What is React.js?

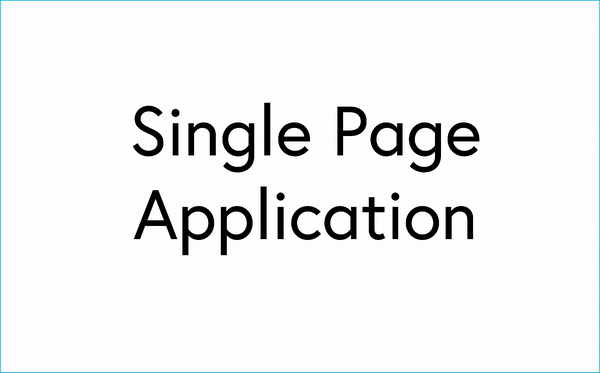
React.js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications.

**Single-Page application:** The single page application is a web application that interacts with the user by dynamically rewriting the current page, rather than loading entire new pages from the server.

On a traditional website, your request for a new tree would cause the server to repaint the entire picture and send it back.



a traditional website

the page can be dynamically rewritten, instead of going through an entire reload. This makes things a whole lot faster.

Why React? **(even we have other options like Angular, Vue.js)**

* "The main advantage of using a Library over a Framework is that Libraries are lightweight, and there is a freedom to choose different tools. The Framework consists of an entire ecosystem to build an application, and you don't have *an easy way*to use any other tools outside the Framework."

## React Has a Great Developer Experience

## Traditional frameworks like Angular and Vue power up the HTML. They use JavaScript inside HTML. They have created HTML attributes that give extra capabilities to it.

## Here's an example of looping in Angular. Look at the \*ngFor attribute:

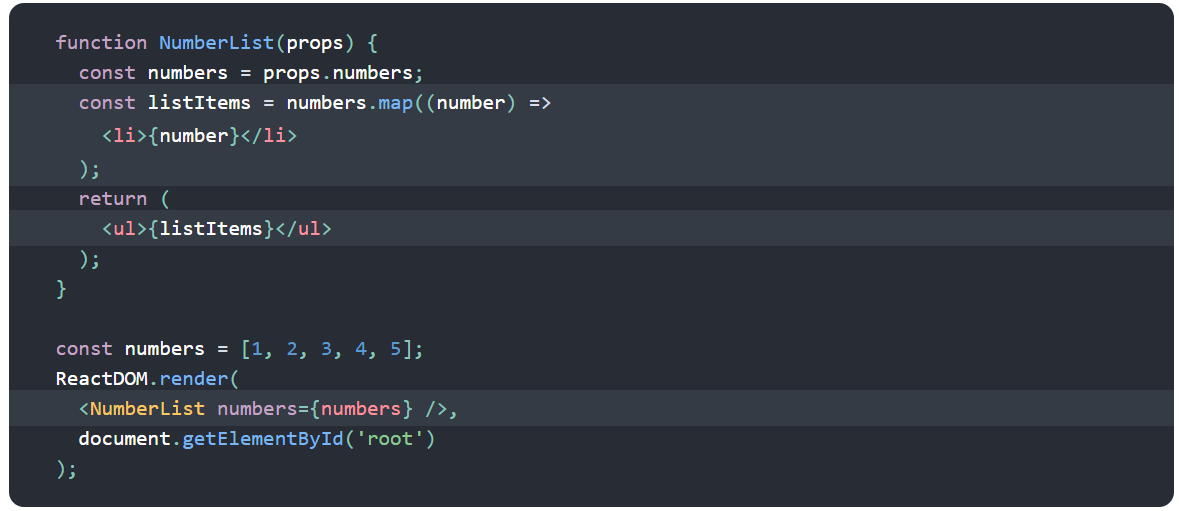
## Angular-Loops

## And here's an example of looping in Vue.js. Look at the v-for attribute:

## vue-looping

React uses the opposite approach. It uses HTML (JSX) inside JavaScript. I like this approach because here, you can deal with pure JavaScript and HTML.

Here's an example of looping in React.js



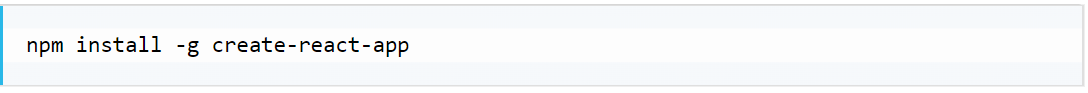
# **Environment Setup**

# **Install Nodejs:**

**Node.js** provides a runtime environment to execute JavaScript code from outside a browser.

**NPM**, the Node package manager is used for managing and sharing the packages for either React or Angular. NPM will be installed along with Nodejs.

