

Backend Javascript Development

Schedule

- Node.js
- Express
- Database
 - **MongoDB**
 - MySQL
- Authentication

Node.js

- Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine.
- Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.
- Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

What you can do with Node.js

- Web server
- IoT (Internet of Things)
- Real-time chat application
- Microservices architecture
- Streaming server - Netflix

“ Any application that can be written in JavaScript, will eventually be written in JavaScript. ”

<p style="float:right">- Atwood's Law</p>

Why Node.js

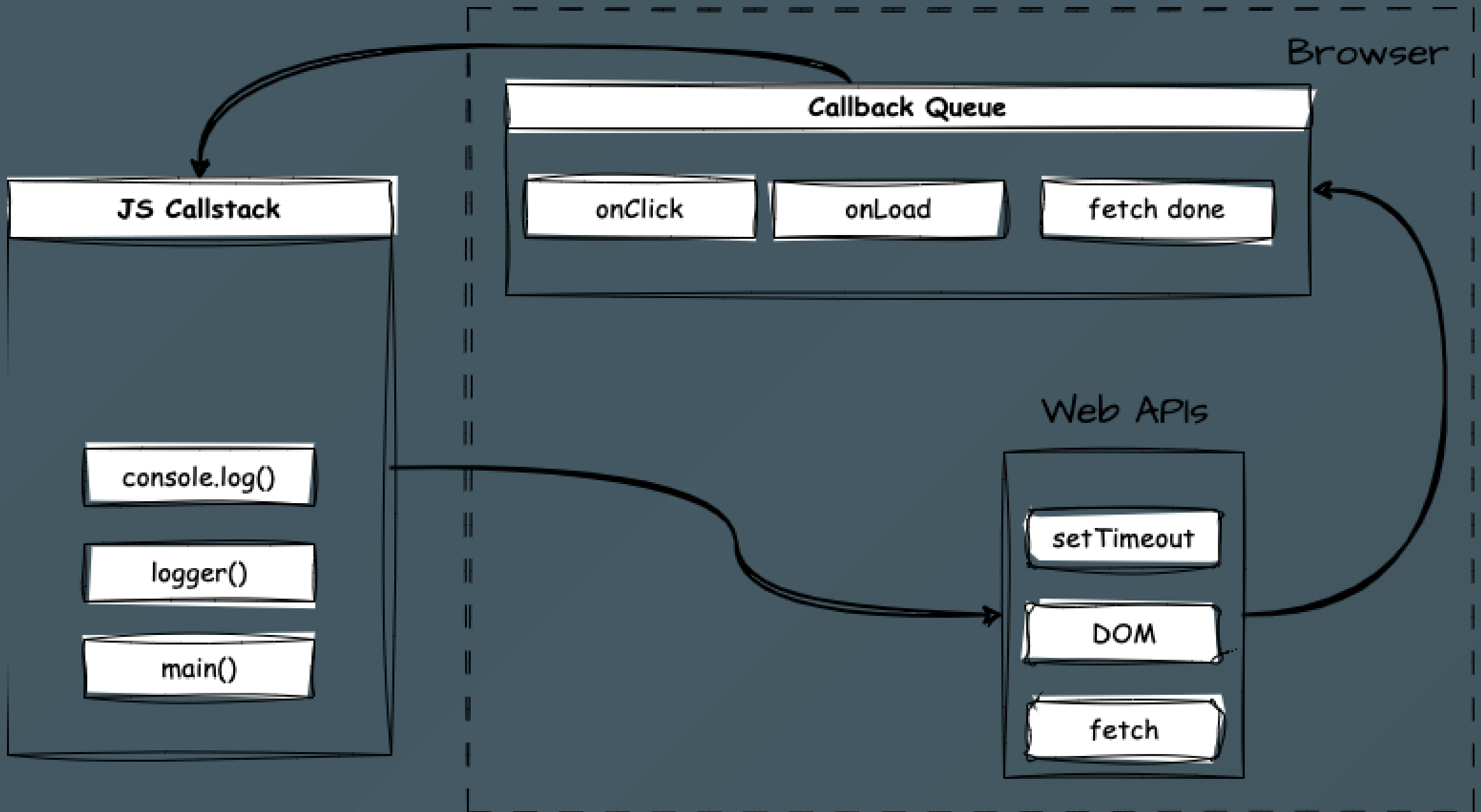
- Scalability
- Performance
- Full-stack JS
- Large ecosystem

