**JAVASCRIPT BASIC & DOM**

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

Ans.1 JavaScript is a scripting language used by developers to create and control dynamic website content.

<script>

document.getElementById("demo").innerHTML = "Hello, JavaScript!";

</script>

Q.2 How many types of Variables in JavaScript?

Ans.2 There are three types of variables which are **var, let, const,**

Q.3 Define a Data Types in js?

Ans.3 JavaScript has several data types:

**Number:** For both integers and floating-point numbers.

**String:** For a sequence of characters.

**Boolean:** For true or false.

**Undefined:** For a variable that has been declared but has not yet been assigned a value.

**Null:** For an absence of any value or object.

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

Ans.4 ‘**mul’** function that multiplies any number of arguments passed to it.

console.log(mul(4, 6)); (output = 24)

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

Ans.5 **Undefined**: A variable is undefined when it has been declared but has not been assigned a value. It is a type and a value in JavaScript.

let x;

console.log(x);

**Undeclared**: A variable is **undeclared** when it has not been declared using any of the variable declaration keywords (**var**, **let**, **const**).

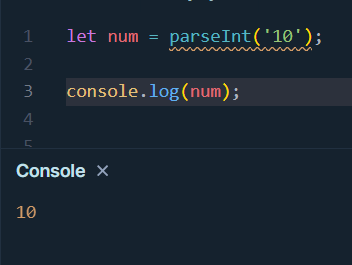
console.log(y);

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:

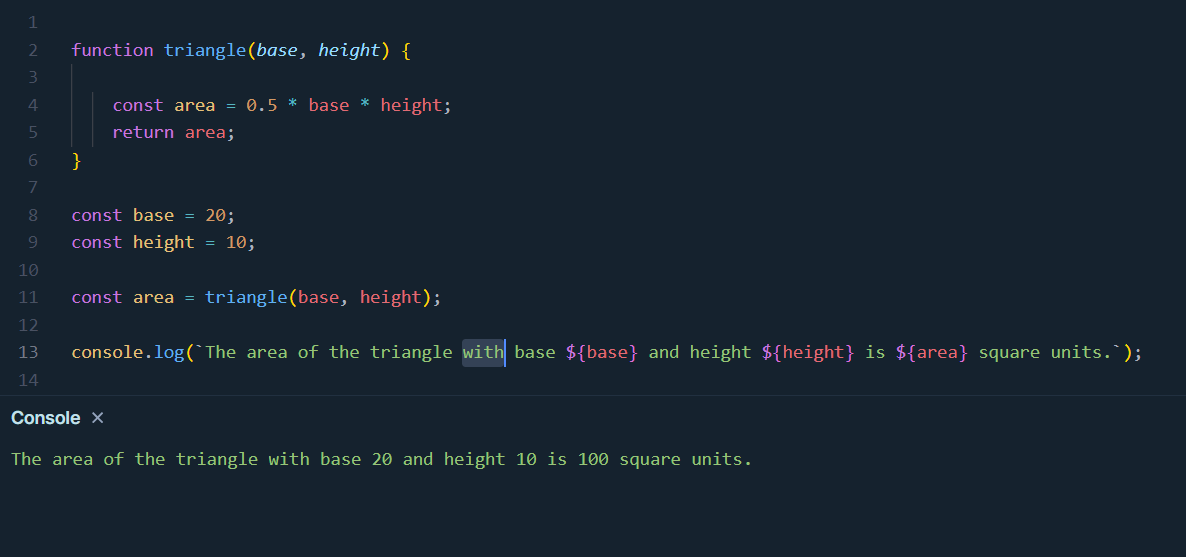
Ans.6 console.log("The quote 'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes teaches us to help one another.");

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

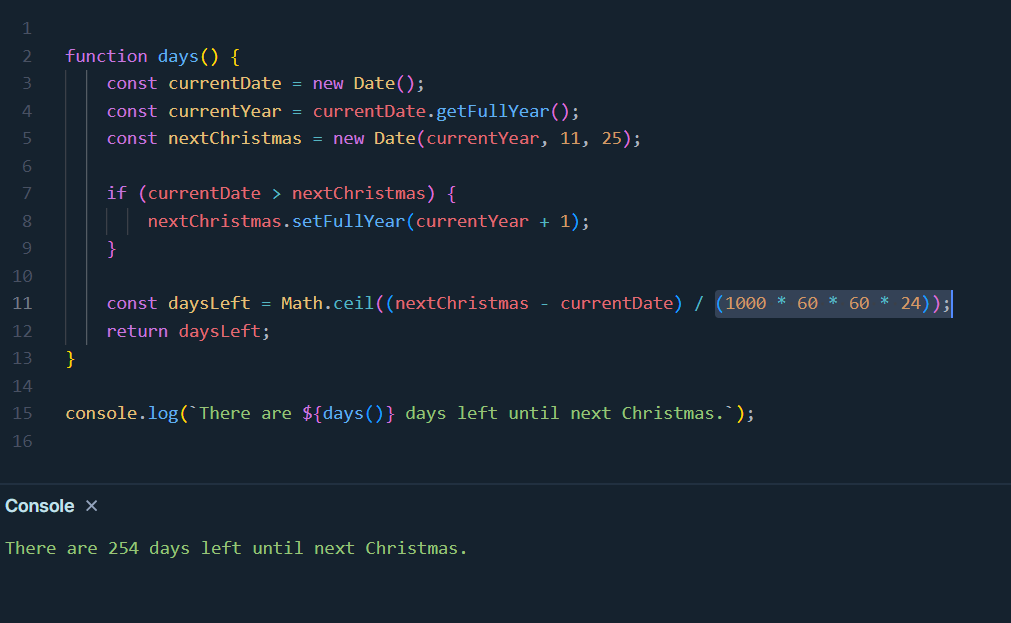
Q.7 Check if typeof '10' is exactly equal to 10. If not make it exactly equal?

