**Module 6) JAVASCRIPT BASIC & DOM**

**(Basic logic Question)**

**Q.1 What is JavaScript. How to use it?**

**Ans-** JavaScript is a high-level programming language commonly used for creating interactive effects within web browsers. It's a versatile language that can be used both on the client-side (in web browsers) and server-side (with Node.js). To use JavaScript, you typically include <script> tags in your HTML file and write your JavaScript code within these tags or link to external JavaScript files.

**Q.2 How many type of Variable in JavaScript?**

**Ans-** In JavaScript, variables can be of three types: var, let, and const. var is function-scoped, while let and const are block-scoped. let allows reassignment of values, whereas const is used for variables whose values won't change.

**Q.3 Define a Data Types in js?**

**Ans-** JavaScript has several data types including: string, number, boolean, null, undefined, object, and symbol.

**Q.4 Write a mul Function Which will Work Properly When invoked With Following Syntax.**

**Ans-** function mul(x) {

return function(y) {

return x \* y;

};

}

console.log(mul(2)(3)); // Output: 6

**Q.5 What the deference between undefined and undeclare in JavaScript?**

**Ans-**

* undefined means a variable has been declared but has not yet been assigned a value.
* Undeclared means a variable has been used without being declared first.

**Q.6 Using console.log() print out the following statement: The quote 'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes teaches us to help one another. Using console.log() print out the following quote by Mother Teresa:**

**Ans-** console.log("Spread love everywhere you go. Let no one ever come to you without leaving happier. - Mother Teresa");

**Q.7 Check if typeof '10' is exactly equal to 10. If not make it exactly equal?**

**Ans-** if (typeof '10' !== typeof 10) {

console.log("The types are not equal, converting...");

var convertedNumber = parseInt('10');

console.log("Converted:", convertedNumber);

}

**Q.8 Write a JavaScript Program to find the area of a triangle?**

**Ans-** function triangleArea(base, height) {

return 0.5 \* base \* height;

}

// Example usage:

console.log(triangleArea(5, 8)); // Output: 20

**Q.9 Write a JavaScript program to calculate days left until next Christmas?**

**Ans-** function daysUntilChristmas() {

const today = new Date();

const christmas = new Date(today.getFullYear(), 11, 25);

if (today.getMonth() === 11 && today.getDate() > 25) {

christmas.setFullYear(christmas.getFullYear() + 1);

}

const oneDay = 1000 \* 60 \* 60 \* 24;

const daysLeft = Math.ceil((christmas - today) / oneDay);

return daysLeft;

}

console.log("Days left until next Christmas:", daysUntilChristmas());

**Q.10 What is Condition Statement?**

**Ans-** A condition statement, like if, else, and else if, allows you to execute different blocks of code based on the evaluation of a condition. It helps control the flow of a program by making decisions based on certain conditions.

**Q.11 Find circumference of Rectangle formula: C = 4 \* a?**

**Ans-** The given formula calculates the circumference of a rectangle. However, the typical formula for the perimeter (circumference) of a rectangle is C = 2 \* (length + width). If **a** represents the length of one side of the rectangle, then the formula for the circumference of the rectangle would be C = 2 \* (a + a) = 4 \* a.

**Q.12 WAP to convert years into days and days into years?**

**Ans-** function yearsToDays(years) {

return years \* 365;

}

function daysToYears(days) {

return days / 365;

}

// Example usage:

console.log(yearsToDays(2)); // Output: 730

console.log(daysToYears(730)); // Output: 2

**Q.13 Convert temperature Fahrenheit to Celsius? (Conditional logic Question)**

**Ans-** function fahrenheitToCelsius(fahrenheit) {

return (fahrenheit - 32) \* (5/9);

}

// Example usage:

console.log(fahrenheitToCelsius(32)); // Output: 0

**Q.14 Write a JavaScript exercise to get the extension of a filename.?**

**Ans-** function getFileExtension(filename) {

return filename.slice(filename.lastIndexOf('.') + 1);

}

// Example usage:

console.log(getFileExtension('example.docx')); // Output: docx

**Q.15 What is the result of the expression (5 > 3 && 2 < 4)?**

**Ans-** The expression evaluates to true. Both conditions (5 > 3 and 2 < 4) are true, and the && operator returns true only if both conditions are true.

**Q.16 What is the result of the expression (true && 1 && "hello")?**

**Ans-** The expression evaluates to "hello". In JavaScript, the && operator returns the last truthy value if all values are true. Since all values in the expression are truthy, "hello" is returned.

**Q.17 What is the result of the expression true && false || false && true?**

**Ans-** The expression evaluates too false. It follows the rules of operator precedence and short-circuit evaluation. The expression is equivalent to (true && false) || (false && true), which becomes false || false, resulting in false.

**Q.18 What is a Loop and Switch Case in JavaScript define that?**

**Ans-**

* **Loop:** A loop is a programming construct that repeats a block of code multiple times until a certain condition is met. JavaScript has several types of loops including for, while, and do-while.
* **Switch Case:** Switch case is a control flow statement used to select one of many code blocks to be executed. It evaluates an expression, matching the expression's value to a case label, and executes the statements associated with that case.

**Q.19 What is the use of is Nan function?**

**Ans-** 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.

**Q.20 What is the difference between && and || in JavaScript?**

**Ans**-

* && is the logical AND operator. It returns true if both operands are true.
* || is the logical OR operator. It returns true if at least one of the operands is true.

