ReactJS

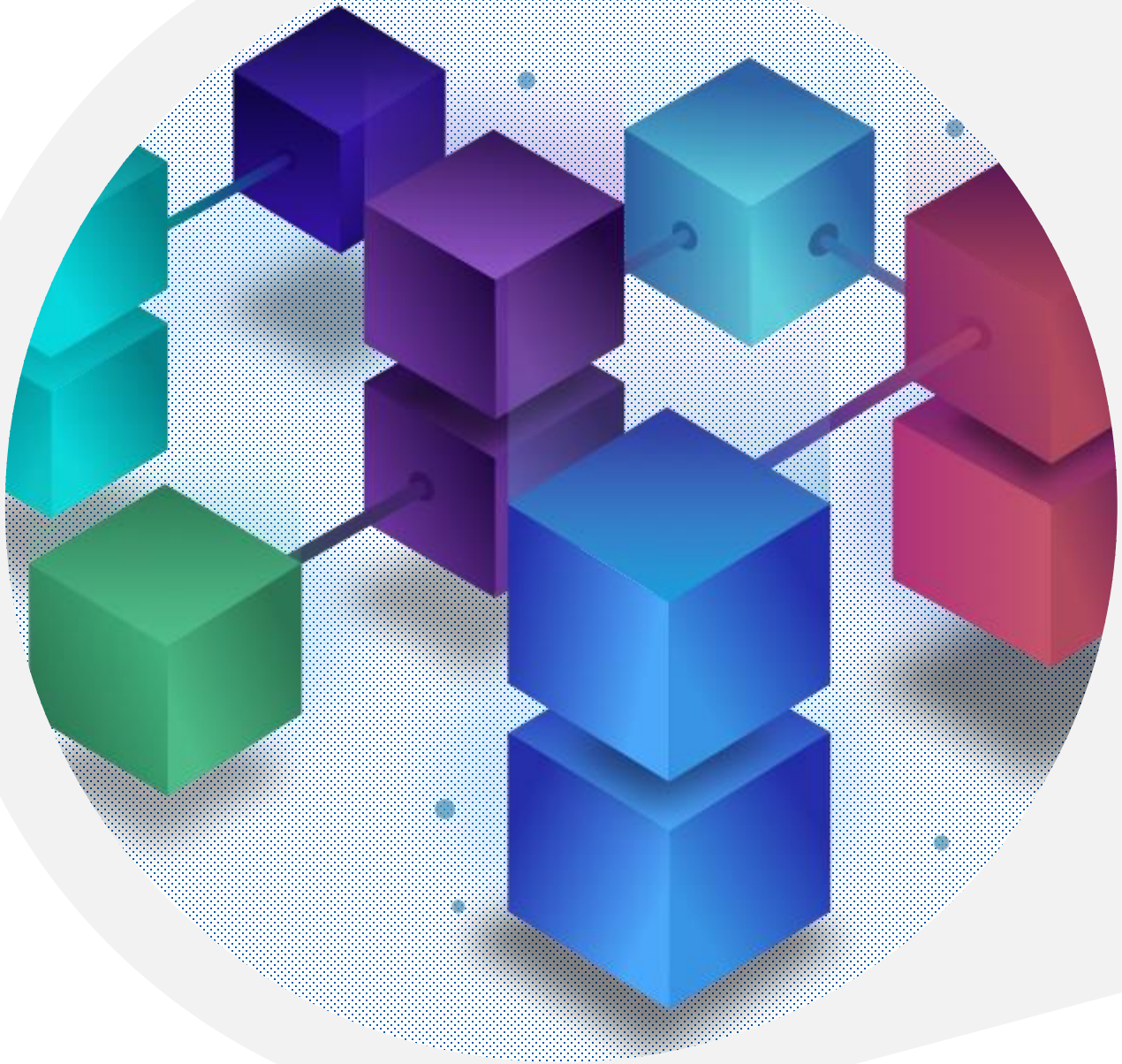
June 2021

Course Objective

- React Introduction
- Environment Setup
- ES6
- React Render Html
- JSX
- React Components
- Props and State
- Component API
- Component Life cycle
- React Forms, Events Refs, Keys
- Rest API
- React Router

Session Plan

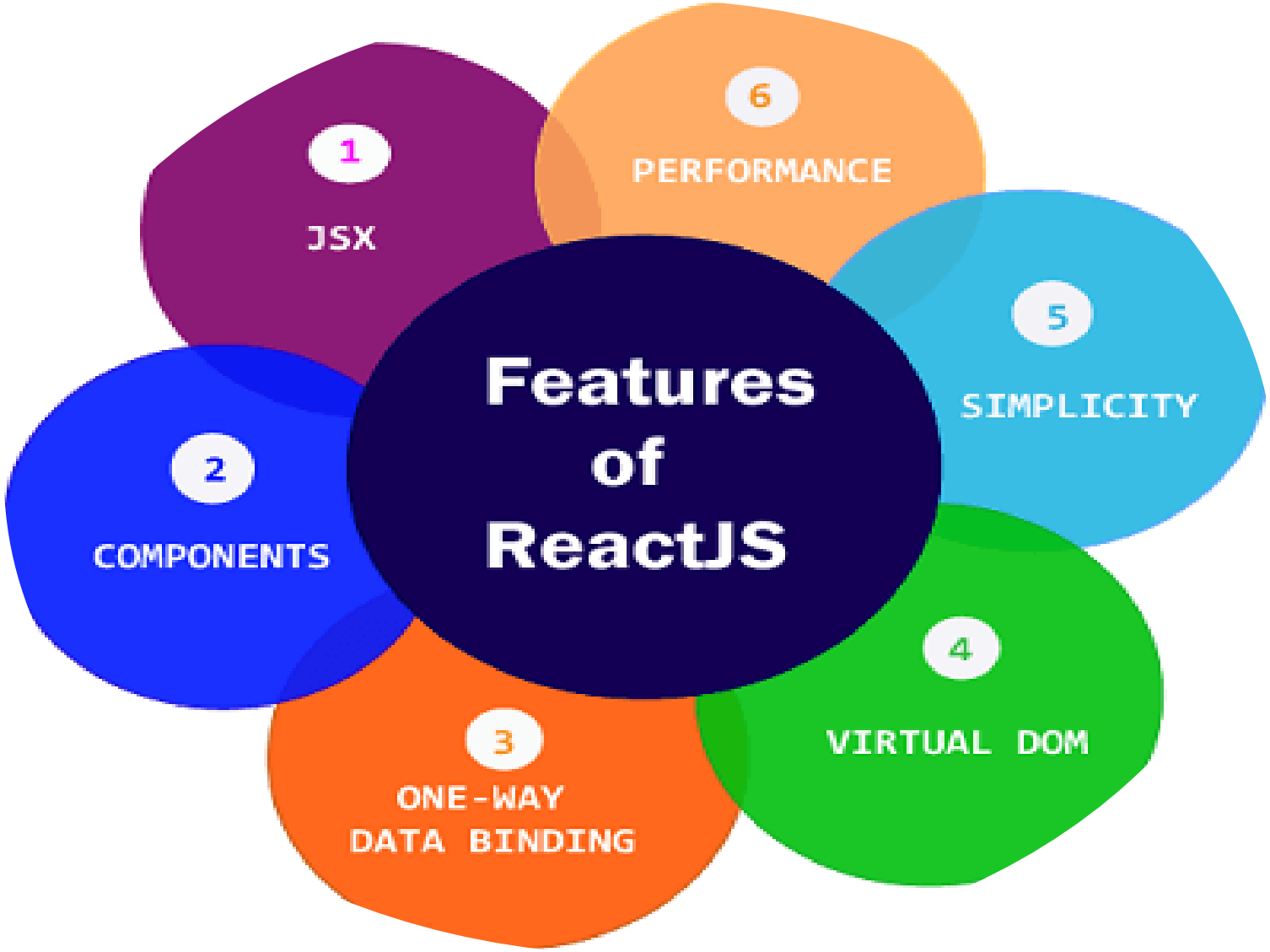
- React Introduction
- Environment Setup
- ES6
- React Render Html
- JSX
- React Components



React Introduction

- **React is a JavaScript library created by Facebook to build UI as SPA.**
- **React is a tool for building reusable UI components.**
- **React Components can be reused in Angular and Vue.**
- **React is Declarative.**
- **Seamlessly integrate React into any of your application.**

Features of ReactJS



Features of ReactJS

JSX

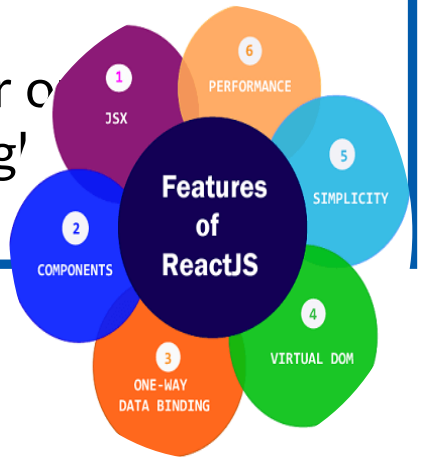
JSX stands for JavaScript XML.

