🟩 Introduction to Node.js 🚀

Node.js is an **open-source**, **cross-platform**, **server-side JavaScript runtime environment** built on **Google Chrome's V8 engine**. It allows JavaScript to be executed **outside the browser**, enabling developers to build **scalable network applications**.

🧠 Key Highlights

- ⚙️ **Open-source** → Anyone can view, modify, and contribute to its codebase.
- 💻 **Server-side JavaScript** → Run JS on your local machine or backend servers.
- ⚡ **Built on V8 Engine** → Compiles JS into machine code for lightning-fast execution.
- 📦 **npm Ecosystem** → Access to the world's largest package manager with millions of libraries.

🔓 What Does Open-Source Mean?

> Node.js is *open-source*, meaning its code is freely available to everyone.

Anyone can:

- 👀 **View:** Check how Node.js works under the hood.
- ✍️ **Modify:** Customize or improve functionality.
- 🤝 **Contribute:** Join a global community improving Node.js daily.

💻 What Does Cross-Platform Mean?

Node.js runs on multiple operating systems **without major code changes**.

✅ Supported Platforms:

- 🪟 Windows
- 🍎 macOS
- 🐧 Linux

This makes Node.js ideal for developers who want portability and flexibility.

⚙️ What is a JavaScript Runtime Environment?

A **JavaScript runtime** is the environment where your JS code executes.

In Browsers:

- JavaScript runs inside browsers like Chrome, Firefox, etc.
- Mainly used for **frontend logic** (UI, interactivity).

With Node.js:

- JavaScript runs **outside the browser**, on the **server**.
- Node.js provides system-level APIs to handle:
    - 🗂️ **File System** (read/write files)
    - 🌐 **Network** (create servers, handle HTTP requests)
    - 🗄️ **Database** (connect to MongoDB, MySQL, etc.)

⚡ Why is Node.js Special as a Runtime?

🧩 V8 Engine

- Compiles JS into machine code for **fast performance**.
- Developed by Google and used in Chrome.

🧰 Built-in APIs

- Node.js provides built-in modules like:
    - `fs` → File system
    - `http` → Server creation
    - `os` → System info
    - `path` → File path utilities
- Helps build powerful apps **without external libraries**.

🚀 Why Do We Need Node.js?

🧠 Single Language for Full Stack

- Use **JavaScript** for both frontend and backend.
- Perfect for **MERN Stack**: MongoDB + Express + React + Node.

⚡ High Performance

- Powered by a V8 engine for **super-fast execution**.
- Handles **thousands of simultaneous connections** efficiently.

🔁 Event-Driven & Non-Blocking I/O

- Node.js utilizes **asynchronous programming**, which means it doesn't block threads.
- Ideal for **real-time applications such as** chat, video, or streaming services.

📈 Scalable for Modern Applications

- Designed for **microservices** and **cloud-native** apps.
- Used in real-time apps like:
    - Chat applications
    - Streaming services
    - Online multiplayer games

🌍 Where Is Node.js Used?

Company	Use Case
🎬 **Netflix**	Fast streaming and server performance
💼 **LinkedIn**	Handles massive concurrent requests
💰 **PayPal**	Scalable backend for secure transactions
🚗 **Uber**	Real-time driver & rider synchronization
📋 **Trello**	Real-time task updates in boards

💡 Million-Dollar Question

🟩 Is Node.js a programming language or a framework?

❌ No!
Node.js is NOT a programming language or a framework.

✅ It is a **JavaScript Runtime Environment**
that allows you to execute JS code **outside the browser**.