

Hands-on Lab: Working with Array Iteration for Restaurant Menu List



Estimated time needed: 20 minutes

What you will learn

In this lab, you will explore how JavaScript dynamically generates restaurant menu sections for breakfast, main course, and dessert using JavaScript's array methods. You will learn to use `map` to populate the breakfast menu, `forEach` to display the main course menu items, and a traditional `for` loop to render the dessert items. Additionally, you will understand how HTML content can be updated dynamically, showcasing the power of JavaScript in modifying webpage elements based on predefined arrays.

Learning objectives

After completing this lab, you will be able to:

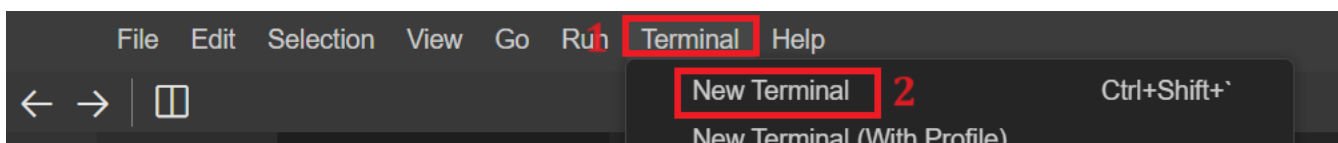
- **Understanding DOM manipulation:** Gain proficiency in accessing HTML elements and dynamically updating their content using JavaScript. Focus on the `getElementById` method to precisely alter specific elements within a webpage.
- **Array iteration methods:** Differentiate between various array iteration methods like `map`, `forEach`, and `for` loops. Observe their distinct functionalities in iterating through arrays and generating structured HTML content.
- **Dynamic web content creation:** Learn the process of dynamically generating and rendering content on a webpage by using JavaScript to create structured menus based on predefined arrays, emphasizing the dynamic nature of web development.
- **Menu section organization:** Grasp the concept of organizing webpage content into sections (e.g., breakfast, main course, dessert) by populating HTML elements with data from corresponding JavaScript arrays, facilitating structured and readable webpage layouts.
- **Application of JavaScript methods:** Apply JavaScript methods (`map`, `forEach`, `getElementById`, `innerHTML`) to manipulate HTML file effectively, showcasing the practical usage of these methods in generating and updating webpage content dynamically based on predefined data.

Prerequisites

- Basic Knowledge of HTML.
- Web browser with a console (Chrome DevTools, Firefox Console, and more).

Step 1: Setting up the environment

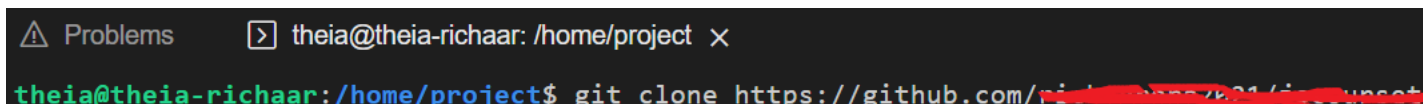
1. Firstly, you need to clone your main repository in the **Skills Network Environment** which you have created in the first lab and where you have pushed all of your previous labs files and folders. Follow given steps:
 - For this click on terminal in right window pane and then select **New Terminal**.



- Perform `git clone` command by writing given command in the terminal.

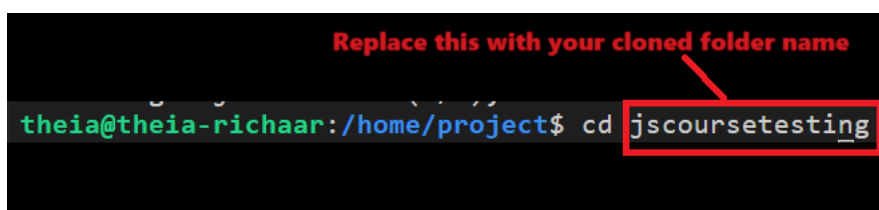
```
git clone <github-repository-url>
```

Note: Put your own GitHub repository link instead of `<github-repository-url>`.



- Above step will clone folder for your GitHub repository under project folder in explorer. You will also see multiple folders inside cloned folder.
- Now you need to navigate inside the cloned folder. For this write given command in the terminal:

```
cd <repository-folder-name>
```



Note: Write your cloned folder name instead of <repository-folder-name>. Perform `git clone` if you have logged out of **Skills Network Environment** and you cannot see any files or folder after you logged in.

