Web Programming 261

JavaScript DOM [Creating HTML Elements]

BELGIUM CAMPUS
iTversity

Outcomes

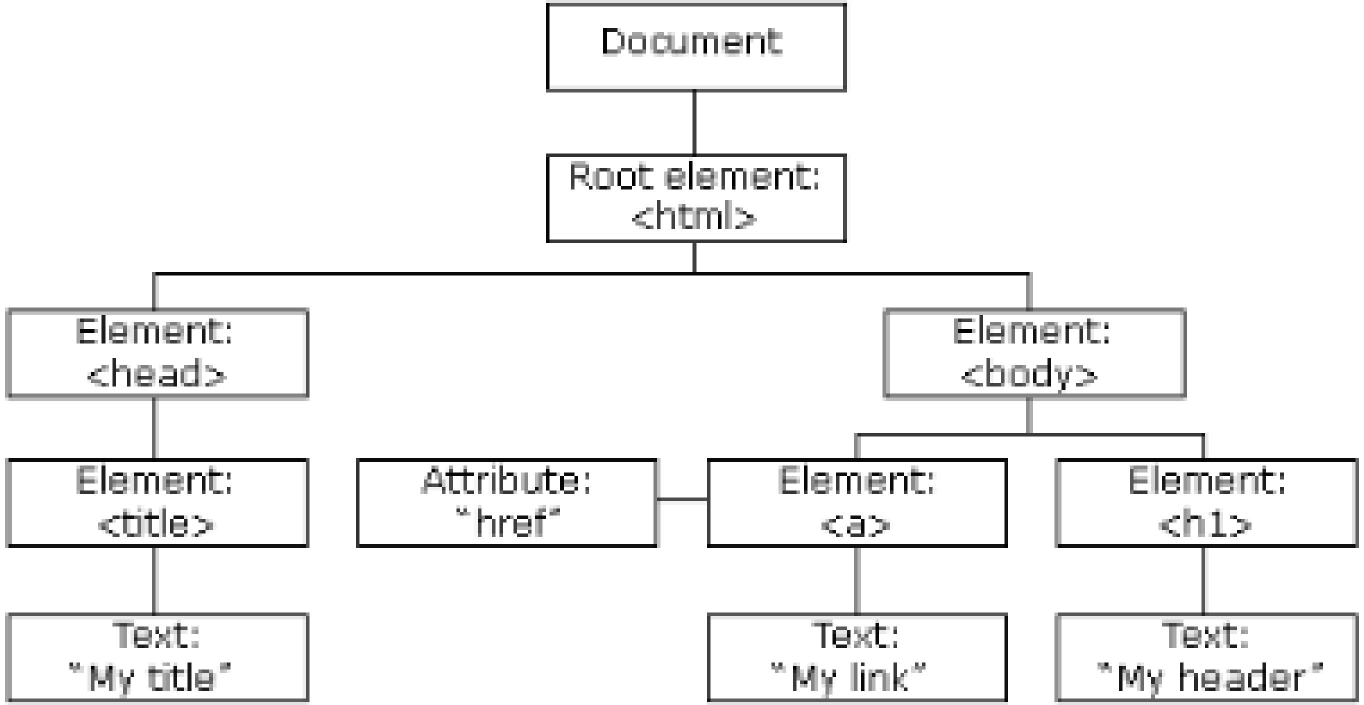
Students should understand the following outcomes, upon successful completion of this module:

- Intro to the DOM
- document.createElement()
- appendChild()
- Creating new HTML Elements
  - Paragraph
  - List
  - Table

DOM

When a web page is loaded, the browser creates a **D**ocument **O**bject **M**ode (DOM) of the page.

DOM, represents *all page content* as *objects* that can be *modified*.



Creating New Elements on the DOM

- In JavaScript, we can create completely new HTML Elements (Nodes) from the app.js file.

- This is useful when you want to dynamically update a page or build User Interfaces (UIs).

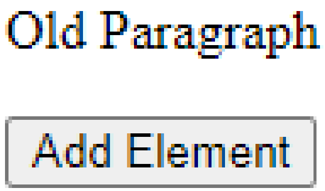
- To add a new element to the HTML DOM, you must create the element (element node) first, and then append it to an existing element.

