**📒 Node.js Notes**

-

**1. What is Node.js?**

* Server-side runtime built on Chrome’s V8 engine.
* Open-source, cross-platform.
* Runs JavaScript outside the browser.
* Perfect for building web servers and networking apps.

**2. Node Package Manager (NPM)**

* Installs and manages libraries.
* Syntax:

A screenshot of a computer program

AI-generated content may be incorrect.

**3. Semantic Versioning:** Because npm set some rules we can use in the package.json file to choose which versions it can update our packages to, when we run npm update

Format: major.minor.patch

* Major → breaking changes
* Minor → new features
* Patch → bug fixes

Examples in package.json:

A black screen with red and green text

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**4. package.json**

Holds project metadata and dependencies.  
Example:

A computer code on a black background

AI-generated content may be incorrect.

**5. Global Objects**

* \_\_filename → current file path
* \_\_dirname → current directory path
* setTimeout(cb, ms), clearTimeout(t)
* console → log info
* process → process info

A black screen with white text

AI-generated content may be incorrect.

**6. Modules -** Module in Node.js is a simple or complex functionality organized in single or multiple JavaScript files which can be reused throughout the Node.js application

Three types:

* Core (http, fs, path, etc.)
* Local (your own files)
* Third-party (via npm)

A green screen with black text

AI-generated content may be incorrect.

Exporting a module:

A screen shot of a computer code

AI-generated content may be incorrect.

**7. Events & EventEmitter**

 Many Node objects generate events (e.g. http.Server emits on a new connection, fs.ReadStream emits when a file opens). Events let you handle async actions without blocking code.

 All event-based objects in Node are instances of events.EventEmitter. Use it to create, fire, and listen for events.

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

**8. HTTP Server**

A computer screen with text

AI-generated content may be incorrect.

**9. File System (fs module)**

* Read, write, delete, rename files.

A screen shot of a computer program

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

**10. Postman:** API testing tool useful for testing Node.js APIs without writing a frontend.

**📒 Node.js Modules & HTTP Servers Notes**

**1. What is a Module?**

* A **self-contained file/package** with its own scope.
* Prevents global pollution, encourages reuse, testing, and versioning.
* Resolution:
  + ./ or ../ → relative paths
  + Bare imports → from node\_modules Cached after first load (single instance).

**2. CommonJS (CJS):** Default in Node (older style).Uses require and module.exports.Loads **synchronously** at runtime.

**Example 1 – Multiple Exports**

A computer screen with text on it

AI-generated content may be incorrect.

**Example 2 – Exports shorthand & destructuring**

A computer code with colorful text

AI-generated content may be incorrect.

**Example 3 – Module Cache**

A screen shot of a computer program

AI-generated content may be incorrect.

**Tips:**

* Avoid hidden shared state unless intentional.
* Circular dependencies only load partially → structure carefully.
* Don’t overwrite exports directly.`

**3. ES Modules (ESM)**

* Uses import/export. Enabled via "type": "module" in package.json or .mjs extension.
* Static imports, supports top-level await.

**Example 1 – Named Exports**

A computer screen shot of a program code

AI-generated content may be incorrect.

**Example 2 – Default + Named**

A computer code on a black background

AI-generated content may be incorrect.

**Example 3 – Dynamic Import + Top-level Await**

A screen shot of a computer code

AI-generated content may be incorrect.

**Interop:**

* ESM can import CJS (default export = module.exports).
* CJS must use await import() for ESM.
* No \_\_dirname/\_\_filename in ESM.

**4. HTTP Basics**

* Request–response over TCP.
* Methods: GET, POST, PUT, PATCH, DELETE.
* Status codes: 2xx (ok), 3xx (redirect), 4xx (client error), 5xx (server error).
* Concerns: logging, auth, validation, CORS, rate limiting.

**5. HTTP Server Examples**

**Example 1 – Native http module**

A computer screen with many colorful text

AI-generated content may be incorrect.

**Example 2 – Express**

A computer screen with text

AI-generated content may be incorrect.

**Example 3 – Fastify**

A computer screen with many colorful text

AI-generated content may be incorrect.

**6. Testing Endpoints**

A screen shot of a computer

AI-generated content may be incorrect.

**7. Best Practices**

* Validate inputs (e.g., JSON Schema).
* Use Helmet, CORS, HTTPS, rate limiting.
* Graceful shutdown and timeouts.
* Don’t block the event loop.