Lab 11: JavaScript Arrays

Due date: Nov 4, 2023, 4:00 am

Overview

In this lab, you will implement the behavior of a simple note-taker application that allows the user to add and remove notes. You will use an array to maintain the notes.

The Starter Project

The starter project consists of

- An HTML file. The HTML file implements a page with three elements.
  - The first element is supposed to display the collection of notes as an unordered list.
  - The second element contains a form with an <input> element of type `text` and an add button. If the add button is clicked, the entered value in the input field should be added as a new note to the collection of notes. Since the functionality is not yet implemented, nothing happens if you click the add button in the starter project.
  - The third element contains a form with a drop-down box that should be populated by the notes and with a delete button. If the delete button is clicked, the selected note should be deleted from the collection of notes.

    You should not make any changes to the HTML file unless you attempt the Bonus part.
- A style sheet. You should not make any changes to the style sheet unless you attempt the bonus part.
- An empty JavaScript file. You will implement the behavior of the note-taker application in this file.

Solution Demo

See the short video clip on the course website under the Lab 11 instructions for a demonstration of the completed functionality of this note-taker application.

Instructions

All code should be implemented in the file `notetaker.js`. For the optional Part 4, you may make changes to the HTML and stylesheet.

Part 1: Updating page content

Declare a variable that is initialized to be an empty array. The variable will track a collection of strings, each string representing a note. In the following, the array is referred to as the array of notes.

Implement a function that updates the page content:

1. If the array of notes is empty, the element with id `display-notes` and the element with id `delete-note` are hidden on the web page. You can use the same technique as used in Lab 10 to hide these elements.
2. If the array of notes is not empty:
    a. Ensure that the element with id `display-notes` and the element with id `delete-note` are shown on the web page.
    b. Iterate through the array elements in the order of the indices. For each array element:
        i. Add an <li> element to the inner HTML of the unordered list with id `notelist`. (An alternative and more elegant way of adding an <li> element is the use of the `appendChild` method. Feel free to explore and use this alternative.)
        ii. Add an <option> element to the inner HTML of the <select> element with id `noteselection`. The value attribute of the option element should be set to the corresponding index of the array element. (Again, you could also use the `appendChild` method.)

Call the function that updates the page content such that the page content is updated whenever the note-taker page is displayed in a browser.

Part 2: Adding notes

Implement a function that adds the note as entered in the <input> element with id `newnote`:

1. Add the value of the <input> element with id `newnote` to the end of the array of notes. You do not need to validate the value. That means it is okay if a new note is an empty string.
2. Clear the input field, i.e., set the value of the <input> element with id `newnote` to an empty string.
3. Update the page content to reflect the changes to the notes.

Register the function as an event handler with the <button> element with id `add` for the click event.

Part 3: Deleting notes

Implement a function that deletes the note selected in the drop-down box with id `noteselection`:

1. Retrieve the value of the <select> element. Note that the value is an index represented as a string.
2. Remove the element of the array of notes at the retrieved index. After the remove operation, all notes should still be stored consecutively in the array, i.e., the array should not have an undefined value at that index. Hint: Check out the method `splice` of the Array class. Due to JavaScript's automatic conversion between types, you may not need to explicitly convert the string with the index to a number.
3. Update the page content to reflect the changes to the notes.

Register the function as an event handler with the <button> element with id `delete` for the click event.

Part 4: Validation (Optional)

Part 4 is optional. You will receive extra credit if you complete this part correctly.

The current solution allows empty notes to be added. Modify the code so that empty notes are not added. If the user attempts to add an empty note, an error message should be displayed. The error message should be hidden again as soon as a valid note is submitted or as soon as an existing note is deleted. Modify the HTML file and the stylesheet as necessary.

Submission

Archive the folder that contains all files with your solution using the ZIP format and submit your solution on the course website before the due date. Make sure that the name of the submitted zip file contains your name.

Grading

One point will be subtracted for each error including missing repository and inappropriate indentation.

Part 1: 6 points

Part 2: 2 points

Part 3: 2 points

Part 4: 2 bonus points