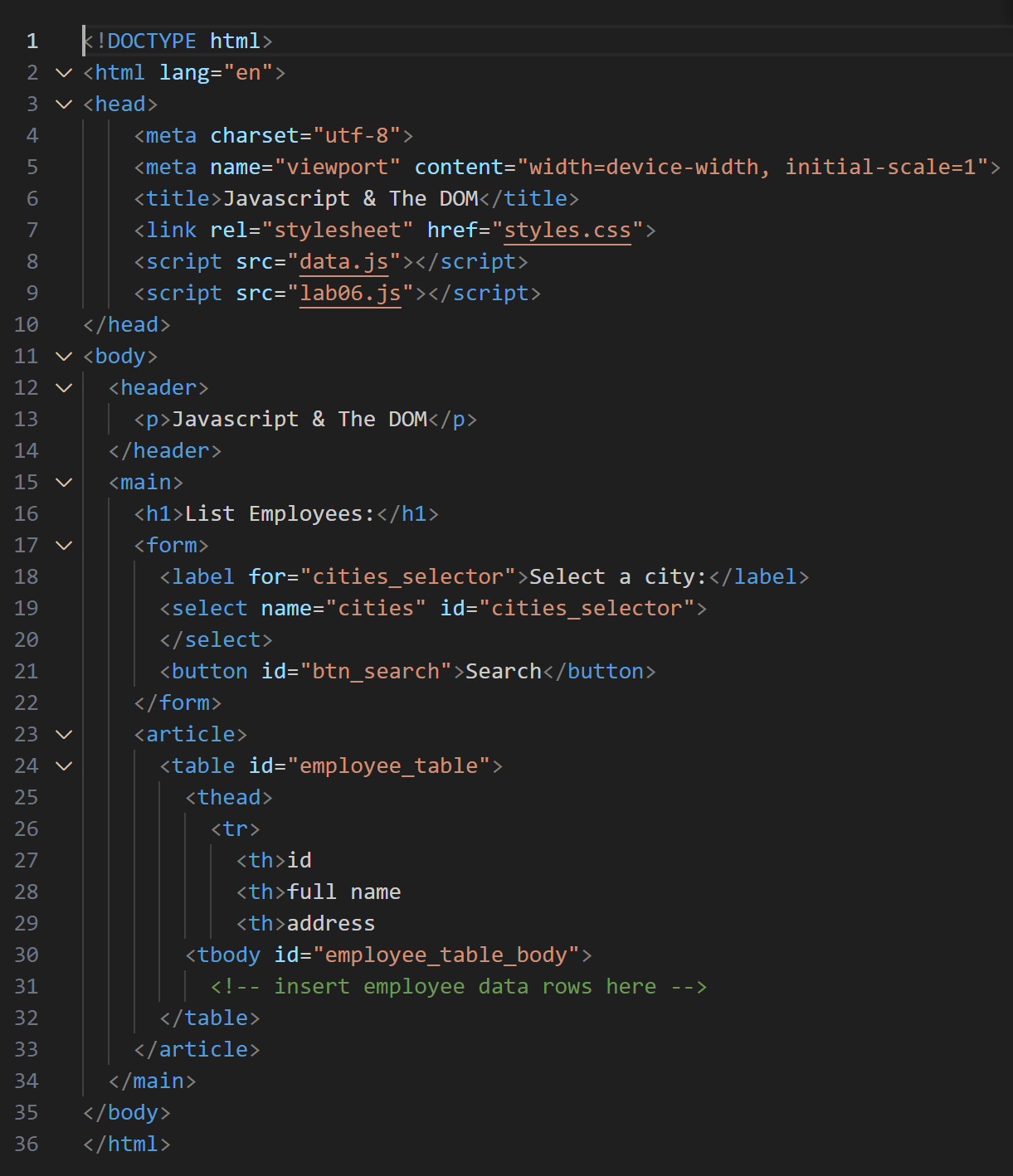
# LAB 06

# JavaScript & the DOM

1. Download the provided starter code from the LMS.
2. Open the index.html file and inspect the code paying attention to the id/names of the different html tags.



