**React Js for beginners (just to get you started)**

Is react a framework or a library? Well, don’t bother. Reactjs.org calls it a library (collection of precompiled objects based on a programming language in general) so let it be that while letting the world ponder over the differences between a library and a framework.

What is the difference between React Js and React Native?

React Js is for web/ browser apps while React Native is for mobile apps.

*Pre-requisite for this tutorial is a basic understanding of HTML, CSS and Javascript.*

The simplest way to get React installed is through Node js (a runtime/ environment\* that allows you to run javascript at the back-end/ server). Install Node js from here: <https://nodejs.org/en/download/>

\* Again you don’t need to stress yourself with what Node js actually is, consider it something similar to a back-end programming language (Javascript here) like PHP.

Select the LTS version and then your operating system.

Once Node js is installed now run:

npx create-react-app your-app-name

If you encounter an error saying “………………..as it does not contain a package,json file”, run:

npm i create-react-app *(both “i” and “install” can be used alternatively in node commands)*

*npm is a package manager of Node js which allows you to install various node packages create-react-app being one of such.*

*Now run:*

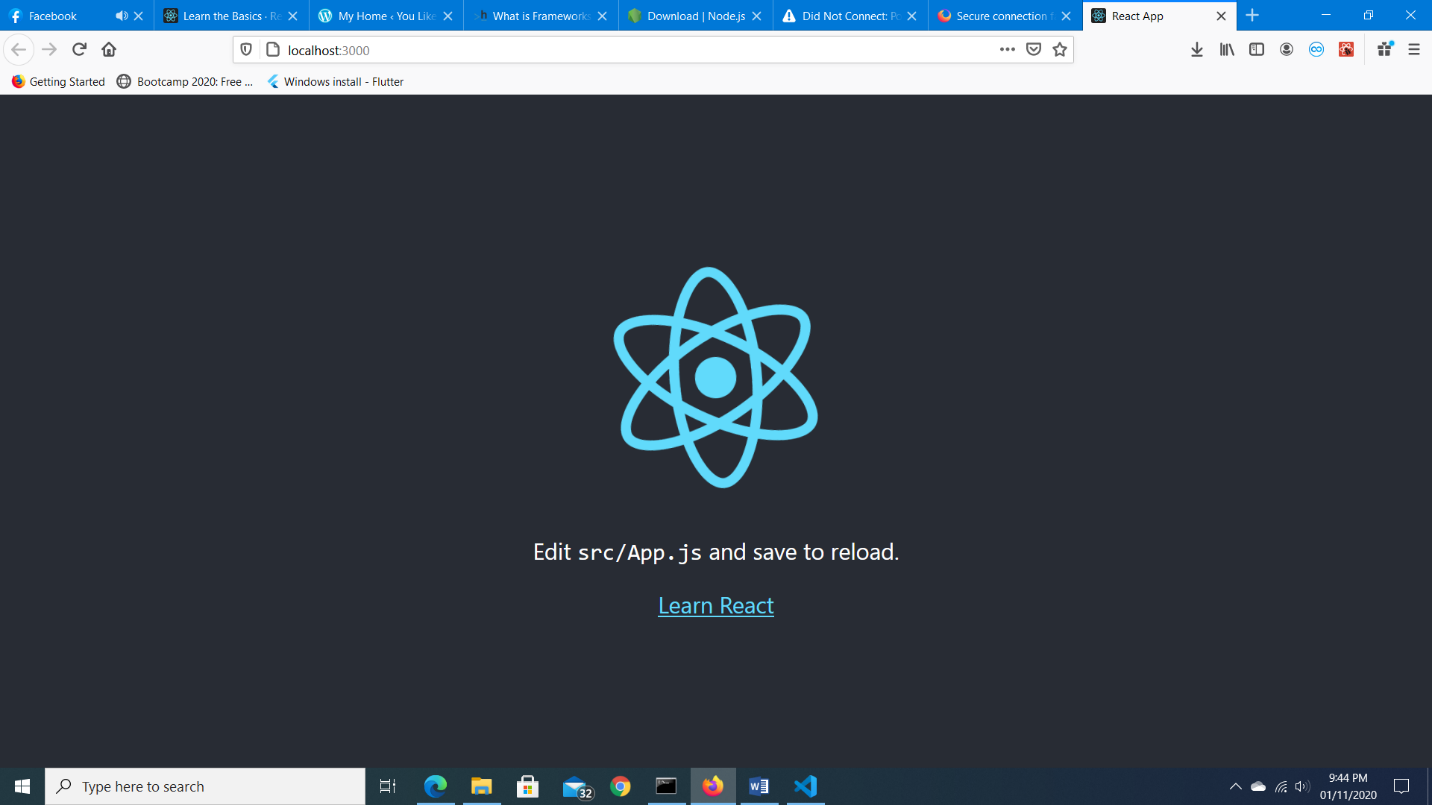
npx create-react-app your-app-name

and your first react app has already been created in the folder named “your-app-name” in this case. You can specify the app name of your choice.

Also run:

npm start

in terminal/ command prompt while in your app’s folder. This would run the development build of your basic react app in the browser at localhost:3000 which would look something like:



Now open that folder in a code editor of your choice.

A few files and folders of our concern for now would be:

The “public” folder containing the index.html

The “src” folder containing the App.js file along with some others.

**The hierarchy**

Let’s now connect some dots.

Open public\index.html in the code editor. Forget the meta and link tags for now and see the body where we have an empty <div> with the id “root”.

Leave that empty and now open: src\index.js which would be having this code:

ReactDOM.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>,

  document.getElementById('root')

);

*Yeah I know there would be something above this too. That code just imports the required modules like React, ReactDOM etc. to compile this code and connects the styling file index.css available in the src folder.*

