



JavaScript DOM

Basic

Duration: 4 Hour

JavaScript Dom

Exercises 01

Add a label to each of the input fields: username, password, confirm password

Exercise 02

Add a required validation to each input that shows an error message next to the entry if it does not have any text entered.

Exercise 03

Add a further validation to check if the user input in the password and confirm password inputs match. Show an error message if they do not.

Exercise 04

Ensure the 'Register' button is disabled until the user has entered valid data into all the input fields. Once they have, the registration button should then be enabled.

Exercise 05

When the user clicks the 'Register' button, a message should be displayed informing them of a successful user registration.

Mid

Exercise 06

Add a required validation to each input that shows an error message next to the entry if it does not have any text entered.

Result :



Registration Form

Username:

Required

Password:

Required

ConfirmPassword:

Required

Register

Exercise 01

Highlight all of the words over 8 characters long in the paragraph text (with a yellow background for example)

Exercise 02

Add a link back to the source of the text after the paragraph tag.

(<https://google.com/>)

Exercise 03

Split each new sentence on to a separate line in the paragraph text.

A sentence can be assumed to be a string of text terminated with a period (.)

Exercise 04

Count the number of words in the paragraph tag and display the count after the heading.

You can assume that all words are separated by one singular whitespace.

Result :



Heading

Hey, you're not permitted in there. It's restricted. You'll be deactivated for sure.. Don't call me a mindless philosopher, you overweight glob of grease! Now come out before somebody sees you. Secret mission? What plans? What are you talking about? I'm not getting in there! I'm going to regret this. There goes another one. Hold your fire. There are no life forms. It must have been short-circuited. That's funny, the damage doesn't look as bad from out here. Are you sure this things safe? Close up formation. You'd better let her loose. Almost there! I can't hold them! It's away! It's a hit! Negative. Negative! It didn't go in. It just impacted on the surface. Red Leader, we're right above you. Turn to point... oh-five, we'll cover for you. Stay there... I just lost my starboard engine. Get set to make your attack run. The Death Star plans are not in the main computer. Where are those transmissions you intercepted? What have you done with those plans? We intercepted no transmissions. Aaah....This is a consular ship. Were on a diplomatic mission. If this is a consular ship...were is the Ambassador? Commander, tear this ship apart until you've found those plans and bring me the Ambassador. I want her alive! There she is! Set for stun! She'll be all right. Inform Lord Vader we have a prisoner. What a piece of junk. She'll make point five beyond the speed of light. She may not look like much, but she's got it where it counts, kid. I've added some special modifications myself. We're a little rushed, so if you'll hurry aboard we'll get out of here. Hello, sir.

Advanced

3-Create a simple shopping list application that allows adding items to the list by inputting them into a form and clicking a button.

When an item is added, it should appear on the list and have a button next to it to delete it.

The form input should be cleared and focused after each item is added.

Output:

My shopping list

Enter a new item: Add item

- Eggs Delete
- Milk Delete
- Bread Delete
- Humous Delete



4-In this exercise, you'll work with an array and you'll add nodes to the DOM to display the Results and the Scores.

1. Open the HTML and JavaScript files Then, run the application to see the user interface shown above, although that interface won't do anything until you develop the JavaScript for it.

2. 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.

3. 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 elements. The easiest way to do that is to use the innerHTML property as shown in figure 6-13.

4. 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.

5. 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. 16 Extra exercises for Murach's JavaScript and jQuery (3rd Edition)

6. 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".

7. 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.



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