Hands-on Lab: Writing Your First JavaScript Function

Estimated time needed: 30 minutes

What you will learn

In this lab, you will delve into foundational JavaScript concepts vital for web development. You will learn how to use functions to create an interactive web application that calculates the area of a rectangle based on user-provided values.

Learning objectives

After completing this lab, you will be able to:

- **Understanding HTML structure:** Recognize the structure of an HTML file and comprehend how to create an interactive user interface by implementing HTML elements like input fields and buttons.
- **JavaScript variable declaration:** Grasp the concept of declaring variables in JavaScript and understand their role in storing data retrieved from user input.
- **Retrieving user input with JavaScript:** Learn how to use JavaScript to fetch and process user-provided data entered into input fields within an HTML document.
- **Function implementation and execution:** Comprehend the creation and execution of JavaScript functions and their role in executing specific tasks, such as performing calculations and dynamically updating HTML content based on user actions.

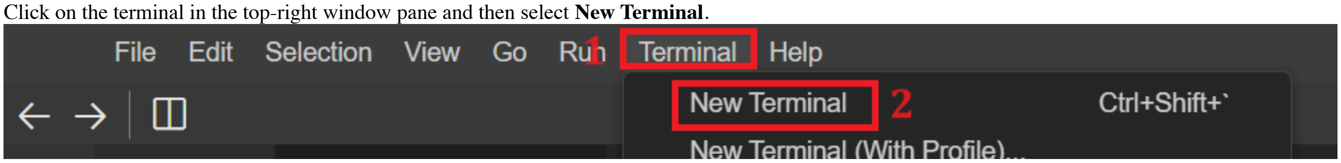
Prerequisites

- Basic Knowledge of HTML and Git commands.
- Basic understanding of functions and it's syntax.
- Web browser with a console (Chrome DevTools, Firefox Console, and so on).

Step 1: Setting up the environment

1. Firstly, you need to clone your main repository in the **Skills Network Environmemnt** which you created in the first lab and where you have been pushing all work related to the previous labs. Follow the given steps to clone this repository:

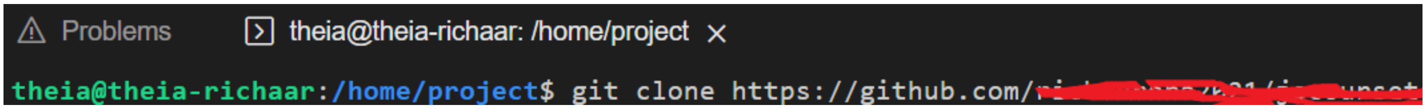
   - Click on the terminal in the top-right window pane and then select **New Terminal**.



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

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

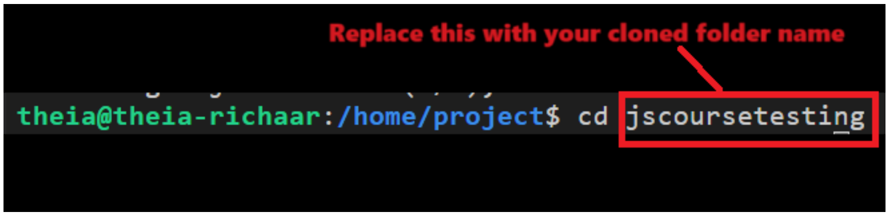
Note: Put your own GitHub repository link instead of `<github-repository-url>`.



```
theia@theia-richaar:/home/project$ git clone https://github.com/[redacted]
```

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

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



Replace this with your cloned folder name

```
theia@theia-richaar:/home/project$ cd jscoursetesting
```

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.

2. Now select **cloned Folder Name** folder, right click on it and click on **New Folder**. Enter folder name as **calculateArea**. It will create the folder for you. Then select **calculateArea** folder, right click and select **New File**. Enter file named as **calculate_Area.html** and click OK. It will create your HTML file.

3. Now, select **calculateArea** folder again, right click and select **New File**. Enter file named as **calculate_Area.js** and click OK to create your javaScript file.