**Q.21 What is the use of Void (0)?**

**Ans-** void(0) is typically used to prevent a page from refreshing when a link is clicked. It evaluates the expression inside void() as undefined, and since undefined has no effect, it effectively stops the browser from navigating to a new page.

**Q.22 Check Number Is Positive or Negative in JavaScript?**

**Ans-** function checkPositiveOrNegative(number) {

if (number > 0) {

return "Positive";

} else if (number < 0) {

return "Negative";

} else {

return "Zero";

}

}

// Example usage:

console.log(checkPositiveOrNegative(-5)); // Output: Negative

**Q.23 Find the Character Is Vowel or Not?**

**Ans-** function isVowel(char) {

const vowels = ['a', 'e', 'i', 'o', 'u'];

return vowels.includes(char.toLowerCase());

}

// Example usage:

console.log(isVowel('a')); // Output: true

**Q.24 Write to check whether a number is negative, positive or zero?**

**Ans-** function checkNumber(number) {

if (number > 0) {

return "Positive";

} else if (number < 0) {

return "Negative";

} else {

return "Zero";

}

}

// Example usage:

console.log(checkNumber(0)); // Output: Zero

**Q.25 Write to find number is even or odd using ternary operator in JS?**

**Ans-** function checkEvenOrOdd(number) {

return number % 2 === 0 ? "Even" : "Odd";

}

// Example usage:

console.log(checkEvenOrOdd(7)); // Output: Odd

**Q.26 Write find maximum number among 3 numbers using ternary operator in JS?**

**Ans-** function findMax(a, b, c) {

return (a > b ? (a > c ? a : c) : (b > c ? b : c));

}

// Example usage:

console.log(findMax(5, 3, 9)); // Output: 9

**Q.27 Write to find minimum number among 3 numbers using ternary operator in JS?**

**Ans-** function findMin(a, b, c) {

return (a < b ? (a < c ? a : c) : (b < c ? b : c));

}

// Example usage:

**console.log(findMin(5, 3, 9)); // Output: 3**

**Q.28 Write to find the largest of three numbers in JS?**

**Ans-** function findLargest(a, b, c) {

return Math.max(a, b, c);

}

// Example usage:

console.log(findLargest(5, 3, 9)); // Output: 9

**Q.29 Write to show**

**i. Monday to Sunday using switch case in JS?**

**Ans-** function getDayName(day) {

switch (day) {

case 1:

return "Monday";

case 2:

return "Tuesday";

case 3:

return "Wednesday";

case 4:

return "Thursday";

case 5:

return "Friday";

case 6:

return "Saturday";

case 7:

return "Sunday";

default:

return "Invalid day";

}

}

// Example usage:

console.log(getDayName(3)); // Output: Wednesday

**ii. Vowel or Consonant using switch case in JS?**

**Ans-** function checkVowelOrConsonant(char) {

switch (char.toLowerCase()) {

case 'a':

case 'e':

case 'i':

case 'o':

case 'u':

return "Vowel";

default:

return "Consonant";

}

}

// Example usage:

console.log(checkVowelOrConsonant('e')); // Output: Vowel

**(Conditional looping logic Question)**

**Q.30 What are the looping structures in JavaScript? Any one Example?**

**Ans-** The looping structures in JavaScript include for, while, and do-while loops. Here's an example of a for loop:

for (let i = 0; i < 5; i++) {

console.log(i);

}

**Q.31 Write a print 972 to 897 using for loop in JS?**

**Ans-** for (let i = 972; i >= 897; i--) {

console.log(i);

}

**Q.32 Write to print factorial of given number?**

**Ans-** function factorial(n) {

if (n === 0 || n === 1) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

// Example usage:

console.log(factorial(5)); // Output: 120

**Q.33 Write to print Fibonacci series up to given numbers?**

**Ans-** function fibonacciSeries(limit) {

let fibArray = [0, 1];

for (let i = 2; fibArray[i - 1] + fibArray[i - 2] <= limit; i++) {

fibArray.push(fibArray[i - 1] + fibArray[i - 2]);

}

return fibArray.join(", ");

}

// Example usage:

console.log(fibonacciSeries(50)); // Output: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

**Q.34 Write to print number in reverse order e.g.: number = 64728 ---> reverse =82746 in JS?**

**Ans-** function reverseNumber(number) {

return parseInt(number.toString().split('').reverse().join(''));

}

// Example usage:

console.log(reverseNumber(64728)); // Output: 82746

**Q.35 Write a program make a summation of given number (E.g., 1523 Ans: - 11) in JS?**

**Ans-** function sumOfDigits(number) {

let sum = 0;

while (number > 0) {

sum += number % 10;

number = Math.floor(number / 10);

}

return sum;

}

// Example usage:

console.log(sumOfDigits(1523)); // Output: 11

**Q.36 Write a program you have to make a summation of first and last Digit. (E.g., 1234 Ans: - 5) in JS?**

**Ans-** function sumOfFirstAndLastDigit(number) {

const lastDigit = number % 10;

while (number >= 10) {

number = Math.floor(number / 10);

}

const firstDigit = number;

return firstDigit + lastDigit;

