Introduction to JavaScript

1. Overview of JavaScript

JavaScript (JS) is a high-level, interpreted programming language that enables interactive web pages. It is one of the core technologies of the World Wide Web, along with HTML and CSS. JavaScript is used for client-side scripting, allowing dynamic content updates, form validations, and event-driven programming.

Key Features of JavaScript

- Lightweight and fast execution
- Interpreted and does not require compilation
- Supports object-oriented, imperative, and functional programming styles
- Can be executed in any web browser without additional plugins
- Provides event-driven functionality (handling user interactions)

2. History and Evolution of JavaScript

- **1995**: Brendan Eich developed JavaScript while working at Netscape.
- **1996**: JavaScript was standardized as ECMAScript (ES1).
- **2009**: ECMAScript 5 (ES5) introduced stricter syntax and additional methods.
- **2015**: ECMAScript 6 (ES6) introduced modern syntax like `let`, `const`, arrow functions, template literals, and promises.
- **Present**: Ongoing updates introduce features such as async/await, optional chaining, and improved modules.

3. JavaScript in the Browser

JavaScript runs inside the browser as part of the Document Object Model (DOM) manipulation. Web developers use JS to:

- Change HTML content dynamically
- Modify CSS properties
- Handle user events (click, hover, etc.)

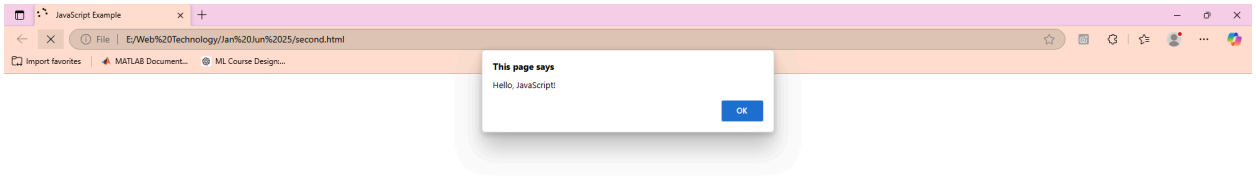
- Communicate with servers using AJAX and Fetch API

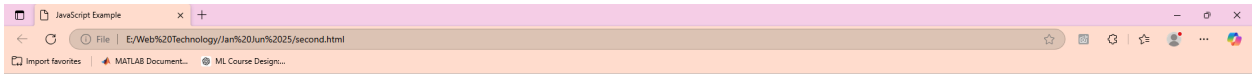
4. JavaScript Syntax and Structure

JavaScript code is written within `<script>` tags in an HTML file or in an external `.js` file.

Example of JavaScript in an HTML file:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>JavaScript Example</title>
  <script>
    alert("Hello, JavaScript!");
  </script>
</head>
<body>
  <h1>Welcome to JavaScript</h1>
</body>
</html>
```





5. Variables and Data Types

Declaring Variables

Variables in JavaScript can be declared using `var`, `let`, and `const`.