Ans.7 

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



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



Q.10 What is Condition Statement?

Ans.10 In JavaScript, there are mainly three types of conditional statements:

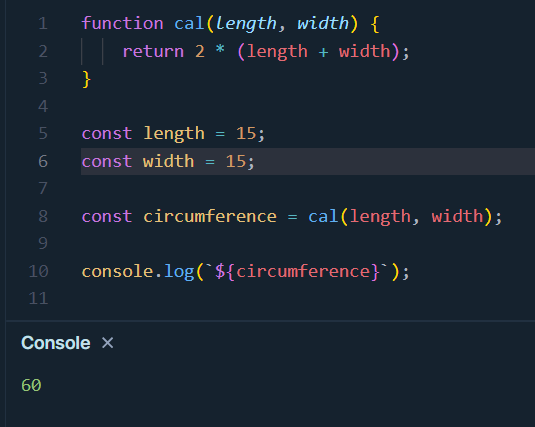
If and else statement: It execute if condition is true or false

Switch statement: It execute days, weeks etc

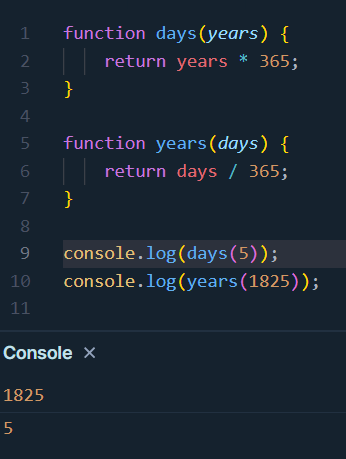
For loop: for (let i = 0; i < 5; console.log(i++));

While loop: let i = 0; while (i < 5) console.log(i++);

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

Ans.11 

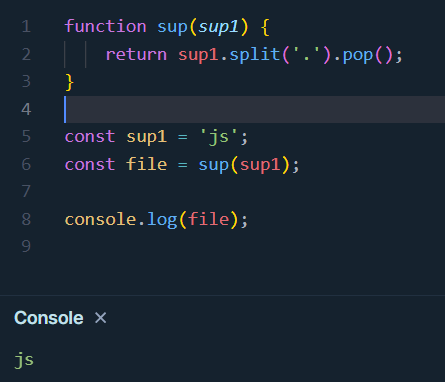
Q.12 WAP to convert years into days and days into years?

Ans.12 

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

Ans.13 

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

Ans.14 

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

Ans.15 The result is **true** because both conditions **5 > 3** and **2 < 4** are true, and **&&** (AND) requires both conditions to be true for the entire expression to be true.

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

Ans.16 The result is **"hello"** because all values are truthy, and **&&** (AND) returns the last value if all values are truthy.

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

Ans.17 The result is **false** because **&&** (AND) has higher precedence than **||** (OR), so it evaluates **true && false** first, which is **false**, then **false && true**, which is also **false**. Finally, **false || false** evaluates to **false**

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

Ans.18 **Loop**: Execute a set of instructions or a block of code a certain number of times without having to write it again until a certain condition is met. Types of loops in JavaScript are **for**, **while**, and **do...while** loops.

**Switch Case**: Switch statements are always evaluated from top to bottom. The order of cases does not matter as long as you always have the reserved word “break” at the end of all cases that shouldn't have fall-through.

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

Ans.19 The **isNaN()** function in JavaScript is used to determine whether a value is **NaN** (Not-a-Number) or not. isNaN() converts the value to a number before testing it. It returns **true** if the value is **NaN**, otherwise it returns **false**.

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

Ans**.**20 **&&** (AND): Returns **true** if both operands are truthy; otherwise, returns **false**.

**||** (OR): Returns **true** if at least one of the operands is truthy; returns **false** if both operands are falsy.

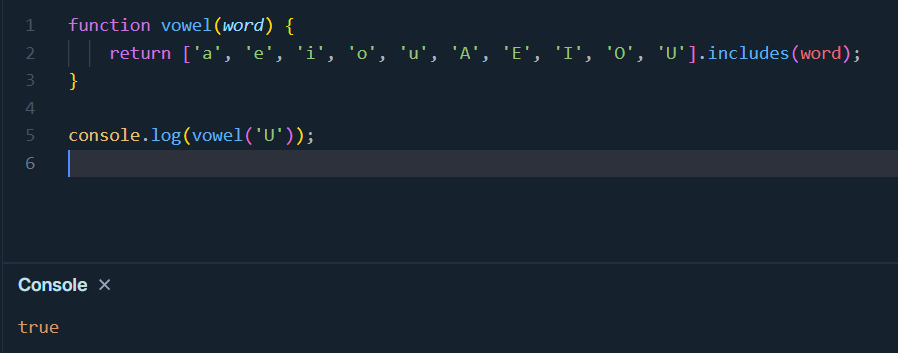
Q.21 What is the use of Void (0)?

Ans**.**21The void operator evaluates an expression and returns undefined. By running void (0) in the URL JavaScript code, nothing is evaluated or returned.

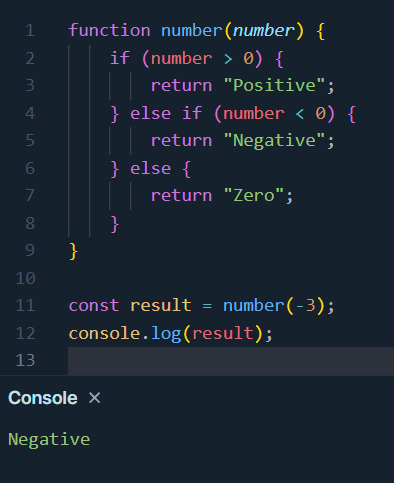
Q.22 Check Number Is Positive or Negative in JavaScript?

