**Module 6) JAVASCRIPT BASIC & DOM**

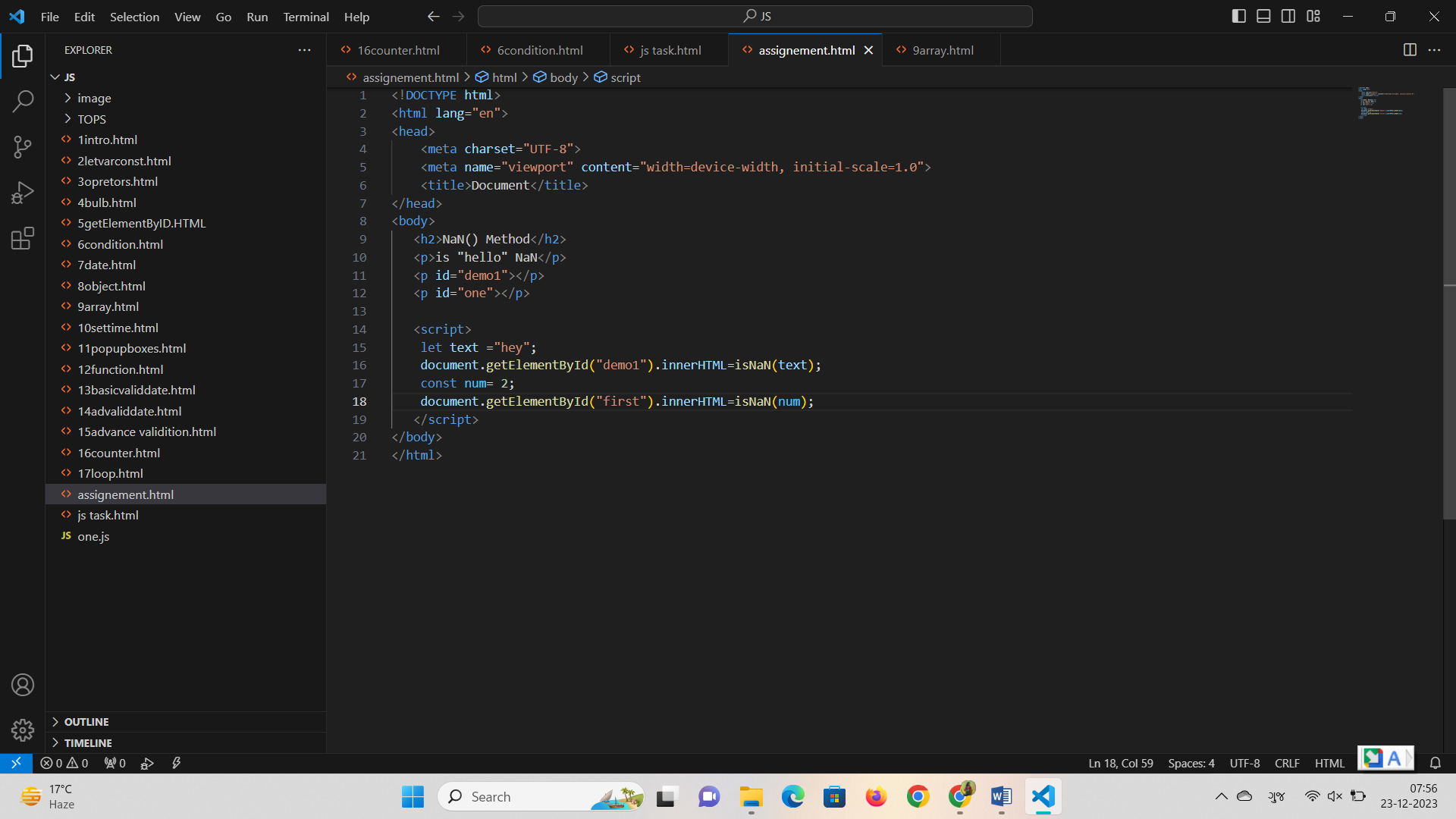
**Q-1 What is JavaScript?**

**Ans-** JavaScript is a versatile and widely used programming language for creating interactive and dynamic web applications. It is commonly used for client-side scripting in web development, allowing developers to add interactivity, manipulate the Document Object Model (DOM), and perform various tasks within web browsers.

