ASSIGNMENT: JAVASCRIPT INTRODUCTION

In this assignment you will get started with the JavaScript language in a browser environment and on how to read and manipulate the HTML DOM.

This assignment will often mention methods and techniques but may not explain everything fully. If this is the case, it is up to you to research said method or technique and implement them accordingly.

Difficulty: ☆☆☆☆

Learning objectives:

- I remember JavaScript can read and manipulate a webpage via the DOM.
- I can find a specific node in the DOM using method *document.getElementById*.
- I can create arrow functions in JavaScript.
- I can add event listeners to DOM nodes.
- I can prevent the default action of a DOM event.
- I can use the following Array methods: 'find', 'map' and 'forEach'.
- I can perform exact comparisons between variables.
- I can log messages to the browser console.

Estimated time required: 60 minutes

1. INSPECTING THE PROVIDED PROJECT

The project consists of a login screen, built using 3 files: an HTML file for the structure of the page, a CSS file for the styling, and a JavaScript file for executing interactive code inside of the browser.

You can open the project folder in Visual Studio Code, to easily access and edit all the concerning files.

The JavaScript file is already set up to be loaded, using a 'script' tag, inside of the HTML file. Of course, for JavaScript to manipulate the HTML DOM, the DOM needs to exist before the JavaScript code gets executed. That is why the 'script' tag is placed at the end of the body. HTML is loaded and executed line by line, and thus by placing the 'script' tag at the end, the JavaScript code will be executed after the rendering of the DOM.

When you open the provided HTML file in your browser, you should be able to see the login screen with custom styling. Now please open the browser developer tools. How to do this may defer between browsers, but mostly pressing 'F12' should do the trick. There are all kinds of tools in here, to assist you during web development, but today we will focus on the 'console'. If all is well, you should already be able to see a first message in there, telling you the users accounts have been initialized. This is done, using the method 'console.log', inside of 'main.js'.

2. Implementing the login feature

To simulate the login feature, we will first need to find the form node in the DOM, add an event listener to that node for the 'submit' event, and then in that event listener check the user accounts and log a message, showing us the result.

Here are the necessary steps:

1. The 'submit' event is fired by the form, when a user pressed the submit button, or does so indirectly by pressing the 'enter' button on their keyboard. Thus let us first find the form node in the DOM, using the method 'document.getElementById' and the unique 'id' that is already set to the node in HTML.
2. Now that we have the form node, we can add an event listener to it using method 'addEventListener'. Again, attach the listener to the 'submit' event.
3. Forms, when submitted, instruct the browser to send its data to another URL and load that page. Since we only want to execute our own code, we want to prevent this behavior. To do this, we will call the method 'preventDefault' on the event object, that we received in our event listener.
4. In the event listener, we will now also need to verify whether the login attempt matches any of the provided accounts. For this, you will need to find the input fields, get their input, and compare it to the predefined user accounts by calling the 'find' method on the array of user accounts, and comparing the usernames and passwords.
5. Finally, log, whether the login attempt was successful, to the console, using the 'console.log' method.

3. Clearing the form

The next feature is to clear the input fields of the form, by pressing the 'clear' button. For this, you will need to find 'clear' button, add an event listener to the 'click' event, and of course find and clear the input fields in said event listener.

4. Logging the usernames

This last feature does not make much sense in the real world but turns out to be a nice way to practice the use of array methods. When pressing the button, we would like to log all the usernames one-by-one.

To do this, first use the array method 'map' to map the list of user accounts to a list of usernames, and then use the array method 'forEach' to loop through this list and log the usernames.