```
var name = "John";  // Function-scoped
let age = 25;  // Block-scoped
const pi = 3.14;  // Constant value
```

Data Types in JavaScript

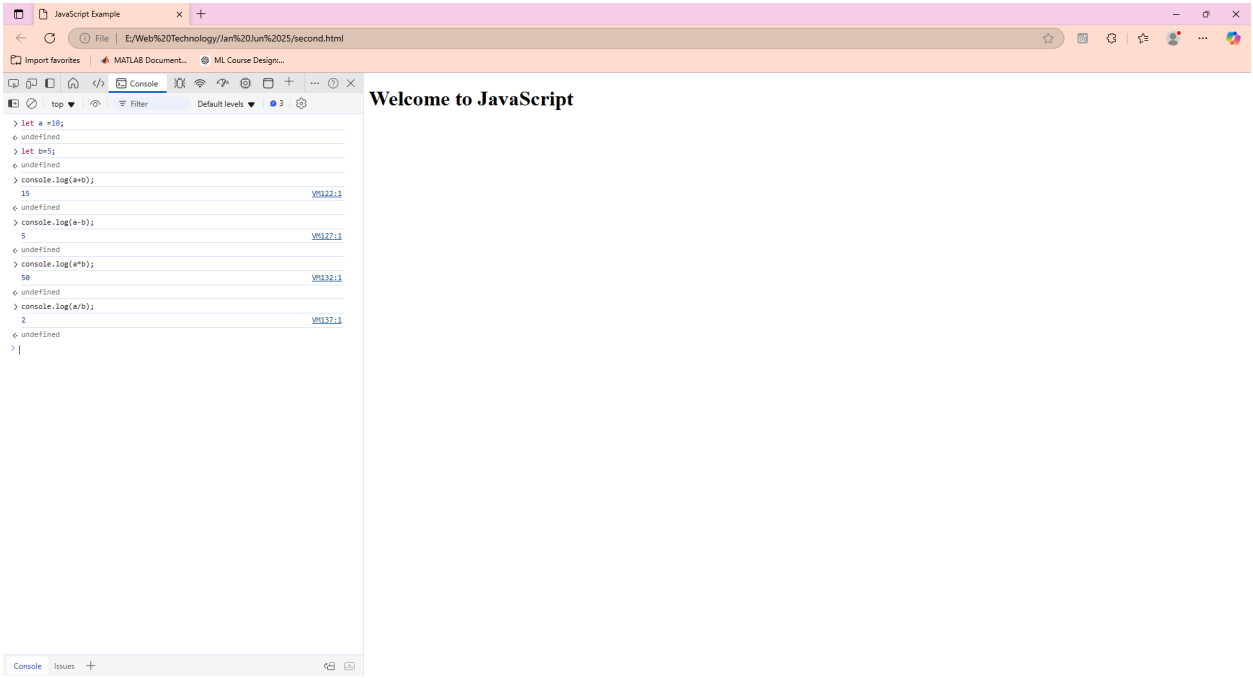
1. **Primitive Data Types**
    - `String`: "Hello"
    - `Number`: 100, 3.14
    - `Boolean`: true, false