**Q-2 What is the use of isNaN function?**

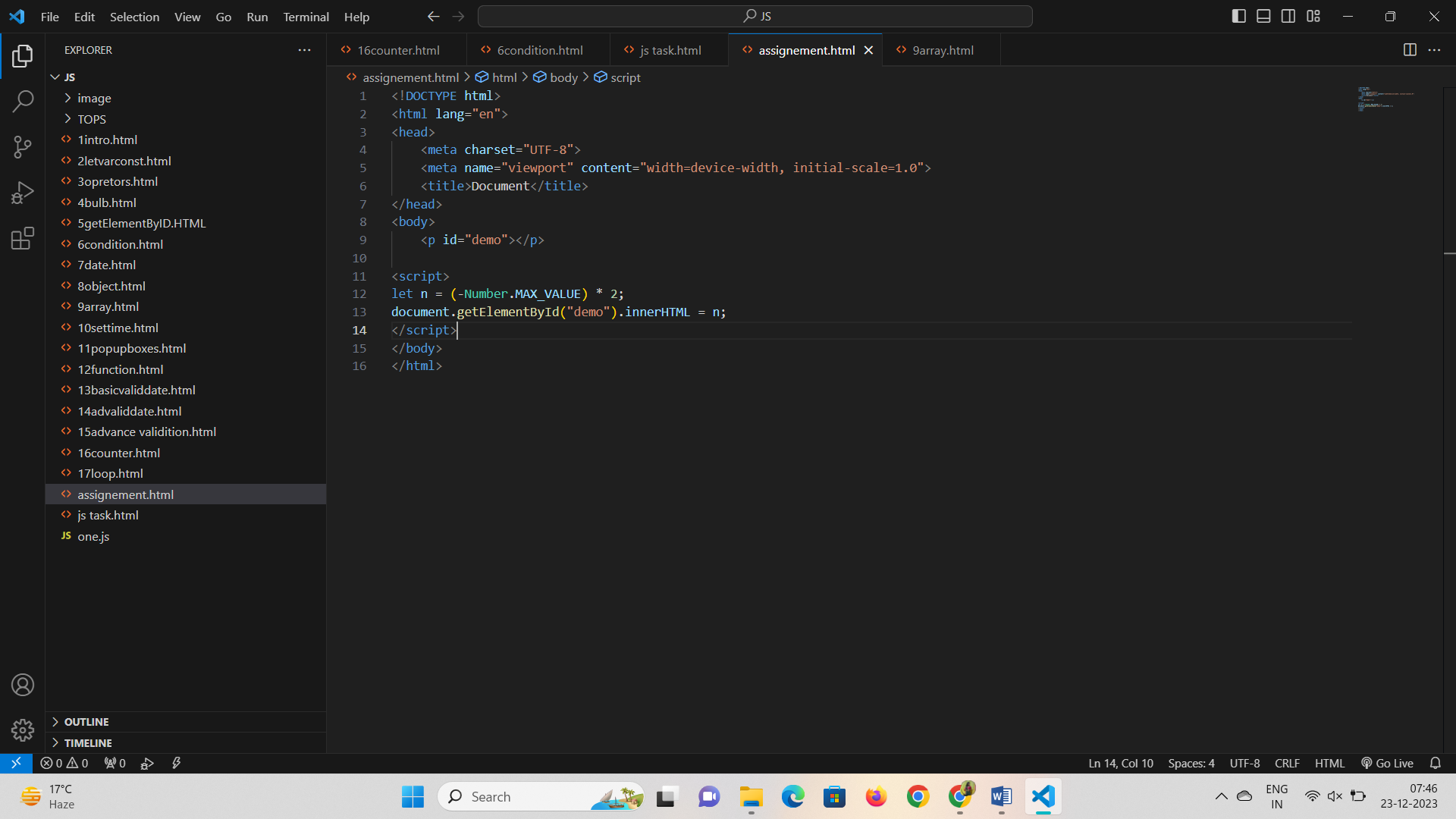
**Ans-** The isNaN function in JavaScript is used to determine whether a value is "Not-a-Number" (NaN) or not. It returns true if the provided value is not a valid number and false if it is a valid number or can be coerced into one.

Ex:



**Q-3 What is negative Infinity?**

**Ans-** Negative Infinity is a special value in JavaScript, representing a number that is smaller (more negative) than the smallest representable number. It is often the result of mathematical operations that approach negative infinity or underflow.



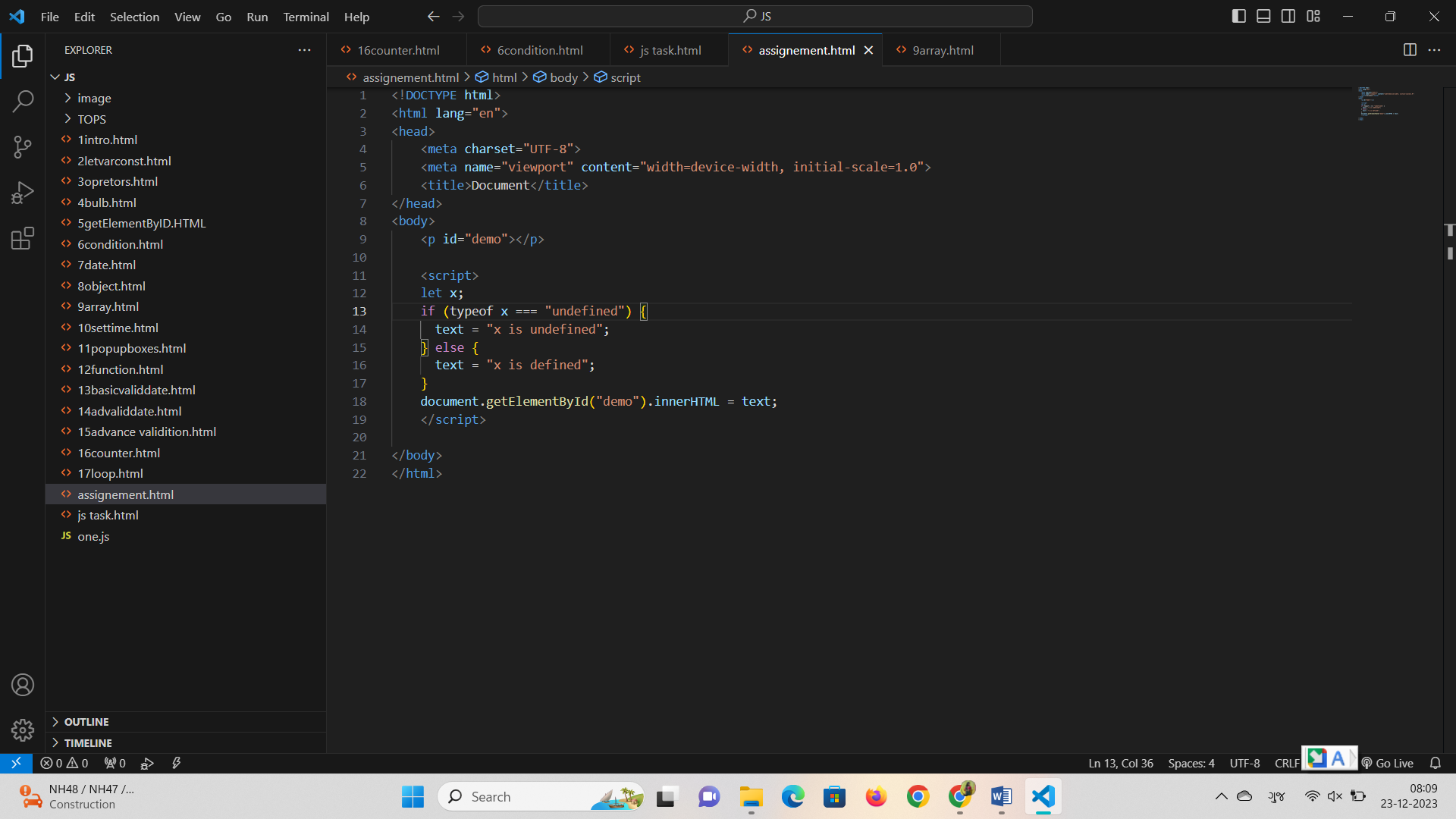
Ex:

**Q-4 Which company developed JavaScript?**

**Ans-** JavaScript was developed by Netscape Communications Corporation. It was originally called "LiveScript" but was later renamed to "JavaScript."

