



Full Stack Web Development Program

Activate Windows
Go to Settings to activate Windows.

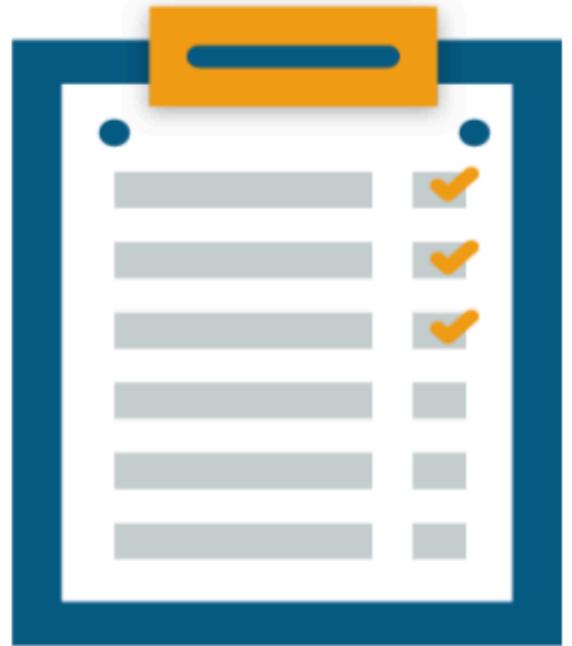


Day 19 - I/O in JavaScript

Activate Windows
Go to Settings to activate Windows.

Titles

- JavaScript Output Statements
 - innerHTML()
 - document.write()
 - windows.alert()
 - console.log()
- JavaScript Input Statements
 - windows.prompt()
 - windows.confirm()
- String Object
- String Methods
- Arrays
- Arrays Operations



Activate Windows
Go to Settings to activate Windows.

Learning Objectives

By the end of this module, you will be able to:

- Identify and explain the purpose of JavaScript input and output statements.
- Construct and manipulate arrays using array operations to store and manage collections of data.
- Develop JavaScript code that utilizes string objects and their associated methods to manipulate and process textual data.



Activate Windows
Go to Settings to activate Windows.

Input and Output Statements in JavaScript

Activate Windows
Go to Settings to activate Windows.

JavaScript Output Statements

- **JavaScript output statements** are essential for communicating with the user, debugging, and testing purposes.
- Understanding different output methods in JavaScript can greatly enhance the way you interact with your web pages and debug your code.

Statement	Description
innerHTML	It writes a data into a certain HTML element
document.write()	It displays a data in the HTML output
window.alert()	It displays the data in the alert box
console.log()	It displays the data in the browser console

Activate Windows
Go to Settings to activate Windows.

JavaScript Output Statements – innerHTML()

- **innerHTML** is a property of the **Document Object Model (DOM)** in JavaScript that is used to get or set the HTML content of an element.
- It is one of the most powerful and commonly used ways to manipulate the content of a webpage dynamically.
- **Syntax:** `element.innerHTML = newHtmlContent;`
Here, **element** is a DOM object and **newHtmlContent** is a string containing the new HTML content you want to insert.

Example:

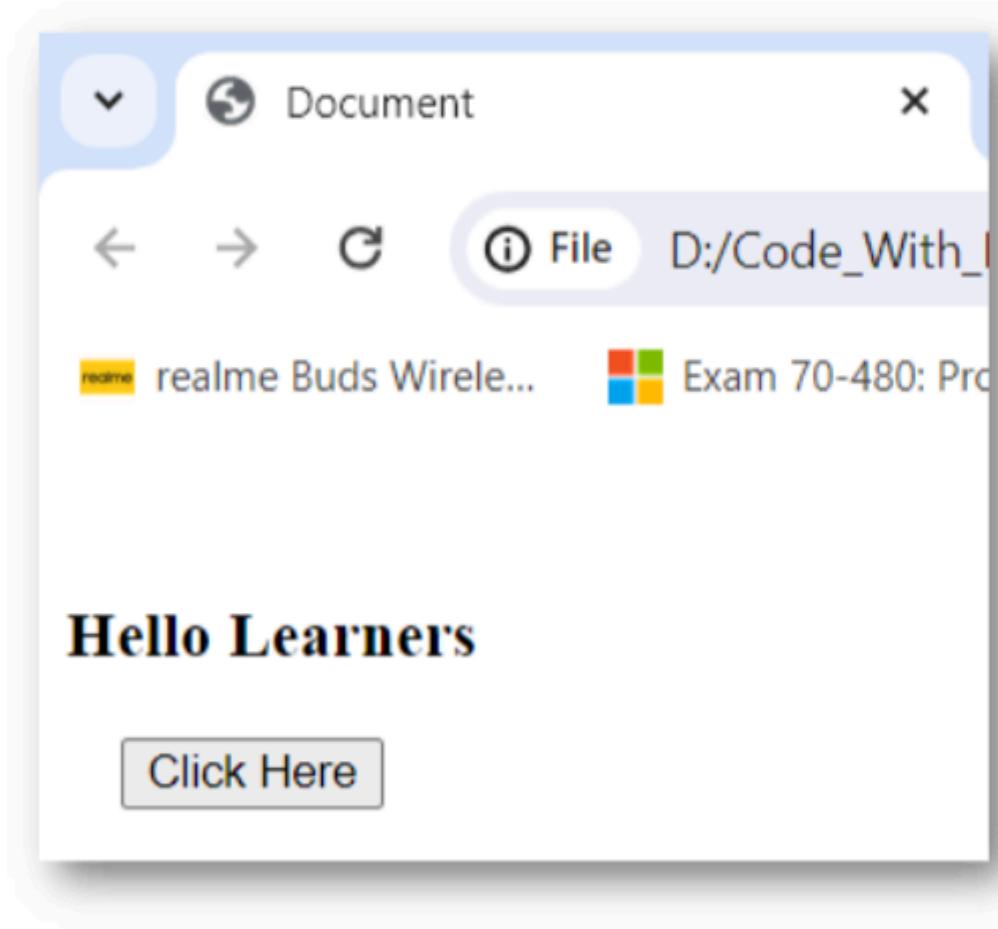
```
<body>
    <h3 id="myH"> Hello Learners </h3>
    <button onclick="myFunc()"> Click Here</button>
    <script>
        function myFunc() {
            var str = "Welcome To Edureka Program"
            const element= document.getElementById("myH");
            element.innerHTML = str;
        }
    </script>
</body>
```

The diagram illustrates the execution flow of the provided JavaScript code. It shows a callout box labeled "Identity of HTML element" pointing to the variable "element" in the code. Another callout box labeled "Insert the data" points to the assignment statement "element.innerHTML = str;" where the value "str" is being assigned to the innerHTML property of the identified element.