}

// Example usage:

console.log(sumOfFirstAndLastDigit(1234)); // Output: 5

**Q.37 Use console.log() and escape characters to print the following pattern in JS?**

**1 1 1 1 1**

**2 1 2 4 8**

**3 1 3 9 27**

**4 1 4 16 64**

**5 1 5 25 125**

**Ans-** for (let i = 1; i <= 5; i++) {

let row = '';

for (let j = 1; j <= i; j++) {

row += Math.pow(i, j) + ' ';

}

console.log(row);

}

**Q.38 Use pattern in console.log in JS?**

**1) 1**

**1 0**

**1 0 1**

**1 0 1 0**

**1 0 1 0 1**

**Ans-** for (let i = 1; i <= 5; i++) {

let row = '';

for (let j = 1; j <= i; j++) {

row += (j % 2 === 0) ? '0 ' : '1 ';

}

console.log(row);

}

**2) A**

**B C**

**D E F**

**G H I J**

**K L M N O**

**Ans-** let charCode = 65;

for (let i = 1; i <= 5; i++) {

let row = '';

for (let j = 1; j <= i; j++) {

row += String.fromCharCode(charCode++) + ' ';

}

console.log(row);

}

**3) 1**

**2 3**

**4 5 6**

**7 8 9 10**

**11 12 13 14 15**

**Ans-** let counter = 1;

for (let i = 1; i <= 5; i++) {

let row = '';

for (let j = 1; j <= i; j++) {

row += counter++ + ' ';

}

console.log(row);

}

**4) \***

**\* \***

**\* \***

**\* \* \* \***

**\* \* \* \* \* \***

**Ans-** for (let i = 1; i <= 5; i++) {

let row = '';

for (let j = 1; j <= i; j++) {

row += '\* ';

}

console.log(row);

}

**Q.39 Accept 3 numbers from user using while loop and check each numbers palindrome?**

**Ans-** function isPalindrome(number) {

const originalNumber = number;

let reverse = 0;

while (number > 0) {

const remainder = number % 10;

reverse = reverse \* 10 + remainder;

number = Math.floor(number / 10);

}

return originalNumber === reverse;

}

let i = 0;

while (i < 3) {

let userInput = prompt("Enter a number:");

const number = parseInt(userInput);

if (isPalindrome(number)) {

console.log(number + " is a palindrome.");

} else {

console.log(number + " is not a palindrome.");

}

i++;

}

**(Array and object Question)**

**Q.40 Write a JavaScript Program to display the current day and time in the following format. Sample Output: Today is Friday. Current Time is 12 PM: 12 : 22 2 ?**

**Ans-** const today = new Date();

const day = today.toLocaleString('default', { weekday: 'long' });

const hours = today.getHours() > 12 ? today.getHours() - 12 : today.getHours();

const minutes = today.getMinutes();

const meridiem = today.getHours() >= 12 ? 'PM' : 'AM';

console.log(`Today is ${day}. Current Time is ${hours} ${meridiem}: ${minutes}`);

**Q.41 Write a JavaScript program to get the current date?**

**Ans-** const currentDate = new Date();

console.log(currentDate.toDateString());

**Q.42 Write a JavaScript program to compare two objects?**

**Ans-** function compareObjects(obj1, obj2) {

return JSON.stringify(obj1) === JSON.stringify(obj2);

}

// Example usage:

const obj1 = { name: 'John', age: 30 };

const obj2 = { name: 'John', age: 30 };

console.log(compareObjects(obj1, obj2)); // Output: true

**Q.43 Write a JavaScript program to convert an array of objects into CSV string?**

**Ans-** function convertArrayToCSV(array) {

const headers = Object.keys(array[0]).join(',');

const values = array.map(obj => Object.values(obj).join(',')).join('\n');

return `${headers}\n${values}`;

}

// Example usage:

const array = [

{ name: 'John', age: 30 },

{ name: 'Alice', age: 25 }

];

console.log(convertArrayToCSV(array));

**Q.44 Write a JavaScript program to capitalize first letter of a string?**

**Ans-** function capitalizeFirstLetter(str) {

return str.charAt(0).toUpperCase() + str.slice(1);

}

// Example usage:

console.log(capitalizeFirstLetter('javascript')); // Output: Javascript

**Q. 45 Write a JavaScript program to determine if a variable is array?**

**Ans-** function isArray(variable) {

return Array.isArray(variable);

}

// Example usage:

console.log(isArray([1, 2, 3])); // Output: true

**Q.46 Write a JavaScript program to clone an array?**

**Ans-** function cloneArray(array) {

return [...array];

}

// Example usage:

const originalArray = [1, 2, 3];

const clonedArray = cloneArray(originalArray);

console.log(clonedArray); // Output: [1, 2, 3]

**Q.47 What is the drawback of declaring methods directly in JavaScript objects?**

**Ans-** The drawback is that when methods are declared directly in JavaScript objects, each instance of the object will have its own copy of the method, consuming more memory. It's more memory-efficient to use prototype-based methods.

**Q.48 Print the length of the string on the browser console using console.log()?**

**Ans-** const str = 'Hello, world!';

console.log(str.length); // Output: 13

**Q.49 Change all the string characters to capital letters using toUpperCase() method?**

**Ans-** const str = 'hello, world!';

const capitalizedStr = str.toUpperCase();

console.log(capitalizedStr); // Output: HELLO, WORLD!