4. Create the basic template structure for **calculate_Area.html** file by adding the provided content.

   - Inside the HTML file, create an input form to collect the length and width of the rectangle along with a button to trigger the calculation.

   - To achive this, include the provided code into the **calculate_Area.html** file.

     ```
     <html>
     <head>
       <title>Rectangle Area Calculator</title>
     </head>
     <body>
       <h1>Rectangle Area Calculator</h1>
       <label for="length">Enter the length: </label>
       <input type="number" id="length"><br><br>
       <label for="width">Enter the width: </label>
       <input type="number" id="width"><br><br>
       <button onclick="calculateArea()">Calculate Area</button><br><br>
       <p id="result"></p>
     <script src="./calculate_Area.js"></script>
     </body>
     </html>
     ```

5. The provided HTML code includes:

   - **Page title and heading:** Sets up a webpage titled "Rectangle Area Calculator" and presents a primary heading `<h1>` displaying the same title, ensuring clarity about the page's purpose.

   - **Input fields for user data:** Provides input fields labeled for length and width `<input type="number" id="length"> and <input type="number" id="width">` to allow users to input numerical values for the rectangle's dimensions.

   - **Calculation trigger and display:** Includes a button `<button onclick="calculateArea()">Calculate Area</button>` to execute a JavaScript function named calculateArea() upon clicking. The calculated result of the rectangle's area will be displayed within the `<p>` element with the ID 'result'.

   - **Dynamic result display:** Prepares a placeholder `<p id="result"></p>` to dynamically display the calculated area, creating a user-friendly interface for real-time feedback after the calculation is performed.

   - One `<script>` tag is added to include the js file in the **calculate_Area.html** file using the **src** attribute.

Step 2: Defining variables and function to calculate area

1. Declare two variables named **length** and **width** in **calculate_Area.js** file but do not assign any values to them yet. These variables will be used to store the length and width of the rectangle provided by the user through a form in HTML file.

   ```
   let length;
   let width;
   ```

2. Now create a function named as **calculateArea** in **calculate_Area.js** file as follows:

```
function calculateArea() {
}
```

3. Inside the above function, fetch values from user as input. For this you need to get values using `document.getElementById` from user input within **calculateArea** function as follows:

```
function calculateArea() {
 length = parseFloat(document.getElementById('length').value);
 width = parseFloat(document.getElementById('width').value);
}
```

4. Above code includes:

   - **document.getElementById ('length'):** This part of the code retrieves an HTML element by its ID, specifically searching for an element with the ID 'length'.
   - **.value:** After accessing the HTML element, .value is used to retrieve the value entered into the input field associated with that element. For instance, if a user enters '5' into the input field for length, .value retrieves the string '5'.
   - **parseFloat(…):** The parseFloat() function converts the string value retrieved from the input field to a floating-point number. This conversion ensures that the input, typically text entered by the user, is treated as a number and can be used in mathematical operations.
   - **length and width:** Finally, the obtained floating-point numbers (representing the length and width values entered by the user) are stored in the variables length and width, respectively. These variables will be utilized for further calculations, such as determining the area of a rectangle in this context.

5. Next, declare a variable named **area** and initialize it with **length * width** in **calculate_Area.js** file as following:

```
function calculateArea() {
 length = parseFloat(document.getElementById('length').value);
 width = parseFloat(document.getElementById('width').value);
let area = length * width;
}
```

6. After the calculation of the rectangle's area is completed and stored within the variable named area, the given code involves presenting or displaying this result to the user interface. Include given code within the function after calculation of area.

```
document.getElementById('result').innerText = `The area of the rectangle is: ${area}`;
```

