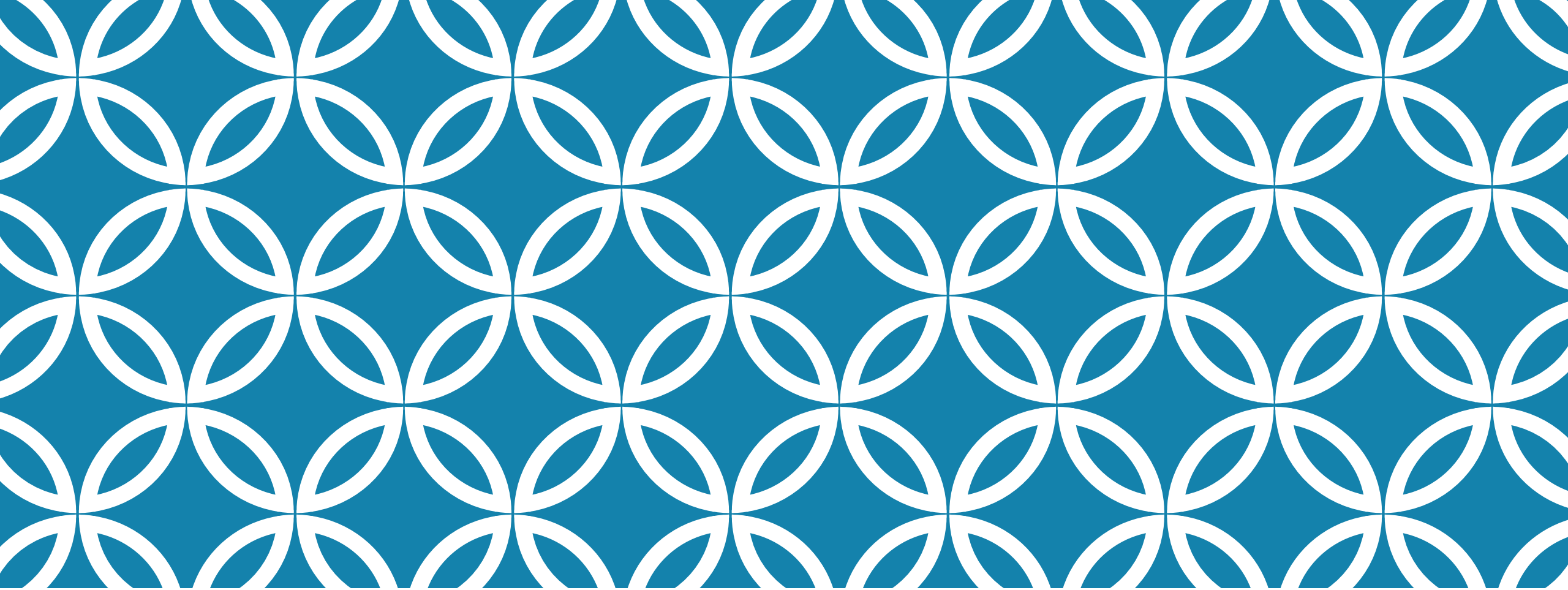


LECTURE #3

By: Aqib Rehman

OUTLINE

Intro to Forms

Create React App

Single Page Apps

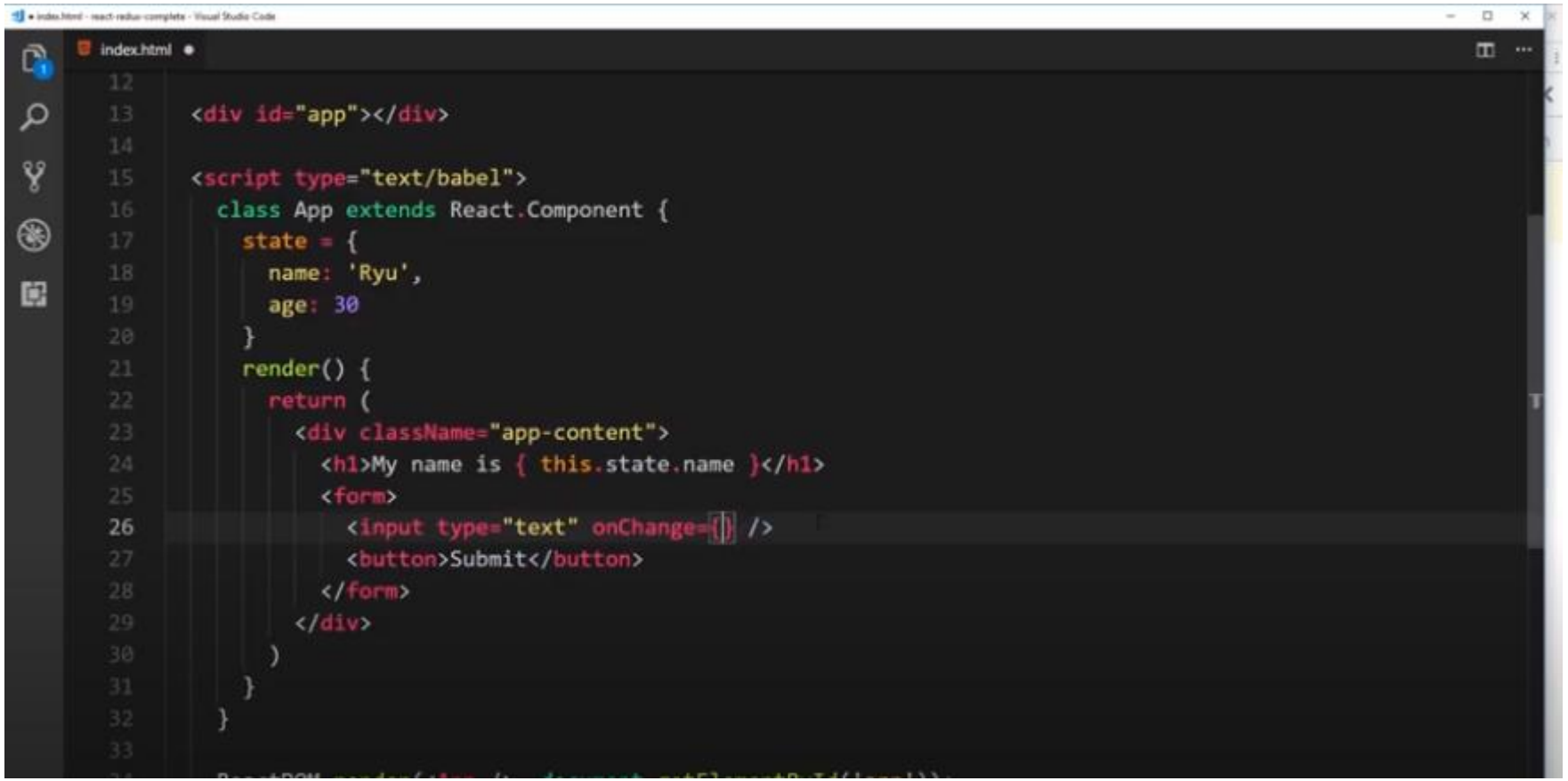
Nesting Components

Props

Outputting Lists

Stateless Components

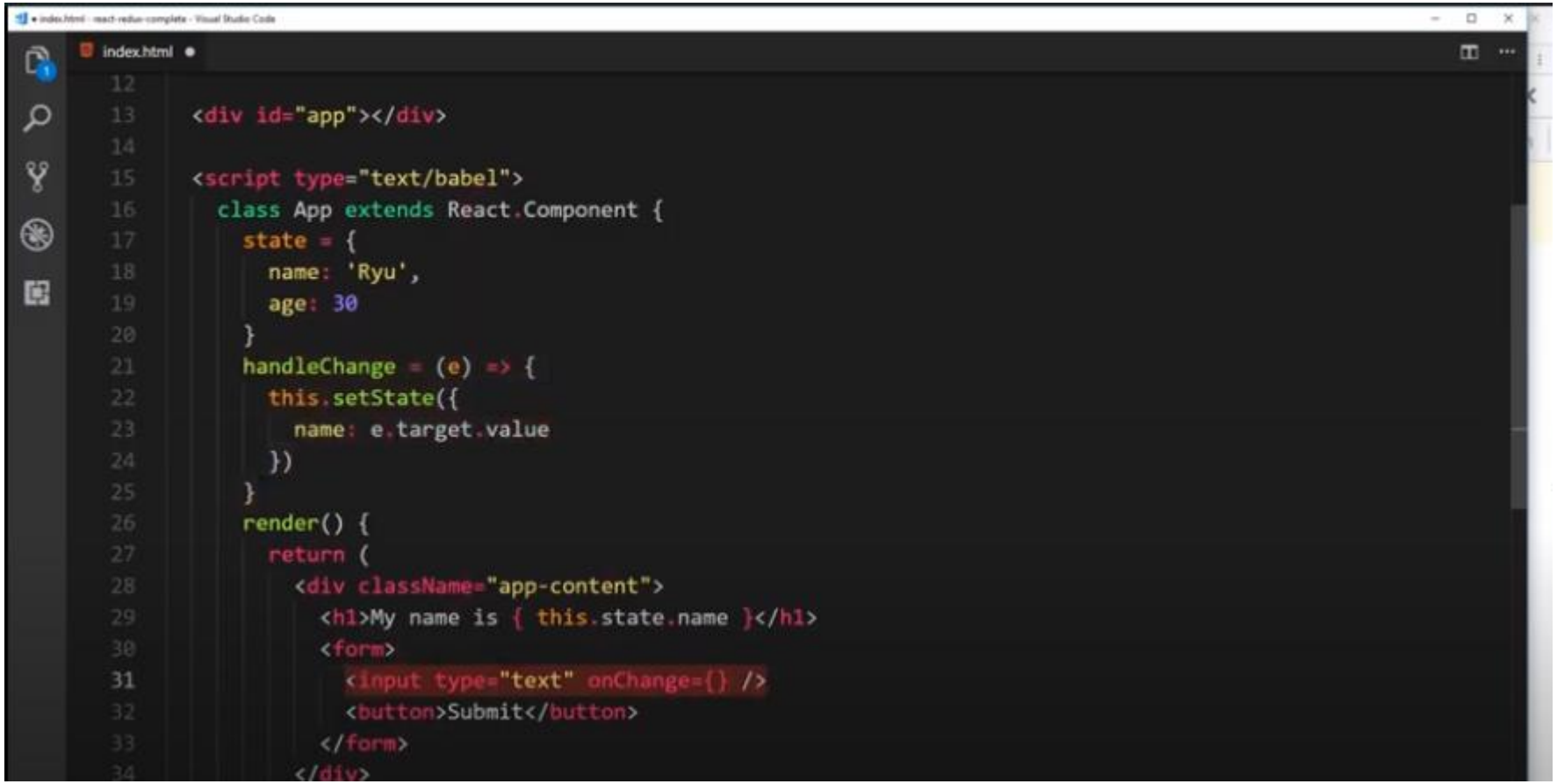
INTRO TO FORMS

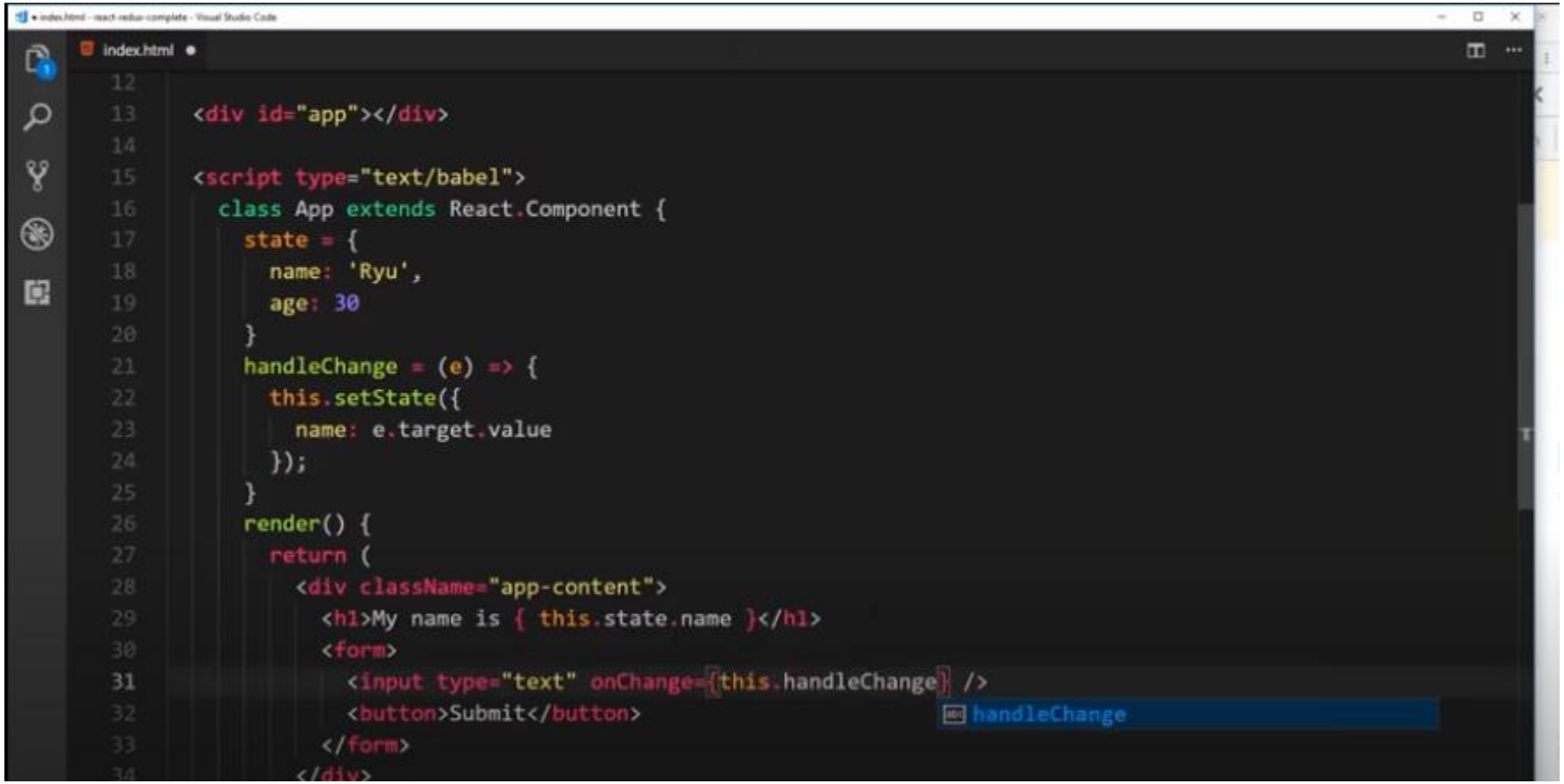

index.html

<script type="text/babel">
class App extends React.Component {
state = {
name: 'Ryu',
age: 30
}
render() {
return (

<h1>My name is { this.state.name }</h1>
<form>
<input type="text" onChange={} />
<button>Submit</button>
</form>

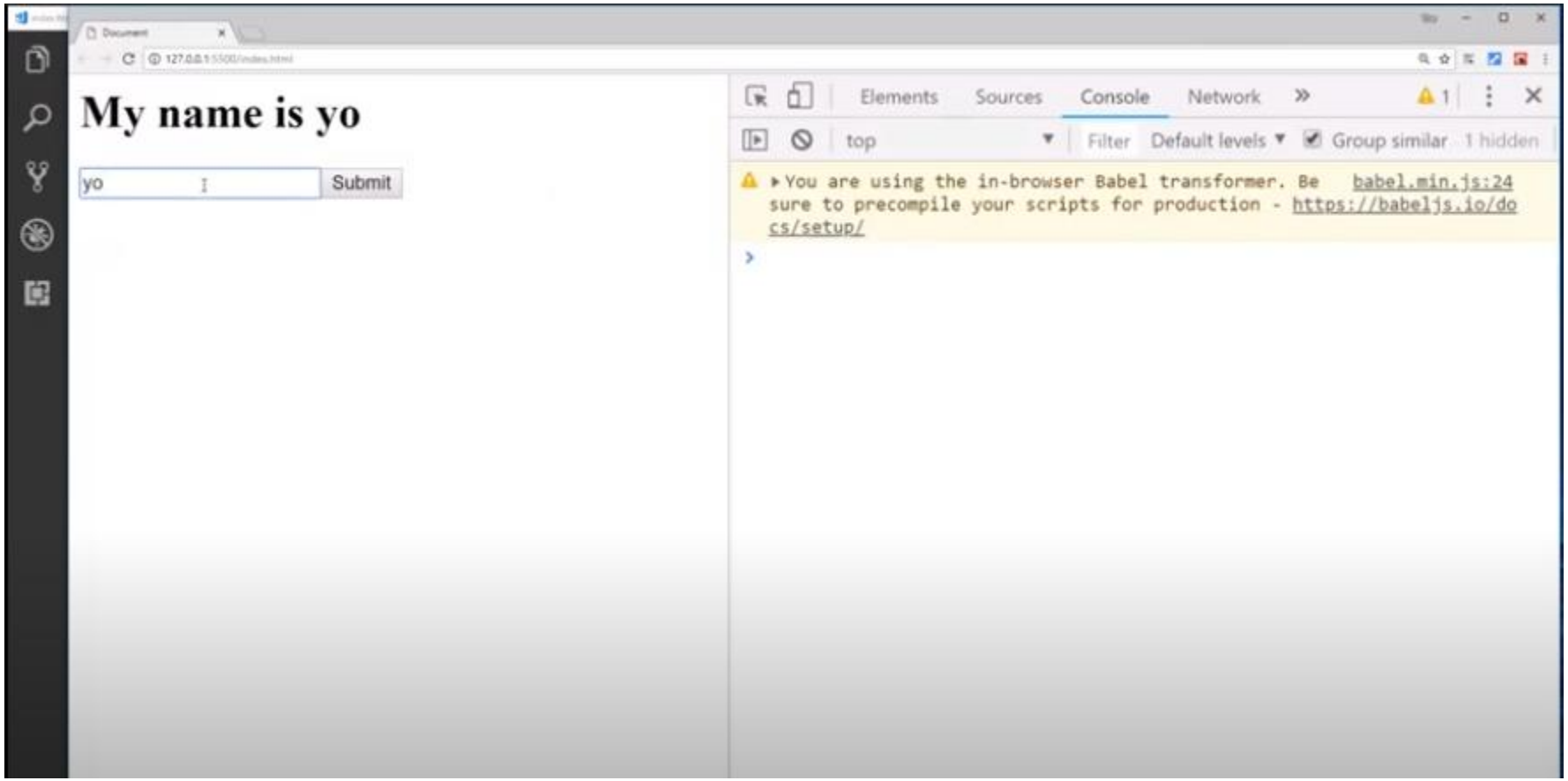
)
}
}


index.html
12
13
14
15 <script type="text/babel">
16 class App extends React.Component {
17 state = {
18 name: 'Ryu',
19 age: 30
20 }
21 handleChange = (e) => {
22 this.setState({
23 name: e.target.value
24 })
25 }
26 render() {
27 return (
28
29 <h1>My name is { this.state.name }</h1>
30 <form>
31 <input type="text" onChange={} />
32 <button>Submit</button>
33 </form>
34

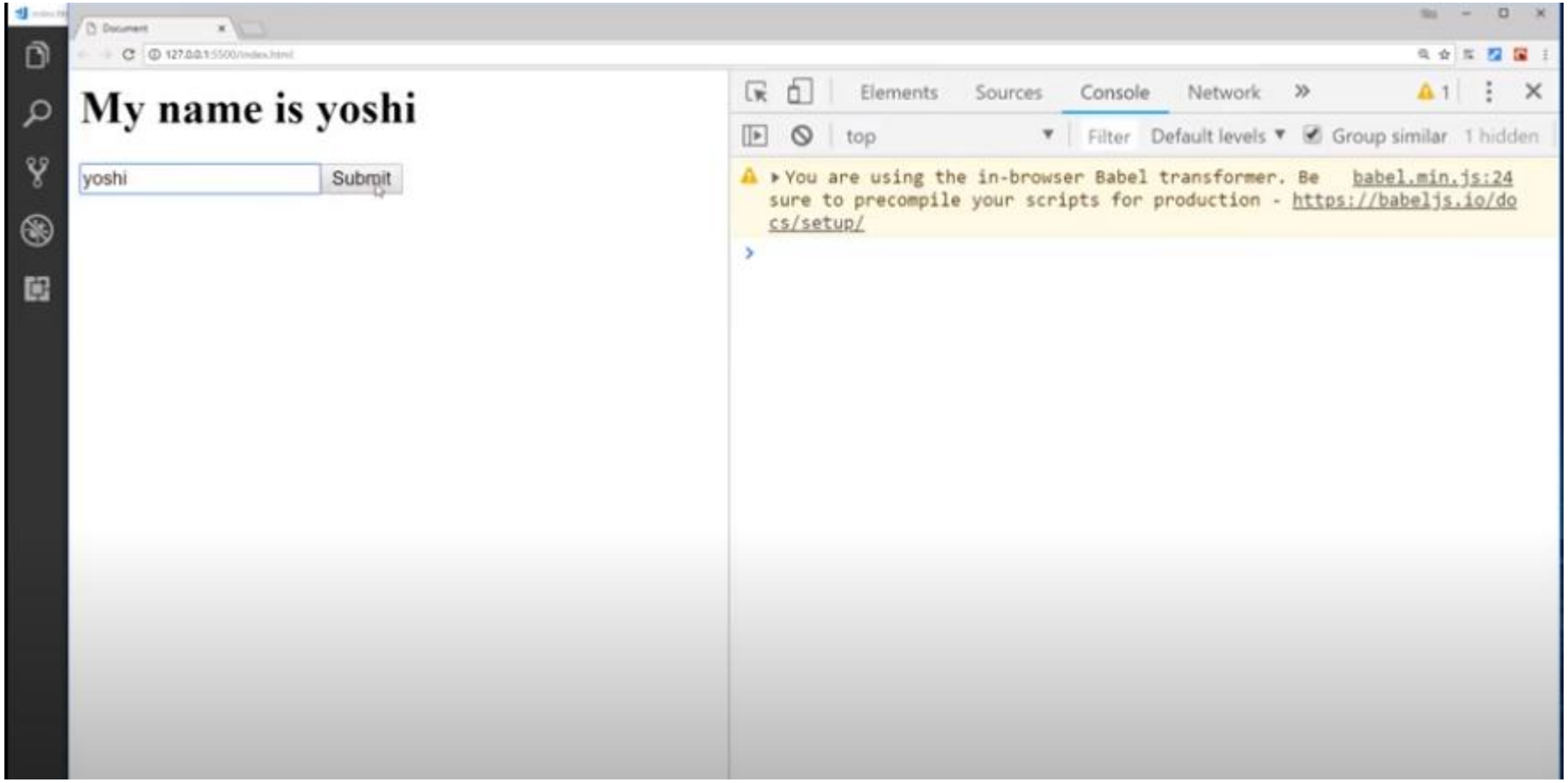

index.html

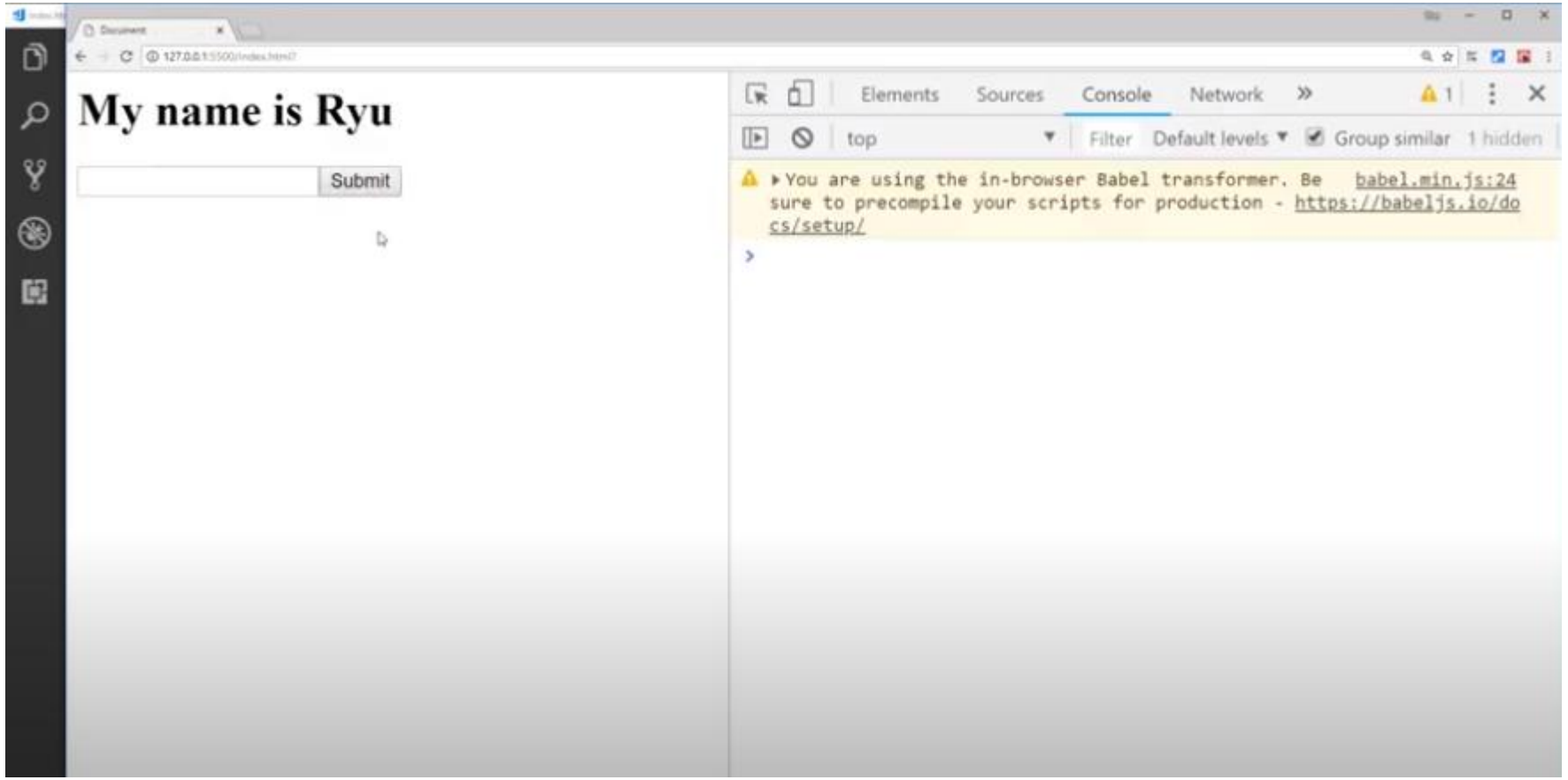
<script type="text/babel">
class App extends React.Component {
state = {
name: 'Ryu',
age: 30
}
handleChange = (e) => {
this.setState({
name: e.target.value
});
}
render() {
return (

<h1>My name is { this.state.name }</h1>
<form>
<input type="text" onChange={this.handleChange} />
<button>Submit</button>
</form>
handleChange

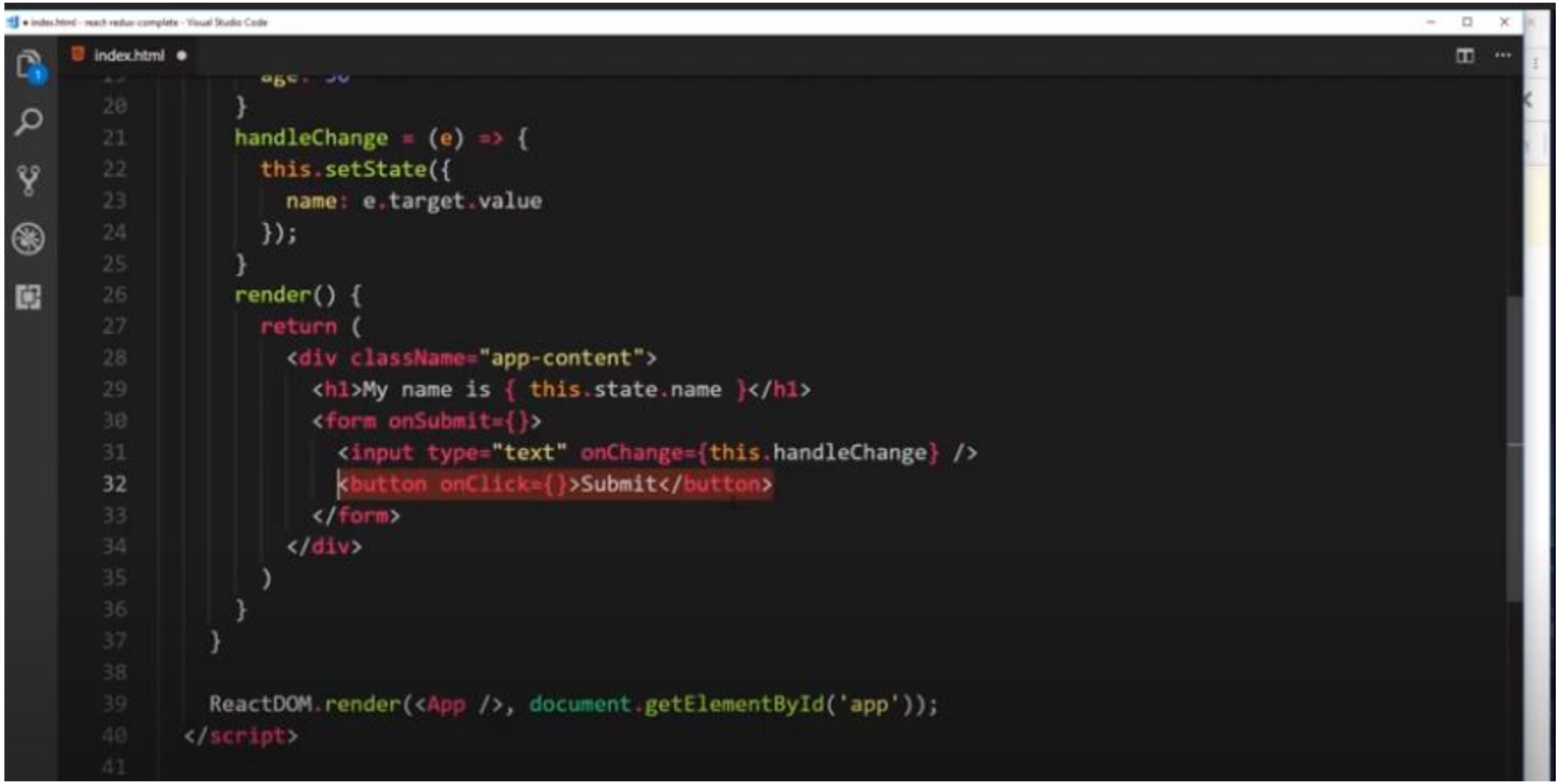

Document
127.0.0.1:5500/index.html
My name is yo
yo
Submit
Elements
Sources
Console
Network
1
top
Filter
Default levels
Group similar
1 hidden
You are using the in-browser Babel transformer. Be sure to precompile your scripts for production - https://babeljs.io/docs/setup/
babel.min.js:24

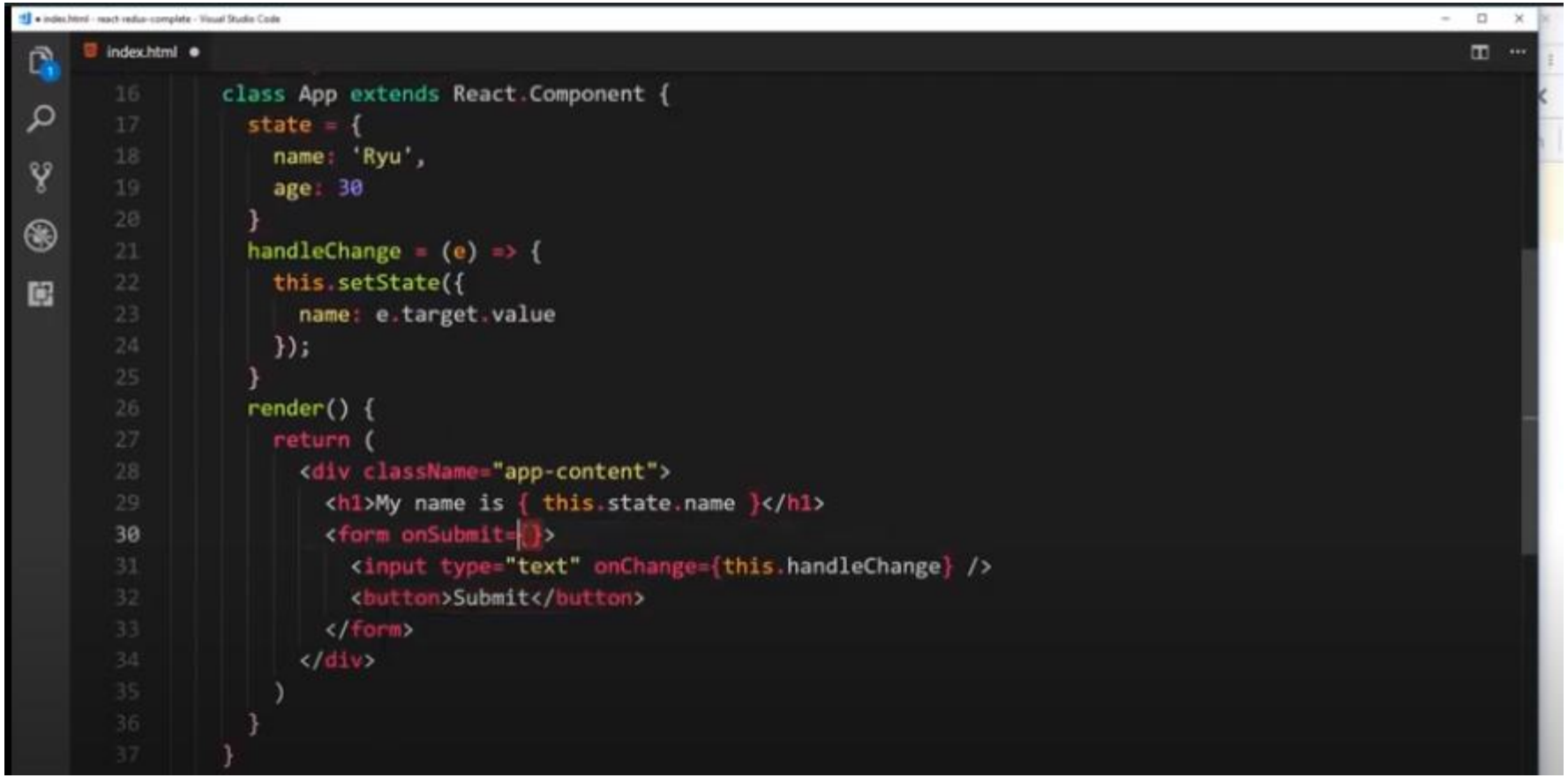
When click on submit page will be refresh default event of form submit




My name is Ryu
Submit
Elements
Sources
Console
Network
top
Filter
Default levels
Group similar
1 hidden
You are using the in-browser Babel transformer. Be sure to precompile your scripts for production - https://babeljs.io/docs/setup/
babel.min.js:24