Ans**.**22 

Q.23 Find the Character Is Vowel or Not?

Ans**.**23 

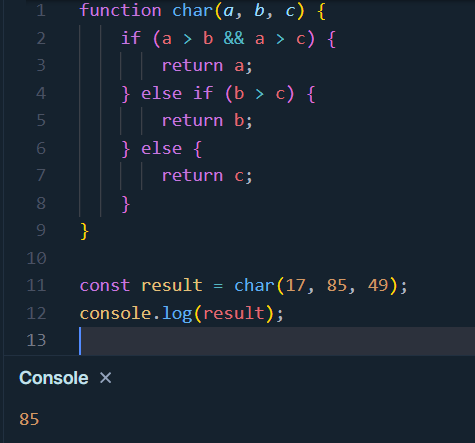
Q.24 Write to check whether a number is negative, positive or zero?

Ans**.**24 

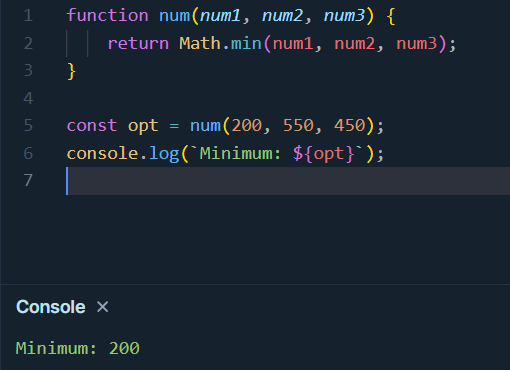
Q.25 Write to find number is even or odd using ternary operator in JS?

Ans**.**25 

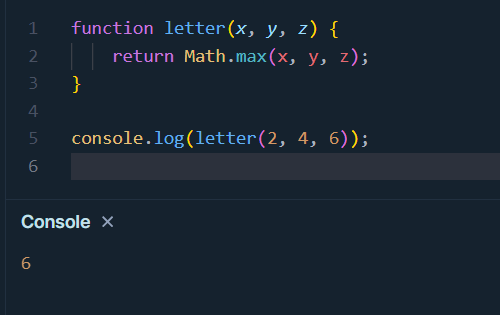
Q.26 Write find maximum number among 3 numbers using ternary operator in JS?

Ans**.**26 

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

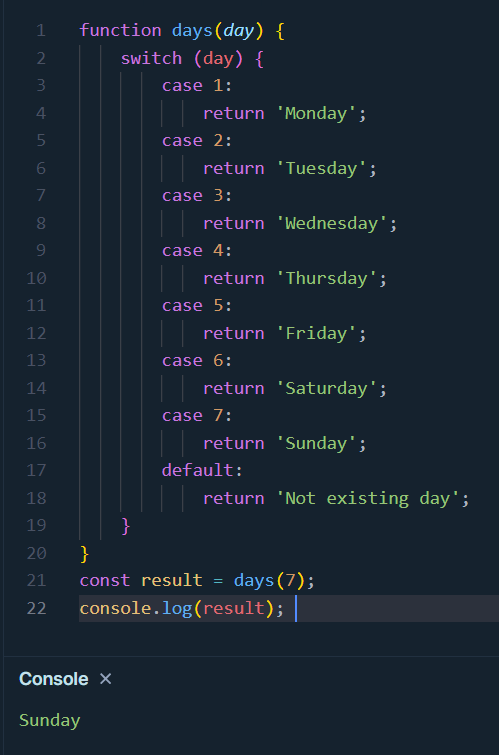
Ans.27 

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

Ans**.**28 

Q.29 Write to show

i. Monday to Sunday using switch case in JS?



ii. Vowel or Consonant using switch case in JS?



**(Conditional looping logic Question)**

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

Ans.30 JavaScript has several looping structures: -

1. **for** loop
2. **while** loop
3. **do...while** loop

For Example: -

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

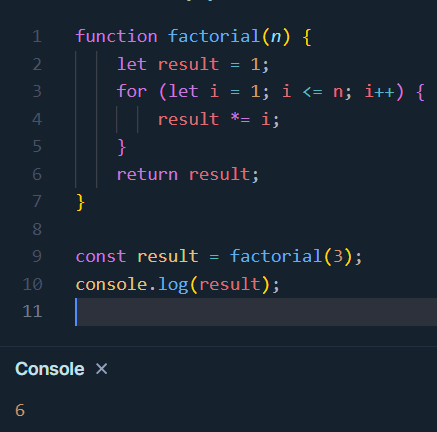
console.log(i);

}

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

Ans.31 

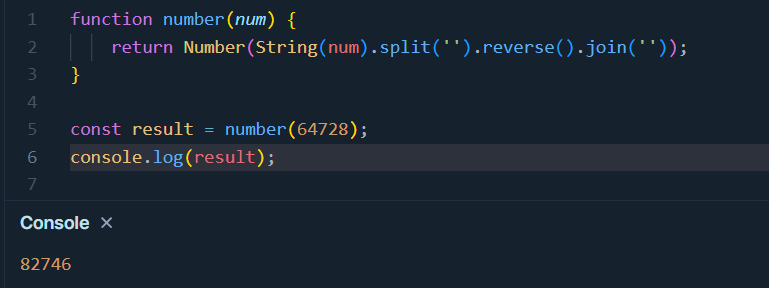
Q.32 Write to print factorial of given number?