7. Above code includes:

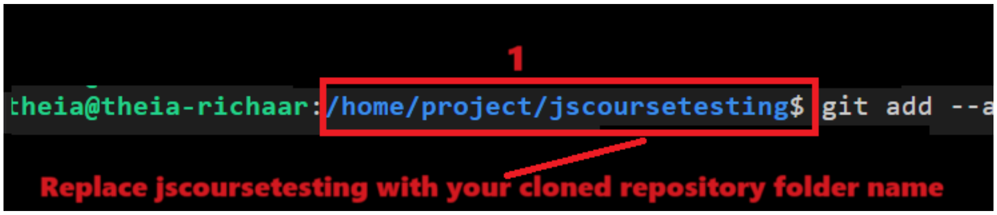
   - d**ocument.getElementById ('result'):** This part of the code retrieves an HTML element by its ID. Specifically, it targets an element with the ID 'result'.
   - **.innerText = The area of the rectangle is: ${area};:** Once the element is accessed, .innerText is used to modify the text content within that HTML element.
   - The backticks and ${} notation allow for the inclusion of JavaScript variables within a string (using template literals). In this case, it sets the text content to display a message along with the calculated area stored in the variable area. For example, if area holds a value of 25, the text displayed will be "The area of the rectangle is: 25".

Step 3: Perform Git commands

1. Perform `git add` to add latest files and folder in git environment.

```
git add --a
```

   - Make sure the terminal has the same path as follows:



```
theia@theia-richaar:/home/project/jscoursetesting$ git add --a
```

Replace jscoursetesting with your cloned repository folder name

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. 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"
```

Now after this you can perform git command as given below:

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

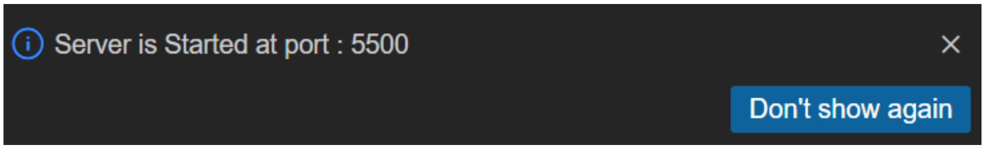
3. Then perform `git push` just by writing given command in terminal.