**Q.50 What is the drawback of declaring methods directly in JavaScript objects?**

**Ans-** The drawback is that when methods are declared directly in JavaScript objects, each instance of the object will have its own copy of the method, consuming more memory. It's more memory-efficient to use prototype-based methods.

**Q.51 Write a JavaScript program to get the current date. Expected Output : mm-dd-yyyy, mm/dd/yyyy or dd-mm-yyyy, dd/mm/yyyy?**

**Ans-** const currentDate = new Date();

const day = currentDate.getDate().toString().padStart(2, '0');

const month = (currentDate.getMonth() + 1).toString().padStart(2, '0');

const year = currentDate.getFullYear();

const separator = '/'; // Change this to '-' or any other separator as needed

console.log(`${month}${separator}${day}${separator}${year}`);

**Q.52 Use indexOf to determine the position of the first occurrence of a in 30 Days Of JavaScript?**

**Ans-** const str = '30 Days Of JavaScript';

const position = str.indexOf('a');

console.log(position); // Output: 4

**Q,53 Use lastIndexOf to determine the position of the last occurrence of a in 30 Days Of JavaScript?**

**Ans-** const str = '30 Days Of JavaScript';

const position = str.lastIndexOf('a');

console.log(position); // Output: 15

**Q.54 Form Validtion in JS?**

**Ans-** Form validation in JavaScript involves ensuring that user input in HTML forms meets certain criteria before it is submitted to the server. This can be done using JavaScript by checking input values against specified conditions, such as required fields, valid email addresses, numeric values, etc.

**Q.55 Form in Email, number, Password, Validation?**

**Ans-** <!DOCTYPE html>

<html>

<head>

<title>Form Validation</title>

<script>

function validateForm() {

const email = document.getElementById('email').value;

const number = document.getElementById('number').value;

const password = document.getElementById('password').value;

// Email validation

if (!/^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/.test(email)) {

alert('Please enter a valid email address');

return false;

}

// Number validation

if (isNaN(number)) {

alert('Please enter a valid number');

return false;

}

// Password validation

if (password.length < 8) {

alert('Password must be at least 8 characters long');

return false;

}

return true;

}

</script>

</head>

<body>

<form onsubmit="return validateForm()">

Email: <input type="text" id="email" name="email"><br>

Number: <input type="text" id="number" name="number"><br>

Password: <input type="password" id="password" name="password"><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

**Q.56 Dynamic Form Validation in JS?**

**Ans-** Dynamic form validation involves validating form fields as the user interacts with them, providing instant feedback. This can be achieved using event listeners such as input, blur, etc., and dynamically updating error messages or visual indicators based on the input.

**Q.57 how many type of JS Event? How to use it?**

**Ans-** JavaScript events can be categorized into different types such as mouse events, keyboard events, form events, document/window events, etc. They can be used by attaching event listeners to HTML elements using event handler attributes (e.g., onclick, onchange, onkeydown) or by using JavaScript methods like addEventListener().

**Q.58 What is Bom vs Dom in JS?**

**Ans-** BOM (Browser Object Model) refers to the browser-specific objects and interfaces that allow JavaScript to interact with the browser window (e.g., window, navigator, screen), whereas DOM (Document Object Model) refers to the tree-like structure of HTML documents and provides methods and properties for accessing and manipulating HTML elements.

**Q.59 Array vs object defences in JS?**

**Ans-** Arrays are ordered collections of values accessed by numerical indices, suitable for storing lists of similar items. Objects are unordered collections of key-value pairs accessed by keys (strings), suitable for representing entities with named properties.

**Q.60 Split the string into an array using split() Method?**

**Ans-** const str = 'Hello, world!';

const arr = str.split(', '); // Splitting by comma and space

console.log(arr); // Output: ['Hello', 'world!']

**Q.61 Check if the string contains a word Script using includes() method?**

**Ans-** const str = '30 Days of JavaScript';

const containsScript = str.includes('Script');

console.log(containsScript); // Output: true

**Q.62 Change all the string characters to lowercase letters using toLowerCase() Method.**

**Ans-** const str = 'Hello, World!';

const lowerCaseStr = str.toLowerCase();

console.log(lowerCaseStr); // Output: hello, world!

**Q.63 What is Character at index 15 in ’30 Days of JavaScript’ string? Use charAt() method.**

**Ans-** const str = '30 Days of JavaScript';

const charAtIndex15 = str.charAt(15);

console.log(charAtIndex15); // Output: J

**Q.64 copy to one string to another string in JS?**

**Ans-** const str1 = 'Hello, world!';

const str2 = str1;

console.log(str2); // Output: Hello, world!

**Q.65 Find the length of a string without using libraryFunction?**

**Ans-** const str = 'Hello, world!';

let length = 0;

for (let char of str) {

length++;

}

console.log(length); // Output: 13

**• What is JavaScript?**

**Ans-** JavaScript is a high-level, interpreted programming language primarily used for creating interactive and dynamic web content. It is widely used in web development for adding behavior to web pages, creating web applications, and building server-side applications using frameworks like Node.js.

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

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

**• What is negative Infinity?**

**Ans-** Negative Infinity is a special value in JavaScript representing the negative infinite number, which is smaller than any other number. It can be obtained by dividing a negative number by zero or by calling Number.NEGATIVE\_INFINITY.

**• Which company developed JavaScript?**

**Ans-** JavaScript was developed by Netscape Communications Corporation, primarily by Brendan Eich. It was originally named Mocha, then LiveScript, and later renamed JavaScript.