**Q-5 What are undeclared and undefined variables?**

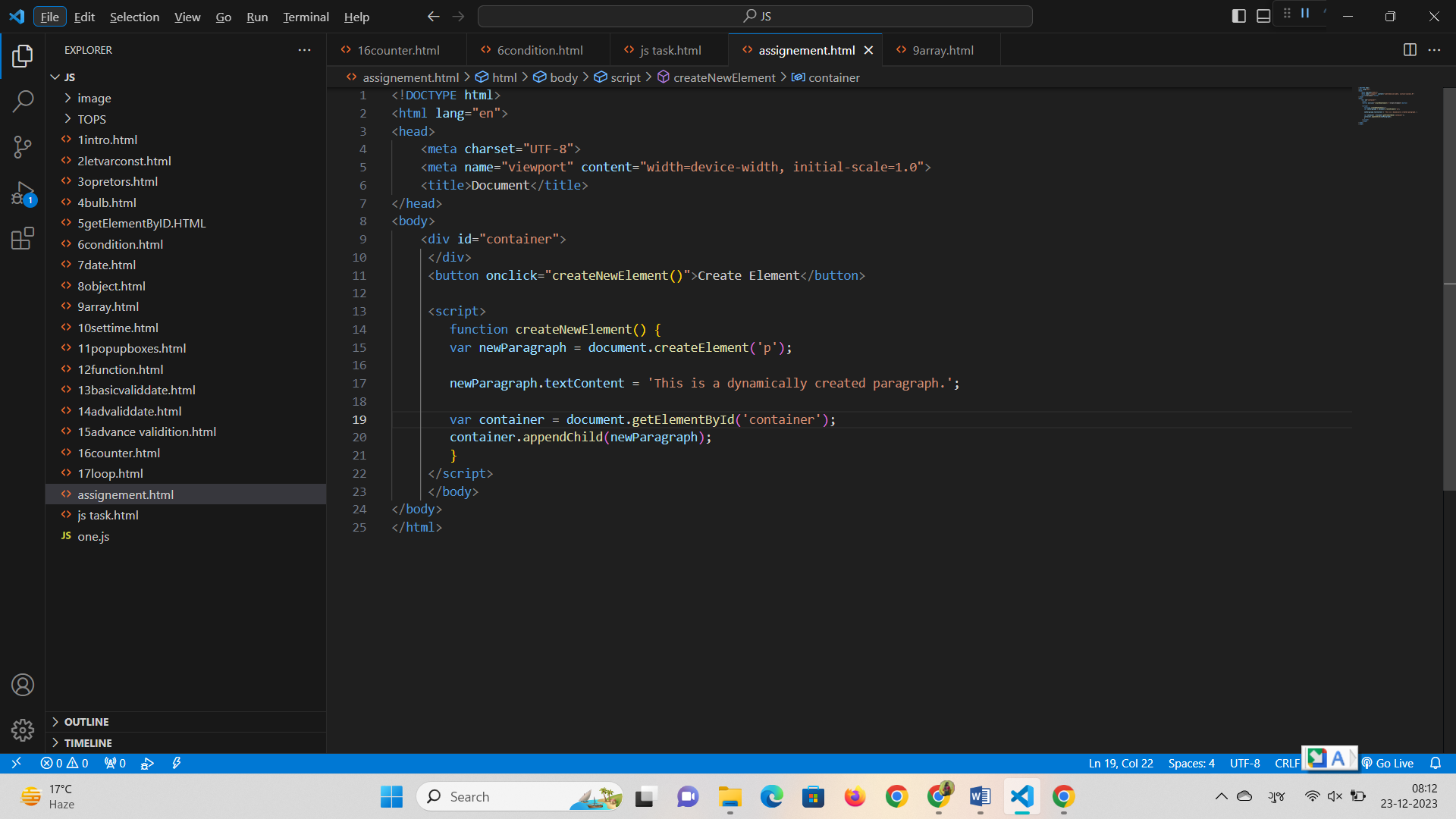
**Ans-** Undeclared variables are those that have been used in code but have not been declared using the var, let, or const keywords. They are typically global and can lead to unexpected behavior.



Undefined variables are those that have been declared but have not been assigned a value. They have the special value undefined.

