**Introduction**

**NodeJS Initialization:**

In order to start project with NodeJS, we’ve to first initialize the project with npm (Node Package Manager) by the following command:  
*“npm init”*

This would create a package.json file.

**Module:**

In Node.js, every simple or complex functionalities, variable or objects are modules.

Modules are those which can export and import from one to other file.

There are **three types** of modules in Node.js:-

1. Importing from **local module**: These modules are created by the user and can be imported as:  
   const var = require('./filename.js'); // OR  
   const var = require('./path/filename.js');
2. Importing from **build-In**/**core modules**: These modules are inbuilt in Node.js and can be imported as:  
   const var = require('fs');  
   All Built-In modules in Node.js:  
   [*https://nodejs.org/api/*](https://nodejs.org/api/)
3. Importing from **third party modules**: These modules are installed using a package manager such as npm. Examples of third party modules are express, mongoose, nodemon, etc. These are imported as:  
   const express = require('express');

Example:

**1. File Based Module:** (Old Traditional Way)

For Export: index.js*module.exports = variable;*

const *a* =2;

module*.*exports = a;

For Import: app.js  
const a = require(“./pathToExportFile”);

const *b* = *require*("./index");

*console.log*(b);

2nd way:-   
For Export: index.js*module.exports = variable;*

module*.*exports  = 2;

For Import: app.js  
const a = require(“./pathToExportFile”);

const *b* = *require*("./index");

*console.log*(b);

3. Multiple Module Import:

For Import: app.js

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

*return* a+b;

};

const *subtract* =(*a*, *b*)=>{

*return* a-b;

};

module*.*exports = {

    add,

    subtract,

}

For Import: app.js

const *obj* = *require*("./index");

*console.log*(*obj.add*(5, 5));

*console.log*(*obj.subtract*(4, 2));

**Note:**

* If you want to use require module then you have to save file with ‘.js’ extension.  
  If you want to use import module (From ES6 Import/Export Feature) then you have to save file with ‘.mjs’ extension.
* You must note that you can’t use require and import at the same time in your node program and it is more preferred to use require instead of import as you are required to use the experimental module flag feature to run import program.
* Nodemon package is used for giving real time update of project on terminal. It help developer not to execute project file every time by writing node ./filename.js  
  In Order to use “*nodemon”* module for particular project(Not Globally follow the bellow steps:  
  1. Install:

npm install nodemon

2. Run Nodemon for particular file Using any of the below command:

nodemon ./filename  
 OR  
 npx nodemon ./filename

* We can use port 80, if you want to run application on server only with hostname i.e. “localhost”

**Deployment:-**

1. Move to package.json file and add any convenient “key name” under “script” object  
   Ex:  
   “script”:{  
    “start”: ”node index.js”  
   }
2. Now using command “*npm start”* we can execute the command written under “start” key.