

# JavaScript

## What is JavaScript?

JavaScript is a high-level, interpreted programming language that is primarily used for creating dynamic and interactive content on websites. It is one of the core technologies of web development, along with HTML and CSS.

**High-level:** You don't need to worry about low-level machine code, JS code looks more like human language.

**Interpreted:** The browser translates your JS code line by line into instructions the computer understands, rather than compiling it all at once.

## Why JavaScript?

**Interactive Web Pages:** JS brings web pages to life. Imagine forms that validate user input before submission, menus that respond to hover events, or content that updates without reloading the entire page.

**Rich Ecosystem:** JavaScript has a vast ecosystem of libraries and frameworks like React, Angular, and Vue.js, which help you build complex applications more efficiently. It also has a strong community and plenty of resources for learning and troubleshooting.

**Full-Stack Development:** With the introduction of Node.js, JavaScript can also be used on the server side. This means you can use one language for both client-side and server-side development, making you a full-stack developer.

**Essential for Web Development:** JavaScript is a core technology for creating web pages. Alongside HTML (for structure) and CSS (for styling), JavaScript brings interactivity to websites. If you want to build web applications or become a web developer, knowing JavaScript is crucial.

## Let's Start

### Printing to the Console:

Use the `console.log()` method to print messages to the browser's developer console. This is where developers inspect code output and errors

For Eg.

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

### 2. Using `document.write()`

The `document.write()` method writes directly to the HTML document. However, it's generally not recommended for modern web development because it can overwrite the entire document if used after the document has loaded.

For Eg.

```
document.write("Hello, World!");
```

```
document.write("<p>This is a paragraph.</p>");
```

### 3. Using `alert()`

The `alert()` method displays a pop-up dialog with the specified message. It's useful for simple notifications or debugging but can be intrusive for users.

For Eg.

```
alert("Hello, World!");
```

### 4. Using `innerHTML`

You can change the content of HTML elements using JavaScript. This is a common way to display information on a web page.

For Eg.

```
document.getElementById("myHeading").innerHTML = "Hello, World!";
```

Printing Expressions in Js:

```
console.log(2 + 3); // Arithmetic expression
```

```
console.log("Hello" + " " + "World"); // String concatenation
```

```
console.log(10 > 5); // Boolean expression
```

## Variables in JS

variables are used to store data values. You can create variables using the var, let, or const keywords.

### Using var

- var is the traditional way to declare variables. It has function scope and is hoisted to the top of its scope.

javascript

Copy code

```
var x = 10;
```

```
console.log(x); // 10
```

### Using let

- let is a newer way to declare variables. It has block scope, meaning it is only accessible within the block it is defined.

javascript

Copy code

```
let y = 20;
```

```
console.log(y); // 20
```

### 🔗 Using const

- const is used to declare constants, which are variables whose value cannot be changed after they are initialized. It also has block scope.

javascript

Copy code

```
const z = 30;  
console.log(z); // 30
```