1. Click on ctrl+shift+i to open the developer tools of chrome. Notice that the data in the **data.js** file has been logged into the console.

const run = () => {

  alert("button pressed");

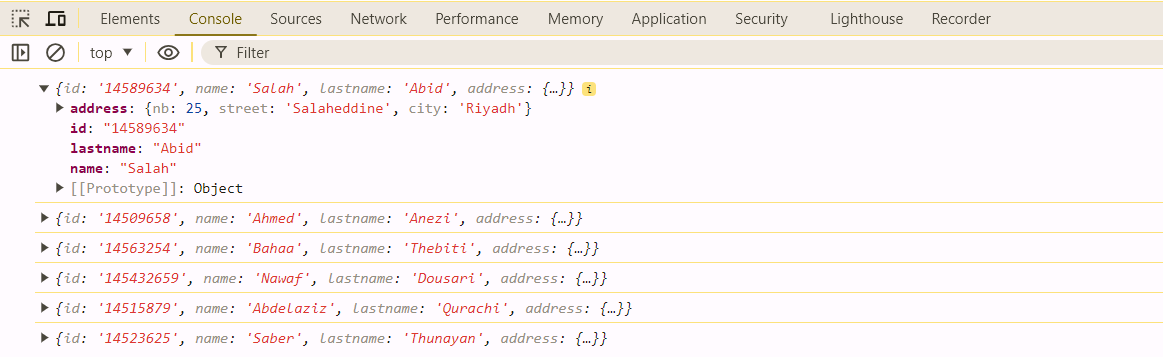
};

for (let d of data) {

  console.log(d);

}

**Result:**



In the following, never modify the **data.js** file

1. In the file lab06.js, create a function expression that accepts an employee object as argument and generates and returns a **tr node** using the DOM API (not string generation).

Use ***document.createElement(“tr”)*** to create a table row for example.

Use ***node.appendChild(newNode)*** to append a new node into an existing one.

**Explaining the requirement:**

Here, we want to create a function that creates the HTML code to host one employee data as shown below:



Therefore, we will need one <tr> and three <td> to form a row that looks like this:

<tr>

<td> </td><td> </td><td> </td>

</tr>

Notice that the questions asks for creating this structure using DOM API which simply means using ***document.createElement and node.appendChild(newNode)*** instead of using string method.

***Follow the following steps to perform this task:***

1. Define a function createTRfromEmployee that takes an employee object as input.
2. Create a new table row node (<tr>) using document.createElement("tr").
3. Create three new (td1, td2, td3) table data nodes (<td>) for the employee's id, name/lastname, and address, using document.createElement("td").
4. Using appendChild, set the text content of each td node using document.createTextNode() to display the employee's data. The name and lastname are concatenated, and the address components (nb, street, city) are also concatenated.
5. append the td1, td2, td3 to the tr node.
6. return created tr node.

1. Use the ***DOMContentLoaded*** event of the ***document*** object to create the DOM tree structure of the table into your page when the page is loaded.

**Explaining the requirements:**

Before we create the table, let us create **a copy** of the data array and store it in a variable named:

employees\_displayed. Then, we will create a function that will create the table.

**Follow the steps below to make copy the data array this task:**

1. Initialize an empty array employees\_displayed =[].
2. Iterate over each element (emp) in the data array.
3. push the element (emp) to the employees\_displayed array.

var employees\_displayed = [];

for(let emp of data) {

  // create a second copy with all employees

  employees\_displayed.push(emp);

}

**Follow the steps below to make the createTable function:**

1. Define a function createTable that takes an array of employees (listOfEmployeesToBeDisplayed) as input.
2. Using getElementById to get a reference to the table body element with its ID ("employee\_table\_body").
3. Clear the current content of the table body table\_body.innerHTML = "";
4. Iterate through each employee object (emp) in the input array.
5. for each employee (emp), call the createTRfromEmployee function to create an HTML table row element. appendChild the created table row to the table body (table\_body).
6. Wait for page load, The code waits until the entire page's Document Object Model (DOM) is loaded document.addEventListener("DOMContentLoaded".
7. Call the createTable function, passing the employees\_displayed array.
8. Using getElementById, Empties the existing options from the dropdown element with the ID "cities\_selector".
9. Create (createElement) a new dropdown option element labeled "all" with value "all"

document.addEventListener("DOMContentLoaded", () => {

    // Create the table

    createTable(employees\_displayed);

    // Refill the selector with cities

    const cities\_selector = document.getElementById("cities\_selector");

    cities\_selector.innerHTML = "";

    let opt\_all = document.createElement("option");

    opt\_all.value = "all";

    opt\_all.appendChild(document.createTextNode("all"));

    cities\_selector.appendChild(opt\_all);

});

**Result:** The page displays the filled table and in the select displays "all"

