NPM Why what and How

→ Hitesh Choudhary &
Coder Coder

→ NPM is a short topic 😊
→ NPM Solves a very Big Problem

(Node Package manager)

→ npm -v or node -v

why ?

Save a user with unique ID

We can use the code written by anyone else
We pull the down and Configure it then
and place all files at We can use it this is a bit lengthy
right place and

this is Solved by npm

Download Code
Configure it

Solution

that every Programmer will use
Set Standards and
use npm

has to follow
if he/she wants
to design anything
in npm
now then can easily
do → npm install

😊

no need for hassle to
Configure and more.

Refer to npm docs → Create a folder in Vscode
→ Use built in editor

→ npm init → initialize (or npm init -y to install with default values)

(this creates a file initializing our program or product that I will using 3rd party things you need data then here which I want.

default Package name will be you folder name but not necessary.

npm uses Semantic Versioning aka numbers

1.0.0
- Patch releases
- minor releases
- major releases

In Description → what Package does.

what is entryPoint ?

=)) from where the Program Starts from where the Software can Start.

Eg: index.js or App.js or anything

Best Practice.

work with these file with these name

git repository:

↓

To connect your package to your git repo.

author : Santosh Mishra.

how Package.json file is Created with what data you have given.

main is the entry Point.

→

npm Config → these are global Configuration

→ npm Config

→ npm Config set key Value

↓

To set as global Value

when we Started ~~coz go~~ when we did npm init we got some License type and something they are step up here.

eg. how to edit global Config :

→ npm ↓ set init.author.email "example@gmail.com"
config

→ npm Config list

→ npm Config delete init.author.email

Hands on Versioning

→ npm install ~~uuid~~ uuid → you will see this in Package.json file you will see version there.

→ npm uninstall uuid

- introduced in version 5 of npm.

Package-lock.json is more Verbose

1) ↓ this does something like time machine you can go back and forth in ~~Versioni~~ Versioning.

2) It is very important for Production. used in Production

Versioning →

NPM

"uuid": "7.0.3" will install 7.1.3

" : "^2" → will install 7.0.5

" : "*" → will install latest

" : "7.0.3" → will install exact Version

^ → this Carrot sign means get the latest Version in minor release but don't change ~~Ver~~ major release.

→ '~' → get the latest Patch release Version don't change the major and minor release.

Command to do this all-ls

→ npm install uuid@7.0.3
or ~7.0.3
or ^7.0.3

you will always see changes in your Package.json

"Script" → we can have Script to run Some thing when we want

Run by the "npm start" Command

"Scripts":
"start": "node main.js or app.js".

So run the application.

→ node main.js or app.js etc. (entry point)

But Before

now you don't need to do it

now you can just type

→ npm start ✓

if you run this Command it will run everything inside Start under the Script Section.

→ nodemon search on google and by this you will be able to use

~~node~~ node Command

file → node main.js

↓

i think so

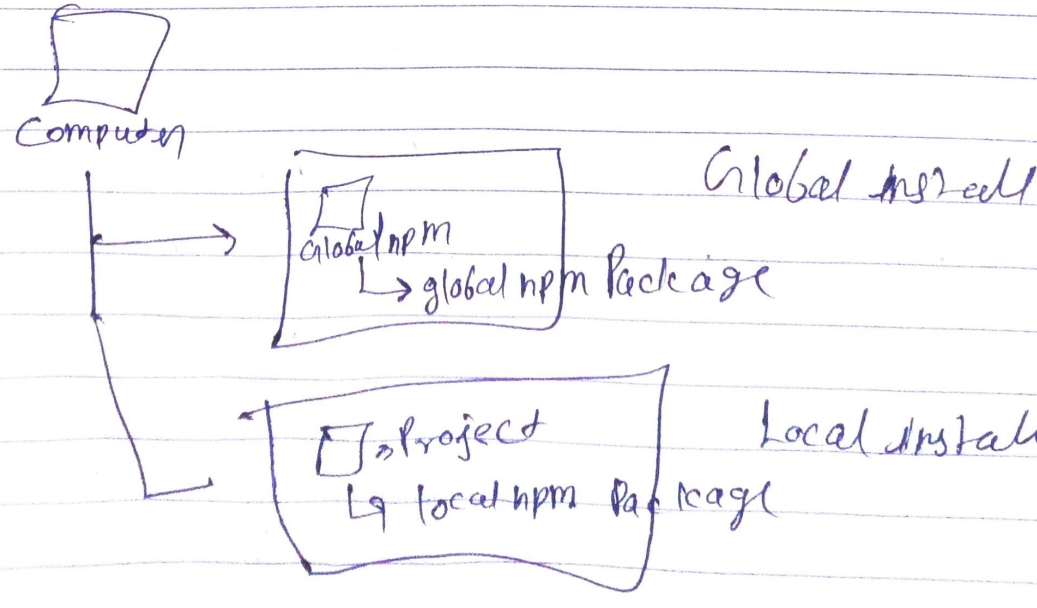
by me.

→ there are kind of global Packages

→ npm install -g packagename.

Read npm documentation.

with the help of npm you can easily use other persons code in yours without breaking anything



Global Package can be accessed from any directory.

Packages are available at npm registry

To install the package.json file

→ npm install package.json

→ In node_module folder npm saves all Packages files.

Bonus →

A package can be dependent on other & And the other one is dependent on other.

→ npm list → To list all Packages

→ npm view chalk versions.

→ Shows all the version available

→ To update npm or npm Packages

→ npm update

The caret sign will update minor and patch version of the major version.

Package-lock.json file is there to save you to from screwing up for example you clone a git repo in which the Package.json file used a version dependencies used a version V.1.0.0 and in copied git repo you delete the Package-lock.json and while installing the Package.json in your machine due to the Carot ~~Carot~~ Sign (^) under dependencies in Package.json file "~~1.0.0~~" "^1.0.0" it will the update the dependencies ~~of~~ package and the newer version will have different updates and roll backes of Commands and there you will screw up but Package-lock.json doesn't let it to update and installs the Package with same ~~dupe~~ dependencies which was used while making the ~~of~~ package. the man who made Package.json file with dependencies version of his choice are all same used or you can say copied him.

Commands →

→ npm -V

→ npm init or npm init -y

→ npm list

→ npm Config

→ npm Config set key value

→ npm Config delete key

→ npm Config list

→ npm install packagename

→ npm install pkg@version (e.g. pkg@7.6.0)

pkg@ ★

pkg@ ^1.0.0

pkg@ ~1.0.0

pkg@ 1.0.0

→ npm uninstall Package name
→ npm install -g Package name
→ npm update → to update all Packages only their minor & Patch Version

→ npm update version
→ npm view package-name version
→ npm view package-name versions.
→ npm install Package.json
→ npm Start
or any which you have set under Scripts in Package.json file.