

npm and node (Optional)

This lesson is a crash course in node and npm. It is not exhaustive, but it will cover all of the necessary tools. If you are familiar with both of them, you can skip this section.

The *node package manager* (npm) installs external node packages from the command line. These packages can be a set of utility functions, libraries, or whole frameworks, and they are the dependencies of your application. You can either install these packages to your global node package folder, or to your local project folder.

Global node packages are accessible from everywhere in the terminal, and only need to be installed to the global directory once. Install a global package by typing the following into a terminal:

```
npm install -g <package>
```



The `-g` flag tells npm to install the package globally. Local packages are used in your application by default. For our purposes, we will install React to the local directory terminal by typing:

```
npm install react
```



The installed package will automatically appear in a folder called *node_modules/* and will be listed in the *package.json* file next to your other dependencies.

To initialize the *node_modules/* folder and the *package.json* file for your project, use the following npm command. Then, you can install new local packages via npm:

```
npm init -y
```



The `-y` flag initializes all the defaults in your package.json. After initializing

The `-y` flag initializes all the defaults in your *package.json*. After initializing your npm project, you are ready to install new packages via `npm install <package>`.

The *package.json* file allows you to share your project with other developers without sharing all the node packages. It will contain references to all node packages used in your project, called **dependencies**. Other users can copy a project without the dependencies using the references in *package.json*, where the references make it easy to install all packages using `npm install`. A `npm install` script will take all the dependencies listed in the *package.json* file and install them in the *node_modules/* folder.

Finally, there's one more command to cover about npm:

```
npm install --save-dev <package>
```



The `--save-dev` flag indicates that the node package is only used in the development environment, meaning it won't be used in when the application is deployed to a the server or used in production. It is useful for testing an application using a node package, but want to exclude it from your production environment.

Some of you may want to use other package managers to work with node packages in your applications. **Yarn** is a dependency manager that works similar to **npm**. It has its own list of commands, but you still have access to the same npm registry. Yarn was created to solve issues npm couldn't, but both tools have evolved to the point where either will suffice today.

Exercises:

- Set up a throw away npm project using the terminal:
 - Create a new folder with `mkdir <folder_name>`
 - Navigate into the folder with `cd <folder_name>`
 - Execute `npm init -y` or `npm init`
 - Install a local package like React with `npm install react`
 - Check the *package.json* file and the *node_modules/* folder
 - Attempt to uninstall and reinstall the *react* node package

Further Readings:

- Read about [npm](#)
- Read about [yarn](#) package manager