- *document.createElement()* is a method in JavaScript that allows you to dynamically create an HTML element within the context of a web page.

- It is part of the Document Object Model (DOM) API, which provides a structured representation of a web page's content and allows you to manipulate that content programmatically.

Creating New Elements on the DOM

- Example 1: Lets create the following in HTML:



Old Paragraph

Add Element

```
<body>
    <p>Old Paragraph</p>

    <div id="container">
        <button id="addButton" onclick="addParagraph()">
            Add Element
        </button>
    </div>
</body>
```

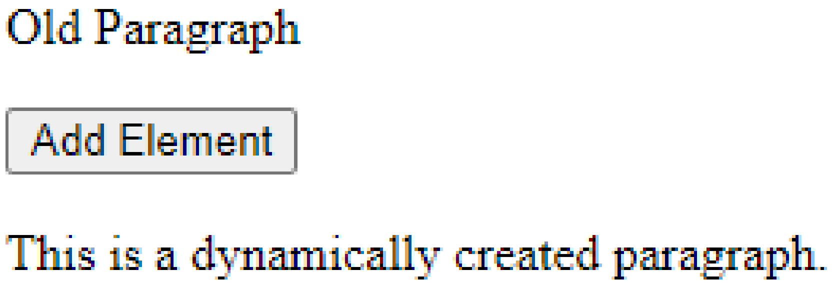
Index.html

document.createElement()

- When the button is clicked, the JavaScript function should run

```
// Function to add a new paragraph
function addParagraph() {
    // Get references to the container div and the add button
    var container = document.getElementById('container');
    // Create a new <p> element
    var newParagraph = document.createElement('p');
    // Set the text content of the paragraph
    newParagraph.textContent = 'This is a dynamically created paragraph.';
    // Append the new paragraph to the container
    container.appendChild(newParagraph);
}
```

app.js



Old Paragraph

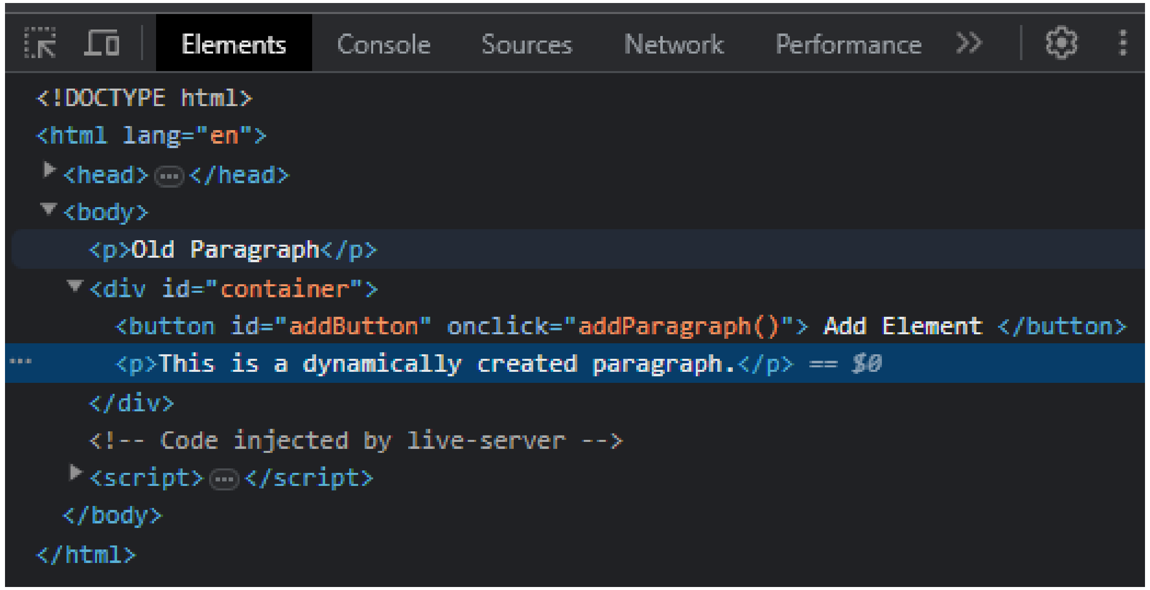
This is a dynamically created paragraph.

document.createElement()

- In this JavaScript file (script.js), we:
  - Define a function called *addParagraph().*
  - Get references to the container <div> using *document.getElementById('container').*
  - Create a new <p> element using *document.createElement().*
  - Sets the new text content of the paragraph.
  - Appends the new paragraph to the container <div> using *container.appendChild(newParagraph).*

Result:

- The page includes a container <div> with an "Add Element" button.
- An onclick attribute points to the addParagraph() function, which runs and as a result, a new paragraph with the text "This is a dynamically created paragraph." is added to the container.

