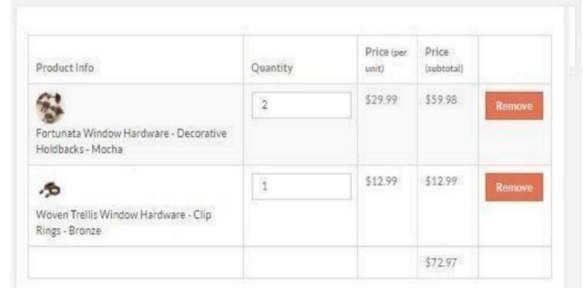
**Module (JavaScript Essentials And Advanced) – 7**

**MODULE: 1 (JavaScript Essentials)**

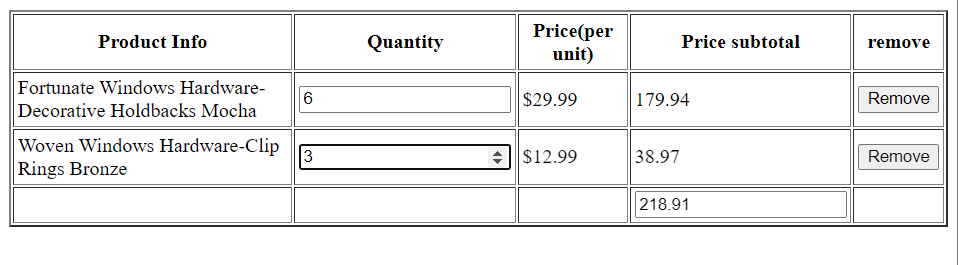
\* **Calculate subtotal price of quantity in JavaScript?**

****

**Answer:**

****

**Output:**

****

\* **What is JavaScript Output method?**

JavaScript Output defines the ways to display the output of a given code.

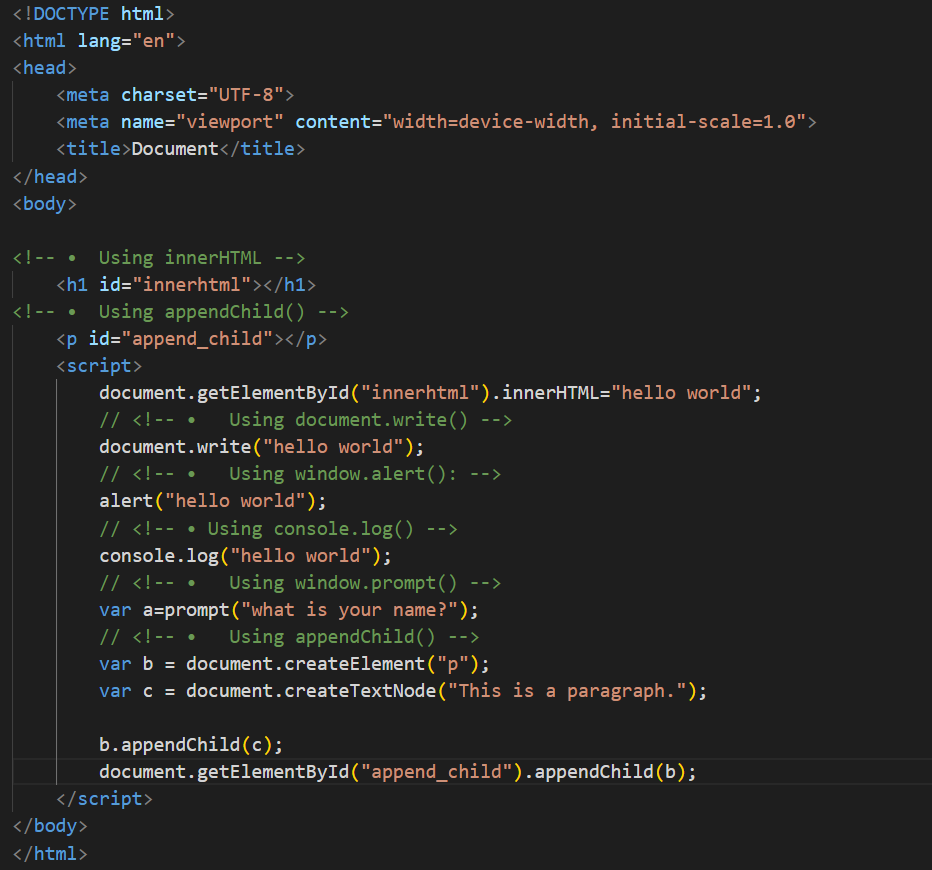
The output can be displayed by using different approaches which are listed below:

**Table of Content**

* [Using innerHTML](https://www.geeksforgeeks.org/javascript-output/#approach-1-using-innerhtml)
* [Using document.write()](https://www.geeksforgeeks.org/javascript-output/#approach-2-using-documentwrite)
* [Using window.alert():](https://www.geeksforgeeks.org/javascript-output/#approach-3-using-windowalert)
* [Using console.log()](https://www.geeksforgeeks.org/javascript-output/#approach-4-using-consolelog)
* [Using window.prompt()](https://www.geeksforgeeks.org/javascript-output/#approach-5-using-windowprompt)
* [Using appendChild()](https://www.geeksforgeeks.org/javascript-output/#approach-6-using-appendchild)

\* **How to used JavaScript Output method?**

**Answer:**

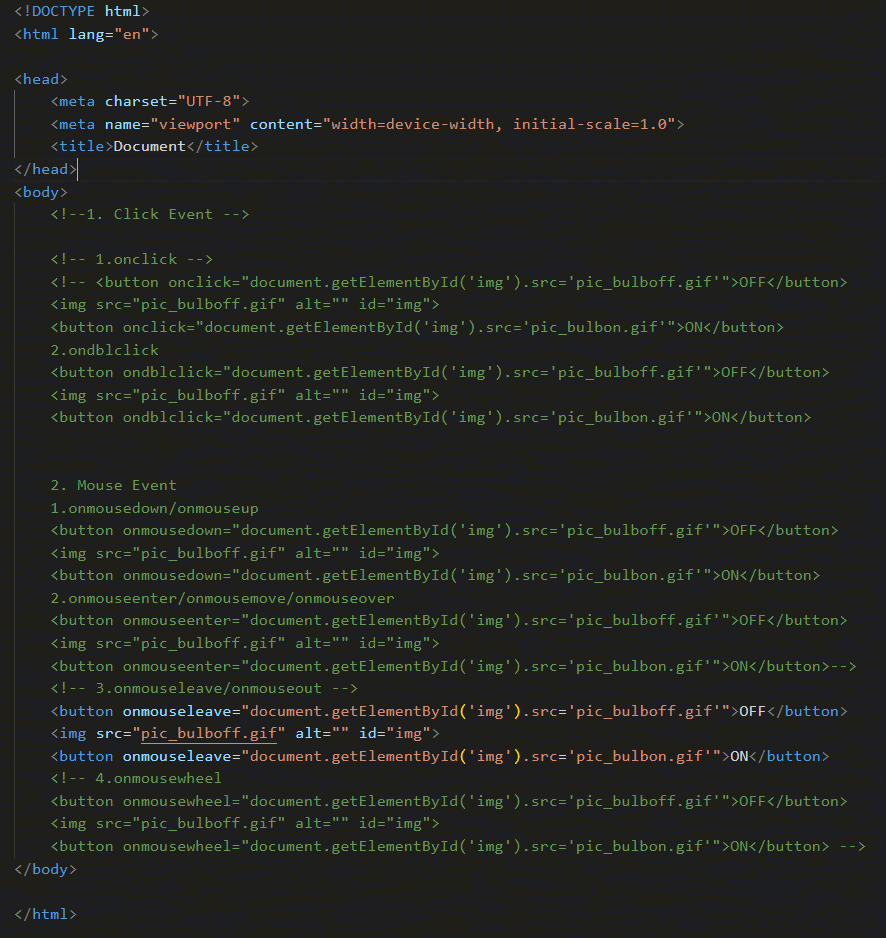
****

**Output:**

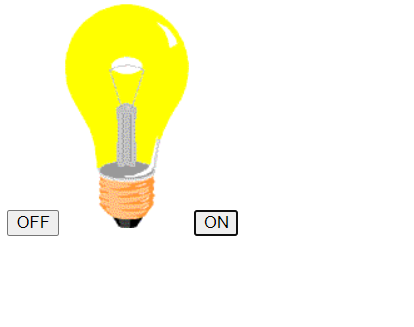
****

\* **How to used JavaScript Events to do all examples?**

**Answer:**



**Output:**



**Module :2(JAVASCRIPT BASIC & DOM)**

\* **What is JavaScript?**

- JavaScript is a high-level, interpreted programming language developed by Netscape.