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

**Ans-**

* Undeclared variables: Variables that are not declared using var, let, or const keywords are considered undeclared. Accessing undeclared variables will result in a reference error.
* Undefined variables: Variables that are declared but not initialized with a value are considered undefined. Accessing undefined variables will result in the value undefined.

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

**Ans-** To add new elements dynamically to the DOM (Document Object Model), you can use methods like createElement() and appendChild():

// Create a new element

const newElement = document.createElement('div');

// Add content to the new element

newElement.textContent = 'New Element';

// Append the new element to an existing element

const parentElement = document.getElementById('parent');

parentElement.appendChild(newElement);

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

**Ans-**

* **ViewState:** ViewState is a client-side state management technique used in ASP.NET web forms. It stores the state of a web page's controls in a hidden field on the page itself. ViewState is used to maintain the state of controls during postbacks within the same page.
* **SessionState:** SessionState is a server-side state management technique used to store user-specific information across multiple requests. It stores data associated with a user session on the server and can be accessed across different pages of the web application.

**• What is === operator?**

**Ans-** The === operator is the strict equality operator in JavaScript. It compares two values without performing type conversion. It returns true if the operands are equal and of the same type, otherwise false.

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

**Ans-** The style or class of an HTML element can be changed using JavaScript:

// Change style

const element = document.getElementById('example');

element.style.color = 'red';

// Change class

element.className = 'new-class';

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

**Ans-** In a browser environment, JavaScript cannot directly read or write files on the user's filesystem due to security restrictions. However, you can read files using file input elements and the FileReader API, and write files using techniques such as blob URLs or data URLs. In a server-side environment like Node.js, you can use the built-in **fs** module to read and write files.

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

**Ans-** JavaScript provides several looping structures:

* **for loop:** Executes a block of code a specified number of times.
* **while loop:** Executes a block of code as long as a specified condition is true.
* **do-while loop:** Similar to a while loop, but the block of code is executed at least once, even if the condition is false.
* **for...in loop:** Iterates over the properties of an object.
* **for...of loop:** Iterates over the iterable objects like arrays, strings, maps, sets, etc.

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

**Ans-** You can convert a string representing a number in any base to an integer using the parseInt() function. You need to specify the string to be converted and the base as arguments. For example:

const binaryString = '1010';

const decimalNumber = parseInt(binaryString, 2);

console.log(decimalNumber); // Output: 10

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

**Ans-** The delete operator in JavaScript is used to remove a property from an object. It can also be used to remove elements from an array. However, it cannot delete variables declared with the var, let, or const keywords.

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

**Ans-** JavaScript provides three types of popup boxes:

**alert():** Displays an alert dialog box with a specified message and an OK button.

**confirm():** Displays a dialog box with a specified message and OK/Cancel buttons. It returns true if the user clicks OK and false if the user clicks Cancel.

**prompt():** Displays a dialog box with a message, an input field for the user to enter data, and OK/Cancel buttons. It returns the text entered by the user if OK is clicked, or null if Cancel is clicked.

**• What is the use of Void (0)?**

**Ans-** The void(0) expression in JavaScript is used to return undefined. It is commonly used as the href attribute value in anchor tags <a> to prevent the page from navigating to a new URL when the link is clicked.

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

**Ans-** You can force a page to load another page in JavaScript using the window.location object. For example:

// Load a new page

window.location.href = 'https://example.com';

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

**Ans-** Using innerHTML to manipulate the HTML content of an element can be less efficient compared to other methods like DOM manipulation. It can also introduce security risks such as cross-site scripting (XSS) vulnerabilities if the content being inserted is not properly sanitized. Additionally, setting innerHTML will destroy any existing event listeners or data associated with the element's children.

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

**Ans-**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    Enter Password:

    <input type="password" id="passwordField">

    <div class="mb-3 form-check">

      <input type="checkbox" class="form-check-input" onclick="togglePasswordVisibility()">

      <label class="form-check-label" for="check1">Show password</label>

    </div>

</body>

</html>

<script>

function togglePasswordVisibility() {

  var passwordField = document.getElementById("passwordField");

  if (passwordField.type === "password") {

    passwordField.type = "text";