Only trigger when button will be clicked not on Enter pressed

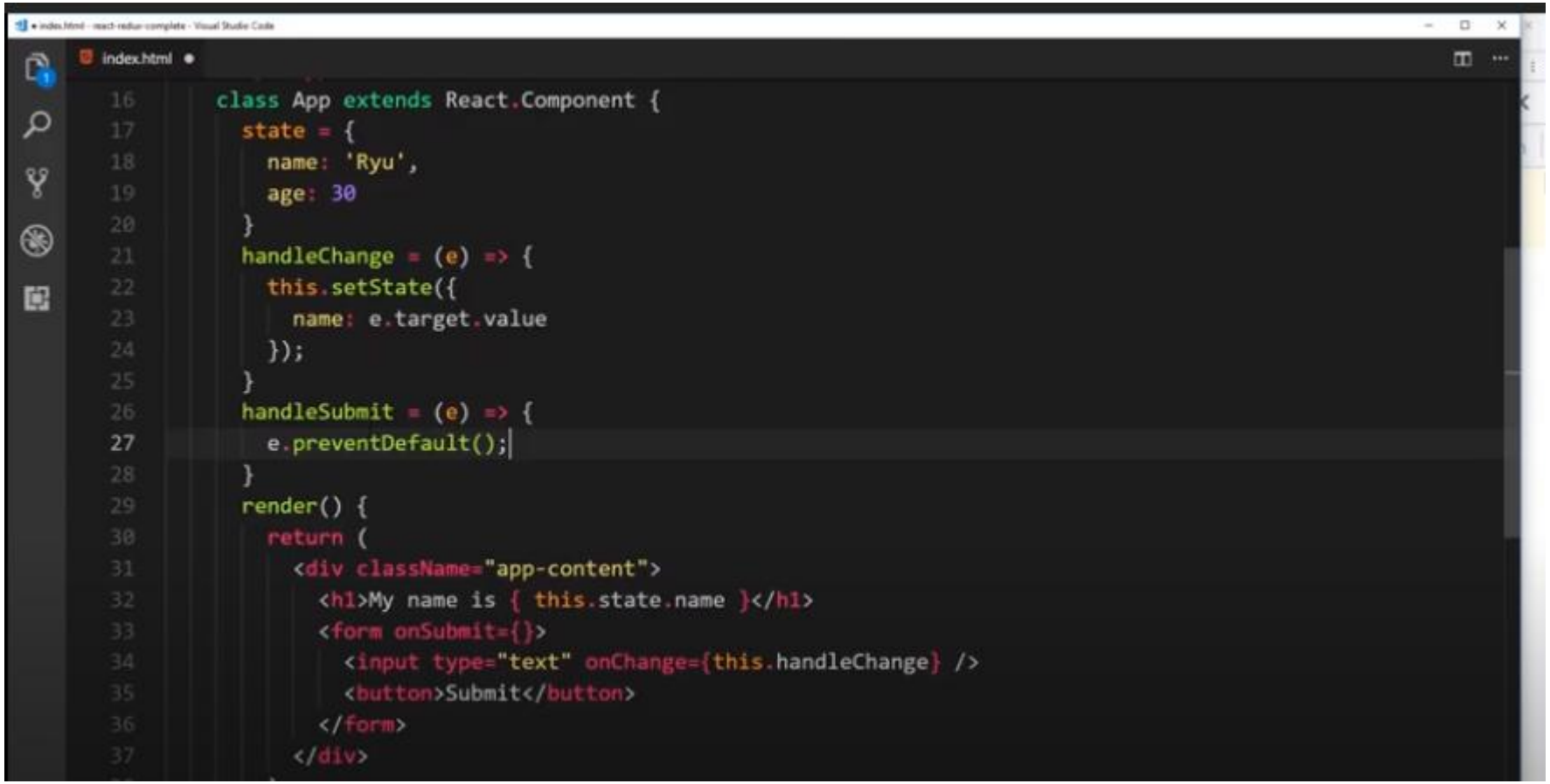


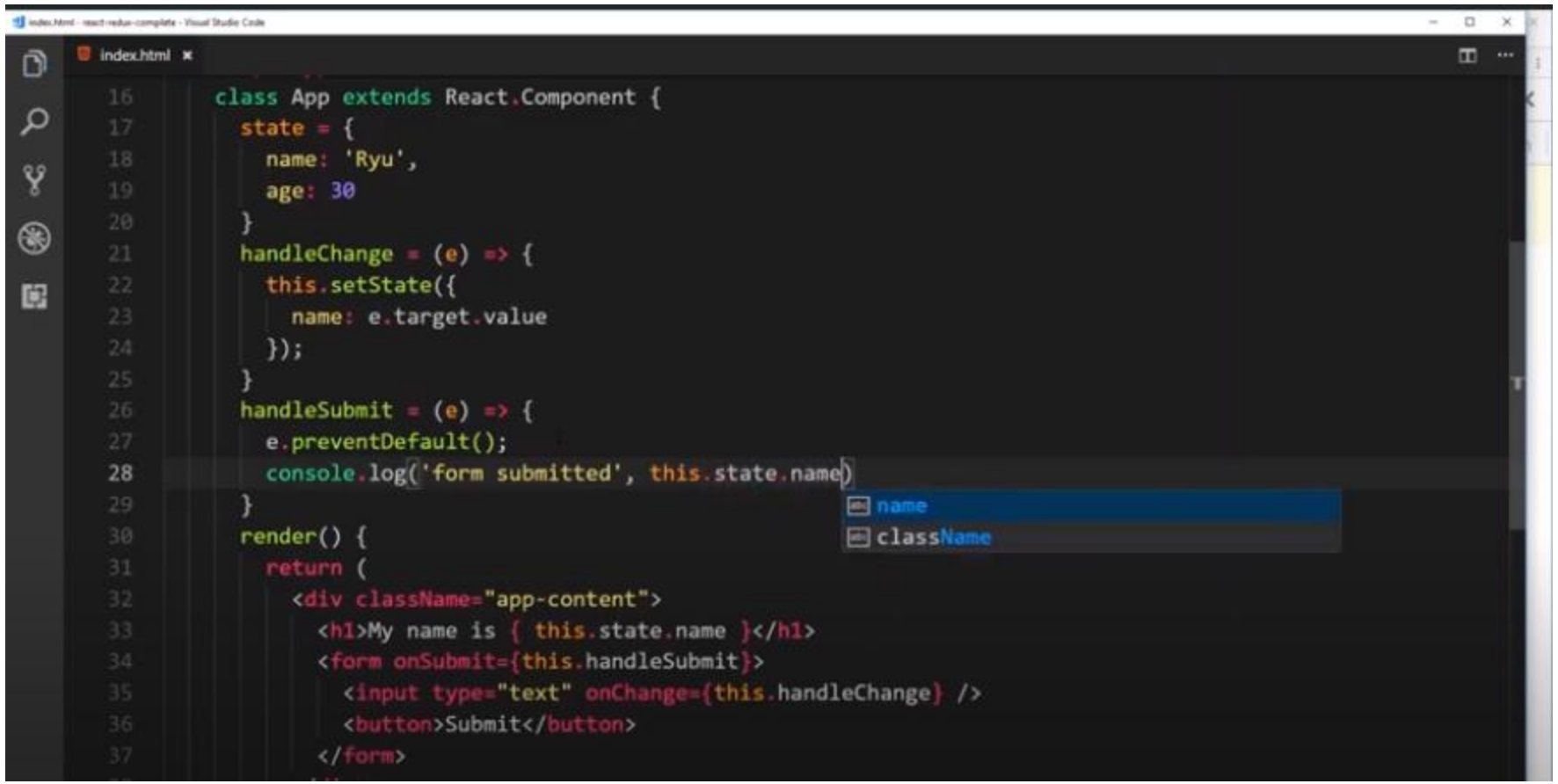

index.html
class App extends React.Component {
state = {
name: 'Ryu',
age: 30
}
handleChange = (e) => {
this.setState({
name: e.target.value
});
}
render() {
return (

<h1>My name is { this.state.name }</h1>
<form onSubmit={}>
<input type="text" onChange={this.handleChange} />
<button>Submit</button>
</form>

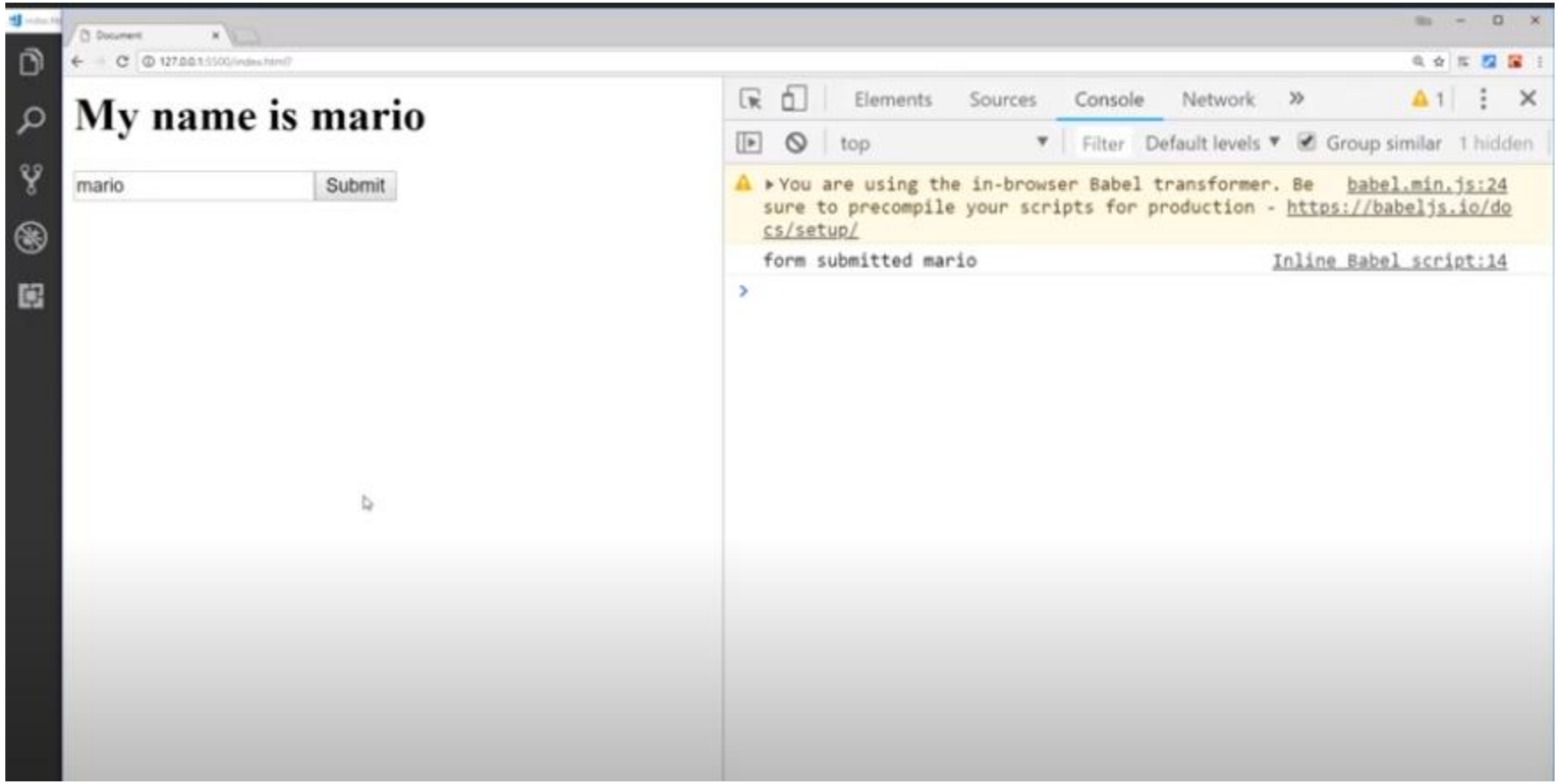
)
}
}

Now page will not be refreshed




index.html
class App extends React.Component {
state = {
name: 'Ryu',
age: 30
}
handleChange = (e) => {
this.setState({
name: e.target.value
});
}
handleSubmit = (e) => {
e.preventDefault();
console.log('form submitted', this.state.name)
}
render() {
return (

<h1>My name is { this.state.name }</h1>
<form onSubmit={this.handleSubmit}>
<input type="text" onChange={this.handleChange} />
<button>Submit</button>
</form>
name
className


Document
127.0.0.1:5500/index.html?
My name is mario
mario
Submit
Elements
Sources
Console
Network
1
top
Filter
Default levels
Group similar
1 hidden
You are using the in-browser Babel transformer. Be sure to precompile your scripts for production - https://babeljs.io/docs/setup/
babel.min.js:24
form submitted mario
Inline Babel script:14

CREATE REACT APP

Create React App

- A command line tool to create React apps
  - Development server
  - Use ES6 features which are not normally supported
  - Keep our code modular
  - Use build tools to create optimized code

Quick Overview

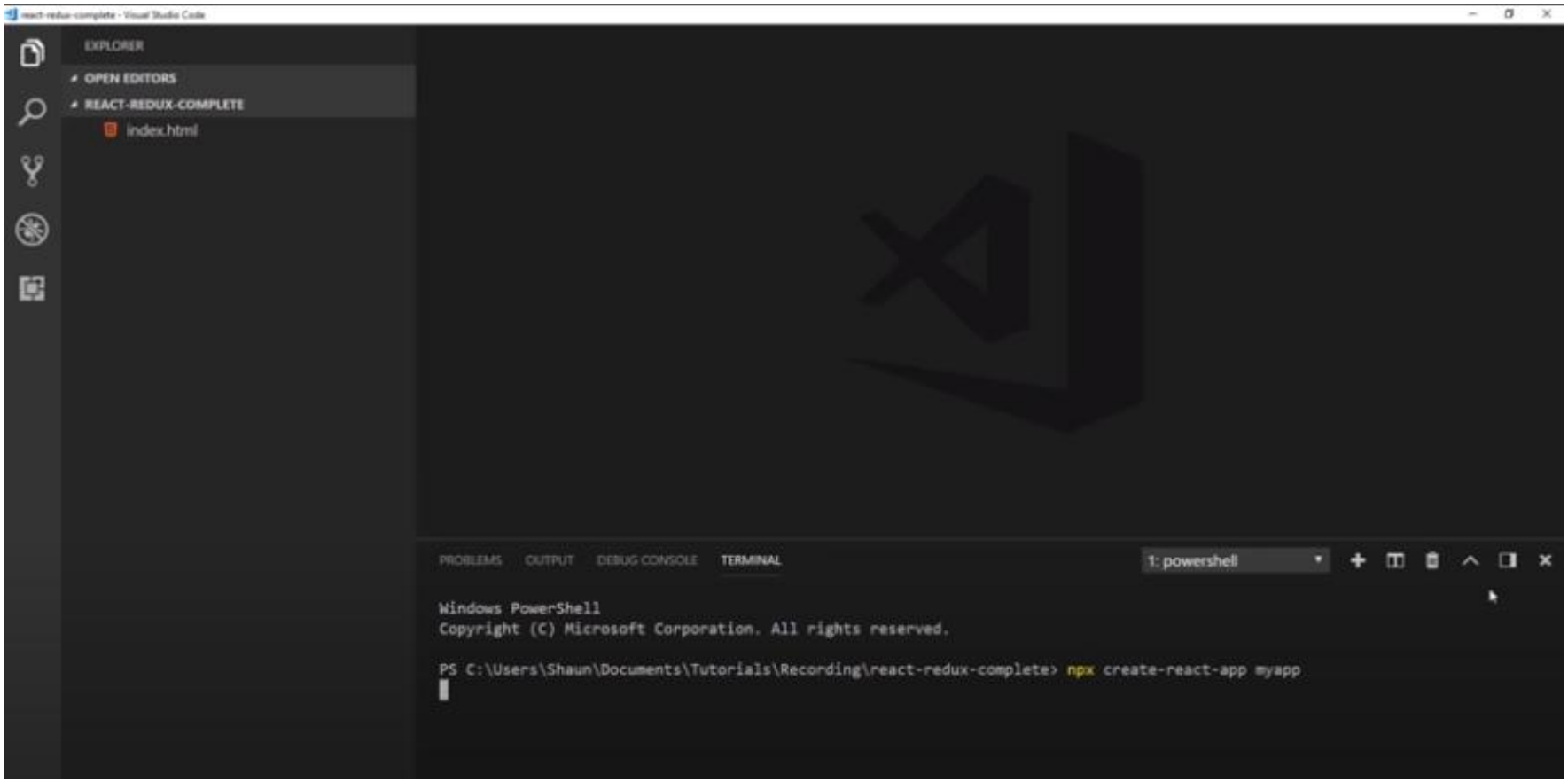
```
npx create-react-app my-app
cd my-app
npm start
```

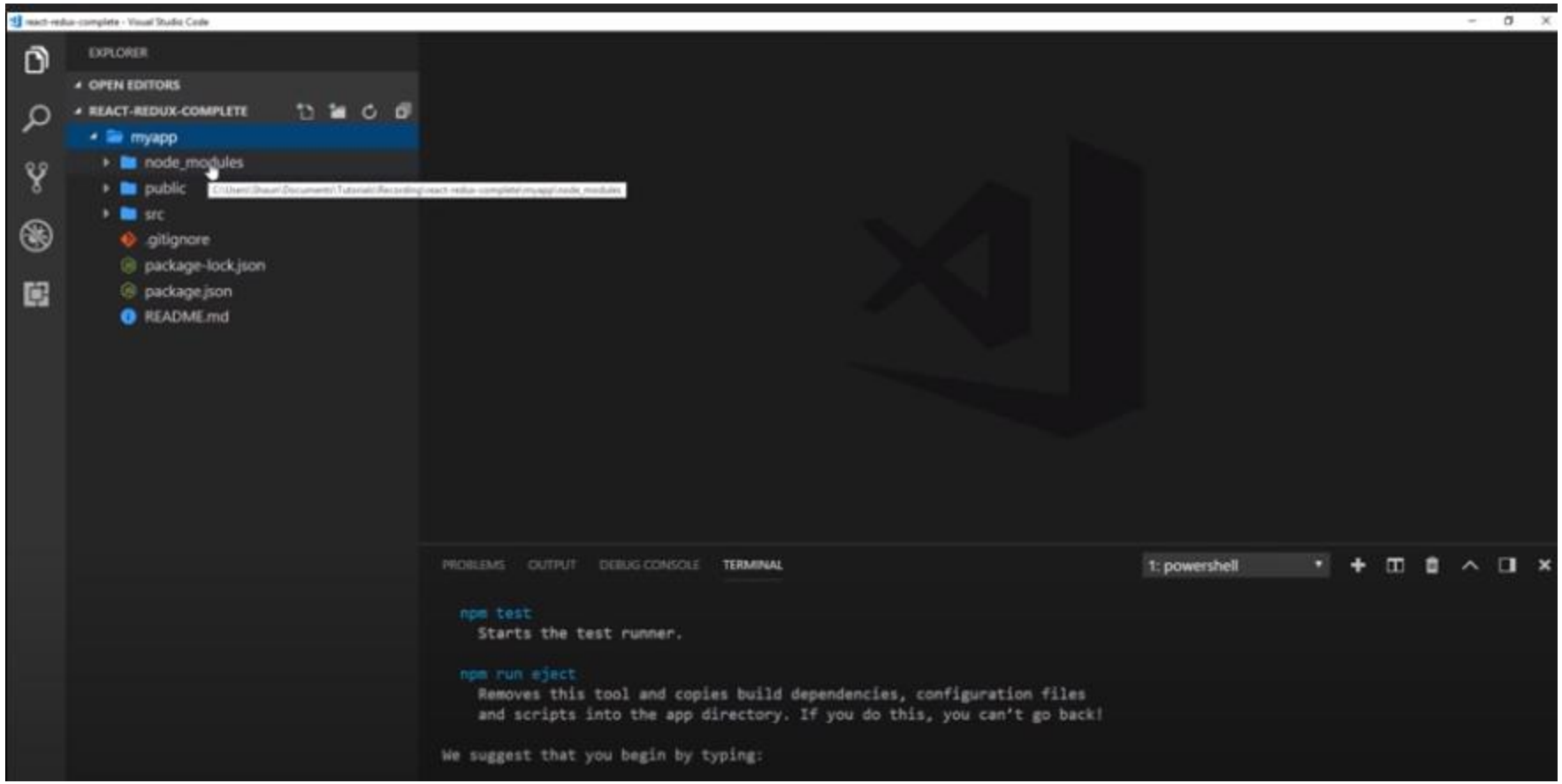
(npx comes with npm 5.2+ and higher, see instructions for older npm versions)

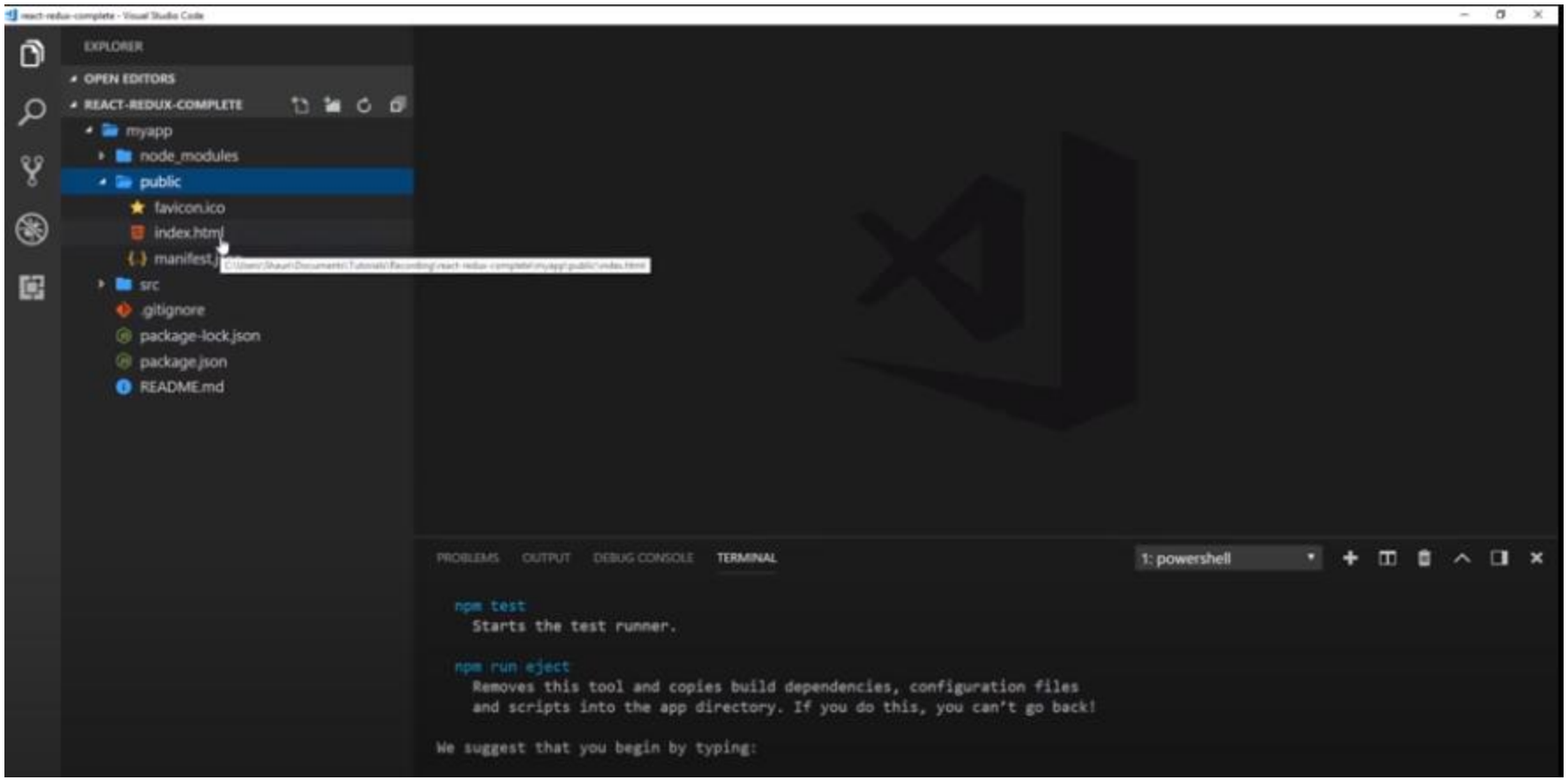
Then open http://localhost:3000/ to see your app.
When you're ready to deploy to production, create a minified bundle with `npm run build`.

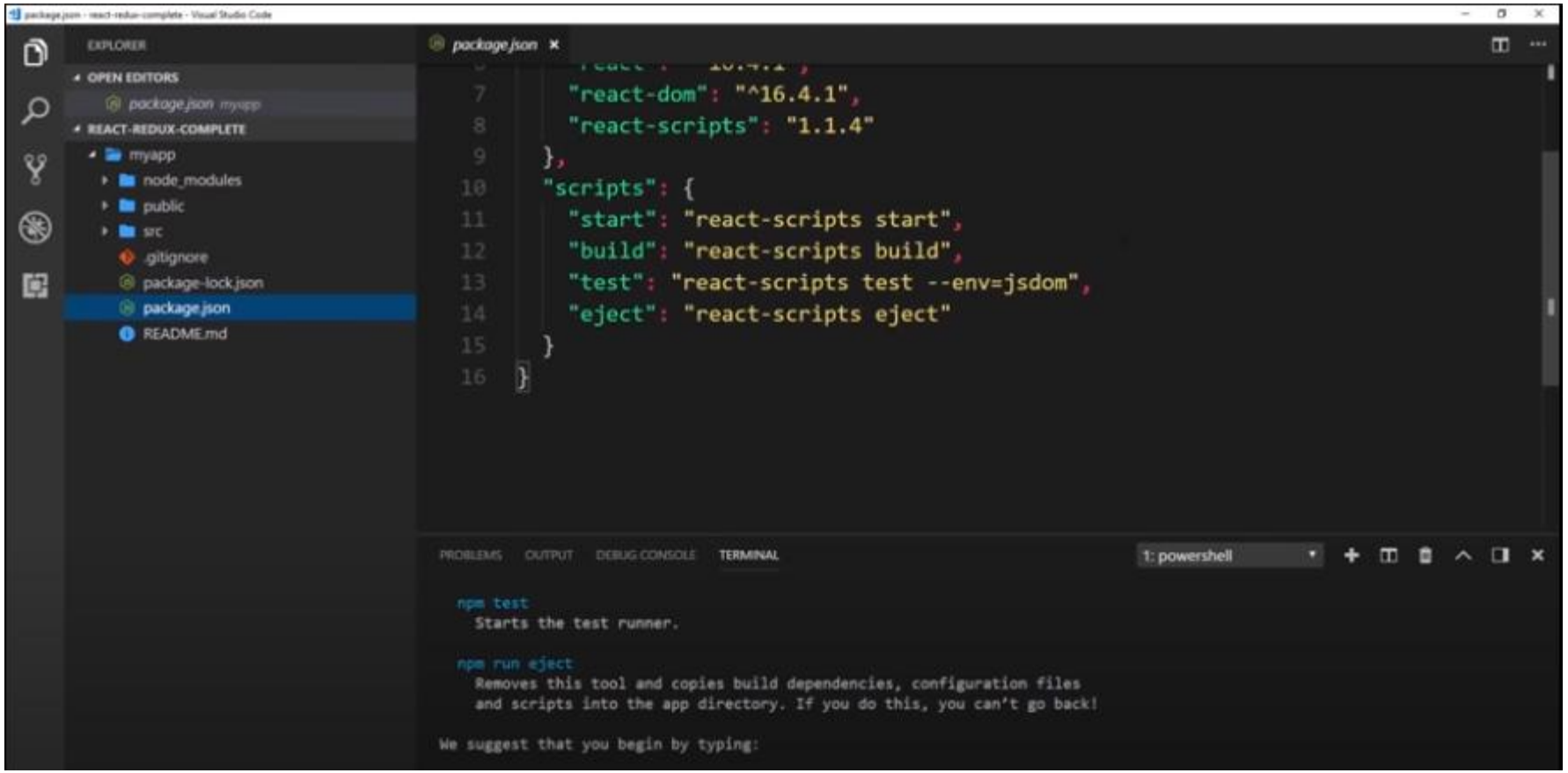

Secure | https://nodejs.org/en/
node
HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS | FOUNDATION
Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.
JS Interactive
Join us at #jsinteractive
October 10-12, 2018
Vancouver, Canada
Download for Windows (x64)
8.11.3 LTS
Recommended For Most Users
10.7.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs
Or have a look at the Long Term Support (LTS) schedule.
Sign up for Node.js Everywhere, the official Node.js Weekly Newsletter.

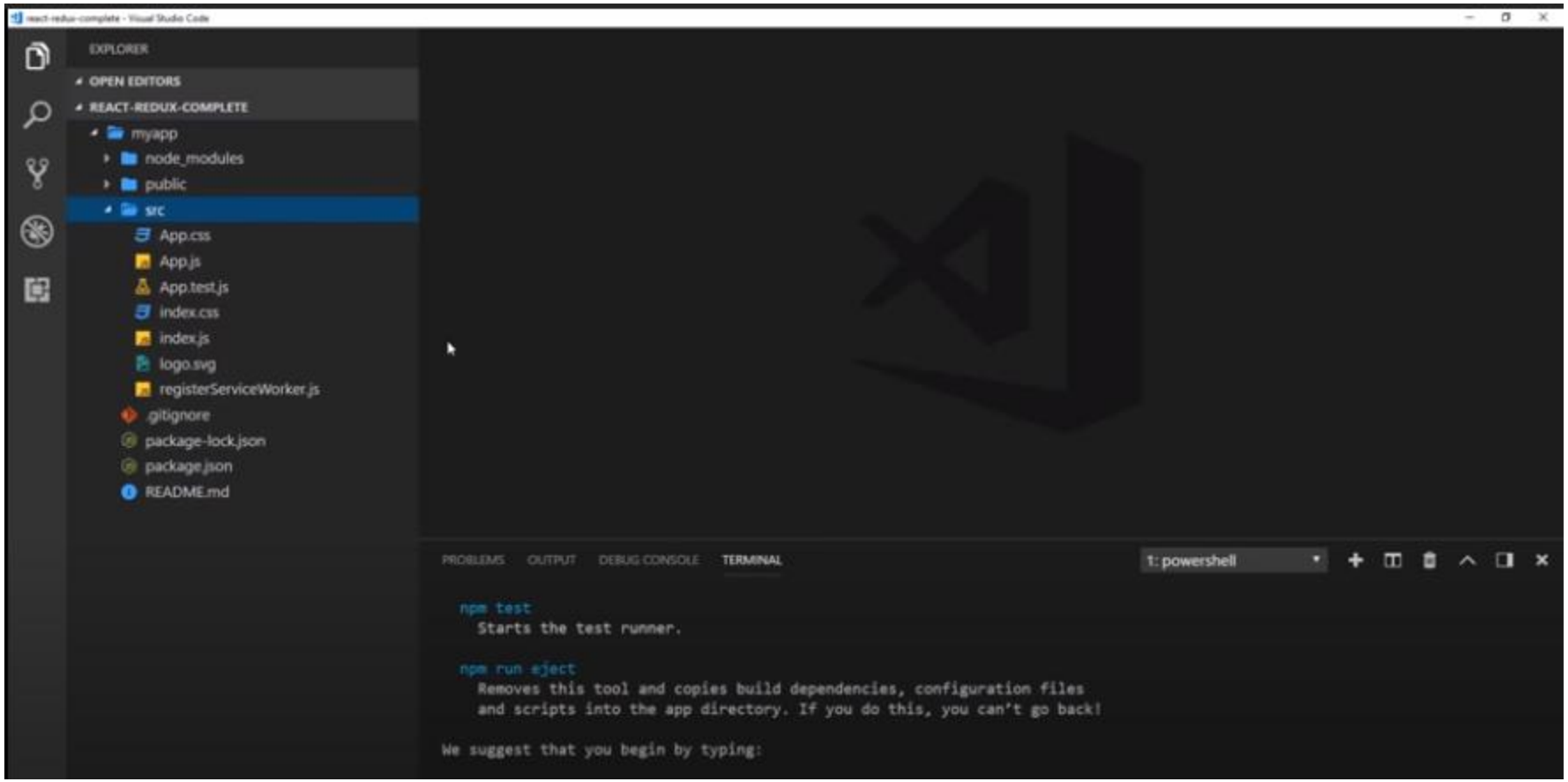
CREATING APP WITH COMMAND

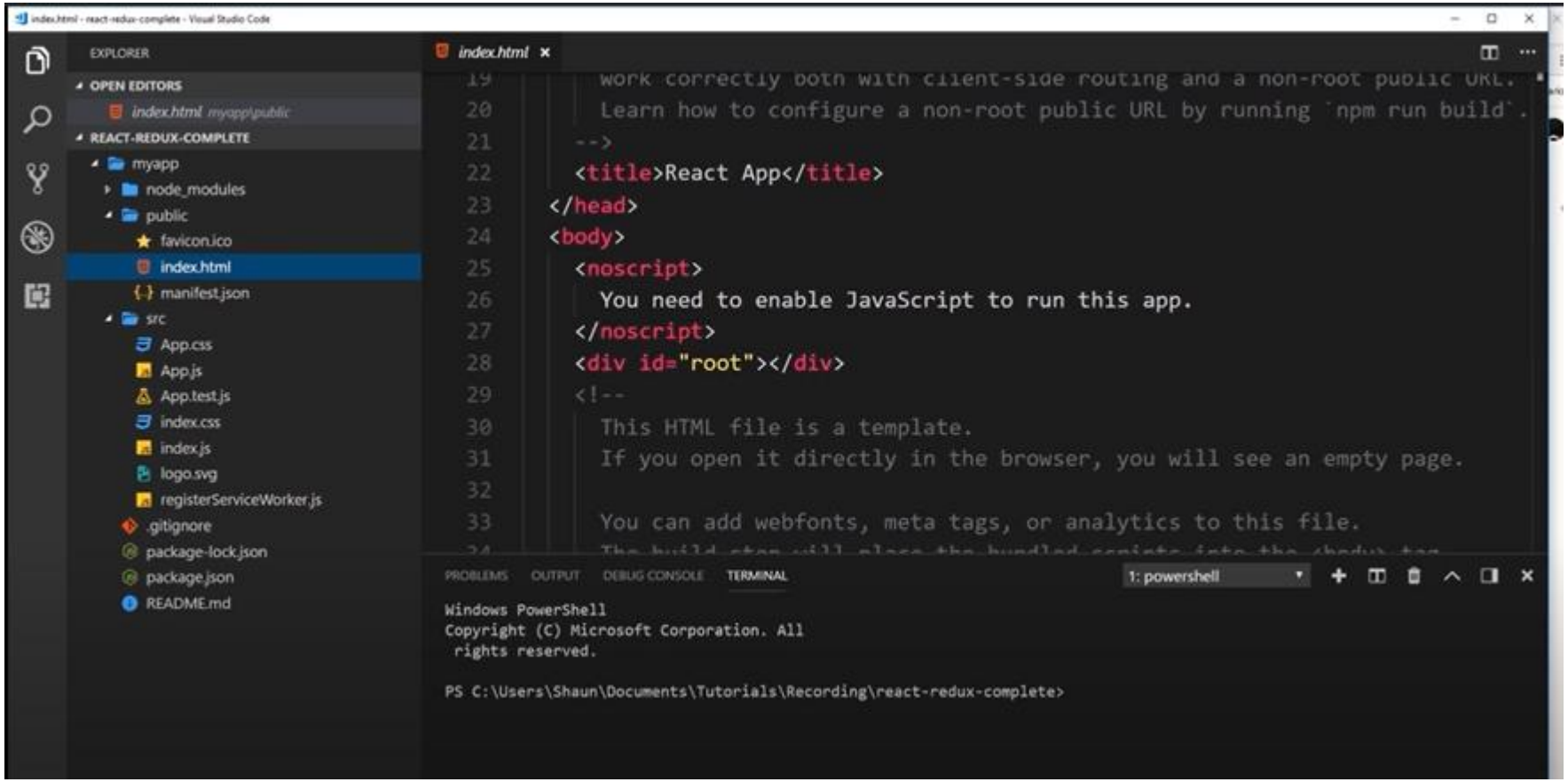



react-redux-complete - Visual Studio Code
EXPLORER
OPEN EDITORS
REACT-REDUX-COMPLETE
myapp
node_modules
public
src
.gitignore
package-lock.json
package.json
README.md
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: powershell
npm test
Starts the test runner.
npm run eject
Removes this tool and copies build dependencies, configuration files
and scripts into the app directory. If you do this, you can't go back!
We suggest that you begin by typing:

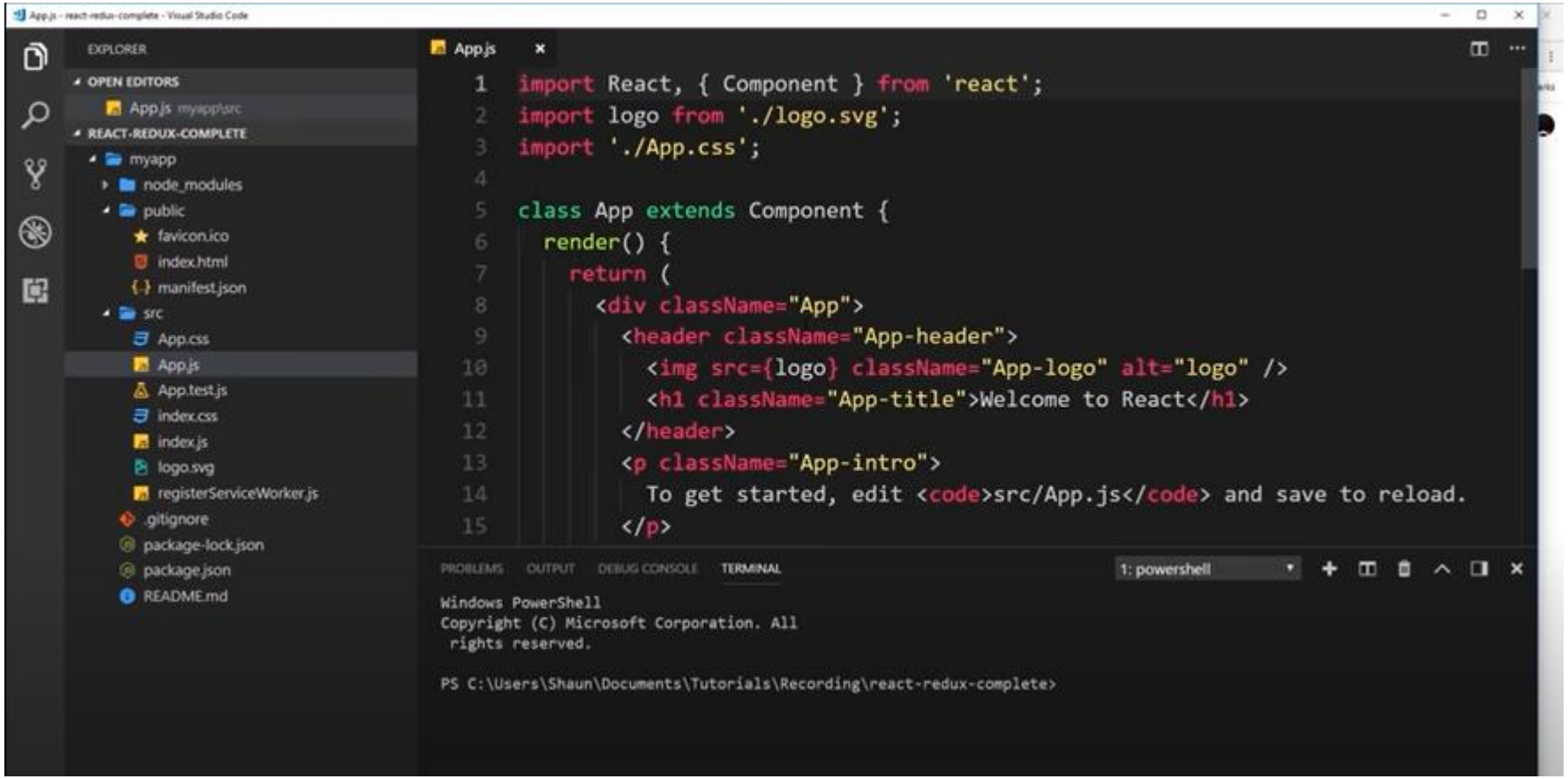

react-redux-complete - Visual Studio Code
EXPLORER
OPEN EDITORS
REACT-REDUX-COMPLETE
myapp
node_modules
public
favicon.ico
index.html
manifest.j
src
.gitignore
package-lock.json
package.json
README.md
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: powershell
npm test
Starts the test runner.
npm run eject
Removes this tool and copies build dependencies, configuration files
and scripts into the app directory. If you do this, you can't go back!
We suggest that you begin by typing:


package.json - react-redux-complete - Visual Studio Code
EXPLORER
OPEN EDITORS
package.json myapp
REACT-REDUX-COMPLETE
myapp
node_modules
public
src
.gitignore
package-lock.json
package.json
README.md
package.json
"react-dom": "^16.4.1",
"react-scripts": "1.1.4"
},
"scripts": {
"start": "react-scripts start",
"build": "react-scripts build",
"test": "react-scripts test --env=jsdom",
"eject": "react-scripts eject"
}
}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: powershell
npm test
Starts the test runner.
npm run eject
Removes this tool and copies build dependencies, configuration files
and scripts into the app directory. If you do this, you can't go back!
We suggest that you begin by typing:

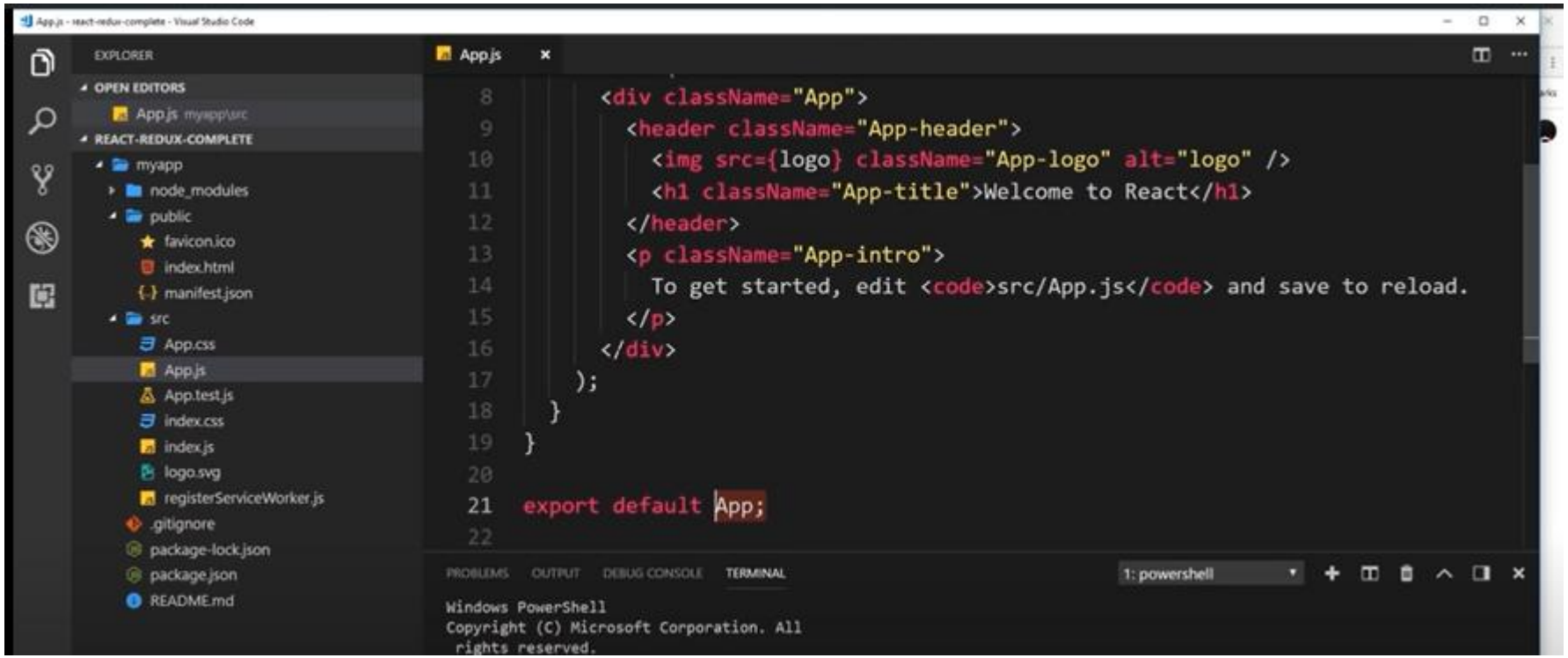

react-redux-complete - Visual Studio Code
EXPLORER
OPEN EDITORS
REACT-REDUX-COMPLETE
myapp
node_modules
public
src
App.css
App.js
App.test.js
index.css
index.js
logo.svg
registerServiceWorker.js
.gitignore
package-lock.json
package.json
README.md
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: powershell
npm test
Starts the test runner.
npm run eject
Removes this tool and copies build dependencies, configuration files
and scripts into the app directory. If you do this, you can't go back!
We suggest that you begin by typing:



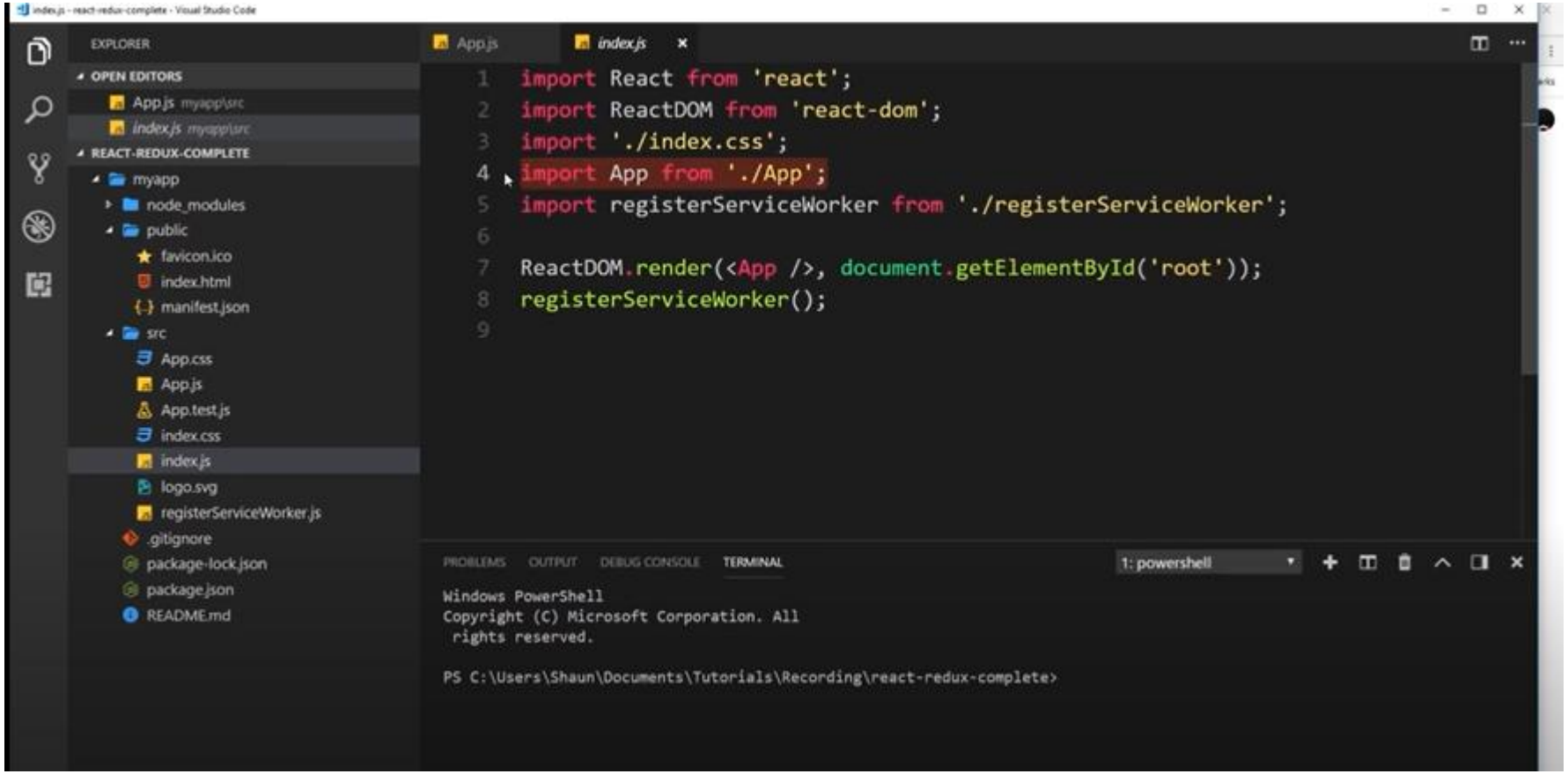
Basically this index.html is our one page app. All the components will be injected dynamically in this root div



First component React created for us with name App.
Its same which we saw earlier like a class based component with render method having a jsx template code, returning a div.

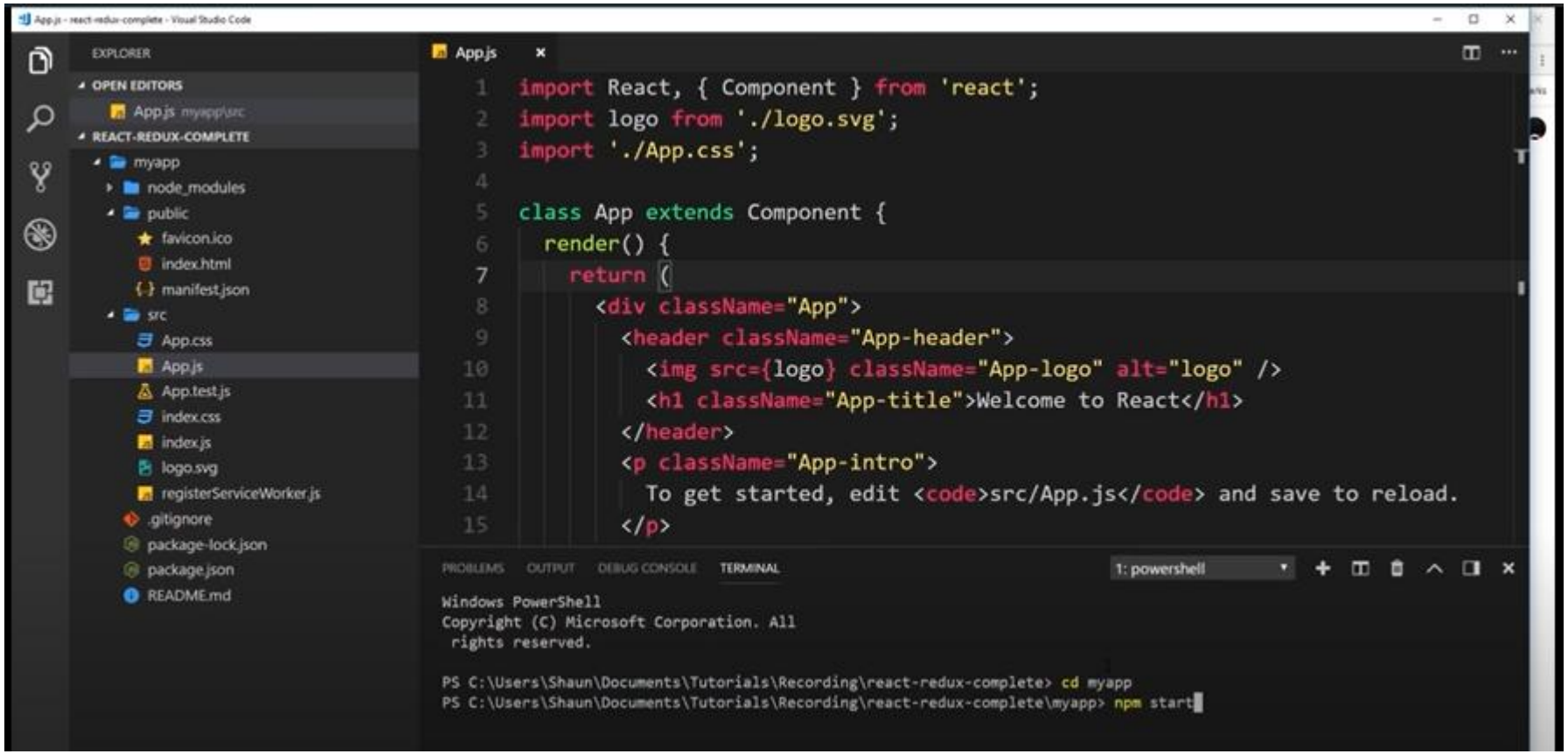


At the end we are exporting that component so that we can render it to the DOM.

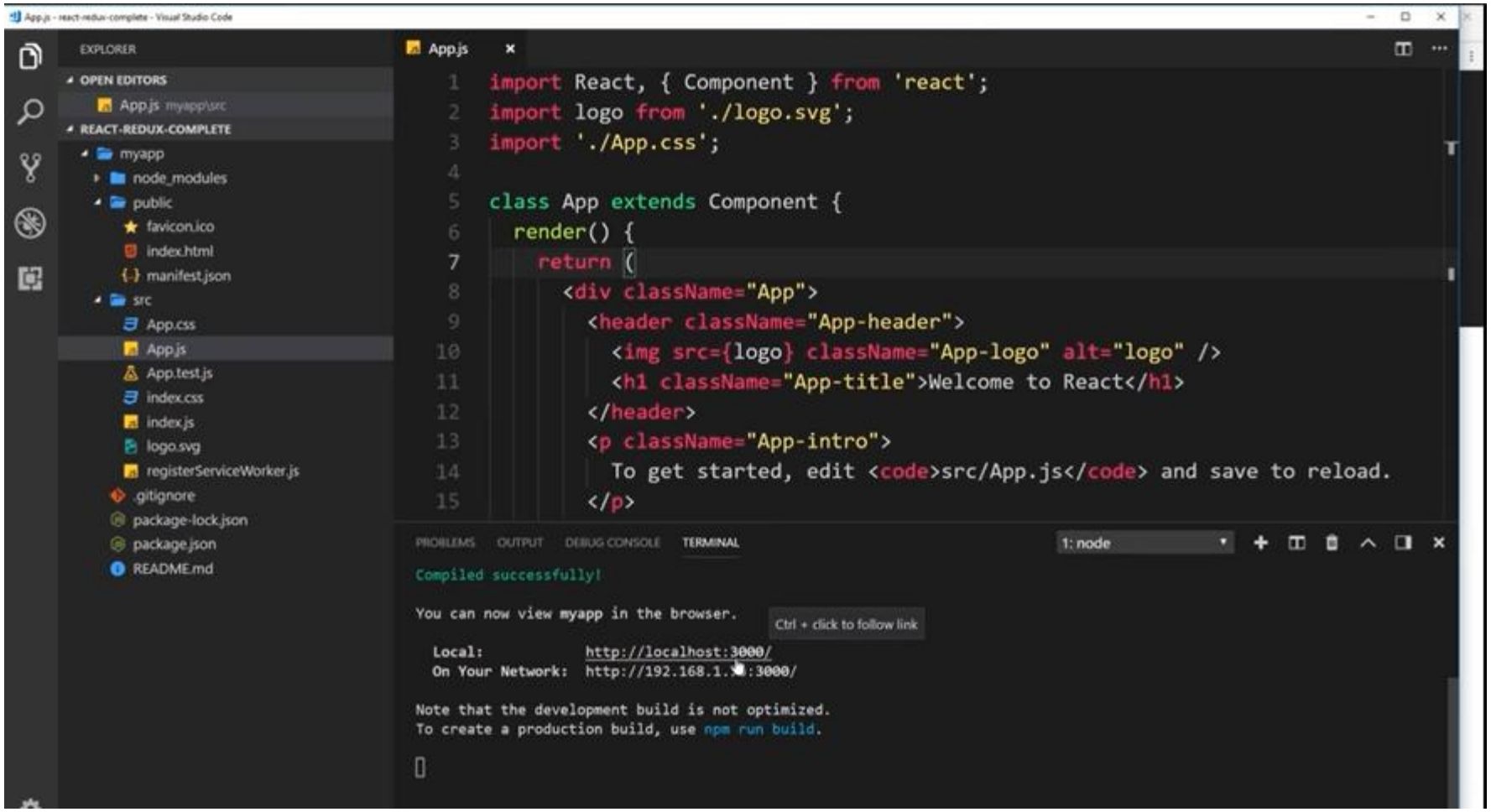


Importing that component in index.js and rendering it to the root element of index.html
So basically here we are inject our component with ReactDOM.render method

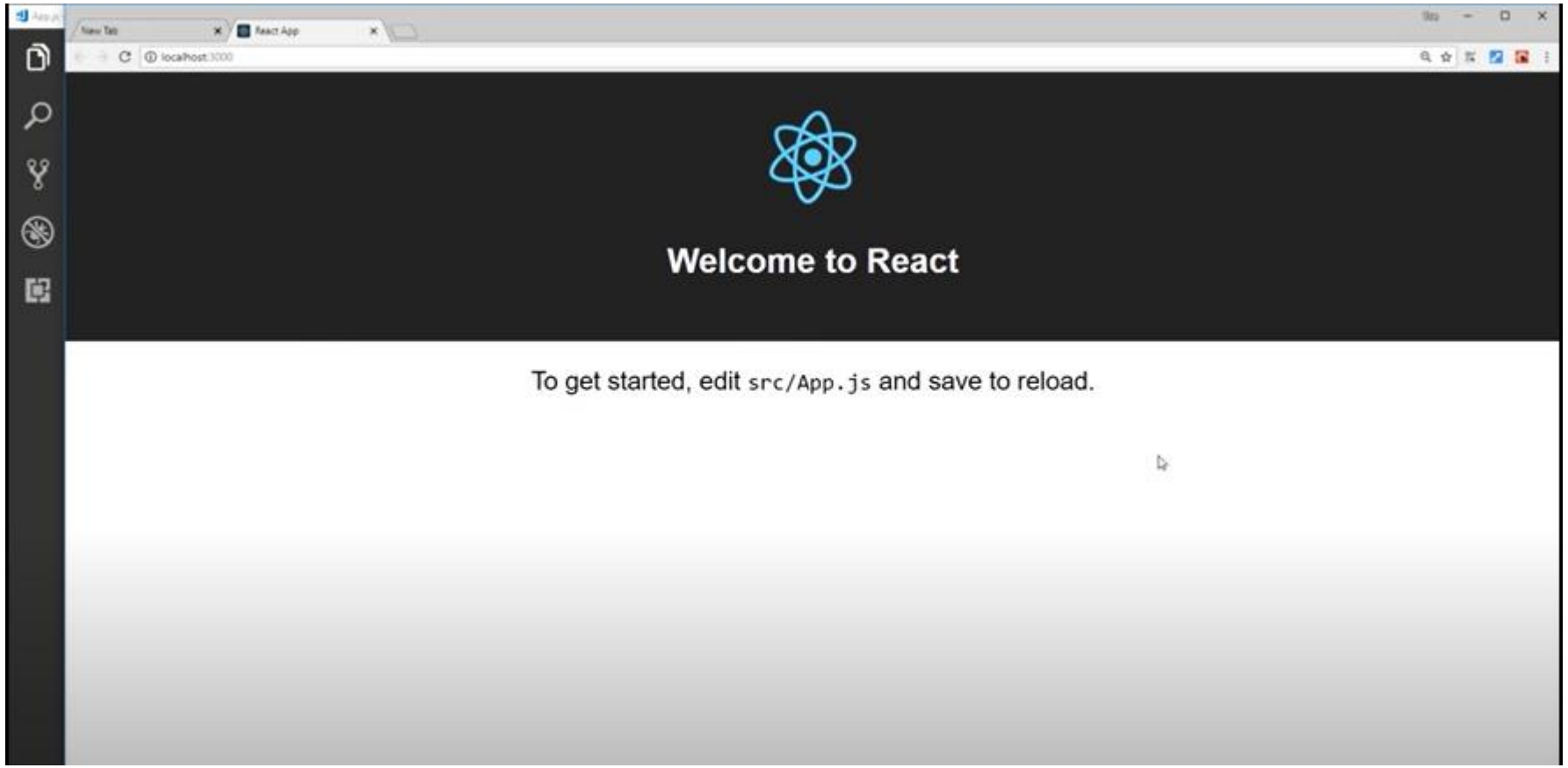
How to run this app



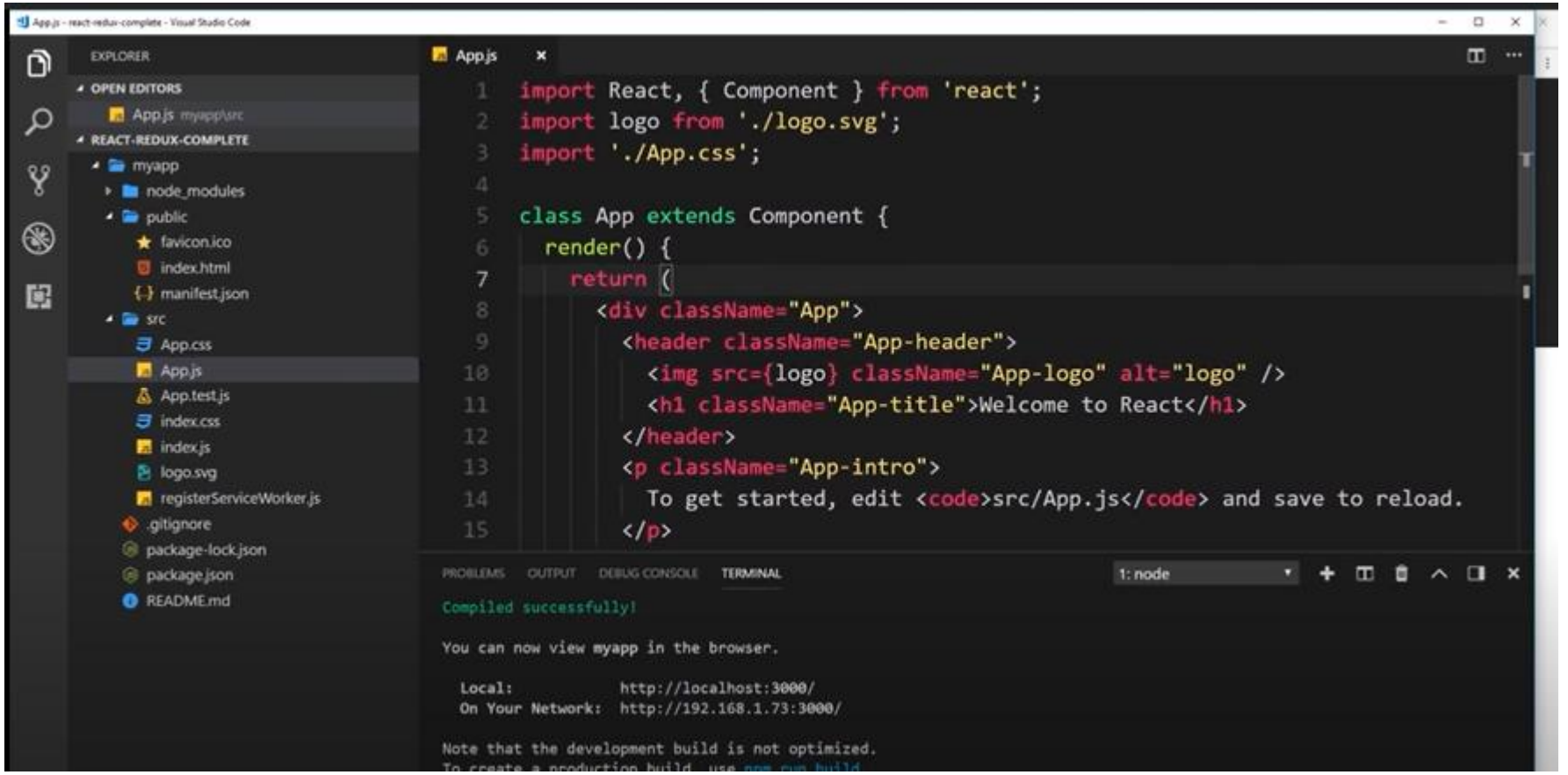
Firstly with cd command will go into the directory
Then will start a local server



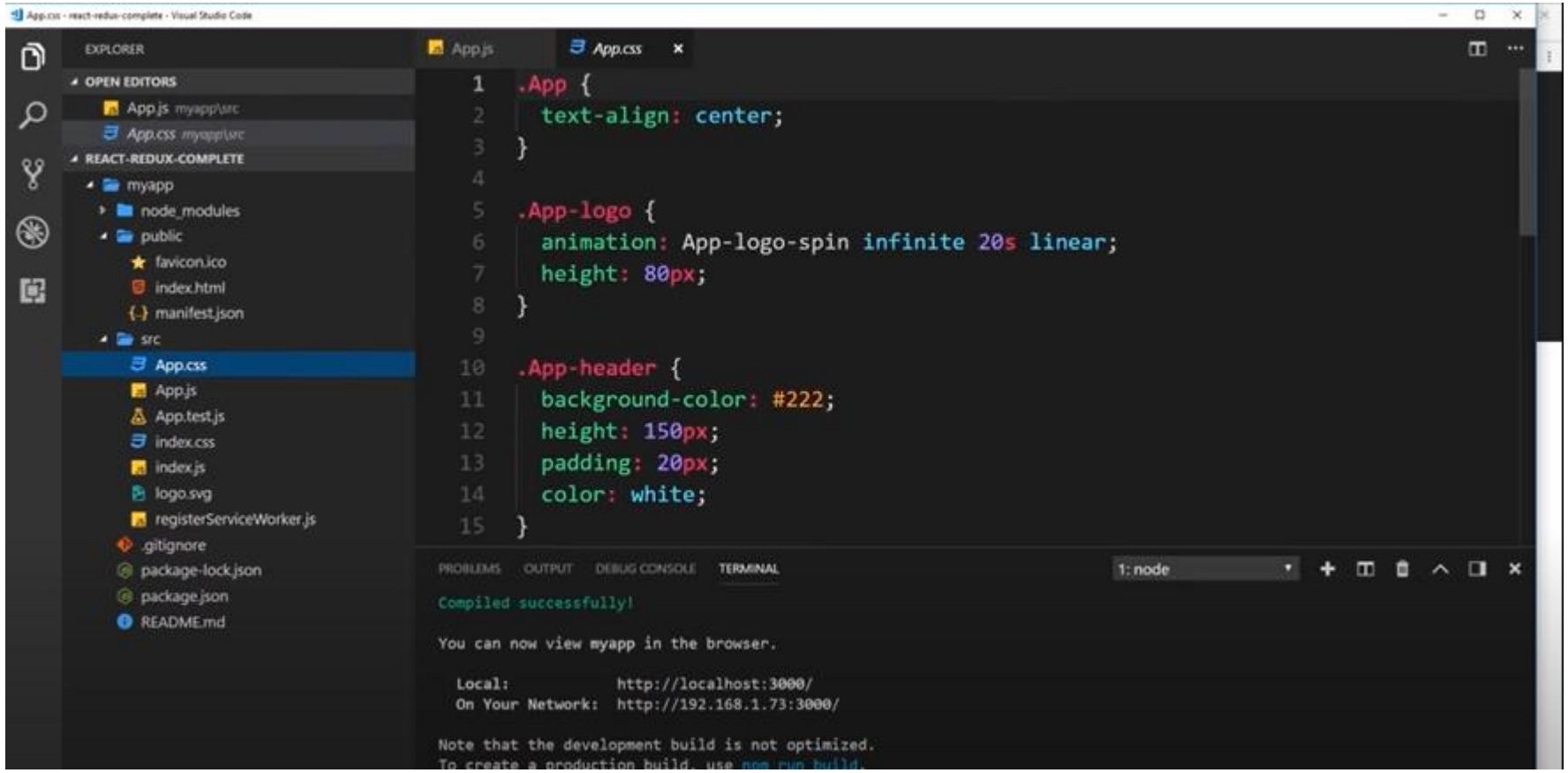
It will shows the ip address where that app is running



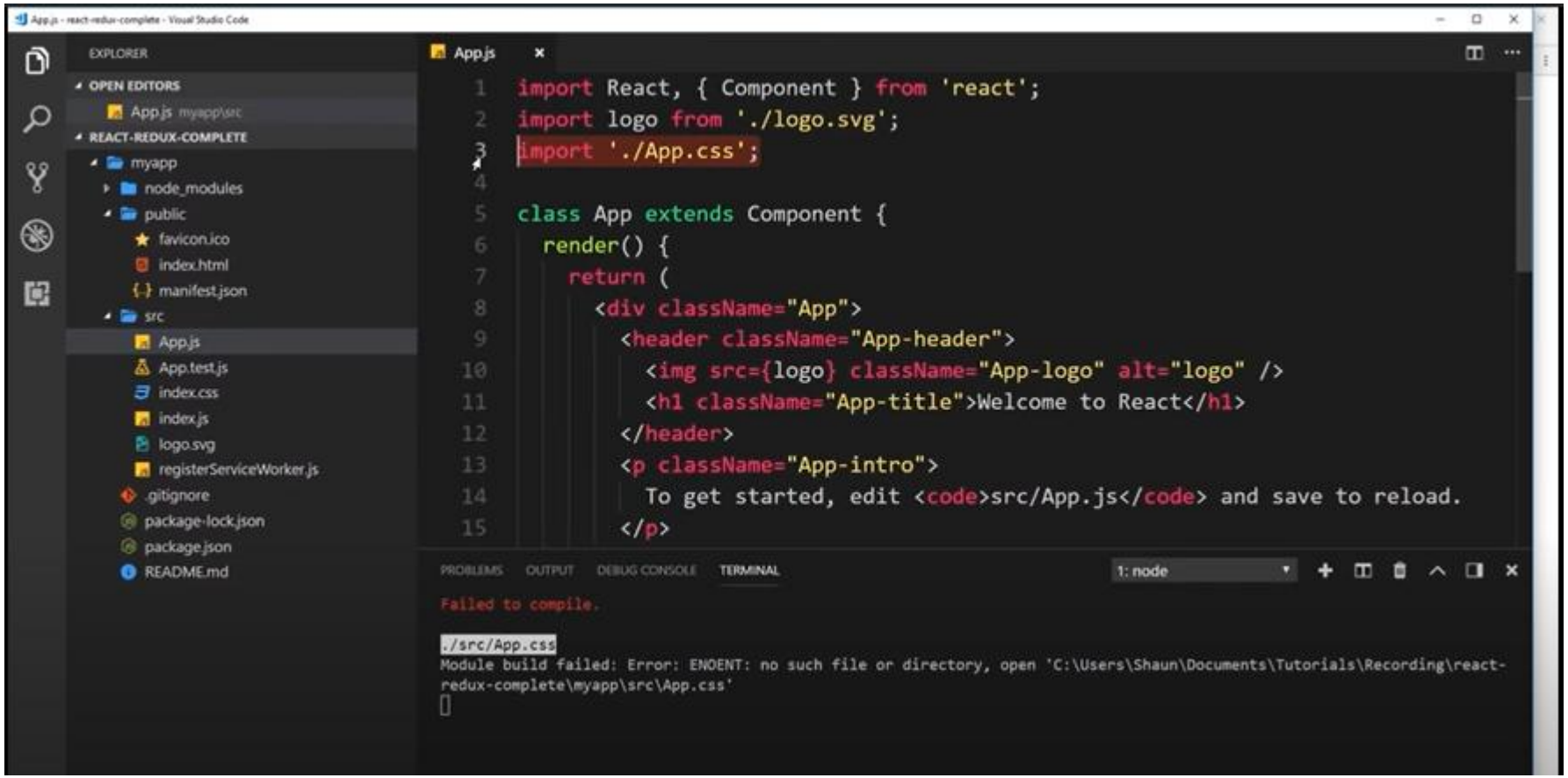
It will also be opened in browser automatically



