**WEEK-10**

1. Write a program in Javascript to create variables to store numeric, float, string and

Boolean values and show them in page with document.write() function.

Code-

A screen shot of a computer program

Description automatically generated

Explanation- This HTML code includes a JavaScript script that declares and initializes four variables: `numValue` for a numeric value of `10`, `floatValue` for `3.14`, `stringValue` for the string `"Hello, World!"`, and `booleanValue` for `true`. Each variable's type and value are then outputted to the webpage using `document.write()`. This code serves as a basic demonstration of JavaScript variable declaration and usage for numeric, floating-point, string, and boolean data types.

Output-

A screen shot of a computer

Description automatically generated

2. Write a program to input two data values for Your Name and Student Id in prompt

dialog boxes and show them in alert dialog box.

Code-

A screen shot of a computer program

Description automatically generated

Explanation- This HTML code incorporates a JavaScript script that prompts the user to input their name and student ID using the `prompt()` function. The entered values are stored in variables `name` and `studentId`, respectively. Afterwards, an alert dialog box displays the entered name and student ID concatenated with corresponding labels. This code essentially collects user input through prompts and presents it via an alert, facilitating a basic interaction between the user and the webpage.

Output-

A screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer

Description automatically generated

3. WAP to show the use if .... Else if .... else for following conditions:

a. For marks > = 80 and marks <=100, show Distinction

b. For marks >=60 and marks < 80, show First Division

c. For marks >=50 and marks < 60, show Second Division

d. For marks >=40 and marks <50, show Third Division

e. For marks < 40, show Fail

Code-

A screen shot of a computer program

Description automatically generated

Explanation- This HTML code contains a JavaScript script that prompts the user to input their marks, which are then converted into a numeric value using the `Number()` function. Based on the entered marks, the script evaluates different conditions using `if`, `else if`, and `else` statements to determine the corresponding grade category. If the marks fall within certain ranges, different alerts are triggered to notify the user of their grade, such as Distinction, First Division, Second Division, Third Division, or Fail. This code essentially provides a simple grading system based on user input, offering immediate feedback on their performance.

Output-

A screenshot of a computer

Description automatically generated

A black rectangular object with white text

Description automatically generated

4. WAP to get number input with prompt dialog dox and show the Day using switch case

a. For Case 1, show Sunday

b. For Case 2, show Monday

c. For Case 3, show Tuesday

d. For Case 4, show Wednesday

e. For Case 5, show Thursday

f. For Case 6, show Friday

g. For Case 7, show Saturday

h. For default, show Invalid

Code-

A screenshot of a computer program

Description automatically generated

Explanation- This HTML code utilizes JavaScript to prompt the user for a number between 1 and 7, representing days of the week. After converting the input to a number, a `switch` statement evaluates it against different cases. Each case corresponds to a day of the week, triggering an alert with the matching day. If the input is outside the valid range, an "Invalid input" alert is displayed. This code offers a straightforward way for users to input a number and receive the corresponding day of the week as output.

Output-

A screenshot of a computer

Description automatically generated

A black rectangular object with white text

Description automatically generated

5. WAP in javascript using for loop to display 4 images with image name as 1.jpg, 2.jpg, 3.jpg and 4.jpg.

Code-

A screen shot of a computer program

Description automatically generated

Explanation- This HTML code displays four images on a webpage. The JavaScript function `displayImages()` creates `<img>` elements for each image, setting their source and alternative text accordingly. These images are then appended to a `<div>` element with the ID "imageContainer". Upon loading the webpage, the images are automatically displayed.

Output-

A collage of different images of trees and mountains

Description automatically generated

A road leading to mountains

Description automatically generated

6. WAP to display multiplication number of 5 as follows:

5 \* 1 = 5

5 \* 2 = 10

5 \* 3 = 15

...............

................

5 \* 10 = 50

Code-

A screen shot of a computer program

Description automatically generated

Explanation- This HTML code contains a JavaScript script that generates multiplication tables for the number 5. It utilizes a `for` loop to iterate from 1 to 10. Within each iteration, it calculates the result of 5 multiplied by the current iteration index (`i`). Then, it creates a paragraph element (`<p>`) to display the multiplication expression and result. Finally, it appends this paragraph element to the body of the HTML document. This code dynamically generates and displays the multiplication tables for 5 from 1 to 10 on the webpage.

Output-

A white background with black numbers

Description automatically generated

7. WAP in javascript using loop to display following table layout and with alternate background

color on data rows.