Components

ReactJS components has its own logic and controls. These components can be reusable.

One-way Data Binding

ReactJS is designed in such a manner that follows unidirectional data flow or o
binding. The benefits of one-way data binding give you better control throug
application.



Features of ReactJS

Virtual DOM

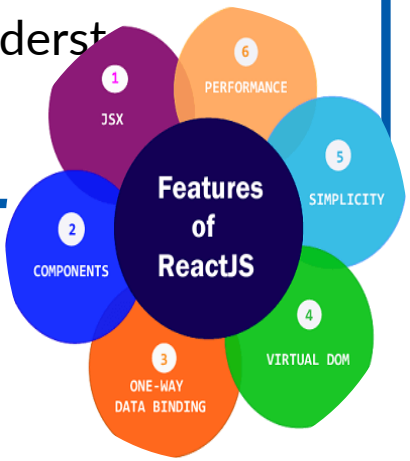
A virtual DOM object is a representation of the original DOM object and works in one-way data binding.
The entire UI is re-rendered in virtual DOM for the modifications in the web application and it checks the difference between the previous DOM representation and new DOM and update the changes.

Simplicity

ReactJS uses JSX file which makes the application simple and to code as well as underst

Performance

ReactJS is a great performer because of virtual DOM.



Mode of installation

- Using the npm command
- Using the create-react-app

Installing ReactJS Using the create-react-app command

Create folder react in Desktop

C:\Users\username>cd C:\Users\28695\Desktop\react

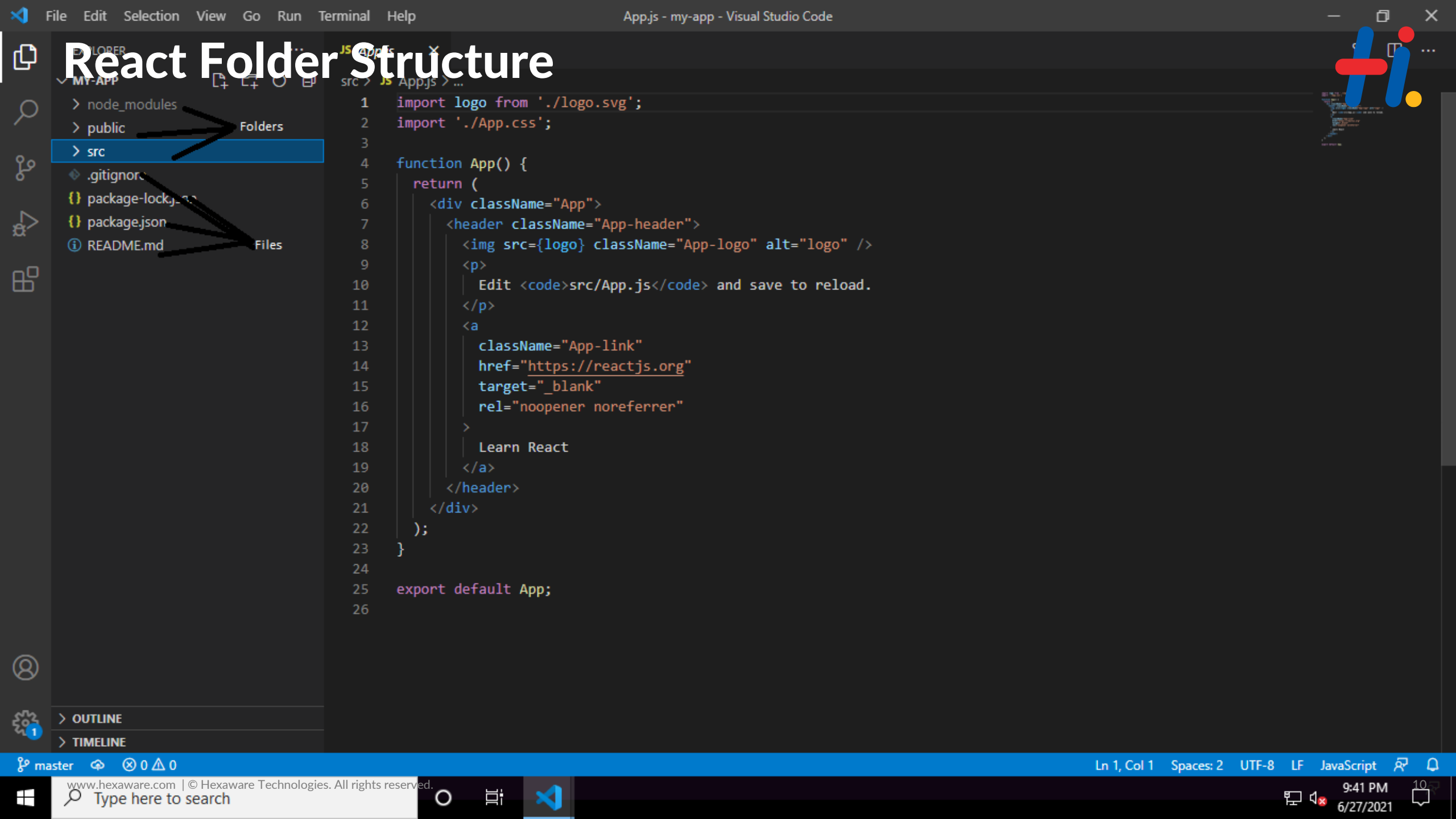
Create the react project using create-react-app

C:\Users\Tutorialspoint\Desktop> create-react-app my-app

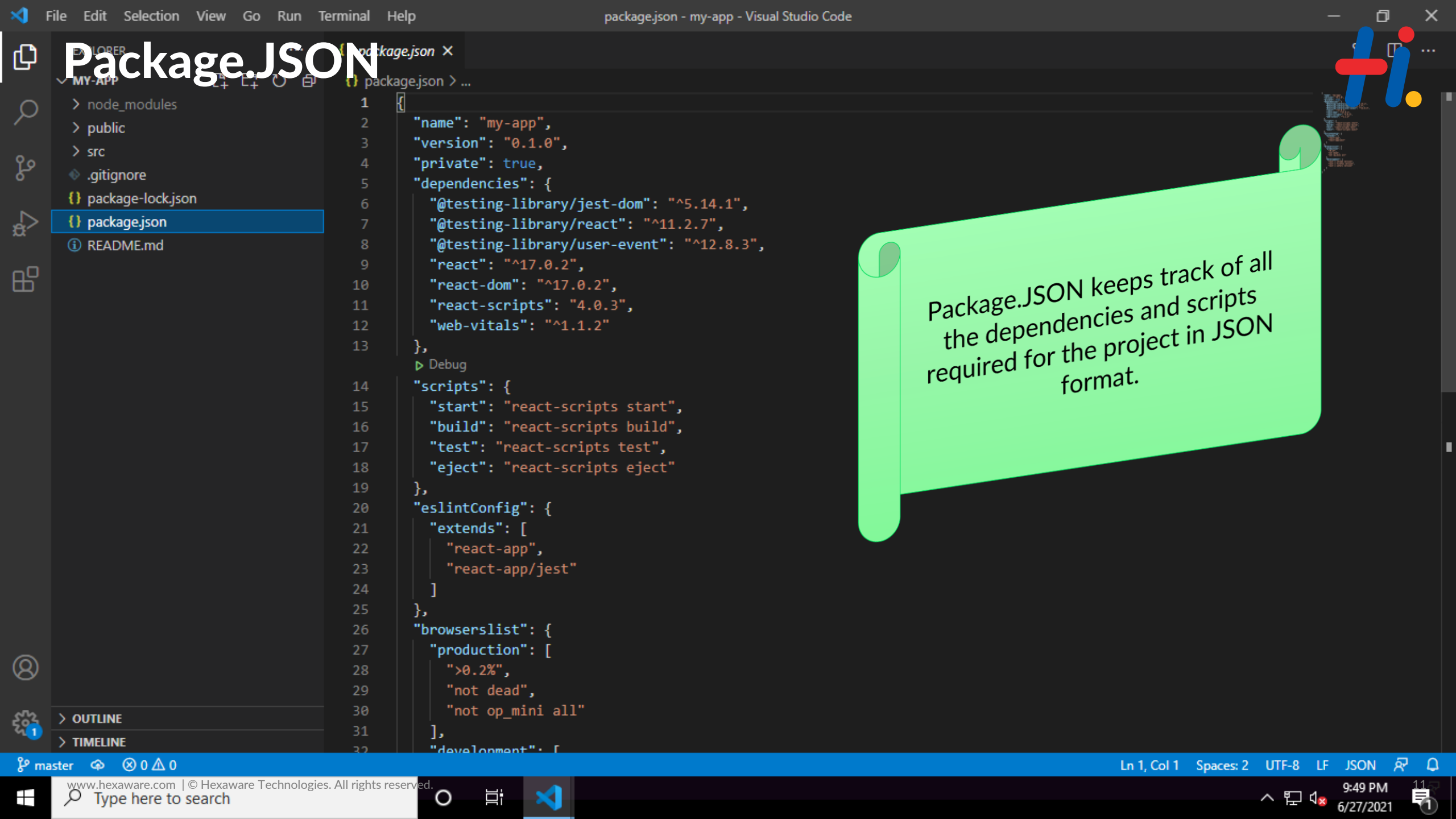
Check the execution of the demo:

npm start

React Folder Structure

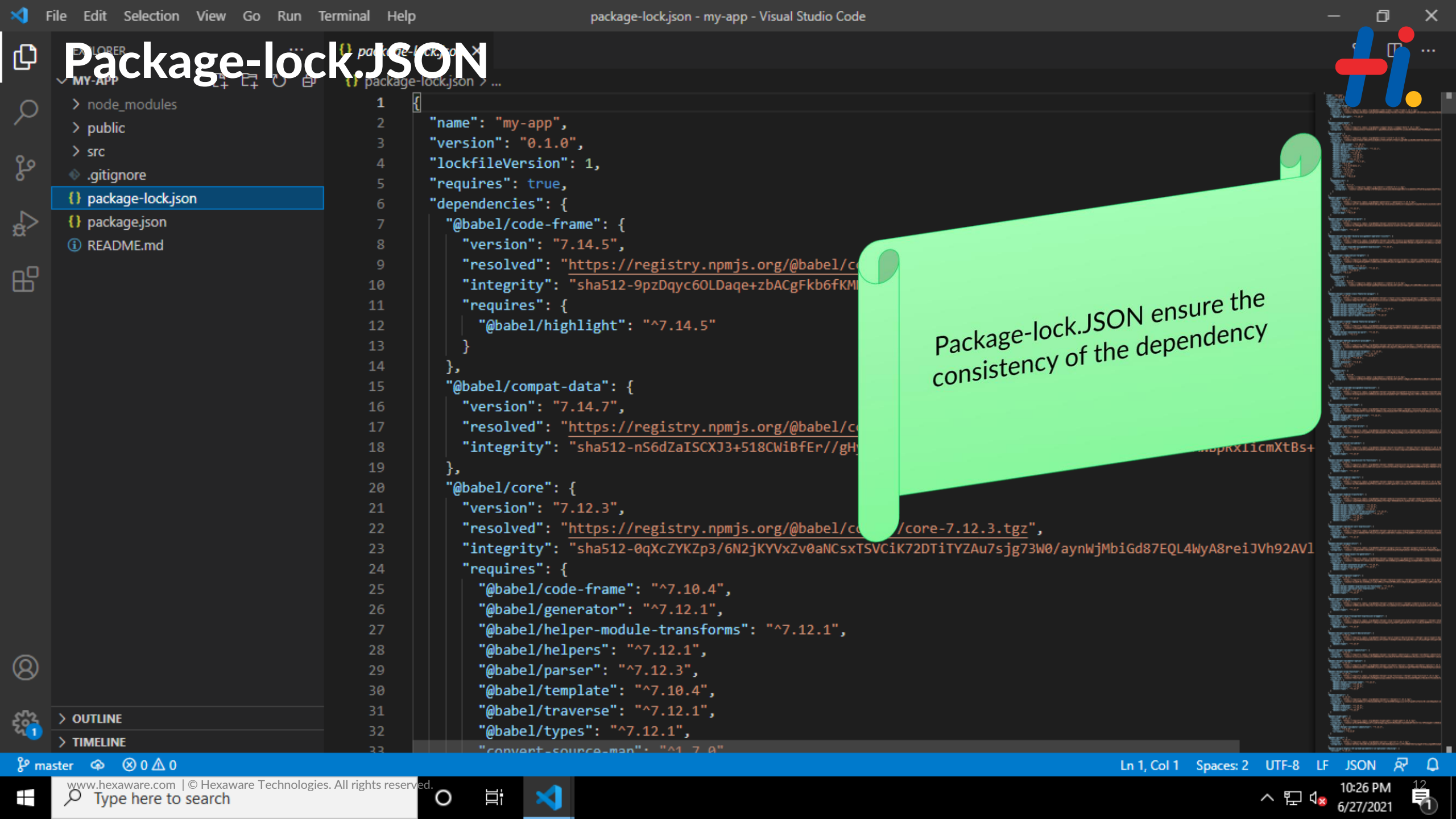


Package.JSON

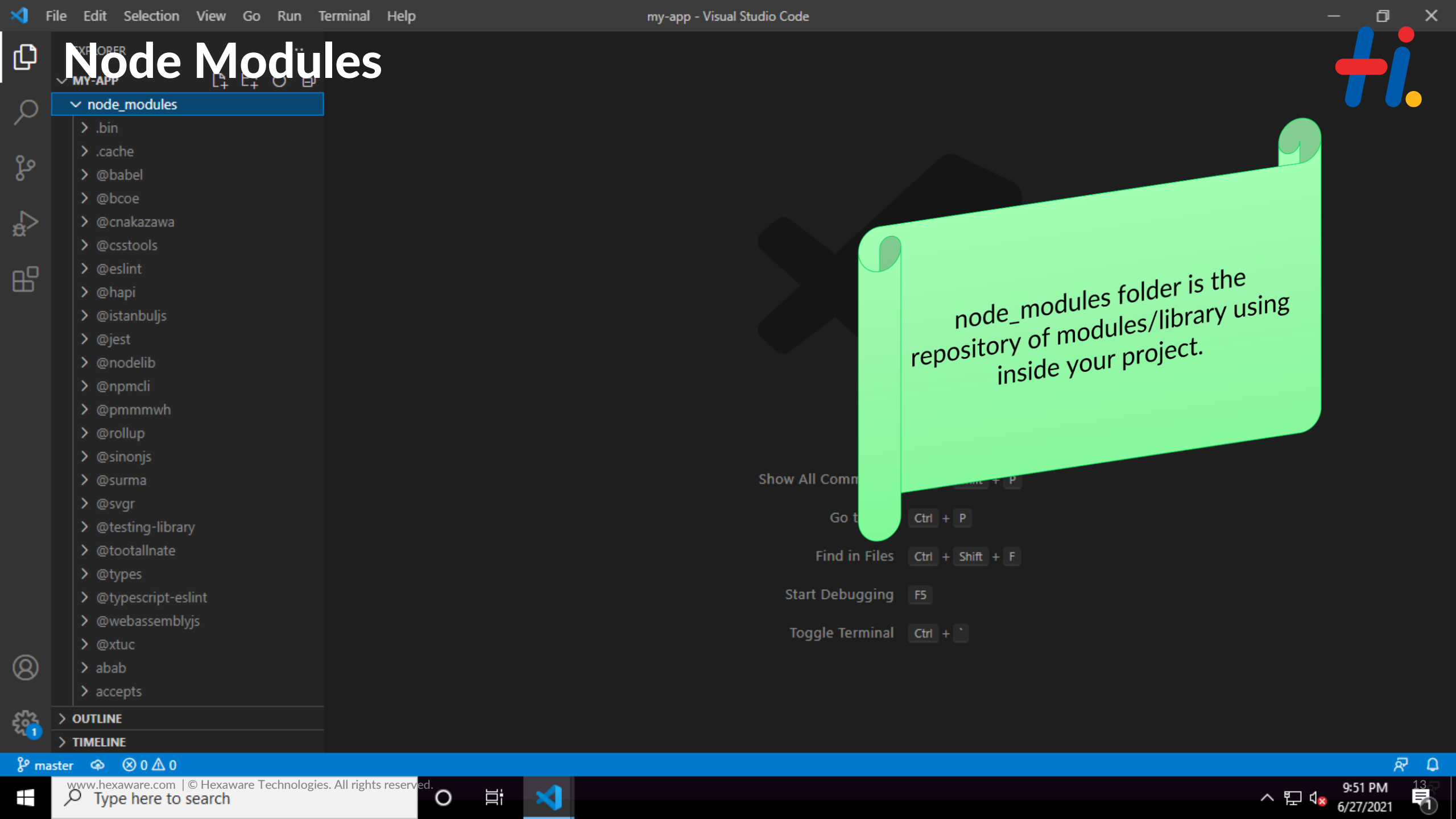


Package.JSON keeps track of all the dependencies and scripts required for the project in JSON format.