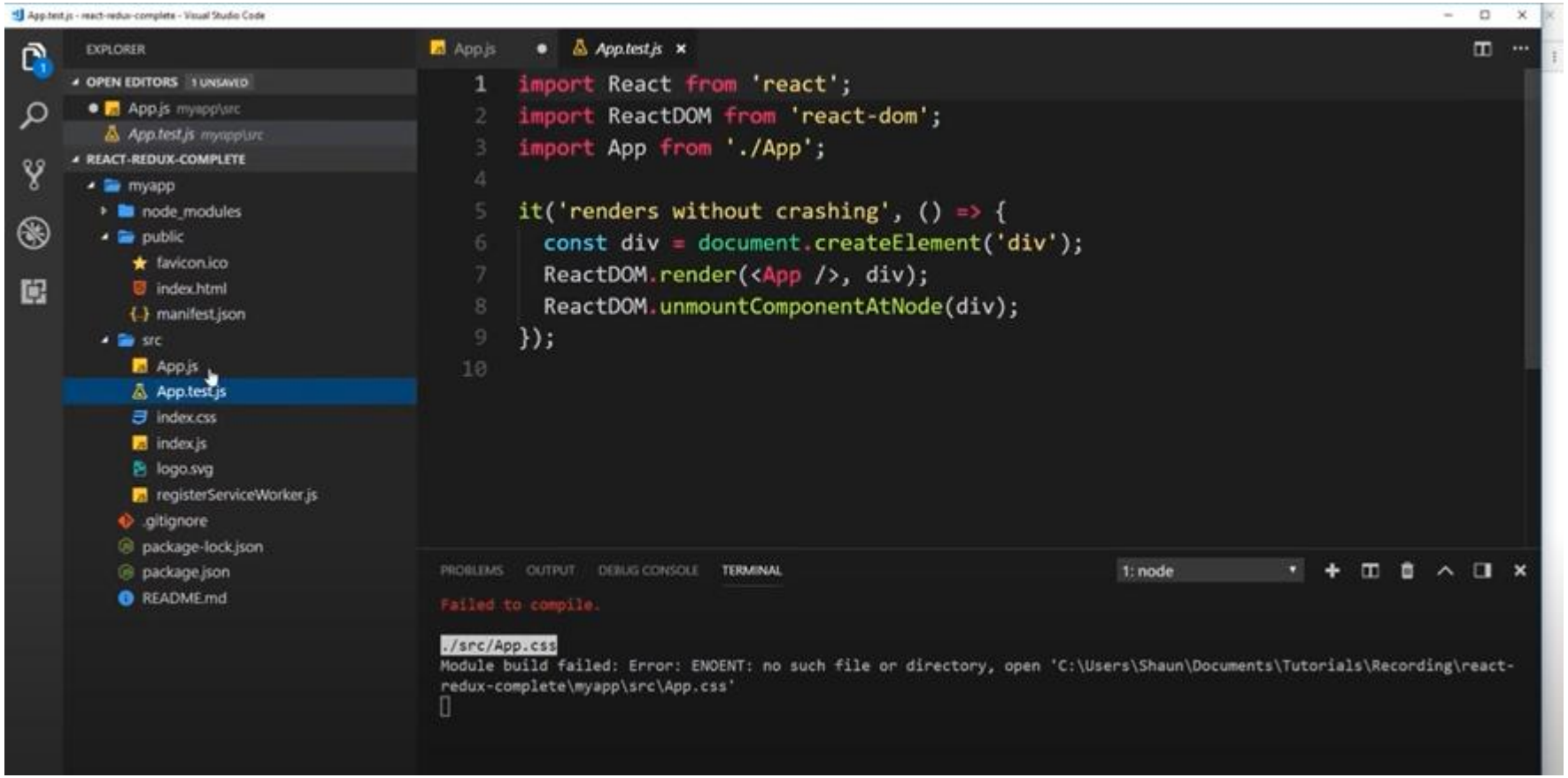
All the text on the browser is coming from here: App component



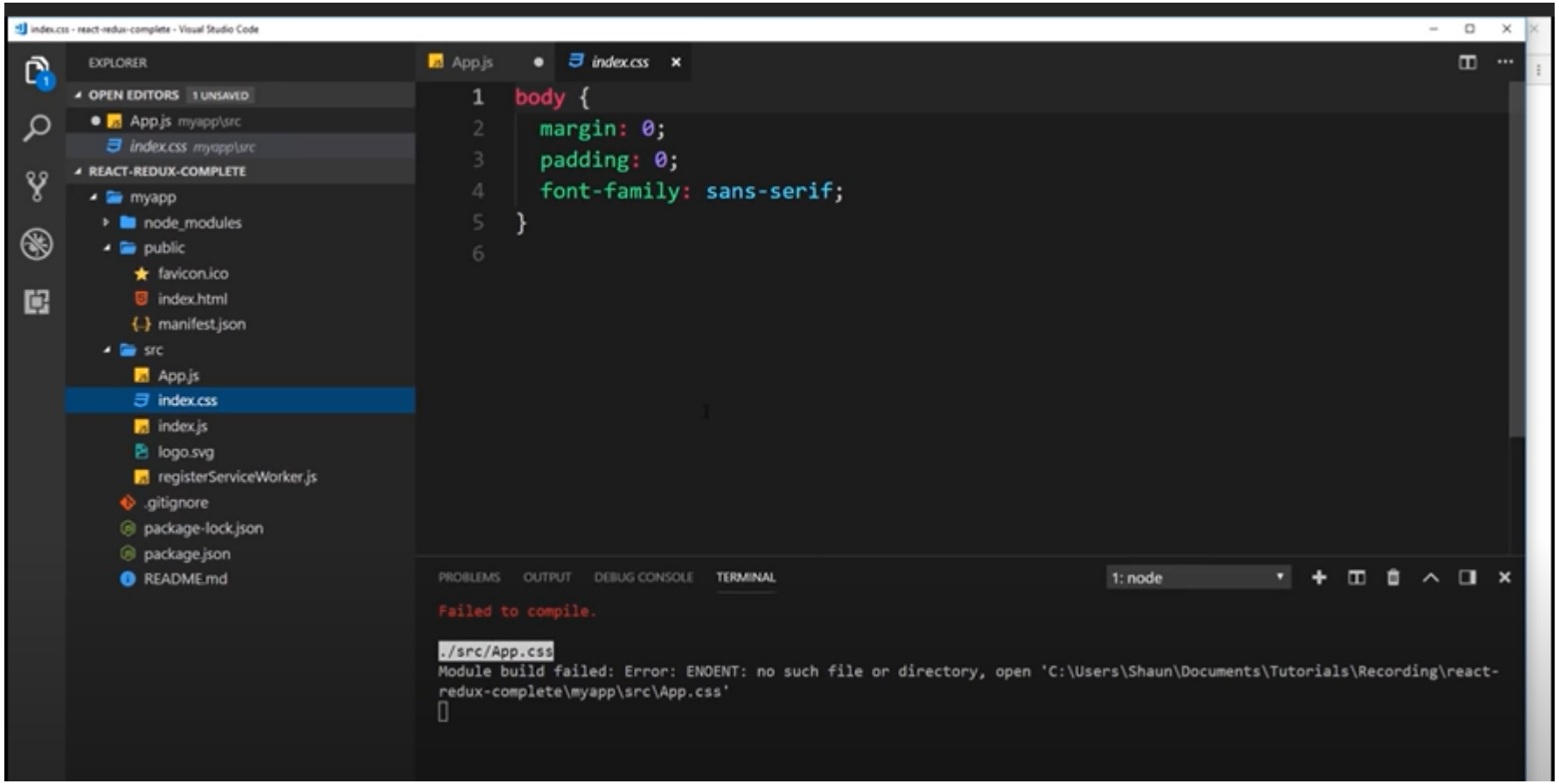
Lets delete the css file. We do not going to show that default text on browser like its logo



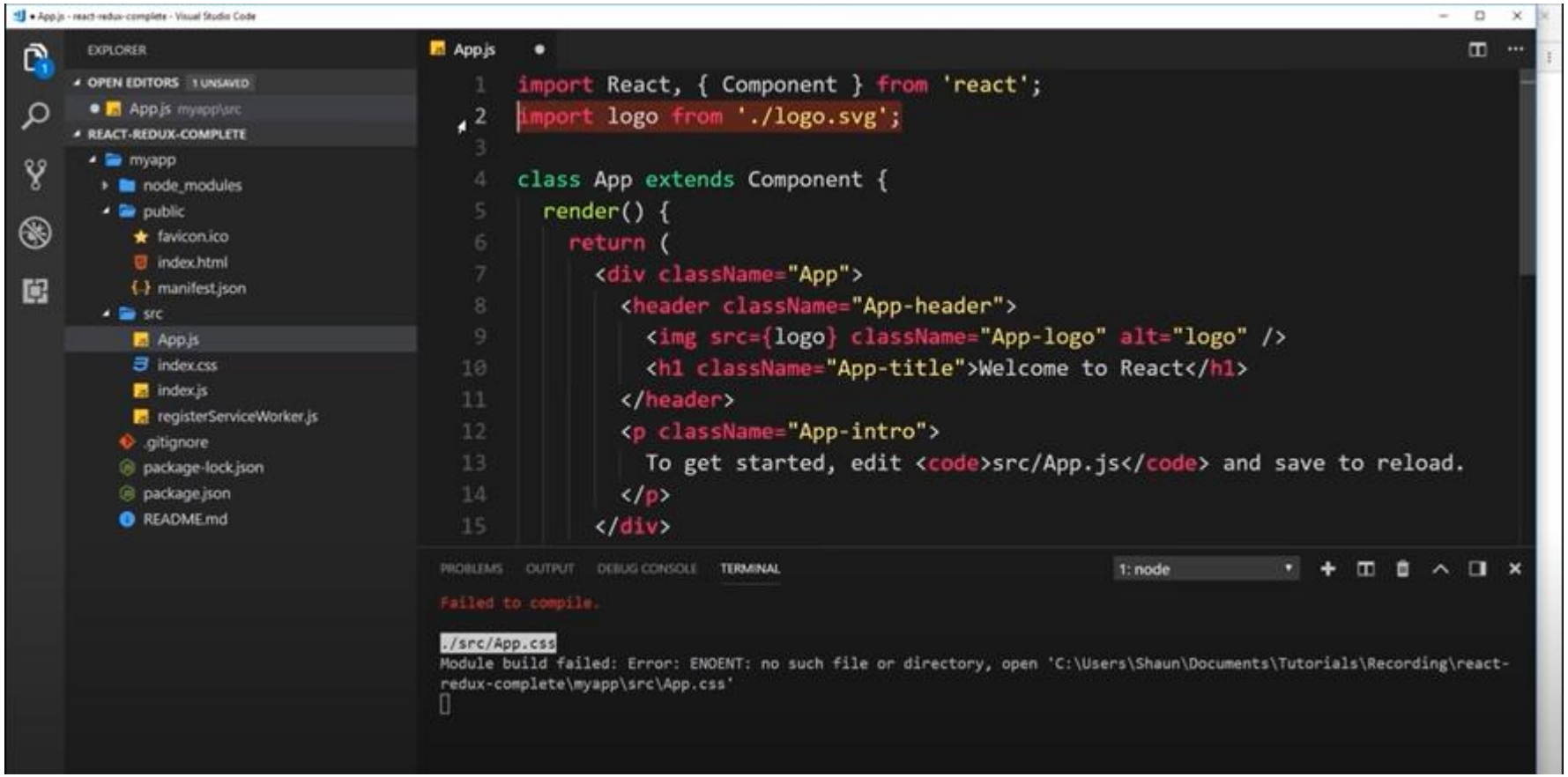
Also delete the reference of the previous css file



Also delete the test file. No need at this time

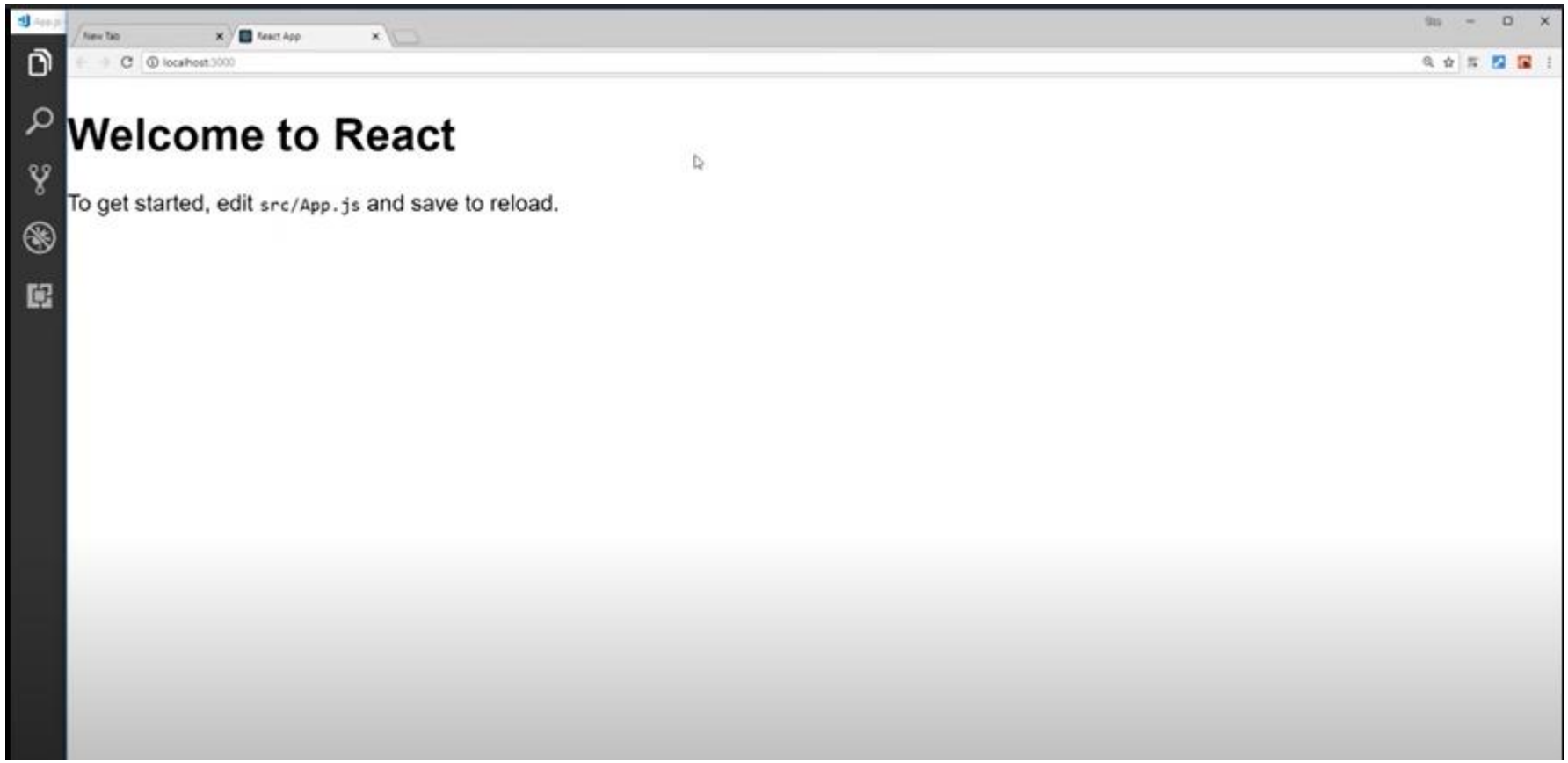


So now basically we have
App.js file which is our component.
Index.css which is a global css file of our project we can also add more css in this file.
Index.js file where we render the component to the DOM.
Logo.svg delete this logo too we have no further need.
registerServiceWorker.js file which manage cache for best user experience leave it…



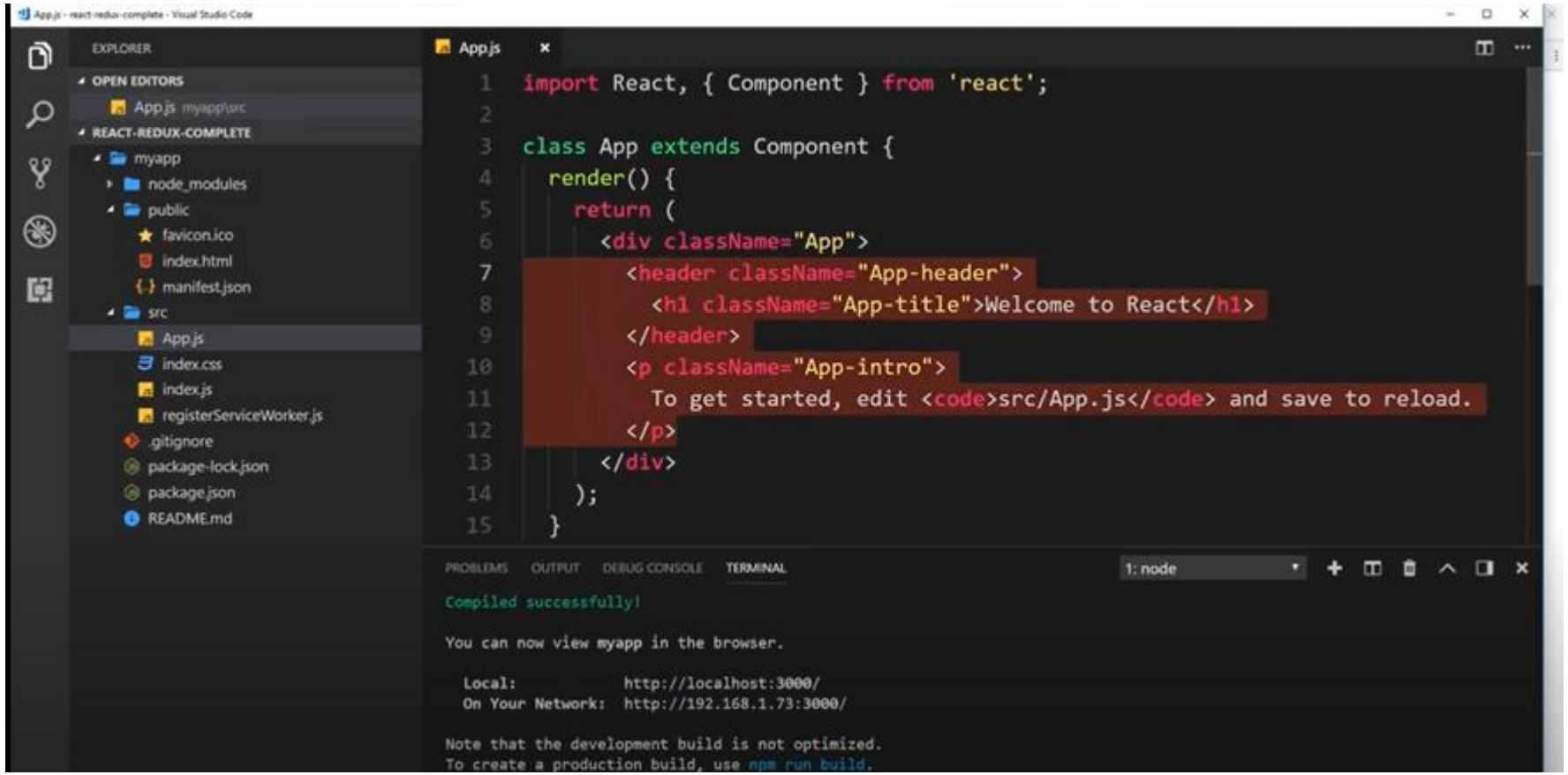
Delete the import logo line we have no more this logo and also delete the image tag where we are putting logo.

Now in the browser we will have:

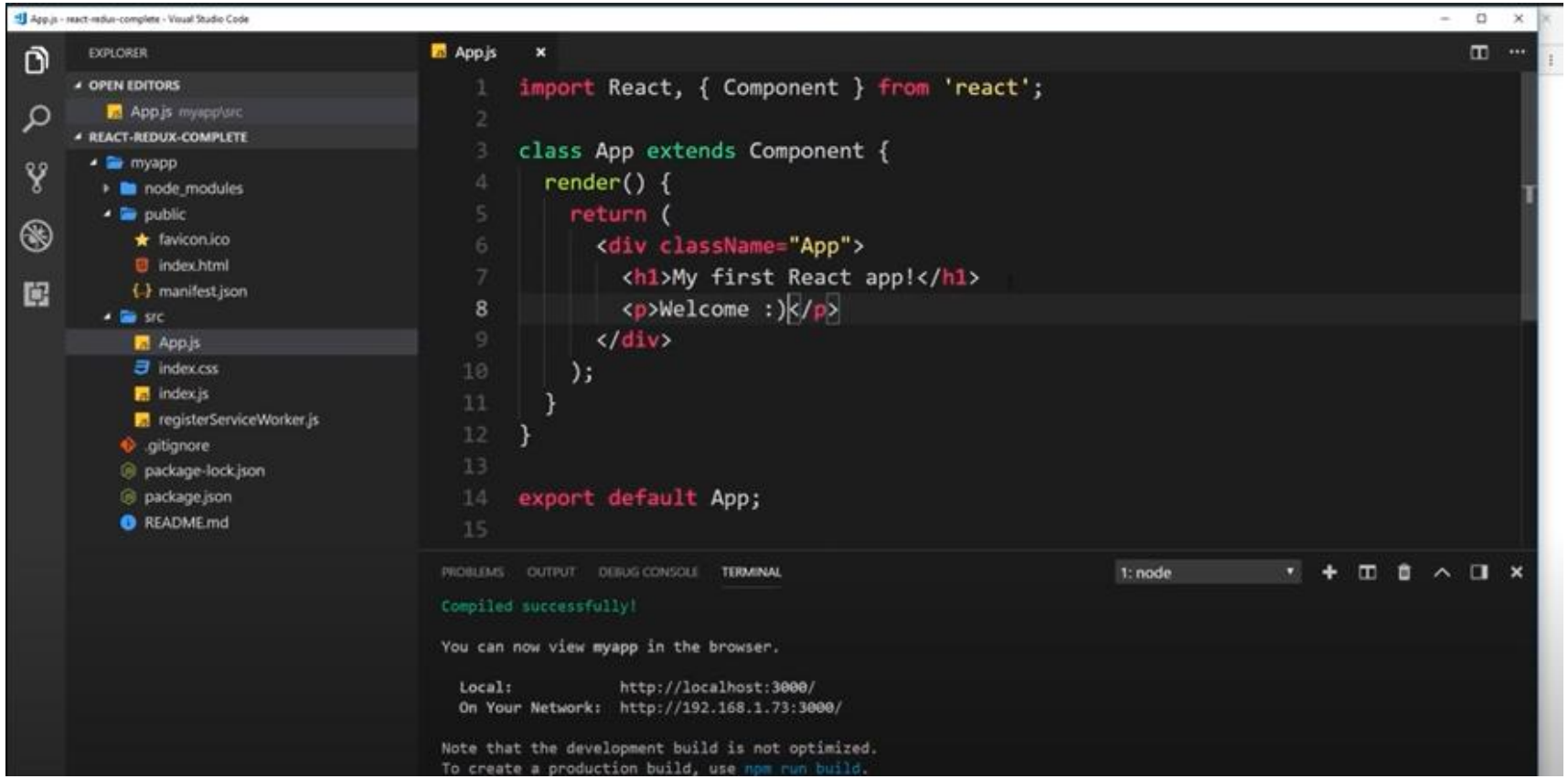


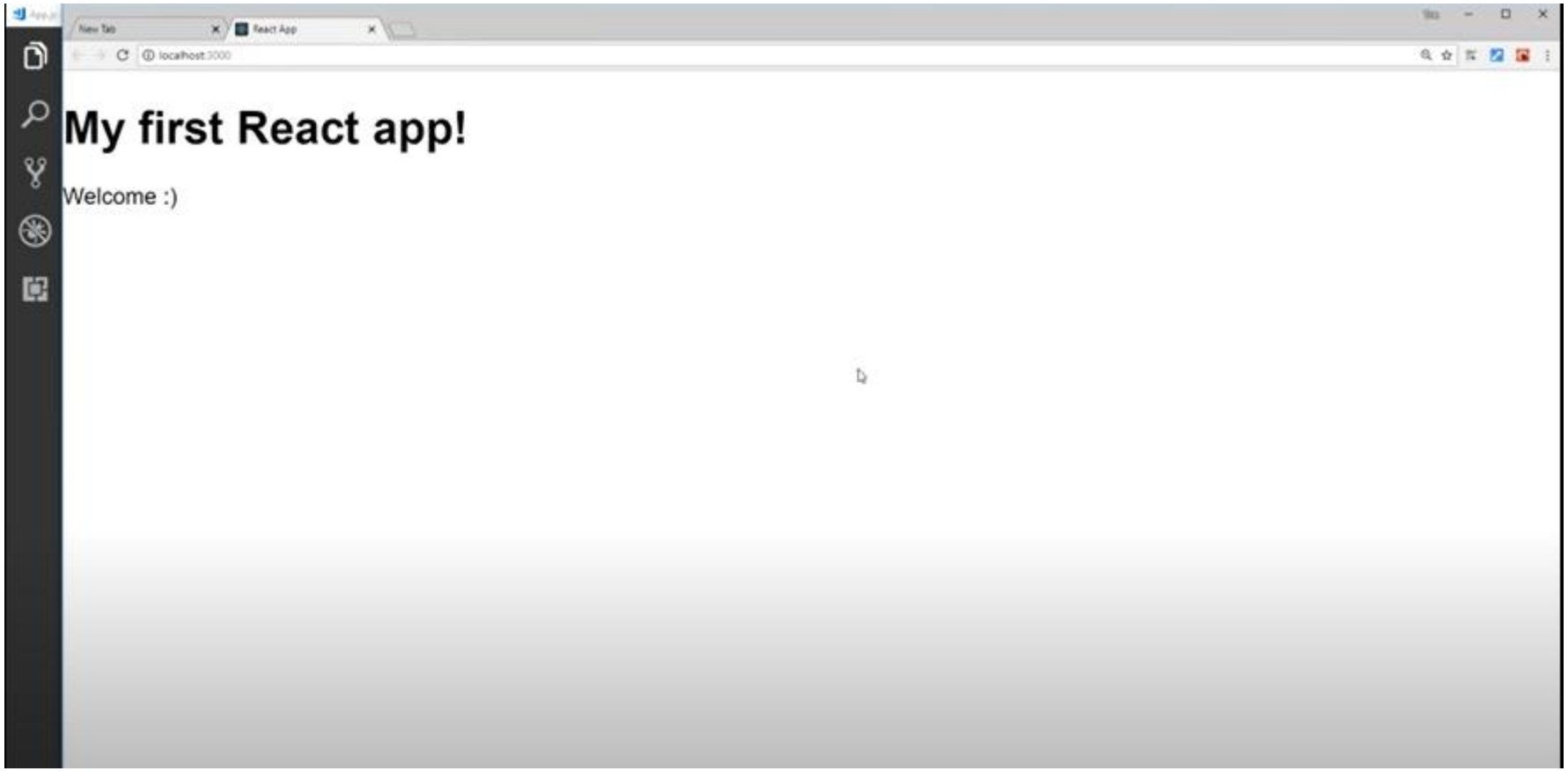
Simple Right?

Delete the rest too



Add simple template



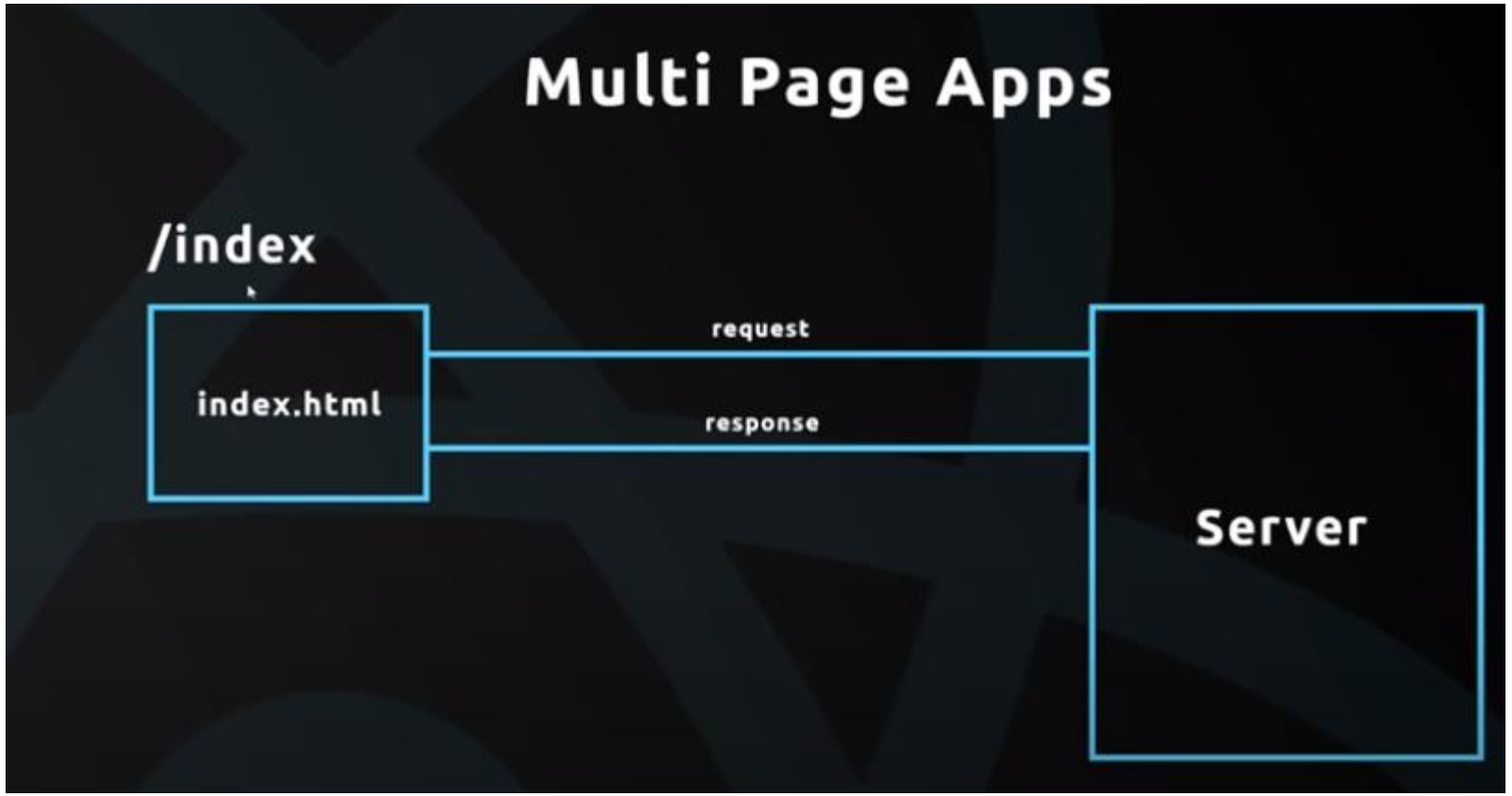


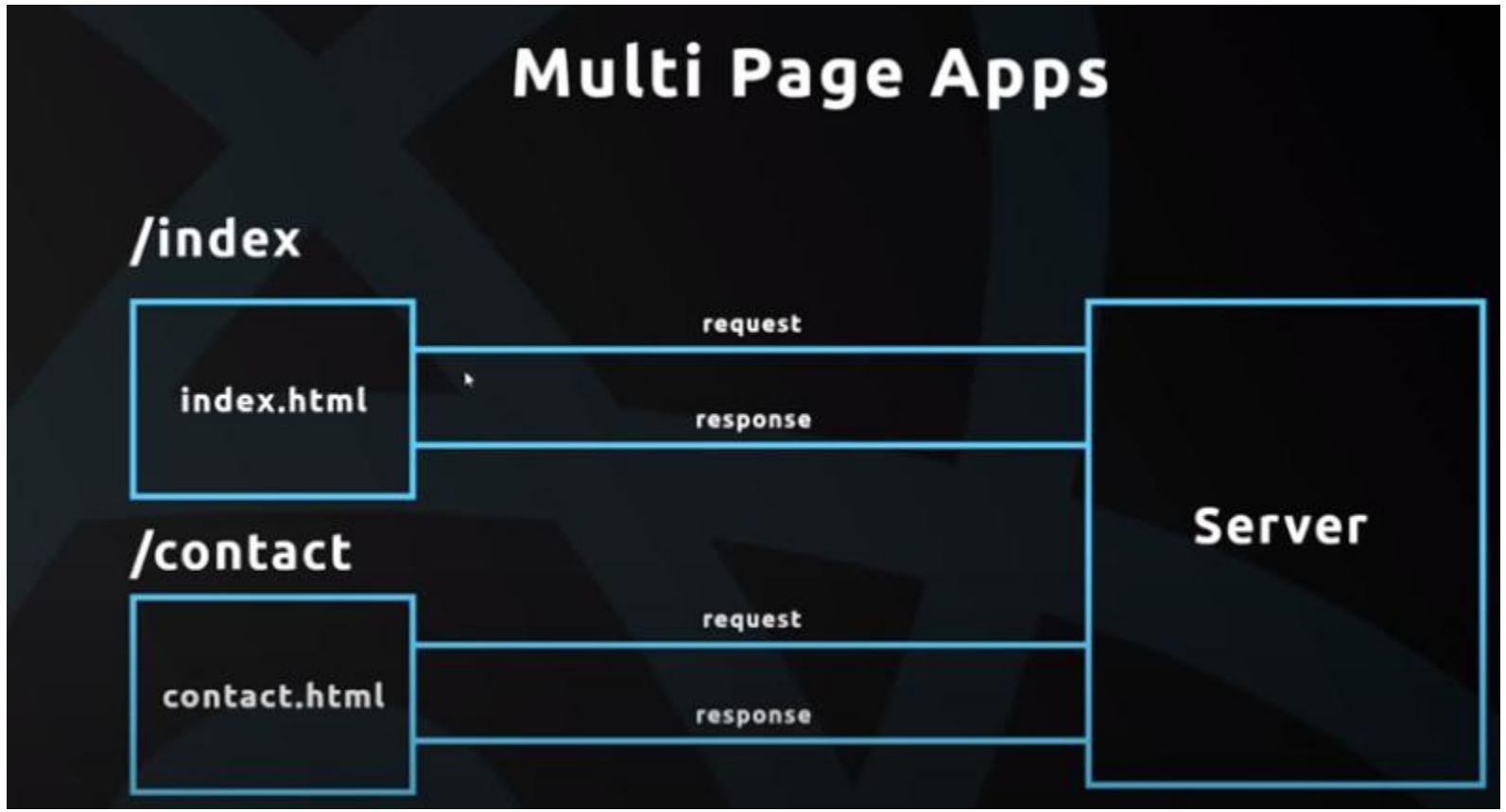
When you hit Ctrl +S key it will automatically updated on the browser

