

# Introduction to NodeJS and NPM

**Relevel**  
by Unacademy



# Topics to be covered

- Intro to Nodejs, Brief discussion on V8 Engine.
- Hello World in NodeJS - writing and executing the code using VSCode Terminal.
- Exporting functionalities in NodeJS using module.exports and exports.
- Importing functionalities in NodeJS using require.
- Intro to NPM
- Installing modules locally and globally using npm.
- Uninstalling modules using npm.
- Initialising an NPM project, brief discussion on package.json.
- What are core modules of Node?
- Introduction to the FS core module of Node.



# Introduction to Node JS

- A JavaScript Runtime Environment
- Built on Chrome's V8 Engine!
- Light, scalable and open-source language platform
- Used for Server-side development



# Hello World in Nodejs

1. Create a file - helloworld.js

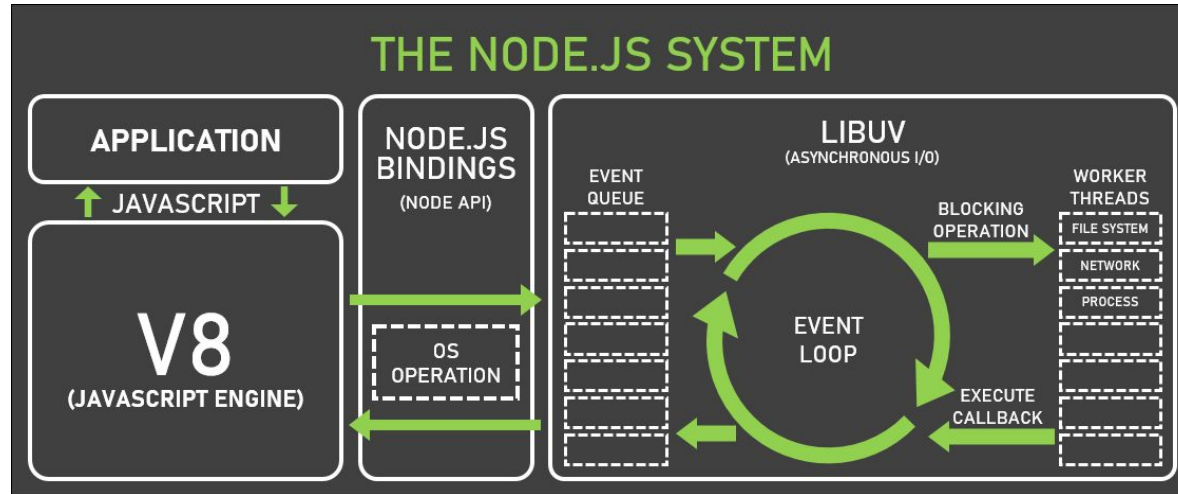
```
console.log("Hello World");
```

2. And save the file after adding this line
3. Run the file - node helloworld.js



# NodeJS Ecosystem and Event Loop

The Node.js JavaScript code runs on a single thread. There is just one thing happening at a time. The **Event Loop** is one of the most important aspects to understand about Node.js. Because it explains how Node.js can be asynchronous and have non-blocking I/O, and so it explains basically the "killer app" of Node.js, the thing that made it this successful.



# Exporting and Importing functionalities in NodeJS

- Node.js has a built-in module system.
- A Node.js file can import functionality exposed by other Node.js files.
- Let's see how to expose a functionality

```
// car.js
```

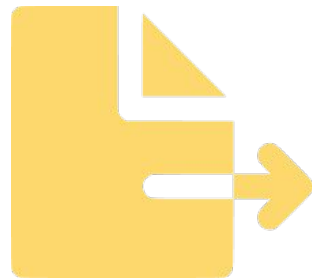
```
const car = {
```

```
  brand: 'Ford',
```

```
  model: 'Fiesta'
```

```
}
```

```
module.exports = car
```

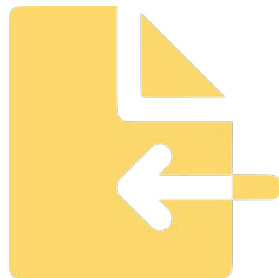


# Exporting and Importing functionalities in NodeJS

- And to use this exported object, we can import it as below

```
// index.js
```

```
const car = require('./car')
```