# **Install Create-React-App Tool:**

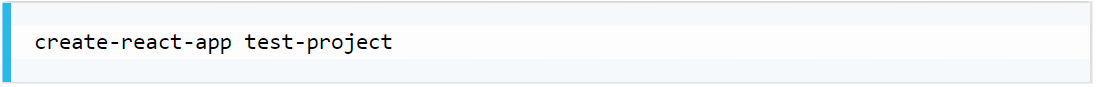


**create-react-app** is a easier way which does all the configuration and necessary package installations automatically and starts a new React app locally, ready for development.

**Create First App:**

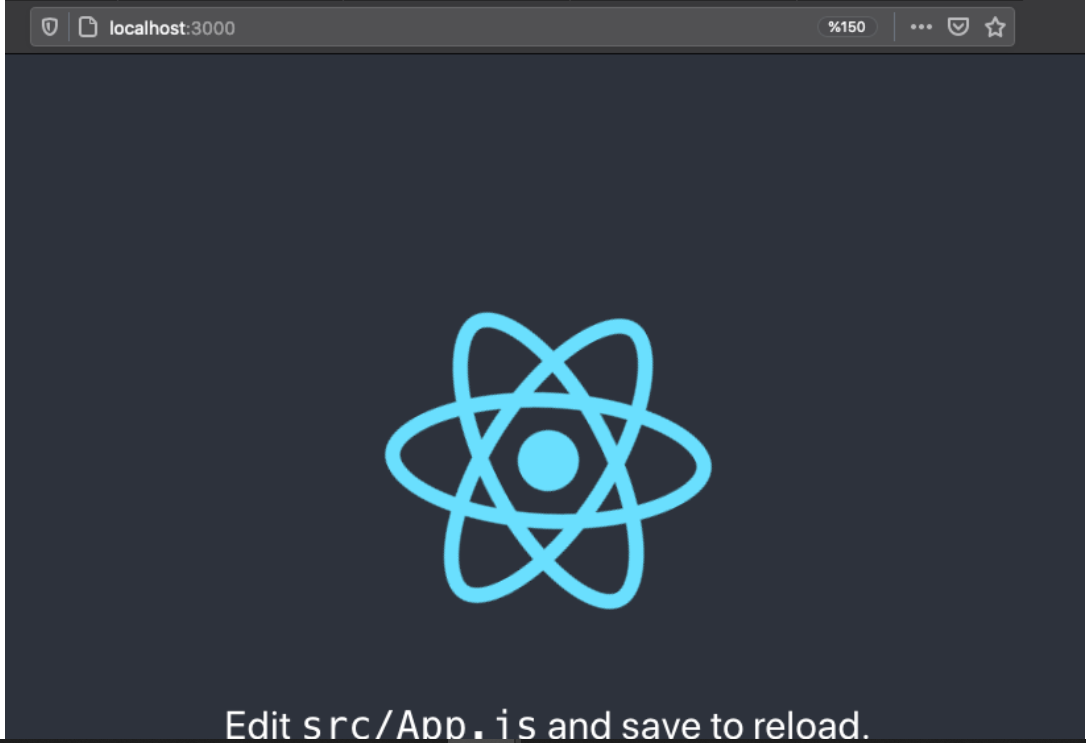
After create-react-app is installed, we can create our first react application. Let's say I want to create the project or application in D:\React\_Programs.

**D:\React\_Programs > create-react-app test-project**

****

**Running First App:**

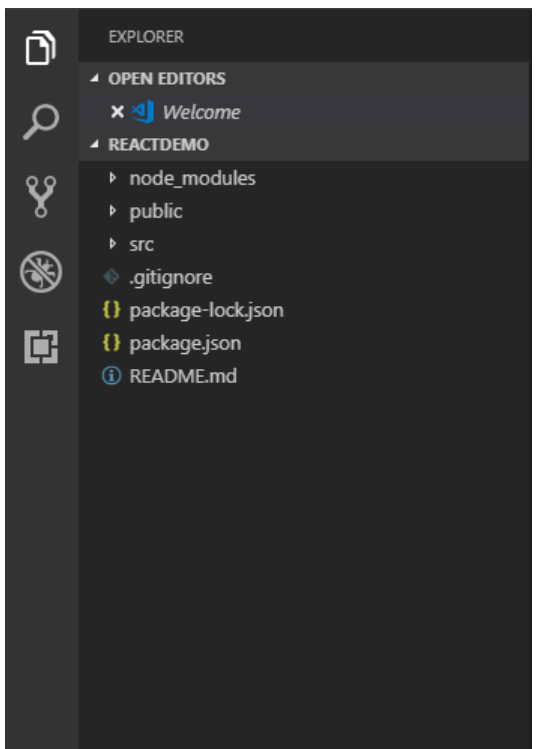
First command prompt point to test-project folder **D:\React\_Programs\test-project>** and we have to use the command **npm start** to run the app.



If you see something like this in your browser then your app is up and running.

