Node js Notes:

In Node.js, the "runtime environment" refers to the comprehensive set of tools and components that enable the execution of JavaScript code outside of a web browser. Historically, JavaScript was primarily confined to browser environments for client-side scripting. Node.js changed this by providing a standalone environment for server-side and general-purpose JavaScript execution.

Key components and aspects of the Node.js runtime environment include:

V8 JavaScript Engine: This is the core component, a high-performance open-source JavaScript engine developed by Google for Chrome. Node.js uses V8 to parse and execute JavaScript code, translating it into machine code for efficient execution.

Libuv: This library provides asynchronous I/O operations, such as file system access, networking, and timers. It enables Node.js's non-blocking, event-driven architecture, crucial for handling concurrent operations efficiently.

Core Modules: Node.js comes with a set of built-in modules that provide essential functionalities, such as fs for file system operations, http for creating web servers, path for path manipulation, and events for event handling.

NPM (Node Package Manager): While technically a separate tool, NPM is an integral part of the Node.js ecosystem. It allows developers to easily manage and share JavaScript packages and modules, facilitating code reuse and project development.

Event Loop: Node.js's single-threaded nature relies on the event loop to manage asynchronous operations. It constantly monitors the call stack and callback queue, moving completed asynchronous tasks back to the call stack for execution.

Common js

require() - import

export - module.exports

import

export default

named export

add.js file

```
const add = (a,b)=>{

    return a+b;

}

export const sub = (a,b)=>{

    return a-b;

}

export default add;
```

math.js file

```
import add from './add.js'

console.log(add(1,2));
console.log(sub(1,2));
```