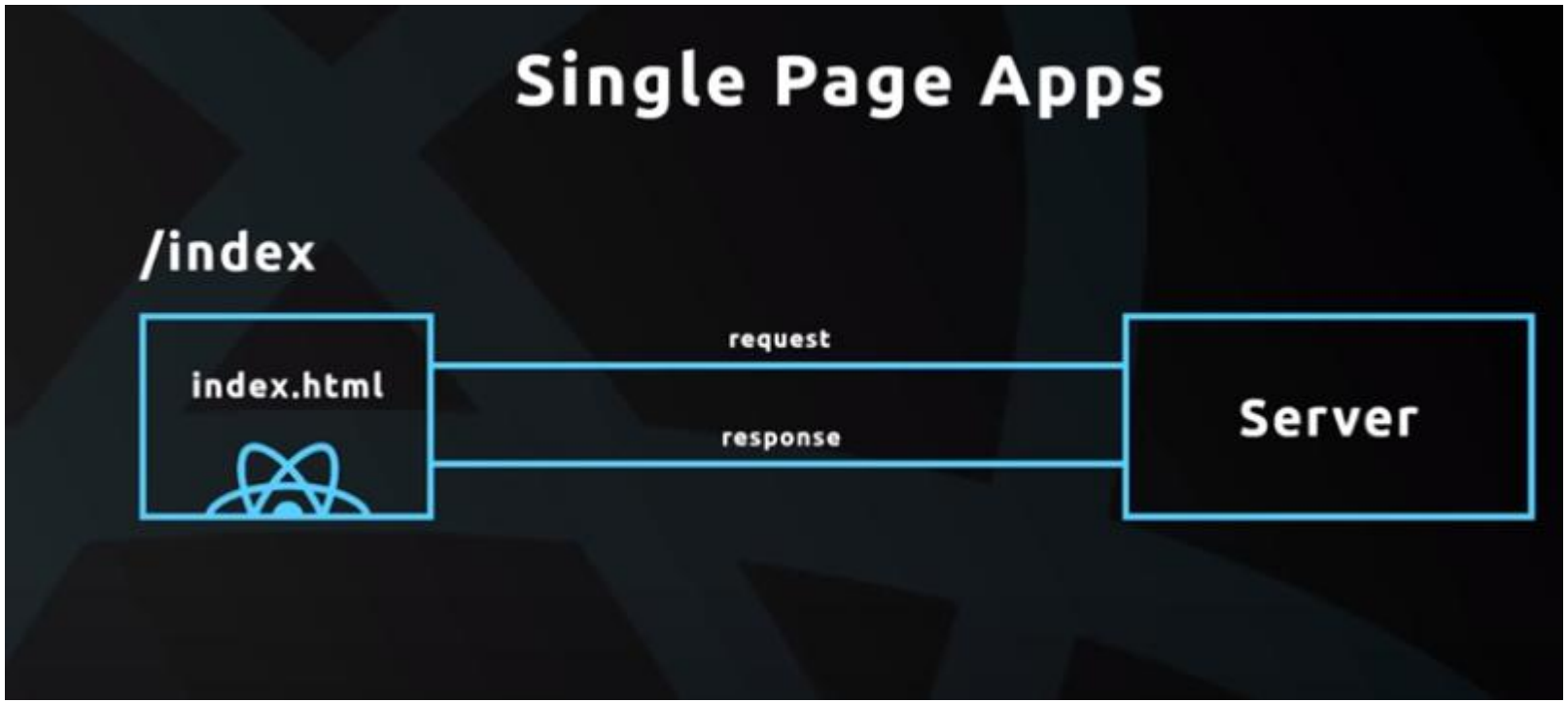
SINGLE PAGE APPS

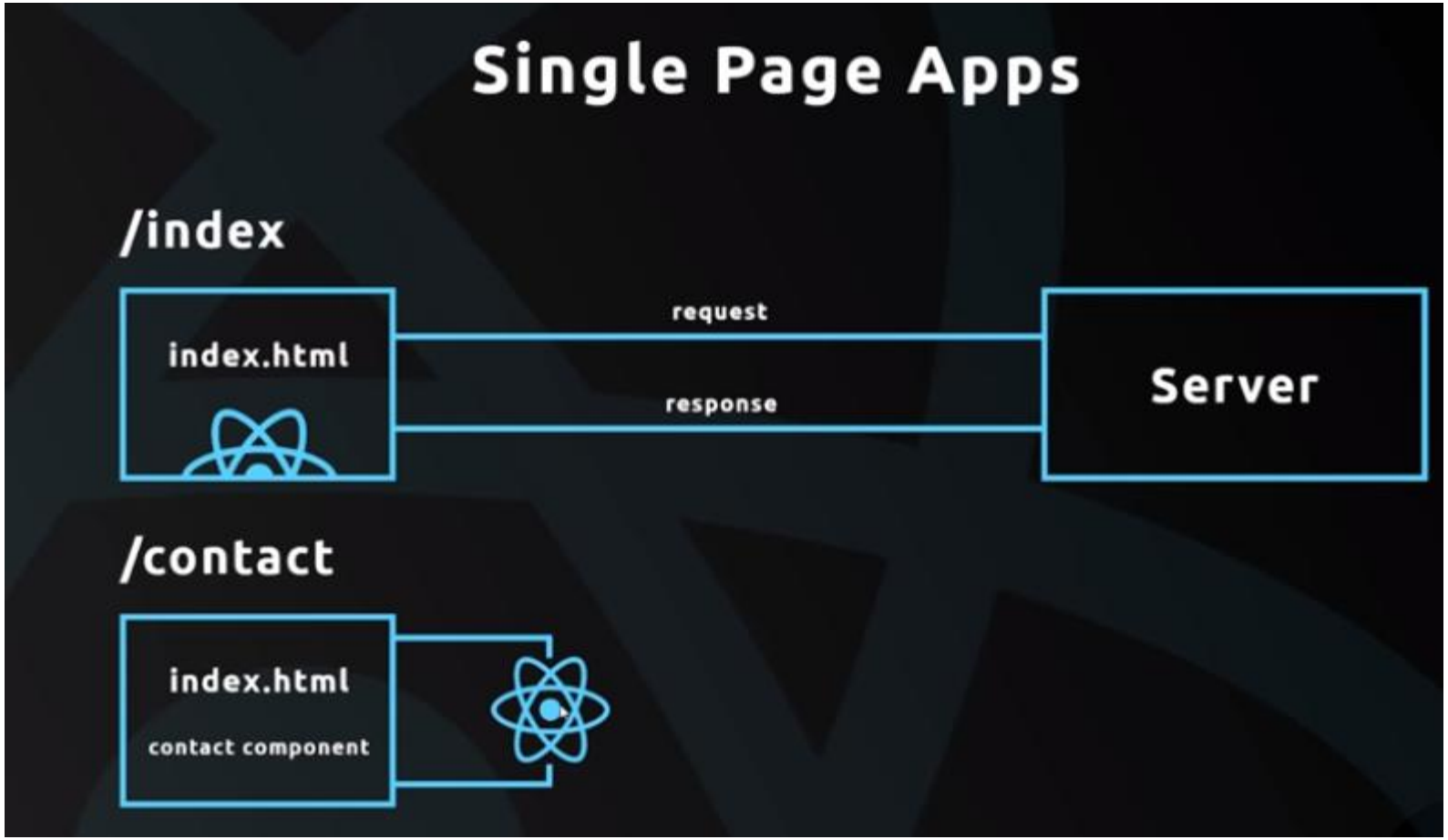
Single Page Apps

- React apps are typically SPA's
- Only ever one HTML page served to the browser
- React then controls what a user sees on that page


Multi Page Apps
/index
index.html
request
response
Server

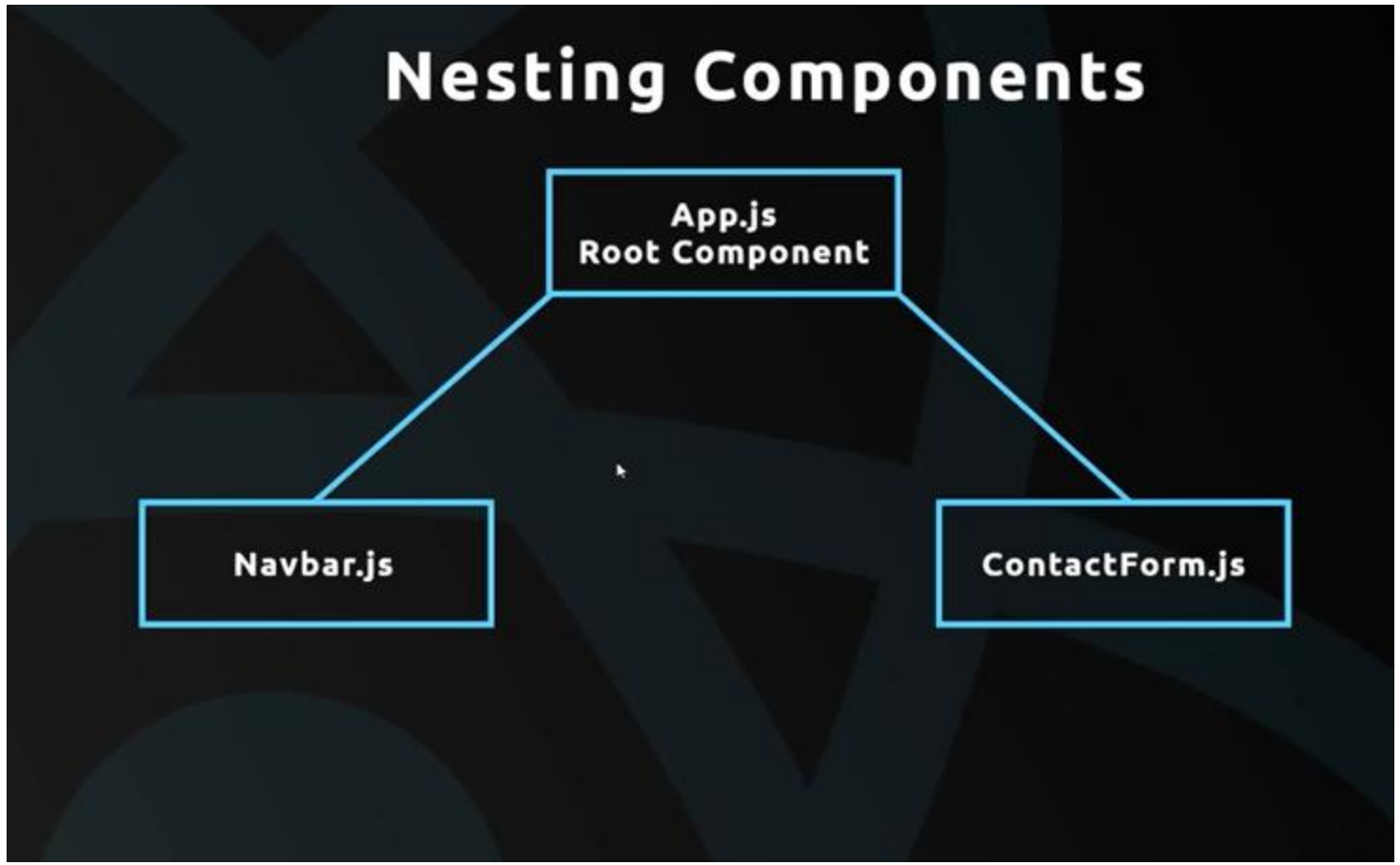

Multi Page Apps
/index
index.html
request
response
/contact
contact.html
request
response
Server

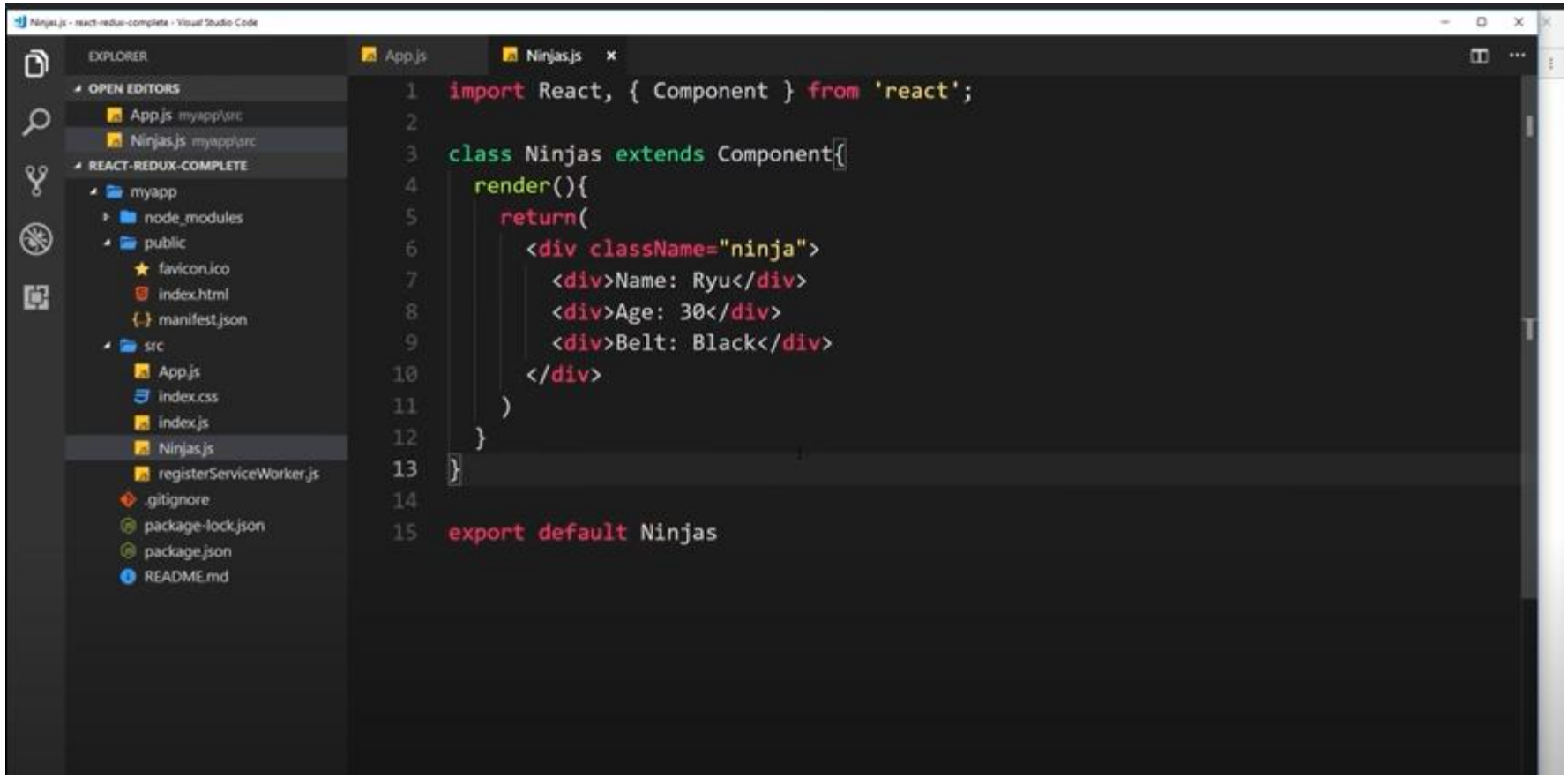

Single Page Apps
/index
index.html
request
response
Server


Single Page Apps
/index
index.html
request
response
Server
/contact
index.html
contact component

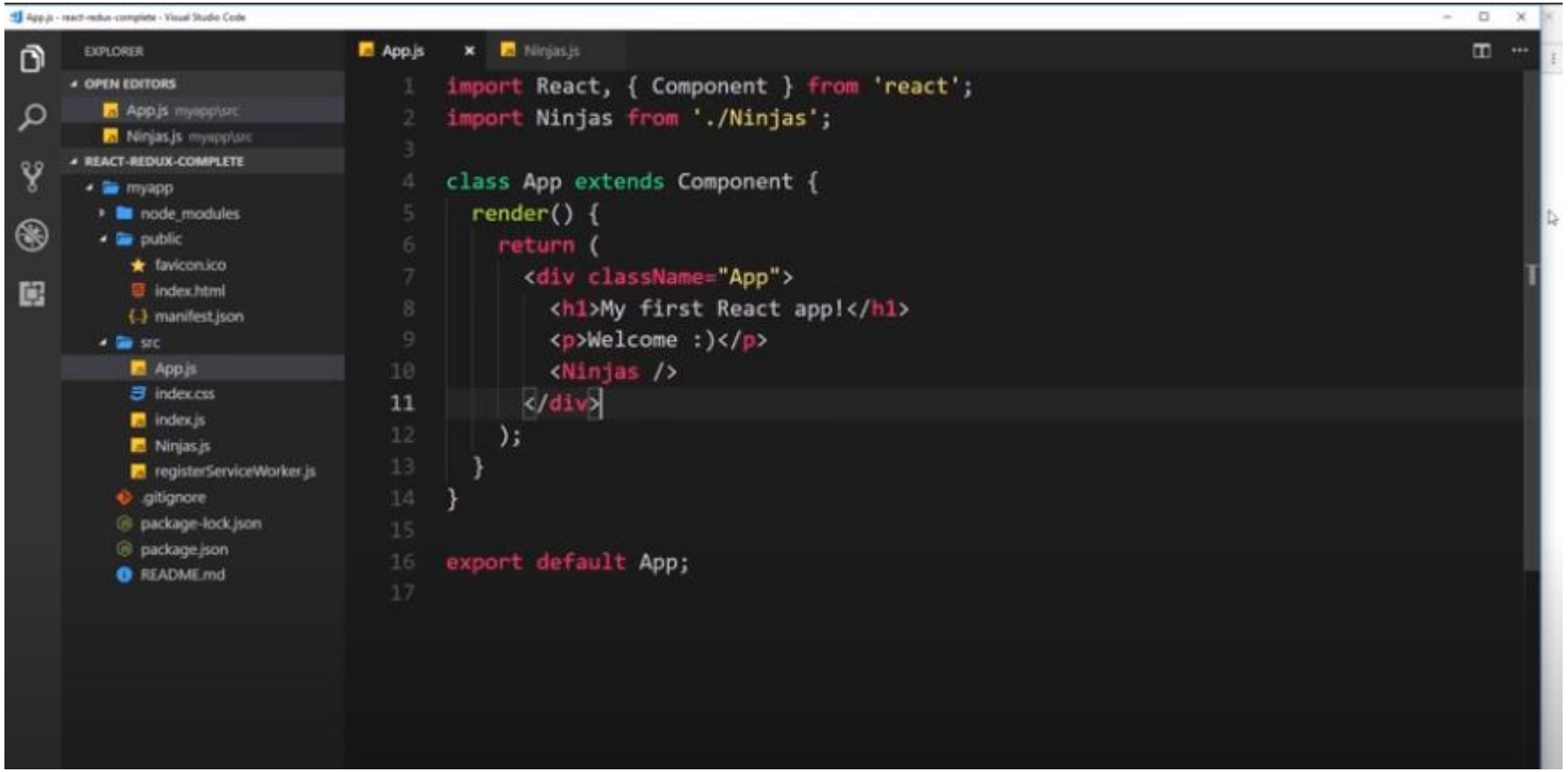
NESTING COMPONENTS

In React root component is App.js we can nesting more components in it

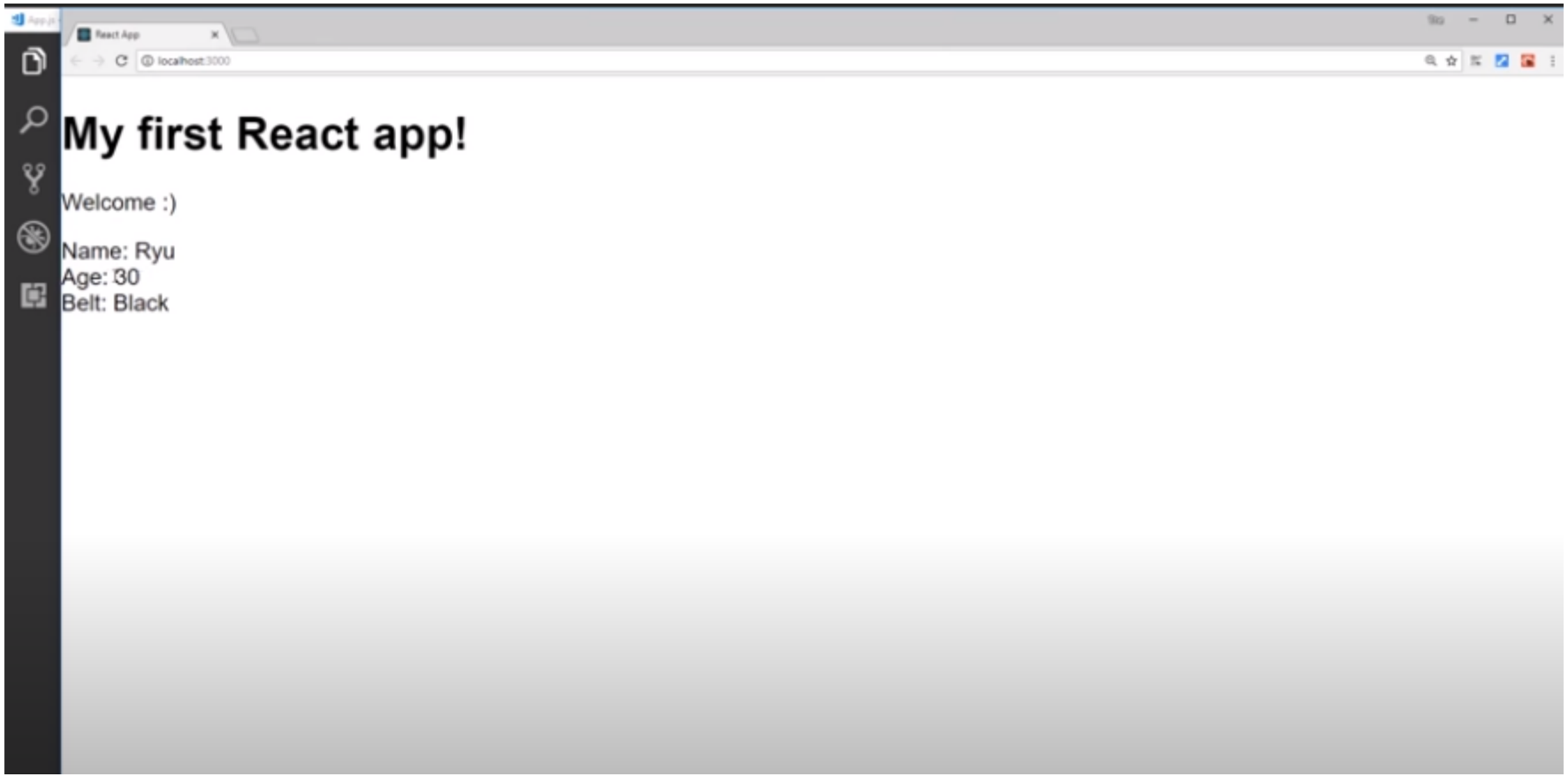

Nesting Components
App.js
Root Component
Navbar.js
ContactForm.js



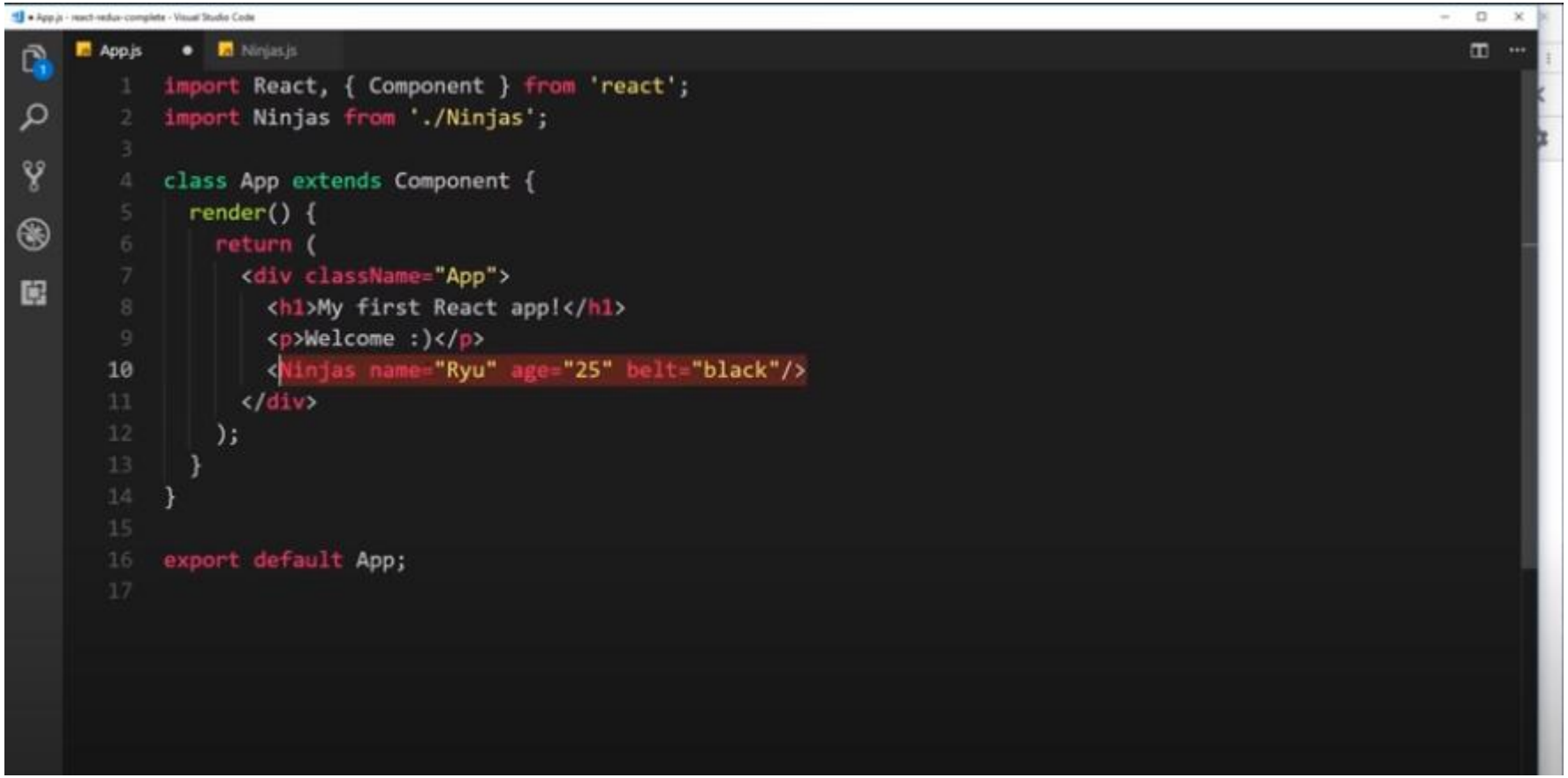
Creating another component with name Ninjas in src directory



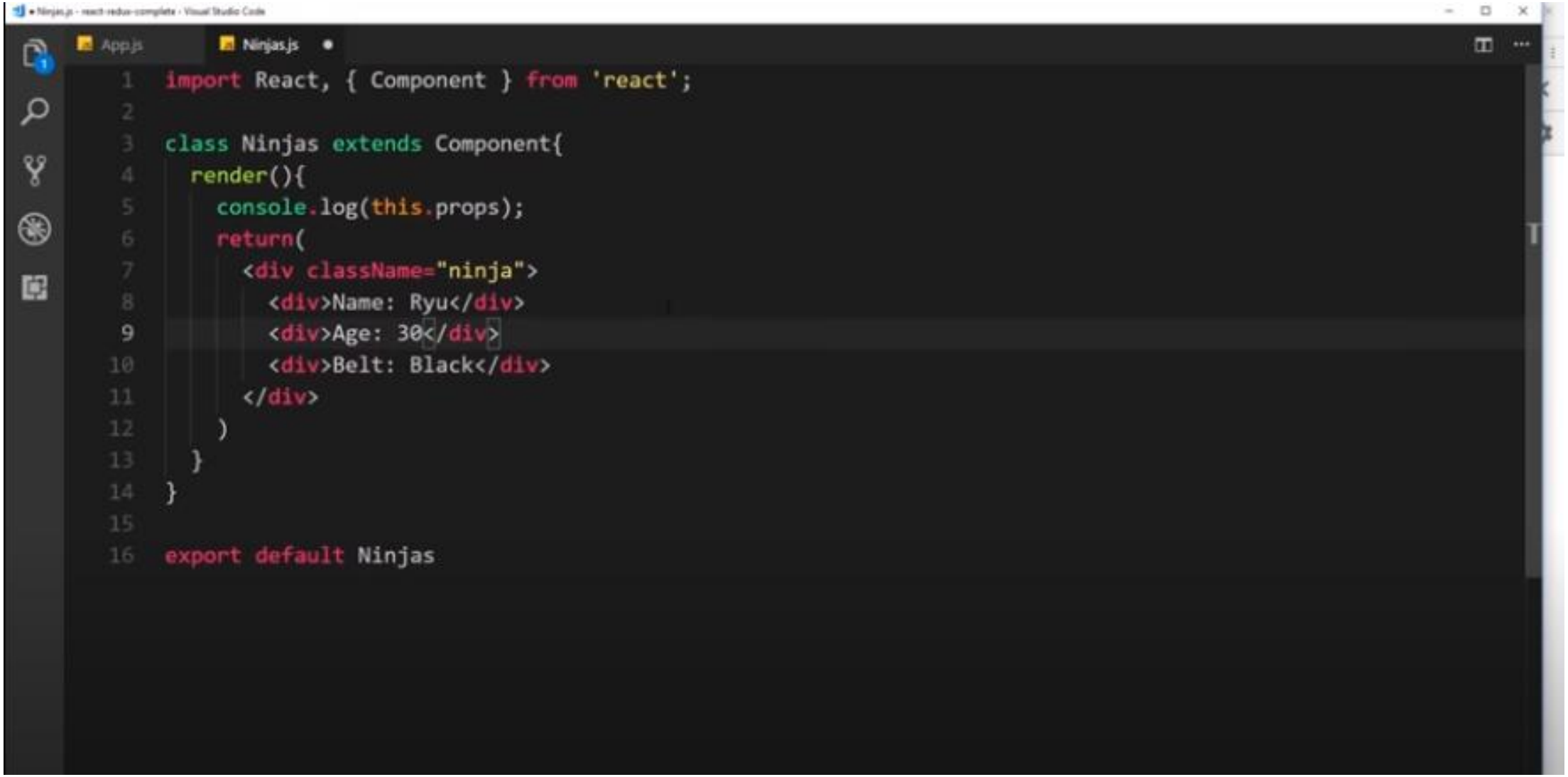
Injecting component


App.js
React App
localhost:3000
My first React app!
Welcome :)
Name: Ryu
Age: 30
Belt: Black

PROPS

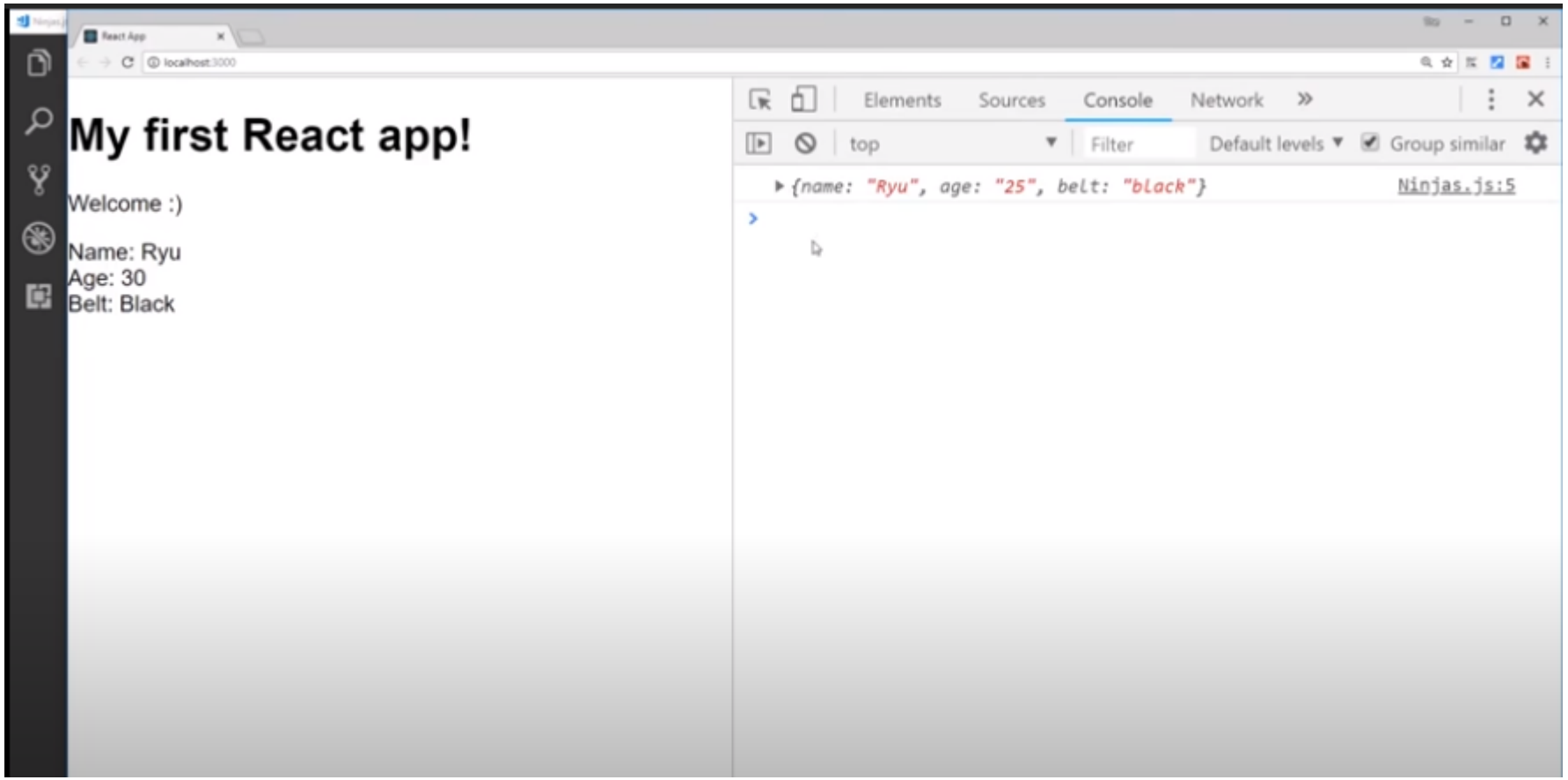


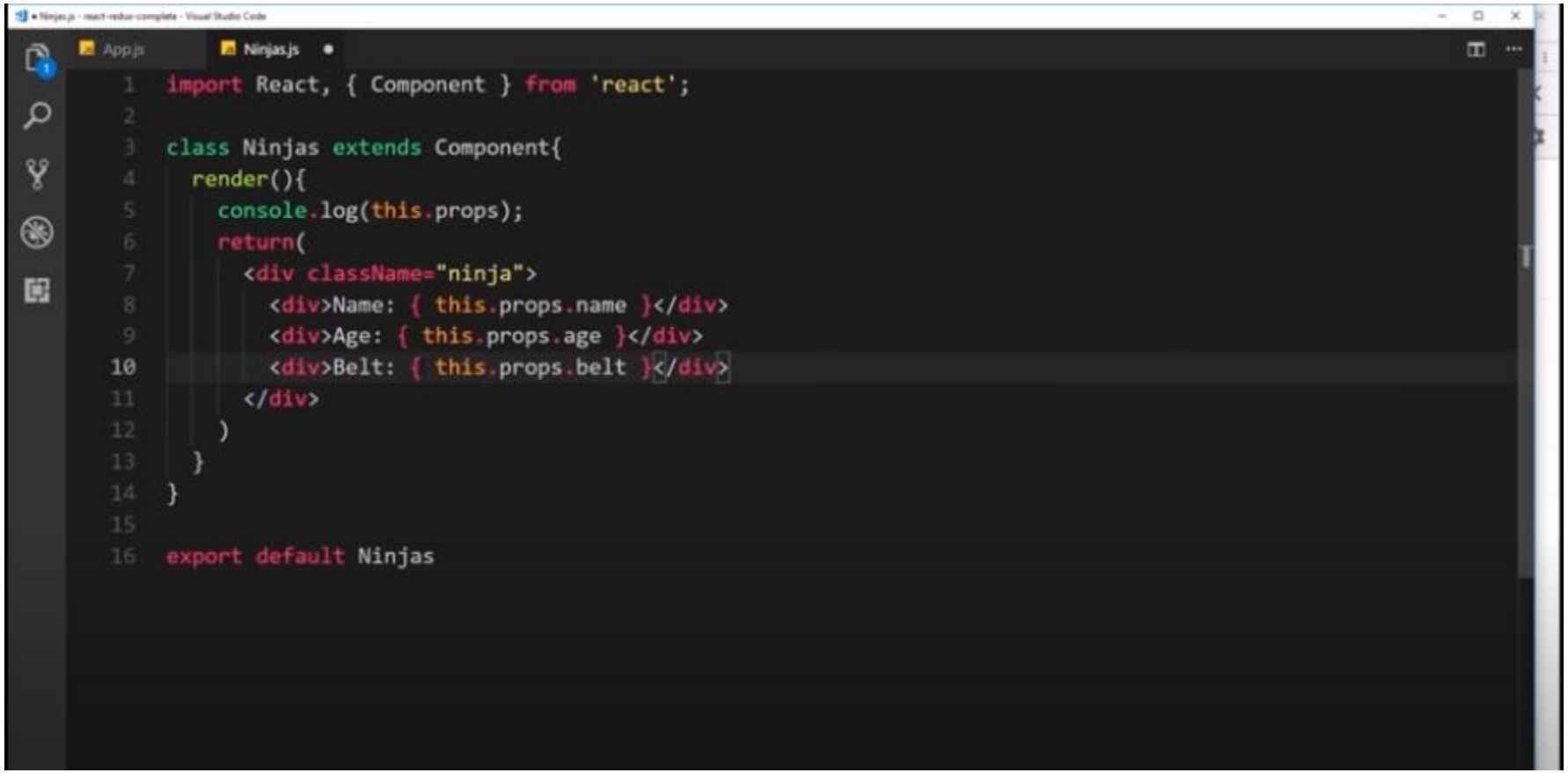
Passing data to component (Parent to child)