Anyways so in the code above, a method (coming from ReactDOM which itself is a React module) we see is:

ReactDO.render(

/\*blah blah blah\*/,

document.getElementById(‘root’)

);

This simply means that render my /\*blah blah blah\*/ stuff on the <div> with the id “root” in that public\index.html file. Try changing this “root” to something else and also change the div id in index.html to the same and the app would still render. (You don’t need to reload the page in your browser, it would reload automatically with the saved changes)

Now what is that:

<React.StrictMode>

    <App />

  </React.StrictMode>,

For the StrictMode part all you need to know now is that it is a React tool which highlights potential problems during development. What we are concerned with here is the <App />.

What is this <App/>? Looks similar to an HTML tag? Well that is because React uses an object oriented language called JSX which is an extension of Javascript. JSX allows us to use HTML tags and HTML like tags in our Javascript files. Again, we do not have to be concerned about the nitty gritty her.

<App/> is a component being exported from App.js (in the same src folder) and imported here in our index.js file through:

import App from './App';

**What are components in React?**

Components are what build a React UI. A React UI is a collection of components out of which some may be parents and some may be children of those parent components which all relate to a single grand parent, the <App /> here.

Now we can sum up the code we went through as:

“Render the App component from App.js in StrictMode on the div with the id “root” in our public\index.html file.”

Why all this mess though? Remember, React is Javascript/ JSX not HTML and browsers recognize only HTML for rendering/ displaying stuff. Js needs to connect with HTML somehow to display what it wants. So, it’s actually that index.html file being rendered with stuff coming from js files working in the background through that linkage we just saw.

**Let’s now build our own stuff**

Open App.js and you’ll see some code similar to the one below:

import logo from './logo.svg';

import './App.css';

function App() {

  return (

    <div className="App">

      <header className="App-header">

        <img src={logo} className="App-logo" alt="logo" />

        <p>

          Edit <code>src/App.js</code> and save to reload.

        </p>

        <a

          className="App-link"

          href="https://reactjs.org"

          target="\_blank"

          rel="noopener noreferrer"

        >

          Learn React

        </a>

      </header>

    </div>

  );

}

export default App;