ONE PAGE SUMMARY

IT & Software › Backend Development › Node.js

Introduction to Node.js

Unlock the power of server-side JavaScript with Node.js!

DEFINITION

> Node.js is a JavaScript runtime built on Chrome's V8 engine, enabling server-side programming. It allows developers to build scalable network applications efficiently using JavaScript.

KEY CONCEPTS

Asynchronous Programming

Node.js uses non-blocking I/O operations, allowing multiple tasks to run simultaneously without waiting.

Event-Driven Architecture

Node.js operates on an event loop, responding to events and executing callbacks, making it efficient for I/O operations.

NPM (Node Package Manager)

NPM is a package manager for Node.js that allows developers to install and manage libraries and dependencies easily.

Single-Threaded Model

Node.js runs on a single thread, using events and callbacks to handle multiple connections concurrently.

Modules

Node.js uses a modular architecture, allowing developers to create reusable code components and manage dependencies.

EXAMPLES

→ Creating a simple web server using Node.js.

→ Building a RESTful API for a to-do list application.

→ Using NPM to install Express.js for web development.

MEMORY TIPS

★ Remember 'Node' as the 'hub' for JavaScript on the server side.

★ Think of 'NPM' as your toolbox for adding features easily.

★ Visualize the event loop as a conductor managing multiple musicians (tasks).

COMMON MISTAKES

✗ Forgetting to handle errors in asynchronous code.

✗ Blocking the event loop with synchronous code.

✗ Neglecting to use 'const' and 'let' for variable declarations.

QUICK RECAP

Node.js enables server-side JavaScript programming, focusing on asynchronous and event-driven architecture. Understanding its key concepts like NPM and modules is essential for effective backend development.