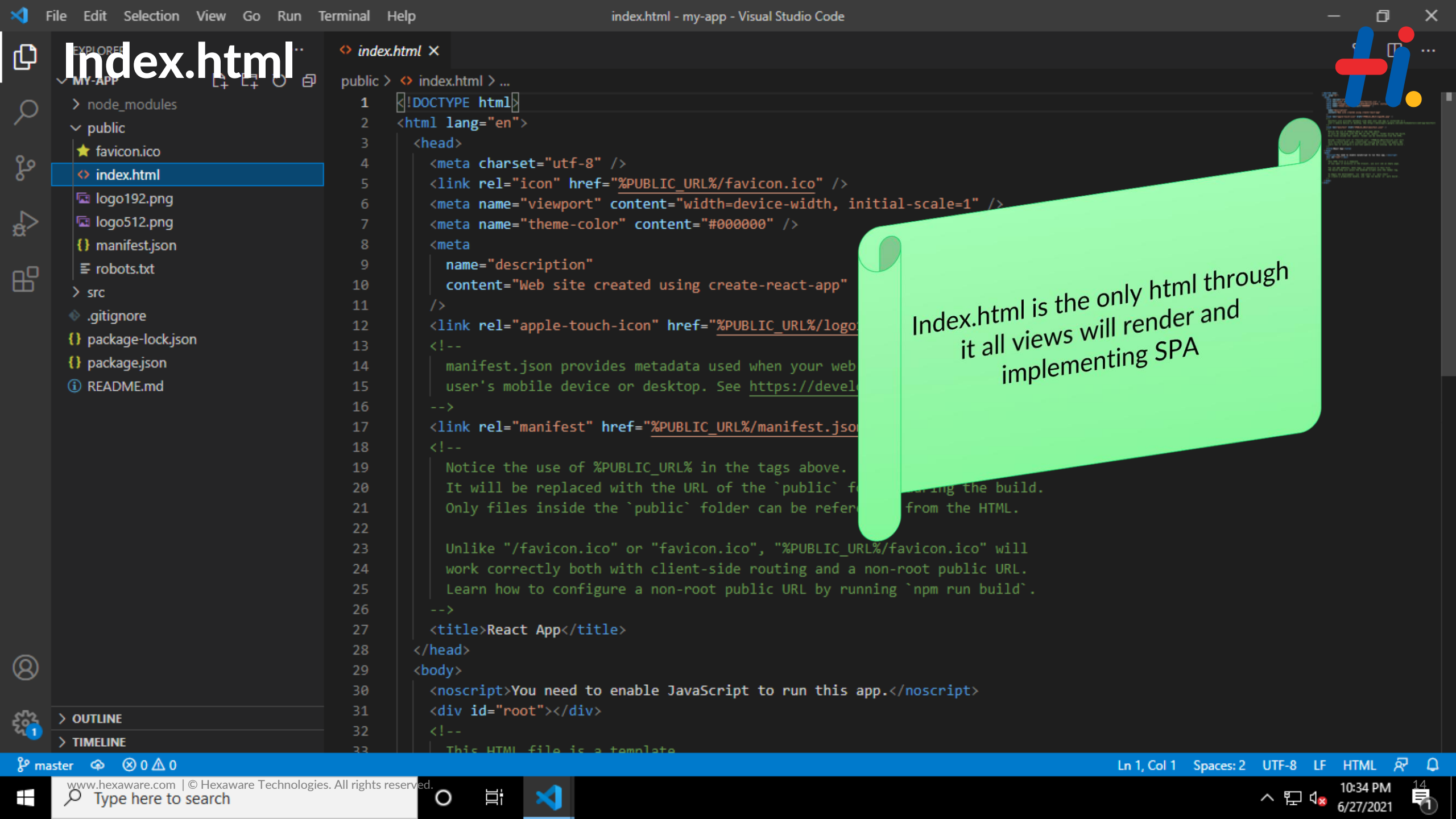
Package-lock.JSON



Node Modules

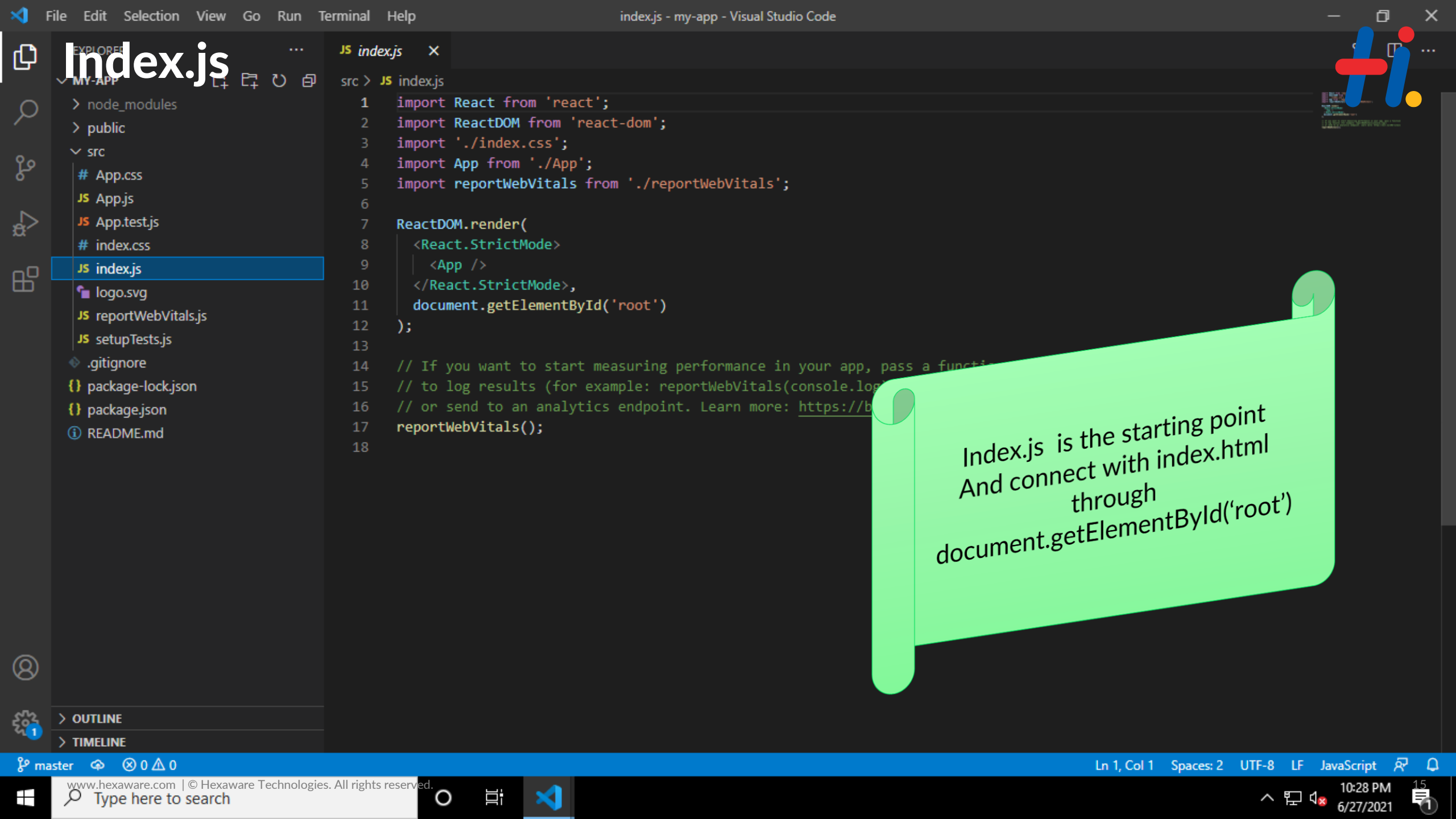


Index.html

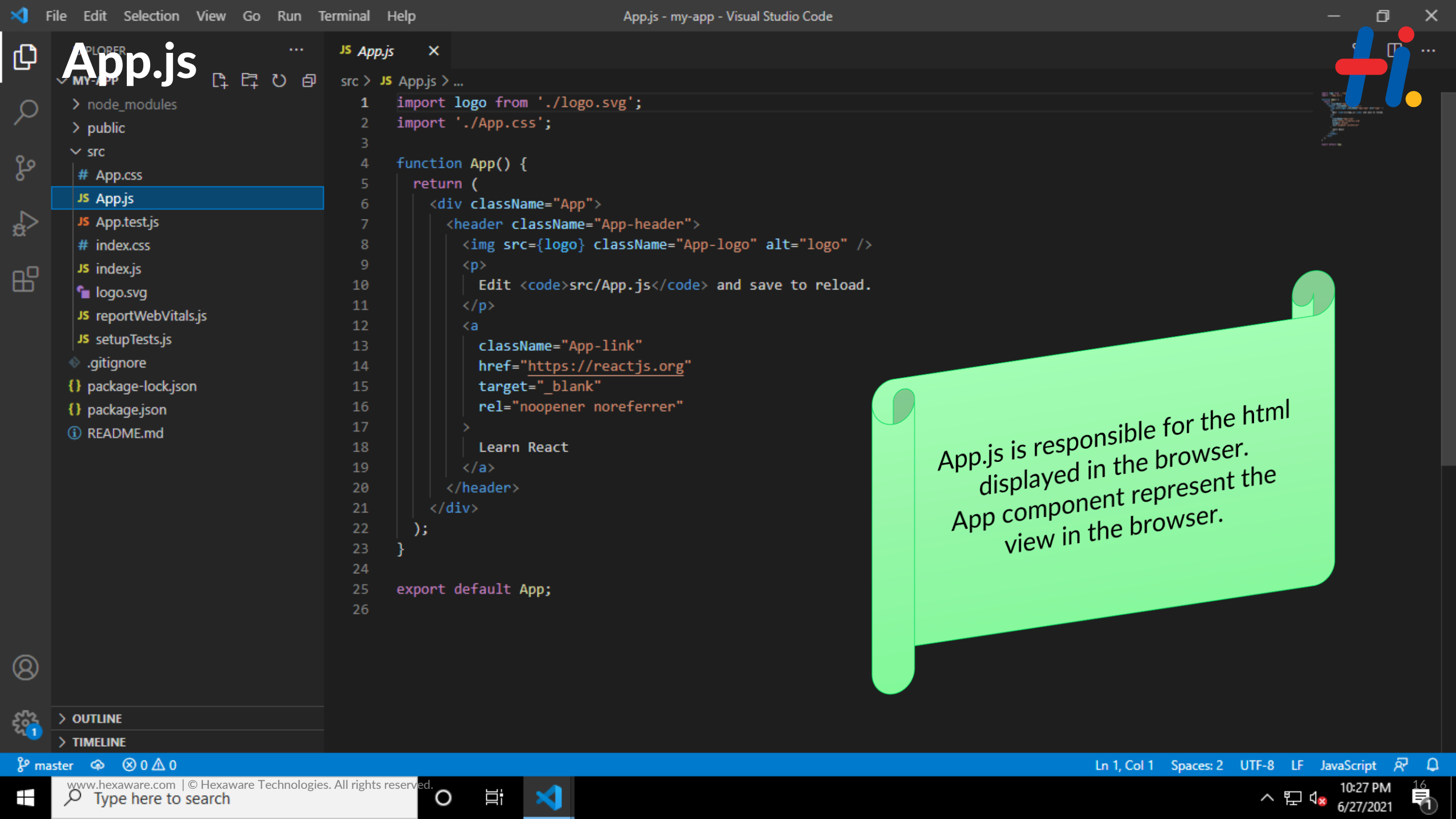


Index.html is the only html through it all views will render and implementing SPA

Index.js



App.js



React Render HTML

React renders HTML to the web page by using a function called ReactDOM.render().

The ReactDOM.render() function takes two arguments, HTML code and an HTML element.

The render function is to display the specified HTML code inside the specified HTML element.

```
ReactDOM.render(<p>Hello</p>, document.getElementById('root'));
```


ES6

ES6

- ES6 stands for ECMAScript 6.

- ECMAScript is a standardize JavaScript.
- React uses ES6 as language.

- Main features of ES6 are:
    - Classes
    - Arrow Functions
    - Variables (let, const, var)

ES6 Class

- A class is a type of function, but instead
- of using the keyword function to initiate it,
- we use the keyword class, and the
- properties are assigned inside a
- constructor() method.

```
<!DOCTYPE html>
<html>

<body>

<script>
class Car {
  constructor(name) {
    this.brand = name;
  }
}

mycar = new Car("Ford");

document.write(mycar.brand);
</script>

</body>
</html>
```

Method in Classes

```
class Car {
  constructor(name) {
    this.brand = name;
  }

  present() {
    return 'I have a ' + this.brand;
  }
}

mycar = new Car("Ford");
mycar.present();
```


Class Inheritance

```
class Car {
  constructor(name) {
    this.brand = name;
  }

  present() {
    return 'I have a ' + this.brand;
  }
}
```

```
class Model extends Car {
  constructor(name, mod) {
    super(name);
    this.model = mod;
  }
  show() {