- Now, select **cloned Folder Name** folder, right-click on it and click on **New Folder**. Enter folder name as **restaurantMenu**. It will create the folder for you. Then select **restaurantMenu** folder, right-click and select **New File**. Enter file named as **restaurant_menu.html** and click OK. It will create your HTML file.
- Now select **restaurantMenu** folder again, right-click and select **New File**. Enter file named as **restaurant_menu.js** and click OK. It will create your JavaScript file.
- Create basic template structure of HTML file by adding the content provided below.

```
<!DOCTYPE html>
<html>
<head>
  <title>Restaurant Menu</title>
</head>
<body>
  <h1>Restaurant Menu Iteration</h1>
  <h2>Breakfast Menu</h2>
  <div id="breakfastTotalItems"></div>
  <div id="breakfastMenuItems"></div>
  <h2>Main Course Menu</h2>
  <div id="maincourseTotalItems"></div>
  <div id="maincourseMenuItems"></div>
  <h2>Dessert Menu</h2>
  <div id="dessertTotalItems"></div>
  <div id="dessertMenuItems"></div>
  <script src="./restaurant_menu.js"></script>
</body>
</html>
```

- This HTML code provides the structural foundation for displaying a restaurant menu across three sections: breakfast, main course, and dessert. It contains elements such as `<h1>` for the overall title, `<h2>` tags for each menu section, and `<div>` elements with unique IDs (`breakfastTotalItems`, `breakfastMenuItems`, `maincourseTotalItems`, `maincourseMenuItems`, `dessertTotalItems`, `dessertMenuItems`) meant for dynamically displaying menu content.
- However, as of now, there's no actual menu content being populated or displayed dynamically. JavaScript code is needed to generate and insert menu items into the respective `<div>` elements identified by their unique IDs.

Note: When you have pasted the code, save your file.

Step 2: Defining variables and map array method

- In **restaurant_menu.js**, initialize three arrays named as **breakfastMenu**, **mainCourseMenu**, and **dessertMenu**. Initialize these arrays with the menu items as given in the code below. Include this code in the **restaurant_menu.js** file.

```
const breakfastMenu = ['Pancakes', 'Eggs Benedict', 'Oatmeal', 'Frittata'];
const mainCourseMenu = ['Steak', 'Pasta', 'Burger', 'Salmon'];
const dessertMenu = ['Cake', 'Ice Cream', 'Pudding', 'Fruit Salad'];
```

- Now, we will use for loop and two array methods: **map** method and **forEach** method to traverse through these arrays and display the menu items on the HTML page.
- Firstly, we are going to iterate through the **breakfastMenu** array using **map** method. Include the following code in **restaurant_menu.js** file.

```
const breakfastMenuItemsHTML = breakfastMenu.map((item, index) => `<p>Item ${index + 1}: ${item}</p>`).join('');
document.getElementById('breakfastMenuItems').innerHTML = breakfastMenuItemsHTML;
```

- The provided code converts breakfast menu array items into HTML strings using `map()` and an arrow function to structure each item's HTML format.
- Subsequently, a string concatenation method has been used to join the generated HTML strings into one cohesive string using `join("")` to prepare for insertion.
- Finally, this concatenated HTML is dynamically updated within the specific HTML element identified by `breakfastMenuItems` ID with the concatenated HTML string, dynamically populating the webpage with the breakfast menu items in formatted paragraphs `<p>` tags.

Step 3: Defining forEach array method and for loop iteration

- We are going to traverse **mainCourseMenu** array using **forEach** array method. For this, include the below code in the **restaurant_menu.js** file after the previous JavaScript code.

```
let mainCourseItem = '';
mainCourseMenu.forEach((item, index) => {
  mainCourseItem += `<p>Item ${index + 1}: ${item}</p>`;
});
document.getElementById('maincourseMenuItems').innerHTML = mainCourseItem;
```

- The variable `mainCourseItem` is initialized as an empty string `""`. This variable will be used to accumulate HTML strings generated for each main course menu item.
- The `forEach` method loops through each element in the `mainCourseMenu` array. For each item in the array, the arrow function `(item, index) => {...}` is executed. Inside the arrow function, an HTML string is composed for each menu item, incorporating the item's content and its corresponding index.
- Then HTML content is updated dynamically by setting the `innerHTML` property of the HTML element with the ID `maincourseMenuItems` to the accumulated `mainCourseItem` string which will insert main course menu items into the specific element within the webpage.