Ans.32 

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

Ans.33 

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

Ans.34 

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.37 

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



3) 1

2 3

4 5 6

7 8 9 10

11 12 13 14 15



4) \*

\* \*

\* \* \*

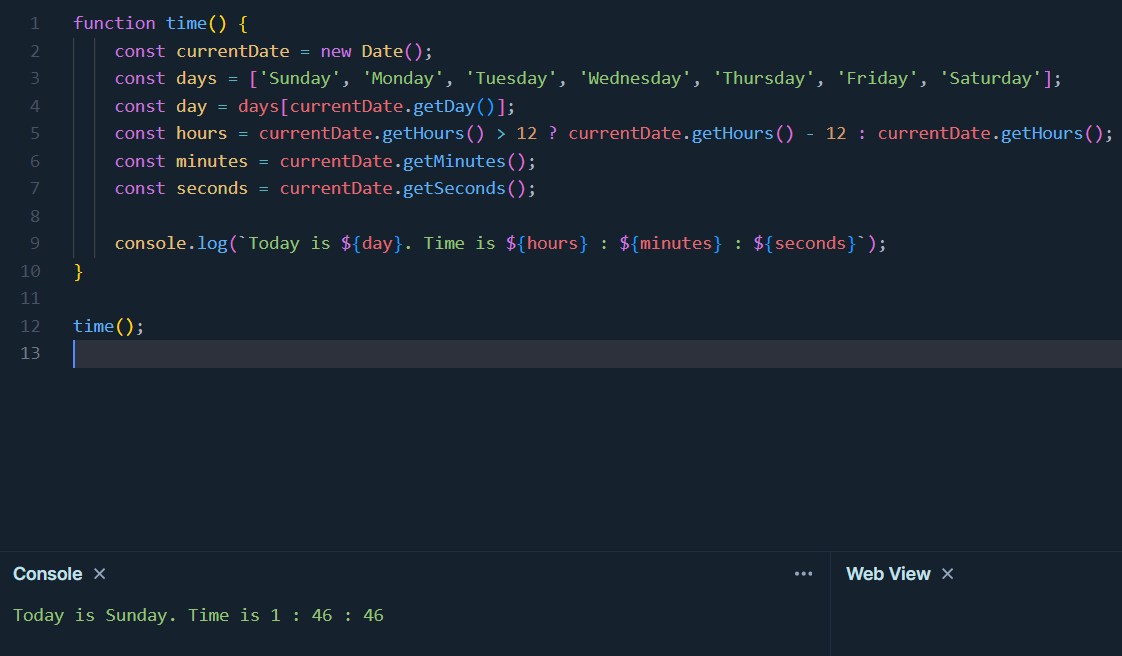
\* \* \* \*

\* \* \* \* \*

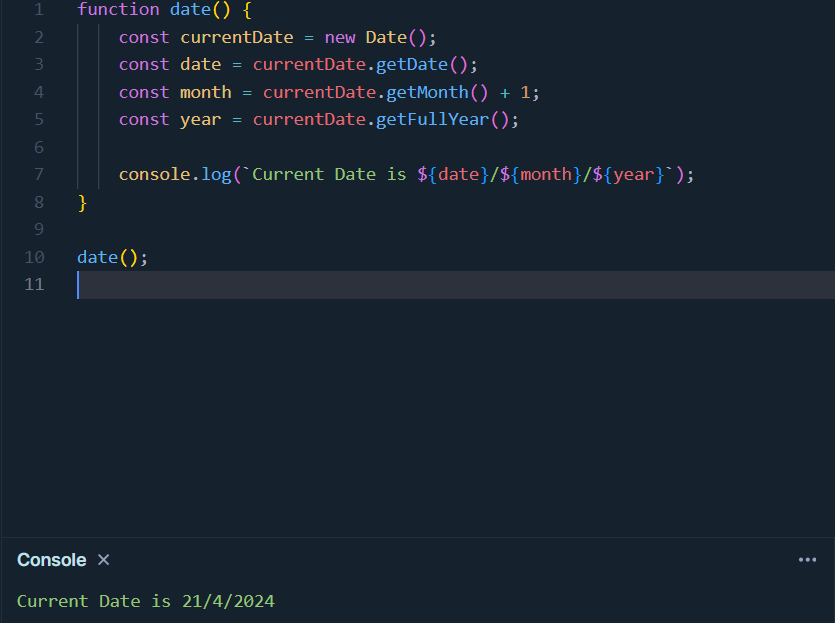


**(Array and object Question)**

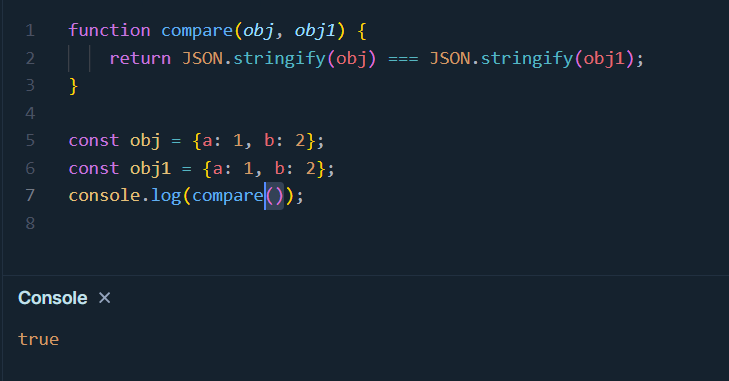
Q.40 Write a JavaScript Program to display the current day and time in the following format. Sample Output: Today is Friday. Current Time is 12 PM: 12 : 22 2 ?

