JS

NPM

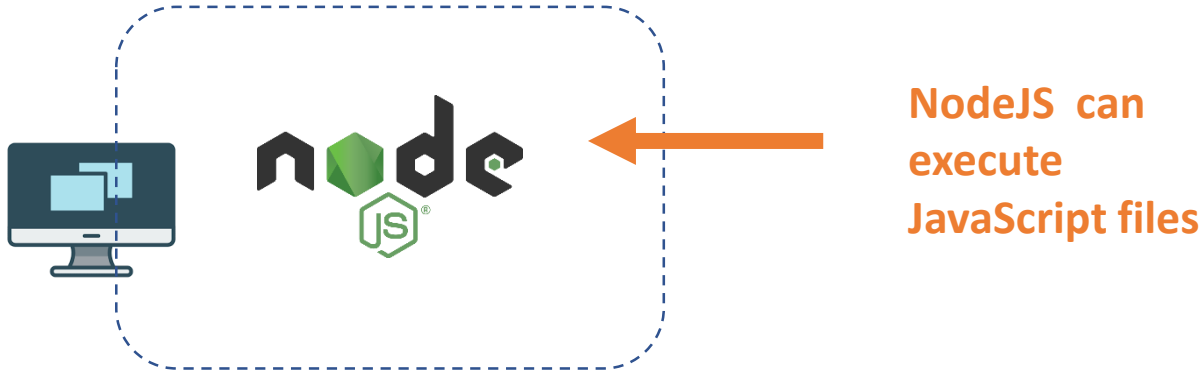
External packages - dependencies

OBJECTIVES

- ✓ What **NPM** is used for ?
- ✓ Install an **NPM PACKAGE**
- ✓ Install **DEPENDANCIES** on an existing project

WHAT DID WE LEARN LAST SESSION?