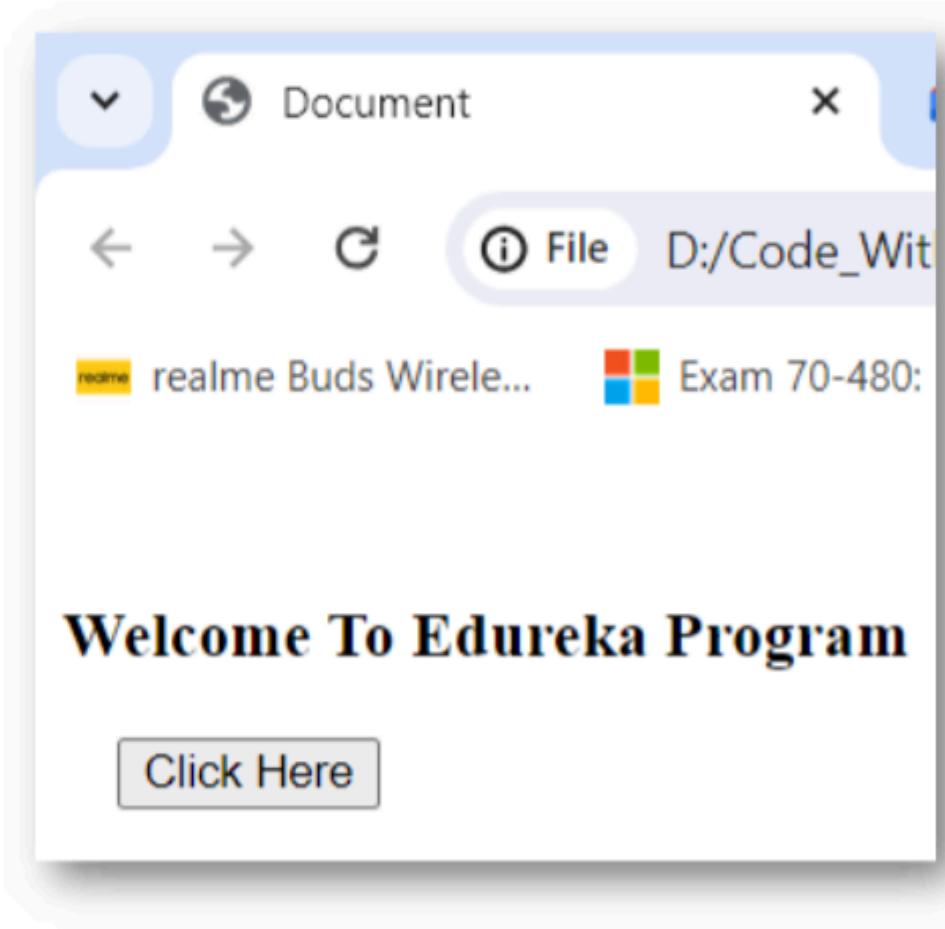
Activate Windows
Go to Settings to activate Windows.

JavaScript Output Statements – innerHTML()

Before clicking the button



After clicking the button



Hello Learners

Click Here

Welcome To Edureka Program

Click Here

After clicking the button, all the previous content of the HTML element will be removed

Activate Windows
Go to Settings to activate Windows.

JavaScript Output Statements – document.write()

- The **document.write()** method in JavaScript is used to write directly to the HTML document.
- This method has been a part of JavaScript since its early days and is primarily used for testing and simple HTML generation.
- Syntax:** document.write(markup)
- Here **document.write()** is a method call where document is a global object representing the HTML document, and **write** is the method. The **markup** parameter is a string that can contain HTML, JavaScript, or plain text intended to be written to the document.

Example:

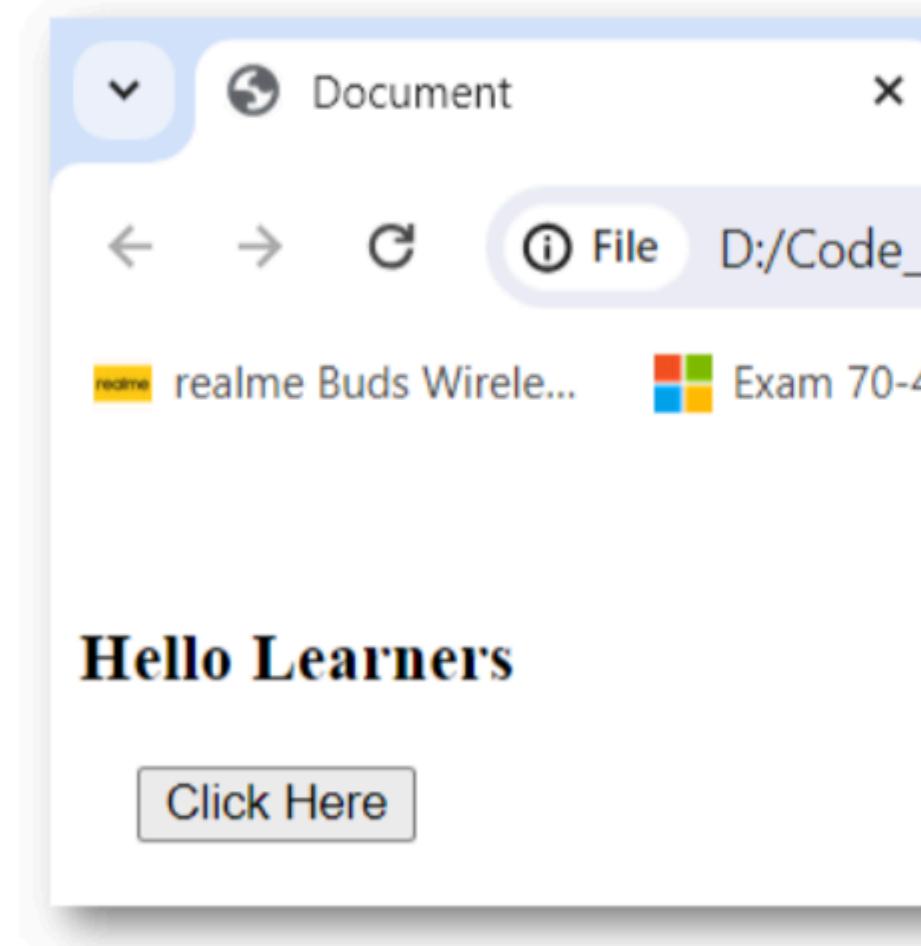
```
<body>
    <h3> Hello Learners </h3>
    <button onclick="myFunc()"> Click Here</button>
    <script>
        function myFunc() {
            var str = "Welcome To Edureka Program";
            document.write(str);
        }
    </script>
</body>
```

