

Here's a **JavaScript prerequisite roadmap** for Node.js (Beginner → Advanced):

● **Basics (must-know before Node.js)**

1. **JavaScript Fundamentals**
 - Variables (`var, let, const`)
 - Data types (string, number, boolean, null, undefined, symbol, bigint)
 - Type conversion & coercion
 - Operators (arithmetic, comparison, logical)
 2. **Control Flow**
 - `if, else, switch`
 - Loops (`for, while, do...while, for...of, for...in`)
 3. **Functions**
 - Function declaration vs expression
 - Arrow functions
 - Parameters, default values, rest parameters
 - Scope (global, local, block)
 - Hoisting
 4. **Objects & Arrays**
 - Object literals & properties
 - Array methods (`map, filter, reduce, forEach, find`)
 - Destructuring (object & array)
 - Spread & rest operators
-

● **Intermediate (important for Node.js)**

5. **DOM & BOM (optional for Node.js)**
 - Good to know but not heavily used in Node.js (since no browser DOM).
 6. **Advanced Functions**
 - Closures
 - Callbacks
 - Higher-order functions
 7. **ES6+ Features**
 - Template literals
 - `let & const`
 - Modules (`import / export`)
 - Classes & inheritance
 - Promises & `async/await`
 8. **Error Handling**
 - `try...catch...finally`
 - Custom errors
-

● Advanced (very useful in Node.js)

9. Asynchronous JavaScript

- Event loop & concurrency model
- Callbacks vs Promises vs Async/Await
- setTimeout, setInterval, setImmediate
- Promise.all, Promise.race

10. Object-Oriented Programming (OOP)

- this keyword & binding
- Prototypes & prototype chain
- Classes, constructors, inheritance
- Encapsulation, polymorphism

11. Functional Programming Concepts

- Pure functions
- Immutability
- Composition & currying

12. JavaScript Internals

- Execution context
- Hoisting & closures deeply
- Garbage collection basics

⌚ If you **master these JavaScript concepts first**, learning Node.js will feel 10x easier.

Do you want me to create a **step-by-step learning order** (with resources/examples) for these JavaScript prerequisites before you jump into Node.js?