A back end is basically a **program** running **all day** on a computer

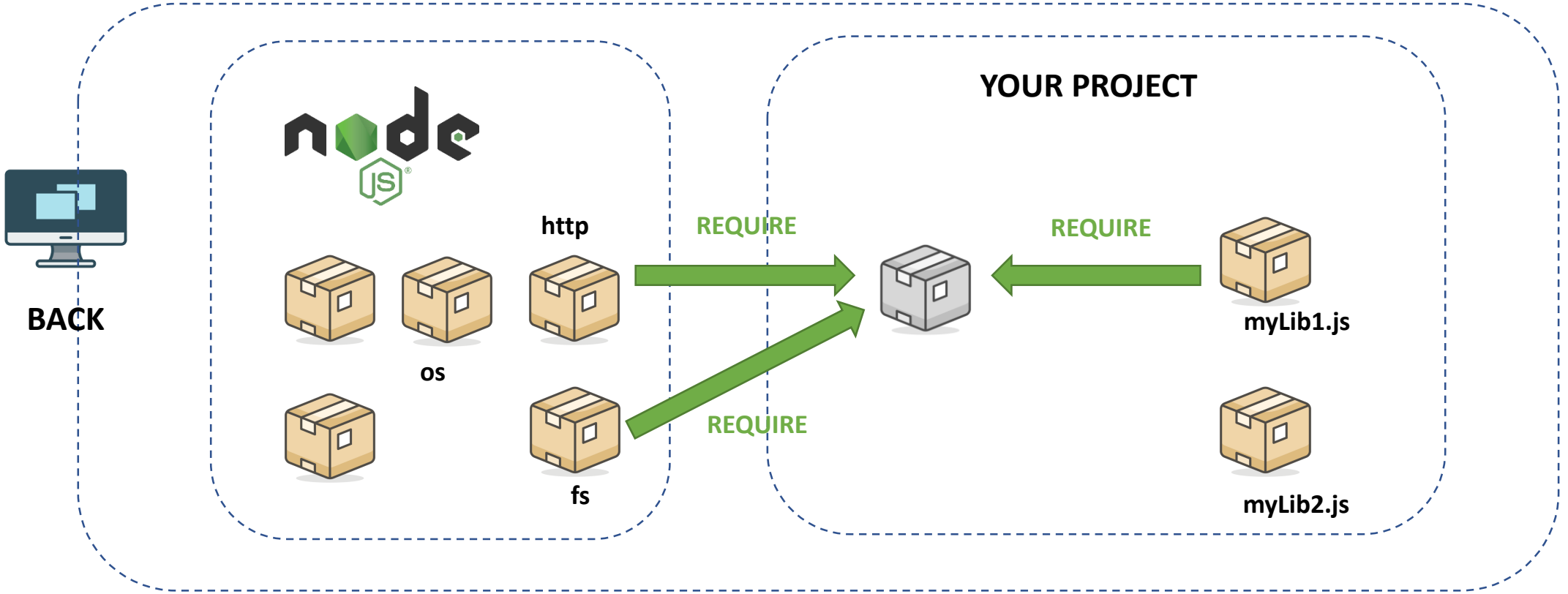


NodeJS can execute JavaScript files

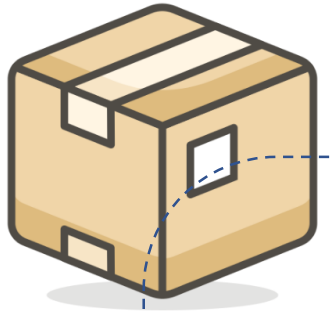
(BACK END)

To write your **code**, you can use other **modules**

- Use Node built-in modules
- Use your own modules



You can export **variable** or **functions**



library.js

```
let a= 2
let b= 2
let c= 2

module.exports.a =a;
module.exports.b =b;
```

main.js

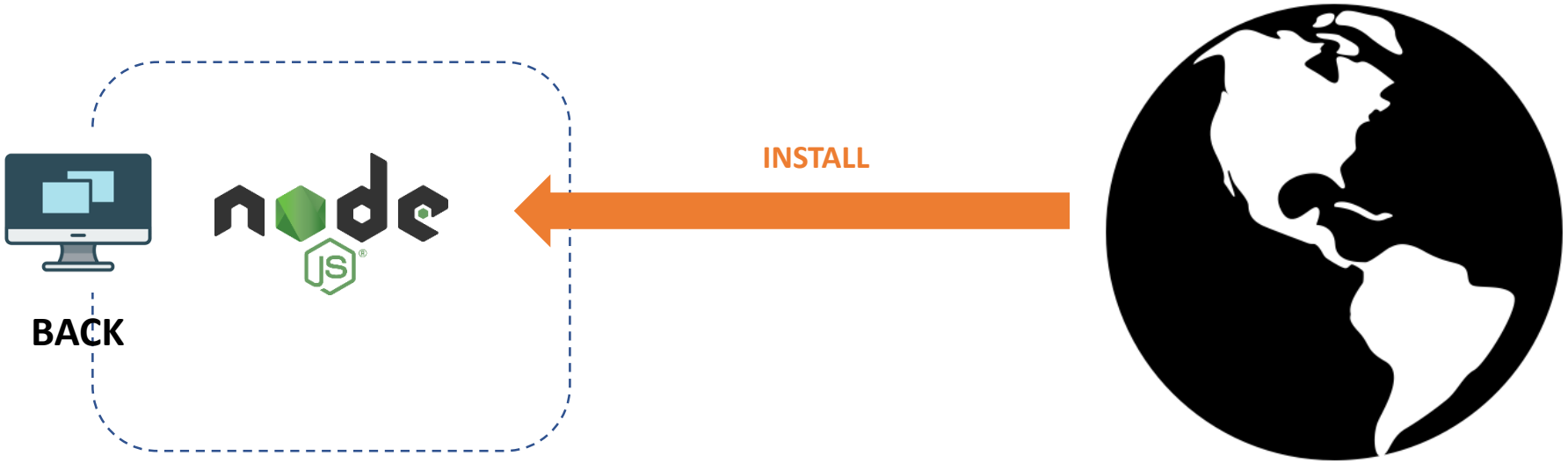
```
let lib = require("./library.js");
```