Add data but
will remove all
the previous
HTML

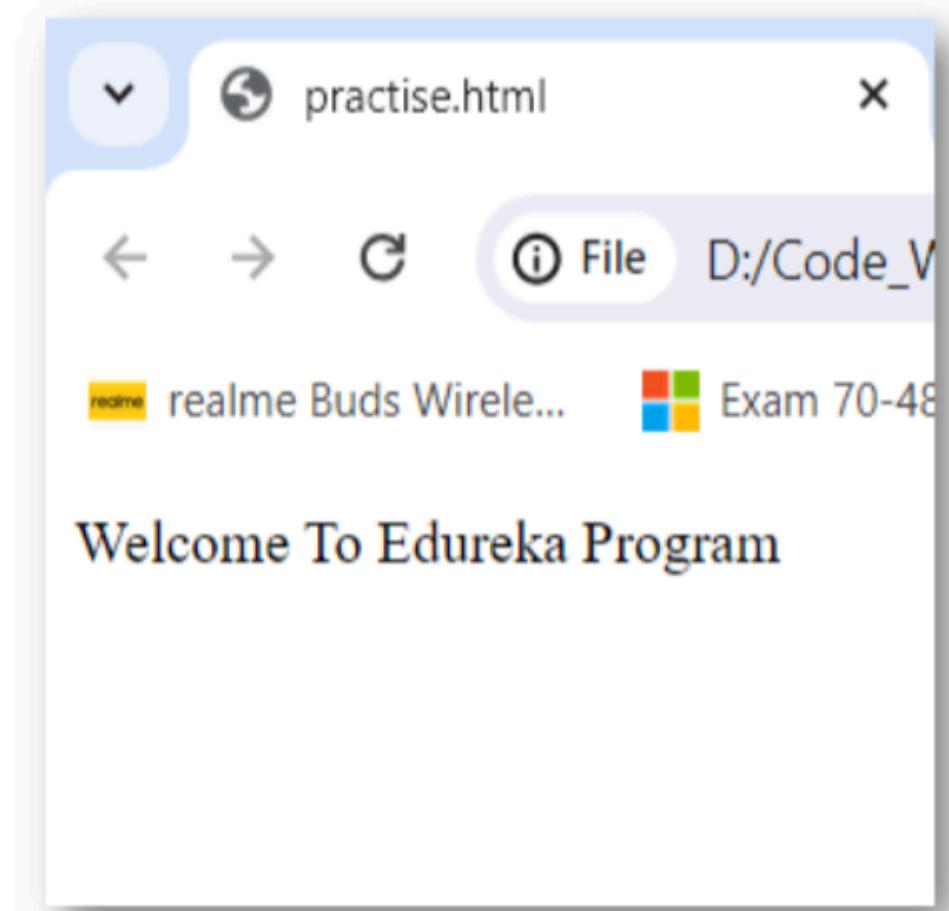
Activate Windows
Go to Settings to activate Windows.

JavaScript Output Statements – document.write()

Before clicking the button



After clicking the button



After clicking the button, all the previous content of the webpage will be removed

Activate Windows
Go to Settings to activate Windows.

JavaScript Output Statements – window.alert()

- The **window.alert()** function is a fundamental part of JavaScript, used to display alert messages to the user. It is a method of the window object, which represents the browser window.
- Syntax:** `window.alert(message);`
- It takes a single argument, **message**, which is the text to be displayed in the alert box.
- Note:** It does not return any value (undefined).

Example:

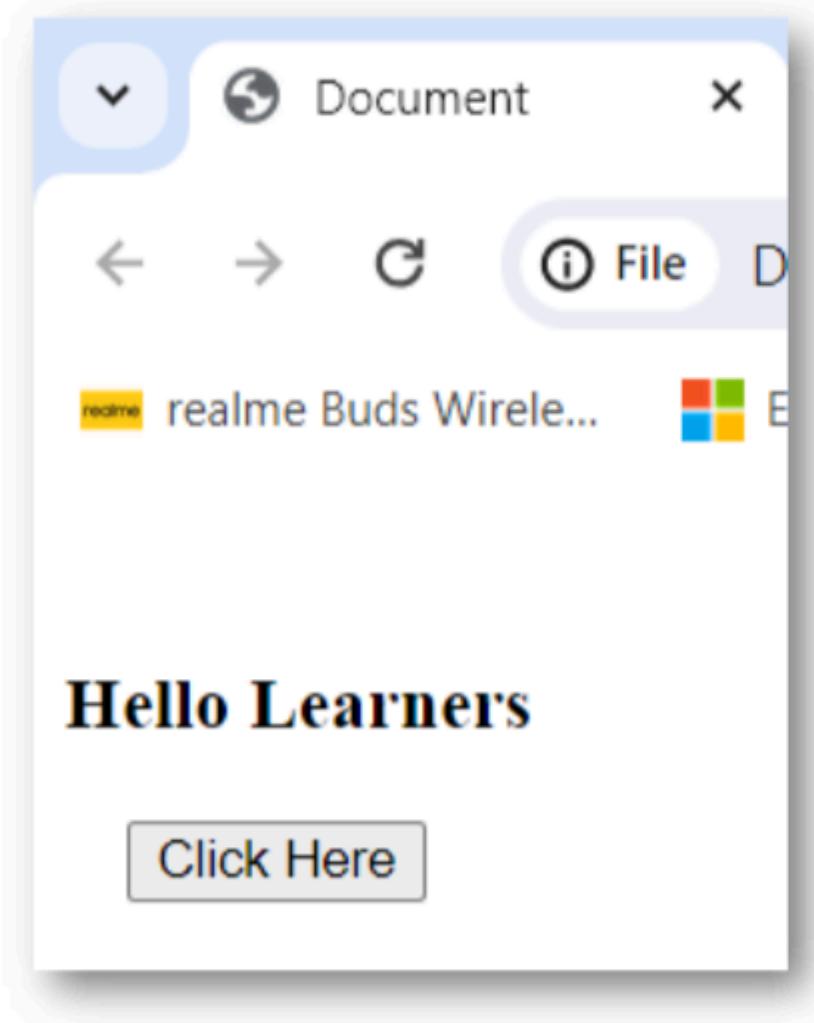
```
<body>
    <h3> Hello Learners </h3>
    <button onclick="myFunc()"> Click Here</button>
    <script>
        function myFunc() {
            var str = "Welcome To Edureka Program";
            window.alert(str);
        }
    </script>
</body>
```

Displays data in
a popup box

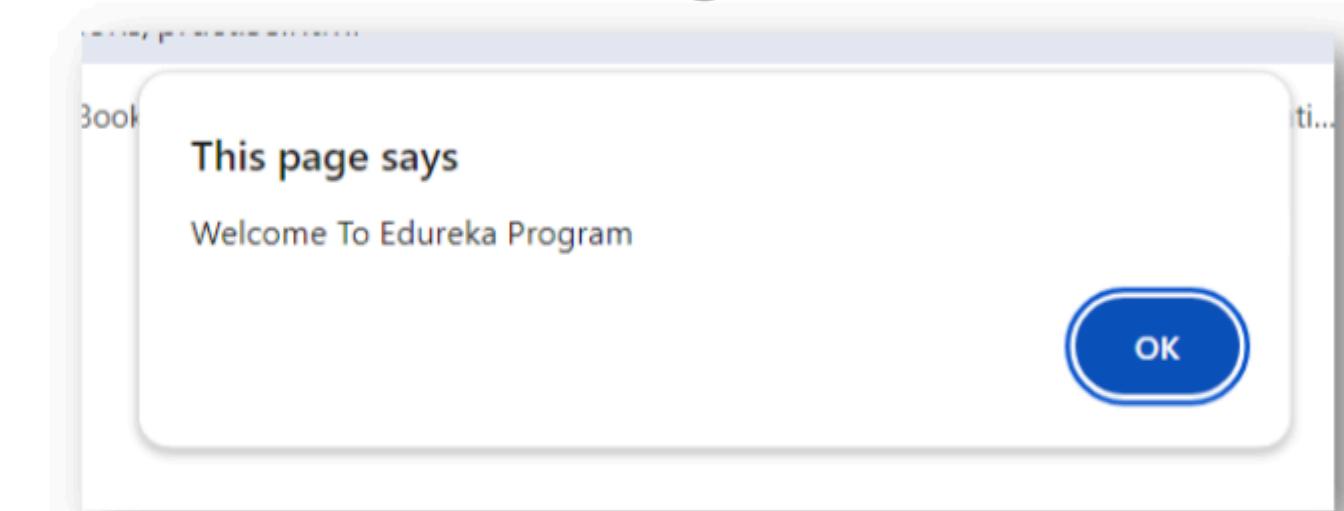
Activate Windows
Go to Settings to activate Windows.

JavaScript Output Statements – window.alert()

Before clicking the button



After clicking the button



The output is displayed by popping a window on the webpage.

Activate Windows
Go to Settings to activate Windows.

JavaScript Output Statements – `console.log()`

- **console.log()** is a JavaScript function used primarily for debugging purposes. It outputs information to the browser's console, making it an invaluable tool for developers to track variable values, program flow, and to print messages during script execution.
- **Syntax:** `console.log(message)`
- It takes a multiple, such as **console.log(message, variable1, variable2, ...)**

Example:

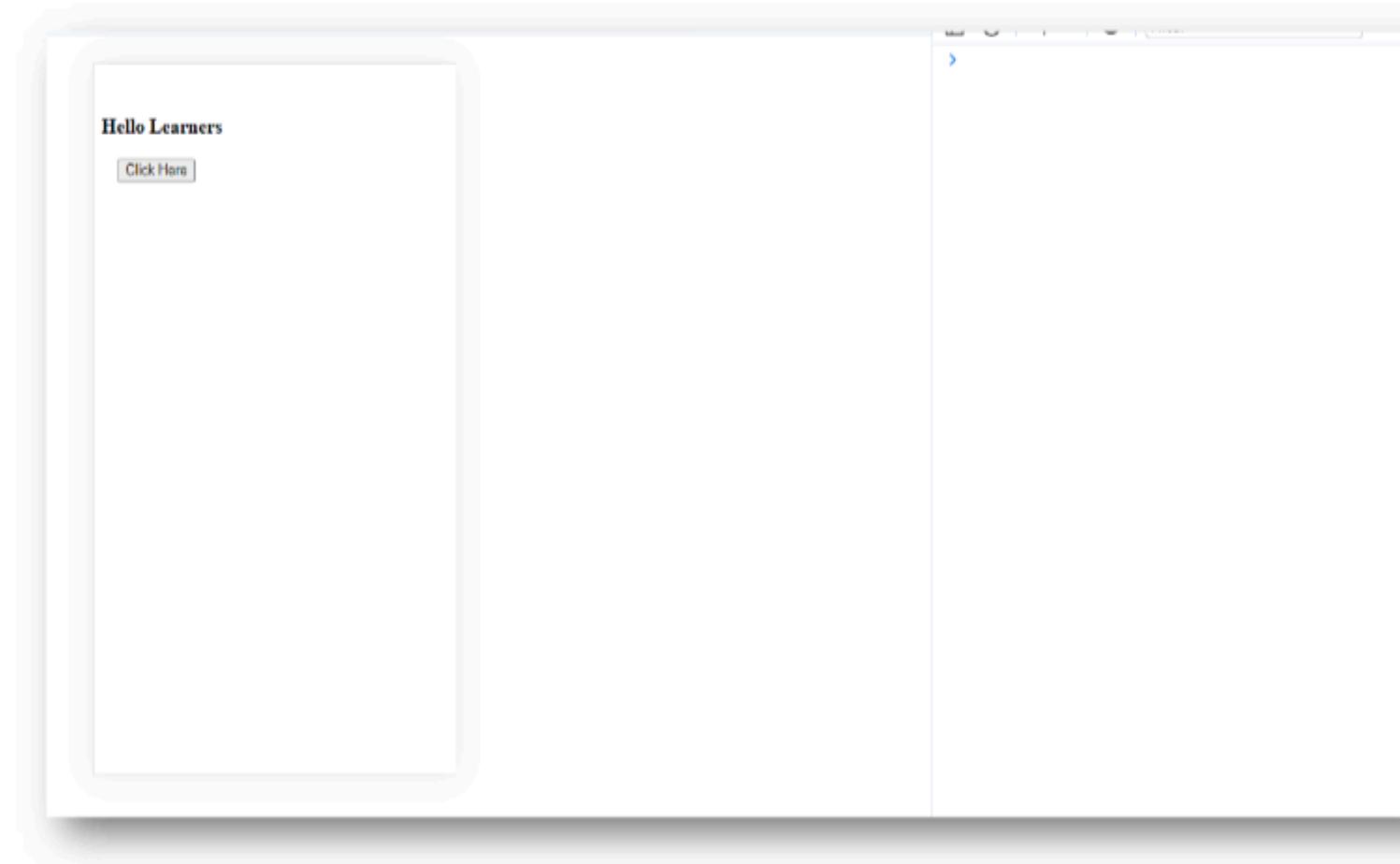
```
<body>
    <h3 id="myH"> Hello Learners </h3>
    <button onclick="myFunc()"> Click Here</button>
    <script>
        function myFunc() {
            var str = "Welcome To Edureka Program";
            console.log(str);
        }
    </script>
</body>
```

Displays data in
browser's console

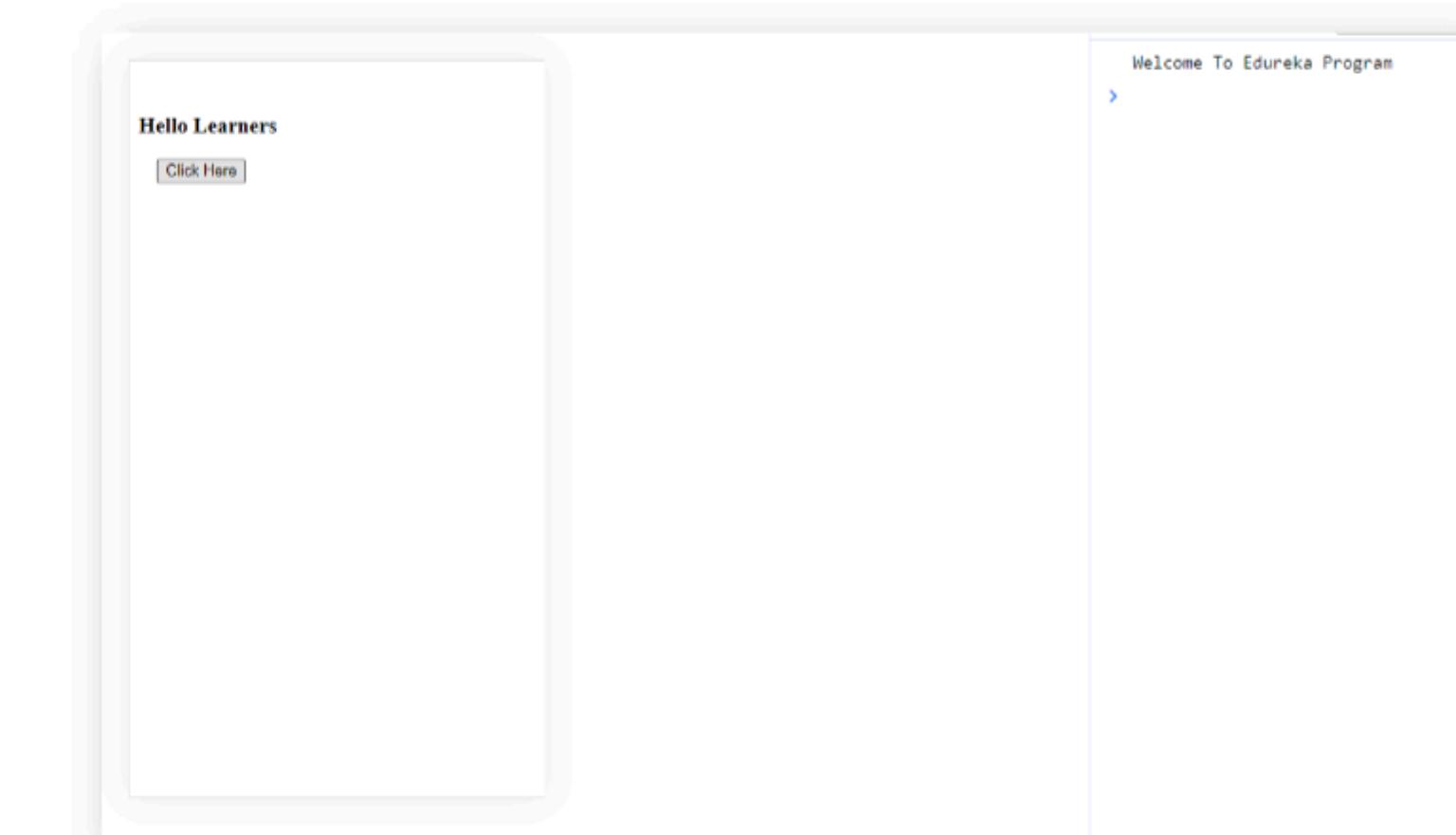
Activate Windows
Go to Settings to activate Windows.

