JAVASCRIPT

What is JavaScript?

JavaScript is a versatile scripting language that brings web pages to life by enabling **dynamic and interactive** elements. It works alongside HTML and CSS to create engaging user experiences, making it an essential part of modern web development.

Unlike static web pages that only display fixed content, JavaScript allows developers to modify page elements, handle user interactions, and control multimedia **without requiring a page reload.** It is a **lightweight, interpreted language** that runs directly in the browser, ensuring fast execution and smooth performance.

Why Use JavaScript?

- **Enhances User Experience –** Enables interactive menus, animations, and real-time updates.
- **Manipulates Web Content –** Dynamically changes HTML and CSS elements on the page.
- **Handles User Input –** Validates forms and process user actions efficiently.
- **Supports Asynchronous Operations –** Fetches data from servers without reloading the page.
- **Cross-Platform Compatibility –** Runs on all major browsers and devices.
- **Extends Beyond the Browser –** With Node.js, JavaScript is also used for backend development.

Example code

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Demo</title>
</head>
<body>
    <p id="text">Click the button to change this text!</p>
    <button onclick="changeText()">Click Me</button>

    <script>
        function changeText() {
            document.getElementById("text").innerText = "JavaScript is awesome!";
        }
    </script>
</body>
</html>
```

Inline JavaScript

(JavaScript is written inside the HTML tag)

```
<!DOCTYPE html>
<html>
<head>
    <title>Inline JavaScript Example</title>
</head>
<body>
     <button onclick="alert('Hello, World!')">Click Me</button>
</body>
</html>
```

Internal JavaScript

(CSS is written inside a **<script>** tag within the **<head>** or **<body>** section)

In <head> tag

```
<!DOCTYPE html>
<html>
<head>
    <title>Internal JavaScript</title>
    <script>
        function greet() {
            alert("Welcome to
JavaScript!");
        }
    </script>
</head>
<body>
    <button onclick="greet()">Click
Me</button>
</body>
</html>
```

In <body> tag

```
<!DOCTYPE html>
<html>
<head>
    <title>Internal JavaScript</title>
</head>
<body>
    <button onclick="greet()">Click
Me</button>
    <script>
        function greet() {
            alert("Welcome to
JavaScript!");
        }
    </script>
</body>
</html>
```

External JavaScript

(JavaScript is written in a separate file and linked to HTML)

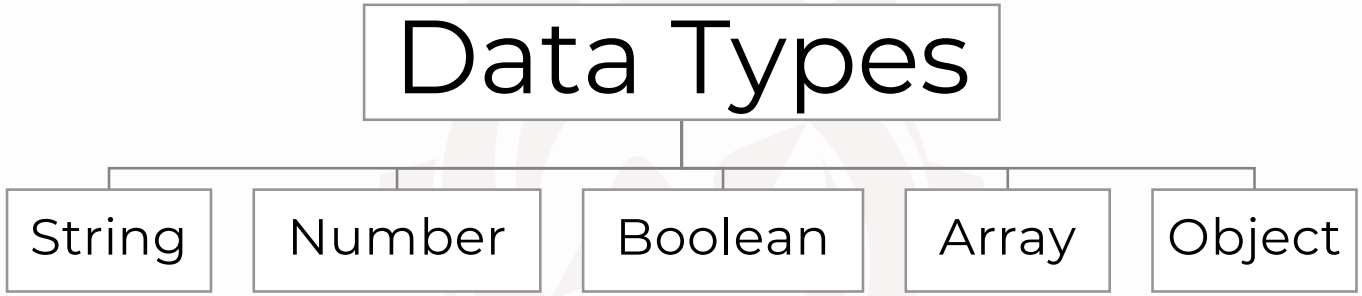
Create a JavaScript file (**app.js**):

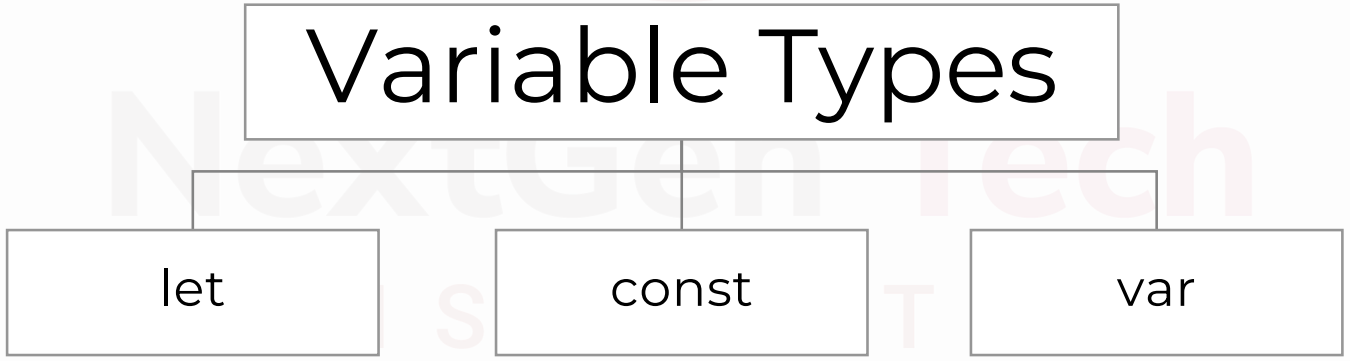
```
function greet() {
    alert("Welcome to JavaScript!");
}
```

Link it in an **HTML** file:

```
<!DOCTYPE html>
<html>
<head>
    <title>External JavaScript
Example</title>
</head>
<body>
     <button onclick="greet()">Click
Me</button>
     <script src="app.js"></script>
</body>
</html>
```

JavaScript Syntax





Variables

Variables store data values and can be changed or updated

- `let name = "John";  // String`
- `let age = 25;       // Number`
- `let isStudent = true; // Boolean`

- **let** and **const** are **recommended over var**.
- **let** allows **reassignment**, while **const** is for values that **do not change**.

Data Types

- `String → "Hello“`
- `Number → 25`
- `Boolean → true or false`
- `Array → ["Apple", "Banana", "Orange"]`
- `Object → { name: "John", age: 25 }`

Output Methods

```
console.log("Hello, World!");
```
console.log() is used for debugging.
Displays in the browser console.

```
alert("Welcome!");
```
alert() is useful for notifications.
Shows a pop-up alert.

```
document.write("Hello!");
```
document.write() is rarely used but can be written directly to the page.
Writes directly to the webpage.

Operators

Arithmetic Operators

Used for mathematical calculations.

Operator	Usage	Output
Addition	let sum = 12 + 8;	20
Subtraction	let difference = 15 - 7;	8
Multiplication	let product = 6 * 4;	24
Division	let quotient = 20 / 5;	4
Modulus (Remainder)	let remainder = 17 % 5;	2

Comparison Operators

Used to compare values.

Operator	Sign	Usage	Output	Explanation
Greater than	**>**	10 > 5	True	Checks if the left value is greater than the right.
Less than	**<**	10 < 5	False	Checks if the left value is smaller than the right.
Equal to	**==**	10 == "10"	True	Checks if two values are equal (but not necessarily the same type).
Strictly equal to	**===**	10 === "10"	False	Checks if two values are equal **and** of the same type.
Not equal to	**!=**	10 != "10"	False	Checks if two values are different (ignores type).
Strictly not equal to	**!==**	10 !== "10"	True	Checks if two values are different **or** of different types.

Logical Operators

Operator	Sign	Usage	Explanation
AND	**&&**	`(true && false)`	Both must be true
OR	**||**	(true || false)	At least one must be true
NOT	**!**	`(!true)`	Reverses the Boolean value

```
let age = 18;
console.log(age >= 18 && age <= 25); // Output: true (Age is between 18 and 25)
console.log(age < 18 || age > 25); // Output: false (Age is not outside this range)
console.log(!age < 18 || age > 25); // Output: true
```

Modulus Operators % (Finding Remainder)

Examples

```
console.log(10 % 3);
```
Output: 1 (10 ÷ 3 = 3 remainder 1)

```
console.log(25 % 7);
```
Output: 4 (25 ÷ 7 = 3 remainder 4)

```
console.log(16 % 4);
```
Output: 0 (16 ÷ 4 = 4 remainder 0)

Referring Sources

JavaScript: https://www.w3schools.com/js