked).
7. Make sure there are no JS errors.
8. Make sure to verify that you have no validation errors on the page. To validate your HTML/CSS you can use either of the three methods- "By URI" (copy paste specific file path on the server), "By file upload" (uploading the file), or "By direct input" (copy paste the content of the file) on the validation link.

Part 3: Upload files to your Exercise folder in your webpages account

Verify that your final webpage is accessible by going to
<https://webpages.uncc.edu/yourUsername/Exercise/Lastname-Activity8.html>

Part 4: Turn in your activity

1. Log in to Canvas.
2. Go to the Activity #8 Submission link on the Canvas page. Upload the upload the .html, .css and .js files to the submission page.
3. In the Comments textbox of Canvas provide:
 - a. the URL of your html file on the web server. You must provide the correct URL to get credit.