      return this.present() + ', it is a ' + this.model
  }
}
mycar = new Model("Ford", "Mustang");
mycar.show();
```


Arrow Functions

Arrow functions allow us to write shorter function syntax:

```
<!DOCTYPE html>
<html>
<body>
<h1>Arrow Function</h1>
<p>A demonstration of a simple arrow function.</p>
<p id="demo"></p>
<script>
hello = () => {
  return "Hello World!";
}
document.getElementById("demo").innerHTML = hello();
</script>
</body>
</html>
```

```
hello = function() {
return "Hello World!";
}
```

Arrow Functions with parameter and return type

```
hello = val => "Hello " + val;
```

Without parameter

```
hello = (val) => "Hello " + val;
```

With parameter

```
hello = function() {
  return "Hello World!";
}
```

With Return Type

Variables

var

var will be considered as global scope if it declared outside of a function

Var will be considered to function scope if it declared inside the function

If you use var inside of a block like inside a for loop, the variable is still available outside of that block.

var has a function scope, not a block scope.

```
var x = 5.6;
```

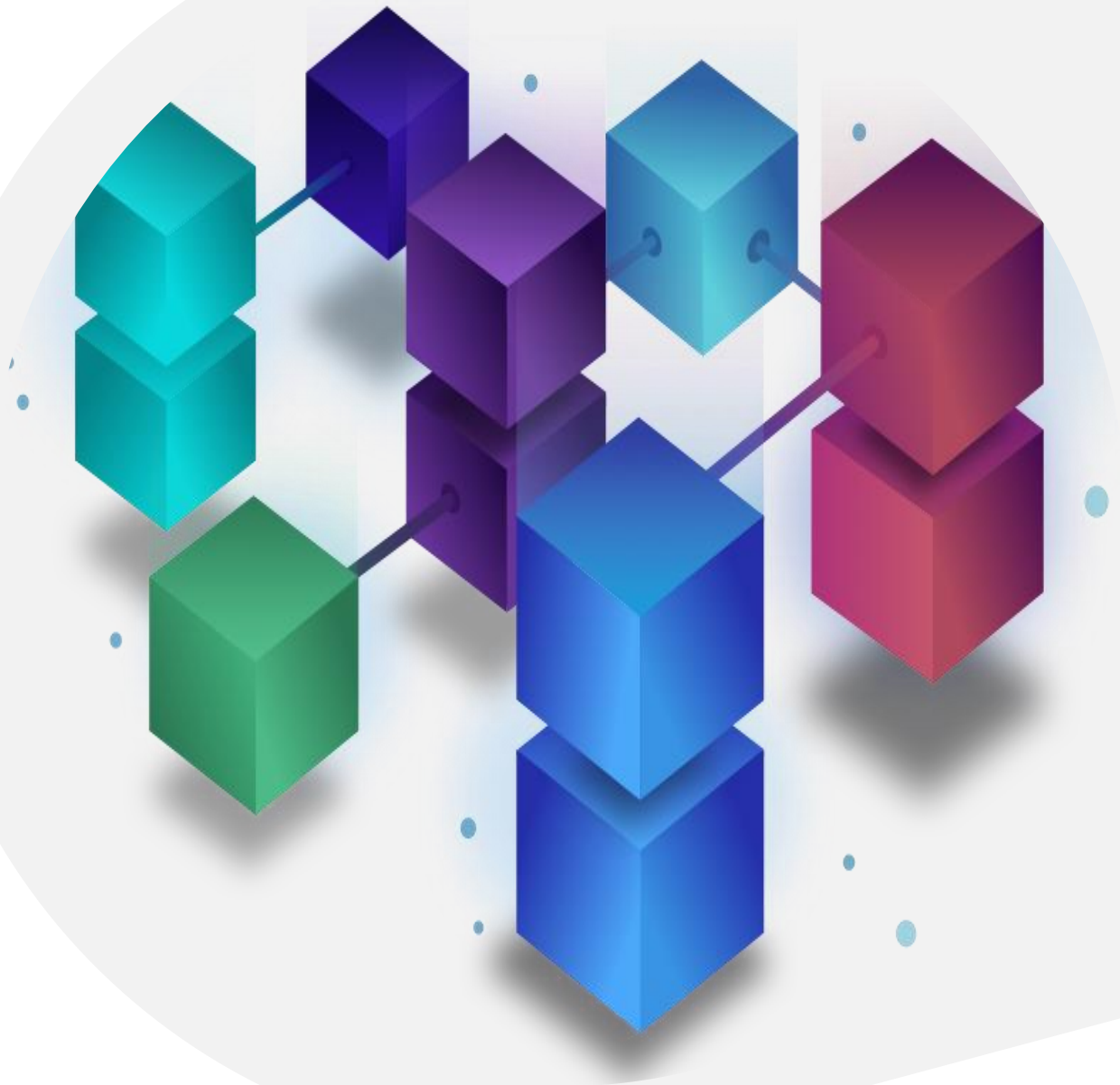
Variables

let

let x = 5.6;
let has a block scope.
let is the block scoped version of var and is limited to the block (or expression)it is defined.
The variables declared inside a block like for loop will be available with the block.

const
const x = 5.6;
const is a variable that once it has been created, its value can never change.
const has a block scope.

Rest Components



React Components

Components are independent and reusable bits of code.
They serve the same purpose as JavaScript functions, but work in isolation and return HTML via a render() function.