```
git push origin
```

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.

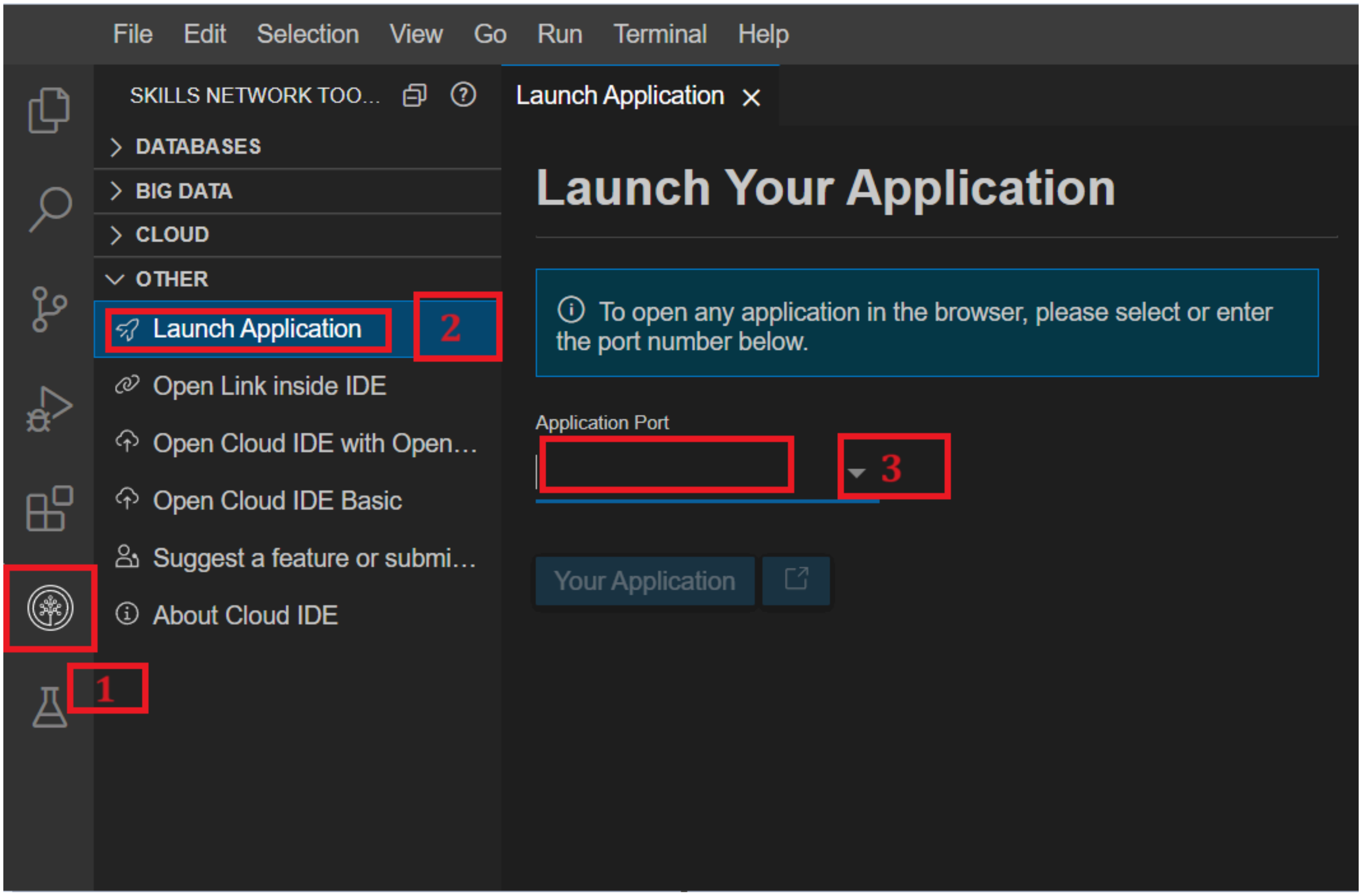
Step 4: Check the output

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



Server is Started at port : 5500

2. Click the Skills Network button on the left (refer to number 1). This action will open the "Skills Network Toolbox." Next, select "Launch Application" (refer to number 2). Once there, enter port number **5500** in "Application Port" (refer to number 3) and click this button .

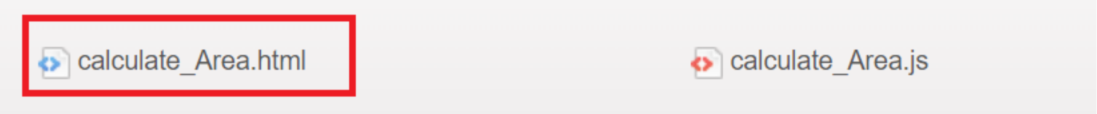


Launch Your Application

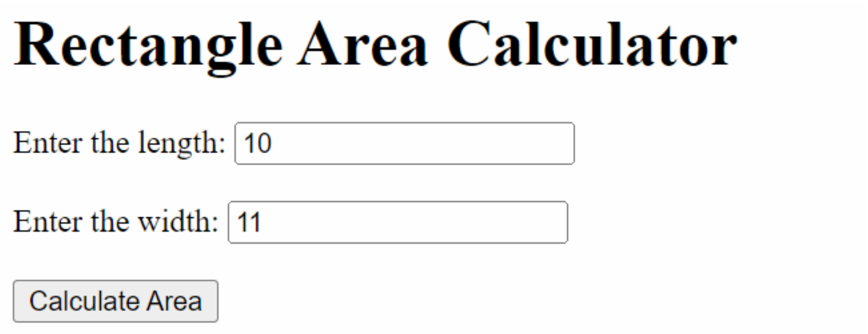
To open any application in the browser, please select or enter the port number below.

Application Port

3. It will open your default browser where you will see **cloned-folder-name** folder name. Click on that **cloned-folder-name** folder name. After clicking you will see multiple folders name, among those folders name click on **calculateArea** folder. You will see files related to this folder where again you will click on **calculate_Area.html** file as shown below.



3. It will open the HTML page where you can then enter the length and width values as shown below.

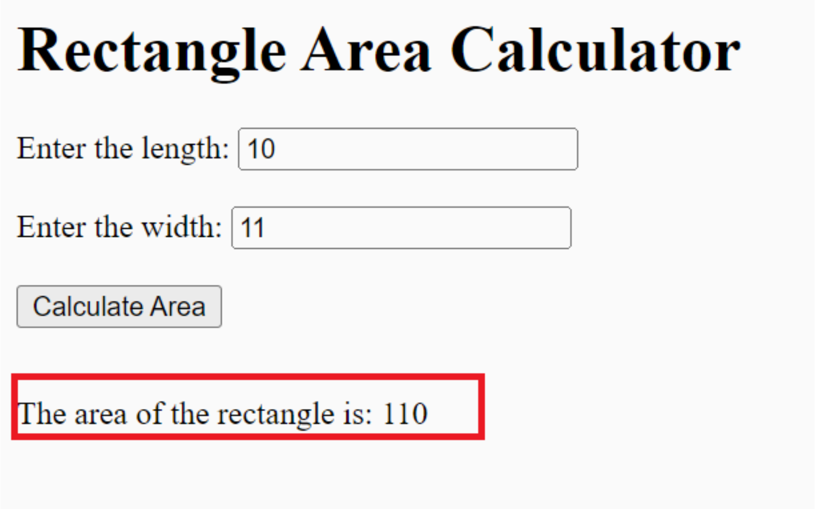


Rectangle Area Calculator

Enter the length: 10

Enter the width: 11

4. Then click on **Calculate Area** button and then you will see the answer.



Rectangle Area Calculator

Enter the length: 10

Enter the width: 11

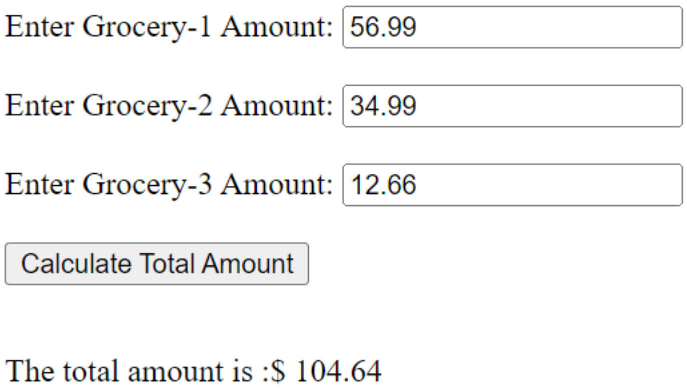
The area of the rectangle is: 110

Note: After pasting the code, remember to save your file. If you edit your code, then just refresh your browser, which is running through port number 5500. This way there is no need to launch the application again and again.