Ans.40 

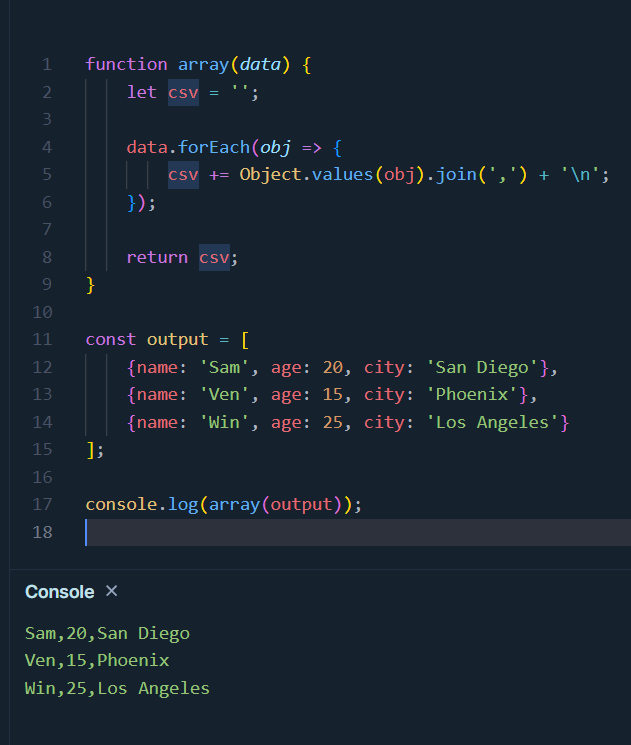
Q.41 Write a JavaScript program to get the current date?

Ans.41 

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

Ans.42 

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

Ans.43 

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

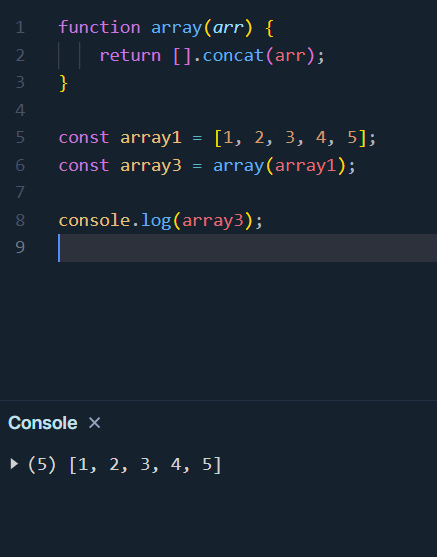
Ans.44 

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

Ans.45 

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

Ans.46 **Using the concat() method**



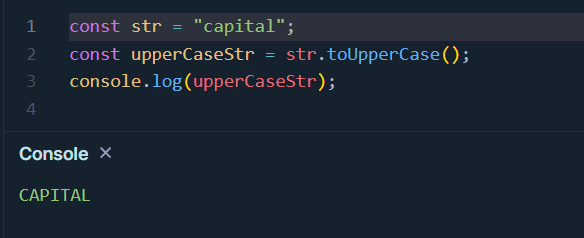
Q.47 What is the drawback of declaring methods directly in JavaScript objects?

Ans.47 Declaring methods directly in JavaScript objects can lead to code duplication and decreased maintainability. If the same method needs to be used across multiple objects, each object will have its own copy of the method. This can make it harder to update the method in the future since changes would need to be made to each object individually.

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

Ans.48 

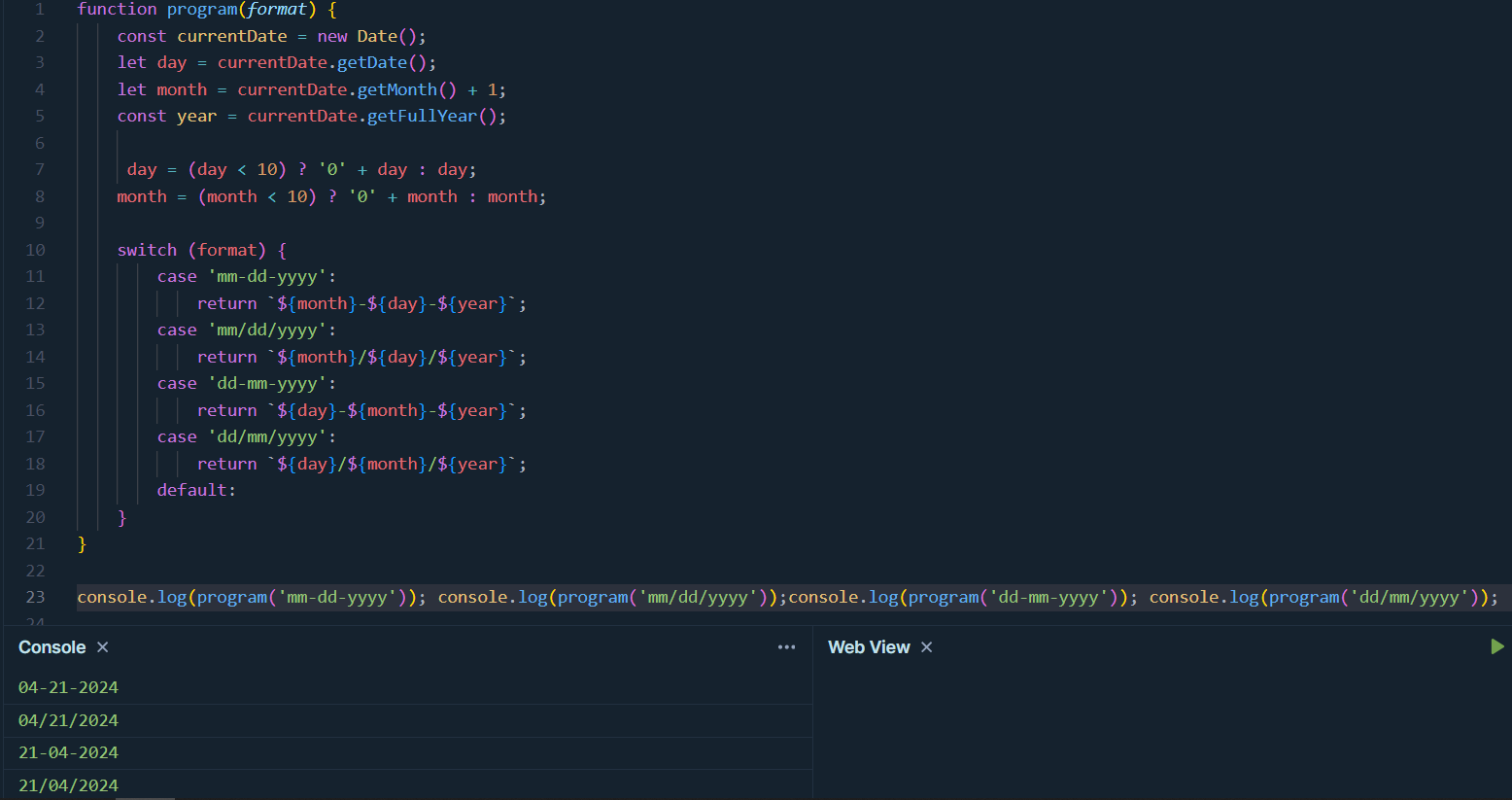
Q.49 Change all the string characters to capital letters using toUpperCase() method?