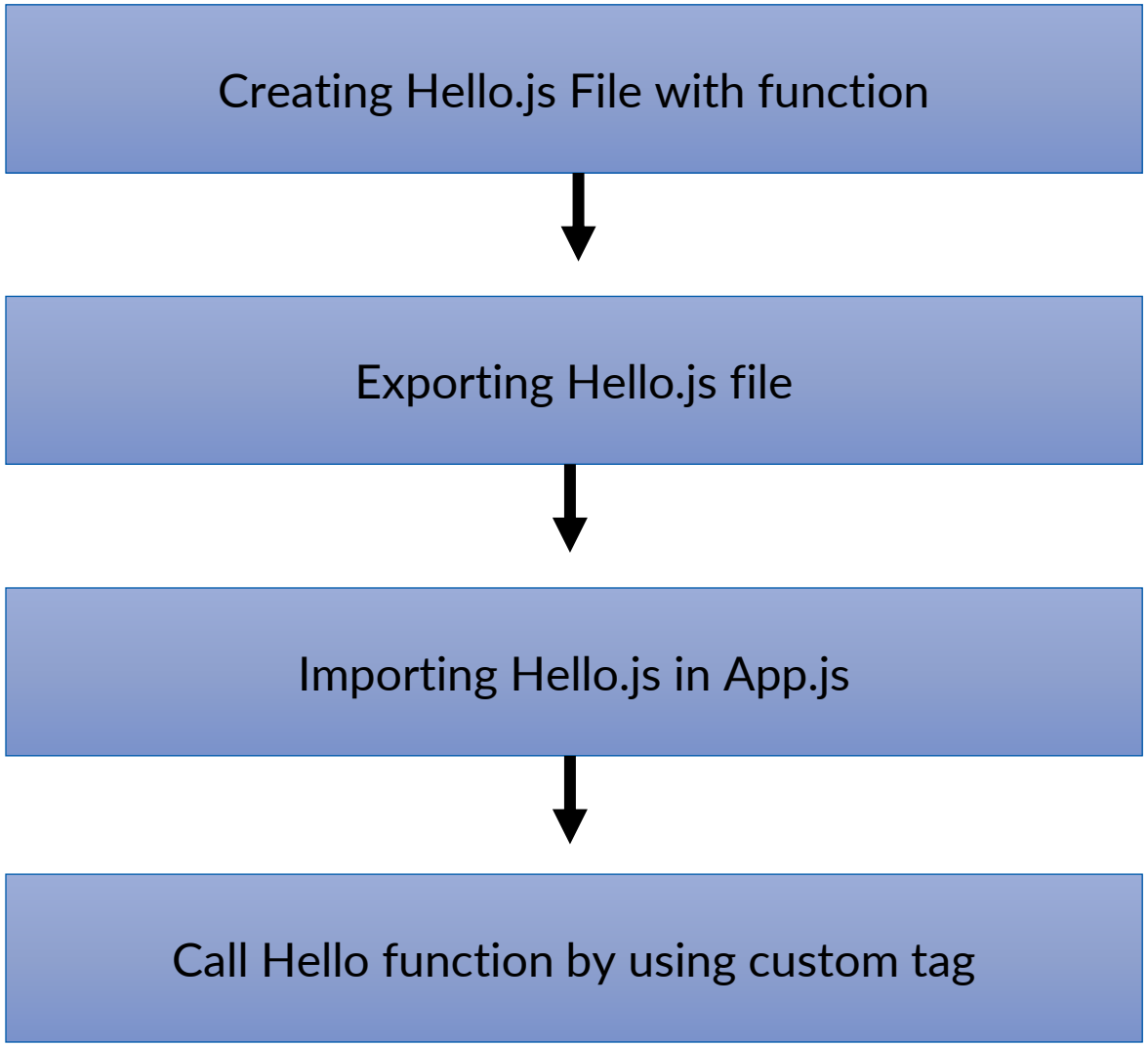
Components come in two types:

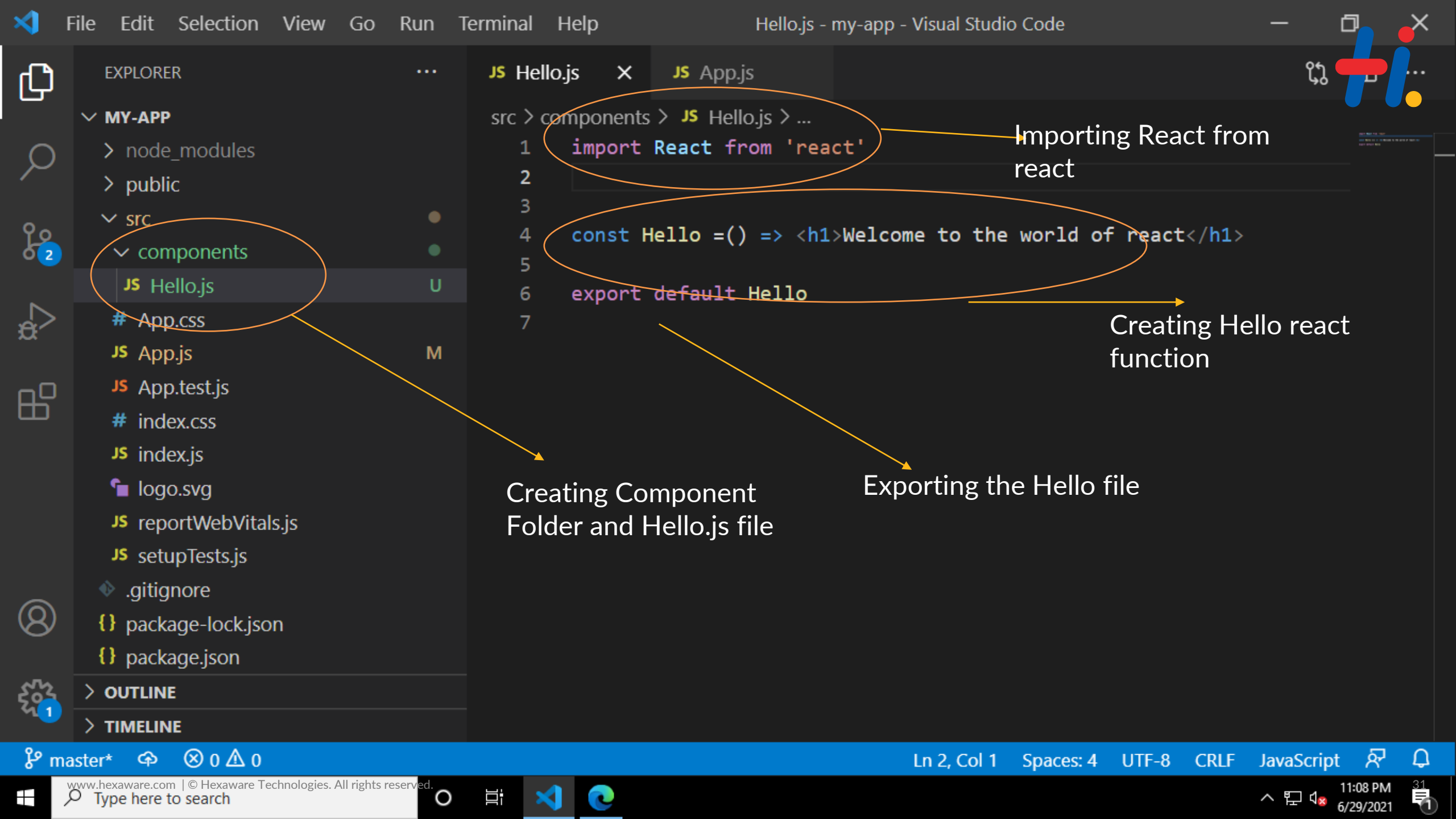
- Class components
- Function components