Practice task

In this task you need to create a function called **groceryTracker** to calculate the total amount of the purchased grocery item. For this:

1. Include the following in the HTML File:
   - You need to create at least three input boxes with ID named as "grocery1" and so on.
   - Also Label them using `<label>` as "Enter first grocery amount" and so on.
   - Create a button that calculates the total expenditure on the grocery purchases.
2. Include the following in the JavaScript file:
   - Create a function which will accept these amount entered by users as a parameter.
   - Then write the logic to calculate the total amount spent on the grocery purchase.
   - Call this function in such a way so that after clicking on the button, it shows the total amount for the grocery purchase.



Enter Grocery-1 Amount: 56.99

Enter Grocery-2 Amount: 34.99

Enter Grocery-3 Amount: 12.66

The total amount is :$ 104.64

Summary

1. **Setting up the environment:** Creating HTML and JavaScript files, initializing the basic HTML structure, input fields, and triggering calculation functionality.
2. **Defining variables and functions:** Declaring variables for user input storage and crafting a calculateArea() function to process this input. Retrieving user-entered values and computing the area of a rectangle based on these values.
3. **Displaying calculated result:** Dynamically updating the HTML content with the calculated area. Utilize JavaScript to modify specific elements within the HTML document, providing real-time feedback to the user.