\* **What is the use of isNaN function?**

- The **isNaN** function in JavaScript is used to determine whether a value is NaN (Not-a-Number) or not. It returns true if the value is NaN; otherwise, it returns false.

\* **What is negative Infinity?**

- Negative Infinity is a special value in JavaScript representing a mathematical concept of negative infinity. It is the result of mathematical operations that lead to a value beyond the representable range.

\* **Which company developed JavaScript?**

- JavaScript was developed by Netscape Communications Corporation.

\* **What are undeclared and undefined variables?**

- An undeclared variable is a variable that has been used in the code without being declared using the **var**, **let**, or **const** keyword. An undefined variable is a variable that has been declared but not assigned a value, or a variable accessed before any value is assigned.

\* **Write the code for adding new elements dynamically?**



**Output:**



\* **What is the difference between ViewState and SessionState?**

- ViewState is used to store state information of individual controls on a page, and it is automatically maintained by ASP.NET. SessionState, on the other hand, is used to store session-specific information that can be accessed across multiple pages during a user's session.

\* **What is === operator?**

- The **===** operator in JavaScript is the strict equality operator. It compares both value and type of the operands and returns **true** if they are equal, and **false** otherwise.

\***How can the style/class of an element be changed?**

- To change the style of an element, you can use the **style** property, and to change the class, you can use the **classList** property.

\* **How to read and write a file using JavaScript?**

- In a web browser environment, JavaScript does not have direct access to the file system for security reasons. However, you can read and write files in a server-side environment using technologies like Node.js. In the browser, you can interact with files through user input (e.g., file input) and use APIs like FileReader for reading.

\* **What are all the looping structures in JavaScript?**

- JavaScript looping structures include the traditional `for` loop, which iterates based on an initialization, condition, and increment/decrement. The `while` loop repeats a block of code while a specified condition holds true. The `do-while` loop is similar but ensures at least one execution. Additionally, `for...in` iterates over an object's properties, while `for...of` iterates over iterable objects like arrays.

\* **How can you convert the string of any base to an integer in JavaScript?**

- You can use the **parseInt** function to convert a string of any base to an integer by providing the base as the second argument.

\* **What is the function of the delete operator?**

- The **delete** operator in JavaScript is used to delete a property from an object or an element from an array. It does not delete variables or functions.

\* **What are all the types of Pop up boxes available in JavaScript?**

- JavaScript provides three types of pop-up boxes:

* **alert**: Displays a message box with an OK button.
* **confirm**: Displays a message box with OK and Cancel buttons.
* **prompt**: Displays a message box with a text input field, OK, and Cancel buttons.

\* **What is the use of Void (0)?**

- **void(0)** is often used in JavaScript to create a hyperlink that doesn't perform any action when clicked. It is commonly used in the **href** attribute to prevent the page from navigating to a new URL.

\* **How can a page be forced to load another page in JavaScript?**

- You can use the **window.location** object to navigate to a different page in JavaScript.

\* **What are the disadvantages of using innerHTML in JavaScript?**

- Using **innerHTML** to manipulate the content of an element has some disadvantages:

* Security: It can expose your application to cross-site scripting (XSS) attacks if the content is not properly sanitized.
* Performance: Manipulating **innerHTML** may be less efficient than using DOM methods for certain operations.
* Overwrites Events: Setting **innerHTML** can remove existing event listeners attached to child elements.

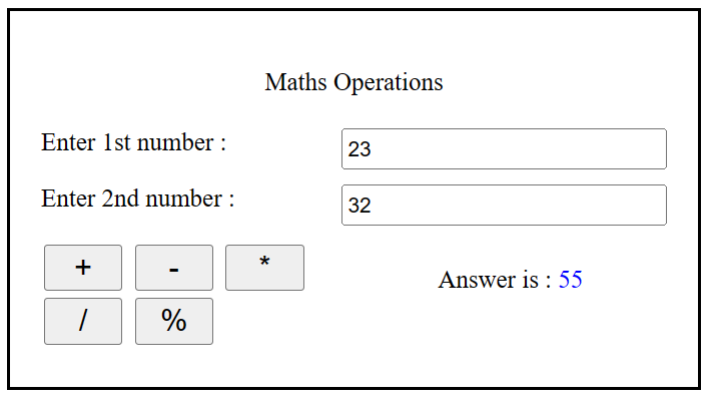
**\*Create password field with show hide functionalities**