**Q-6 Write the code for adding new elements dynamically?**

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



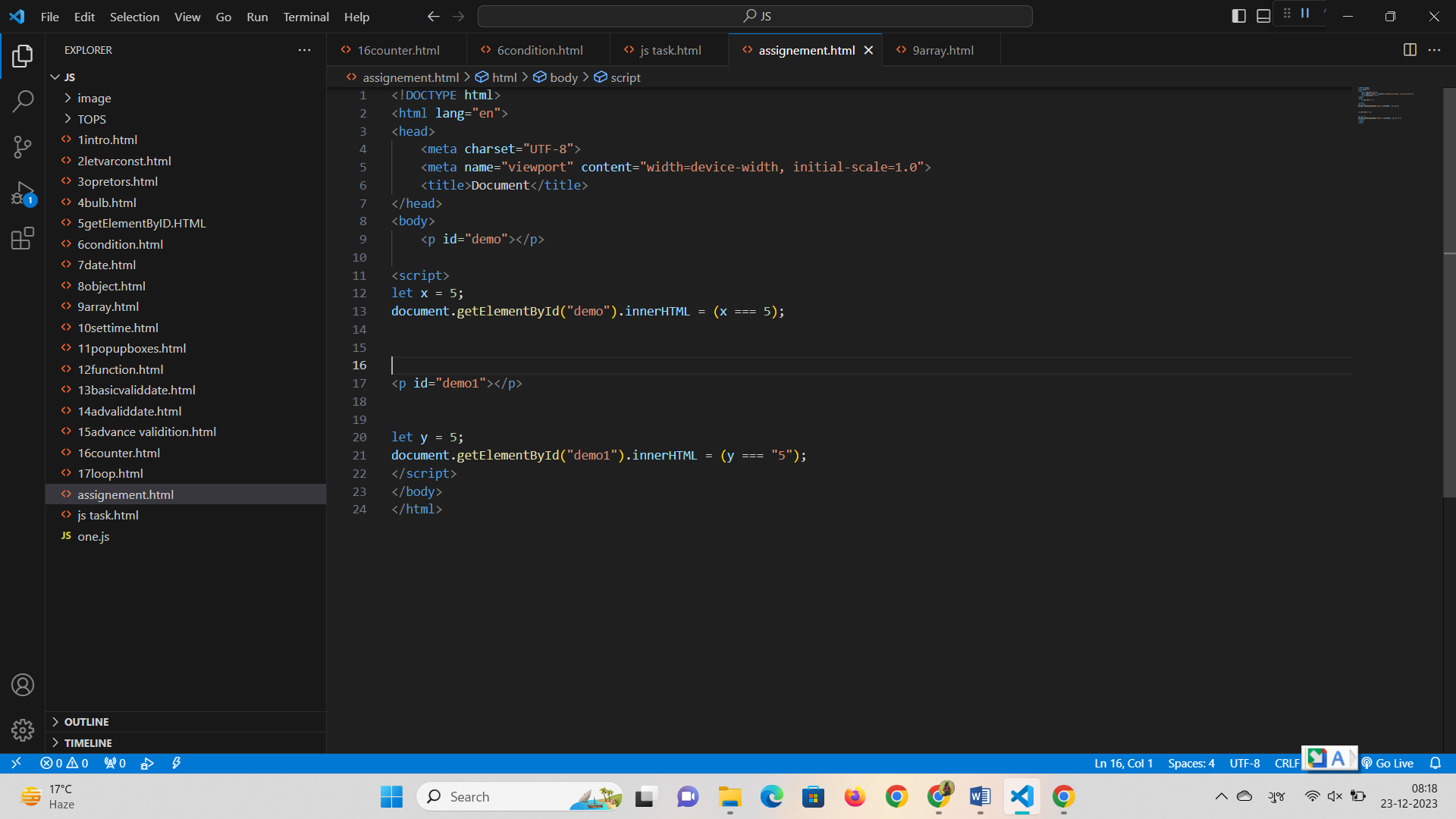
**Q-7 What is the difference between ViewState and SessionState?**

**Ans**

| **ViewState** | **SessionState** |
| --- | --- |
| Maintained at page level only. | Maintained at session level. |
| View state can only be visible from a single page and not multiple pages. | Session state value availability is across all pages available in a user session. |
| It will retain values in the event of a postback operation occurring. | In session state, user data remains in the server. Data is available to user until the browser is closed or there is session expiration. |
| Information is stored on the client’s end only. | Information is stored on the server. |
| used to allow the persistence of page-instance-specific data. | used for the persistence of user-specific data on the server’s end. |
| ViewState values are lost/cleared when new page is loaded. | SessionState can be cleared by programmer or user or in case of timeouts. |

