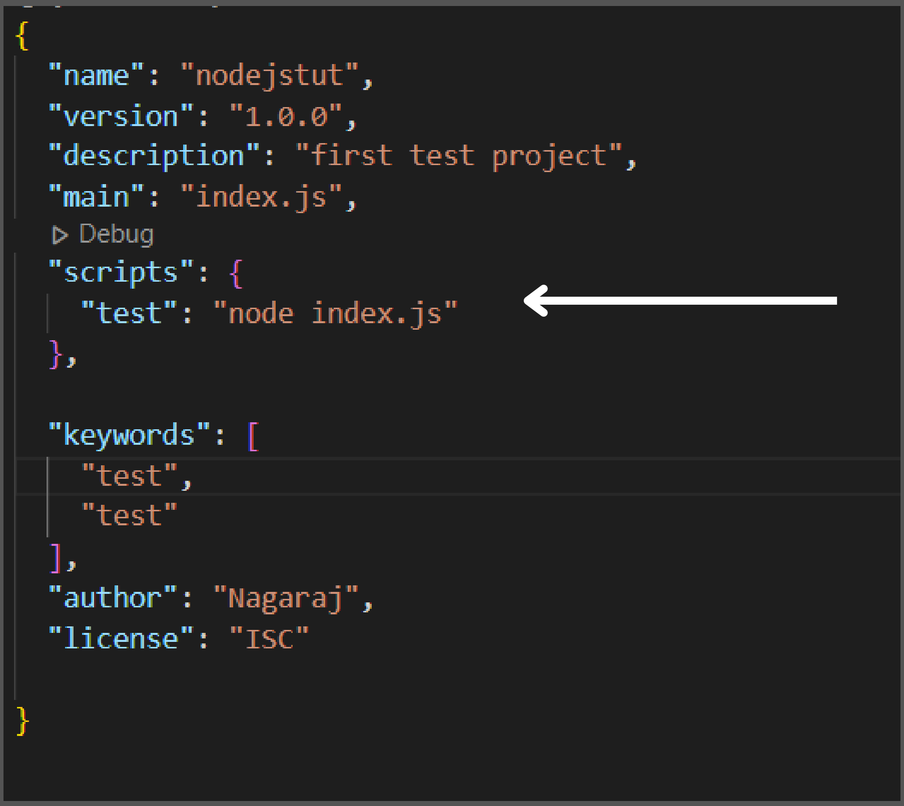
Package.json

- A package.json is a file that contains related information to that project like license, name, entry point and dependency.
- It is initiated by command **npm init.**
- After that it will ask details like name, entry point, license, author name, description, git repo (if any). After entering all details it will generate the package.json

Initial package.json

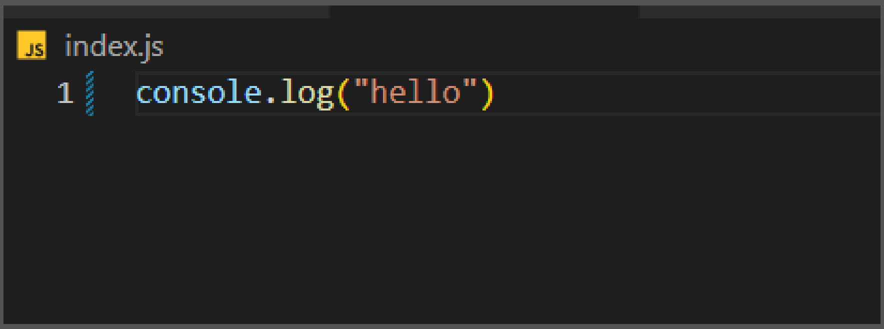


```
{
  "name": "nodejstut",
  "version": "1.0.0",
  "description": "first test project",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "test": "node index.js"
  },

  "keywords": [
    "test",
    "test"
  ],
  "author": "Nagaraj",
  "license": "ISC"

}
```

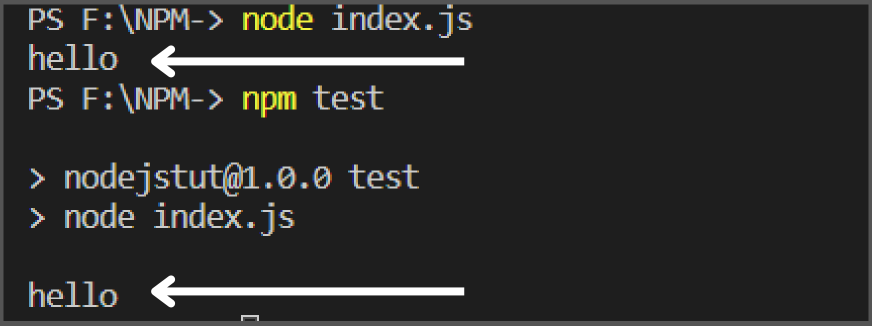
Index.js File



```
index.js
console.log("hello")
```