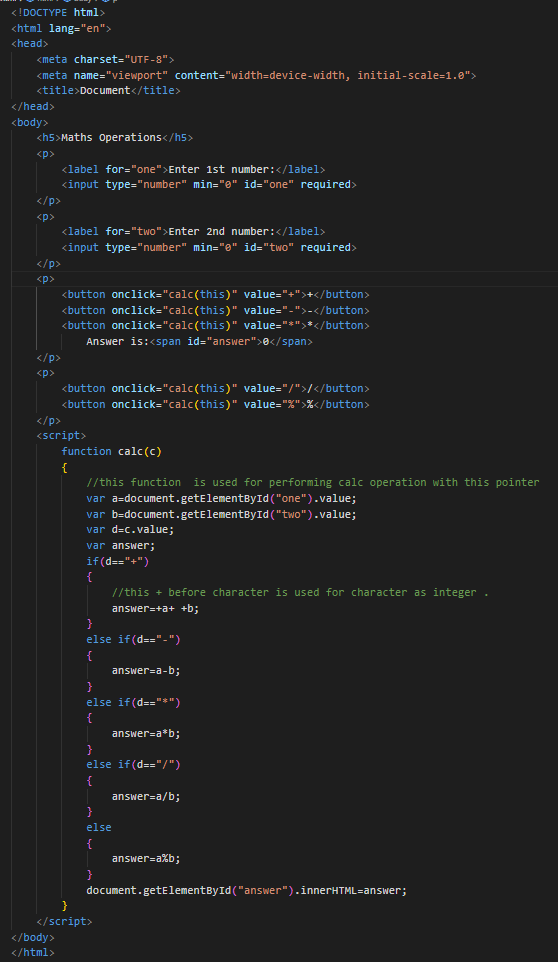
****

**Output:**

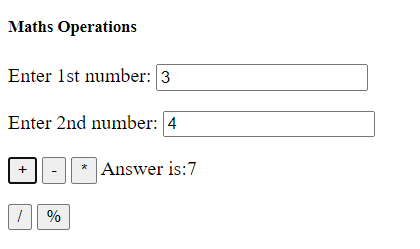


**\*Create basic math operation in JS**

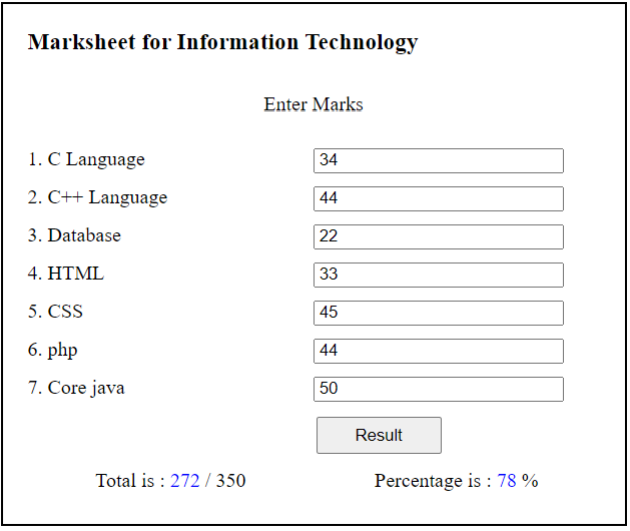


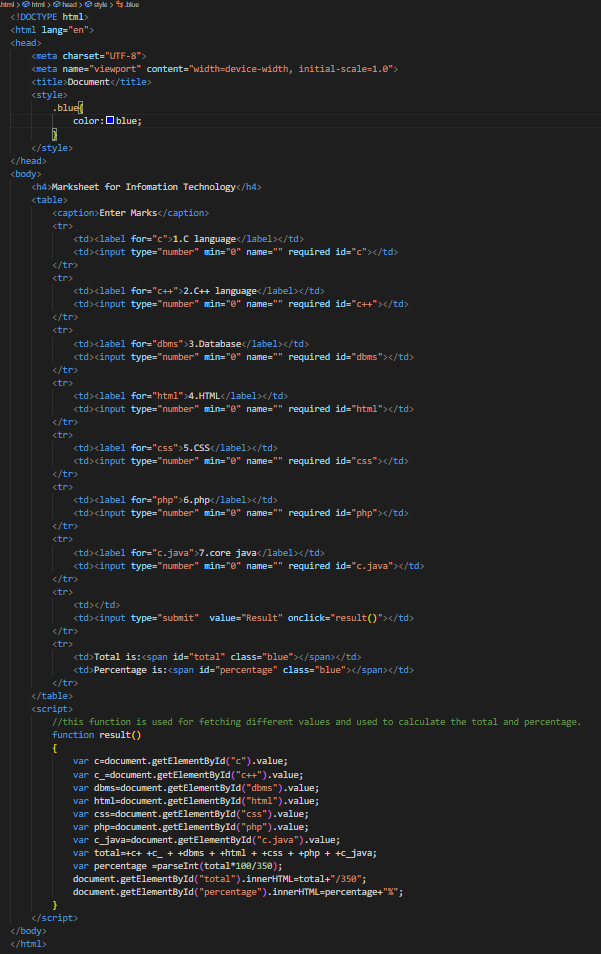


**Output:**

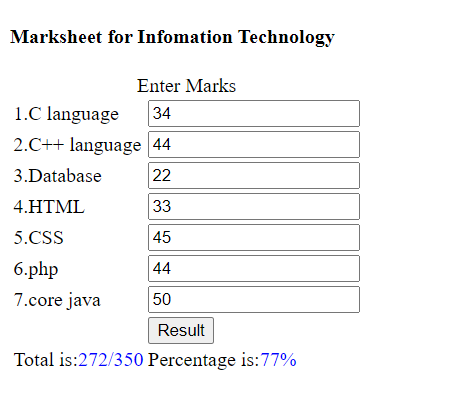


**\*Create result**

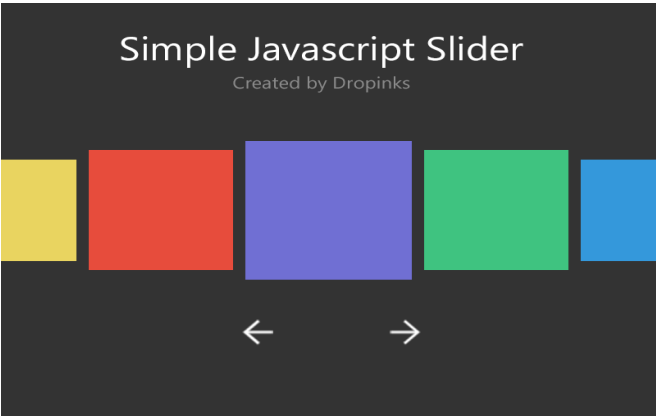
****

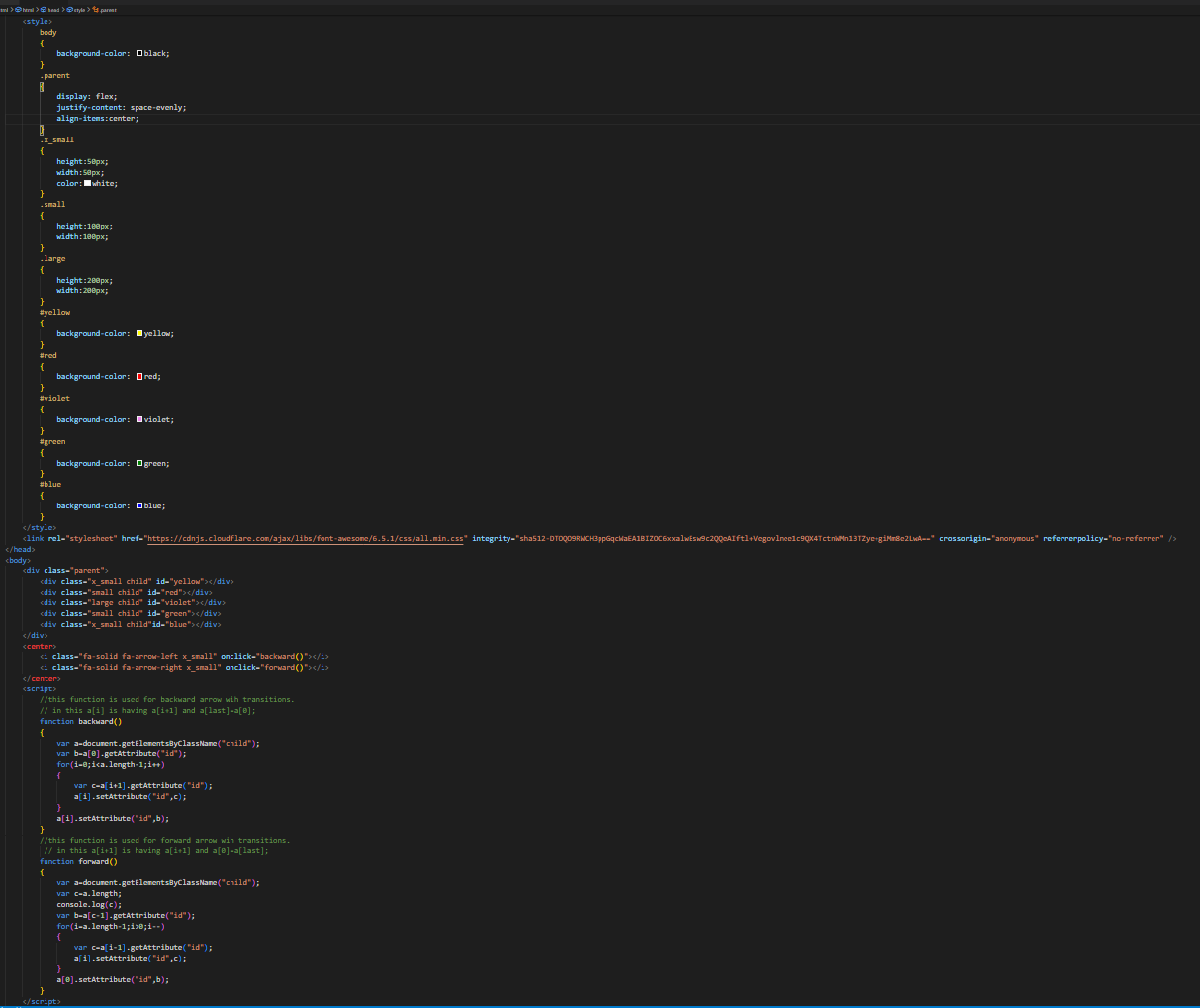
****

**Output**

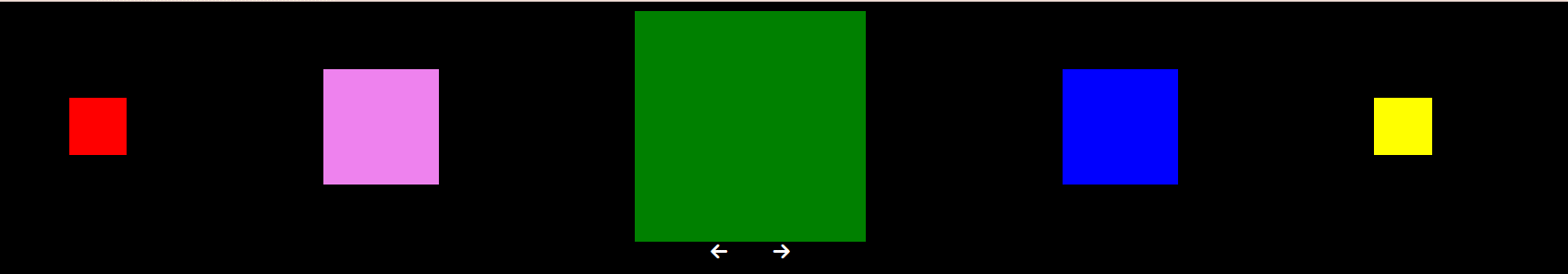


**\*Create a slider using JavaScript**

****

****

**Output:**



**MODULE: 3 (Introduction and Code Quality)**

\* **Write a program to Show an alert**

- 

    -----

\* **What will be the result for these expressions?**

1. 5 > 4

2. "apple" > "pineapple"

3. "2" > "12"