App.js
Ninjas.js
import React, { Component } from 'react';
class Ninjas extends Component{
render(){
console.log(this.props);
return(

Name: Ryu
Age: 30
Belt: Black

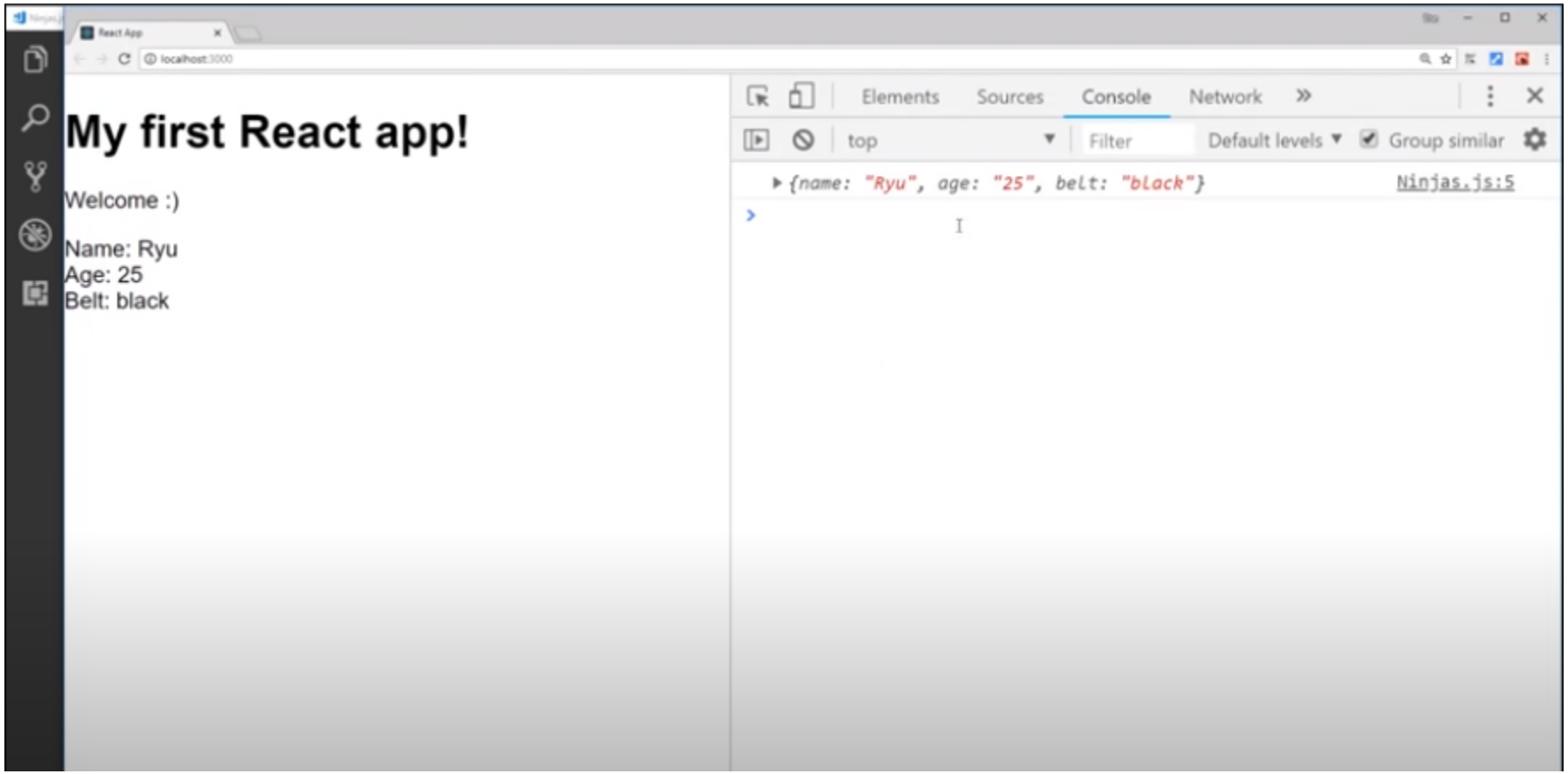
)
}
}
export default Ninjas

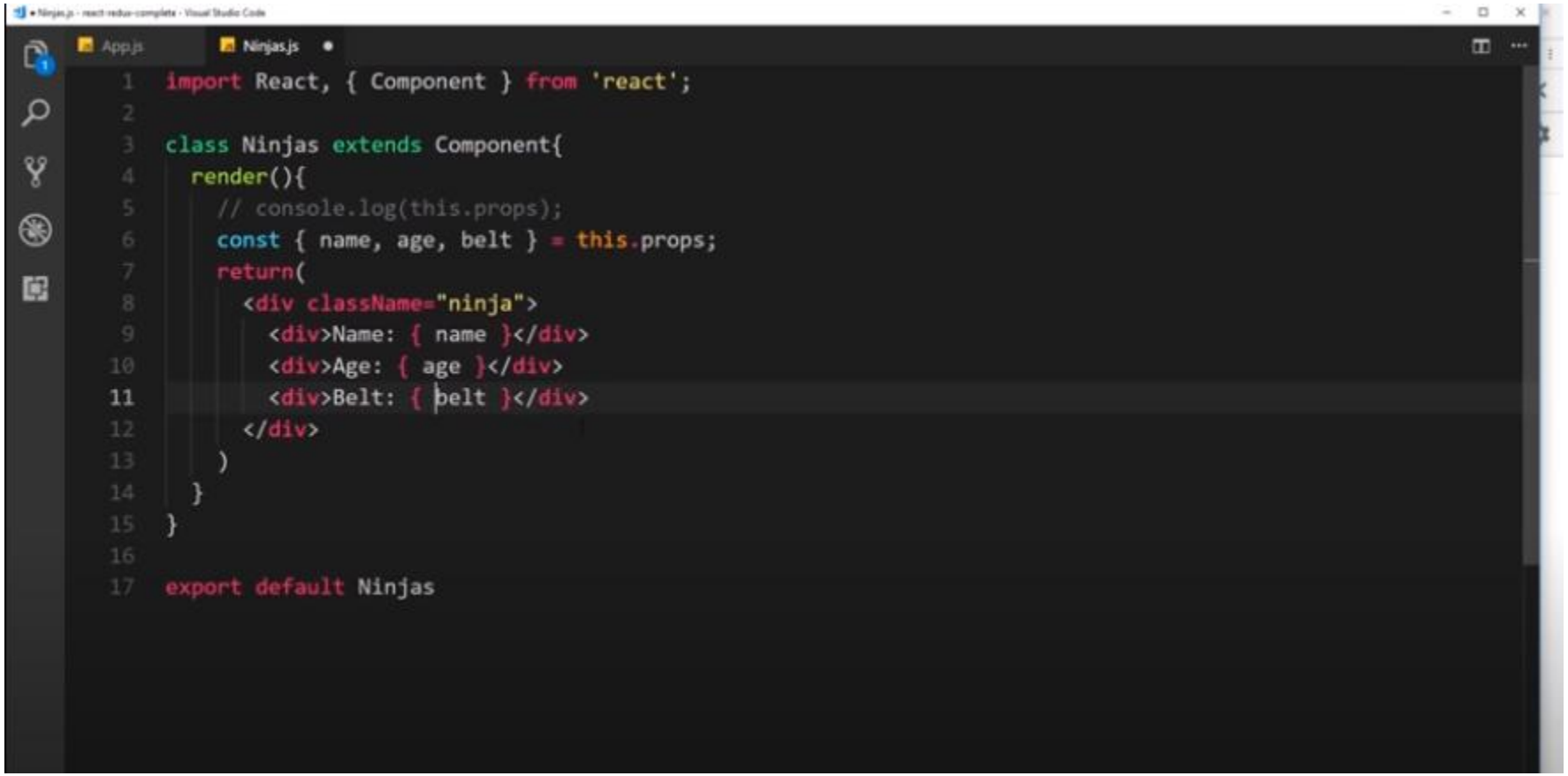

React App
localhost:3000
My first React app!
Welcome :)
Name: Ryu
Age: 30
Belt: Black
Elements
Sources
Console
Network
top
Filter
Default levels
Group similar
{name: "Ryu", age: "25", belt: "black"}
Ninjas.js:5


App.js
Ninjas.js
import React, { Component } from 'react';
class Ninjas extends Component{
render(){
console.log(this.props);
return(

Name: { this.props.name }
Age: { this.props.age }
Belt: { this.props.belt }

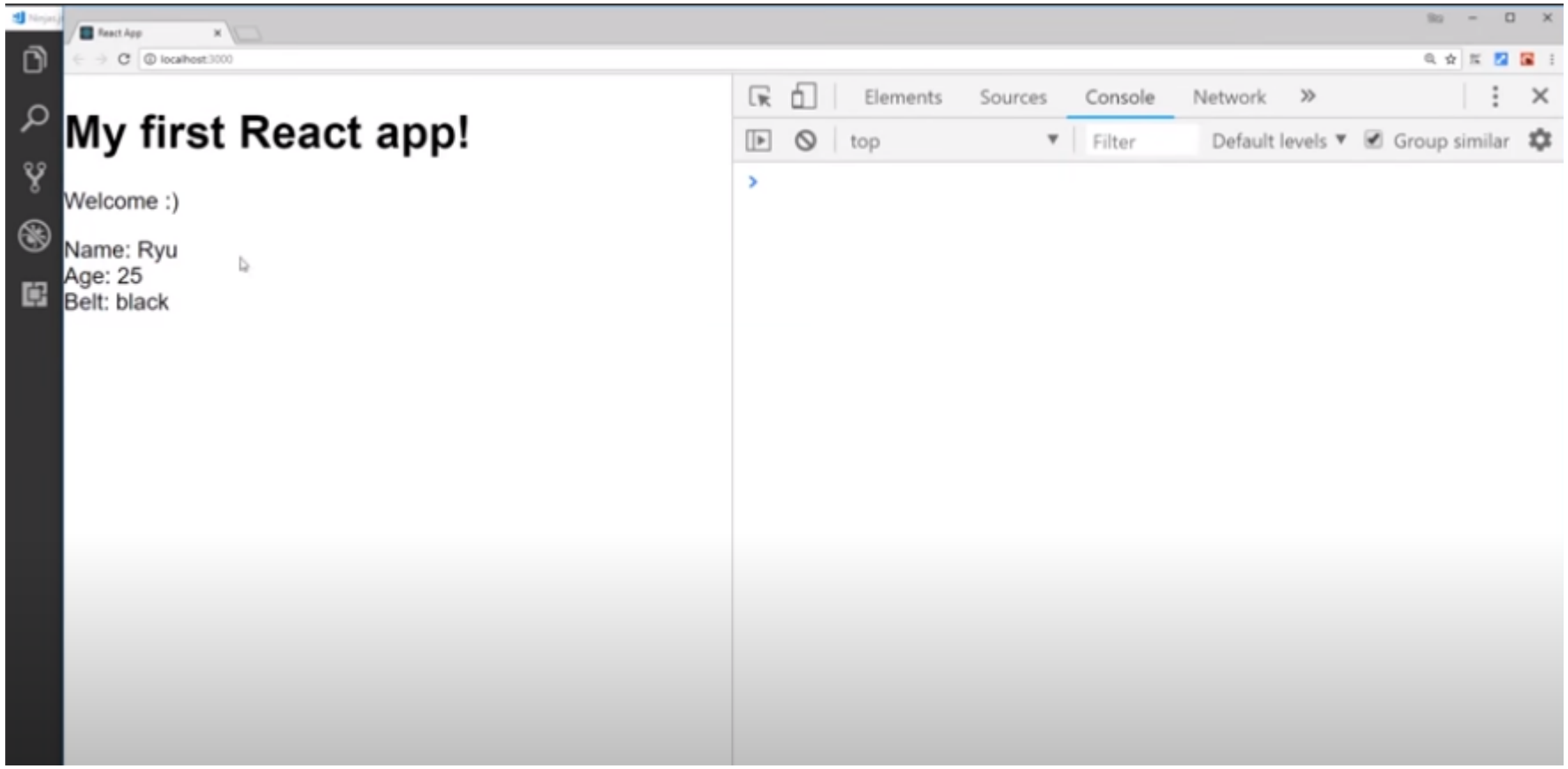
)
}
}
export default Ninjas


React App
localhost:3000
My first React app!
Welcome :)
Name: Ryu
Age: 25
Belt: black
Elements
Sources
Console
Network
top
Filter
Default levels
Group similar
{name: "Ryu", age: "25", belt: "black"}
Ninjas.js:5

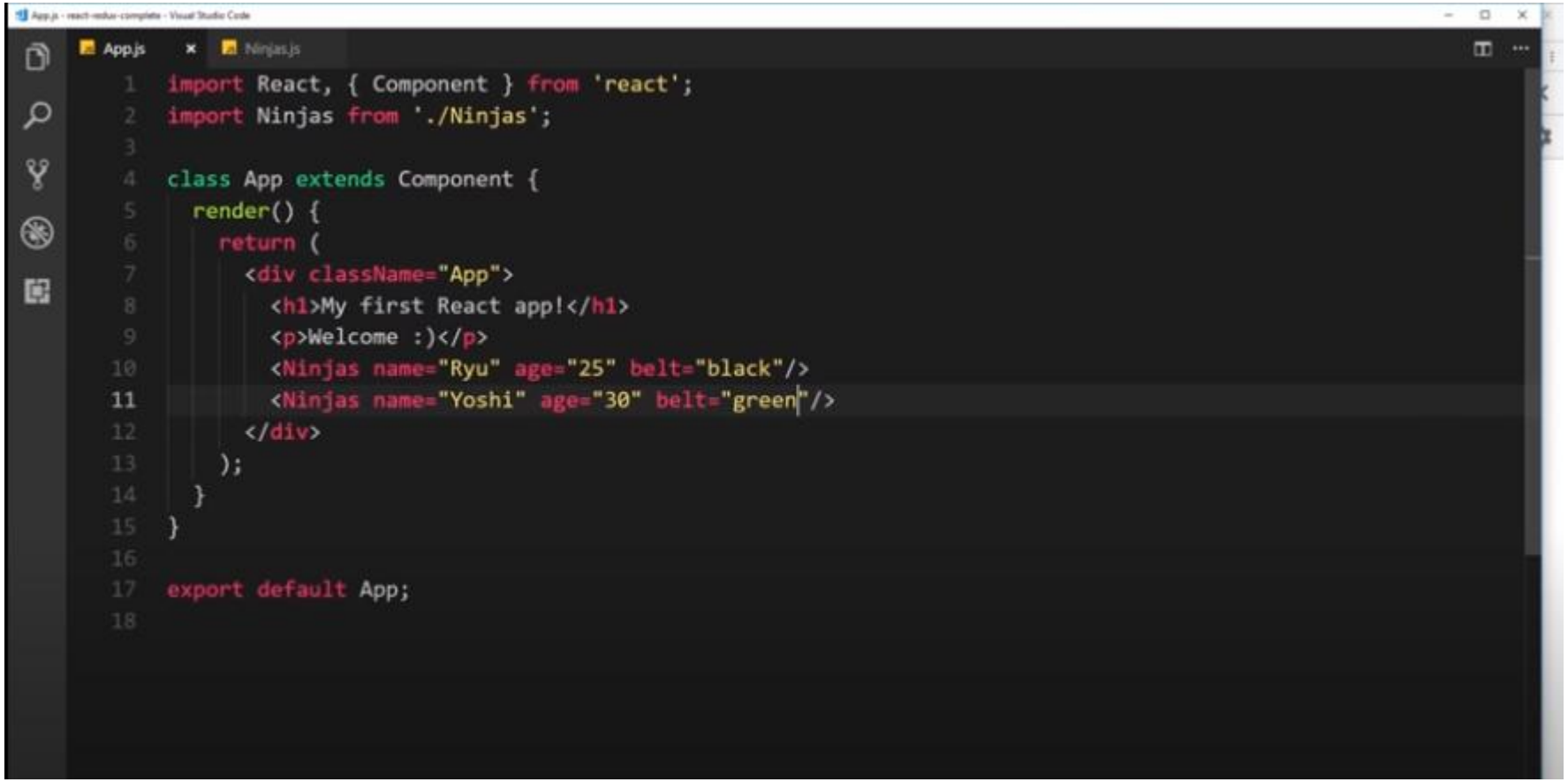

App.js
Ninjas.js
import React, { Component } from 'react';
class Ninjas extends Component{
render(){
// console.log(this.props);
const { name, age, belt } = this.props;
return(

Name: { name }
Age: { age }
Belt: { belt }

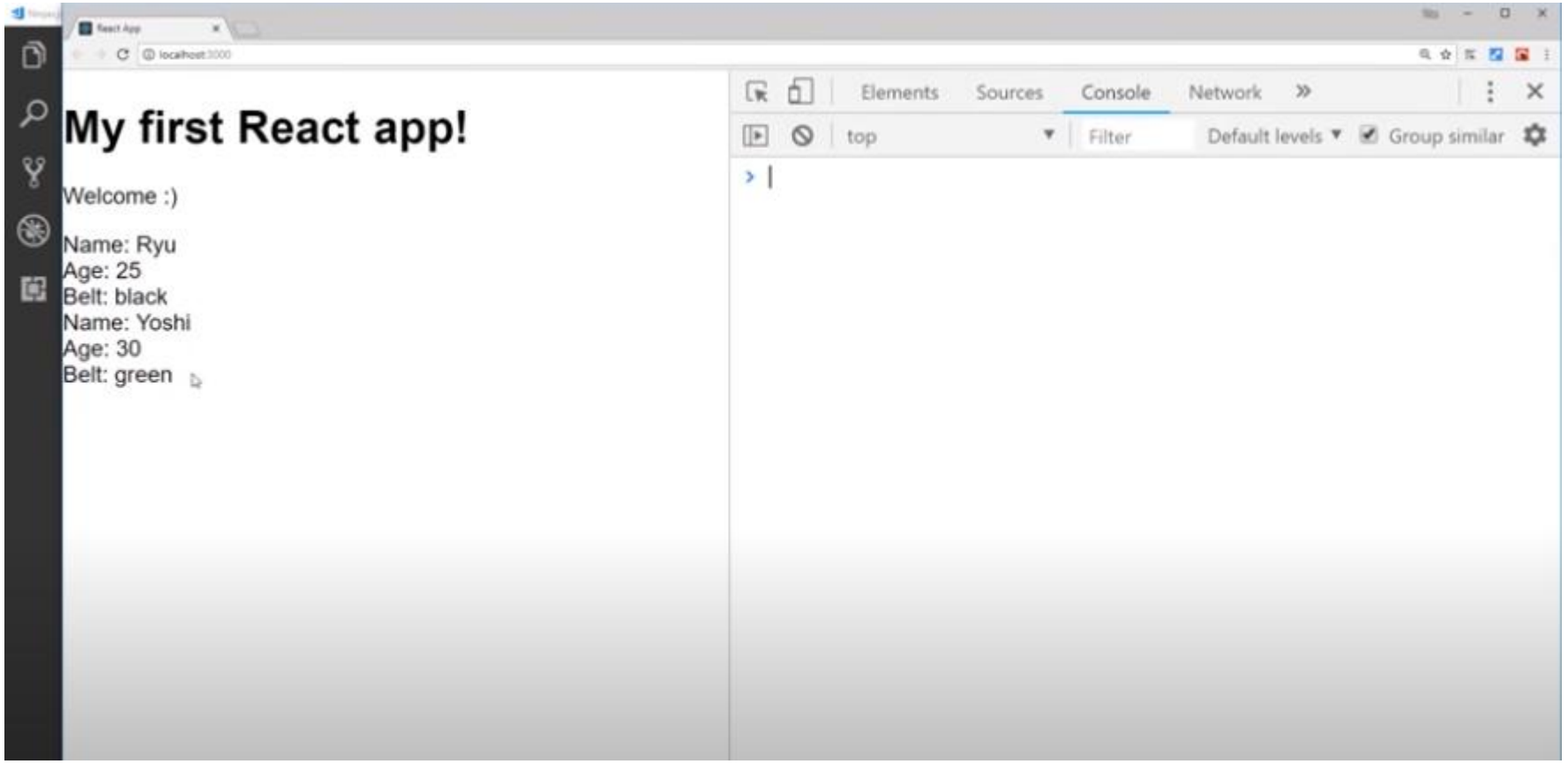
)
}
}
export default Ninjas



Still same output

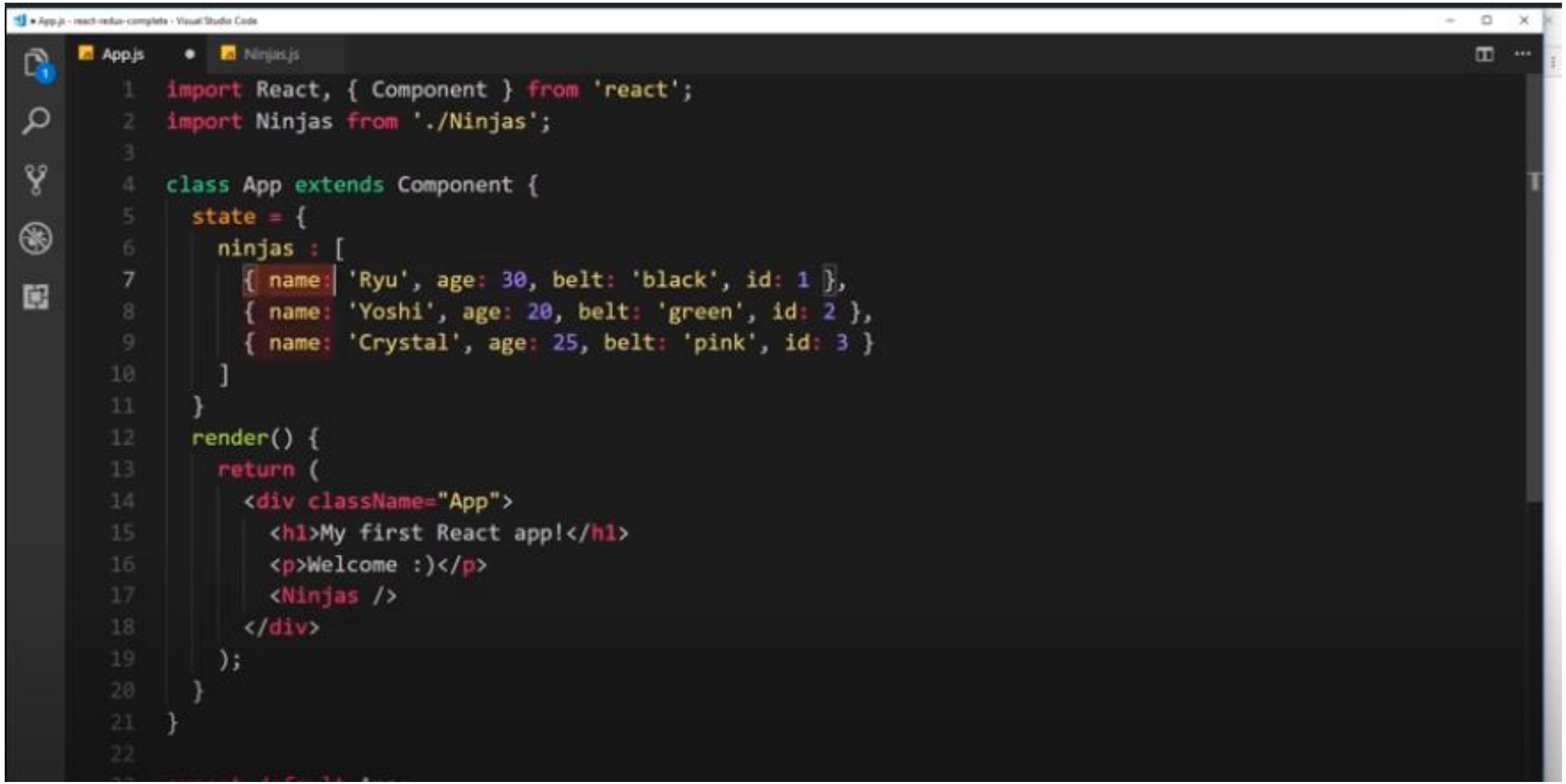


Use same component with diff values: dynamic behavior



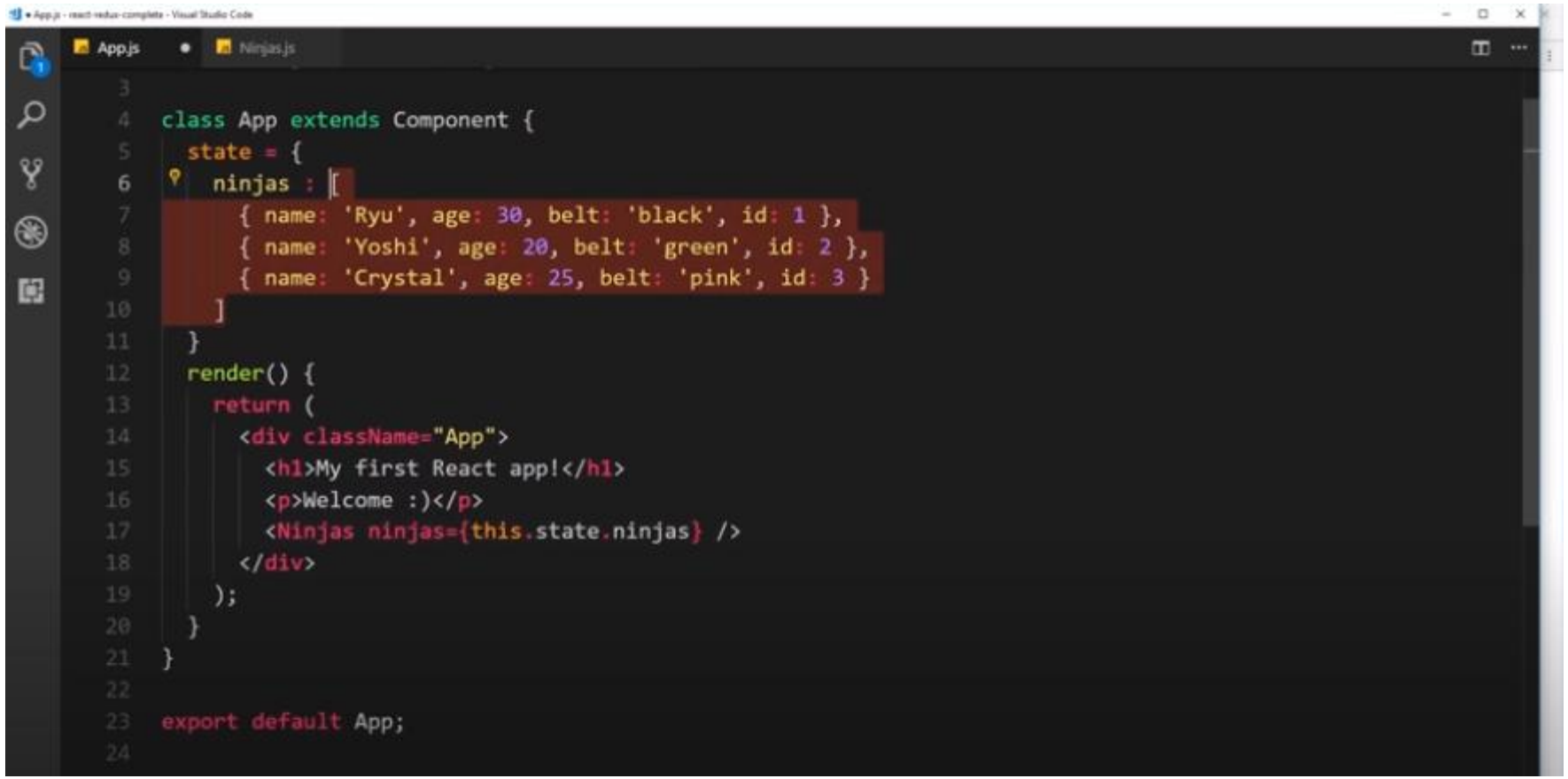
What it looks on browser

OUTPUTTING LISTS


App.js
Ninjas.js
import React, { Component } from 'react';
import Ninjas from './Ninjas';
class App extends Component {
state = {
ninjas : [
{ name: 'Ryu', age: 30, belt: 'black', id: 1 },
{ name: 'Yoshi', age: 20, belt: 'green', id: 2 },
{ name: 'Crystal', age: 25, belt: 'pink', id: 3 }
]
}
render() {
return (

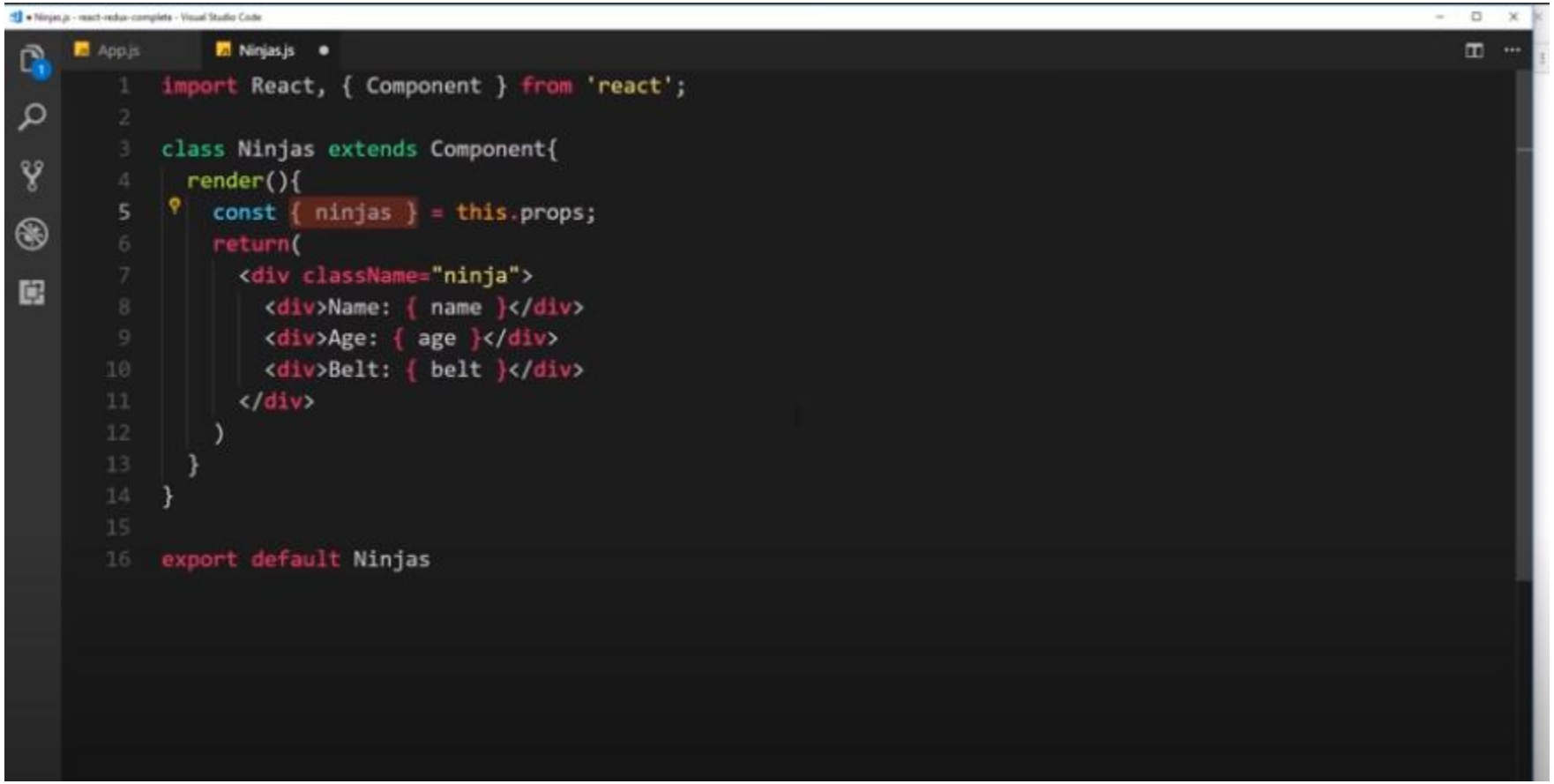
<h1>My first React app!</h1>
Welcome :)
<Ninjas />

);
}
}


App.js
Ninjas.js
class App extends Component {
state = {
ninjas : [
{ name: 'Ryu', age: 30, belt: 'black', id: 1 },
{ name: 'Yoshi', age: 20, belt: 'green', id: 2 },
{ name: 'Crystal', age: 25, belt: 'pink', id: 3 }
]
}
render() {
return (

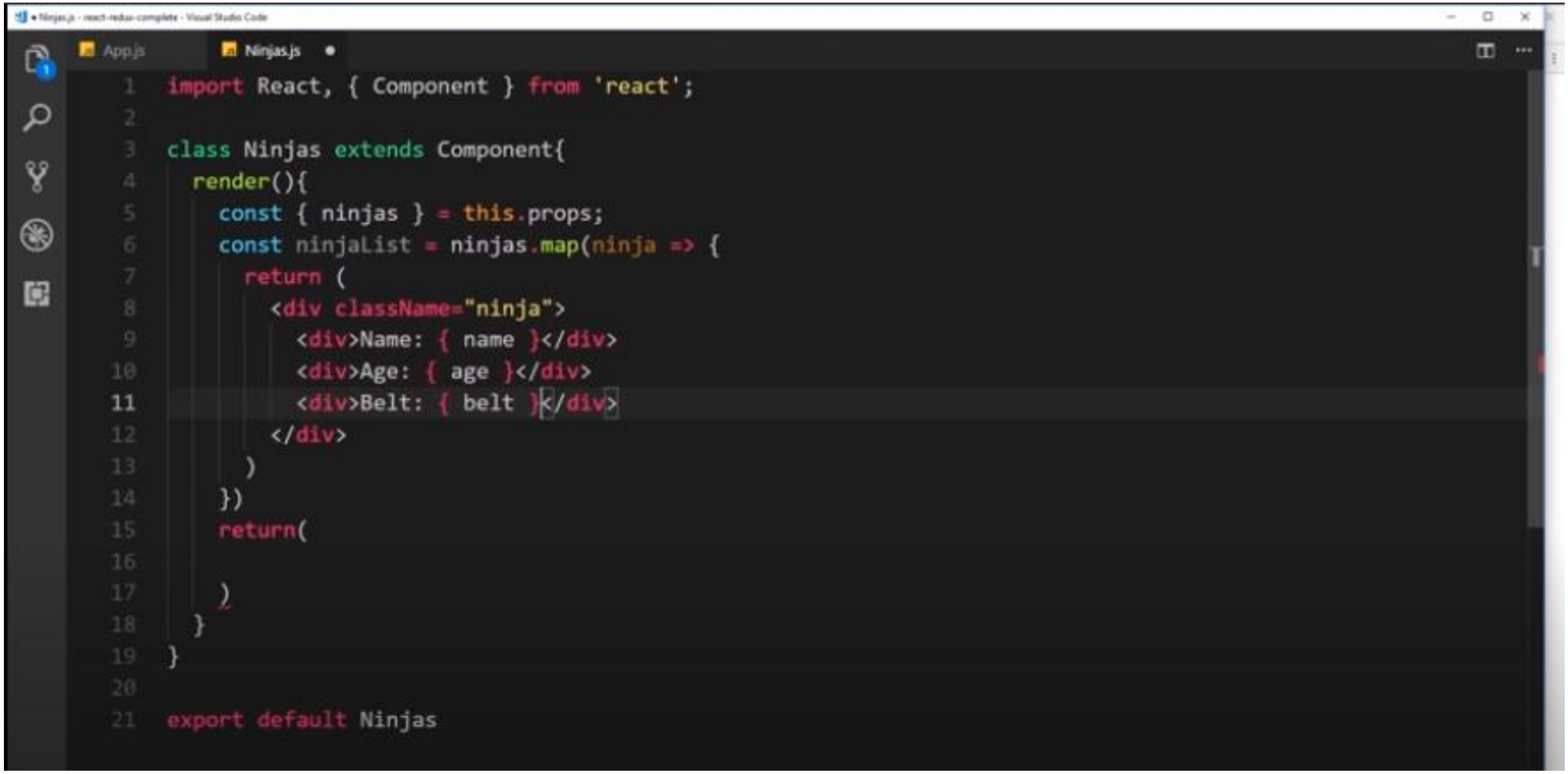
<h1>My first React app!</h1>
Welcome :)
<Ninjas ninjas={this.state.ninjas} />