Ans.49 

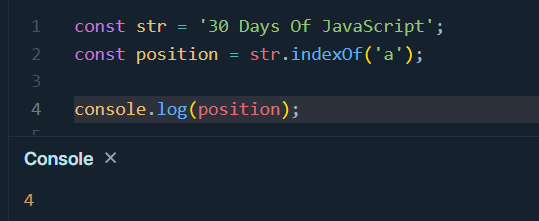
Q.50 What is the drawback of declaring methods directly in JavaScript objects?

Ans.50 In JavaScript objects drawback of declaring methods directly in JavaScript objects can lead to increased memory usage, decreased code readability, and issues with inheritance and maintainability.

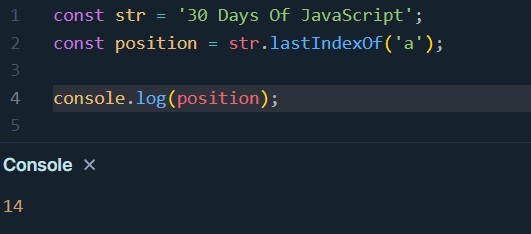
Q.51 Write a JavaScript program to get the current date. Expected Output : mm-dd-yyyy, mm/dd/yyyy or dd-mm-yyyy, dd/mm/yyyy?

Ans.51 

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

Ans.52 

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

Ans.53 

Q.54 Form Validtion in JS?

Ans.54 <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Document</title>

</head>

<body>

    <form method="post" onsubmit="return ValidData(input, msg)">

        <table border="1" align="center">

            <tr>

                <td>Name:</td>

                <td><input type="text" name="name" id="name" onblur="ValidData(this,'nameerror')" onkeydown="NameValid(this,'nerror')">

                    <span id="nameerror"></span>

                    <span id="nerror"></span>

                </td>

            </tr>

            <tr>

                <td>Email:</td>

                <td><input type="email" name="email" id="email" onblur="ValidData(this,'emailerror')" onkeydown="EmailValid(this,'eerror')">

                    <span id="emailerror"></span>

                    <span id="eerror"></span>

                </td>

            </tr>

            <tr>

                <td>Password:</td>

                <td><input type="password" name="pwd" id="pwd" onblur="ValidData(this,'pwderror')" onkeydown="PwdValid(this,'perror')">

                    <span id="pwderror"></span>

                    <span id="perror"></span>

                </td>

            </tr>

            <tr>

                <td>Mobile:</td>

                <td><input type="text" name="mobile" id="mobile" onblur="ValidData(this,'mobileerror')" onkeydown="MobileValid(this,'merror')">

                    <span id="mobileerror"></span>

                    <span id="merror"></span>

                </td>

            </tr>

            <tr>

                <td colspan="2" align="center"><input type="submit" name="submit" value="submit"></td>

            </tr>

        </table>

    </form>

</body>

</html>

<script>

    function ValidData(input, msg) {

        if (input.value == " ") {

            document.getElementById(msg).style.color = "red";

            document.getElementById(msg).innerText = "Please ENter Data!";

            return false;

        }

        else {

            document.getElementById(msg).innerText = "";

            return true;

        }

    }

    function NameValid(input, msg) {

        var name = /^[a-zA-Z]+$/

        if (name.test(input.value)) {

            document.getElementById(msg).innerText = "";

            return true;

        }

        else {

            document.getElementById(msg).style.color = "red";

            document.getElementById(msg).innerText = "Please ENter only Characters!";

            return false;

        }

    }

    function EmailValid(input, msg) {

        var email = /^([a-z0-9.-\_])+@+([a-z0-9-.\_]).([a-z]{3})/

        if (email.test(input.value)) {

            document.getElementById(msg).innerText = "";

            return true;

        }

        else {

            document.getElementById(msg).style.color = "red";

            document.getElementById(msg).innerText = "Please ENter valid Email!";

            return false;

        }

    }

    function PwdValid(input, msg) {

        var pwd = /^[a-zA-Z0-9.-]{3,5}$/

        if (pwd.test(input.value)) {

            document.getElementById(msg).innerText = "";

            return true;

        }

        else {

            document.getElementById(msg).style.color = "red";

            document.getElementById(msg).innerText = "Please ENter min 3 and max 5 characters!";

            return false;