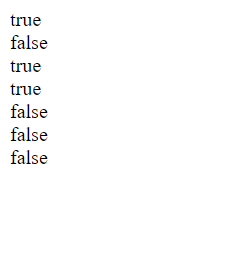
4. undefined == null

5. undefined === null

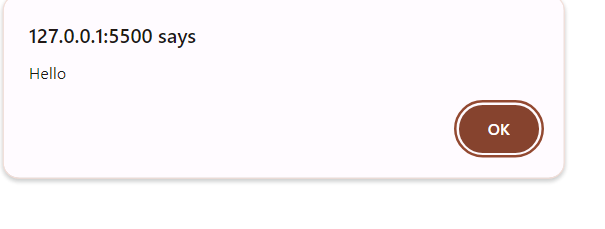
6. null == "\n0\n"

7. null === +"\n0\n"

Answer---

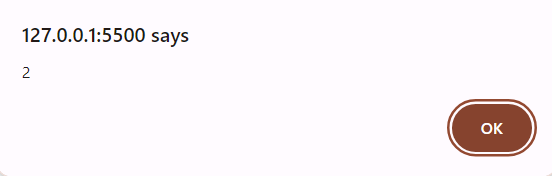


    ----

\* **Will alert be shown? if ("0") { alert( 'Hello');  }**

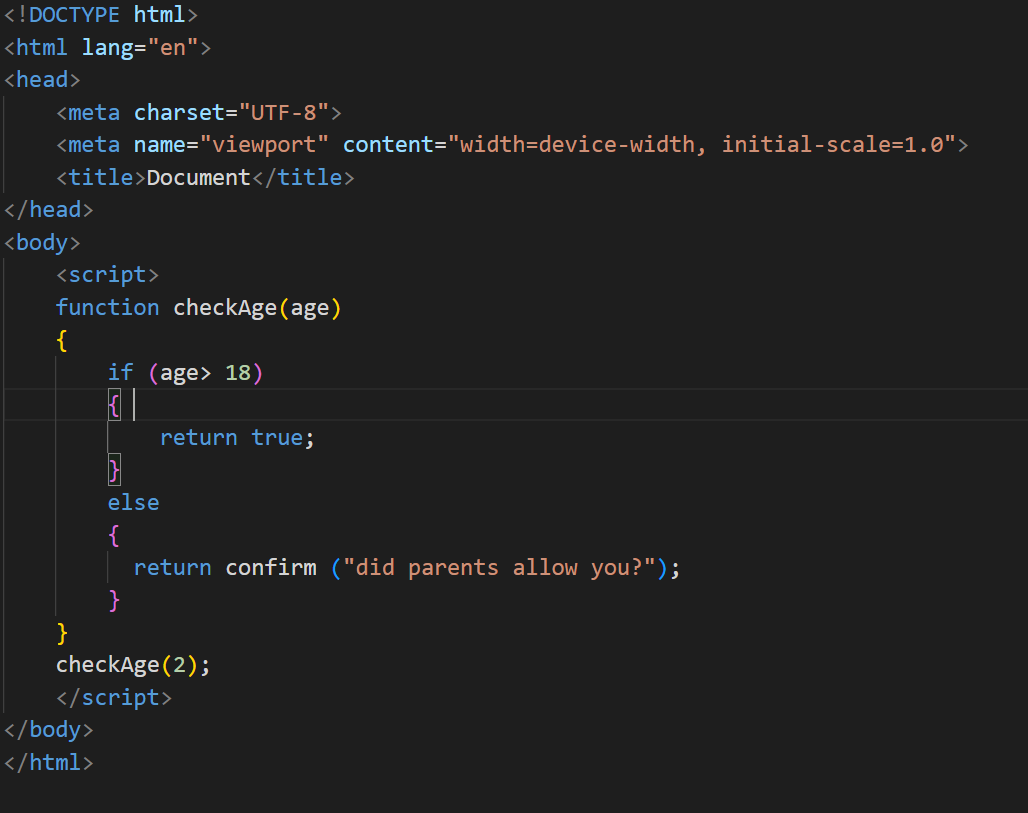
    ----

\* **What is the code below going to output? alert( null || 2 || undefined );**



    ----

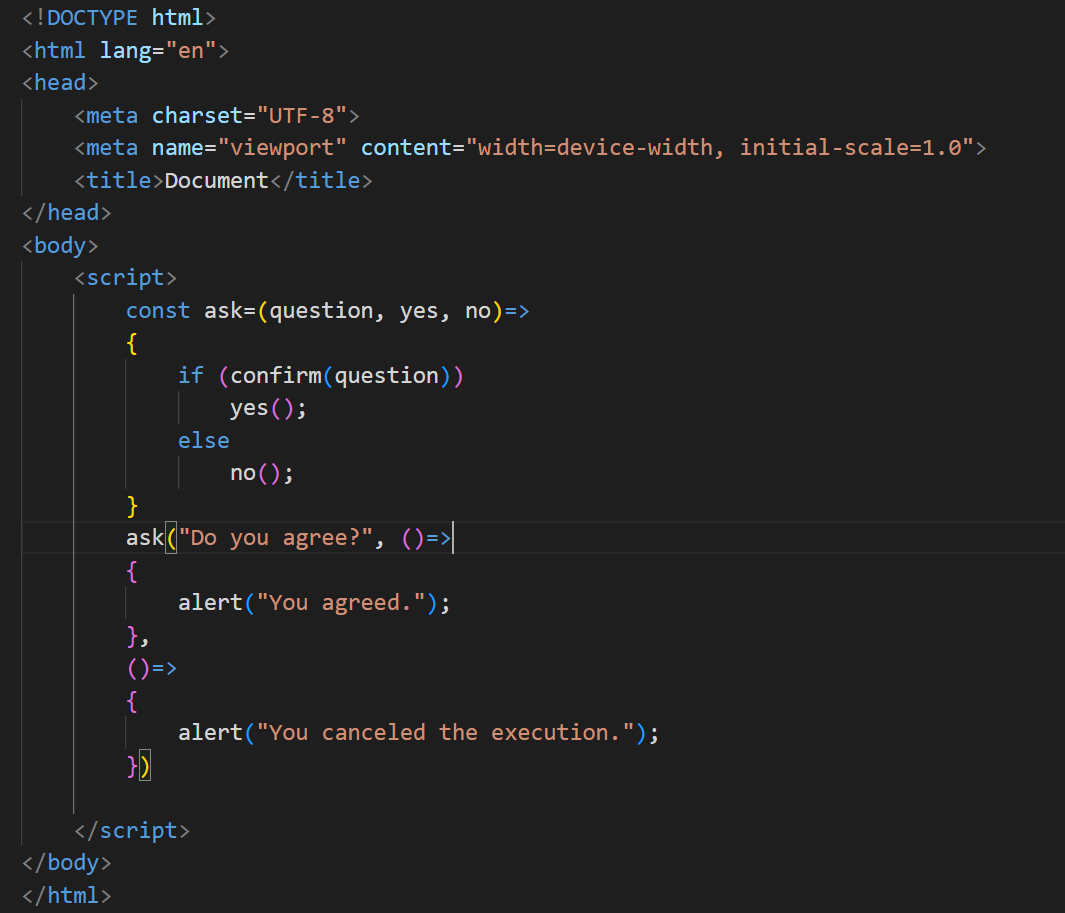
**\* The following function returns true if the parameter age is greater than 18. Otherwise it asks for a confirmation and returns its result: function checkAge(age) { if (age> 18) { return true; } else { // ...return confirm (‘did parents allow you?'); } }**