JavaScript Output Statements – console.log()

Before clicking the button



After clicking the button

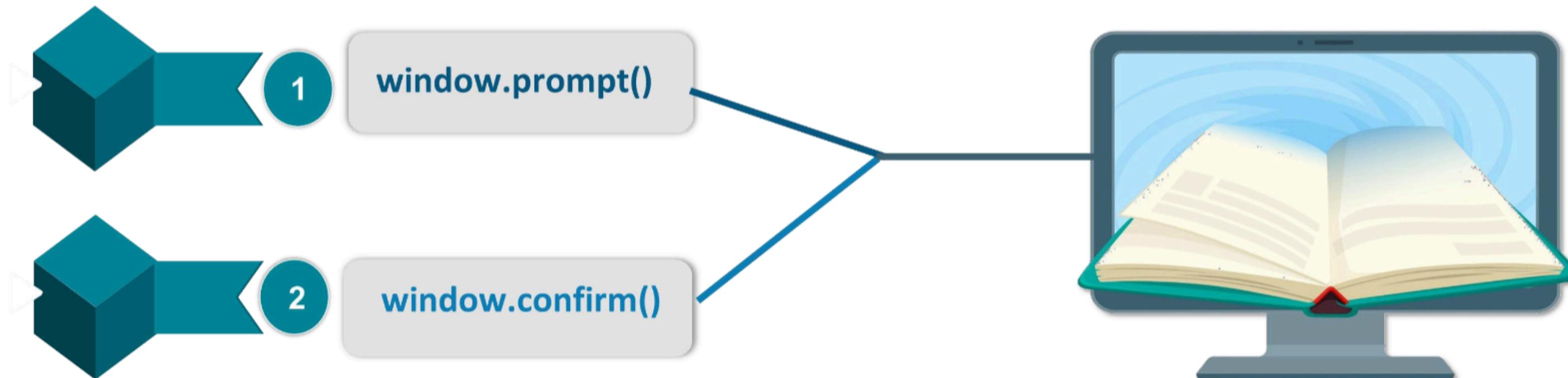


To open the console, right-click on your HTML page, click on the inspect and click on the console.

Activate Windows
Go to Settings to activate Windows.

JavaScript Input Statements

In JavaScript, receiving input from users is a critical aspect of creating interactive web applications. Unlike output methods, JavaScript doesn't have built-in input statements like some other languages. However, you can achieve user input through various methods:



Activate Windows
Go to Settings to activate Windows.

JavaScript Input Statements – `window.prompt()`

- `window.prompt()` displays a dialog box that prompts the user for input.
- It can display a message and an optional default input value.
- **Syntax:** `var userInput = window.prompt(message, defaultInput);`
- **message** denotes text to display in the dialog box, and **defaultInput** is an optional parameter and used to set a default value for the input field.

Example:

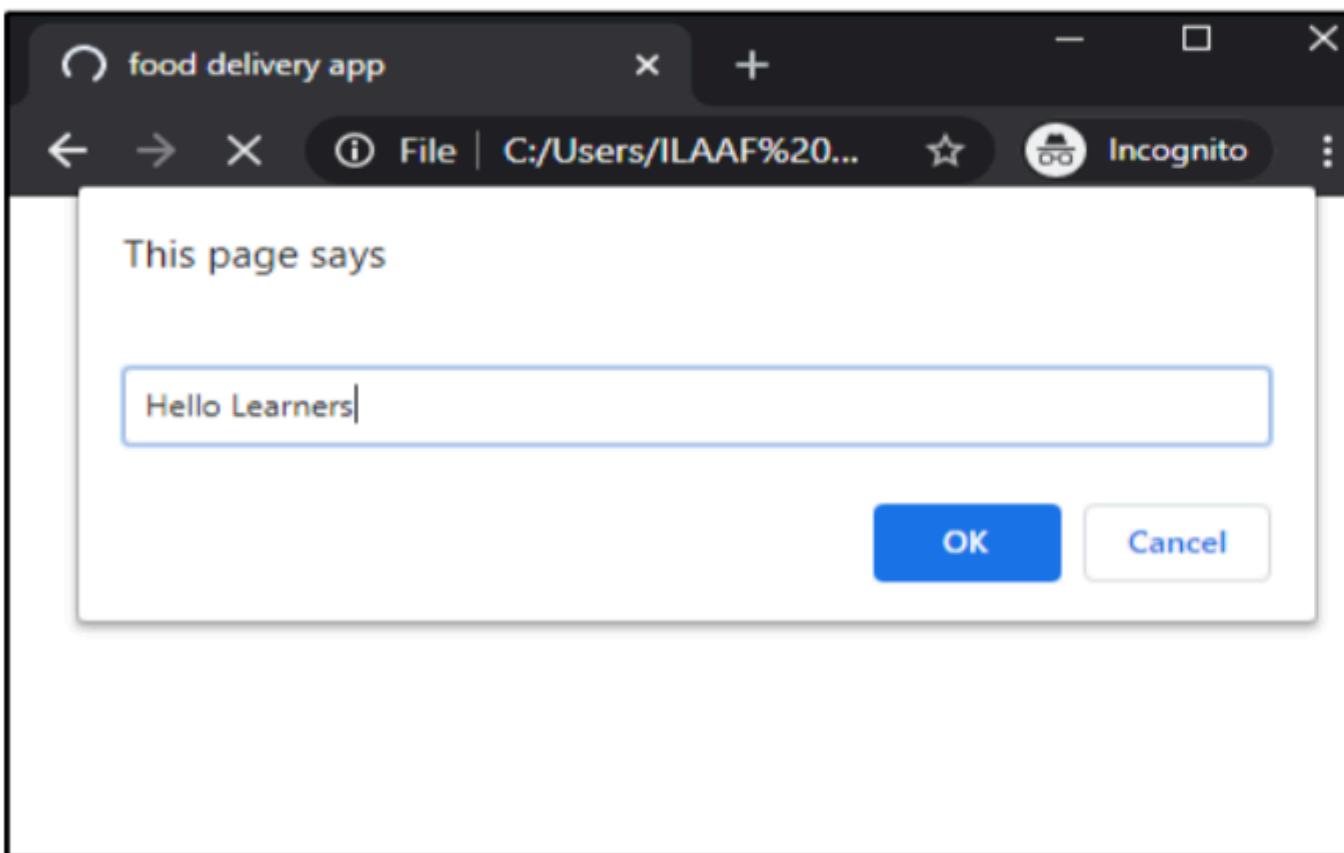
```
<body>
    <p>Enter Something</p>
    <h3 id="myHeader"> </h3>
    <script>
        var x = window.prompt("Enter value");
        element = document.getElementById("myHeader");
        element.innerHTML = x;
    </script>
</body>
```