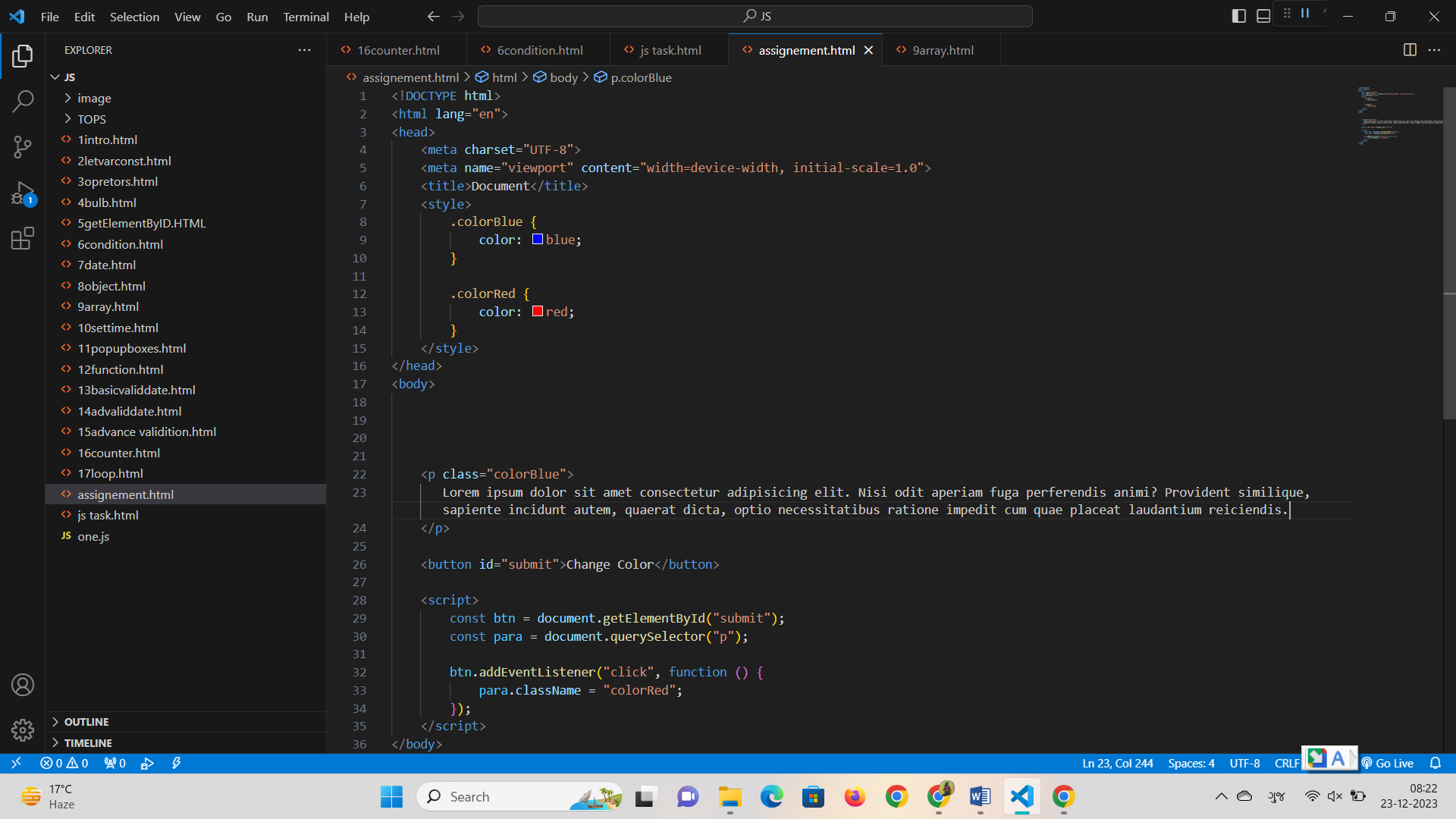
**Q-8 What is === operator?**

**Ans-** ViewState and SessionState are terms associated with ASP.NET and not JavaScript. ViewState is a client-side state management mechanism, and SessionState is server-side state management. They are used to store and retrieve data between HTTP requests in ASP.NET applications.



**Q-9 How can the style/class of an element be changed?**

**Ans-** To change the style or class of an element in JavaScript, you can use the following methods:

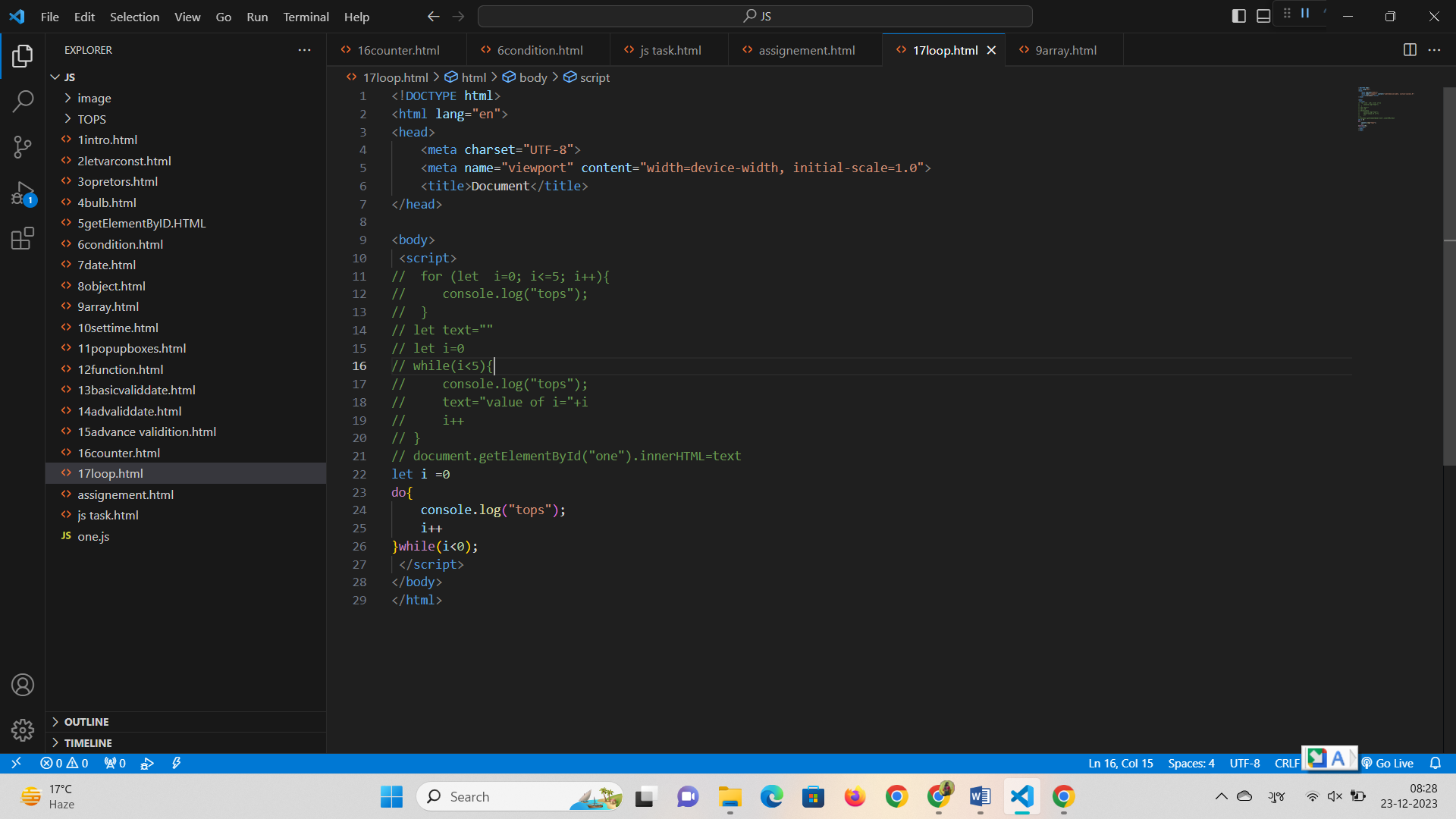


**Q-10 How to read and write a file using JavaScript?**

**Ans-** Reading and writing files using JavaScript is typically not possible in a web browser environment for security reasons. However, in server-side JavaScript (e.g., Node.js), you can use modules like fs for file I/O operations.

**Q-11 What are all the looping structures in JavaScript?**

**Ans-** Looping structures in JavaScript include for, while, do...while, and the newer for...of and for...in loops for iterating through arrays and objects.