- 

* Output
* 

    ----

\* **Replace Function Expressions with arrow functions in the code below: Function ask(question, yes, no) { if (confirm(question))yes(); else no(); } ask("Do you agree?", function() { alert("You agreed."); }, function() { alert("You canceled the execution."); } }**

- 

    ---

**Data Types and Objects**

\***Write the code, one line for each action:**

a) Create an empty object user.

b) Add the property name with the value John.

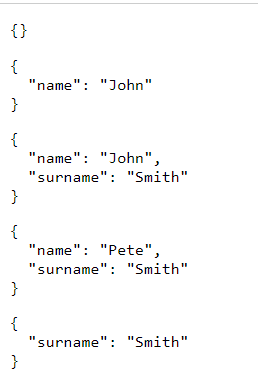
c) Add the property surname with the value Smith.

d) Change the value of the name to Pete.

e) Remove the property name from the object.

- 

**Output:**



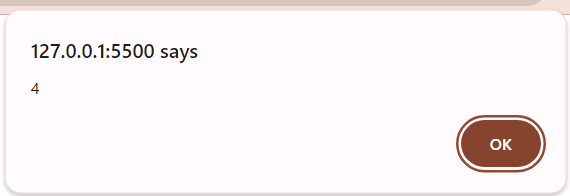
\***Is array copied?**

let fruits = ["Apples", "Pear", "Orange"]; // push a new value into the "copy" let shoppingCart = fruits; shoppingCart.push("Banana"); // what's in fruits? alert( fruits.length ); // ?

- In the given code, shoppingCart is not actually a copy of the fruits array; it's a reference to the same array. When you modify shoppingCart by pushing "Banana" into it, the fruits array will also be affected because both variables point to the same underlying array in memory.



Output:



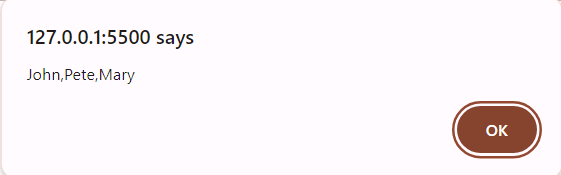
\***Map to names**

let john = { name: "John", age: 25 }; let pete = { name: "Pete", age: 30 }; let mary = { name: "Mary", age: 28 }; let users = [ john, pete, mary ]; let names = /\* ... your code \*/ alert( names ); // John, Pete, Mary

-



**Output:**



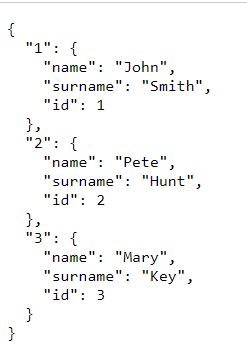
\***Map to objects**

let john = { name: "John", surname: "Smith", id: 1 }; let pete = { name: "Pete", surname: "Hunt", id: 2 }; let mary = { name: "Mary", surname: "Key", id: 3 }; let users = [ john, pete, mary ]; let usersMapped = /\* ... your code ... \*/

/\* usersMapped = [ { fullName: "John Smith", id: 1 }, { fullName: "Pete Hunt", id: 2 }, { fullName: "Mary Key", id: 3 } ] \*/ alert( usersMapped[0].id ) // 1 alert( usersMapped[0].fullName ) // John Smith

-

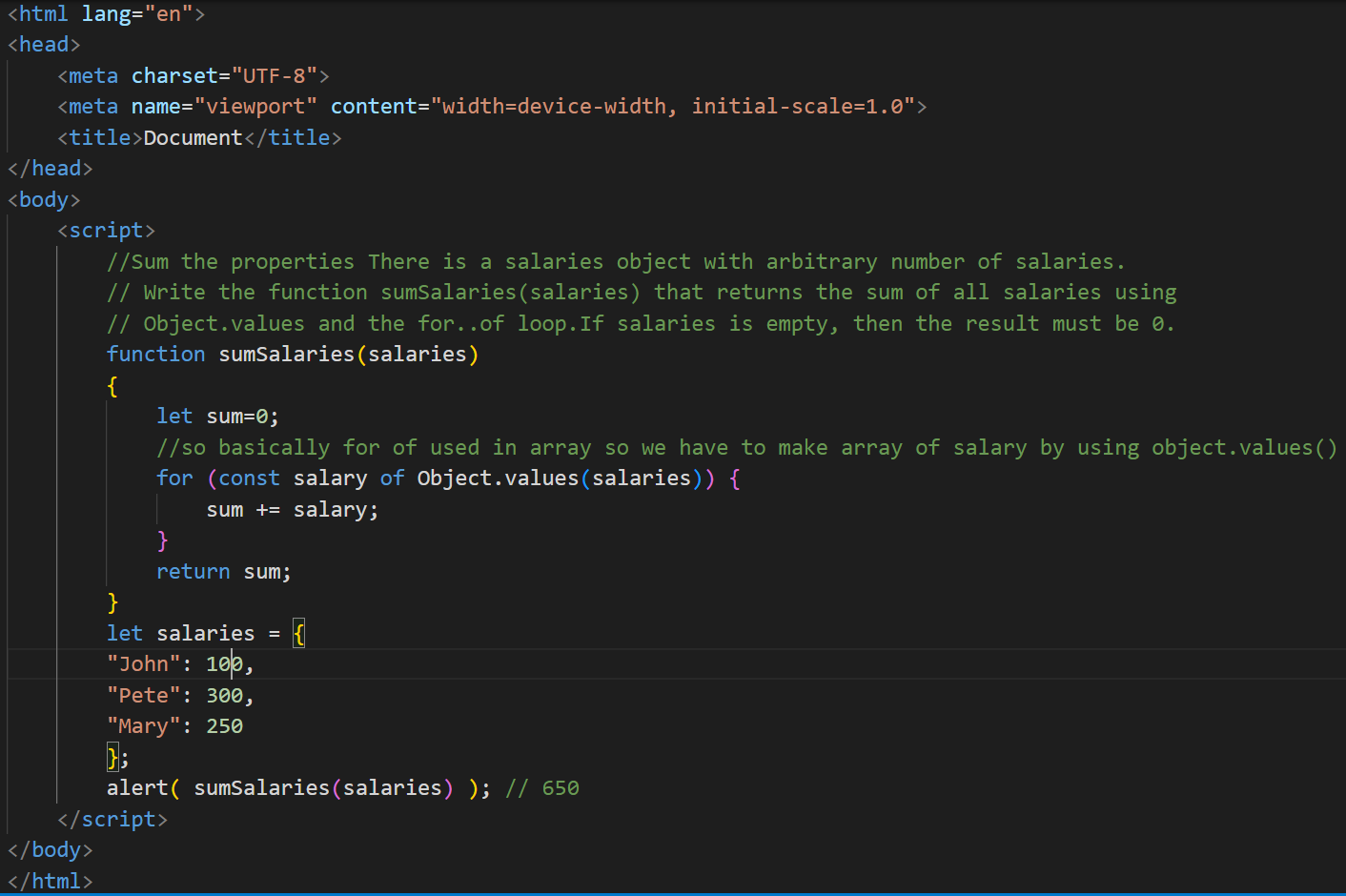
**Output:**



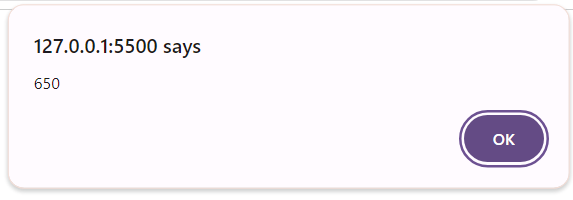
\***Sum the properties There is a salaries object with arbitrary number of salaries.**