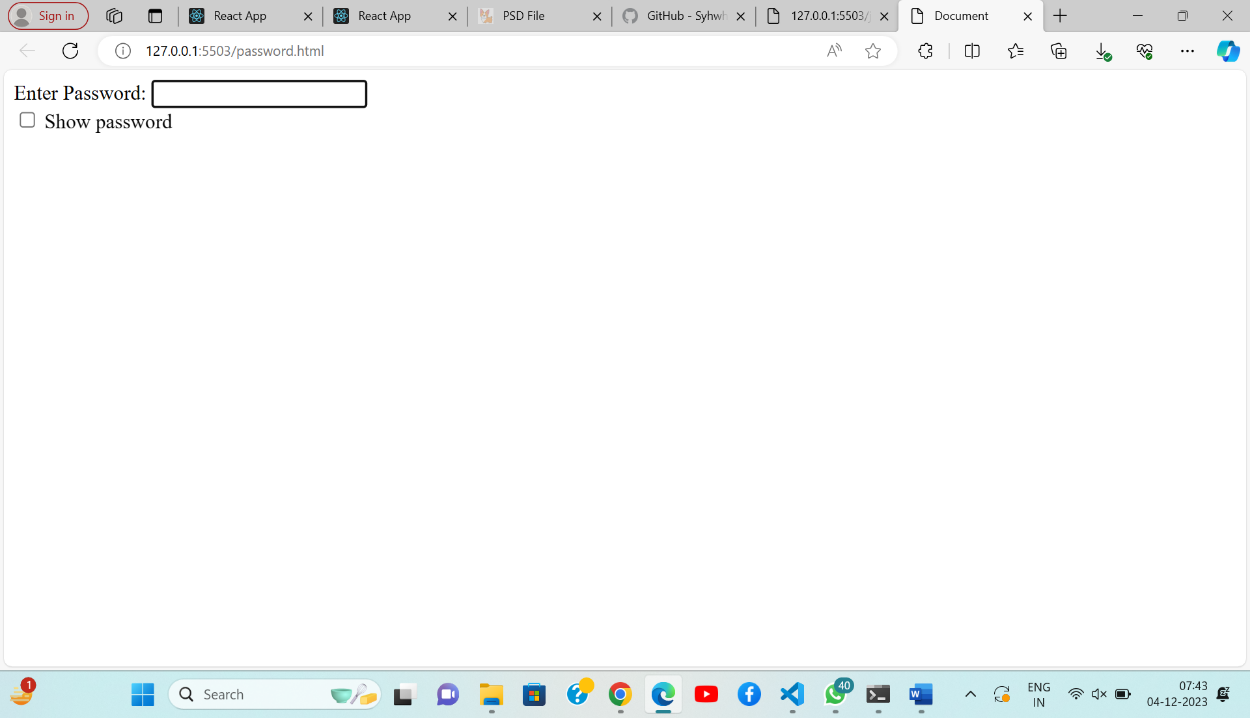
  } else {

    passwordField.type = "password";

  }

}

</script>



* **Create basic math operation in JS**

**Ans-**

<!DOCTYPE html>

<html>

<head>

    <title>Calculator</title>

</head>

<body>

    <h2>Calculator</h2>

    <label for="value1">Enter 1st value: </label>

    <input type="number" id="value1" /><br>

    <label for="value2">Enter 2nd value: </label>

    <input type="number" id="value2" /><br>

    <button onclick="calculate('+')">+</button>

    <button onclick="calculate('-')">-</button>

    <button onclick="calculate('\*')">\*</button> <br>

    <button onclick="calculate('/')">/</button>

    <button onclick="calculate('%')">%</button>

 <p>Answer is: <span id="result">-</span></p>

    <script>

        function calculate(operator) {

            var value1 = parseFloat(document.getElementById('value1').value);

            var value2 = parseFloat(document.getElementById('value2').value);

            var resultElement = document.getElementById('result');

            if (isNaN(value1) || isNaN(value2)) {

                resultElement.textContent = "Invalid input";

            } else {

                switch (operator) {

                    case '+':

                        resultElement.textContent =(value1 + value2);

                        break;

                    case '-':

                        resultElement.textContent =(value1 - value2);

                        break;

                    case '\*':

                        resultElement.textContent = (value1 \* value2);

                        break;

                    case '/':

                            resultElement.textContent = (value1 / value2);

                        break;

                     case '%':

                        resultElement.textContent = (value1 / 100)\*value2;

                        break;

                    default:

                        resultElement.textContent = "Invalid operator";

                        break;