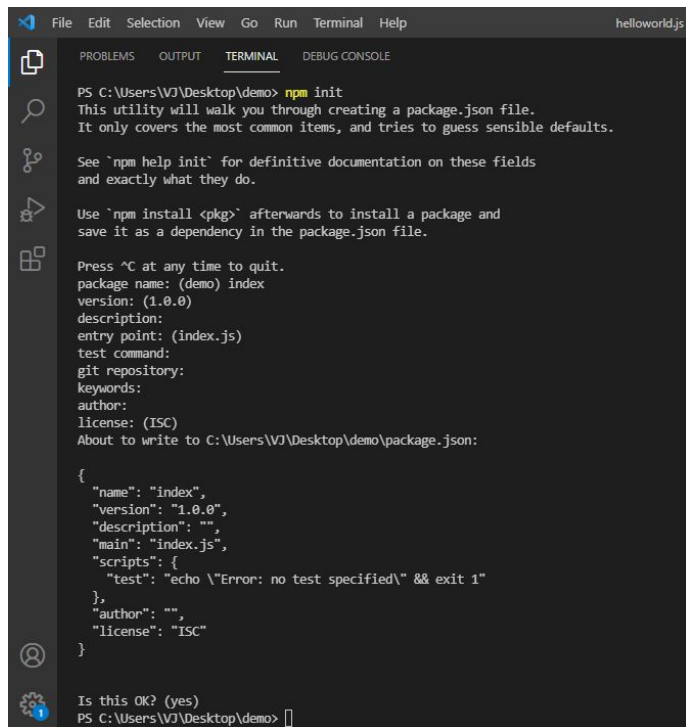
# Introduction to npm

- NPM stands for Node Package Manager.
- npm is the standard package manager for NodeJS
- npm is the world's largest software registry





# Initialising an NPM Project



```
File Edit Selection View Go Run Terminal Help helloworld.js -
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\VJ\Desktop\demo> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (demo) index
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\VJ\Desktop\demo\package.json:

{
  "name": "index",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

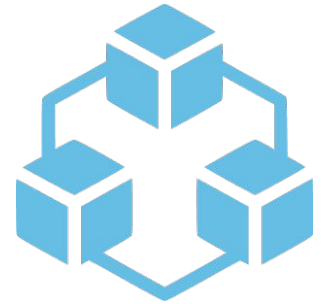
Is this OK? (yes)
PS C:\Users\VJ\Desktop\demo> 
```

# Initialising an NPM Project

```
{ } package.json X
{ } package.json > ...
1  {
2    "name": "index",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC"
11 }
```

# Node Module Types

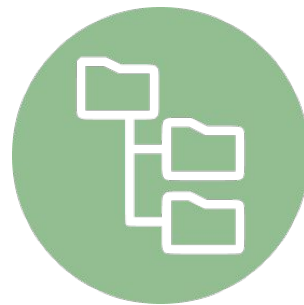
- There are three types of modules in NodeJS
  - Core Modules - Provided by nodeJs Out-of-the-box. Need not to be installed
  - Local Modules - Those are user-defined modules
  - Third Party Modules - Those are module which are created by someone else and available for use by all



# fs module in NodeJs

- fs stands for File System
- fs module is one of the core modules provided by NodeJS to interact with files
- You do not need to install it to use it.
- fs module provides a bunch of methods to deal with files, refer <https://nodejs.dev/learn/the-nodejs-fs-module>
- To read a file

```
var fs = require('fs');  
  
fs.readFile("filename.txt", "utf8",  
function(err, data){  
    console.log(data);  
})
```



## Practice/HW



Write a nodeJS program that would read content from the file and also append new content into the file.



Explore setTimeout and setInterval and write a program that would make use of setTimeout and call a function after 10 sec once the app starts, and setInterval would print number 1 to 10 after interval of 5 secs and then it should stop.

## Next Class



In the next class we would be looking into Client Server Architecture, HTTP and REST. And we would also be setting up a HTTP server using nodeJS.

**Thank you!**