Now the index.js executed by two commands

1. node index.js
2. npm start (which is refer to node index.js in package.json)

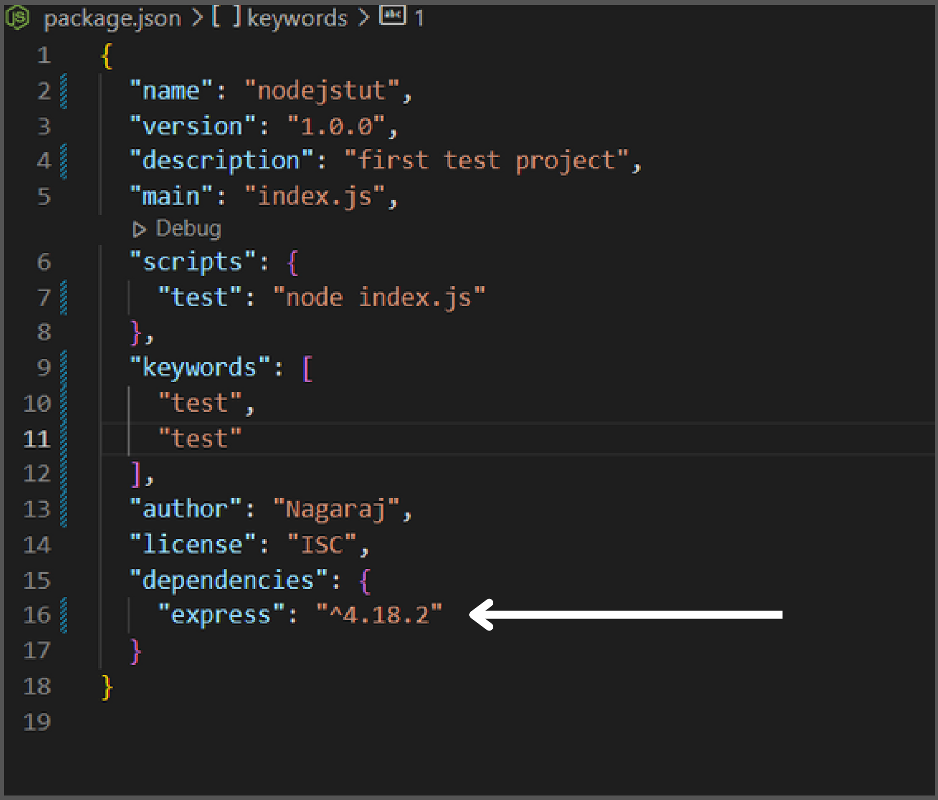


```
PS F:\NPM-> node index.js
hello
PS F:\NPM-> npm test

> nodejstut@1.0.0 test
> node index.js

hello
```

- Now after installing npm install express



```json
{
  "name": "nodejstut",
  "version": "1.0.0",
  "description": "first test project",
  "main": "index.js",
  "scripts": {
    "test": "node index.js"
  },
  "keywords": [
    "test",
    "test"
  ],
  "author": "Nagaraj",
  "license": "ISC",
  "dependencies": {
    "express": "^4.18.2"
  }
}
```

In npm the version denoted by three digits.
X.Y.Z
X - Major Version
Y - Minor Version
Z - Patch Version

From above package.json it is clearly visible that 4.18.2 version of express was installed.

What is meant by ^ , ~ prefix in package version?

Suppose after 6 month you imported the project into another laptop.

Now new laptop make utilize of this package.json file and install the dependency of these project.

Express package is frequently updated by express team. After 6 months this could not be a latest version

- ^ - Tells hey install the latest minor version - 4.latest.latest
- ~ - Tells hey install the latest patch version - 4.18.latest

These was one of the cool feature of NPM.

Now if you want hey, I want the exact version which is 4.18.2 in new laptop. Simple keep **"express": "4.18.2"**.