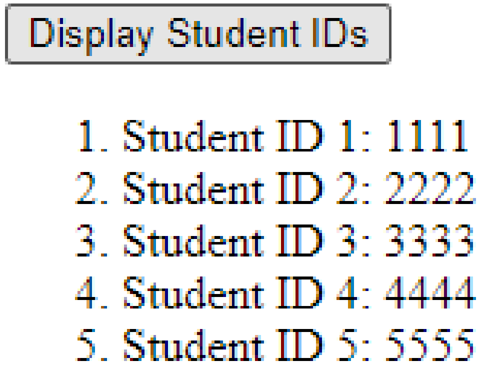


```
<!DOCTYPE html>
<html lang="en">
 ▶<head>⋯</head>
 ▼<body>
    <p>Old Paragraph</p>
   ▼<div id="container">
      <button id="addButton" onclick="addParagraph()"> Add Element </button>
      <p>This is a dynamically created paragraph.</p> == $0
    </div>
    <!-- Code injected by live-server -->
   ▶<script>⋯</script>
  </body>
</html>
```

document.createElement()

Example 2:

- Get 5 student IDs from the browser using prompt in JavaScript.
- Use a *displayStudentIDs()* function, get each ID, store each in an array.
- Create a <ol> element, and pass to it new <li> elements, which will then display IDs as follows:



1. Student ID 1: 1111
2. Student ID 2: 2222
3. Student ID 3: 3333
4. Student ID 4: 4444
5. Student ID 5: 5555

Hint: make use of the onclick() attribute, loops, arrays, querySelector(), createElement(), appendChild() to achieve this.

Inline Event Handling Using onclick()

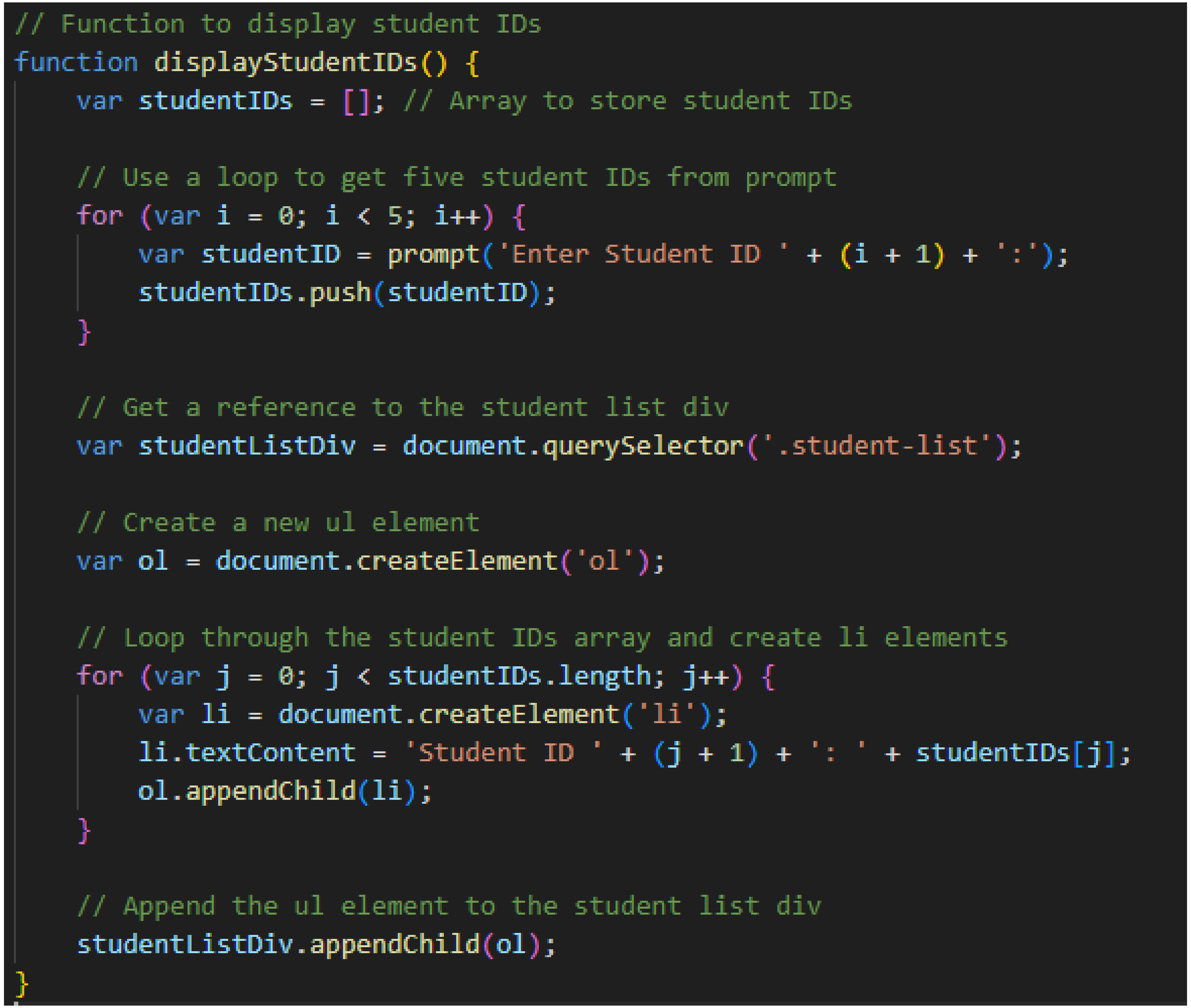
Solution:

- Lets create the following in HTML:

```
<div>
    <button id="displayButton" onclick="displayStudentIDs()">
        Display Student IDs
    </button>
</div>

<div class="student-list"></div>
```

Index.html

Solution:



```javascript
// Function to display student IDs
function displayStudentIDs() {
    var studentIDs = []; // Array to store student IDs

    // Use a loop to get five student IDs from prompt
    for (var i = 0; i < 5; i++) {
        var studentID = prompt('Enter Student ID ' + (i + 1) + ':');
        studentIDs.push(studentID);
    }

    // Get a reference to the student list div
    var studentListDiv = document.querySelector('.student-list');

    // Create a new ul element
    var ol = document.createElement('ol');

    // Loop through the student IDs array and create li elements
    for (var j = 0; j < studentIDs.length; j++) {
        var li = document.createElement('li');
        li.textContent = 'Student ID ' + (j + 1) + ': ' + studentIDs[j];
        ol.appendChild(li);
    }

    // Append the ul element to the student list div
    studentListDiv.appendChild(ol);
}
```

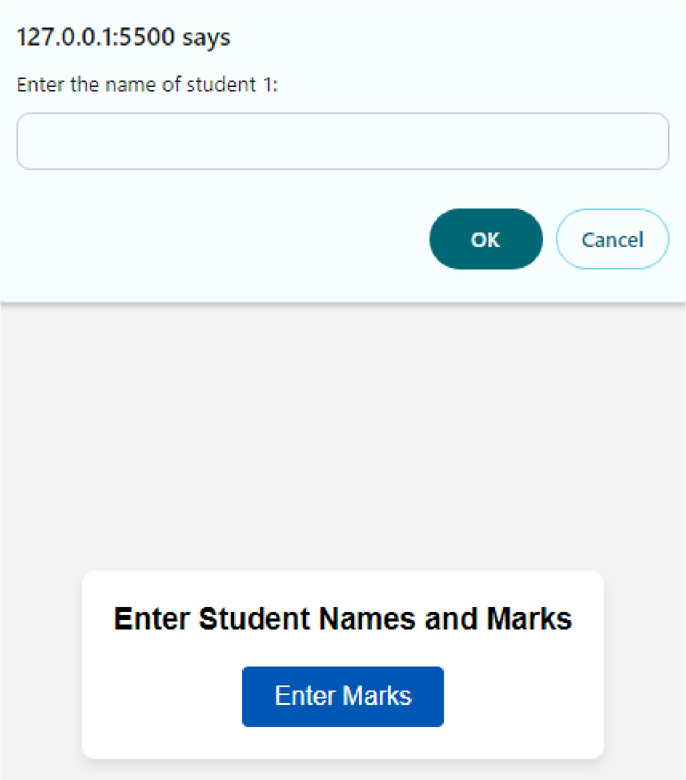
app.js

document.createElement()