                }

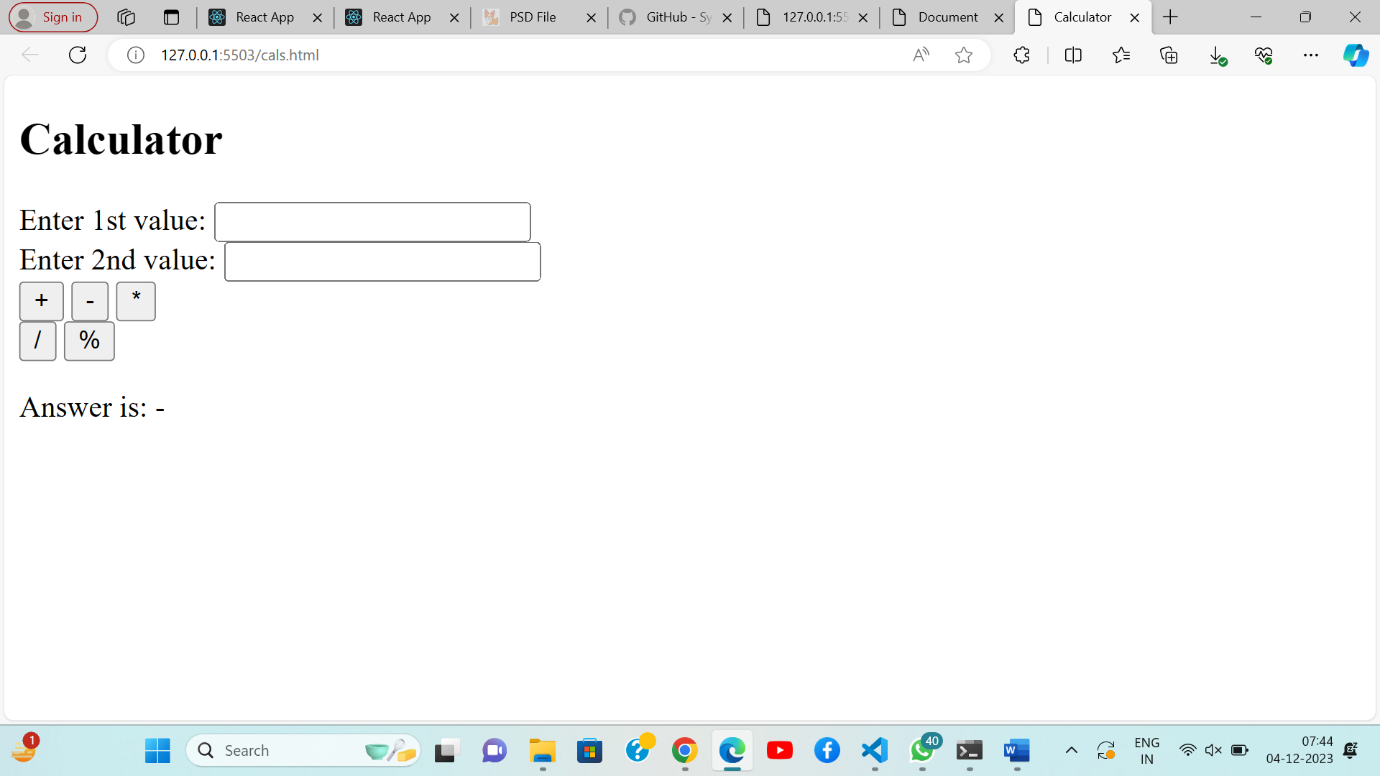
            }

        }

    </script>

</body>

</html>



* **Create result**

**Ans-**

<!DOCTYPE html>

<html>

<head>

    <title>Grade Calculator</title>

    <style>

        input{

            margin-left: 30px;

        }

        li{

            list-style: none;

        }

    </style>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

</head>

<body>

    <table>

        <tr>

            <th colspan="2" class="heading fs-4">Marksheet For Information Technology</th>

            <!-- <td></td> -->

        </tr>

        <tr>

            <!-- <td></td> -->

            <td colspan="2" class="heading2 text-center pt-3 pb-3">Enter Marks</td>

        </tr>

        <tr>

            <td>1. C language</td>

            <td class="s1 pb-2"><input type="number" id="subject1" /></td>

        </tr>

        <tr>

            <td>2. C++ language</td>

            <td class="s1 pb-2"><input type="number" id="subject2" /></td>

        </tr>

        <tr>

            <td>3. Database</td>

            <td class="s1 pb-2"><input type="number" id="subject3" /></td>

        </tr>

        <tr>

            <td>4. HTML</td>

            <td class="s1 pb-2"><input type="number" id="subject4" /></td>

        </tr>

        <tr>

            <td>5. CSS</td>

            <td class="s1 pb-2"><input type="number" id="subject5" /></td>

        </tr>

        <tr>

            <td>6. php</td>

            <td class="s1 pb-2"><input type="number" id="subject6" /></td>

        </tr>

        <tr>

            <td>7. Core java</td>

            <td class="s1 pb-2"><input type="number" id="subject7" /></td>

        </tr>

        <tr>

            <td></td>

            <td class="s1 pb-2"><button type="button" onclick="calculateResult()">Result</button></td>

        </tr>

        <tr>

            <td id="total"></td>

            <td id="percentage"></td>

        </tr>

    </table>

 <script>

    function calculateResult() {

        // Get subject marks

        var subject1 = parseFloat(document.getElementById('subject1').value) || 0;

        var subject2 = parseFloat(document.getElementById('subject2').value) || 0;

        var subject3 = parseFloat(document.getElementById('subject3').value) || 0;

        var subject4 = parseFloat(document.getElementById('subject4').value) || 0;

        var subject5 = parseFloat(document.getElementById('subject5').value) || 0;

        var subject6 = parseFloat(document.getElementById('subject6').value) || 0;

        var subject7 = parseFloat(document.getElementById('subject7').value) || 0;

        // Calculate total and percentage

        var totalMarks = subject1 + subject2 + subject3 + subject4 + subject5 + subject6 + subject7 ;

        var percentage = (totalMarks / 700) \* 100;

        // Update the result in the HTML

        document.getElementById('total').textContent = 'Total is: ' + totalMarks + '/700';

        document.getElementById('percentage').textContent = 'Percentage is: ' + percentage.toFixed(2) + '%';