Write the function sumSalaries(salaries) that returns the sum of all salaries using Object.values and the for..of loop.If salaries is empty, then the result must be 0. let salaries = { "John": 100, "Pete": 300, "Mary": 250 }; alert( sumSalaries(salaries) ); // 650

-



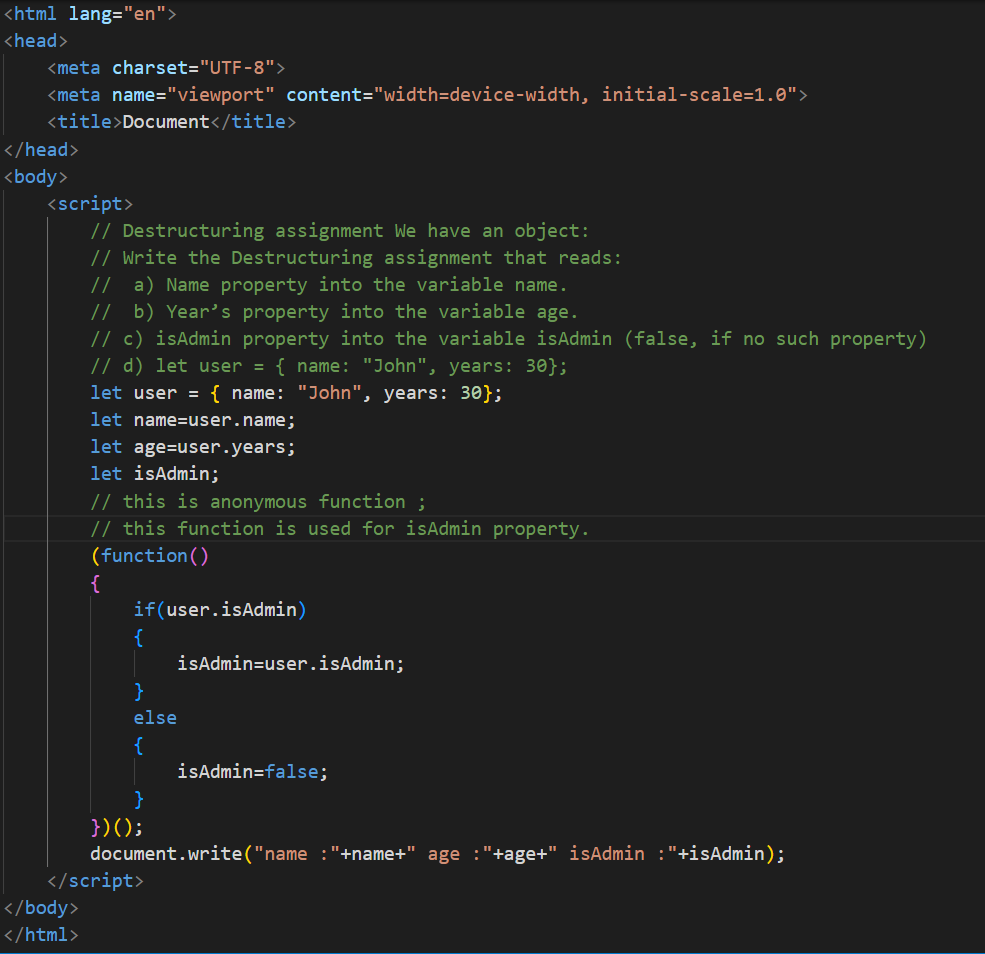
**Output:**



\***Destructuring assignment We have an object:**

Write the Destructuring assignment that reads: a) Name property into the variable name. b) Year’s property into the variable age. c) isAdmin property into the variable isAdmin (false, if no such property) d) let user = { name: "John", years: 30};

-

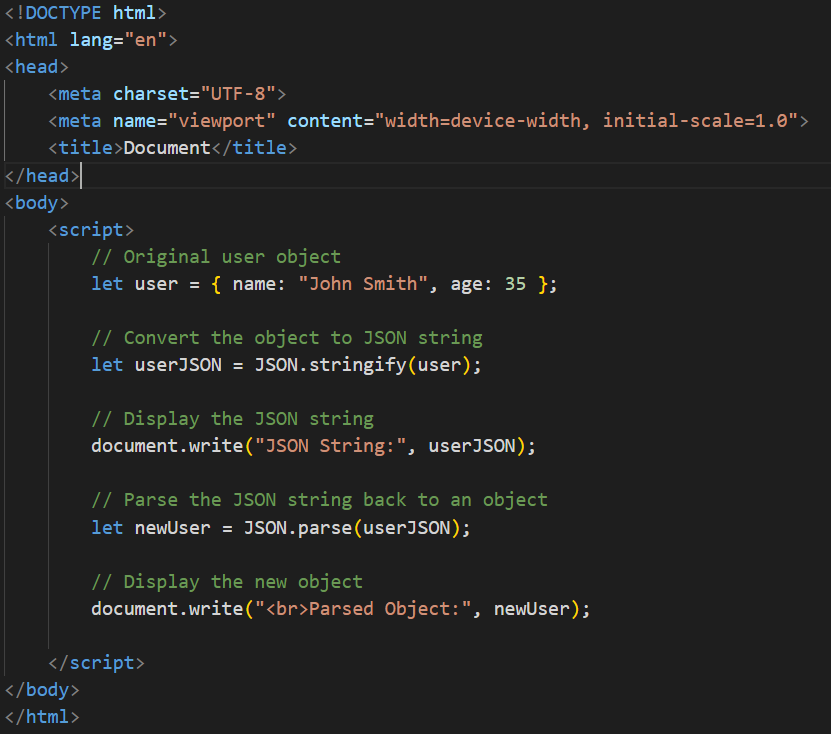


**Output:**

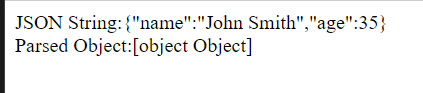
****

\***Turn the object into JSON and back Turn the user into JSON and then read it back into another variable.** user = { name: "John Smith", age: 35};

