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

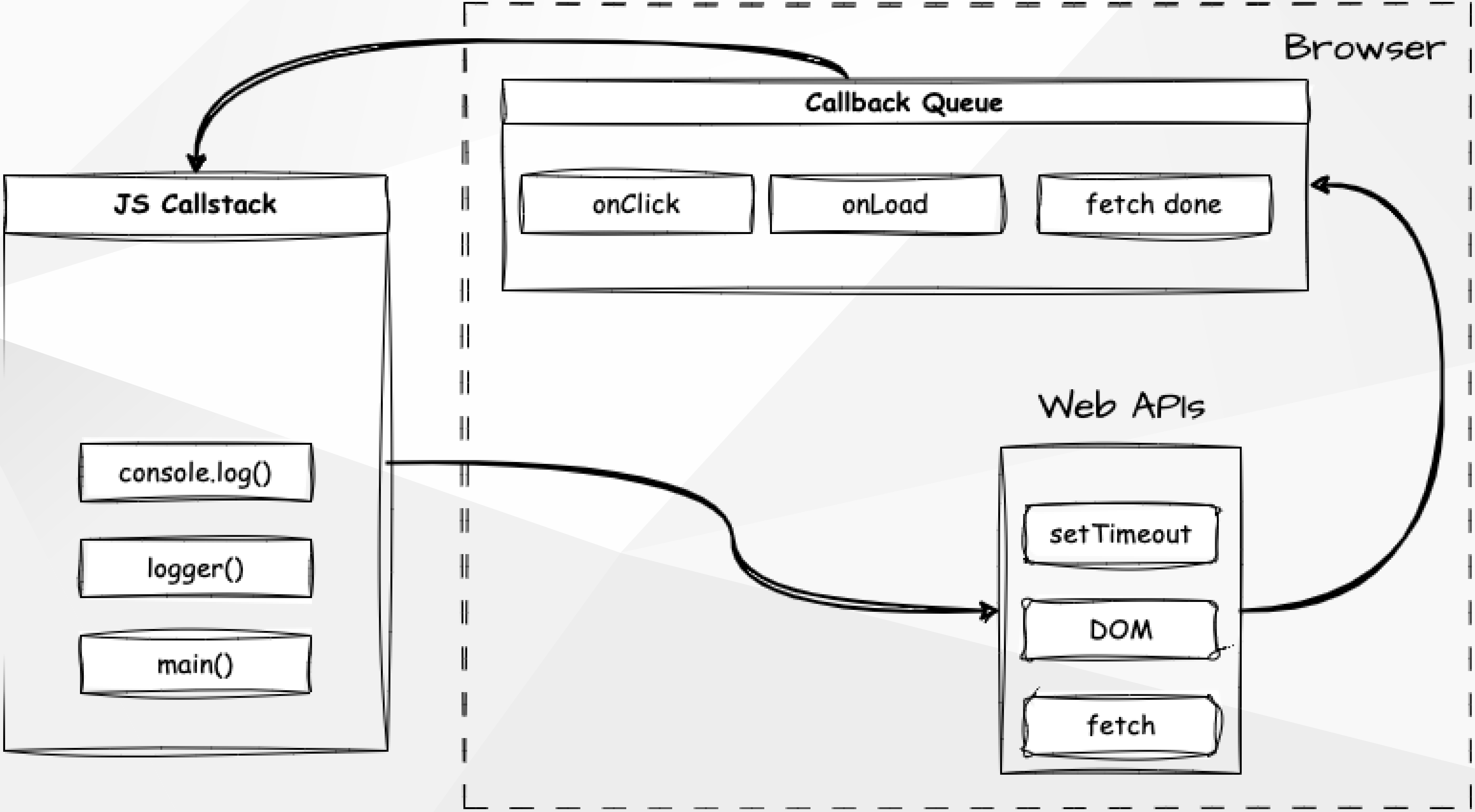
- Web server
- IoT (Internet of Things)
- Real-time chat application
- Microservices architecture
- Streaming server - Netflix