Example 3:

- Use the addendum provided to get 5 student names and their marks from the browser, using prompt in JavaScript.

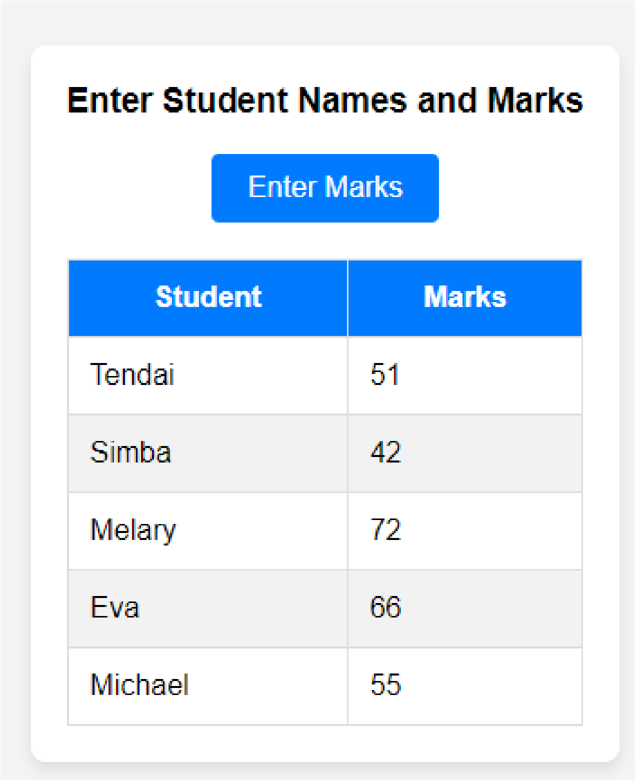
  - Display the marks in a new table element:



127.0.0.1:5500 says

Enter the name of student 1:

Enter Student Names and Marks

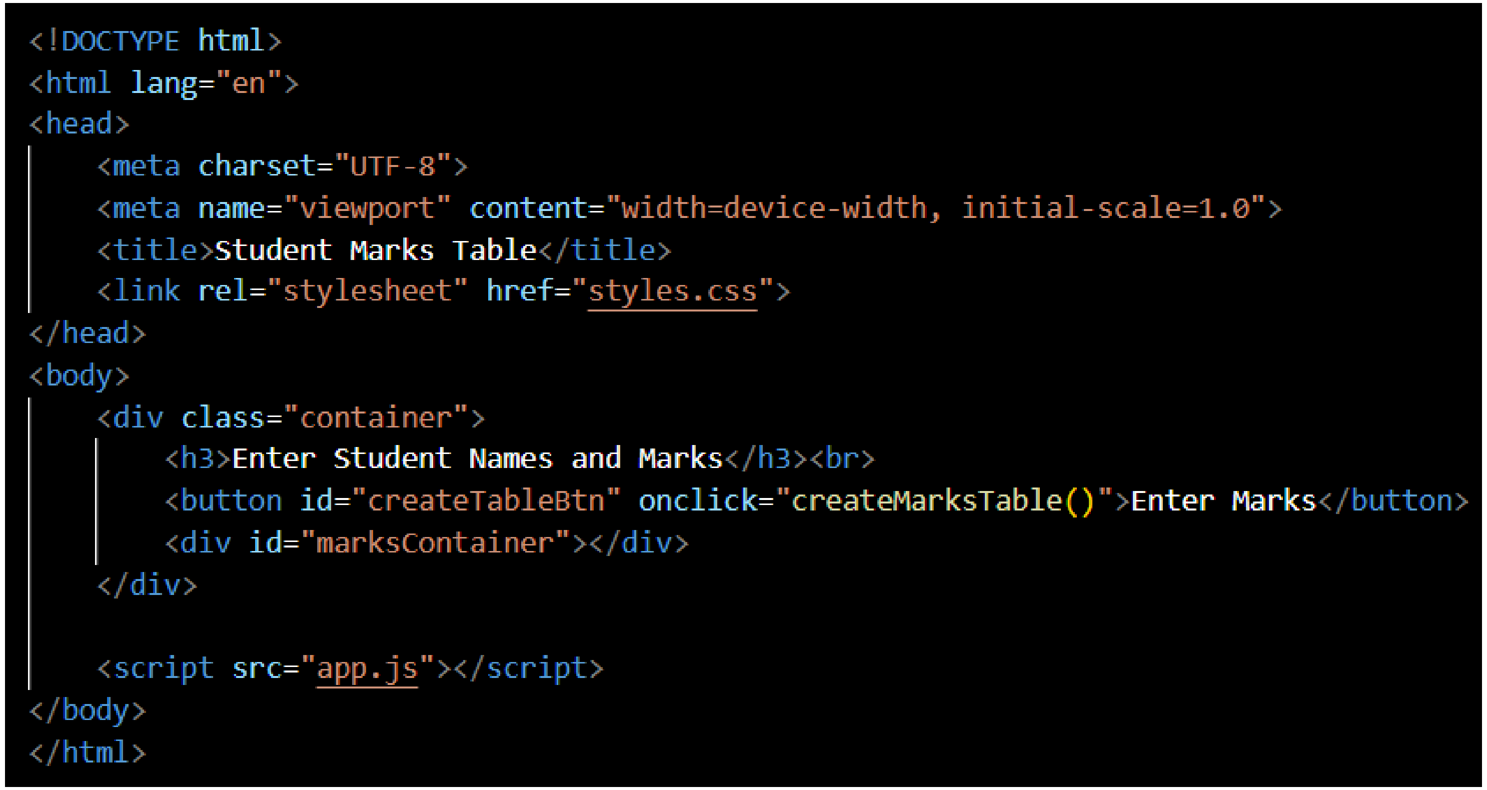


Enter Student Names and Marks

Student	Marks
Tendai	51
Simba	42
Melary	72
Eva	66
Michael	55

Hint: make use of the onclick() attribute, loops, arrays, querySelector(), createElement(), appendChild() to achieve this.

Solution:

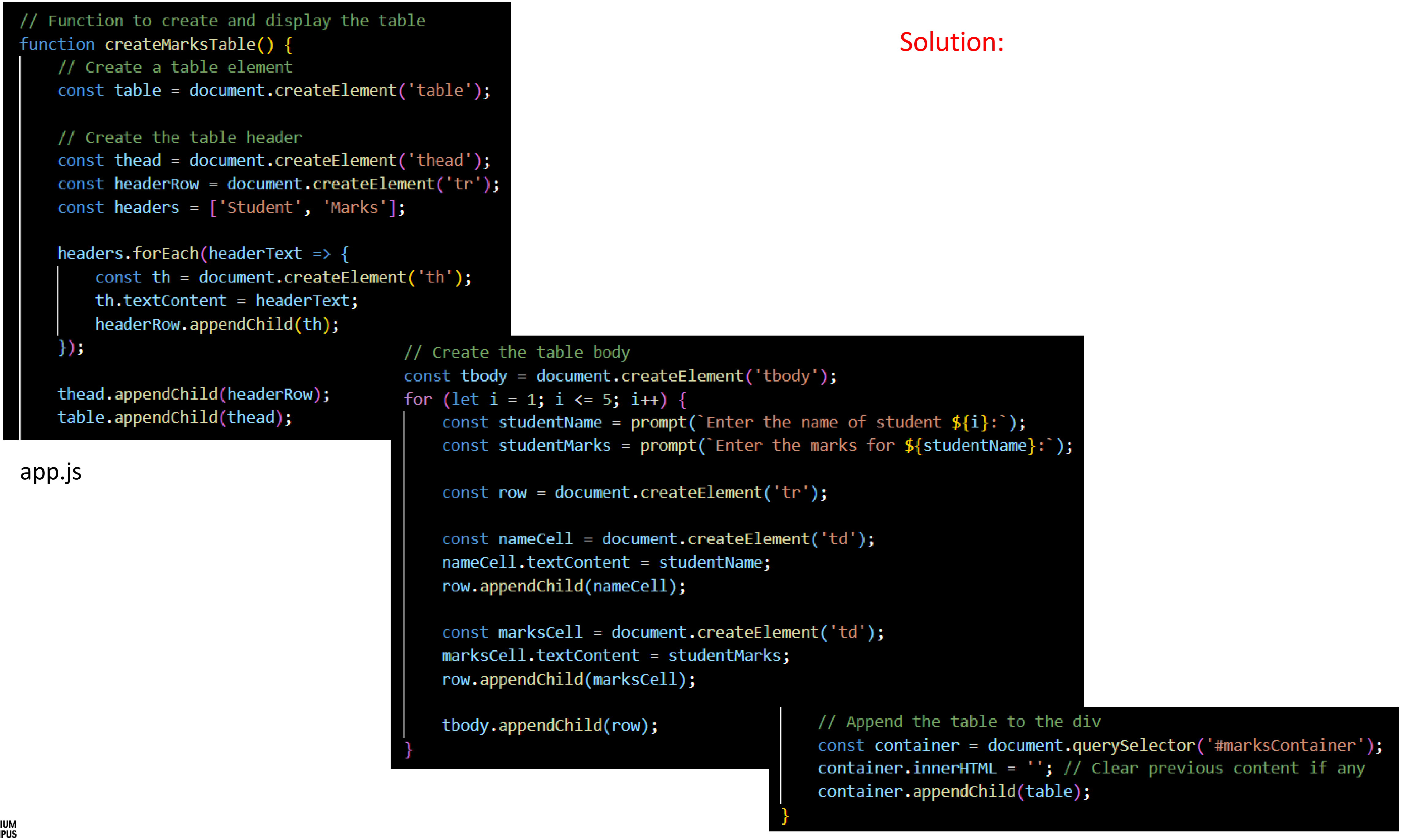


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Marks Table</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h3>Enter Student Names and Marks</h3><br>
        <button id="createTableBtn" onclick="createMarksTable()">Enter Marks</button>
        <div id="marksContainer"></div>
    </div>

    <script src="app.js"></script>
</body>
</html>
```

Index.html

- Use the CSS file in the addendum

Solution:



```javascript
// Function to create and display the table
function createMarksTable() {
    // Create a table element
    const table = document.createElement('table');

    // Create the table header
    const thead = document.createElement('thead');
    const headerRow = document.createElement('tr');
    const headers = ['Student', 'Marks'];

    headers.forEach(headerText => {
        const th = document.createElement('th');
        th.textContent = headerText;
        headerRow.appendChild(th);
    });