-

****

**Output:**



**Document, Event and Controls**

\***Create a program to hide/show the password**

**-**

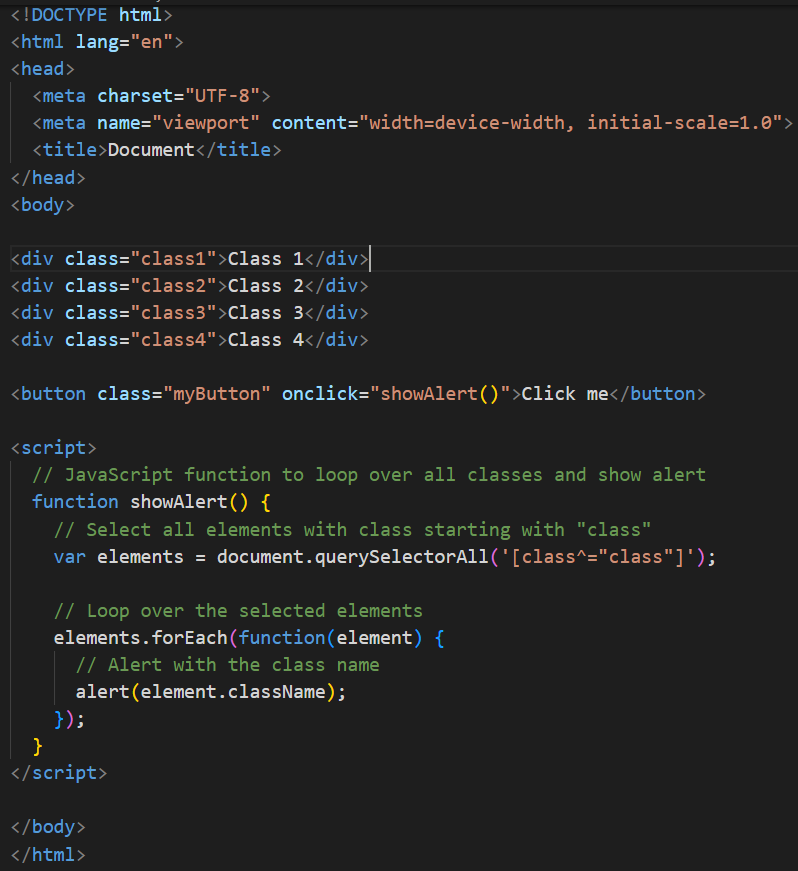


**Output:**

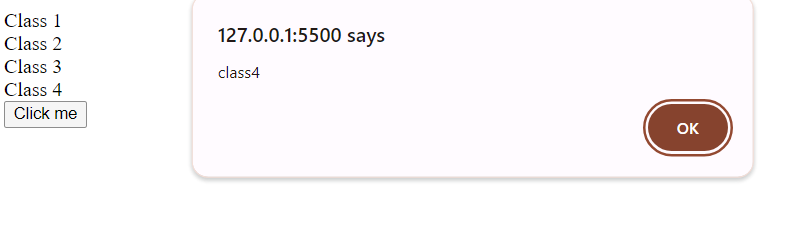


\***Create a program that will select all the classes and loop over and whenever i click the button the alert should show**