);
}
}
export default App;


App.js
Ninjas.js
import React, { Component } from 'react';
class Ninjas extends Component{
render(){
const { ninjas } = this.props;
return(

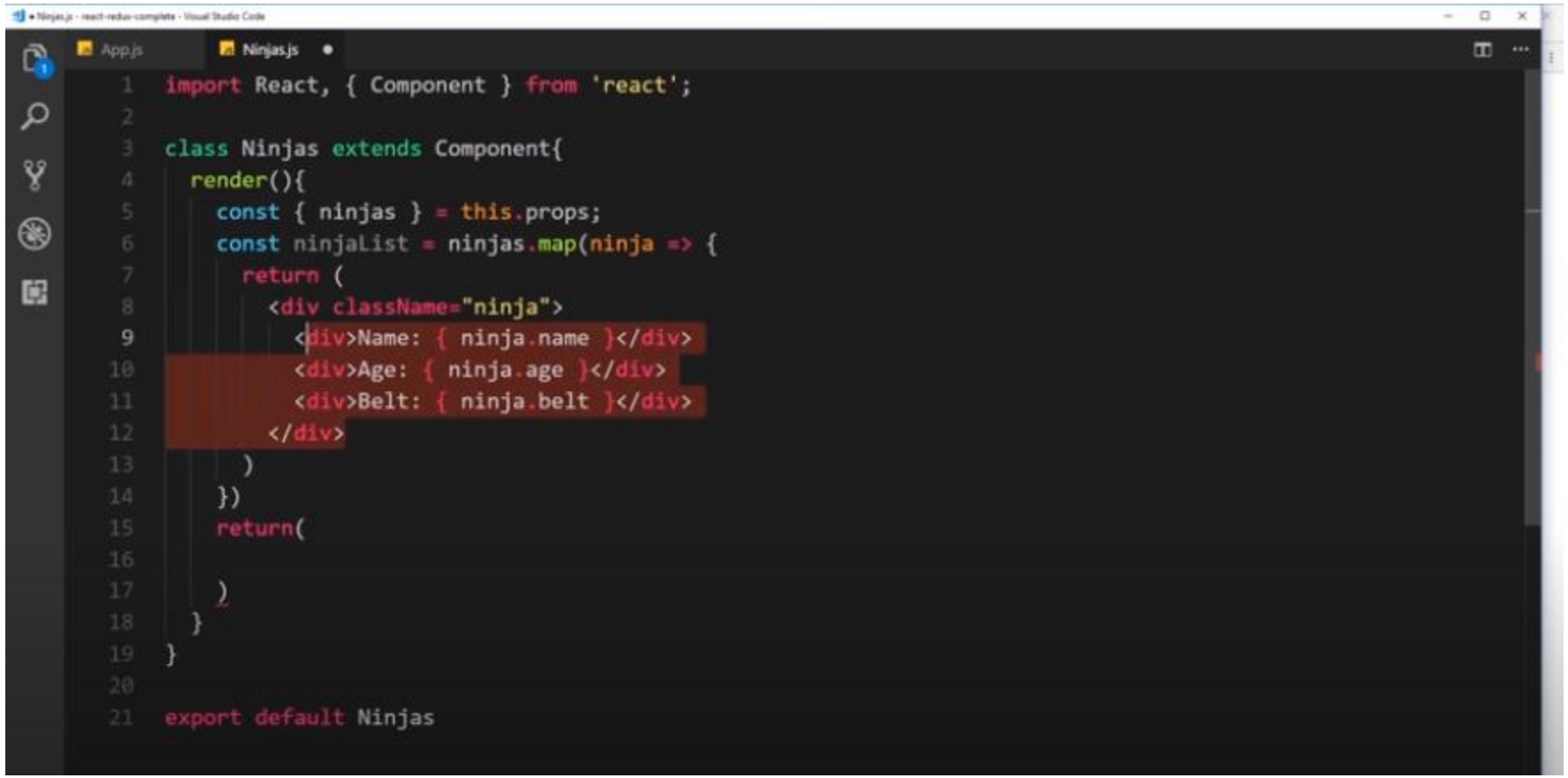
Name: { name }
Age: { age }
Belt: { belt }

)
}
}
export default Ninjas


App.js
Ninjas.js
import React, { Component } from 'react';
class Ninjas extends Component{
render(){
const { ninjas } = this.props;
const ninjaList = ninjas.map(ninja => {
return (

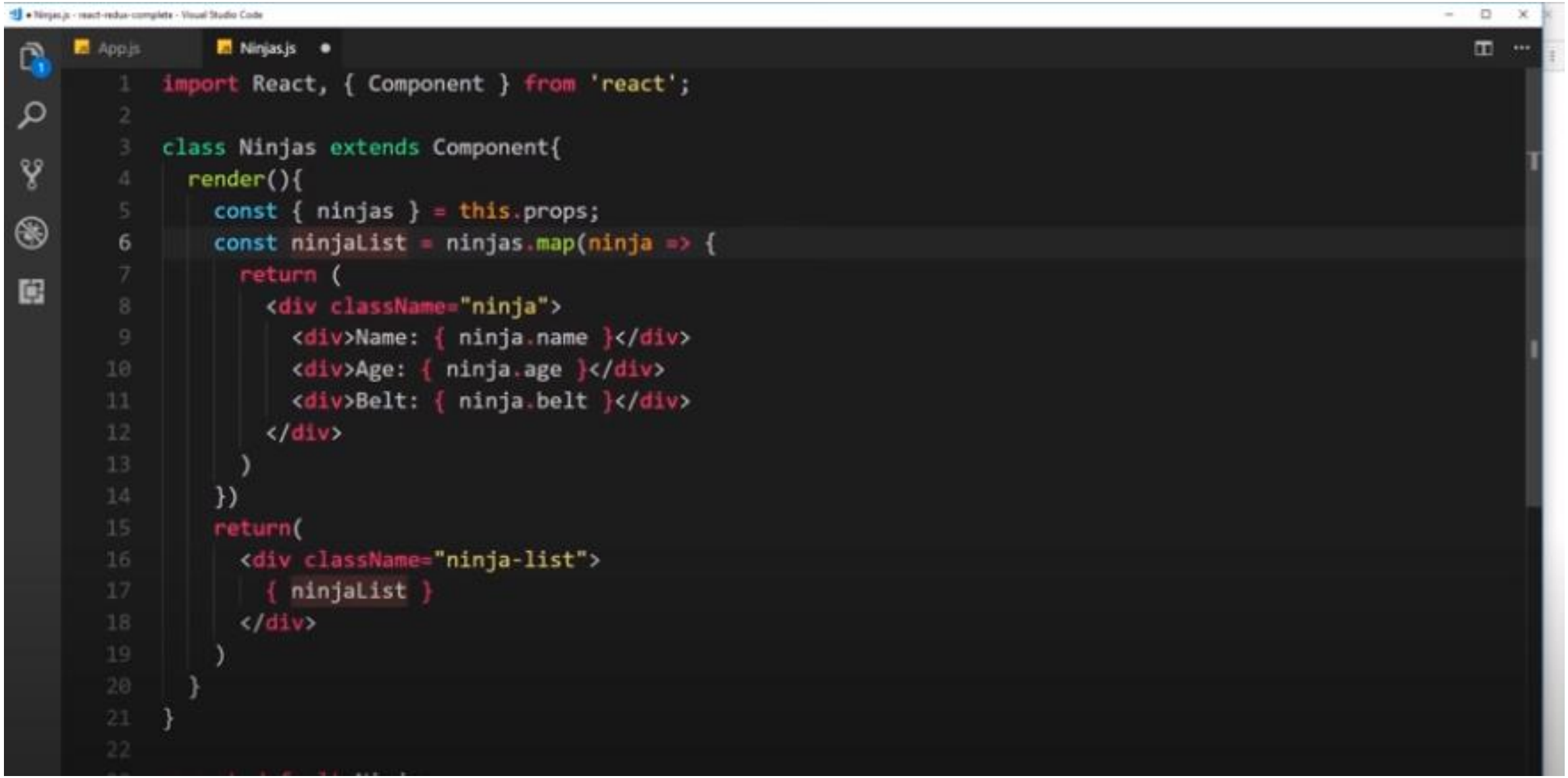
Name: { name }
Age: { age }
Belt: { belt }

)
})
return(
)
}
}
export default Ninjas


App.js
Ninjas.js
import React, { Component } from 'react';
class Ninjas extends Component{
render(){
const { ninjas } = this.props;
const ninjaList = ninjas.map(ninja => {
return (

Name: { ninja.name }
Age: { ninja.age }
Belt: { ninja.belt }

)
})
return(
)
}
}
export default Ninjas

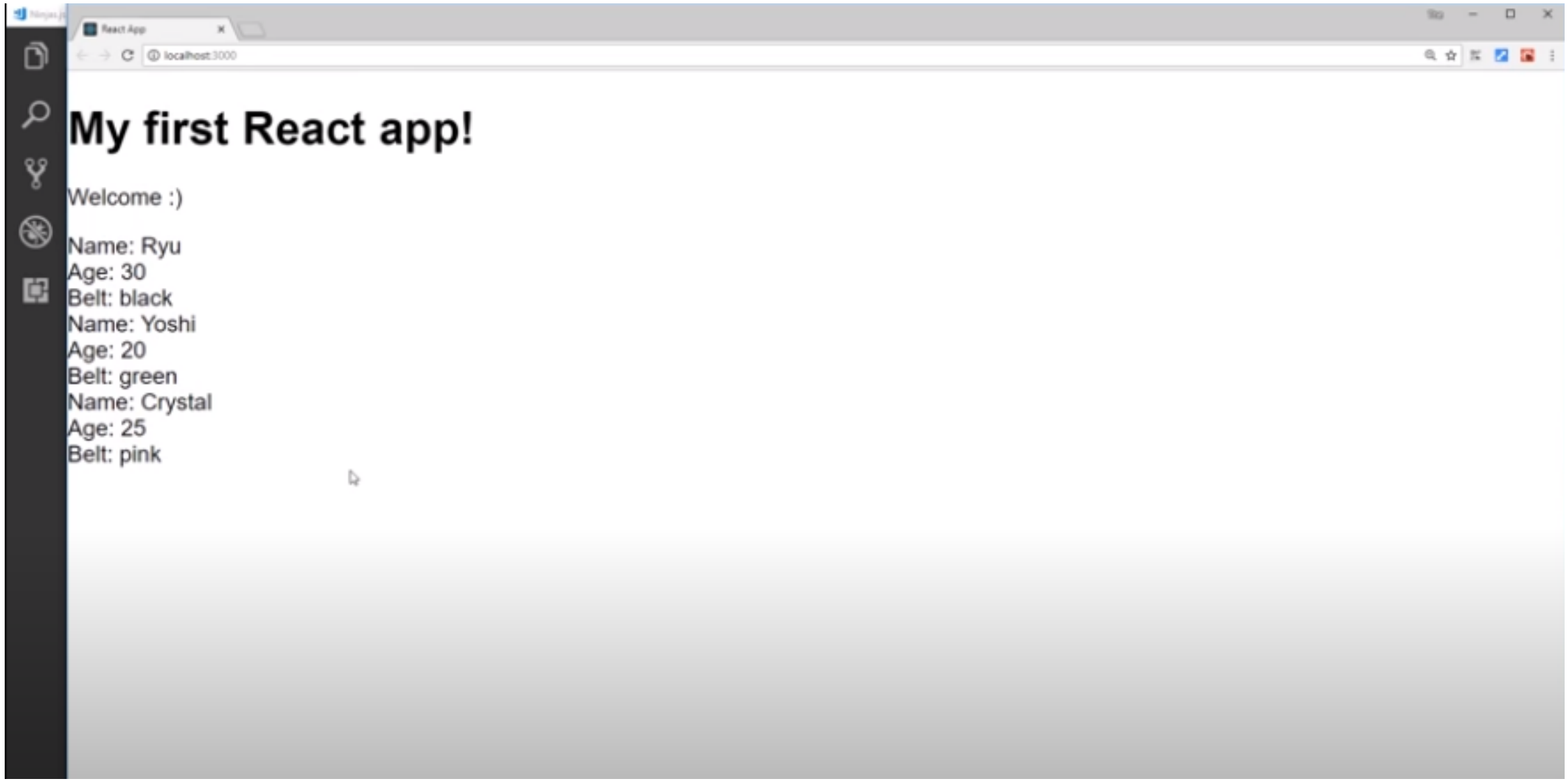

App.js
Ninjas.js
import React, { Component } from 'react';
class Ninjas extends Component{
render(){
const { ninjas } = this.props;
const ninjaList = ninjas.map(ninja => {
return (

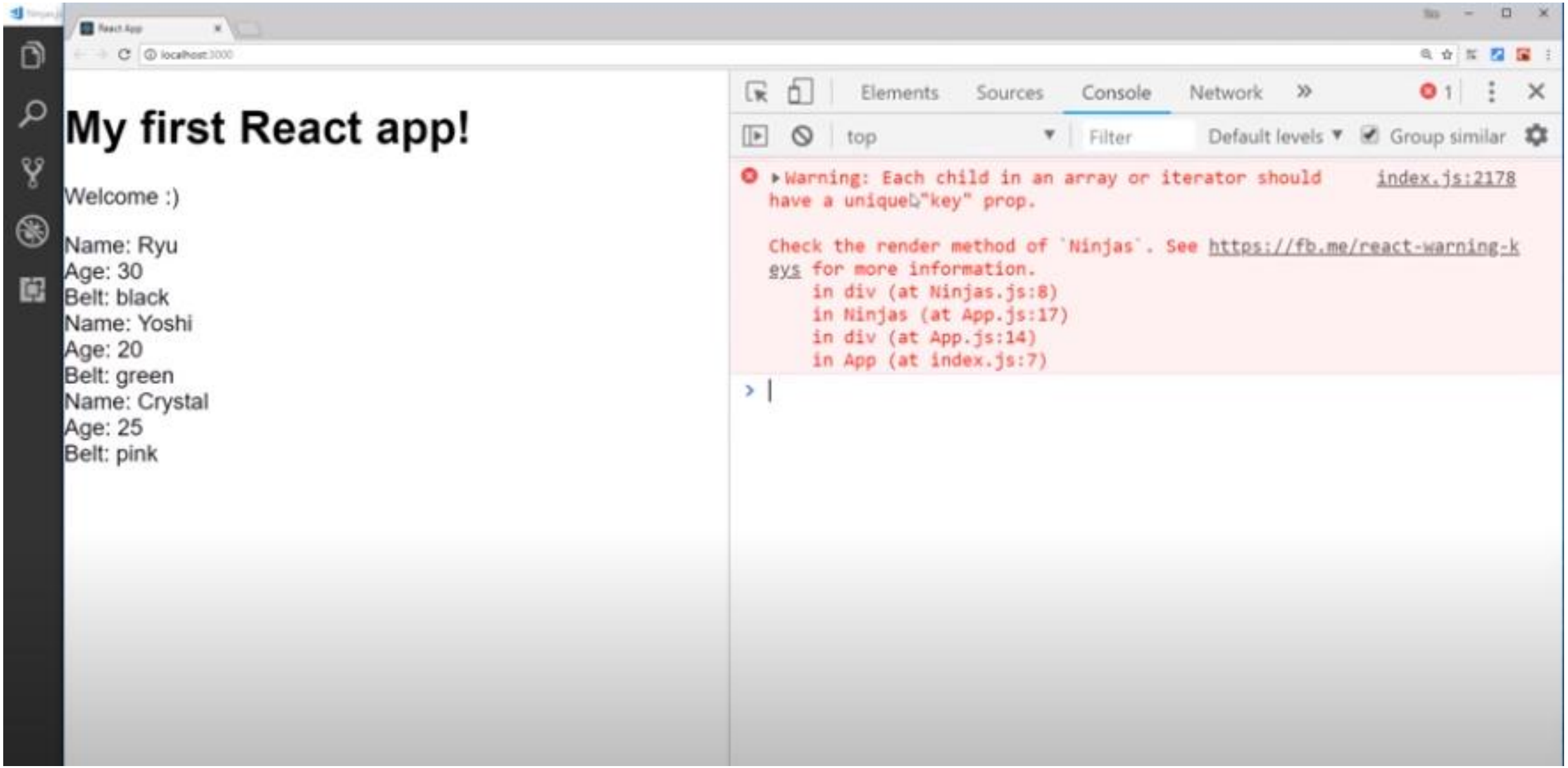
Name: { ninja.name }
Age: { ninja.age }
Belt: { ninja.belt }

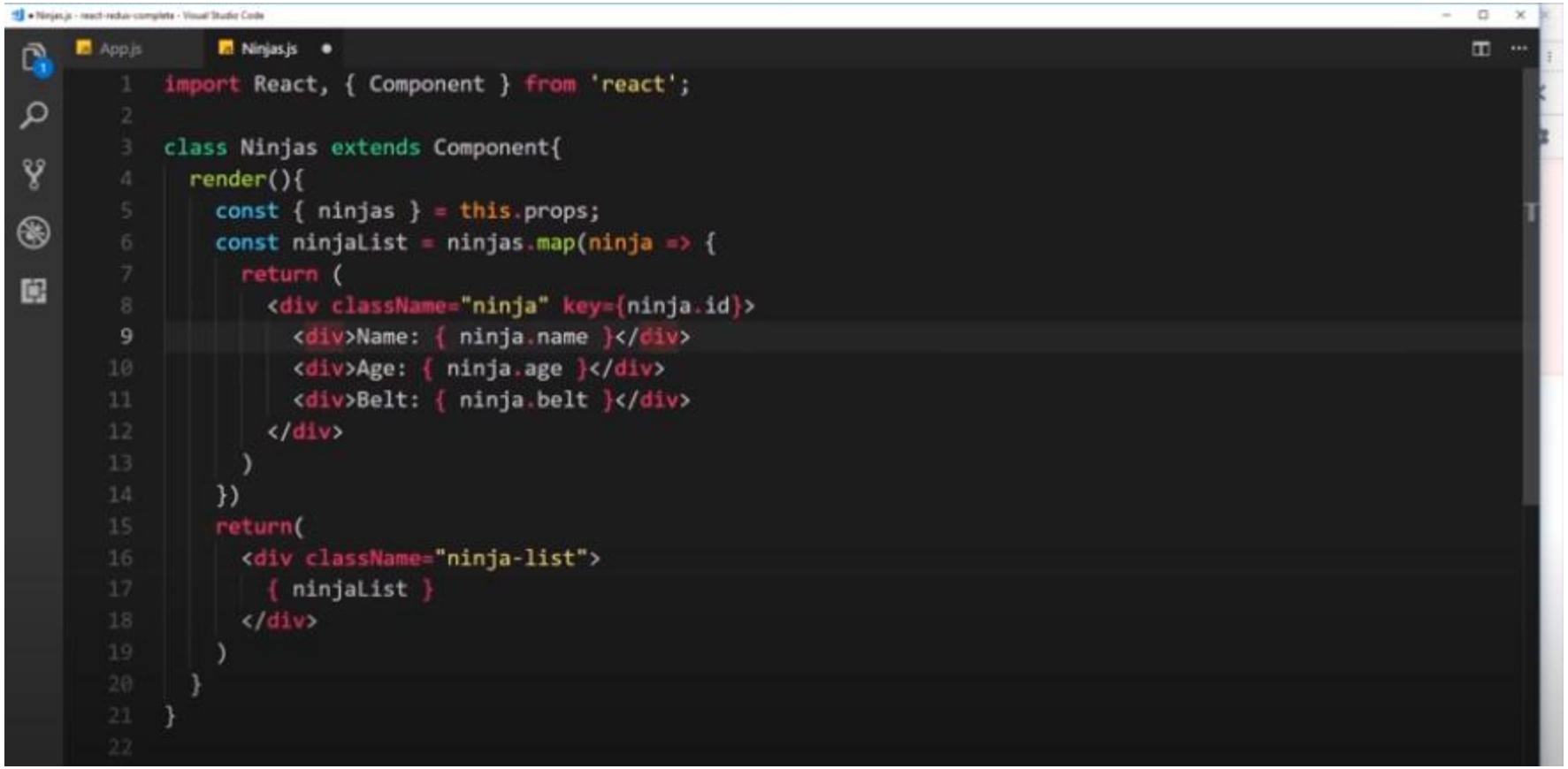
)
})
return(

{ ninjaList }

)
}
}


React App
localhost:3000
My first React app!
Welcome :)
Name: Ryu
Age: 30
Belt: black
Name: Yoshi
Age: 20
Belt: green
Name: Crystal
Age: 25
Belt: pink


React App
localhost:3000
My first React app!
Welcome :)
Name: Ryu
Age: 30
Belt: black
Name: Yoshi
Age: 20
Belt: green
Name: Crystal
Age: 25
Belt: pink
Elements
Sources
Console
Network
top
Filter
Default levels
Group similar
Warning: Each child in an array or iterator should have a unique "key" prop.
index.js:2178
Check the render method of `Ninjas`. See https://fb.me/react-warning-keys for more information.
in div (at Ninjas.js:8)
in Ninjas (at App.js:17)
in div (at App.js:14)
in App (at index.js:7)

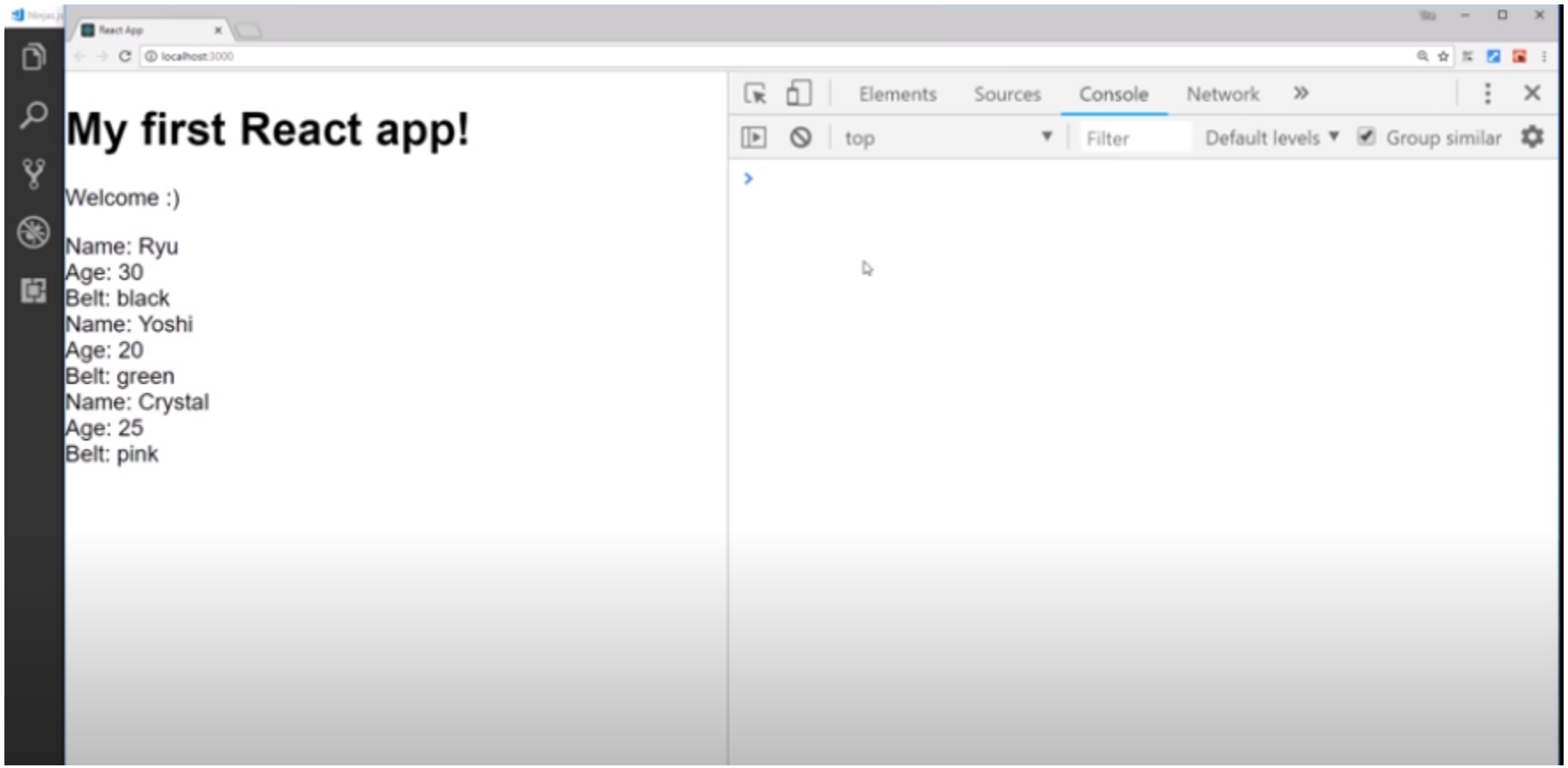

App.js
Ninjas.js
import React, { Component } from 'react';
class Ninjas extends Component{
render(){
const { ninjas } = this.props;
const ninjaList = ninjas.map(ninja => {
return (

Name: { ninja.name }
Age: { ninja.age }
Belt: { ninja.belt }

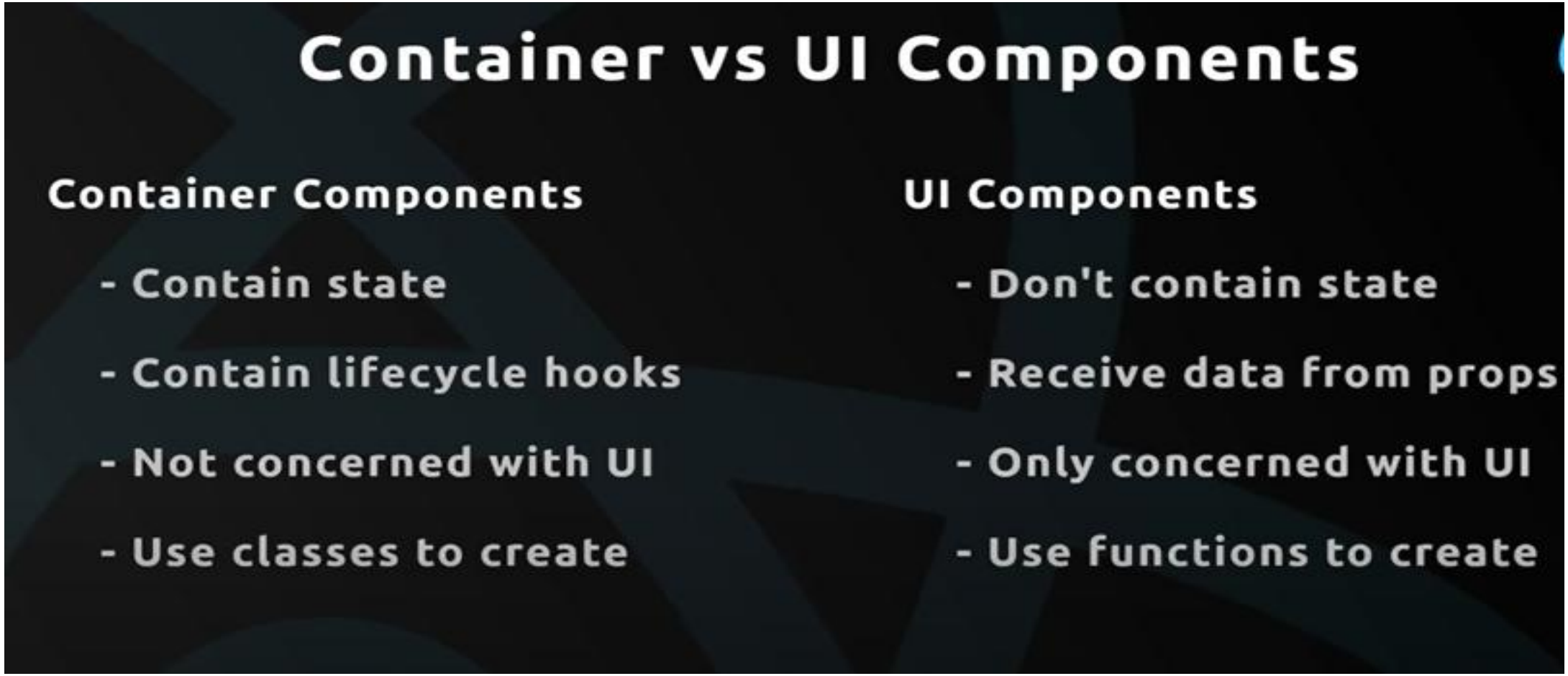
)
})
return(

{ ninjaList }

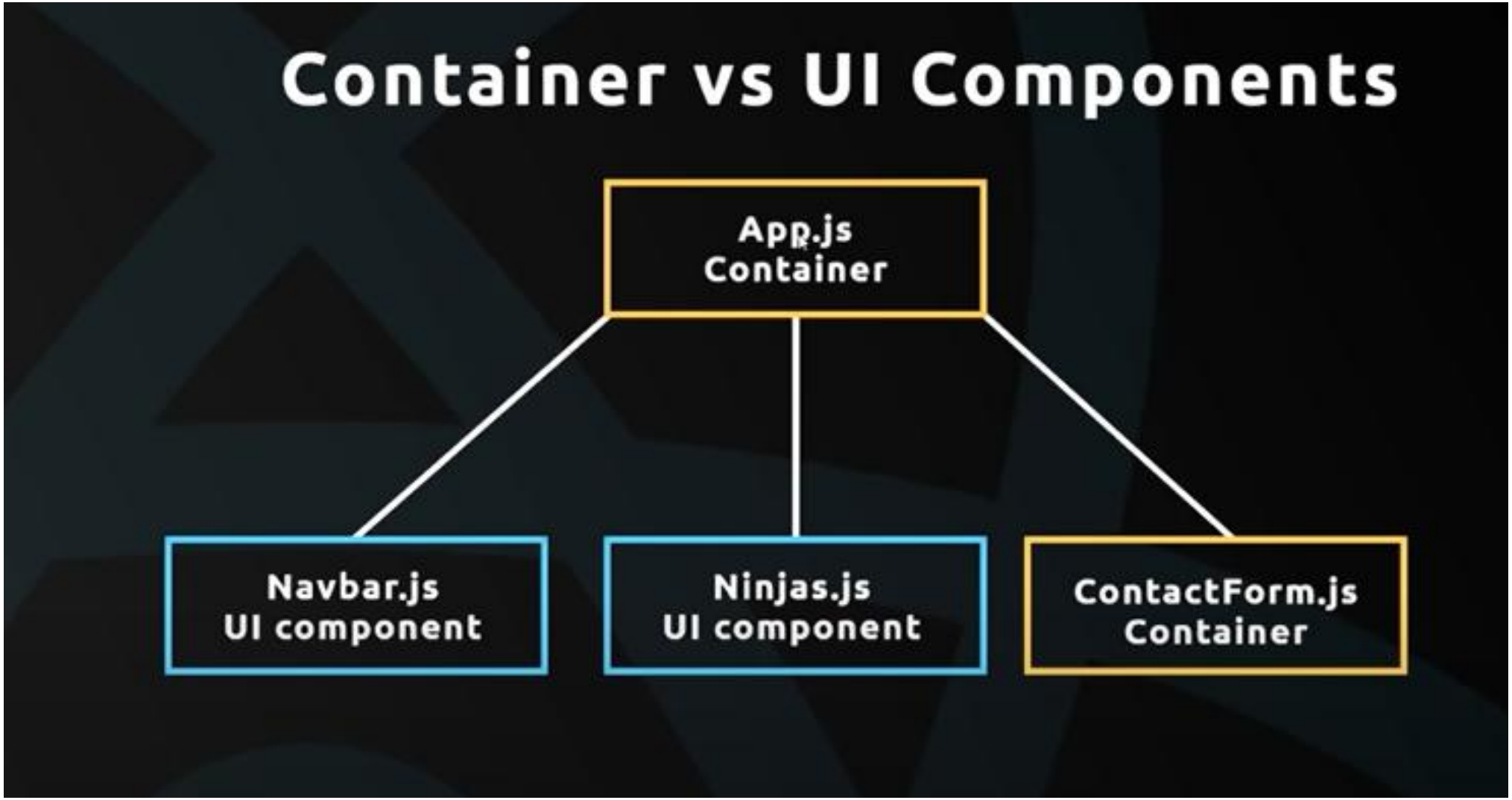
)
}
}

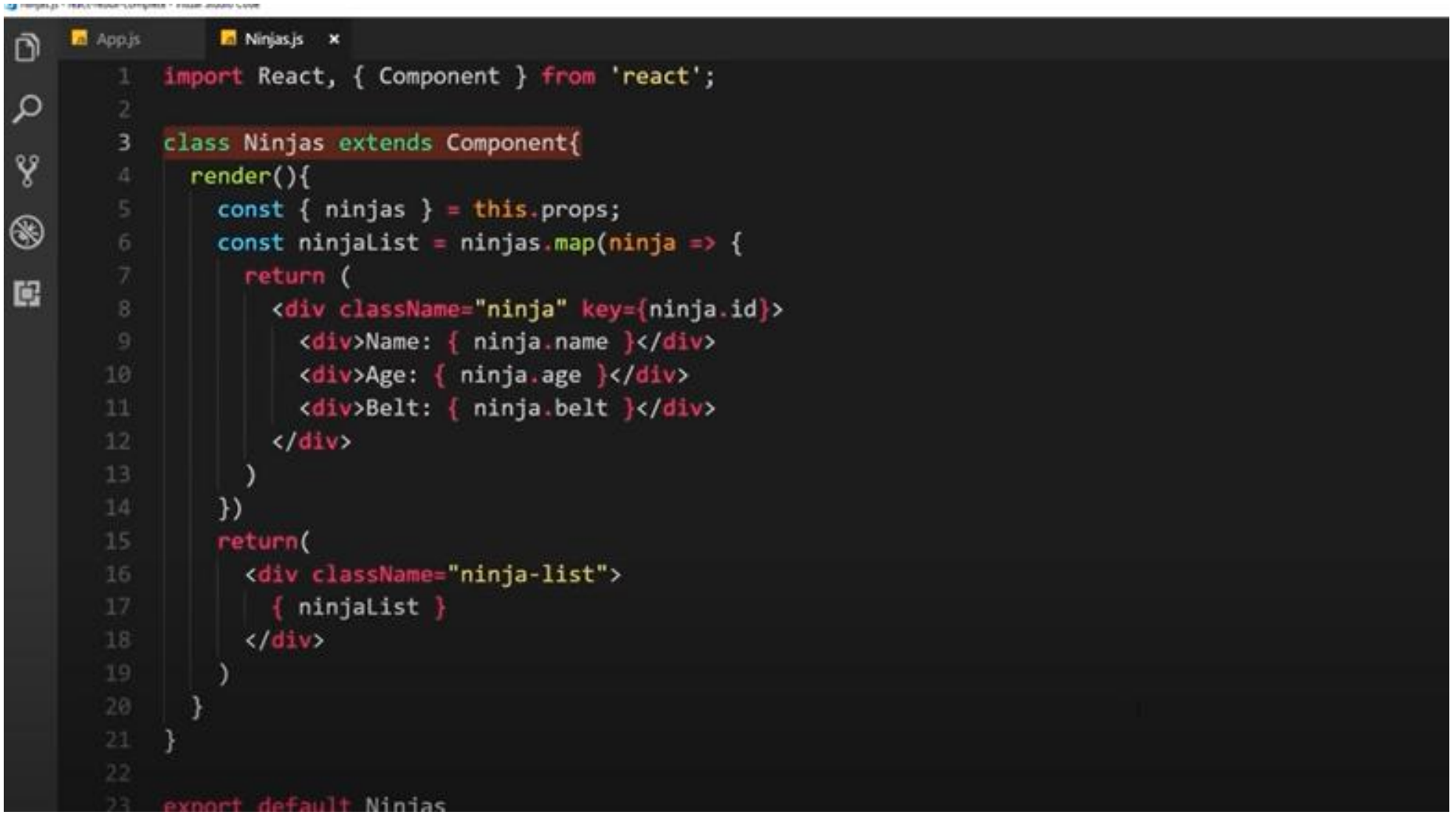

React App
localhost:3000
My first React app!
Welcome :)
Name: Ryu
Age: 30
Belt: black
Name: Yoshi
Age: 20
Belt: green
Name: Crystal
Age: 25
Belt: pink
Elements
Sources
Console
Network
top
Filter
Default levels
Group similar

STATELESS COMPONENTS



Container components are not concerned with UI or look of app
Container Comp.=Statefull Comp.=Class based Comp.
UI Comp.=Stateless Comp.=Functional Comp.