**Q-12 How can you convert the string of any base to an integer in JavaScript?**

**Ans-** To convert a string of any base to an integer in JavaScript, you can use the parseInt function with a specified radix (base). For example:

<script>

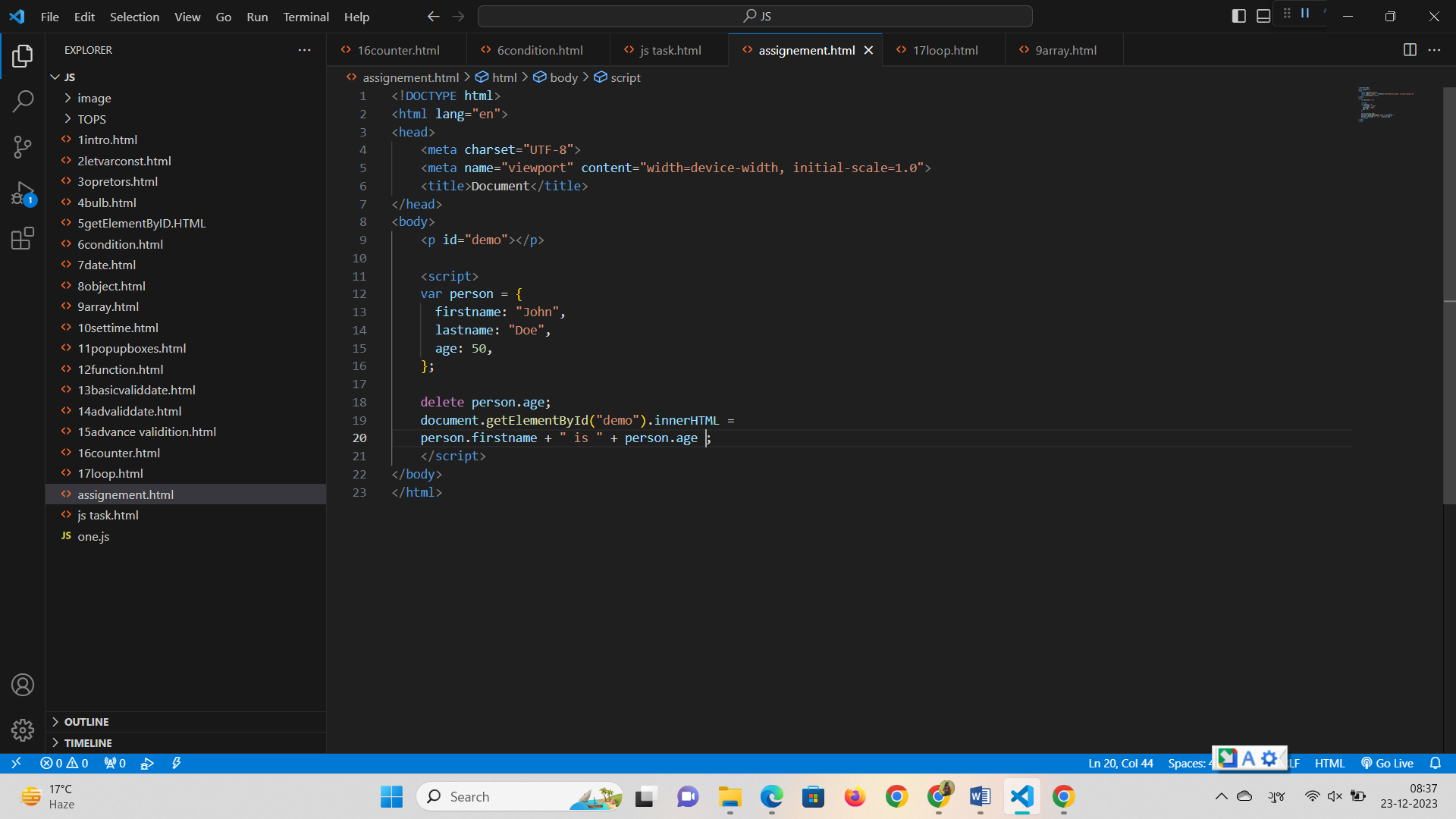
var binaryString = "1010";

var decimalValue = parseInt(binaryString, 2); // Converts binary to decimal

</script>

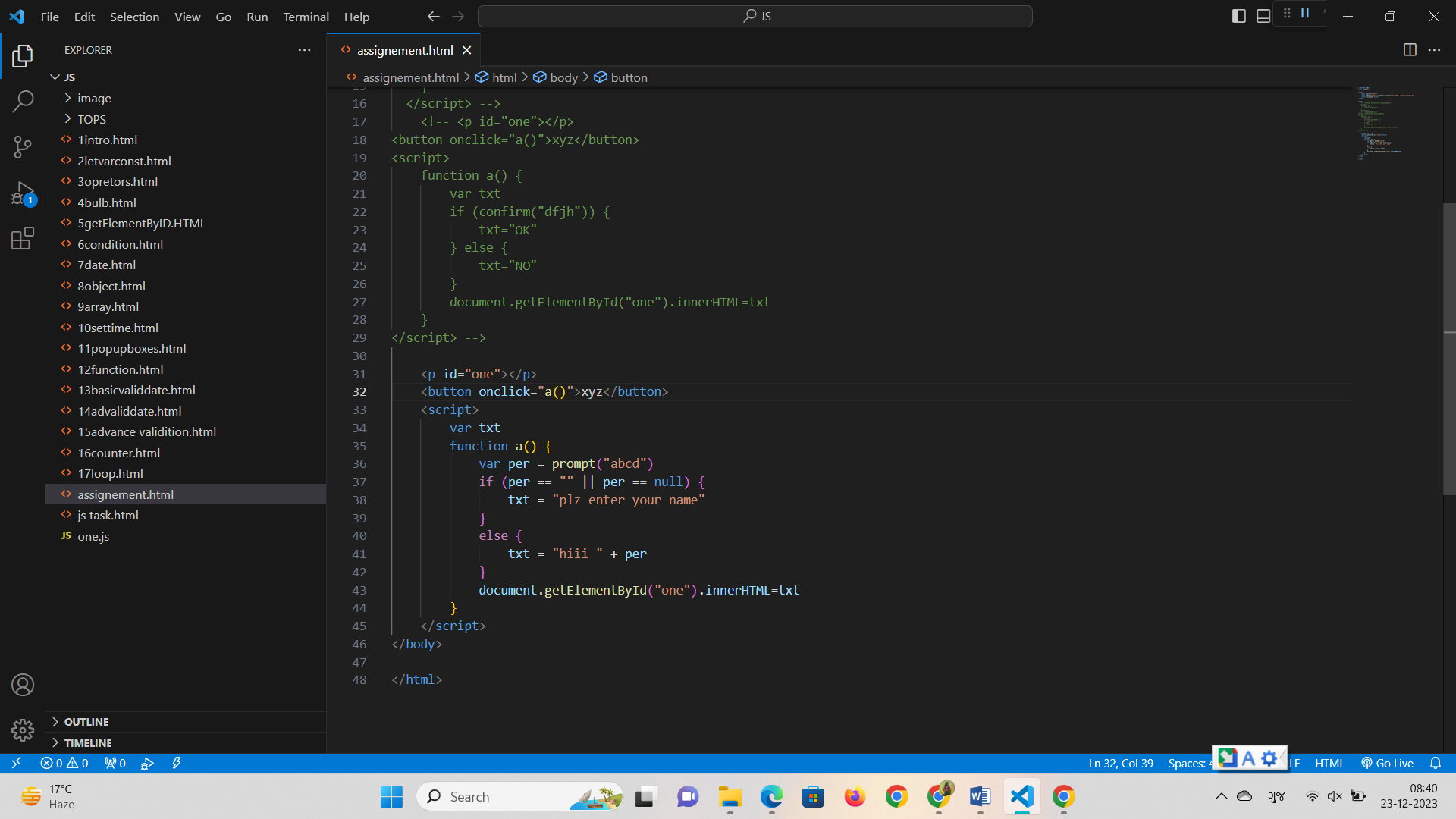
**Q-13 What is the function of the delete operator?**