        }

    }

    function MobileValid(input, msg) {

        var mobile = /^[0-9]{10}$/

        if (mobile.test(input.value)) {

            document.getElementById(msg).innerText = "";

            return true;

        }

        else {

            document.getElementById(msg).style.color = "red";

            document.getElementById(msg).innerText = "Please ENter valid mobile number!";

            return false;

        }

    }

</script>

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

Ans.55 <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Document</title>

</head>

<body>

    <form method="post" onsubmit="return validateForm()">

        <table border="1" align="center">

            <tr>

                <td>Name:</td>

                <td><input type="text" name="name" id="name" onblur="validateInput(this,'nameerror')" required>

                    <span id="nameerror" style="color: red;"></span>

                </td>

            </tr>

            <tr>

                <td>Email:</td>

                <td><input type="email" name="email" id="email" onblur="validateInput(this,'emailerror')" required>

                    <span id="emailerror" style="color: red;"></span>

                </td>

            </tr>

            <tr>

                <td>Password:</td>

                <td><input type="password" name="pwd" id="pwd" onblur="validateInput(this,'pwderror')" required>

                    <span id="pwderror" style="color: red;"></span>

                </td>

            </tr>

            <tr>

                <td>Mobile:</td>

                <td><input type="text" name="mobile" id="mobile" onblur="validateInput(this,'mobileerror')" required>

                    <span id="mobileerror" style="color: red;"></span>

                </td>

            </tr>

            <tr>

                <td colspan="2" align="center"><input type="submit" name="submit" value="submit"></td>

            </tr>

        </table>

    </form>

    <script>

        function validateInput(input, msg) {

            if (!input.value.trim()) {

                document.getElementById(msg).innerText = "Please enter data!";

                return false;

            } else {

                document.getElementById(msg).innerText = "";

                return true;

            }

        }

        function validateForm() {

            return validateInput(document.getElementById("name"), 'nameerror'),

                   validateInput(document.getElementById("email"), 'emailerror'),

                   validateInput(document.getElementById("pwd"), 'pwderror'),

                   validateInput(document.getElementById("mobile"), 'mobileerror');

        }

    </script>

</body>

</html>

Q.56 Dynamic Form Validation in JS?

Ans.56 <!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Document</title>

  <style>

    .msg {

      color: red;

    }

  </style>

</head>

<body>

  <form id="form">

    <div>

      <label for="name">Name:</label>

      <input type="text" id="name" name="name">

      <span id="nerror" class="msg"></span>

    </div>

    <div>

      <label for="pwd">Password:</label>

      <input type="password" id="pwd" name="pwd">

      <span id="perror" class="msg"></span>

    </div>

    <div>

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

    </div>

  </form>

  <script>

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

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

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

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

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

    form.addEventListener('submit', function (event) {

      let isValid = true;

      // For name

      const nameValue = name.value.trim();

      if (nameValue === '') {

        nerror.text = 'Required!';

        isValid = false;

      } else {

        nerror.text = '';

      }

      // For password

      const passwordValue = pwd.value.trim();

      if (passwordValue === '') {

        perror.text = 'Required!';

        isValid = false;

      } else {

        perror.text = '';

      }

      if (!isValid) {

        event.preventDefault();

      }

    });

  </script>

</body>

</html>

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

Ans.57 There are various types of events in JavaScript, such as **click**, **mouseover**, **keydown**, **submit**, etc.

<button id="myButton">Click Me</button>

<script>

    document.getElementById('myButton').addEventListener('click', function() {

        alert('Button clicked!');

    });

</script>

Q.59 What is Bom vs Dom in JS?

Ans.59 **BOM (Browser Object Model)**: It provides objects and methods to interact with the browser. It includes objects like **window**, **navigator**, **screen**, **history**, and **location**.

**DOM (Document Object Model)**: It represents the document as a tree of objects. It provides a structured representation of the document, enabling programs to modify document structure, style, and content.

Q.60 Array vs object defences in JS?

Ans.60 **Array**: An array is an ordered collection of values, accessed by indices.

**Object:** It's best suited for storing structured data.

Q.61 Split the string into an array using split() Method?

Ans.61 const str = "Hello,World";

const arr = str.split(",");

console.log(arr);

Q.62 Check if the string contains a word Script using includes() method?

Ans.62 const str = "JavaScript";

const check = str.includes("Script");

console.log(check);

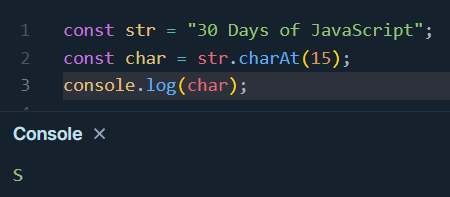
Q.63 Change all the string characters to lowercase letters using toLowerCase() Method.

Ans.63 const str = "JAVASCRIPT";

const Case = str.toLowerCase();

console.log(Case);

Q.64 What is Character at index 15 in ’30 Days of JavaScript’ string? Use charAt() method.

Ans.64 

Q.65 copy to one string to another string in JS?

Ans.65 const str1 = "JavaScript";

const str2 = str1;

console.log(str2);

• What is JavaScript?

JavaScript is a scripting language used by developers to create and control dynamic website content. JavaScript is a high-level, interpreted programming language primarily used for web development to add interactivity and dynamic content to websites.

• What is the use of isNaN function?

=> The **isNaN** function in JavaScript is used to determine whether a value is NaN (Not-a-Number)

• What is negative Infinity?

=> Negative Infinity represents the smallest negative number in JavaScript, indicating a value that is lower than any other number.

• Which company developed JavaScript?

=> JavaScript was developed by Netscape Communications Corporation. It was originally called "LiveScript" but was later renamed to "JavaScript."

• What are undeclared and undefined variables?