**-**

****

**Output:**

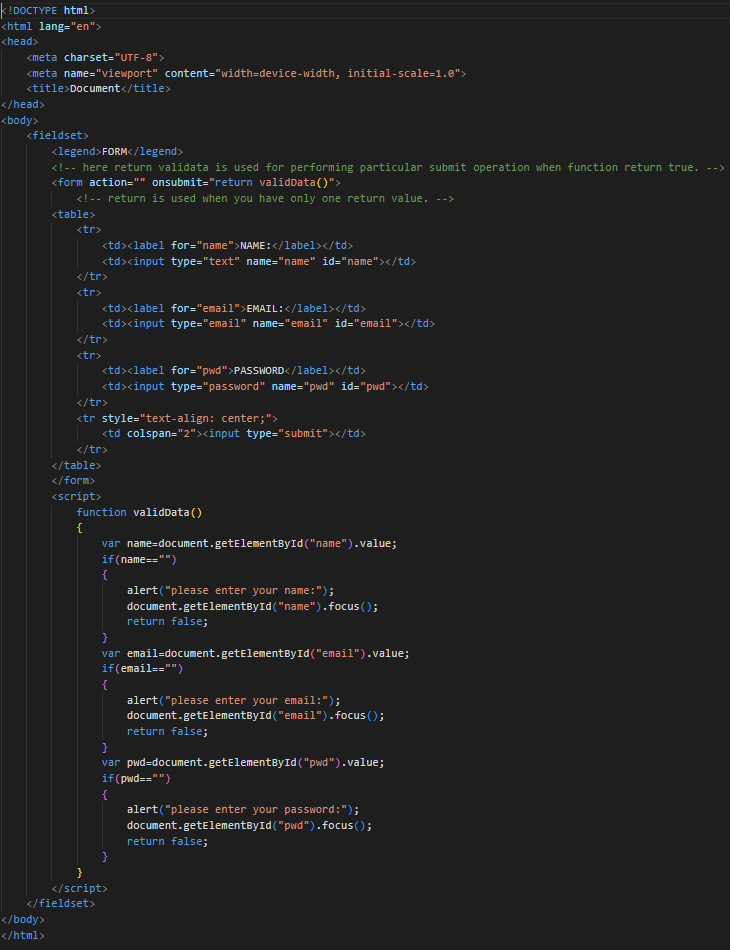


\***Create a responsive header using proper JavaScript**

**-**

\***Create a form and validate using JavaScript**

**-**

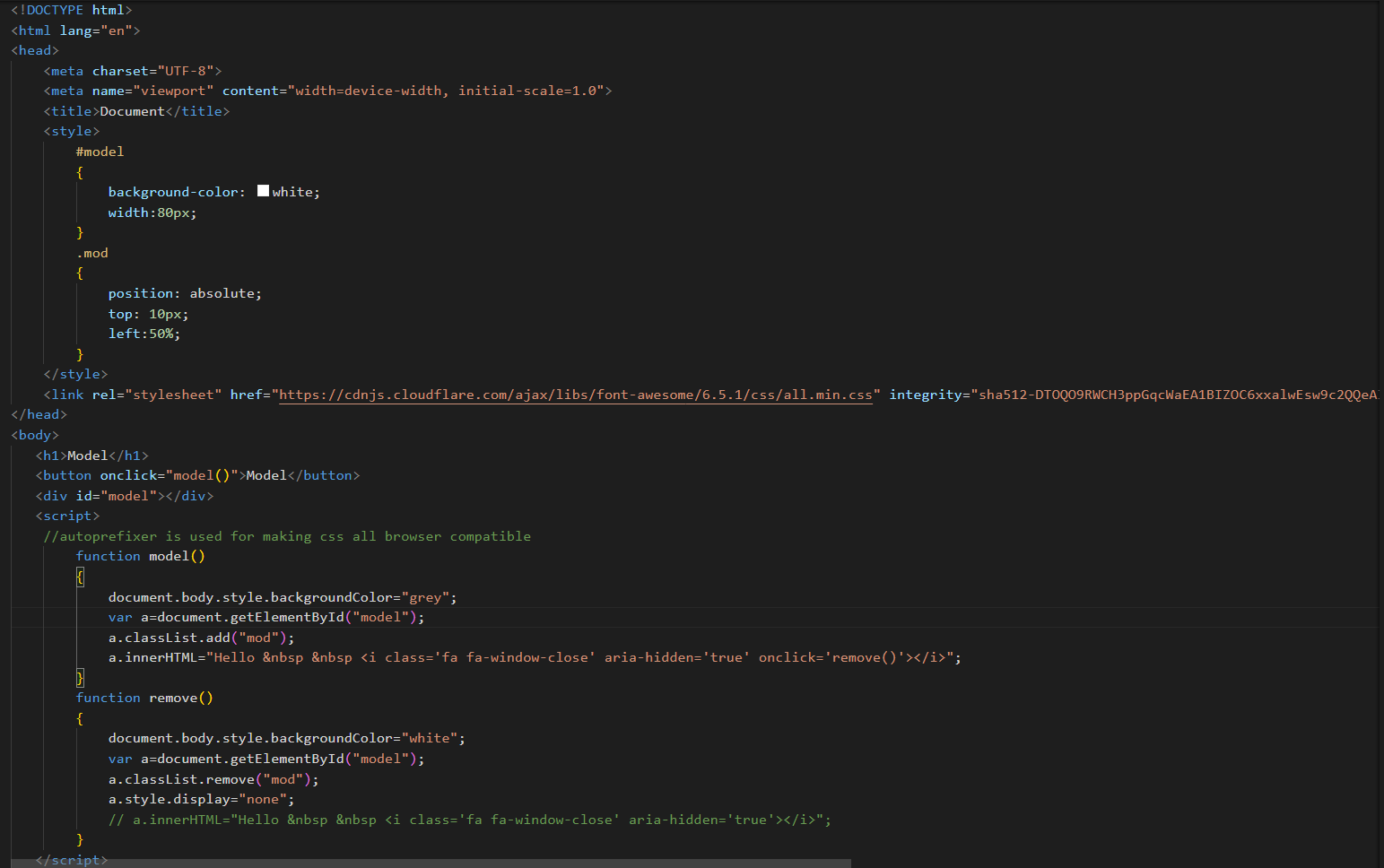


**Output:**

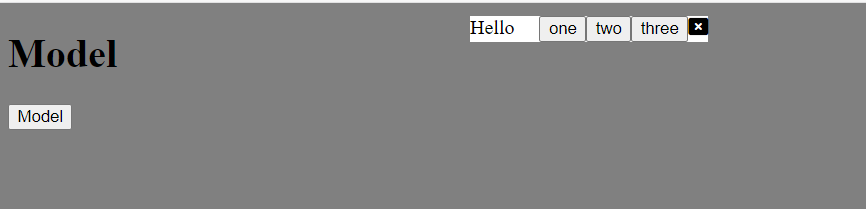


\* **Create a modal box using css and Js with three buttons**

**-**



**Output:**

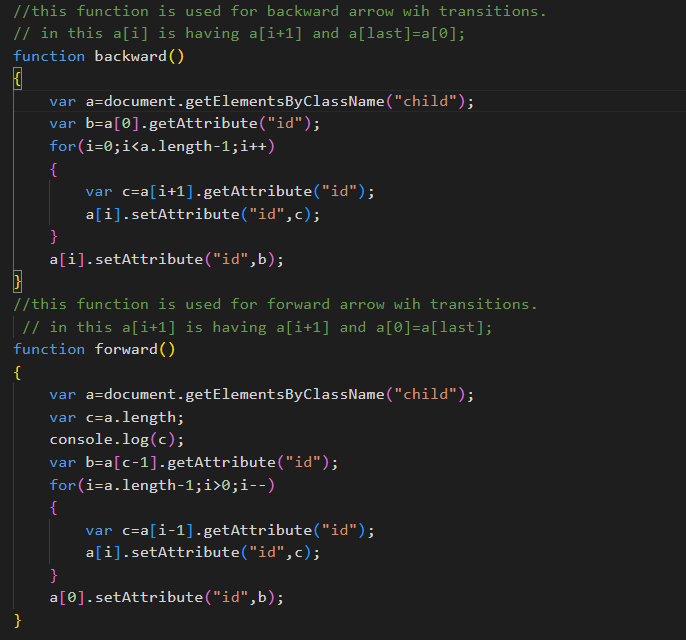


\***Use external js library to show slider**

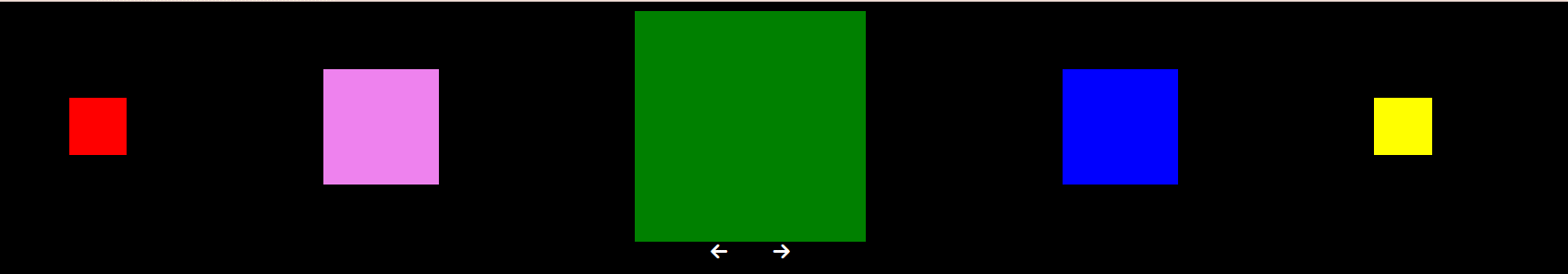
**-**



**External.js:**

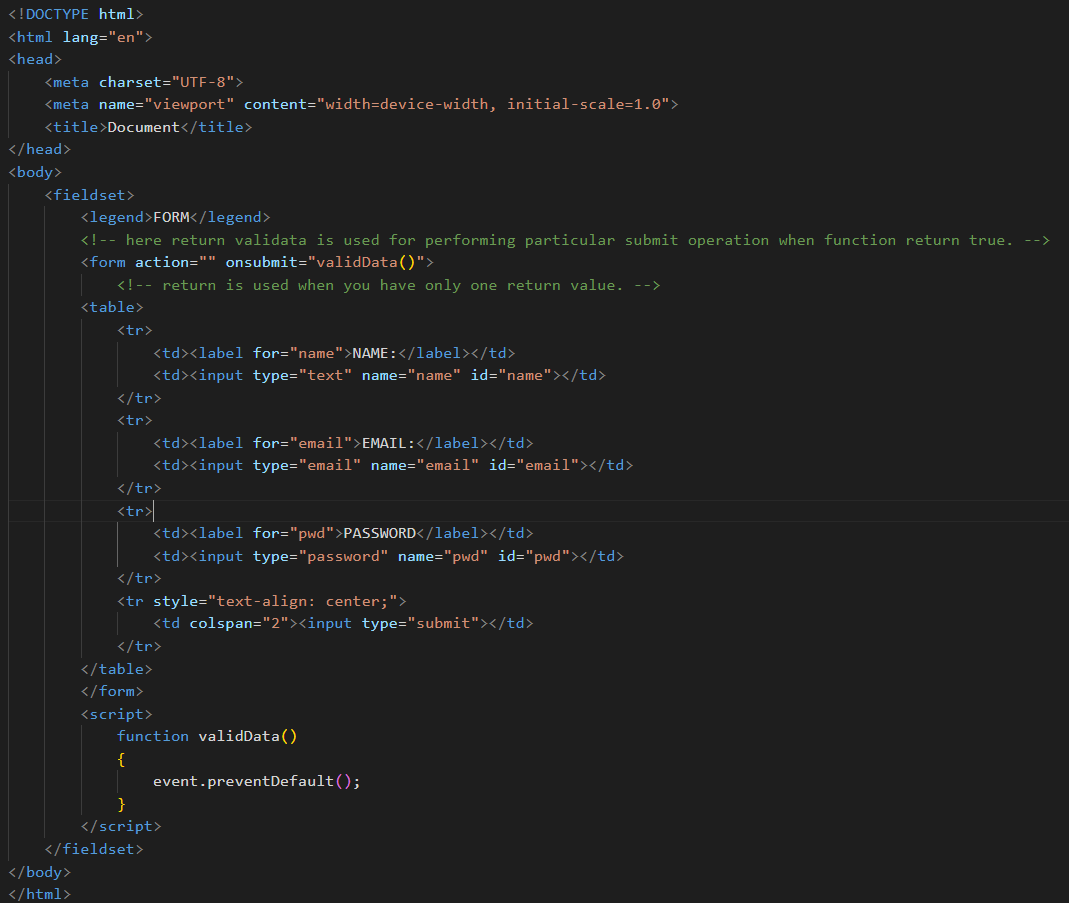
****

**Output:**

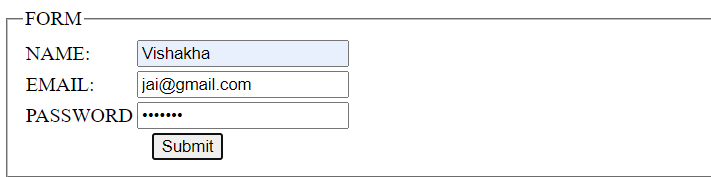


\***Prevent the browser when i click the form submit button**

**-**

****

**Output:**

****

**New Request**

\***What is JSON**

**-** JSON (JavaScript Object Notation) is a lightweight data interchange format used to transmit and store data. It consists of key-value pairs, supports various data types, and is easy for both humans and machines to read and write. JSON is commonly used in web development and APIs.

\***What is promises**

**-** Promises are JavaScript objects that represent the eventual success or failure of an asynchronous operation. They have three states: pending, fulfilled, or rejected. Promises provide a structured way to handle asynchronous code and avoid callback hell. They are often used with the `then` and `catch` methods to handle successful outcomes and errors, respectively. Promises form the foundation for async/await, a more modern and readable way to work with asynchronous operations in JavaScript.

\***Write a program of promises and handle that promises also**

\***Use fetch method for calling an api https://fakestoreapi.com/products**

\***Display all the product from the api in your HTML page**