**Basic Project Structure Of React Project**

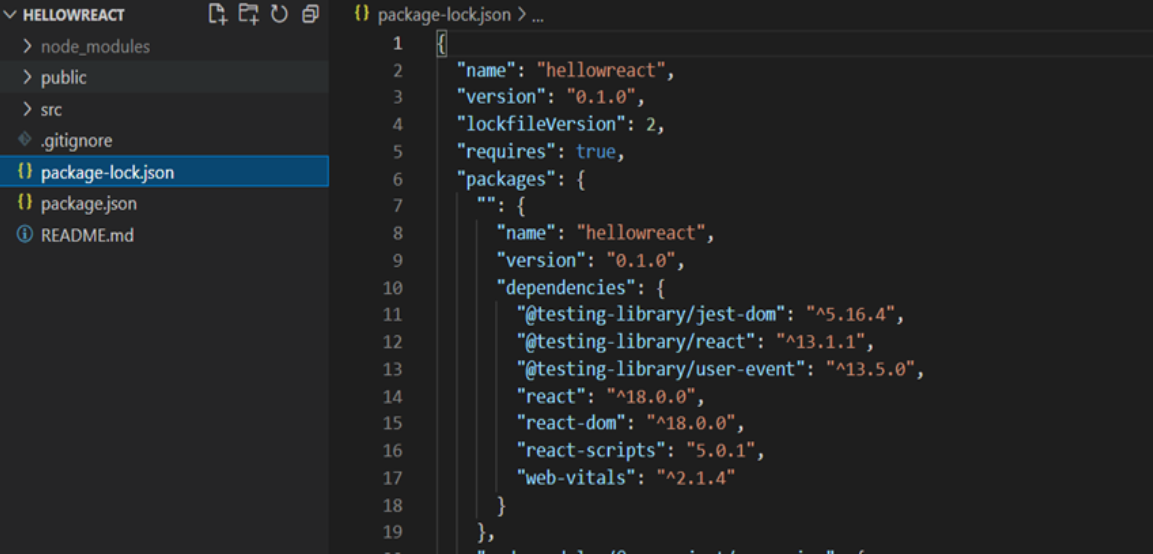
The React application automatically creates required folders, as shown below.

****

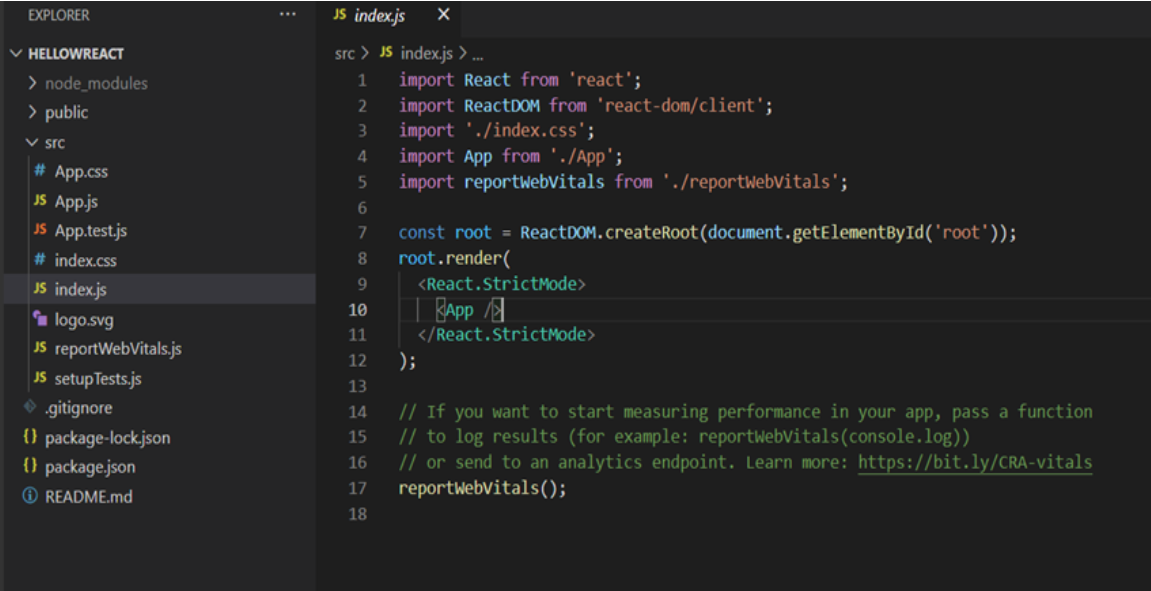
**node\_modules**: This folder will contain all react js dependencies.

**.gitignore:** This file is used by source control tool to identify which files and folders should be included or ignored during code commit.

**package.json**: This file contains dependencies and scripts required for the project.



**Src folder**: src is one of the main folder in react project.



**Index.js**: index.js is the file that will be called once we will run the project. Something like the "entry point" of the application. It just has code of what to render and where to render.

**App.js**: App.js is a component that will get loaded under index.js file. If we do any change in app.js file HTML component and save it it will reflect in localhost://3000