Container vs UI Components
App.js
Container
Navbar.js
UI component
Ninjas.js
UI component
ContactForm.js
Container

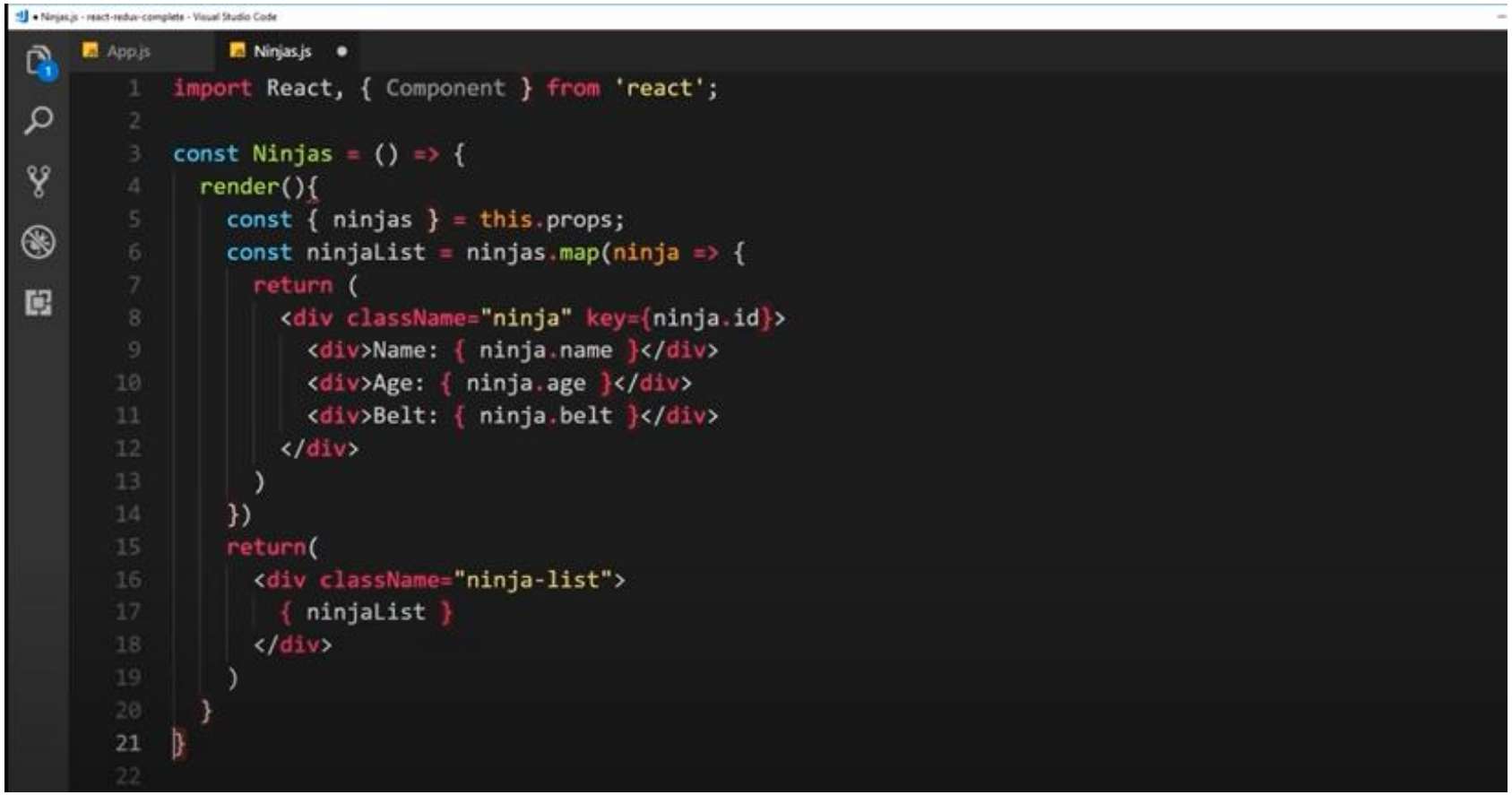

App.js
Ninjas.js
import React, { Component } from 'react';
class Ninjas extends Component{
render(){
const { ninjas } = this.props;
const ninjaList = ninjas.map(ninja => {
return (

Name: { ninja.name }
Age: { ninja.age }
Belt: { ninja.belt }

)
})
return(

{ ninjaList }

)
}
}
export default Ninjas

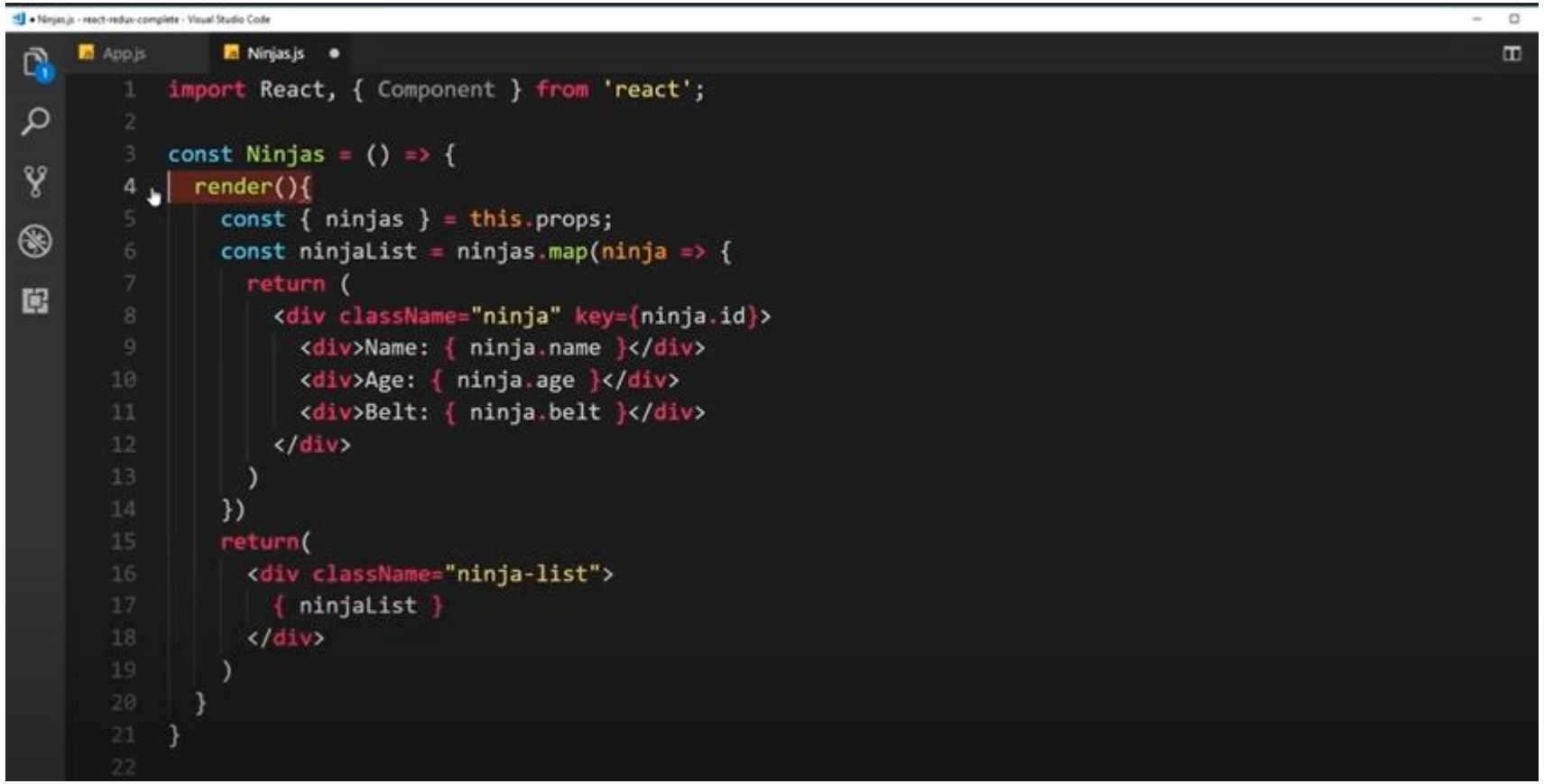

Ninjas.js - react-redux-complete - Visual Studio Code
App.js
Ninjas.js
import React, { Component } from 'react';
const Ninjas = () => {
render(){
const { ninjas } = this.props;
const ninjaList = ninjas.map(ninja => {
return (

Name: { ninja.name }
Age: { ninja.age }
Belt: { ninja.belt }

)
})
return(

{ ninjaList }

)
}
}

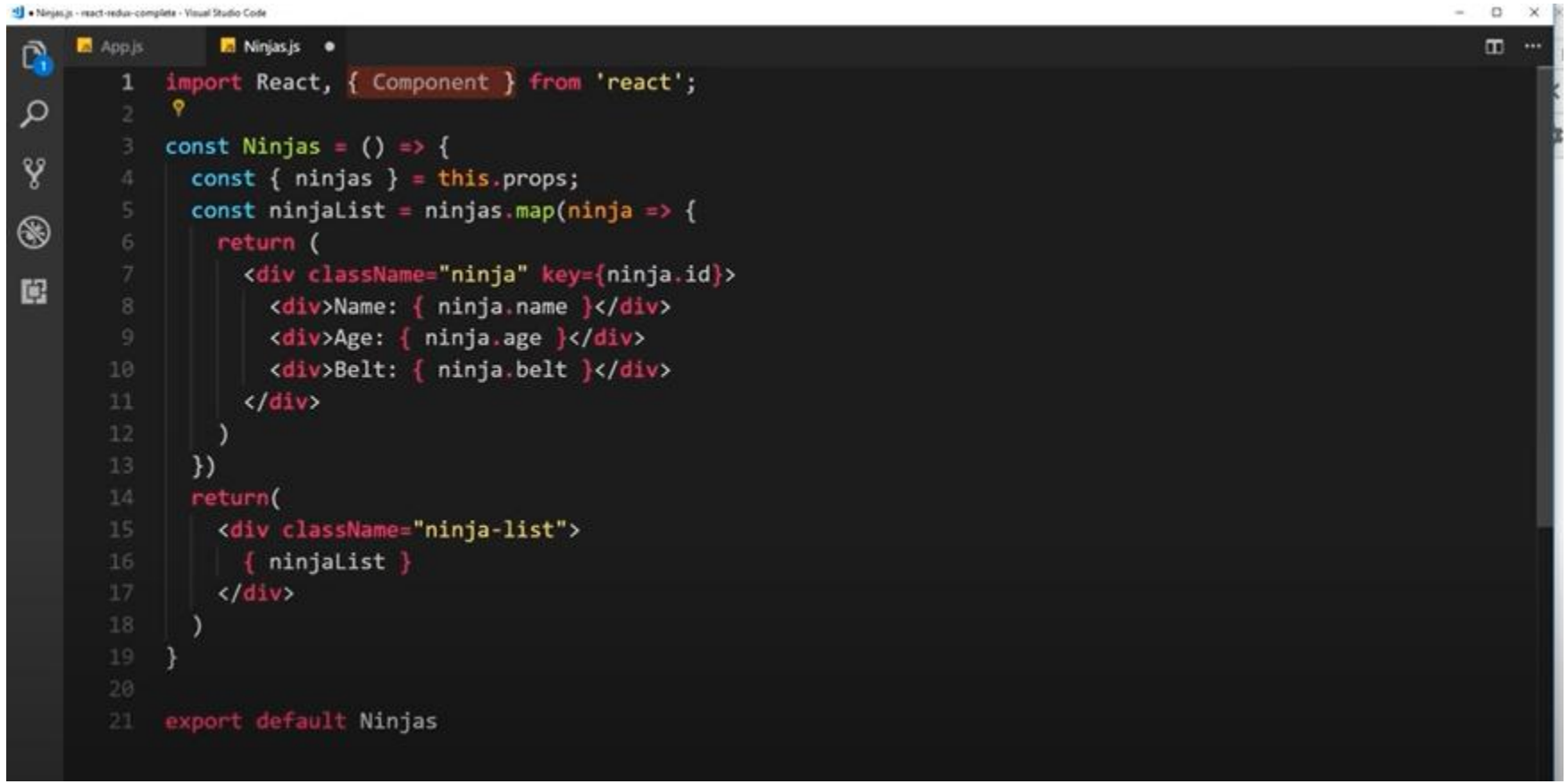

Ninjas.js - react-redux-complete - Visual Studio Code
App.js
Ninjas.js
import React, { Component } from 'react';
const Ninjas = () => {
render(){
const { ninjas } = this.props;
const ninjaList = ninjas.map(ninja => {
return (

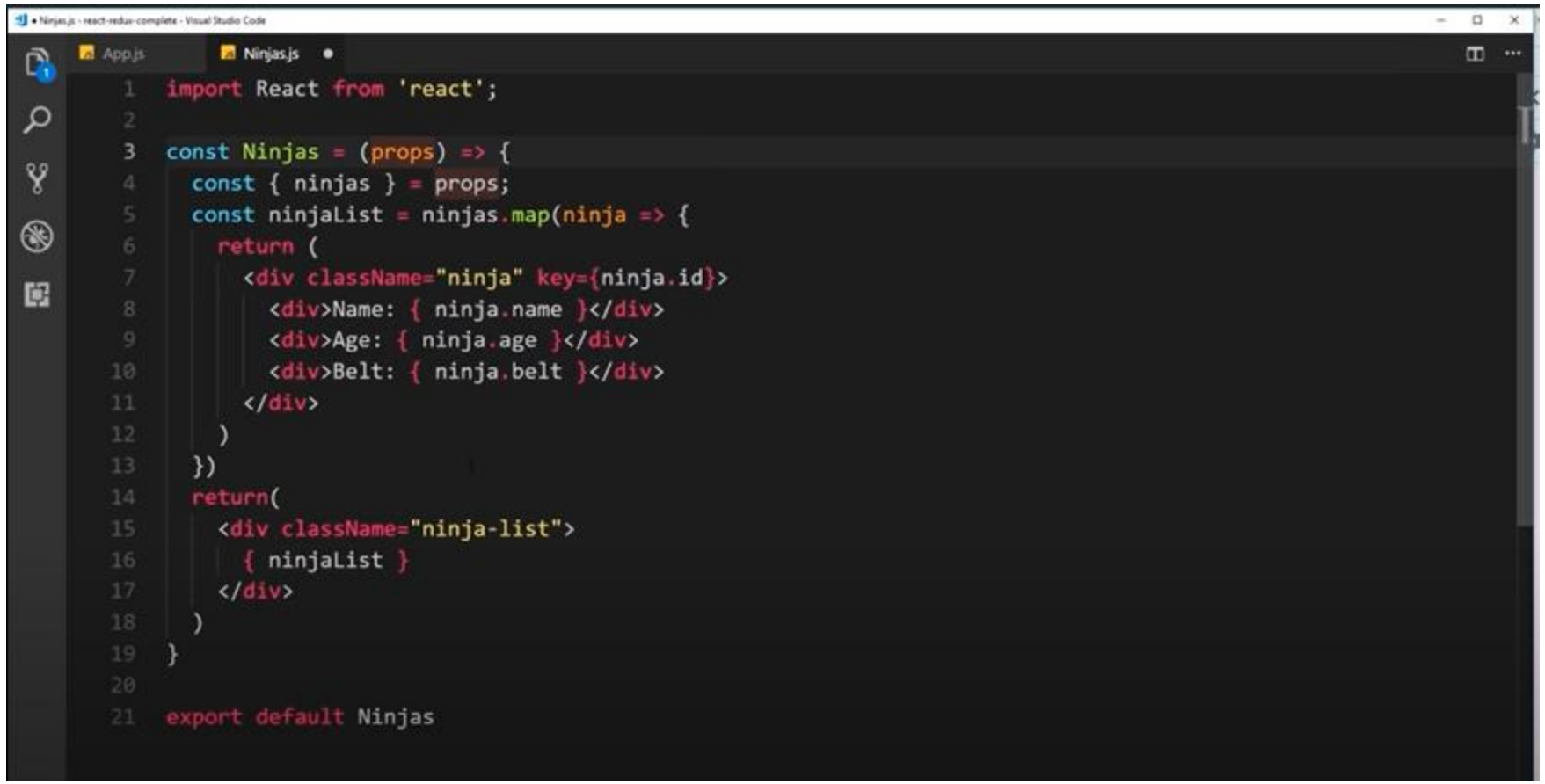
Name: { ninja.name }
Age: { ninja.age }
Belt: { ninja.belt }

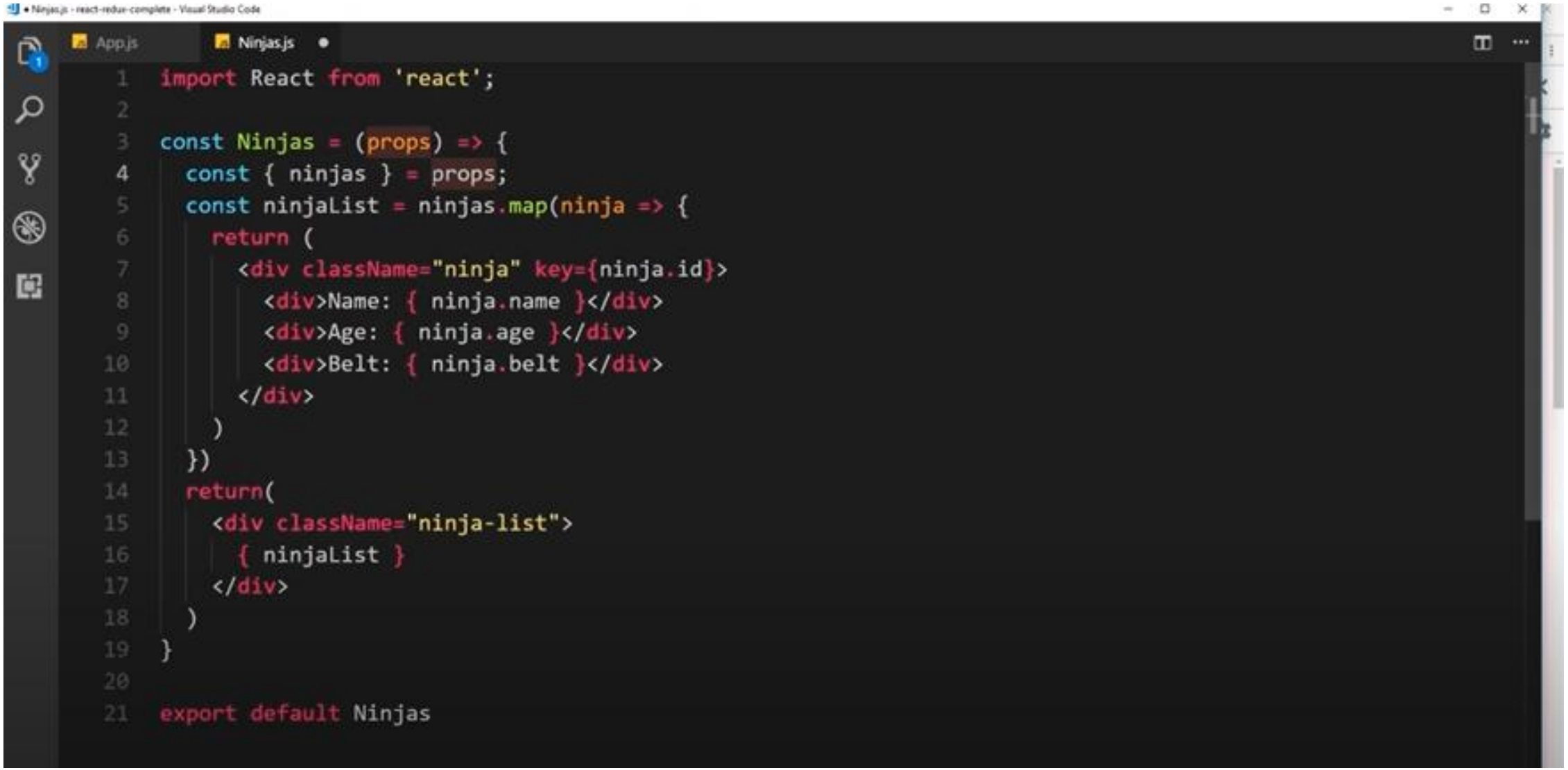
)
})
return(

{ ninjaList }

)
}
}


• Ninjas.js - react-redux-complete - Visual Studio Code
App.js
Ninjas.js
1 import React, { Component } from 'react';
2
3 const Ninjas = () => {
4 const { ninjas } = this.props;
5 const ninjaList = ninjas.map(ninja => {
6 return (
7
8 Name: { ninja.name }
9 Age: { ninja.age }
10 Belt: { ninja.belt }
11
12)
13 })
14 return(
15
16 { ninjaList }
17
18)
19 }
20
21 export default Ninjas


• Ninjas.js - react-redux-complete - Visual Studio Code
App.js
Ninjas.js
1 import React from 'react';
2
3 const Ninjas = (props) => {
4 const { ninjas } = props;
5 const ninjaList = ninjas.map(ninja => {
6 return (
7
8 Name: { ninja.name }
9 Age: { ninja.age }
10 Belt: { ninja.belt }
11
12)
13 })
14 return(
15
16 { ninjaList }
17
18)
19 }
20
21 export default Ninjas


App.js
Ninjas.js
import React from 'react';
const Ninjas = (props) => {
const { ninjas } = props;
const ninjaList = ninjas.map(ninja => {
return (

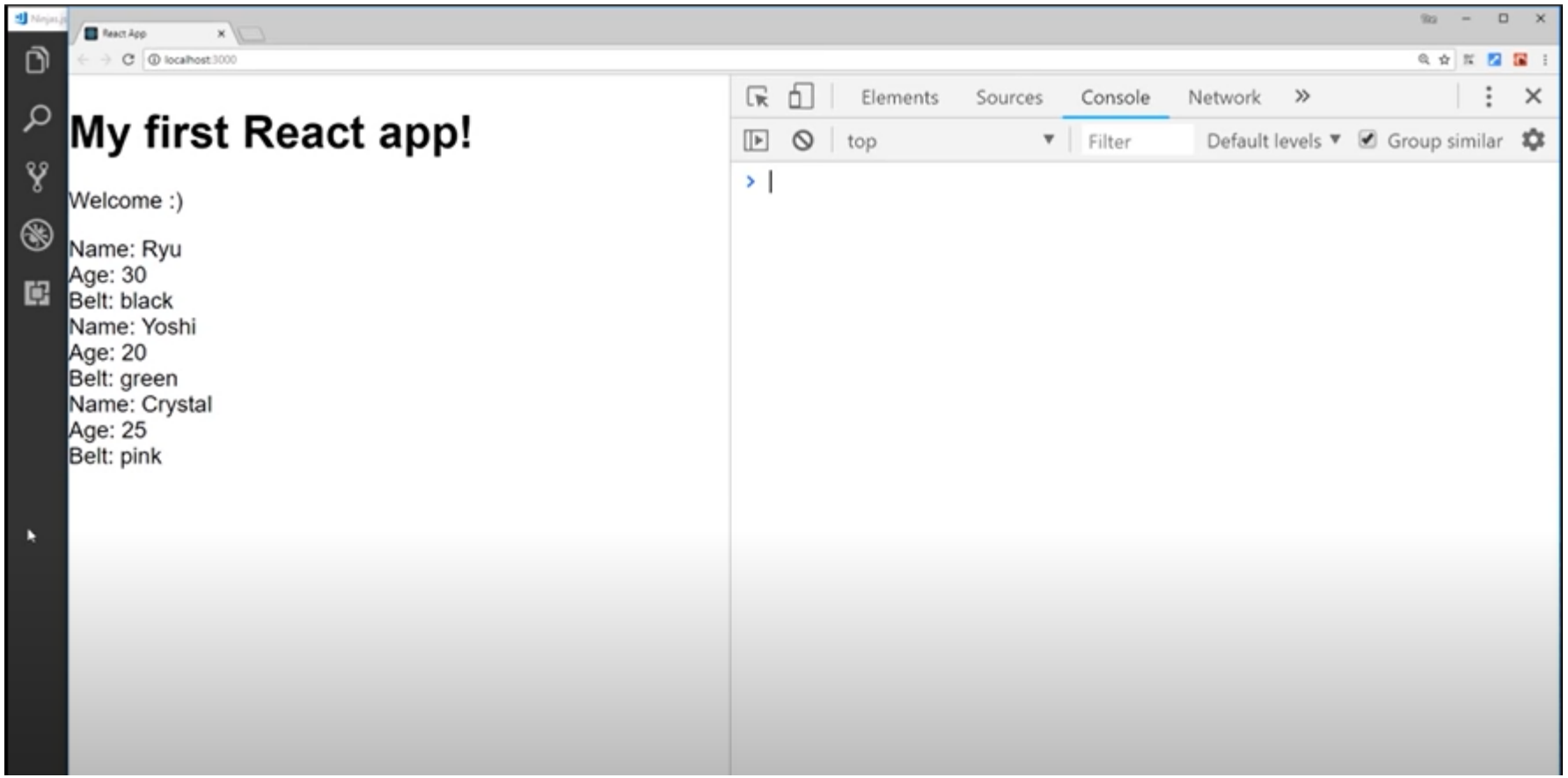
Name: { ninja.name }
Age: { ninja.age }
Belt: { ninja.belt }

)
})
return(

{ ninjaList }

)
}
export default Ninjas

Final One


React App
localhost:3000
My first React app!
Welcome :)
Name: Ryu
Age: 30
Belt: black
Name: Yoshi
Age: 20
Belt: green
Name: Crystal
Age: 25
Belt: pink
Elements
Sources
Console
Network
top
Filter
Default levels
Group similar

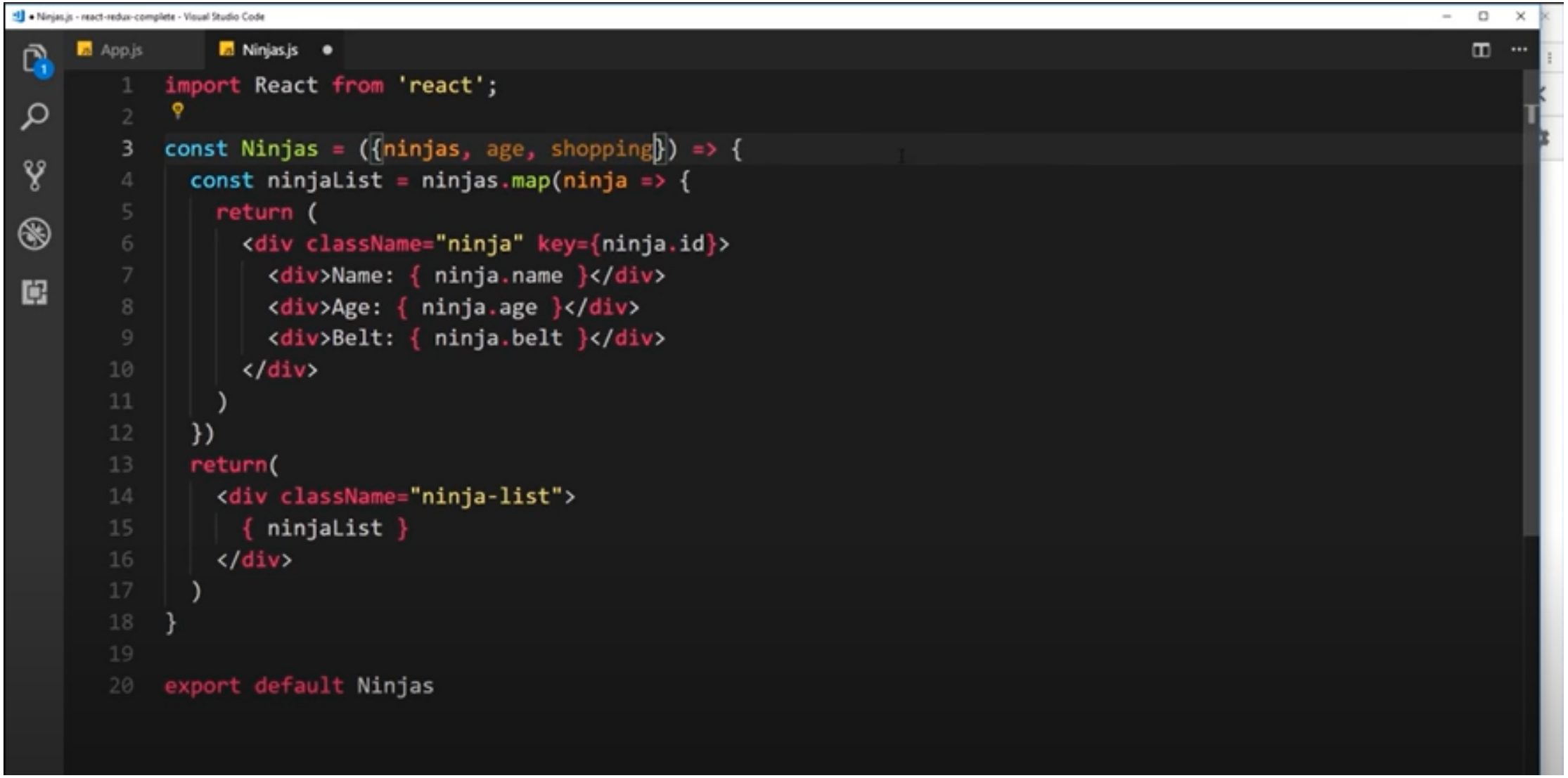

Ninjas.js - react-redux-complete - Visual Studio Code
App.js
Ninjas.js
import React from 'react';
const Ninjas = ({ninjas}) => {
const ninjaList = ninjas.map(ninja => {
return (

Name: { ninja.name }
Age: { ninja.age }
Belt: { ninja.belt }

)
})
return(

{ ninjaList }

)
}
export default Ninjas


Ninjas.js - react-redux-complete - Visual Studio Code
App.js
Ninjas.js
import React from 'react';
const Ninjas = ({ninjas, age, shopping}) => {
const ninjaList = ninjas.map(ninja => {
return (

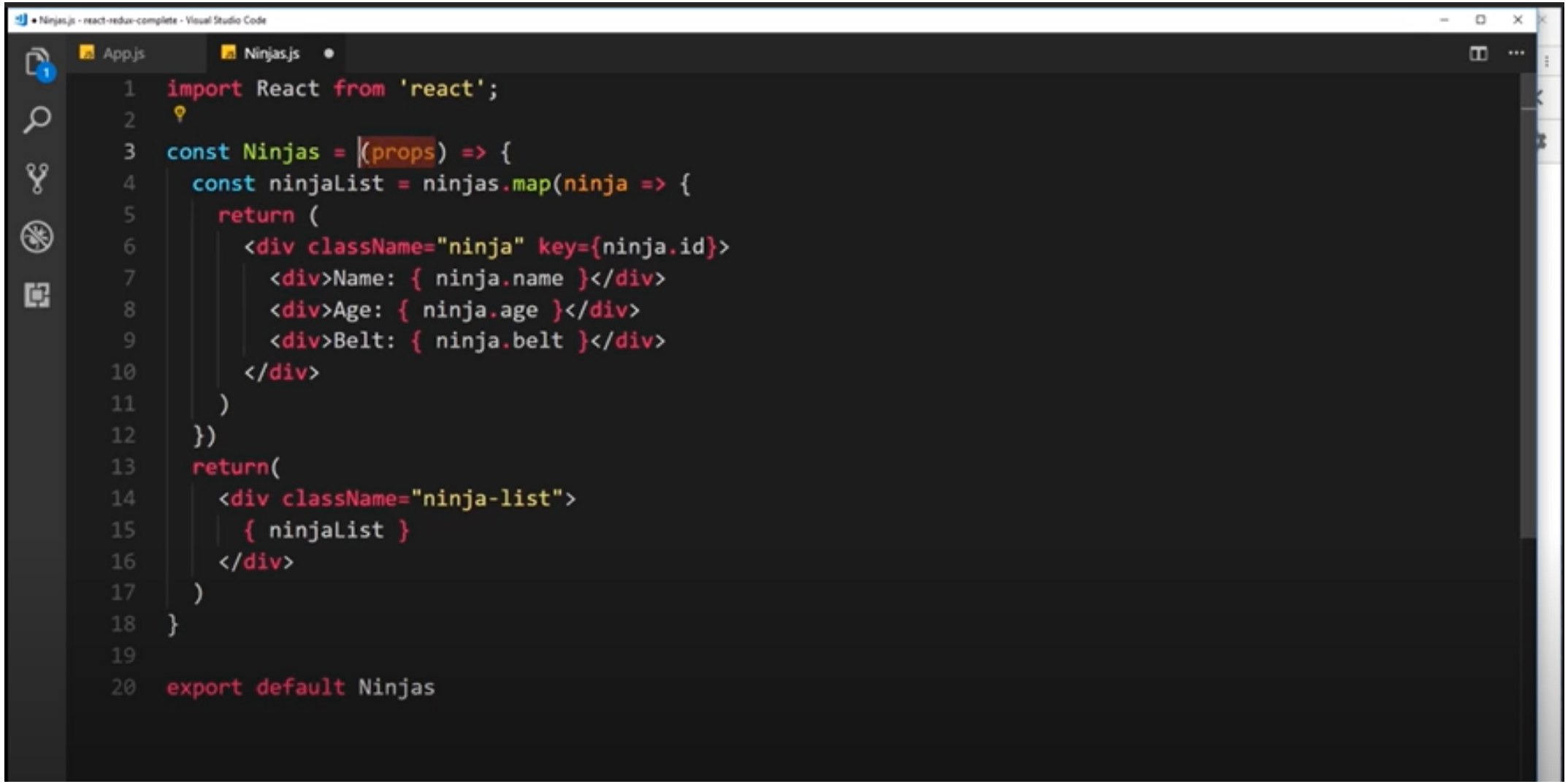
Name: { ninja.name }
Age: { ninja.age }
Belt: { ninja.belt }

)
})
return(

{ ninjaList }

)
}
export default Ninjas

If have many



Also write this