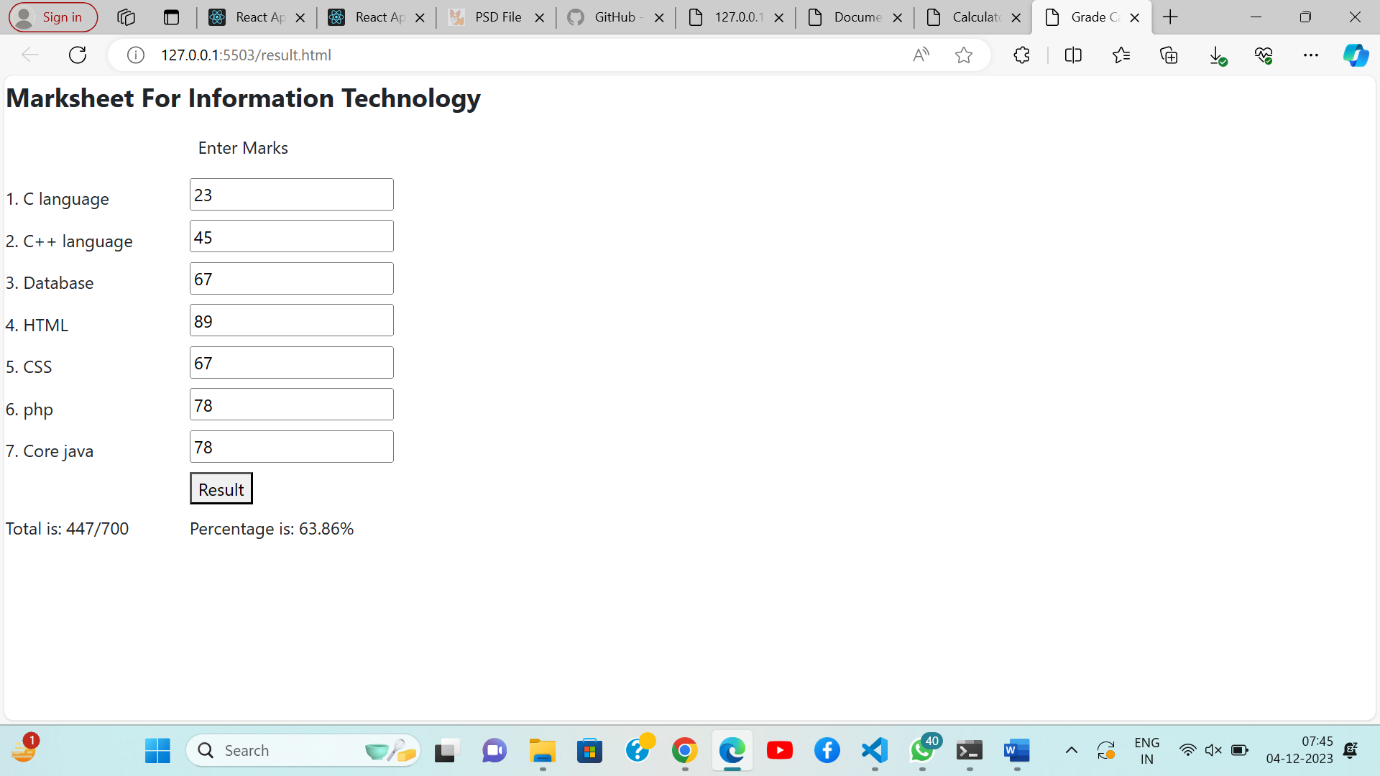
    }

</script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>

</body>

</html>

****

* **Create a slider using JavaScript**

**Ans-**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <style>

    #slider-container {

      position: relative;

      max-width: 1500px; /\* Adjust the width to fit 5 slides \*/

      margin: auto;

      overflow: hidden;

      margin-top: 100px;

    }

    #slider {

      display: flex;

      transition: transform 0.5s ease-in-out;

      justify-content: space-around;

    }

    .large-slide {

      width: 220px; /\* Slightly larger than other slides \*/

      height: 220px; /\* Slightly larger than other slides \*/

    }

    .slide {

      width: 200px;

      box-sizing: border-box;

      height: 200px;

      display: flex;

      align-items: center;

      justify-content: center;

      border-radius: 20px;

    }

    #prev, #next {

      position: absolute;

      top: 50%;

      transform: translateY(-50%);

      font-size: 20px;

      cursor: pointer;

    }

    #prev {

      position: relative;

      margin: 90px 0px 0px 500px;

    }

    #next {

        position: relative;

        margin: -27px 0px 0px 700px;

    }

  </style>

</head>

<body>

<div id="slider-container">

  <div id="slider">

    <div class="slide" style="background-color: #ff9999;">Slide 1</div>

    <div class="slide" style="background-color: #99ff99;">Slide 2</div>

    <div class="slide" style="background-color: #9999ff;">Slide 3</div>

    <div class="slide" style="background-color: #ffcc99;">Slide 4</div>

    <div class="slide" style="background-color: #cc99ff;">Slide 5</div>

  </div>

<div class="arrow">

  <div id="prev" onclick="prevSlide()">&#10094;</div>

  <div id="next" onclick="nextSlide()">&#10095;</div>

</div>

</div>

<script>

    let currentSlide = 0;

    const slides = document.querySelectorAll('.slide');

    const slider = document.getElementById('slider');

    function showSlide(index) {

      const newPosition = -index \* slides[0].offsetWidth;

      slider.style.transform = `translateX(${newPosition}px)`;

    }

    function nextSlide() {

      currentSlide = (currentSlide + 1) % slides.length;

      showSlide(currentSlide);

      updateSlideOrder();

    }

    function prevSlide() {

      currentSlide = (currentSlide - 1 + slides.length) % slides.length;

      showSlide(currentSlide);

      updateSlideOrder();

    }

    function updateSlideOrder() {

      const slidesArray = Array.from(slides);

      const adjustedSlides = slidesArray.slice(currentSlide).concat(slidesArray.slice(0, currentSlide));

      slider.innerHTML = ''; // Clear existing slides

      adjustedSlides.forEach((slide) => slider.appendChild(slide.cloneNode(true)));

    }

    showSlide(currentSlide); // Initialize the display

  </script>

</body>

</html>

****