1. Define JavaScript? Explain JavaScript in Different Browsers.

JavaScript is a **scripting and programming language** used to make web pages **interactive, dynamic, and user-friendly**. It is mainly used on the **client side** of web development. JavaScript helps in form validation, displaying alert messages, creating animations, handling events like button clicks, and changing web page content without reloading the page.

JavaScript runs inside a web browser using a **JavaScript engine**. Different browsers use different engines to execute JavaScript code efficiently.

- **Google Chrome** uses the **V8 Engine**
- **Mozilla Firefox** uses **SpiderMonkey**
- **Microsoft Edge** uses **Chakra / V8**
- **Safari** uses **JavaScriptCore**

Although engines are different, JavaScript code works similarly in all modern browsers.

2. Explain HTML Documents in JavaScript with Example.

JavaScript can access and modify **HTML documents** using the **Document Object Model (DOM)**. An HTML document is treated as a tree of elements, and JavaScript can change its content, structure, and style.

Using JavaScript, we can:

- Change text of HTML elements
- Change colors and styles
- Show or hide elements
- Respond to user actions

Example:

```
<p id="demo">Hello</p>
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript";
</script>
```

Explanation:
JavaScript accesses the paragraph element using its `id` and changes its content.

3. Explain Different Data Types and Variables Used in JavaScript.

A **variable** is used to store data values in JavaScript. Variables are declared using `let`, `var`, or `const`.

Example:

```
let name = "Ram";
```

Data Types in JavaScript:

1. **Number** – Used to store numbers
   Example: `let age = 15;`
2. **String** – Used to store text
   Example: `let city = "Delhi";`
3. **Boolean** – Stores `true` or `false`
   Example: `let isStudent = true;`
4. **Undefined** – Variable declared but not assigned value
   Example: `let x;`
5. **Null** – Represents empty value
   Example: `let y = null;`
6. **Object** – Stores multiple values
   Example: `let student = {name:"Ram", age:15};`

4. Explain Meaning of HTML DOM and Different DOM Methods.

HTML DOM (Document Object Model) is a programming interface that represents an HTML document as a **tree structure of objects**. It allows JavaScript to access, modify, add, or delete HTML elements dynamically.

Using DOM, JavaScript can:

- Change HTML content
- Change styles
- Handle events

Common DOM Methods:

- `getElementById()` – Access element using id
- `getElementsByTagName()` – Access elements by tag
- `getElementsByClassName()` – Access elements by class
- `querySelector()` – Selects first matching element

Example:

```
document.getElementById("demo").style.color = "red";
```

5. Explain Different DOM Elements.

DOM elements are HTML elements that JavaScript can access and manipulate through the DOM. Each HTML tag becomes an object in the DOM.

Common DOM Elements:

- `<p>` – Paragraph
- `<div>` – Division
- `<button>` – Button
- `<input>` – Input field
- `<h1>` to `<h6>` – Headings

JavaScript can perform operations such as:

- Changing text
- Changing color
- Hiding or showing elements
- Adding new elements

Example:

```
document.getElementById("btn").innerHTML = "Clicked";
```

6. Explain Different Types of Control Flow Statements in JavaScript.

Control flow statements control the **execution order** of a program.

Types of Control Flow Statements:

1. Conditional Statements

Used to make decisions.

- `if`
- `if...else`
- `switch`

Example:

```
if (age >= 18) {
  alert("Eligible to vote");
}
```

2. Looping Statements

Used to repeat code.

- `for`
- `while`
- `do...while`

Example:

```
for (let i = 1; i <= 5; i++) {
  console.log(i);
}
```

7. Explain About JavaScript Function.

A **function** is a block of reusable code that performs a specific task. Functions help in reducing code repetition and make programs easier to understand and maintain.

Advantages of Functions:

- Code reusability
- Easy debugging
- Better readability

Example:

```
function greet() {
  alert("Welcome!");
}
greet();
```

The function is executed when it is called.

8. Discuss How JavaScript Is Used in Present Day Web Applications.

JavaScript plays an important role in modern web applications. It helps make websites interactive and user-friendly.

JavaScript is used for:

- Form validation
- Dynamic content loading
- Interactive buttons and menus
- Online games

- Social media platforms
- E-commerce websites

Examples:

- Login validation
- Live chat
- Image sliders
- Search suggestions

9. Short Notes

Prompt

The `prompt()` method is used to take input from the user.

```
let name = prompt("Enter your name");
```

Confirm

The `confirm()` method displays a dialog box with **OK and Cancel** buttons. It returns `true` or `false`.

```
confirm("Are you sure?");
```

Alert

The `alert()` method displays a message box with an OK button.

```
alert("Welcome to JavaScript");
```

PRACTICAL QUESTION:

1. Current Date

Q: Create a program in JavaScript to obtain the current date.

A:

<!-- HTML -->

<button onclick="showDate()">Show Current Date</button>

```
<p class="output" id="dateOutput"></p>

<!-- JavaScript -->
<script>
function showDate() {
  let today = new Date();
  document.getElementById("dateOutput").innerText = "Current Date: " + today.toDateString();
}
</script>
```

2. Compare Positive and Negative Values

Q: Create a JavaScript program that compares the positive and negative values of two integers.