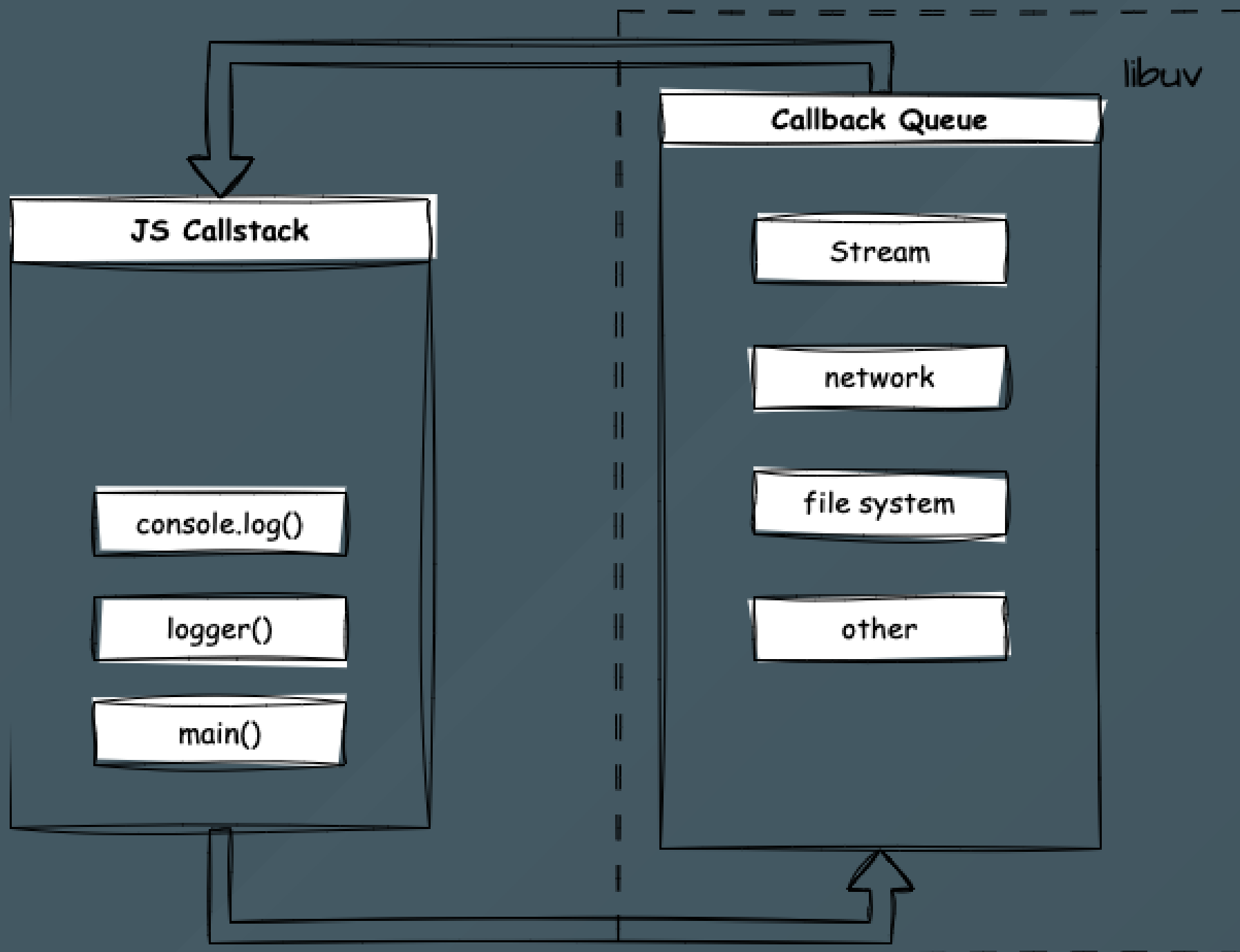
Setup Node.js

- [Node.js](#)
 - LTS
- [nvm](#) - Node Version Manager

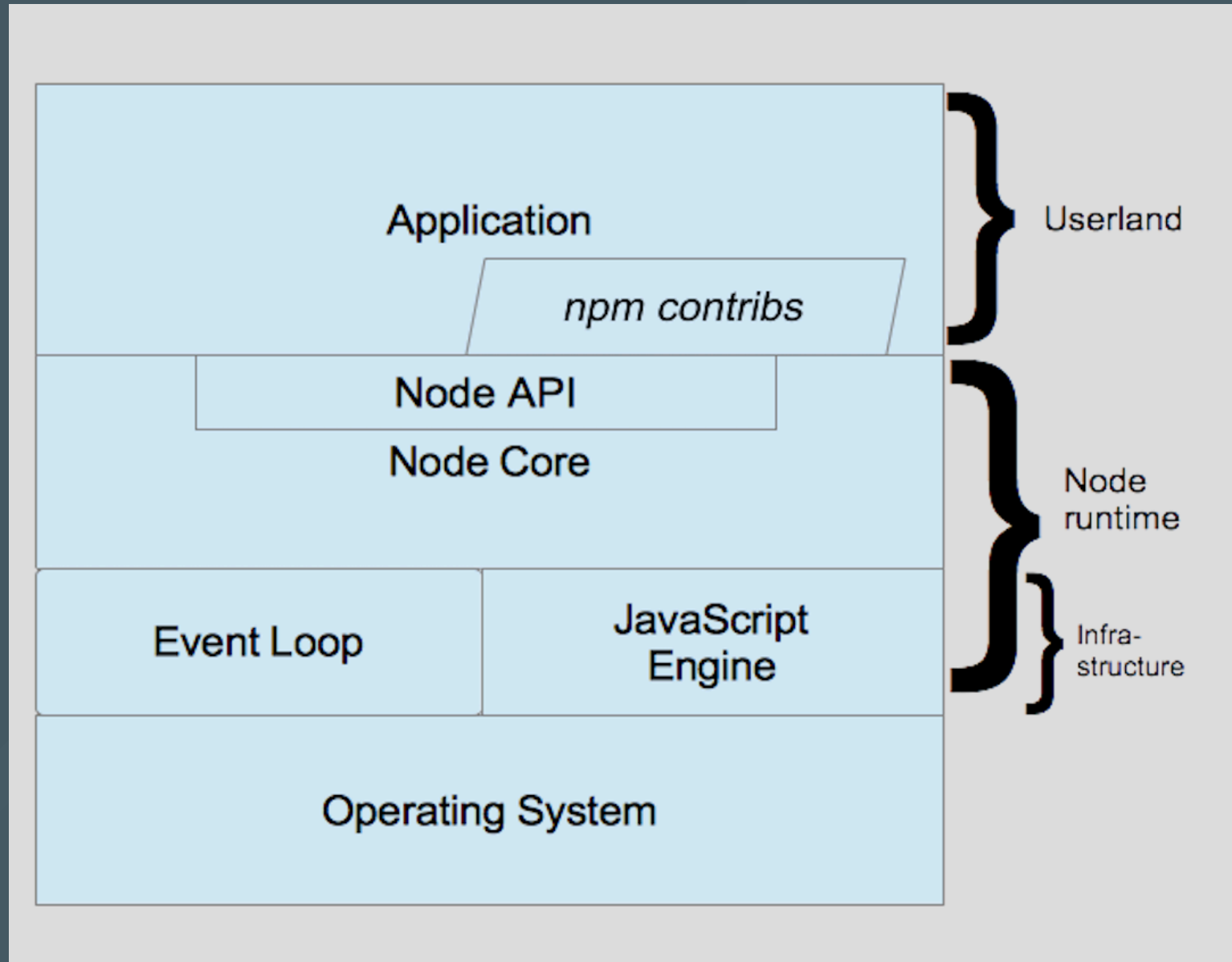
Event Loop

- Node.js is single-threaded.
- Node.js uses **event loop** to handle concurrency.
- Node.js uses **libuv** to implement event loop.





Node.js Architecture



Modules

<p class="mark"> module-use.js esmodule-use.js </p>

- Packaging and Namespacing
- CommonJS modules vs. ECMAScript modules
- To prevent name collision
- `module.exports` and `require()`

Types of Modules

- Core Modules

- `http`, `fs`, `path`, `url`, `querystring`, `crypto`, `os`, `util`, `events`, `stream`, `zlib`, `child_process`, ...

- Local Modules

- User developed modules
- `module.exports` and `require()`

- Third-party Modules

- `npm` (Node Package Manager)

Node Package Manager (NPM)

- NPM is the default package manager for Node.js.
- [npmjs](https://npmjs.com)
- NPM CLI commands
- NPM Packages
 - [lodash](https://lodash.com)
 - [express](https://express.com)
- <https://risingstars.js.org/2023/en>
- <https://npmtrends.com/>

Core Modules

Module	Description
<code>buffer</code>	For I/O operations (file and network)
<code>http</code>	HTTP server and client
<code>fs</code>	File system utilities
<code>path</code>	File path utilities
<code>url</code>	URL parser utilities
<code>child_process</code>	Functions for running external programs
<code>os</code>	Operating system utilities

Global Variables

<p class="mark">global-variables.js</p>