Popups a box to take input.

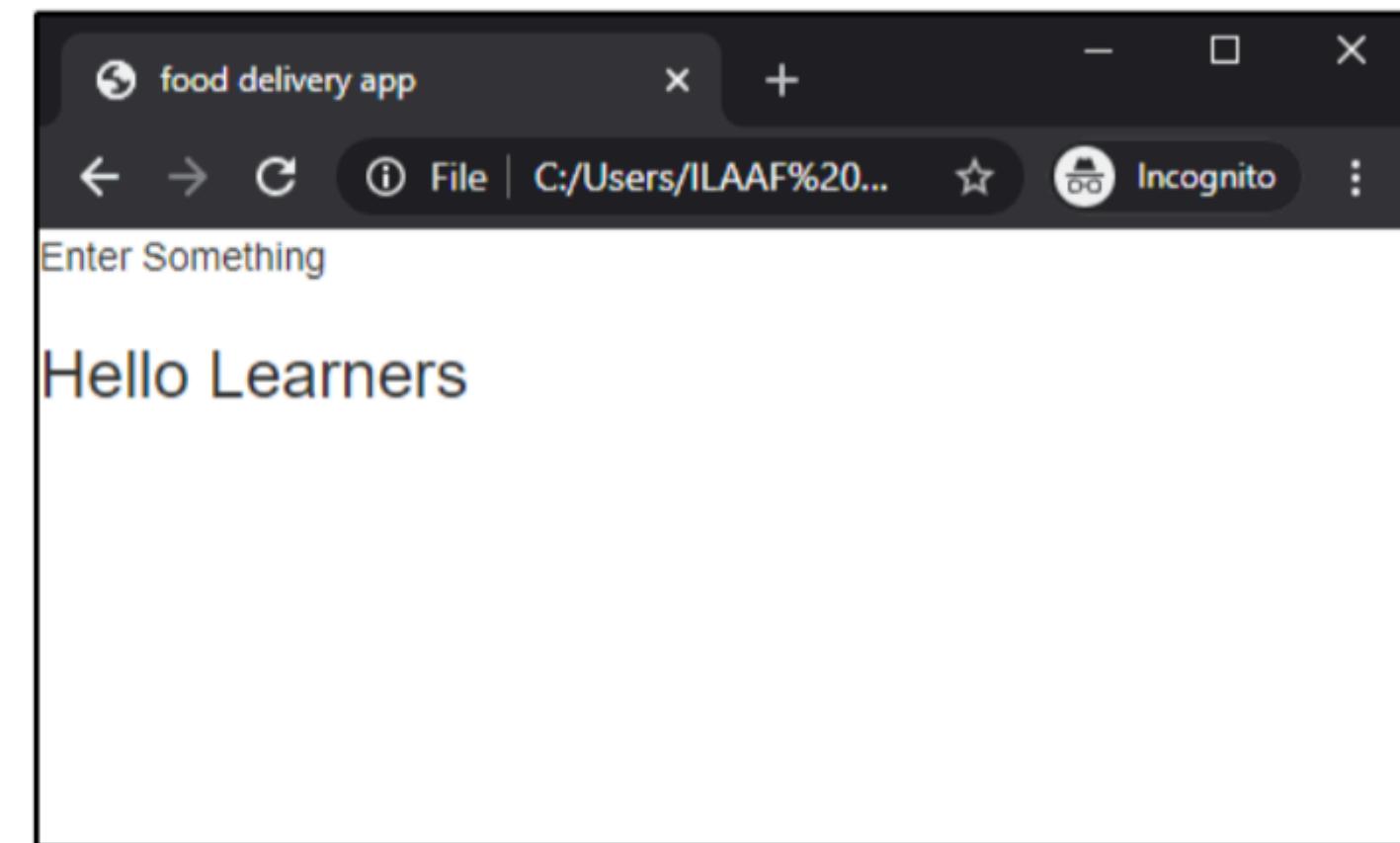
Activate Windows
Go to Settings to activate Windows.

JavaScript Input Statements – `window.prompt()`

Before clicking the ok button



After clicking the ok button



Activate Windows
Go to Settings to activate Windows.

JavaScript Input Statements – `window.confirm()`

- `window.confirm()` displays a dialog box with a specified message, along with "OK" and "Cancel" buttons.
- It's used to verify or accept something from the user.
- **Syntax:** `var userChoice = window.confirm(message);`
- **message** denotes text to display in the dialog box.