A:

```
<!-- HTML -->
<button onclick="compareNumbers()">Compare Numbers</button>
<p class="output" id="compareOutput"></p>

<!-- JavaScript -->
<script>
function compareNumbers() {
  let a = parseInt(prompt("Enter first number:"));
  let b = parseInt(prompt("Enter second number:"));
  let result = (a > b) ? a + " is greater" : b + " is greater";
  document.getElementById("compareOutput").innerText = result;
}
```

```
</script>
```

3. Sum of Two Variables

Q: Utilizing two variables x and y, display the sum of 10 and 15.

A:

```
<!-- HTML -->
<button onclick="sumXY()">Sum of 10 and 15</button>
<p class="output" id="sumOutput"></p>

<!-- JavaScript -->
<script>
function sumXY() {
  let x = 10, y = 15;
  let sum = x + y;
  document.getElementById("sumOutput").innerText = "Sum: " + sum;
}
</script>
```

4. Assign x + y to z

Q: Assign x and y to a variable called z, display the result in an alert box.

A:

```
<!-- HTML -->
<button onclick="assignToZ()">Assign x + y to z</button>
<p class="output" id="zOutput"></p>

<!-- JavaScript -->
```

```
<script>
function assignToZ() {
   let x = 10, y = 15;
   let z = x + y;
   alert("Value of z: " + z);
   document.getElementById("zOutput").innerText = "z = " + z;
}
</script>
```

5. Change Paragraph Background Color

Q: Create a JavaScript program to change a paragraph's background color.

A:

```
<!-- HTML -->
<p id="para">This is a paragraph</p>
<button onclick="changeParaBG()">Change Background Color</button>


<!-- JavaScript -->
<script>
function changeParaBG() {
   let para = document.getElementById("para");
   para.style.backgroundColor = "yellow";
   para.style.color = "red";
   para.style.padding = "10px";
}
</script>
```

6. Add Table Rows

Q: Create a JavaScript function to add table rows.

A:

```
<!-- HTML -->
<table border="1" id="myTable">
  <tr><th>Name</th></tr>
</table>
<button onclick="addTableRow()">Add Table Row</button>

<!-- JavaScript -->
<script>
function addTableRow() {
  let table = document.getElementById("myTable");
  let row = table.insertRow();
  let cell = row.insertCell();
  cell.innerText = "New Row";
}
</script>
```

7. List Object Properties

Q: Create a JavaScript program to list an object's properties.

A:

```
<!-- HTML -->
<button onclick="listProperties()">Show Object Properties</button>
<p class="output" id="propertiesOutput"></p>
```

```
<!-- JavaScript -->

<script>

const student = {name:"Ram", age:15, grade:9};

function listProperties() {

    let props = "";

    for (let key in student) {

        props += key + " ";

    }

    document.getElementById("propertiesOutput").innerText = "Properties: " + props;

}

</script>
```

8. Print Object Methods

Q: Create a JavaScript function to print all of an object's methods.

A:

```
<!-- HTML -->

<button onclick="showMethods()">Show Object Methods</button>

<p class="output" id="methodsOutput"></p>
```

```
<!-- JavaScript -->

<script>

const person = {name: "Ravi", greet: function(){}, walk: function(){}};

function showMethods() {

    let methods = "";

    for (let key in person) {

        if(typeof person[key] === "function") {
```

```
      methods += key + " ";
    }
  }
  document.getElementById("methodsOutput").innerText = "Methods: " + methods;
}
</script>
```

9. Reverse a Number

Q: Create a JavaScript function that reverses a number.

A:

```
<!-- HTML -->
<button onclick="reverseNumber()">Reverse Number</button>
<p class="output" id="reverseOutput"></p>

<!-- JavaScript -->
<script>
function reverseNumber() {
  let num = prompt("Enter a number to reverse:");
  let rev = num.toString().split("").reverse().join("");
  document.getElementById("reverseOutput").innerText = "Reversed: " + rev;
}
</script>
```

10. Create Variable from User Input

Q: Create a JavaScript exercise to create a variable with a name that is specified by the user.

A:

```
<!-- HTML -->
<button onclick="createUserVariable()">Create Variable</button>
<p class="output" id="userVarOutput"></p>


<!-- JavaScript -->
<script>
function createUserVariable() {
  let varName = prompt("Enter variable name:");
  let value = prompt("Enter variable value:");
  window[varName] = value;
  document.getElementById("userVarOutput").innerText = varName + " = " + window[varName];
}
</script>
```

11. Check Multiple of 3 or 7

Q: Create a JavaScript application to determine whether a given positive number is a multiple of 3 or 7.

A:

```
<!-- HTML -->
<button onclick="checkMultiple()">Check Multiple of 3 or 7</button>
<p class="output" id="multipleOutput"></p>


<!-- JavaScript -->
<script>
function checkMultiple() {
  let num = parseInt(prompt("Enter a positive number:"));
```

```
    let result = (num % 3 === 0 || num % 7 === 0) ? "Yes, multiple of 3 or 7" : "No, not a multiple";
    document.getElementById("multipleOutput").innerText = result;
}
</script>
```