If you're learning Node.js, here are the **main topics** you should cover, from basics to advanced:

---

1. Introduction to Node.js

✅ What is Node.js?
✅ Features & Benefits
✅ Installation and Setup
✅ Running a Basic Script (node app.js)

---

2. Node.js Fundamentals

✅ Understanding Non-Blocking I/O
✅ Event Loop & Event-Driven Architecture
✅ Global Objects (__dirname, __filename, process, Buffer)
✅ Using REPL (Read-Eval-Print Loop)

---

3. Modules & NPM (Node Package Manager)

✅ Built-in Modules (fs, http, path, os, etc.)
✅ Creating Custom Modules
✅ Import/Export (require and import)
✅ Package Management with npm & yarn
✅ package.json and Dependency Management

---

4. File System & Streams

✅ Reading & Writing Files (fs module)
✅ Asynchronous vs. Synchronous Operations
✅ File Streams (Readable, Writable, Duplex, Transform)
✅ Handling Large Files

---

5. HTTP & Web Servers

✅ Creating an HTTP Server (http module)
✅ Handling Requests & Responses
✅ Serving Static Files
✅ Using express.js for Web Apps
✅ Middleware in Express

---

6. Working with Databases

✅ Connecting to MongoDB (mongoose)
✅ CRUD Operations (Create, Read, Update, Delete)
✅ Using MySQL/PostgreSQL with sequelize
✅ Redis for Caching

---

7. Event-Driven Programming

✅ EventEmitter Class (events module)
✅ Handling Custom Events
✅ Event-Driven Architecture in Node.js

---

8. Building RESTful APIs

✅ Creating REST APIs with express.js
✅ Middleware (e.g., cors, morgan, body-parser)
✅ Handling Authentication (jsonwebtoken, OAuth)
✅ API Validation (joi, express-validator)

---

9. WebSockets & Real-Time Communication

✅ Using socket.io for Real-Time Apps
✅ Broadcasting Messages
✅ Building a Chat Application

---

10. Authentication & Security

✅ User Authentication (JWT, bcrypt, OAuth)
✅ Securing APIs (helmet, cors)
✅ Data Validation & Sanitization
✅ Preventing Common Attacks (XSS, CSRF, SQL Injection)

---

11. Microservices & Message Queues

✅ Building Microservices with Node.js
✅ Inter-Process Communication (IPC)
✅ Using Message Queues (RabbitMQ, Kafka)
✅ API Gateway (express-gateway, kong)

---

12. Testing & Debugging

✅ Unit Testing (mocha, chai, jest)
✅ Integration Testing (supertest)
✅ Debugging (node --inspect, chrome devtools)

---

13. Deployment & Scaling

✅ Deploying on Cloud (AWS, Heroku, Vercel)
✅ Process Management (PM2)
✅ Containerization (Docker)
✅ Load Balancing & Clustering

---

14. Performance Optimization

✅ Caching (Redis, memory-cache)
✅ Profiling & Debugging (node --inspect)
✅ Worker Threads for CPU-Intensive Tasks
✅ Optimizing Event Loop Performance

---

15. Serverless with Node.js

✅ Using AWS Lambda, Google Cloud Functions
✅ Writing Serverless APIs
✅ Managing Serverless Deployments

➢ What's Next?

After mastering Node.js, you can explore:

- Full-Stack Development with React, Vue, or Angular
- Blockchain Development with Node.js
- DevOps & CI/CD Pipelines

Would you like detailed tutorials on any of these topics?