- Lastly, we will iterate **dessertMenu** array using **for loop** iteration. For this include the below code in **restaurant_menu.js** file after the previous JavaScript code.

```
let dessertItem = '';
for (let i = 0; i < dessertMenu.length; i++) {
  dessertItem += `<p>Item ${i + 1}: ${dessertMenu[i]}</p>`;
}
document.getElementById('dessertMenuItems').innerHTML = dessertItem;
```


- The variable `dessertItem` is initialized as an empty string ("). This variable serves as a container to accumulate HTML strings generated for each dessert menu item.
- The for loop iterates through the `dessertMenu` array, starting from index 0 and continuing until reaching the length of the array. For each iteration, an HTML string is created using the current item in the `dessertMenu` array, including the item's content and its index (`i + 1`).
- The generated HTML content, composed of individual paragraphs `<p>` containing dessert menu items with their respective indices, is assigned to the inner HTML of the HTML element identified by the ID `dessertMenuItems` which will insert the dessert menu items into the specific element within the webpage.

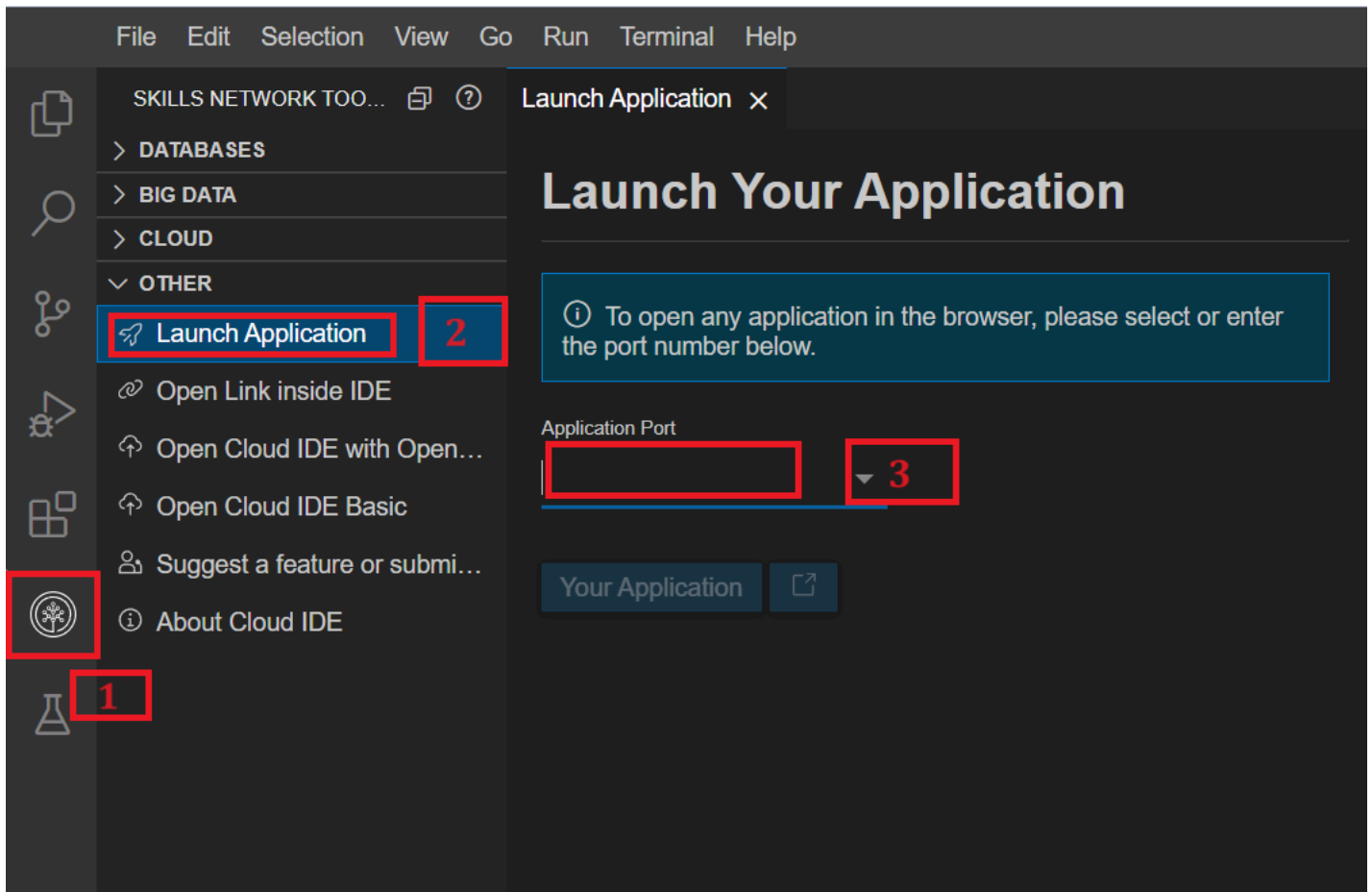
Step 4: Check the output

1. To view how your HTML page, right-click the **restaurant_menu.html** file after selecting this file, then select "Open with Live Server."
2. The server should start on port 5500, indicated by a notification on the bottom right side.