lib.a ⟵ AVAILABLE

lib.b ⟵ AVAILABLE

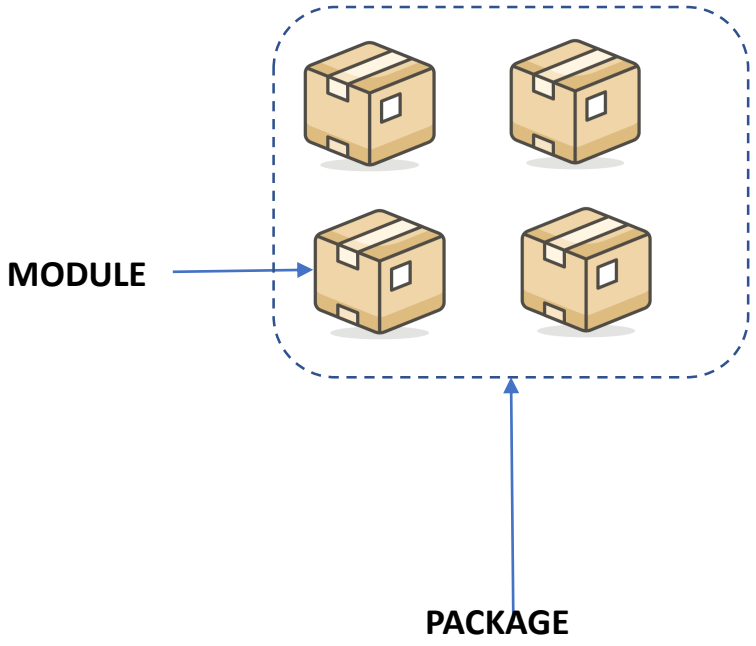
lib.c ⟵ NOT AVAILABLE

What about installing a MODULE from the WEP?

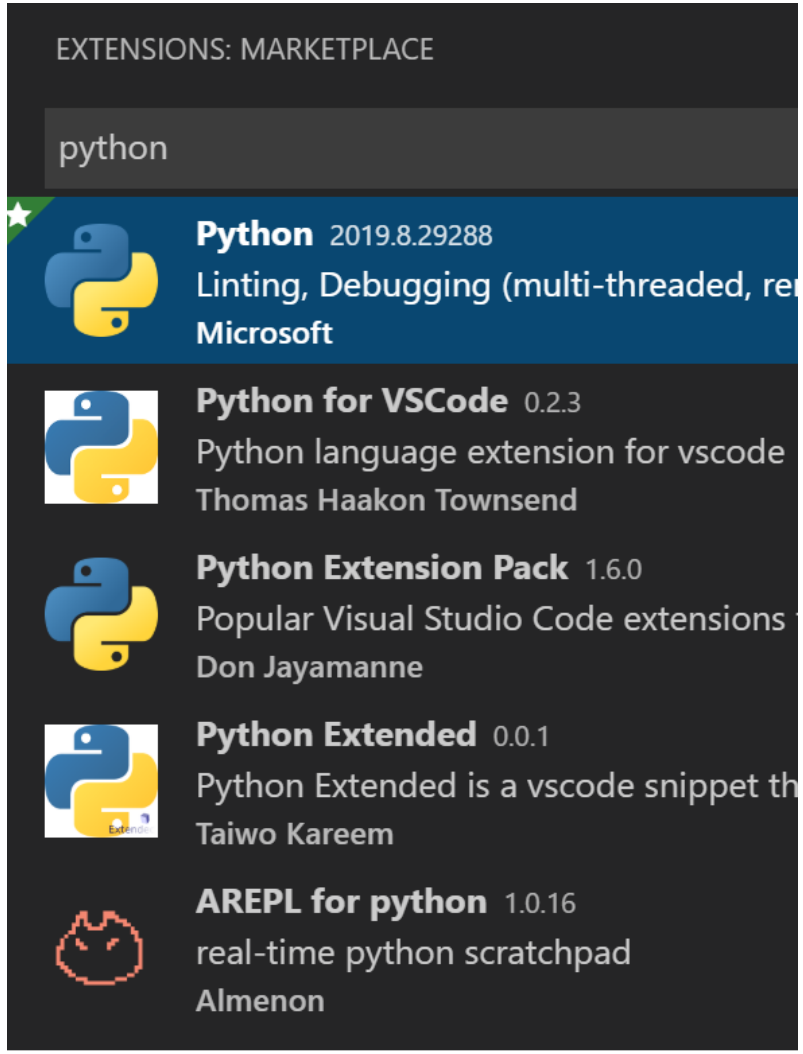


npm
PACKAGES
ARE THE TOOLS
OF YOUR APPLICATION
SEE ALL PACKAGES HERE
https://www.npmjs.com/

