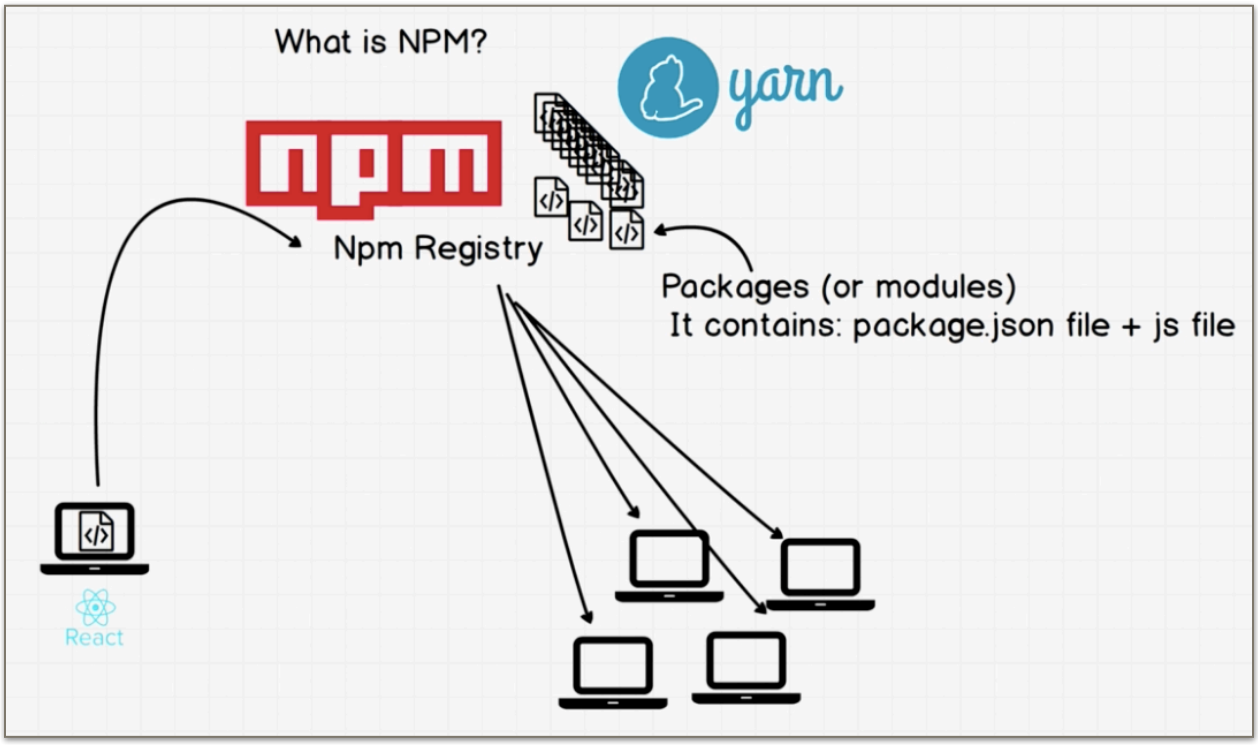
Intro to NPM

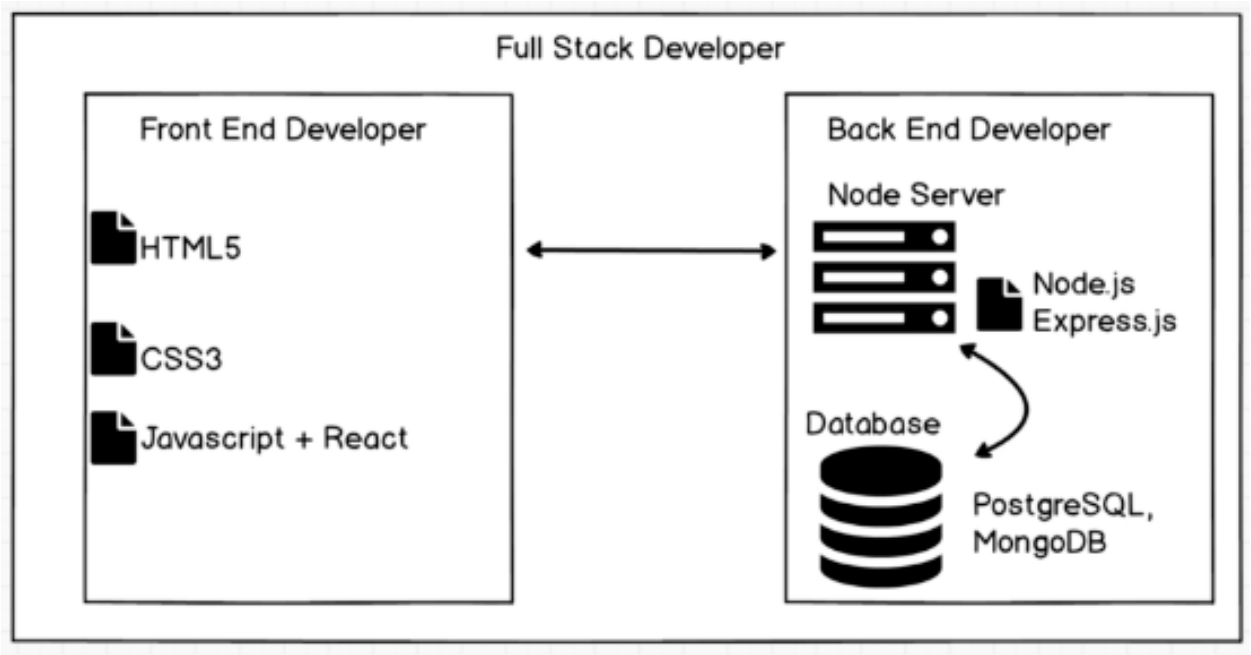
NPM - “Node Package Manager” is a knowledge base of scripts at www.npmjs.com



downsides: not always the best code
increases your own project size (decreases speed)

Yarn is very similar to NPM

package.json - meta file that describes the package



3 Types of Packages:

1. **Front End** - use on the browser
   ex: HTML, CSS, JS, React, Jquery

2. **Command Line** - new commands you can use

3. **Back End** - use on Node.js for back end

3 Must Haves for All Projects:

1. HTML + CSS + Javascript
2. Github Repo + Git
3. Package.json file

We must manage dependencies (packages that other people have written with NPM) using package.json file

Node.js - allows us to run JavaScript outside of the browser

Let's download Node.js (automatically installs NPM too)….

`node -v` << to see node version in terminal
`npm -v`

`npm install npm@latest -g` << to update version

`npm init` << creates a package.json file in your directory (keep pressing enter thru setup)
DO THIS FOR EVERY PROJECT

- If you ever get permission issue, you may need to run the commands with `sudo` in front

NPM allows you to install things two ways:
- globally `-g` can use on whole computer (terminal)
- locally can only use within folder

Favorite Packages

React - package created by Facebook
live-server - creates fake server and then watch for changes
lodash - extension of javaScript

live-server

```
npm install -g live-server      << to install (globally)
sudo npm install -g live-server << if EACESS error
(cd to repo you want to work with)
live-server   << opens webpage that reacts to code changes
ctrl-c << to exit!
```

lodash

```
npm install lodash     << to install (locally)
```

package.json just added a new line >> ”dependencies”: ”lodash”
node_modules folder created, contains lodash files

dependencies - packages that our website needs to work

node_modules folder - where all these packages live

Progressing:
180. Installing And Using Packages 24:08