

Installing NPM Modules

**Open the project
make sure your in the
project root folder.**

Open VSCode Terminal.

npm init

npm init

Creates a package.json file

package.json

**records the external modules
used by a project and the
project metadata**

your package.json should look similar
but not exactly the same

```
{  
  "name": "project",  
  "version": "1.0.0",  
  "main": "index.html",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "your name",  
  "license": "MIT",  
  "description": "simple demo of setting up vscode project",  
  "keywords": "npm, installing modules, setup"  
}
```

**There are types of modules
that you can install.**

devDependencies

Development or dev dependencies are installed and used while your building your project. However when you go to deploy your project to production the project doesn't actually need the modules to run.

An example of this type of dependency would be a local dev server, a css preprocessor, testing tools or a linting tool. Though these tools are useful to us in development and help speed up our workflow they are not required by your app at runtime.

dependencies

Dependencies are the opposite of devDependencies. They are required by your app at runtime.

For example if your building a web app using react.js then the app relies on the react code to work. Without the react.js code (module) your app would fail.

Installing dependencies

Use the npm install for your project dependencies

```
npm install
```

Install React

```
npm install react
```

your package.json should look similar
but may not exactly the same version number

```
{  
  "devDependencies": {},  
  "dependencies": {  
    "react": "^17.0.1"  
  }  
}
```

Uninstall React

```
npm uninstall react
```

your package.json should look similar
but may not exactly the same version number

```
{  
  "devDependencies": {},  
  "dependencies": {}  
}
```

Installing devDependencies

Use the `npm install -D` for your project devDependencies. The difference between the two installs is just the `-D` flag so pay attention. You can always uninstall a dependency using `npm uninstall`.

```
npm install -D
```


Install live-server

```
npm install -D live-server
```

your package.json should look similar
but may not exactly the same version number

```
"devDependencies": {  
  "live-server": "^1.2.1"  
},  
"dependencies": {}
```

Uninstall live-server

```
npm uninstall live-server
```

your package.json should look similar
but may not exactly the same version number

```
"devDependencies": {},  
"dependencies": {}
```

Let's Go Write Code