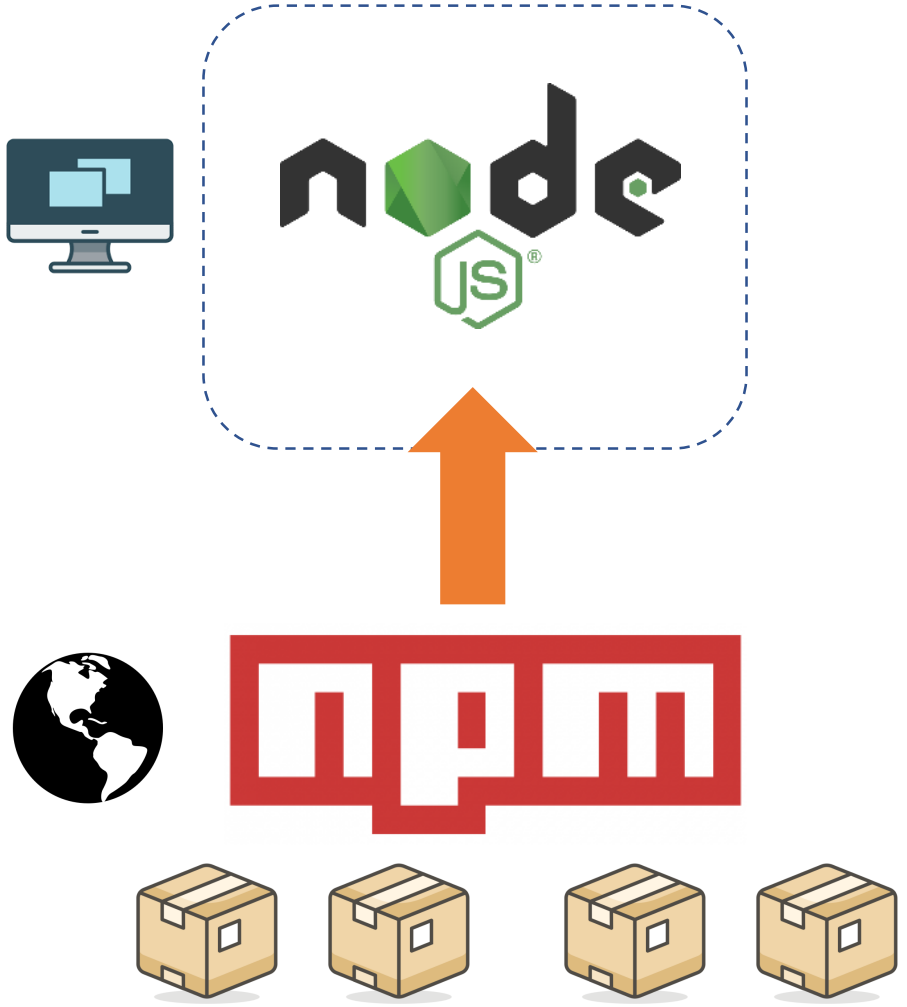
A **package** is just of list of **modules**



IN VSCODE, your install **EXTENSIONS**
Using the **MARKETPLACE**
to get some new **TOOLS**



IN NODE, your install **PACKAGES**
Using **NPM**
to get some new **MODULES**



#STEP 1 : Make your project an NPM project

Create a project and run:

```
npm init
```

What happen on your project?

#STEP 2 : Install an NPM package

Install a package

```
npm install chalk
```

What happen on your project?