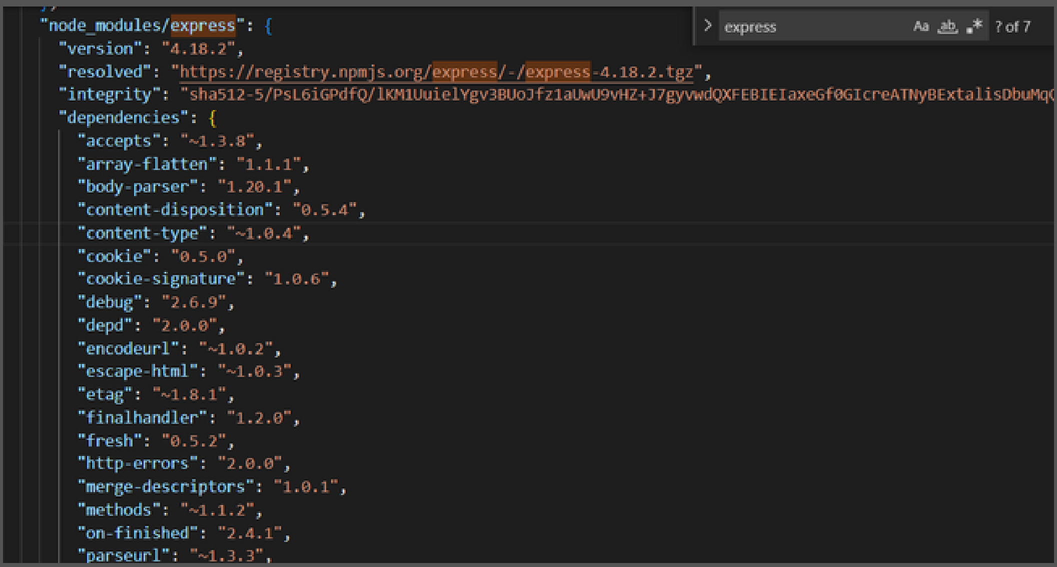
- The default prefix is ^.

Now consider there are 100 packages. You want exact version on our new laptop. Is I have to manually edit 100 times in package.json?

Answer is Big No.

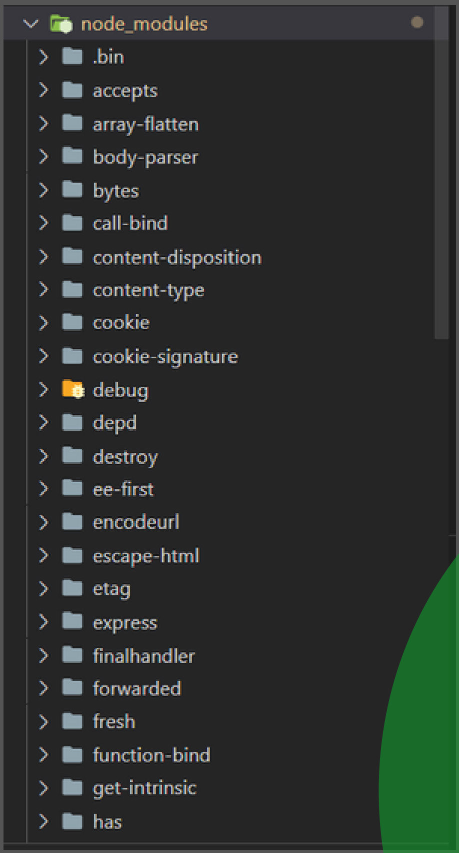
Package-lock.json

- After installing express, there is one new folder created called node module which contains package.
- Even though I install express only, these are dependency and there dependencies got installed also.
- And also one more file called **package-lock.json** was generated.
- These file contains all the information about the package details with exact version. It will not only contain package whatever we installed.
- It also contains all dependency details up to root level.
- So we can replicate the same setup with same version in another environment.



```
"node_modules/express": {
  "version": "4.18.2",
  "resolved": "https://registry.npmjs.org/express/-/express-4.18.2.tgz",
  "integrity": "sha512-5/PsL6iGPdfQ/lKM1UuielYgv3BUoJfz1aUwU9vHZ+J7gyvwdQXFEBIEIaxeGf0GIcreATNyBExtalisDbuMq
  "dependencies": {
    "accepts": "~1.3.8",
    "array-flatten": "1.1.1",
    "body-parser": "1.20.1",
    "content-disposition": "0.5.4",
    "content-type": "~1.0.4",
    "cookie": "0.5.0",
    "cookie-signature": "1.0.6",
    "debug": "2.6.9",
    "depd": "2.0.0",
    "encodeurl": "~1.0.2",
    "escape-html": "~1.0.3",
    "etag": "~1.8.1",
    "finalhandler": "1.2.0",
    "fresh": "0.5.2",
    "http-errors": "2.0.0",
    "merge-descriptors": "1.0.1",
    "methods": "~1.1.2",
    "on-finished": "2.4.1",
    "parseurl": "~1.3.3",
```

Package-lock.json



```
node_modules
  .bin
  accepts
  array-flatten
  body-parser
  bytes
  call-bind
  content-disposition
  content-type
  cookie
  cookie-signature
  debug
  depd
  destroy
  ee-first
  encodeurl
  escape-html
  etag
  express
  finalhandler
  forwarded
  fresh
  function-bind
  get-intrinsic
  has
```