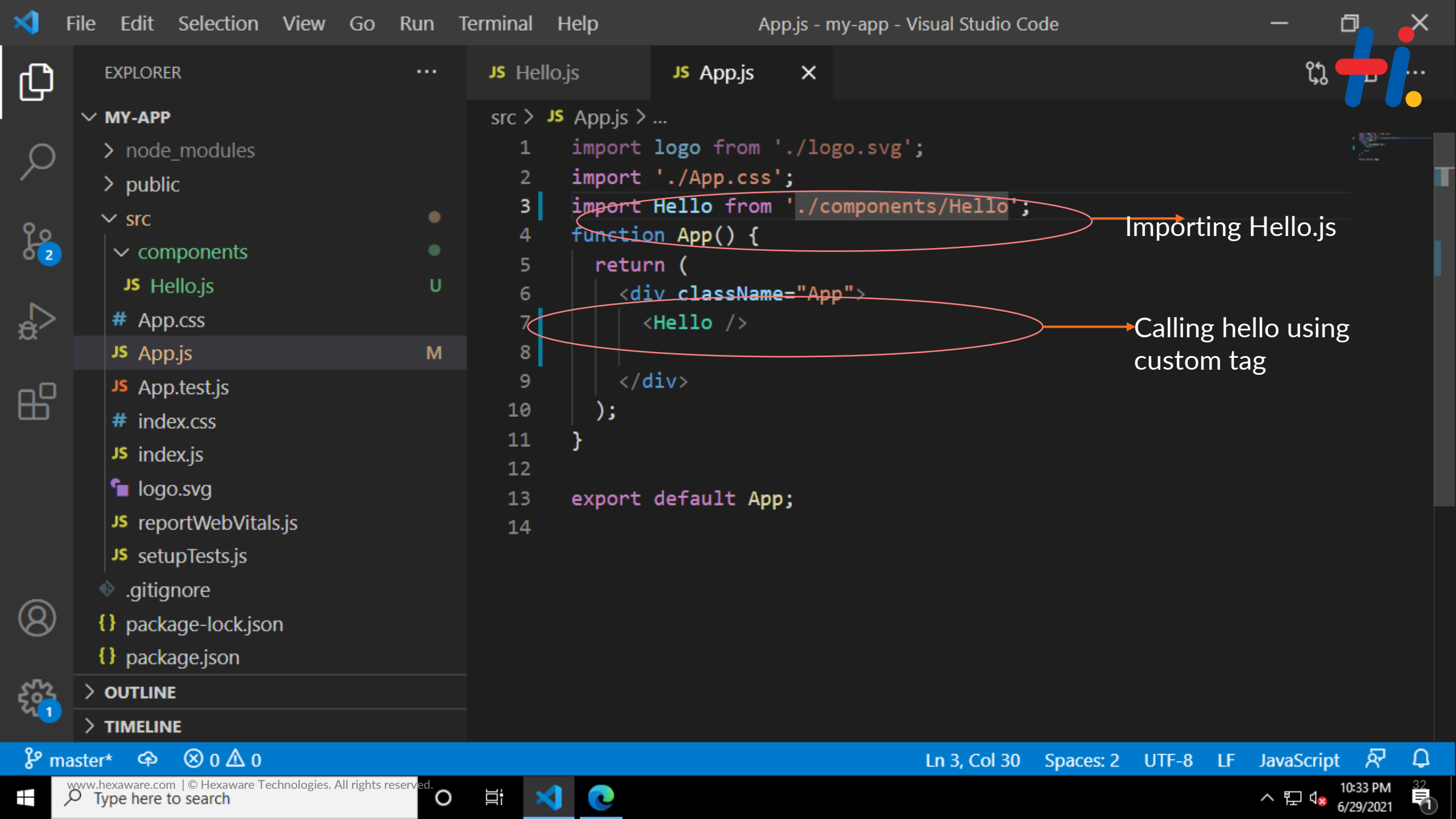
Functional Component

- In React, function components are JavaScript functions
- It receives optionally object of the properties(props) and returns html(JSX) for UI

