NodeJS cheat sheet

Event-Driven Architecture

→ Node.JS employs an event driven architecture where actions on events trigger callbacks.

→ This enables Node.JS to manage multiple connection simultaneously without blocking the execution thread.

Asynchronous Programming

→ It allow operations to be executed independently of the main program flow.

→ Non-blocking I/O operations ensure efficient resource utilization and responsiveness.

npm (node package manager).

→ npm is the default package manager for Node.JS, facilitating the installation, sharing and management of dependencies.

→ It boasts a vast repository of open source packages, enabling developers to leverage existing solutions in their projects effortlessly.

Modules

→ Node.JS follow a modular approach, allowing developers to organize code into reusable modules.

→ Modules encapsulate functionality, promoting code maintainability and reusability.

CommonJS Module system

- Node.js utilizes the commonJS module system for modular development.
- Modules in commonJS are encapsulated units of code with their own scope, facilitating dependency management and code organization.

Streams

- Streams are a key feature of Node.JS, enabling efficient handling of data.
- They allow data to be processed in chunks, enhancing performance and scalability for tasks like file I/O and network communication.

Express.JS

→ It is a popular web application framework for Node.JS, simplifying the development of server-side applications

→ It provides robust routing, middleware support, and a host of features for building scalable web applications.

Promises and Async/Await

- Promises and Async/Await are mechanisms for managing asynchronous operations in Node.JS.
- They provide cleaner and more readable syntax for handling asynchronous code compared to traditional callback-based approaches.