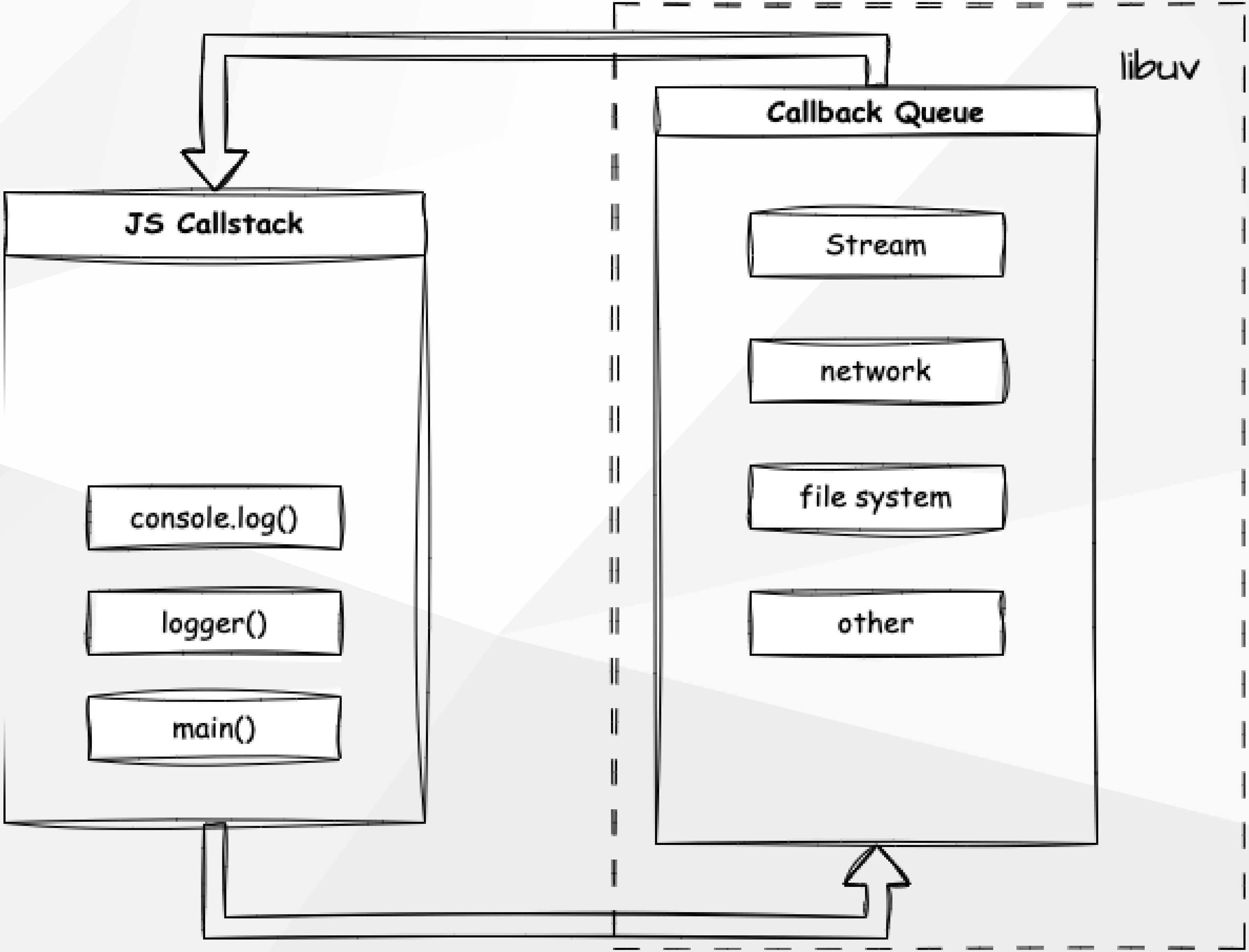
Why Node.js

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

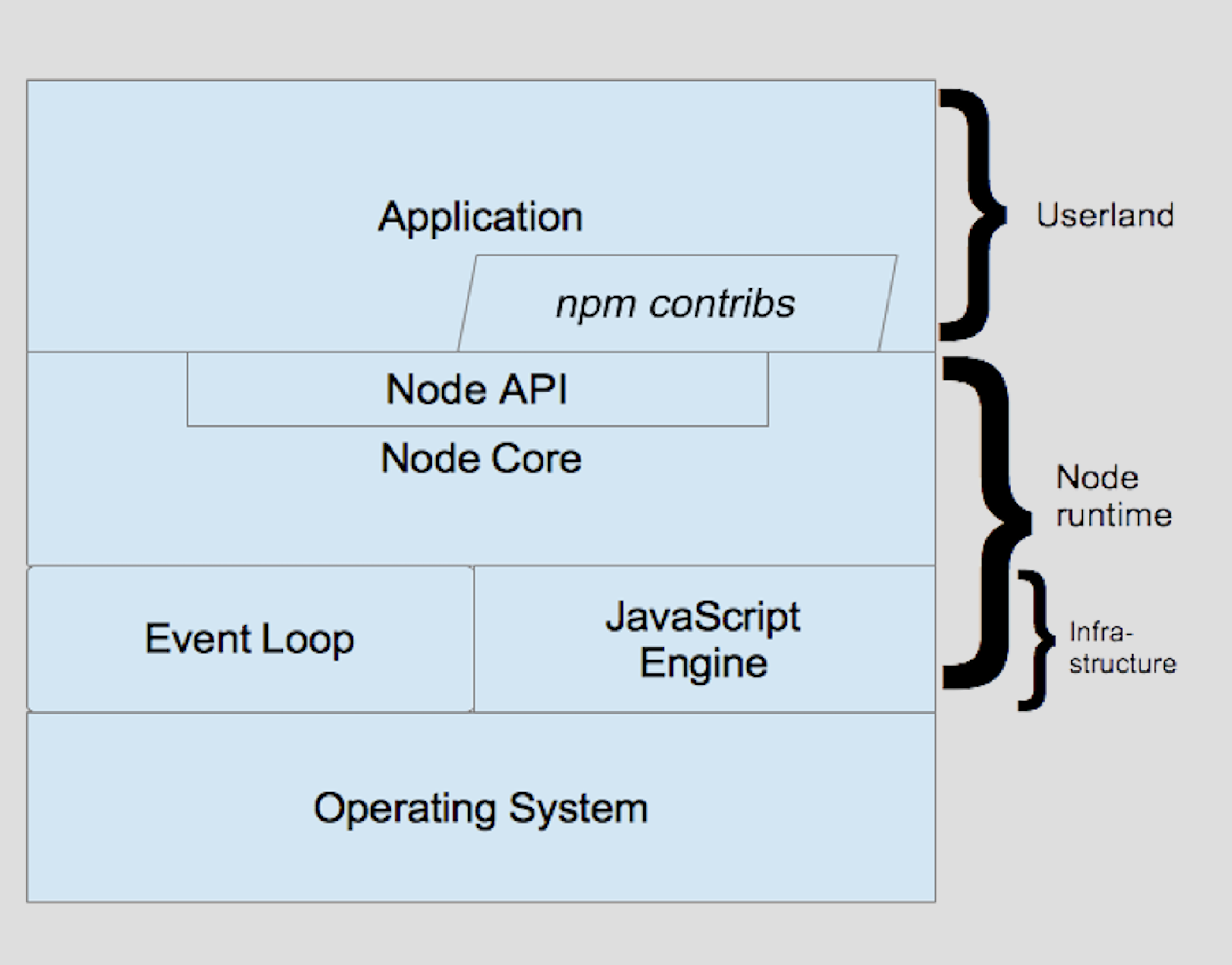
Event Loop

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


Browser
Callback Queue
onClick
onLoad
fetch done
JS Callstack
console.log()
logger()
main()
Web APIs
setTimeout
DOM
fetch


libuv
Callback Queue
Stream
network
file system
other
JS Callstack
console.log()
logger()
main()

Node.js Architecture



Modules

module-use.js esmodule-use.js

- 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)

Core Modules

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

Global Variables

global-variables.js

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

File System Module

fs-use.js fs-esmodule-use.js

- `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

http-use.js

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

Building a Web Server

web-server.js

- `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

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`