Steps to creating Functional Component







React App

localhost:3000

welcome to the world of React

Output in browser

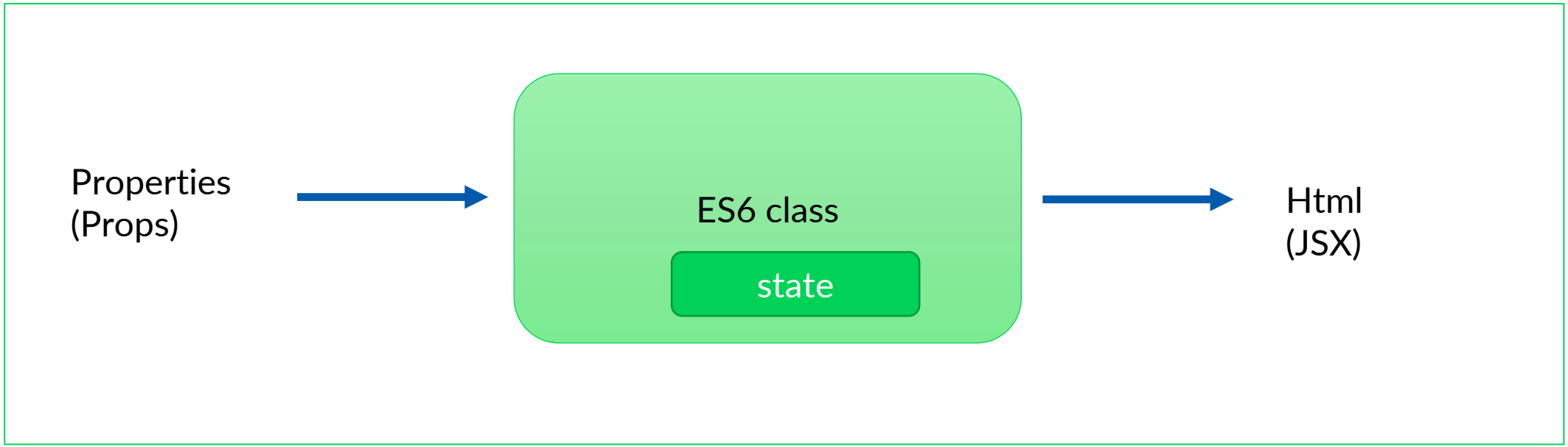
Type here to search

10:33 PM
6/29/2021

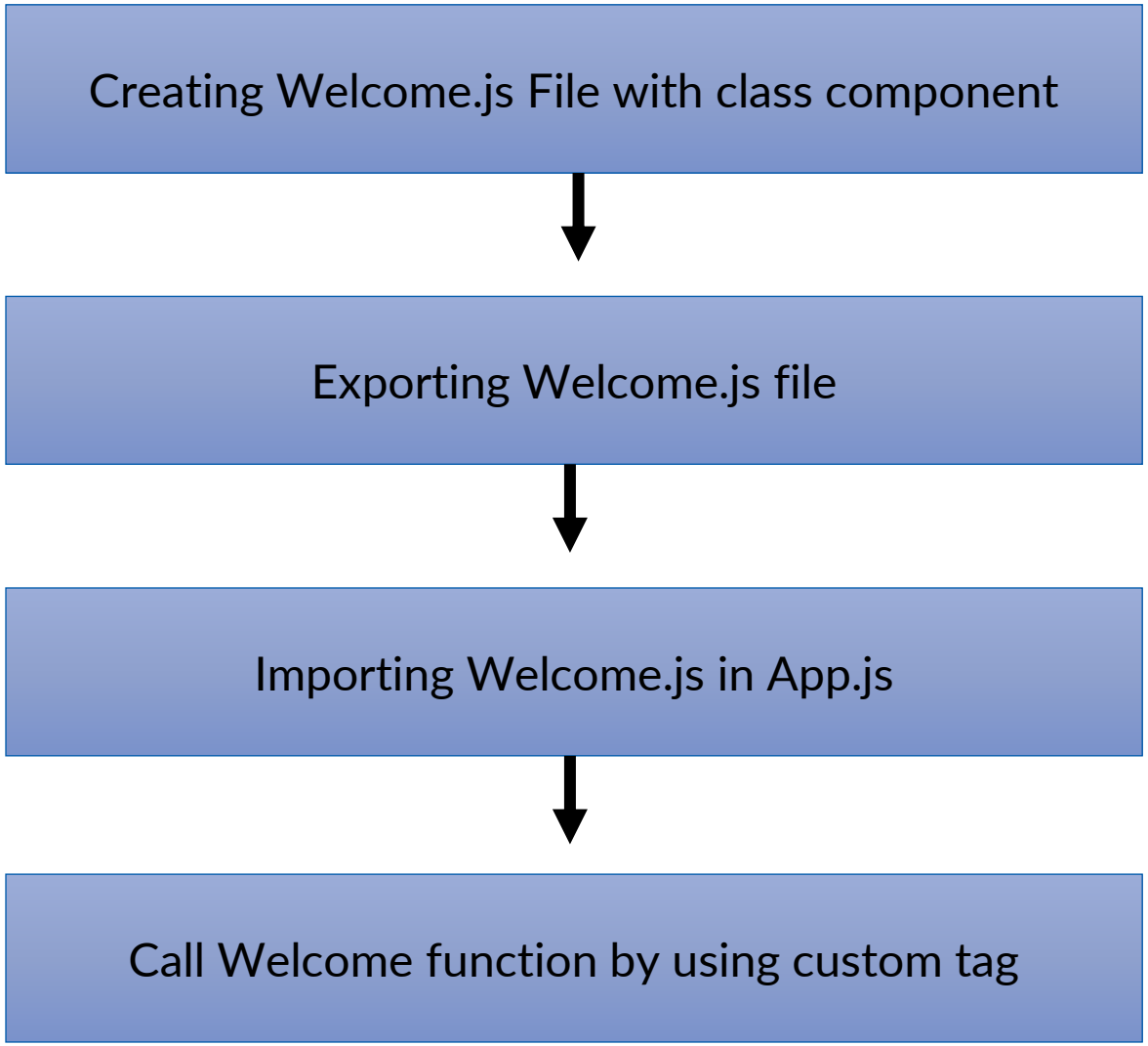
Class Component

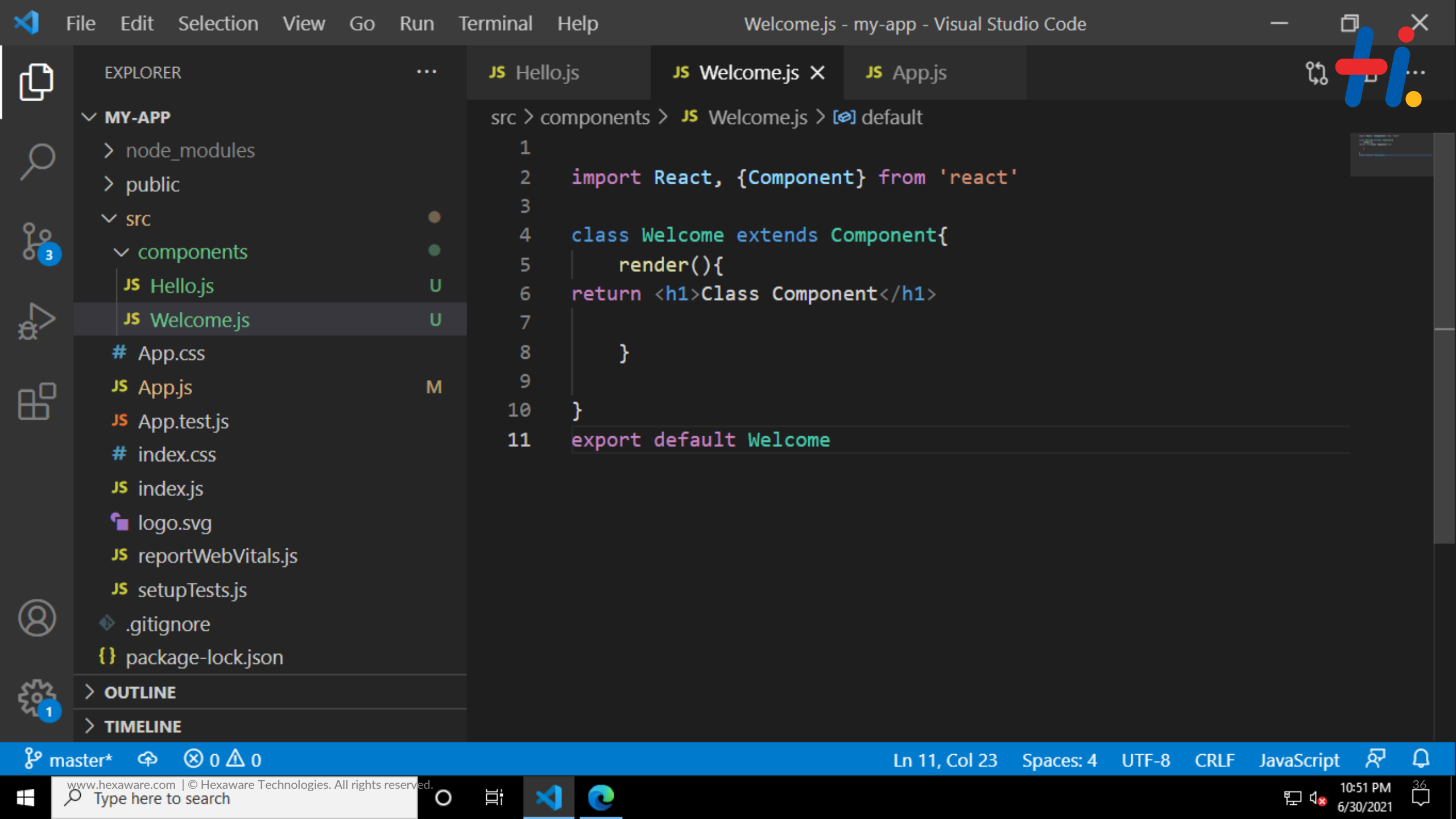
class components are basically EX6 classes

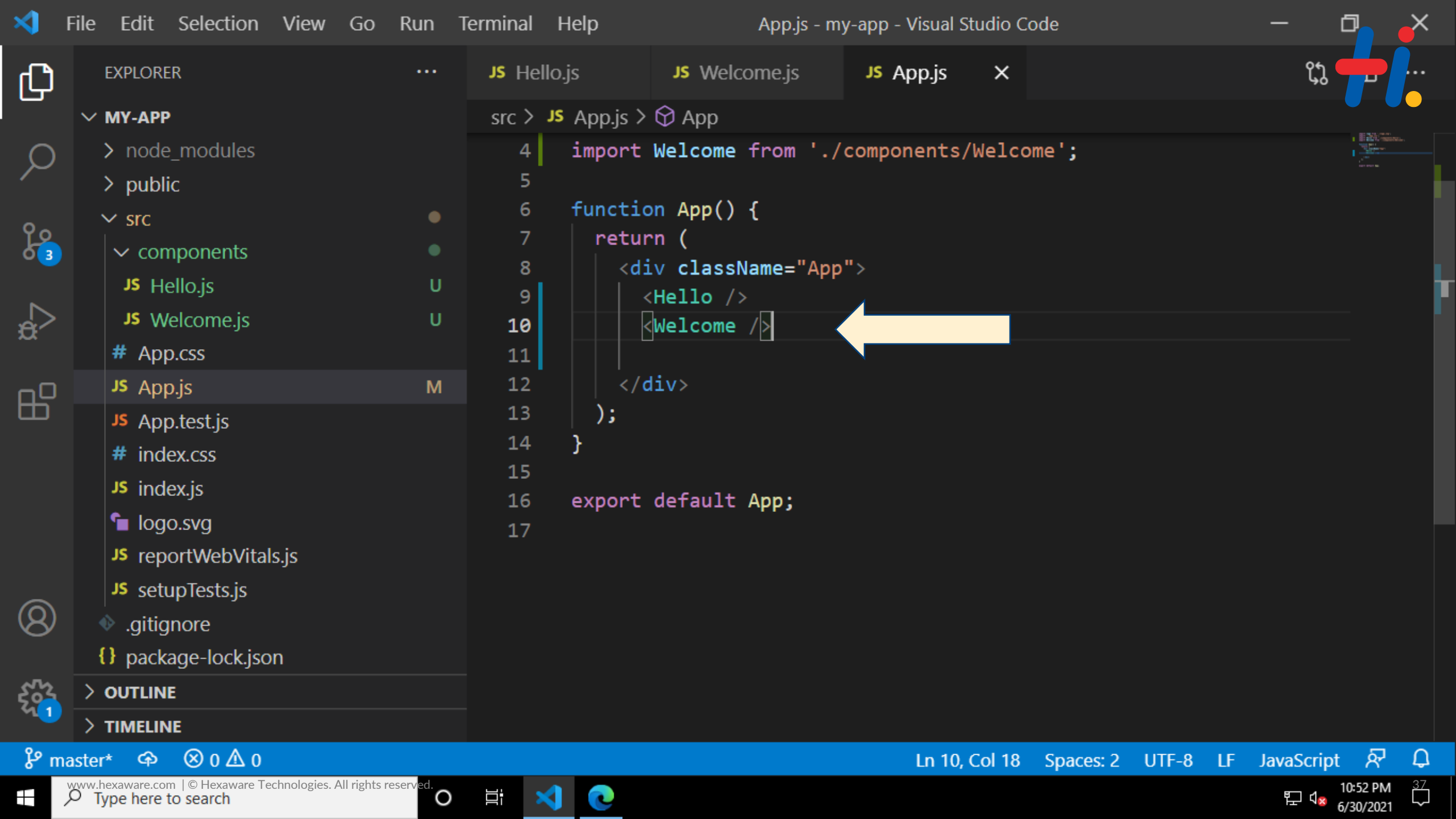
class component will maintain internal state, private to that component



Steps to creating Class Component







Functional Vs Class Component

Functional	Class
Simple Function	Rich and Complex feature
State was not maintained	Maintain their private state
Used to create simple UI	Used to create complex UI
Doesn't have Life cycle hook	Provide Life cycle hook
Stateless/Dump/Presentational	Stateful/Smart/Container

JSX

JSX stands for JavaScript XML.

React uses JSX for templating instead of regular JavaScript.

It is faster because it performs optimization while compiling code to JavaScript.

It is also type-safe and most of the errors can be caught during compilation.

It makes it easier and faster to write templates, if you are familiar with HTML.

JSX - Syntax

```
const Hello =() =>{
return (
  <div>
       <h1> Hello world </h1>
  </div>
 )   }
```

Without JSX - Syntax

```
const Hello =() =>{
return  React.createElement(
  'div', null,React.createElement('h1',null, 'Hello World')
)
}
```

Thank you

Innovative Services

Passionate Employees

Delighted Customers