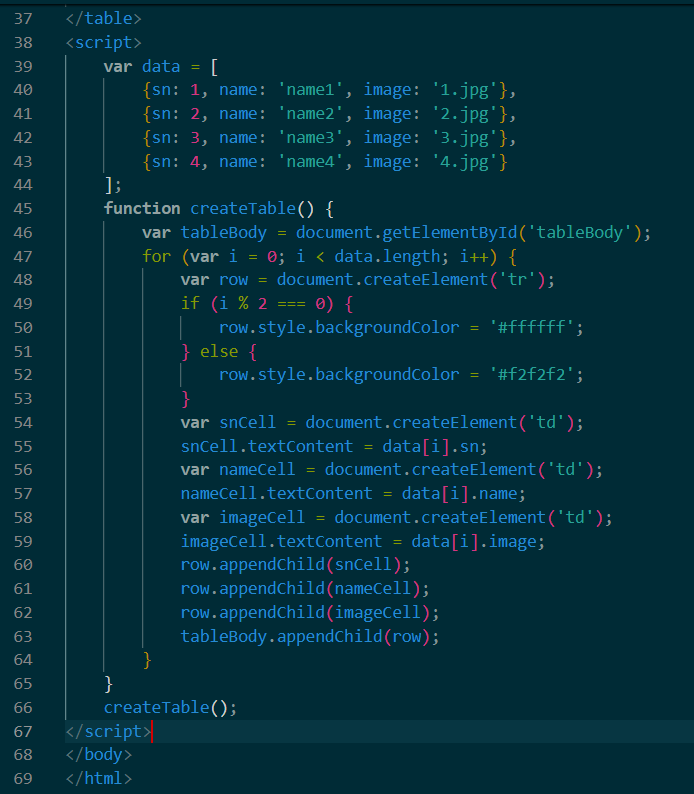
A table with a number and text

Description automatically generated with medium confidence

Code-

A screen shot of a computer program

Description automatically generated



Explanation- This HTML code creates a table with alternating row background colors using CSS. JavaScript dynamically populates the table with data from an array. The table structure is defined in the HTML, with headers for "SN", "Name", and "Image". CSS styling is applied to format the table, including borders and alternating row colors. JavaScript iterates over the data array, creating table rows and cells for each item. It alternates row background colors based on the index of the data item. Finally, the populated table is displayed on the webpage. This code efficiently generates a styled table with dynamic content using HTML, CSS, and JavaScript.

Output-

A screenshot of a computer

Description automatically generated

8. Write a simple JavaScript program to join all elements of the following array into a

string. Sample array: myColor = ["Red", "Green", "White", "Black"];

Expected Output:

"Red,Green,White,Black"

"Red,Green,White,Black"

"Red+Green+White+Black"

Hint: Use these array functions

myColor.toString();

myColor.join();

myColor.join('+');

Code-

A screen shot of a computer program

Description automatically generated

Explanation- This HTML code demonstrates how to join elements of an array into a single string using JavaScript. It defines an array of colors and showcases three methods: `toString()` for comma-separated values, `join()` without a separator for the same result, and `join('+')` for elements joined with a plus sign. The results are displayed on the webpage, showing the array elements joined in different ways. This succinctly illustrates array manipulation in JavaScript for string concatenation.

Output-

A screenshot of a computer

Description automatically generated

9. Write a JavaScript program to compute the sum and product of an array of integers.

Code-



Explanation- This HTML code creates a simple calculator for computing the sum and product of integers entered by the user. It consists of a form with an input field where users can enter integers separated by commas. Upon clicking the "Calculate" button, a JavaScript function is triggered. This function retrieves the input, splits it into an array of integers, and calculates the sum and product iteratively using a loop. The results are then displayed in a div below the input field. This code enables users to quickly compute the sum and product of a list of integers they provide.

Output-

A black text and black text

Description automatically generated

10. Write a Javascript function to calculate area of rectangle which accepts two parameters– length and breadth.

Code-

A screenshot of a computer program

Description automatically generated

Explanation- This HTML code creates a simple calculator to compute the area of a rectangle. It features input fields for users to enter the length and breadth of the rectangle. Upon clicking the "Calculate Area" button, a JavaScript function is invoked. This function retrieves the values entered by the user, validates them to ensure they are valid numbers, and then calculates the area of the rectangle using the formula: length multiplied by breadth. If the inputs are not valid numbers, an error message is displayed. Otherwise, the calculated area is displayed below the input fields. This code provides users with a convenient tool to quickly determine the area of a rectangle based on their inputs.

Output-

A rectangular object with black text

Description automatically generated

11. Write a JavaScript program to add items in an blank array and display the items using

function.

Sample screen:

A screenshot of a computer

Description automatically generated

Code-

A screenshot of a computer program

Description automatically generated

Explanation- This HTML code allows users to add items to an array and display them. It provides an input field for users to enter text, along with "Add" and "Display" buttons. The "Add" button calls a function to add the entered text to the array, alerting the user about the addition and clearing the input field. The "Display" button calls a function to generate an HTML string displaying each element of the array along with its index. This string is then shown in a designated `<div>` element.

Output-