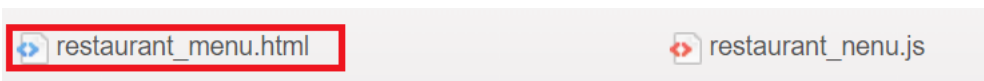


3. Click on the Skills Network button on the left (refer to number 1), it will open the "Skills Network Toolbox". Then click **Launch Application** (refer to number 2).

From there you enter the port no. as 5500 at number 3 and click on this button .



4. It will open your default browser where you will see **cloned-folder-name** folder name. Click on that **cloned-folder-name** folder name and then click on **restaurantMenu** folder name. You will see files related to this folder where you will click again on **restaurant_menu.html** file as shown below.



5. It will open the HTML page and you will see the menu items for **Breakfast**, **Main Course** and **Dessert**.

Restaurant Menu Iteration

Breakfast Menu

Item 1: Pancakes

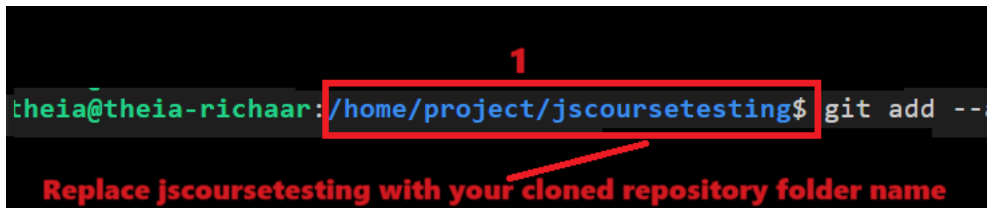
Item 2: Eggs Benedict

Step 5: Perform Git commands

1. Perform `git add` to add latest files and folder by writing given command in terminal in git environment.

```
git add --a
```

Make sure terminal has the path as follows:



2. Then perform `git commit` in the terminal. While performing `git commit`, terminal can show message to set up your `git config --global` for user.name and user.email. If yes, then you need to perform `git config` command as well for user.name and user.email as given.

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"
```

Note: Replace data within quotes with your own details.

Then perform commit command as given:

```
git commit -m "message"
```

3. Then perform `git push` just by writing given command in terminal.

```
git push origin
```

- o After the push command, the system will prompt you to enter your username and password. Enter the username for your GitHub account and the password that you created in the first lab. After entering the credentials, all of your latest folders and files will be pushed to your GitHub repository.

Note: After pasting the code, save your file. You can use any output method for this. If you edit your code, simply refresh your browser running through port number 5500. This way, there is no need to launch the application again and again.

Practice task

1. You can include prices for all the items in breakfastMenu, mainCourseMenu, and dessertMenu.
2. For this you can take include the prices as follows:

```
const breakfastMenu = ['Pancakes- $12', 'Eggs Benedict -$22.99', 'Oatmeal -$21.99', 'Frittata -$15'];
```

3. Then, access it in the similar manner and you will see the output for the menu items along with prices as well.

Restaurant Menu Iteration

Breakfast Menu

Item 1: Pancakes- \$12

Item 2: Eggs Benedict -\$22.99

Item 3: Oatmeal -\$21.99

Item 4: Frittata -\$15

Summary

- HTML structure:** Defines an HTML structure for a restaurant menu, categorizing items into Breakfast, Main Course, and Dessert sections using appropriate <h2> headings and <div> elements with unique IDs for content display.
- JavaScript dynamic content:** Utilizes JavaScript to dynamically populate menu items within their respective sections.
 - Breakfast menu: Uses the map method to generate HTML strings for breakfast items and inserts them into the designated <div> element.
 - Main course menu: Utilizes forEach to construct HTML strings for main course items, updating the related <div> with the generated content.
 - Dessert menu: Utilizes a for loop to create HTML strings for dessert items and inserts them into the designated <div> element for dessert items.
- Data handling:** Stores menu item data in separate JavaScript arrays (breakfastMenu, mainCourseMenu, dessertMenu) to facilitate dynamic content creation without hardcoding within the HTML structure.
- Dynamic rendering:** Illustrates how JavaScript interacts with HTML, dynamically generating and updating content within the webpage, demonstrating different iteration methods to present menu items under specific categories.

© IBM Corporation. All rights reserved.