**Node JS**

NPM: Node Package Manager

Var:

Const:

Let:

**Commands**

npm init

npm install <module-name> / npm i <module-name>

nodemon

**Double Equals (==)**:

* The double equals (**==**) is known as the "loose equality" operator.
* It performs type coercion before comparing two values. It attempts to convert the operands to the same type before making the comparison.
* It may not consider the type of values strictly; for instance, it can equate different data types after converting them to a common type.

**Triple Equals (===)**:

* The triple equals (**===**) is known as the "strict equality" operator.
* It checks for both value equality and data type equality without performing any type coercion.
* It requires that both the value and the data type of the operands match for the comparison to evaluate to true.

Core Modules

In Node.js, core modules refer to the built-in modules that come bundled with the Node.js installation. These modules provide essential functionalities, allowing developers to perform various tasks without requiring external dependencies. Core modules can be directly imported into Node.js applications using the require () function.

Example,

const http = require('http');

const fs = require('fs');

const path = require('path');

1. Global
2. Non-Global

Package.Json

Project details, dependencies, code version

Nodemon: Update program without refresh

**Arrow Function**

Arrow functions in Node.js (and JavaScript in general) are a concise way to write functions. They were introduced in ECMAScript 6 (ES6) and provide a more concise syntax compared to traditional function expressions.

Here's a basic example of a traditional function expression:

function add(a, b) {

return a + b;

}

And here's the equivalent written as an arrow function:

const add = (a, b) => a + b;

**How Node.JS Works**

Call Stack ()

Node APIS

Callback Queue

Event Loop

**Express JS**

Express.js is a web application framework for Node.js that simplifies the process of building web applications and APIs by providing a robust set of features and tools. It's fast, unopinionated (gives you a lot of freedom in how you structure your application), and minimalist.

* 1. Routing
  2. Middleware
  3. Template
  4. Static files
  5. HTTP Utility Methods