    - `Undefined`: A declared variable without a value
    - `Null`: An intentional absence of value
    - `Symbol`: Unique identifier (ES6)
2. **Reference Data Types**
    - Arrays

- Objects
- Functions

6. Operators in JavaScript

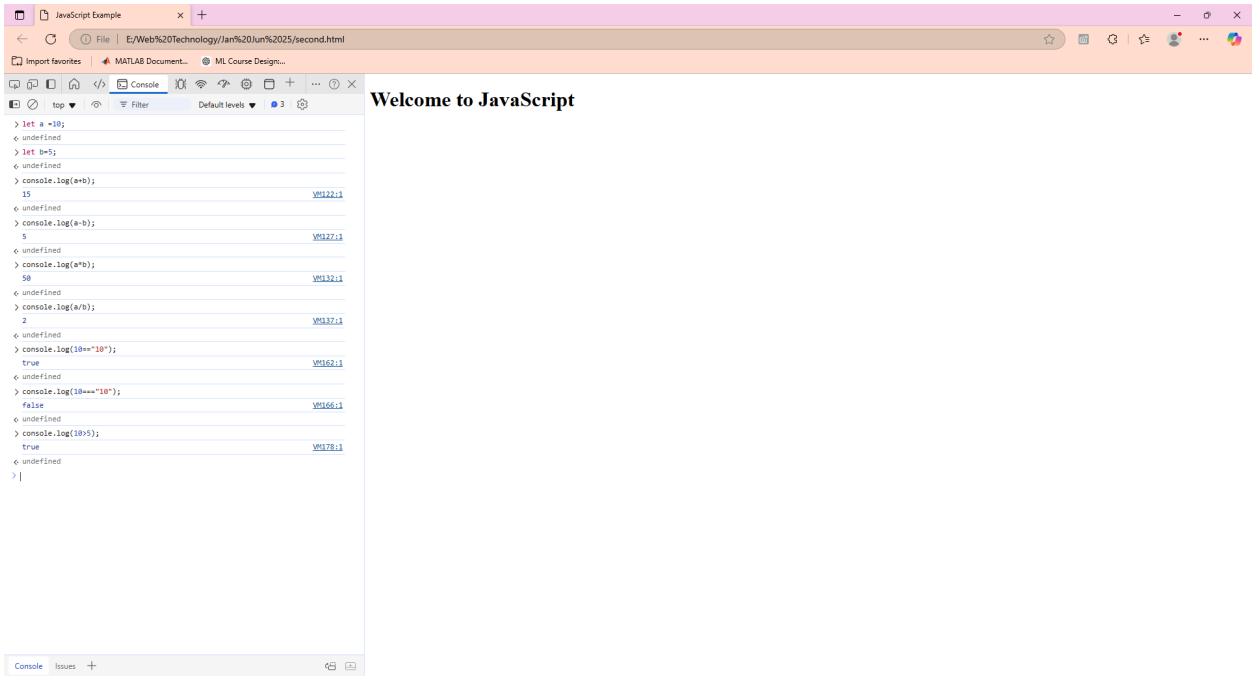
Arithmetic Operators:

```
let a = 10;
let b = 5;
console.log(a + b); // Addition: 15
console.log(a - b); // Subtraction: 5
console.log(a * b); // Multiplication: 50
console.log(a / b); // Division: 2
```



Comparison Operators:

```
console.log(10 == "10");  // true (type conversion)
console.log(10 === "10"); // false (strict comparison)
console.log(10 > 5);      // true
```

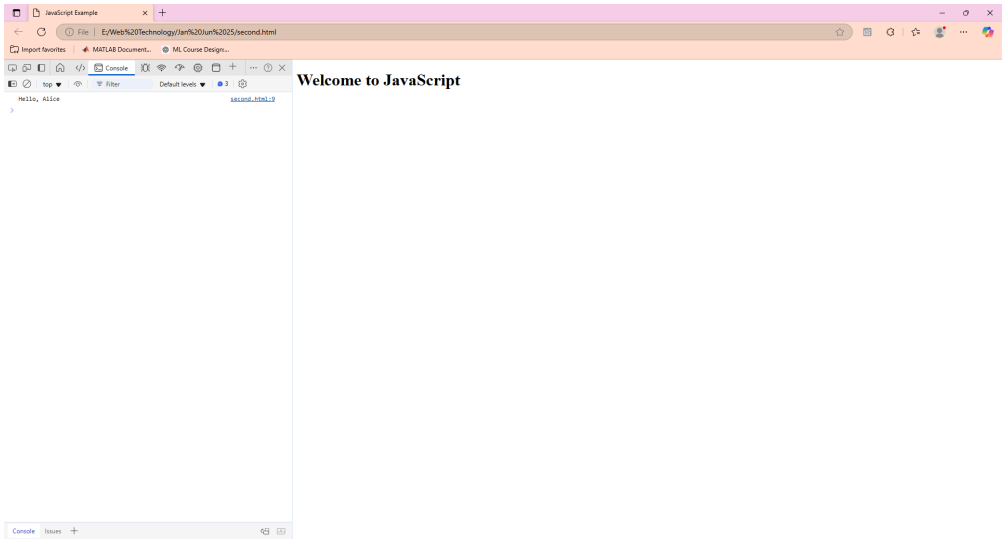


7. Functions in JavaScript

JavaScript functions allow code reuse and modularity.

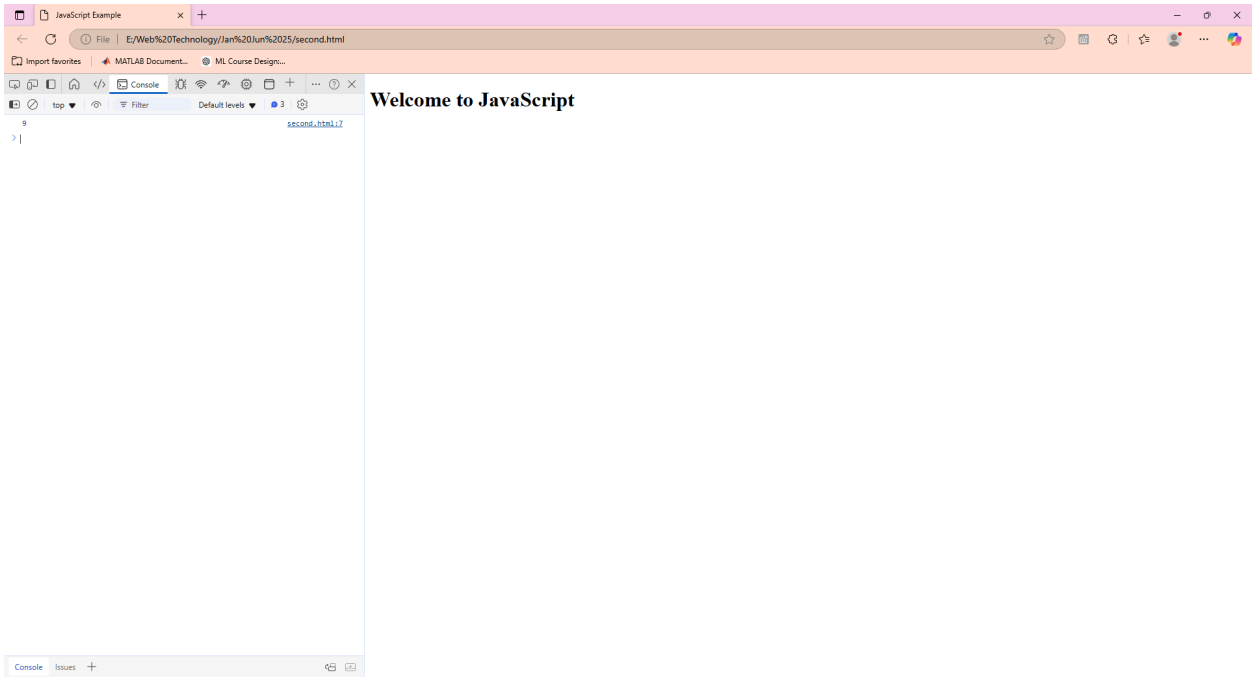
Function Declaration:

```
function greet(name) {
   return "Hello, " + name;
}
console.log(greet("Alice"));
```



Arrow Function (ES6):

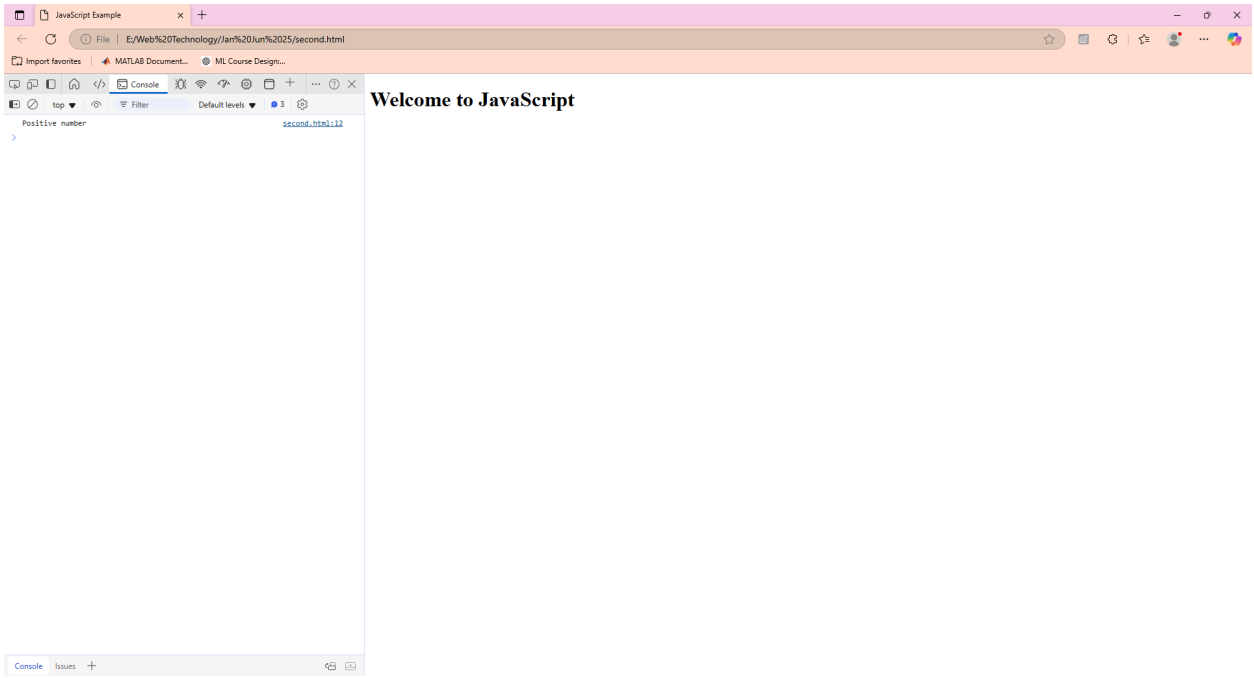
```
const add = (x, y) => x + y;
console.log(add(4, 5));
```



8. Control Structures

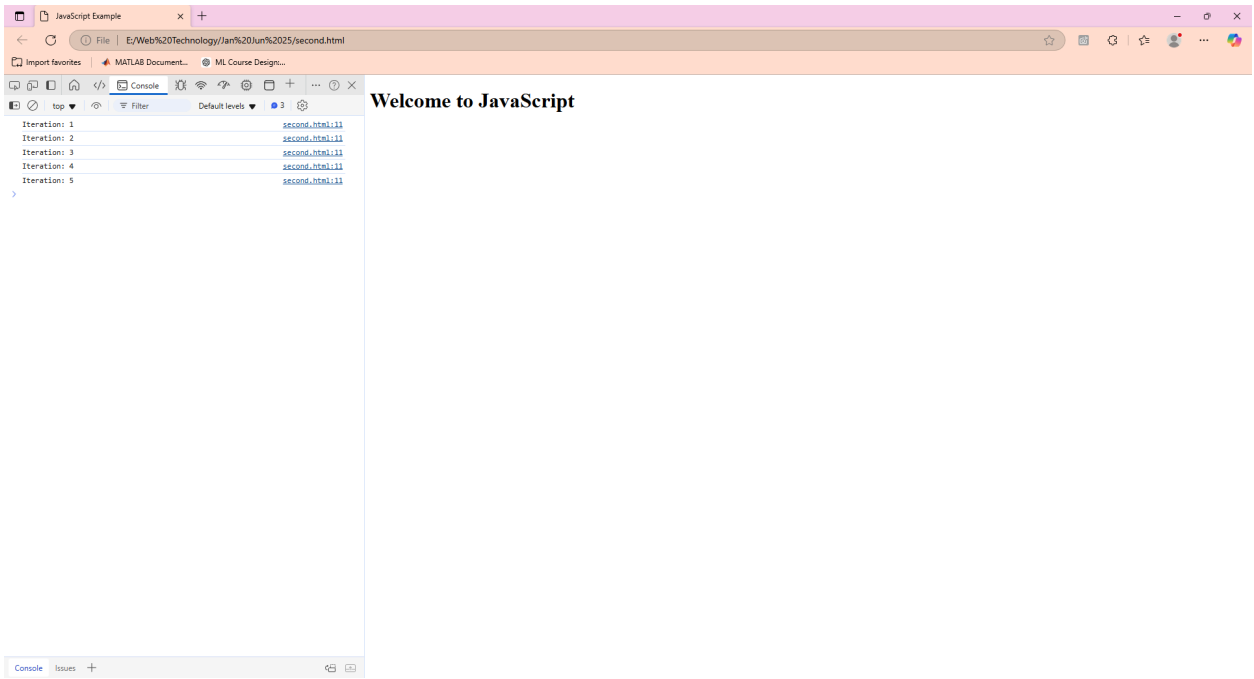
Conditional Statements:

```
let num = 10;
if (num > 0) {
  console.log("Positive number");
} else {
  console.log("Negative number");
}
```



Loops:

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



9. DOM Manipulation

JavaScript interacts with HTML and CSS via the Document Object Model (DOM).

document.getElementById("demo").innerHTML = "Changed Text";