What is JavaScript (JS)?

JavaScript is a high-level, interpreted programming language used mainly to make web pages interactive and dynamic.

It runs inside the browser (like Chrome, Edge, or Firefox), but can also run on servers using Node.js.

In short:

HTML → structures the webpage
CSS → styles the webpage
JavaScript → makes the webpage come alive

Why Learn JavaScript?

Here's why JavaScript is so important and widely used

---

1. It Runs Everywhere

- **Works in all modern browsers (no installation needed).**
- **Also runs on servers (Node.js), mobile apps (React Native), and desktop apps (Electron).**

---

2. Adds Interactivity to Webpages

JS lets you respond to user actions — like clicks, inputs, or scrolling.

```
document.querySelector("button").addEventListener("click", () => {
  alert("Hello, Rakshith!");
});
```

Without JS, webpages are static (just text and images).

---

3. Controls the Browser (DOM Manipulation)

You can change HTML and CSS dynamically:

- **Show/hide elements**
- **Update content live**
- **Animate components**
- **Validate forms before submission**

4. Used for Full-Stack Development

With tools like:

- **Frontend: React, Angular, Vue**
- **Backend: Node.js, Express.js**

- **Database: MongoDB**

5. Foundation for Modern Frameworks

Learning JavaScript first helps you easily learn:

- **React**
- **Next.js**
- **Vue.js**
- **Angular**

6. High Demand in Jobs

JavaScript is one of the most in-demand programming languages in the world — used by startups, tech giants, and freelancers alike.

7. Easy to Start, Hard to Master

- **Beginner-friendly syntax**
- **Runs directly in your browser**
- **Massive community and resources**

ES6 features :

ES stands for ECMAScript — it's the official standard that defines how JavaScript should work.

In simple terms:

ECMAScript (ES) = The official language specification
JavaScript (JS) = The practical implementation of that specification in browsers

1. let and const

- let and const are block-scoped (unlike var which is function-scoped).
- const is used for variables that should not be reassigned.

```
let name = "Rakshith";

const PI = 3.14;
```

---

2. Arrow Functions

- Shorter syntax for writing functions.
- this keyword is **lexically bound** (it doesn't create its own this).

```
// Traditional function
```

```
function add(a, b) {
  return a + b;
}

// Arrow function
const add = (a, b) => a + b;
```

---

3. Template Literals

- Use backticks (`) to embed variables or expressions using ${}.

```
let name = "Rakshith";
console.log(`Hello, ${name}! Welcome to ES6.`);
```

---

4. Default Parameters

- Assign default values to function parameters.

```
function greet(name = "Guest") {
  console.log(`Hello, ${name}`);
}
greet(); // Hello, Guest
```

---

5. Destructuring Assignment

- Extract values from arrays or objects easily.

```
// Array destructuring
const [a, b] = [10, 20];

// Object destructuring
const { name, age } = { name: "Rakshith", age: 22 };
```

---

6. Spread and Rest Operators (...)

- **Spread**: Expands arrays or objects.
- **Rest**: Collects arguments into an array.

```
// Spread
```

```
const nums = [1, 2, 3];
const moreNums = [...nums, 4, 5];

// Rest
function sum(...args) {
  return args.reduce((a, b) => a + b, 0);
}
```

---

7. Classes

- Simplifies object-oriented programming syntax.

```
class Person {
  constructor(name) {
    this.name = name;
  }

  greet() {
    console.log(`Hello, I'm ${this.name}`);
  }
}

const p = new Person("Rakshith");
p.greet();
```

---

8. Promises

- Handle asynchronous operations more cleanly.

```
const fetchData = new Promise((resolve, reject) => {
  setTimeout(() => resolve("Data loaded"), 1000);
});

fetchData.then(data => console.log(data));
```

---

9. Modules (import/export)

- Helps organize code into separate files.

```
// export.js
export const PI = 3.14;


// import.js
import { PI } from './export.js';
```

---

10. for...of Loop

- Iterates over iterable objects (like arrays, strings, etc.)

```
const nums = [10, 20, 30];
for (let num of nums) {
  console.log(num);
}
```

---

11. Map and Set

- New data structures for unique values and key-value pairs.

```
const set = new Set([1, 2, 2, 3]); // {1, 2, 3}
const map = new Map();
map.set("name", "Rakshith");
```

---

12. Enhanced Object Literals

- Easier way to define object properties and methods.

```
const name = "Rakshith";
const person = {
  name,
  greet() {
    console.log(`Hello, ${this.name}`);
  }
}
```

What is the DOM?

DOM stands for **Document Object Model**.

It is a **programming interface** that allows JavaScript to **interact with, access, and change** the structure, style, and content of a web page.