=> Undeclared variables are those that have been used in code but have not been declared using the var, let, or const keywords. They are typically global and can lead to unexpected behavior. Undefined variables are those that have been declared but have not been assigned a value. They have the special value undefined.

• Write the code for adding new elements dynamically?

=> To add new elements dynamically in the DOM using JavaScript, you can use methods like createElement, appendChild, and manipulate the DOM as follows:

Example of Create a new element

var newElement = document.createElement("div");

• What is the difference between ViewState and SessionState?

=> **ViewState** is used to maintain the state of the server-side controls on a single web page across postbacks. It is lightweight but can contribute to the page size.

**SessionState** is used to store user-specific data across multiple pages within the same web application. It is stored on the server and has a defined lifetime.

• What is === operator?

=> The **===** operator in JavaScript is a strict equality operator. It checks both the value and the type of the operands, ensuring they are identical.

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

=> You can change the style or class of an element in JavaScript using the **style** property or **classList** property, respectively.

• How to read and write a file using JavaScript?

=> You can read and write a file using JavaScript with the **File API**.

• What are all the looping structures in JavaScript?

=> The looping structures in JavaScript are **for**, **while**, **do-while**, and **for...in**

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

=> You can convert a string of any base to an integer using **parseInt()** with a radix parameter specifying the base.

• What is the function of the delete operator?

=> The **delete** operator in JavaScript is used to remove a property from an object

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

=> The types of pop-up boxes available in JavaScript are **alert**, **confirm**, and **prompt**

• What is the use of Void (0)?

=> **void(0)** is used to prevent the browser from loading a new page when clicked.

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

=> To force a page to load another page in JavaScript, you can use **window.location.href** to set the URL.

• What are the disadvantages of using innerHTML in JavaScript?

=> The main disadvantages of using `innerHTML` in JavaScript are potential security risks from cross-site scripting (XSS) attacks and performance issues due to re-parsing and re-rendering the entire HTML content.

• Create password field with show hide functionalities

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Create Password</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>

**Q** **Create basic math operation in JS:**

<!DOCTYPE html>

<html>

<head>

    <title>Basic Math Operation</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:**

<!DOCTYPE html>

<html>

<head>

    <title>Create result</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**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Slider</title>

  <style>

    #slider-container {

      position: relative;

      max-width: 2000px;

      margin: auto;

      overflow: hidden;

      margin-top: 50px;

      background-color: black;

    }

    #slider {

      display: flex;

      transition: transform 0.5s ease-in-out;

      justify-content: space-around;

    }

    .slide {

      width: 250px;

      box-sizing: border-box;

      height: 250px;

      display: flex;

      align-items: center;

      justify-content: center;

    }

    #prev, #next {

      position: absolute;

      top: 50%;

      font-size: 20px;

      cursor: pointer;

      color: white;

    }

    #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: #ea1818;"></div>

    <div class="slide" style="background-color: #0aef0a;"></div>

    <div class="slide" style="background-color: #3e3ef3;"></div>

    <div class="slide" style="background-color: #bb7a3a;"></div>

    <div class="slide" style="background-color: #562685;"></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 = '';

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

  }

  showSlide(currentSlide);

</script>

</body>

</html>

• What is JavaScript?

JavaScript is a scripting language used by developers to create and control dynamic website content. JavaScript is a high-level, interpreted programming language primarily used for web development to add interactivity and dynamic content to websites.

• What is the use of isNaN function?

=> The **isNaN** function in JavaScript is used to determine whether a value is NaN (Not-a-Number)

• What is negative Infinity?

=> Negative Infinity represents the smallest negative number in JavaScript, indicating a value that is lower than any other number.

• Which company developed JavaScript?

=> JavaScript was developed by Netscape Communications Corporation. It was originally called "LiveScript" but was later renamed to "JavaScript."

• What are undeclared and undefined variables?

=> Undeclared variables are those that have been used in code but have not been declared using the var, let, or const keywords. They are typically global and can lead to unexpected behavior. Undefined variables are those that have been declared but have not been assigned a value. They have the special value undefined.

• Write the code for adding new elements dynamically?

=> To add new elements dynamically in the DOM using JavaScript, you can use methods like createElement, appendChild, and manipulate the DOM as follows:

Example of Create a new element

var newElement = document.createElement("div");

• What is the difference between ViewState and SessionState?

=> **ViewState** is used to maintain the state of the server-side controls on a single web page across postbacks. It is lightweight but can contribute to the page size.

**SessionState** is used to store user-specific data across multiple pages within the same web application. It is stored on the server and has a defined lifetime.

• What is === operator?

=> The **===** operator in JavaScript is a strict equality operator. It checks both the value and the type of the operands, ensuring they are identical.

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

=> You can change the style or class of an element in JavaScript using the **style** property or **classList** property, respectively.

• How to read and write a file using JavaScript?

=> You can read and write a file using JavaScript with the **File API**.

• What are all the looping structures in JavaScript?

=> The looping structures in JavaScript are **for**, **while**, **do-while**, and **for...in**

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

=> You can convert a string of any base to an integer using **parseInt()** with a radix parameter specifying the base.

• What is the function of the delete operator?

=> The **delete** operator in JavaScript is used to remove a property from an object

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

=> The types of pop-up boxes available in JavaScript are **alert**, **confirm**, and **prompt**

• What is the use of Void (0)?

=> **void(0)** is used to prevent the browser from loading a new page when clicked.

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

=> To force a page to load another page in JavaScript, you can use **window.location.href** to set the URL.

• What are the disadvantages of using innerHTML in JavaScript?

=> The main disadvantages of using `innerHTML` in JavaScript are potential security risks from cross-site scripting (XSS) attacks and performance issues due to re-parsing and re-rendering the entire HTML content.