Node JS : Cheat Sheet :-

Defination : Node.js is an open-source, server-side Javascript runtime environment built on chrome's V8 Javascript Engine. It allows developers to run Javascript outside of a web-browser, enabling server-side Scripting to produce dynamic webpage Content before the page is sent to the user's browser.

features :

1) Asynchronous I/O : Node.js uses asynchronous, event-driven architecture to handle I/O operations, allowing it to process multiple requests concurrently without blocking.

2) Serve-Side JavaScript : Node.js enables developers to write server side Code in Javascript leveraging their existing skills and the vast ecosystem or Javascript library and framework.

3) NPM (Node Package Manager) : npm is the default package manager for node.js, providing access to rich ecosystem of over a million reusable packages, libraries and tools using simple command like 'npm install'.

4) Event-driven architecture : Node.js uses event emitters and Callbacks to handle asynchronous operations allowing developers to write Code that responds to events such as incoming http request or data Streams.

5) Modules : Node.js uses a modular architecture, allowing code organizat-ion into reusable modules. Modules encapsulated code logic, making it easier to manage and maintain application. Modules can be created using the CommonJS module system.

7) CommoJS Module System : CommonJS is module System used in Node.js for defining reusable modules. Modules using the require() function. Modules expose functionality using the 'module.exports' or 'exports' objects.

8) Error Handling : Node.js emphasizes error-first Callback Convention for handling errors in asynchronous operations. Error handling can be implemented using try catch blocks, error-first ~~blocks~~, callbacks, or by using libraries like 'async' for more advance error hand-ling patterns.

9) Streams : Streams are a core concept in Node.js for handling data flow. They enable effient processing of large data sets by breaking them into smaller chunks and processing of them asynchrono-usly. Streams can be readable, writable or duplex (both readable and writable).

11) Event Emitters: Event Emitters are objects in Node.js that emit named events asynchronously. They allow communication b/w different part of an application by enabling one part to subscribe to an event and another part to emit that event when certain conditions are met.

12) Express.js: Express.js is a popular web application framework of Node.js, providing features for building robust and scalable web application and APIs. It simplifies the process of handling HTTP requests, routing, middleware integration, and more.

13) Core Modules: Node.js provides a set of Core modules, including 'http', 'fs', 'path' and 'util', among others. These modules offer built-in functionality for common tasks, such as creating web servers, reading and writing files.

14) Web Sockets: WebSockets is a communication protocol that provides full duplex communication channels over a single TCP connection, enabling real-time communication b/w client and Server. WebSockets support in Node.js can be achieved using libraries like 'ws' or frameworks like Socket.io.