**Ans-** The delete operator in JavaScript is used to delete a property from an object or an element from an array. For example:



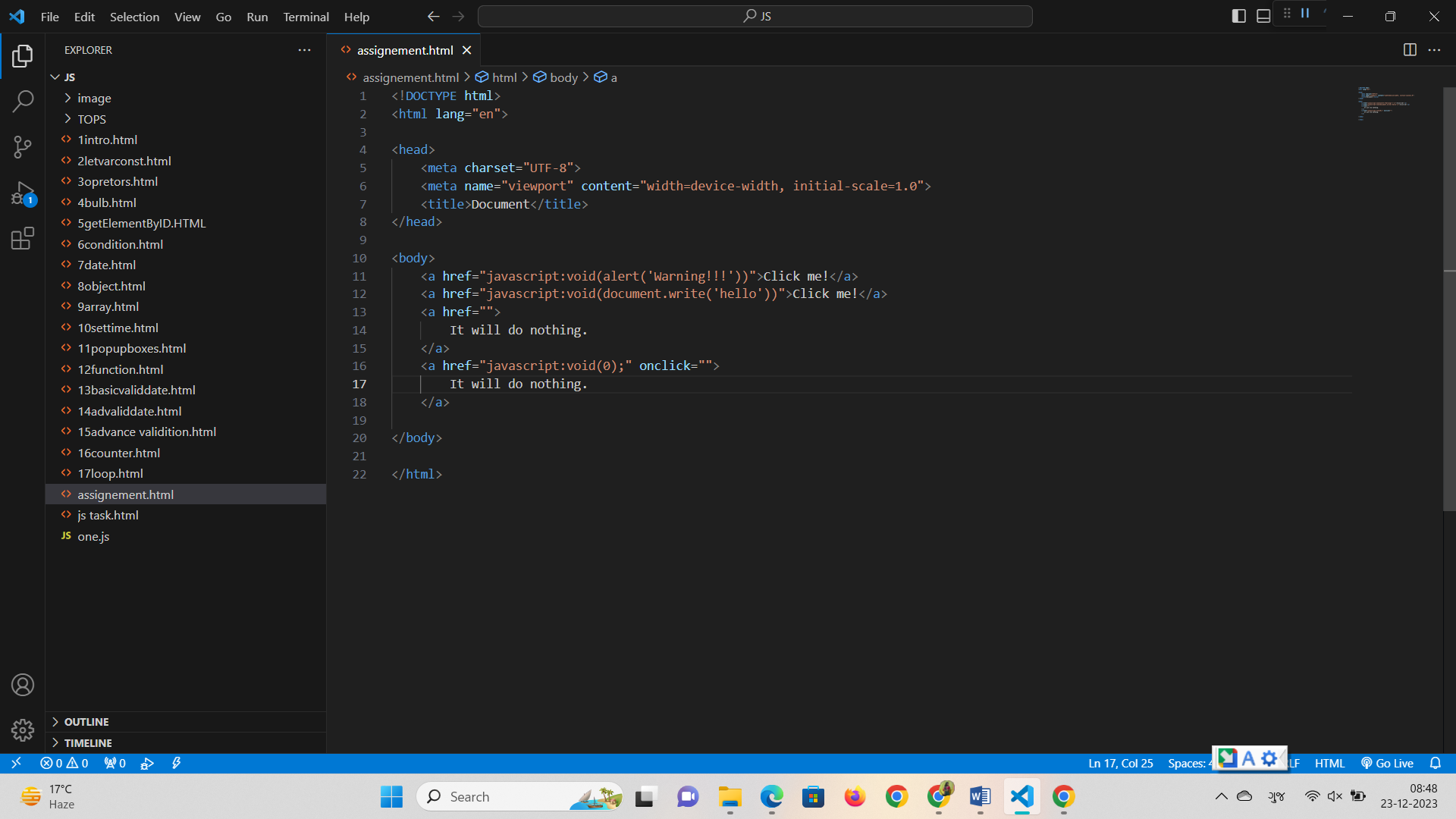
**Q-14 What are all the types of Pop up boxes available in JavaScript?**

**Ans-** In JavaScript, there are three types of pop-up boxes: alert, confirm, and prompt. These are used for displaying messages, confirming actions, and taking user input, respectively.



**Q-15 What is the use of Void (0)?**

**Ans-** void(0) is used to create a hyperlink with no action. It is often used in href attributes to prevent the page from navigating to a new URL when clicked.