- `__dirname`
- `__filename`
- `exports`
- `module`
- `require()`
- `process`

File System Module

<p class="mark"> fs-use.js fs-esmodule-use.js </p>

- `fs` module provides file system related functionality.
- `fs` module provides both synchronous and asynchronous methods.
 - `readFileSync()` and `readFile()`
 - `writeFileSync()` and `writeFile()`
 - `appendFileSync()` and `appendFile()`
 - `readdirSync()` and `readdir()`

HTTP Module

<p class="mark">http-use.js</p>

- `http` module provides HTTP server and client functionality.
- `http` module provides both synchronous and asynchronous methods.
 - `createServer()`
 - `request()`

Building a Web Server

<p class="mark">web-server.js</p>

- `http` provides `createServer()` method to create a web server.
- Route: a section of code that is responsible for handling a request with a specific HTTP method and URL path.
- Static website vs. Dynamic website

Node Package Manager (NPM)

- NPM is the default package manager for Node.js.
- NPM is one of the largest ecosystem of open source libraries in the world.
- NPM is a command-line utility for interacting with a repository that aids in the package discovery, installation, and removal.
- [npmjs](https://npmjs.com)

Use Module to Build a Static Web Server

1. new folder for the project
2. create `package.json` - `npm init`
3. install npm modules - `npm install <module>`
4. implement server code
5. create static folder and files
6. run the server - `npm start`