    thead.appendChild(headerRow);
    table.appendChild(thead);
```

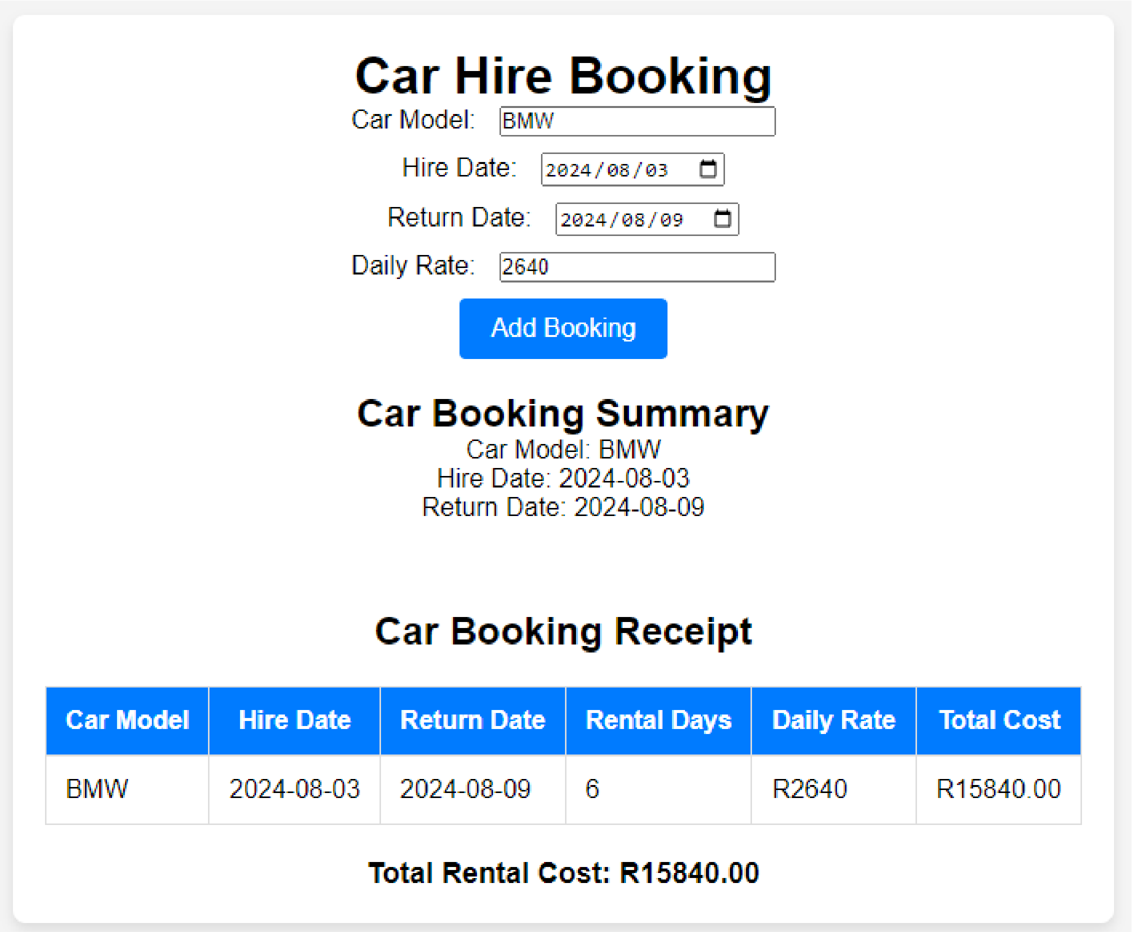
```javascript
    // Create the table body
    const tbody = document.createElement('tbody');
    for (let i = 1; i <= 5; i++) {
        const studentName = prompt(`Enter the name of student ${i}:`);
        const studentMarks = prompt(`Enter the marks for ${studentName}:`);

        const row = document.createElement('tr');

        const nameCell = document.createElement('td');
        nameCell.textContent = studentName;
        row.appendChild(nameCell);

        const marksCell = document.createElement('td');
        marksCell.textContent = studentMarks;
        row.appendChild(marksCell);

        tbody.appendChild(row);
    }
```

```javascript
    // Append the table to the div
    const container = document.querySelector('#marksContainer');
    container.innerHTML = ''; // Clear previous content if any
    container.appendChild(table);
}
```

app.js

Exercise:

Use the addendum with the HTML and CSS and add functionality to the following website in JavaScript:



Car Hire Booking

Car Model:	BMW
Hire Date:	2024/08/03
Return Date:	2024/08/09
Daily Rate:	2640

Car Booking Summary

Car Model: BMW
Hire Date: 2024-08-03
Return Date: 2024-08-09

Car Booking Receipt

Car Model	Hire Date	Return Date	Rental Days	Daily Rate	Total Cost
BMW	2024-08-03	2024-08-09	6	R2640	R15840.00

Total Rental Cost: R15840.00

- The website is a car booking application that adds a booking from details on the form.
- Use JavaScript to add a new paragraph element to display the Car Booking Summary.
- Use JavaScript to add a new table element to display all data of the booking as a receipt, including the total cost, as shown.

Thank You!

THE END

info@belgiumcampus.ac.za +27 10 593 53 68

united we conquer

/belgiumcampusSA

#Belgium Campus

/belgiumcampus

References

https://www.freecodecamp.org/news/here-are-the-new-built-in-methods-and-functions-in-javascript-8f4d2fd794fa/

https://javascript.info/array-methods

https://www.tutorialrepublic.com/javascript-examples.php

https://ultimatecourses.com/blog/array-reduce-javascript