**Q-16 How can a page be forced to load another page in JavaScript?**

**Ans-** You can force a page to load another page in JavaScript by setting the window.location property to the new URL:

**Q-17 What are the disadvantages of using innerHTML in JavaScript?**

**Ans-** Disadvantages of using innerHTML in JavaScript include the risk of introducing cross-site scripting (XSS) vulnerabilities if you insert unescaped user-generated content directly into the DOM, and it can be slower for complex operations compared to other DOM manipulation methods.

**Q-18 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>

**Q19- 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>

**Q-20 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>

**Q21- 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>

**Advance JavaScript for Front-End Introduction and Code Quality**

**Q1- Write a program to Show an alert**

**Ans-** To show an alert in JavaScript, you can use the alert function. Here's a simple program to show an alert:

alert("This is an alert message");

**Q2- What will be the result for these expressions?**

1. **5 > 4**

**Ans**- True

1. **"apple" > "pineapple"**

**Ans-** False

1. **"2" > "12"**

**Ans-** True (String comparison; "2" is considered greater)

1. **undefined == null**

**Ans-** True (They are considered equal in loose comparison)

1. **undefined === null**

**Ans-** False (They are not of the same data type)

1. **null == "\n0\n"**

**Ans-** False (null is not considered equal to a string)

1. **null === +"\n0\n"**

**Ans-** False (strict comparison; null and +"\n0\n" are not of the same data type)

**Q3- Will alert be shown?**

**if ("0") { alert( 'Hello'); }**

**Ans-** Yes, an alert will be shown because the string "0" is considered truthy, so the condition in the if statement is met.

**Q4- What is the code below going to output? alert( null || 2 || undefined );**

**Ans-** The code will output 2. It evaluates the logical OR (||) operator, and it returns the first truthy value it encounters, which is 2. If all values are falsy, it returns the last value, which is undefined in this case.

**Q5- 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?');**

**}**

**}**

**Ans:**

<script>

    function checkAge(age) {

  if (age > 18) {

    return true;

  } else {

    return confirm('Did parents allow you?');

  }

}

</script>

**Q6- 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."); }**

}

**Ans-**

<script>

const ask = (question, yes, no) => {

  if (confirm(question)) {

    yes();

  } else {

    no();

  }

};

ask(

  "Do you agree?",

  () => {

    alert("You agreed.");

  },

  () => {

    alert("You canceled the execution.");

  }

);

</script>

**JavaScript Essentials**

**Q1-** **Calculate subtotal price of quantity in JavaScript?**

**Ans-**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Subtotal Price</title>

    <style>

        table {

          border-collapse: collapse;

          width: 100%;

        }

        th, td {

          border: 1px solid #ddd;

          padding: 8px;

          text-align: left;

        }

        th {

          background-color: #f2f2f2;

        }

        .remove-button {

          background-color: orange;

          color: white;

          border: none;

          padding: 5px 10px;

          cursor: pointer;

        }

      </style>

</head>

<body>

    <table>

        <thead>

          <tr>

            <th>Product Info</th>

            <th>Quantity</th>

            <th>Price per Unit</th>

            <th>Price Subtotal</th>

            <th></th>

          </tr>

        </thead>

        <tbody>

          <!-- First Row -->

          <tr>

            <td>Product 1</td>

            <td><input type="number" id="n1" onkeyup="a()"></td>

            <td>$20</td>

            <td id="r1">0</td>

            <td><button class="remove-button">Remove</button></td>

          </tr>

          <!-- Second Row -->

          <td>Product 2</td>

          <td><input type="number" id="n2" onkeyup="a()"></td>

          <td>$40</td>

          <td id="r2">0</td>

          <td><button class="remove-button">Remove</button></td>

        </tr>

          <!-- third Row -->

          <tr>

            <td></td>

            <td></td>

            <td></td>

            <td id="total">0</td>

            <td></td>

          </tr>

        </tbody>

      </table>

      <script>

     function a() {

        var n1=document.getElementById("n1").value

        var r1=document.getElementById("r1").innerHTML='$' + (20\*n1)

        var n2=document.getElementById("n2").value

        var r2=document.getElementById("r2").innerHTML= '$' + (40\*n2)

        var total=document.getElementById("total").innerHTML= '$' + (20\*n1 + 40\*n2)

     }

      </script>

</body>

</html>

**Q-2 What is JavaScript Output method?**

**Ans-** JavaScript doesn't have a specific "output method" in the same way some other languages do. However, JavaScript is commonly used to manipulate and interact with the Document Object Model (DOM) of a webpage, and this interaction often involves displaying output to the user. The primary methods for output in JavaScript include:

1. console.log(): This method is used for logging messages to the browser console. It's commonly used for debugging purposes.

console.log("Hello, world!");

1. DOM Manipulation: JavaScript can dynamically manipulate the content of an HTML document by interacting with the DOM. For example, changing the text content of an HTML element:

document.getElementById("exampleElement").innerText = "New Text";

or appending HTML content:

document.getElementById("exampleElement").innerHTML = "<p>New HTML content</p>";

**Q-3 How to used JavaScript Output method?**

**Ans-** Using JavaScript to output information typically involves using **console.log()** for debugging or updating HTML content through the DOM for user-facing output.

Here's an example of using console.log():

console.log("Hello, world!");

And an example of updating HTML content using the DOM:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>JavaScript Output Example</title>

</head>

<body>

<p id="output">Initial content</p>

<script>

// JavaScript code to update the content of the paragraph

document.getElementById("output").innerText = "Updated content";

</script>

</body>

</html>

**Q-4 How to used JavaScript Events to do all examples**

**Ans-** JavaScript events are actions or occurrences that happen in the browser, such as a user clicking a button or resizing the window. You can use events to trigger JavaScript code. Here's an example of using JavaScript events:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>JavaScript Events Example</title>

<style>

button {

padding: 10px;

font-size: 16px;

cursor: pointer;

}

</style>

</head>

<body>

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

<script>

// JavaScript code to handle the click event

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

alert("Button clicked!");

});

</script>

</body>

</html>

In this example:

The HTML button element has an ID of "myButton."

The JavaScript code uses addEventListener to listen for a "click" event on the button.

When the button is clicked, the provided function is executed, showing an alert with the message "Button clicked!"

This is a basic example, and JavaScript events can be used for a wide range of interactions, such as handling form submissions, responding to keyboard input, and more.