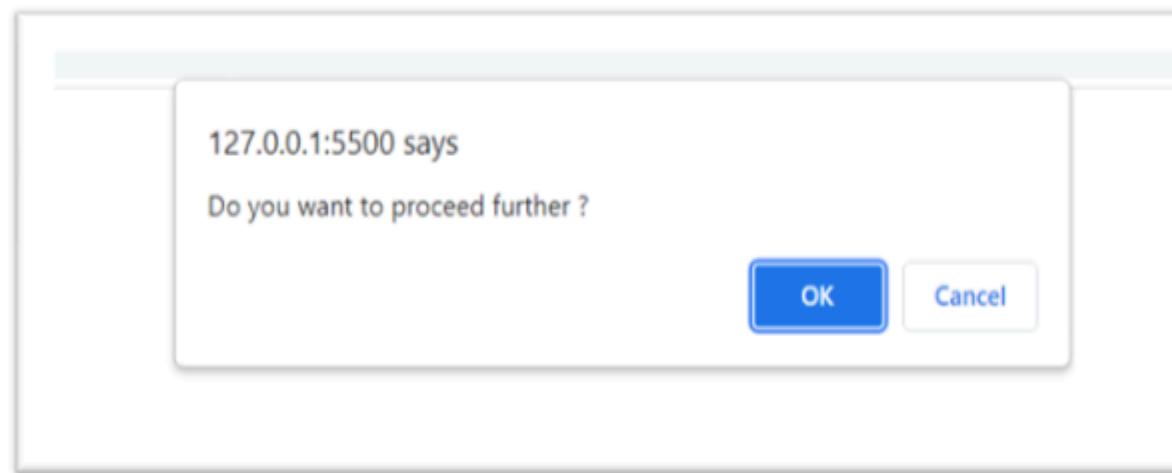
Example:

```
<body>
    <centre><h1 id="myHeader"> </h1></centre>
    <script>
        element = document.getElementById("myHeader");
        if (window.confirm("Do you want to proceed further ? ","OK","Cancel") == true)
            element.innerHTML = "Welcome to Edureka";
        else
            element.innerHTML = "Please come back later"
    </script>
</body>
```

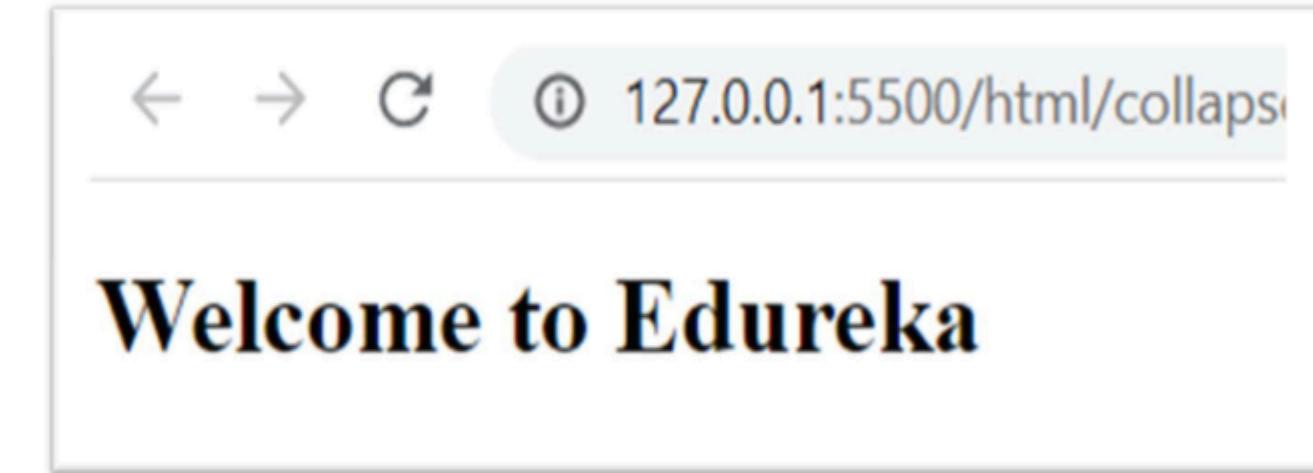
Activate Windows
Go to Settings to activate Windows.

JavaScript Input Statements – `window.confirm()`

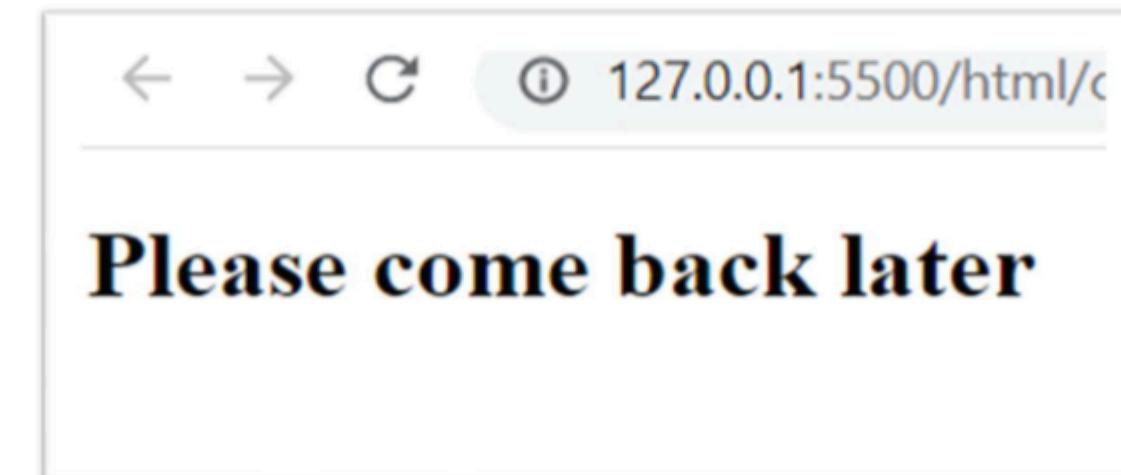
Before clicking any button



After clicking on **OK** button



After clicking on **Cancel** button



Activate Windows
Go to Settings to activate Windows.