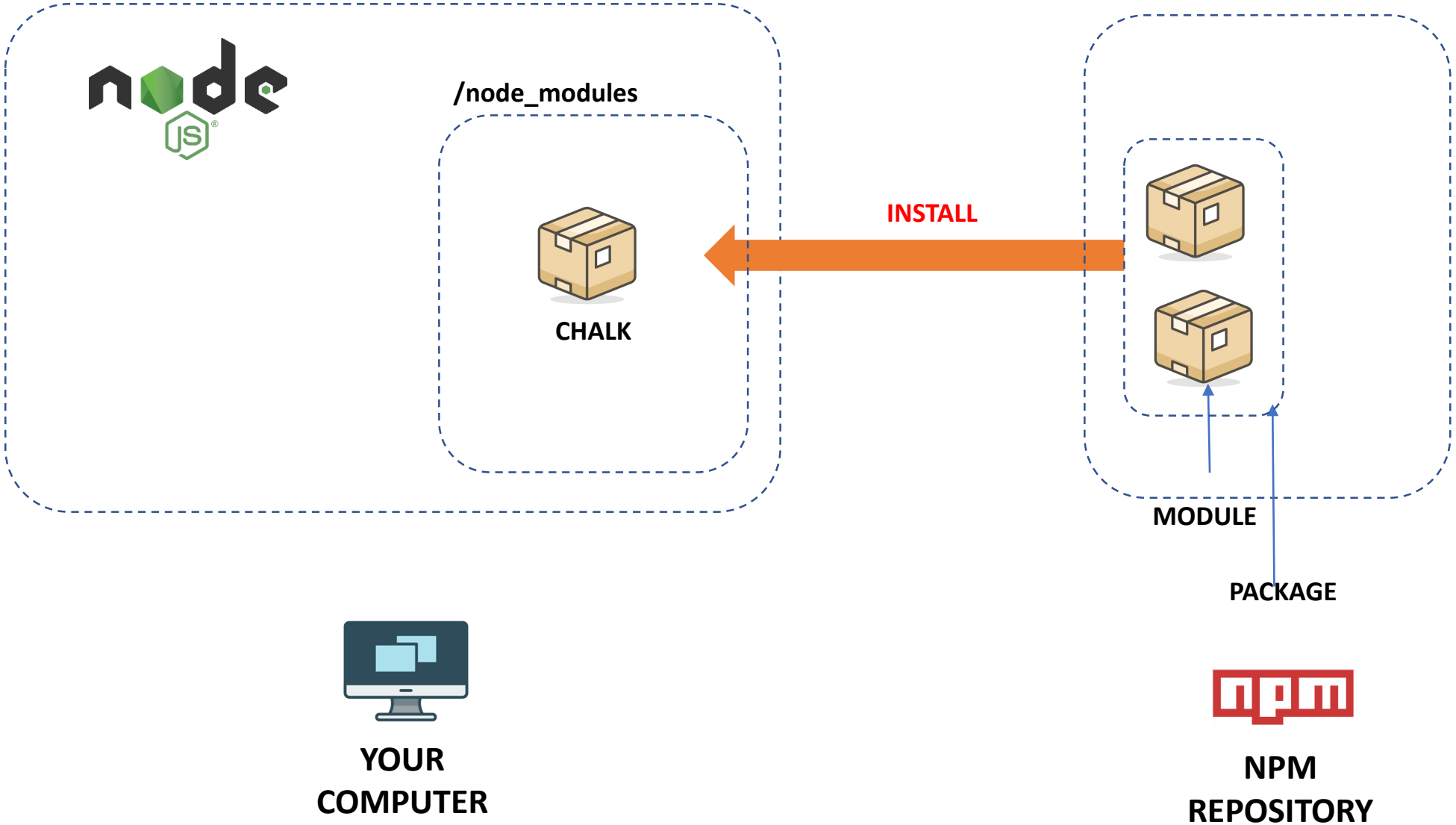
#STEP 3 : Use your package

Create a JS file and require the package

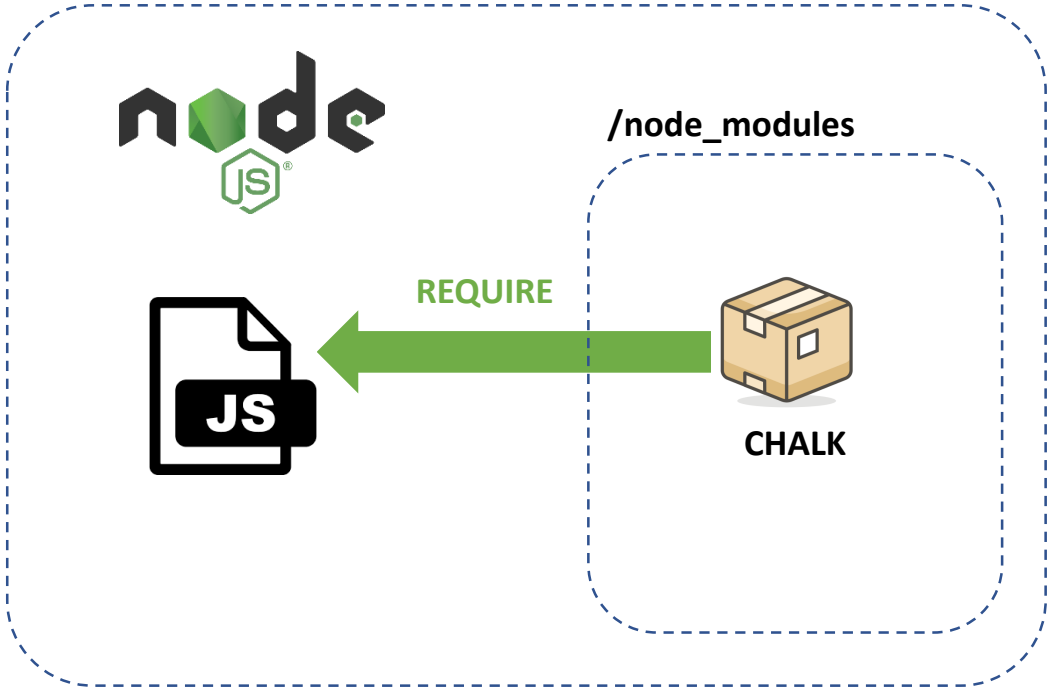
```
const chalk = require('chalk');

console.log(chalk.blue('Hello world!'));
```

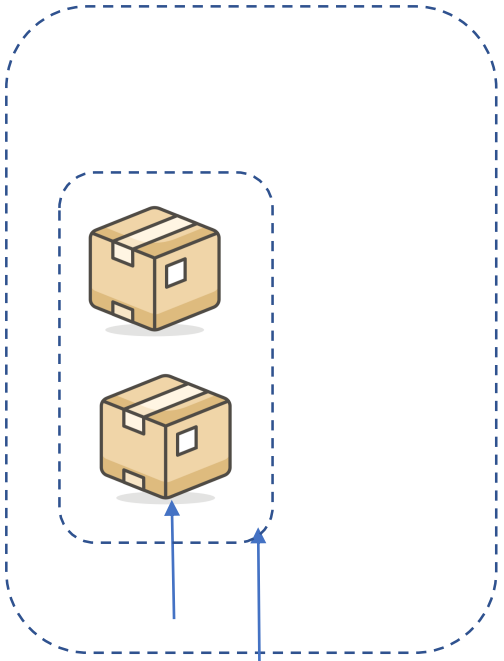
What happen when you run your code ?

NPM downloads the package to the folder /node_modules



Then you can use it : **require**





YOUR COMPUTER

NPM REPOSITORY

What it the package.json ?

Project informations

Packages installed
= dependencies

```
{
  "name": "npm-test",
  "version": "1.0.0",
  "description": "folder to test stuff about npm for the npm course",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "chalk": "^4.1.1"
  }
}
```

Name of package

Version

Can we remove /node_modules ?

1 – Install another package : axios

```
npm install axios
```

What happen on your package.json ?

What happen on your /node_modules?

2 – Remove the folder /node_modules

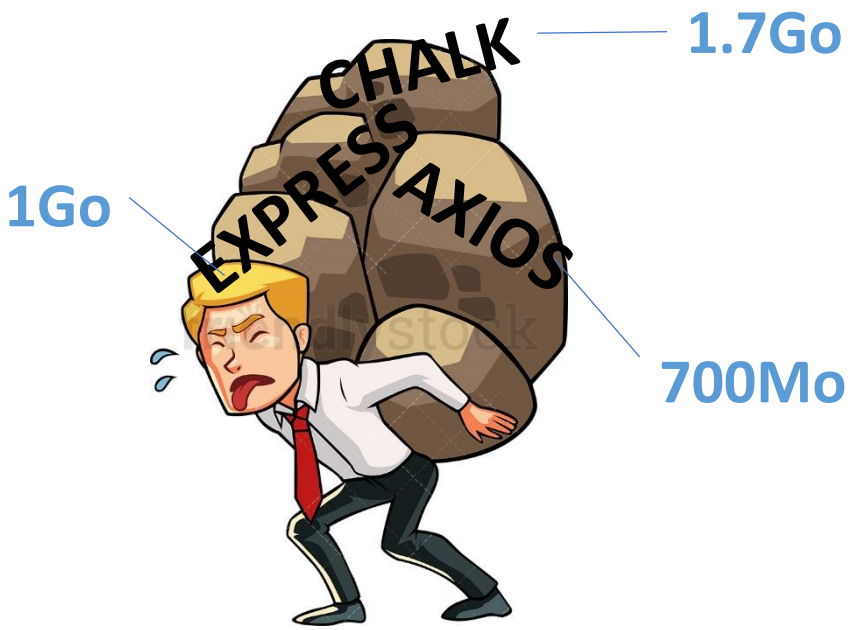
3 – Install all your package dependencies again:

```
npm install
```

What happen on your /node_modules?

You should not send **/node_modules** folder to your friend!

Contain a lot of data



Instead run:

```
npm install
```

Will download dependancies defined in package.json

ACTIVITY

ACTIVITY-1

Follow instructions on main.js

ACTIVITY

ACTIVITY-2

Follow instructions on main.js

OBJECTIVES

- ✓ What **NPM** is used for ?
- ✓ Install an **NPM PACKAGE**
- ✓ Install **DEPENDANCIES** on an existing project