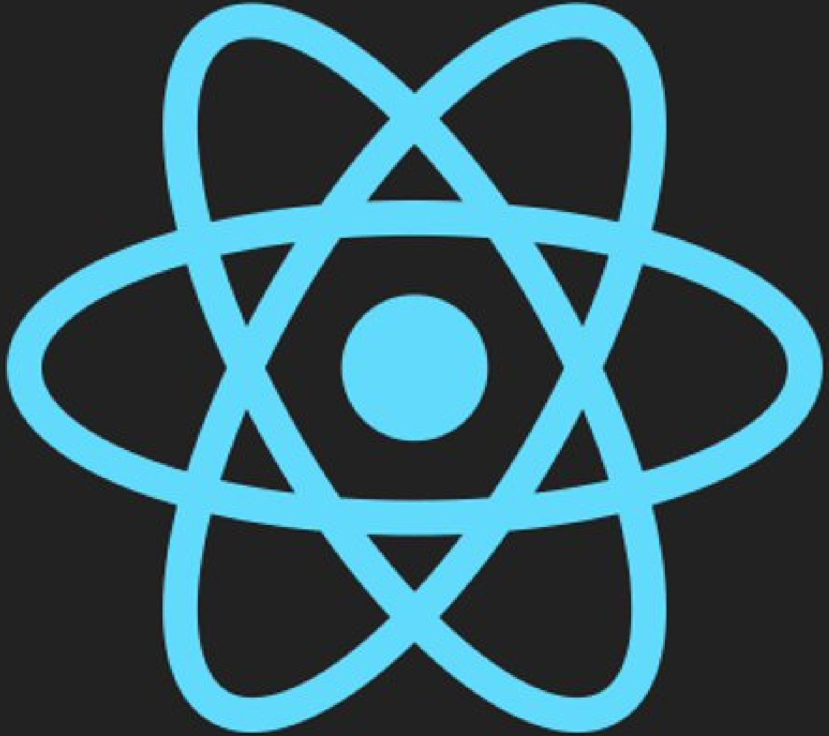
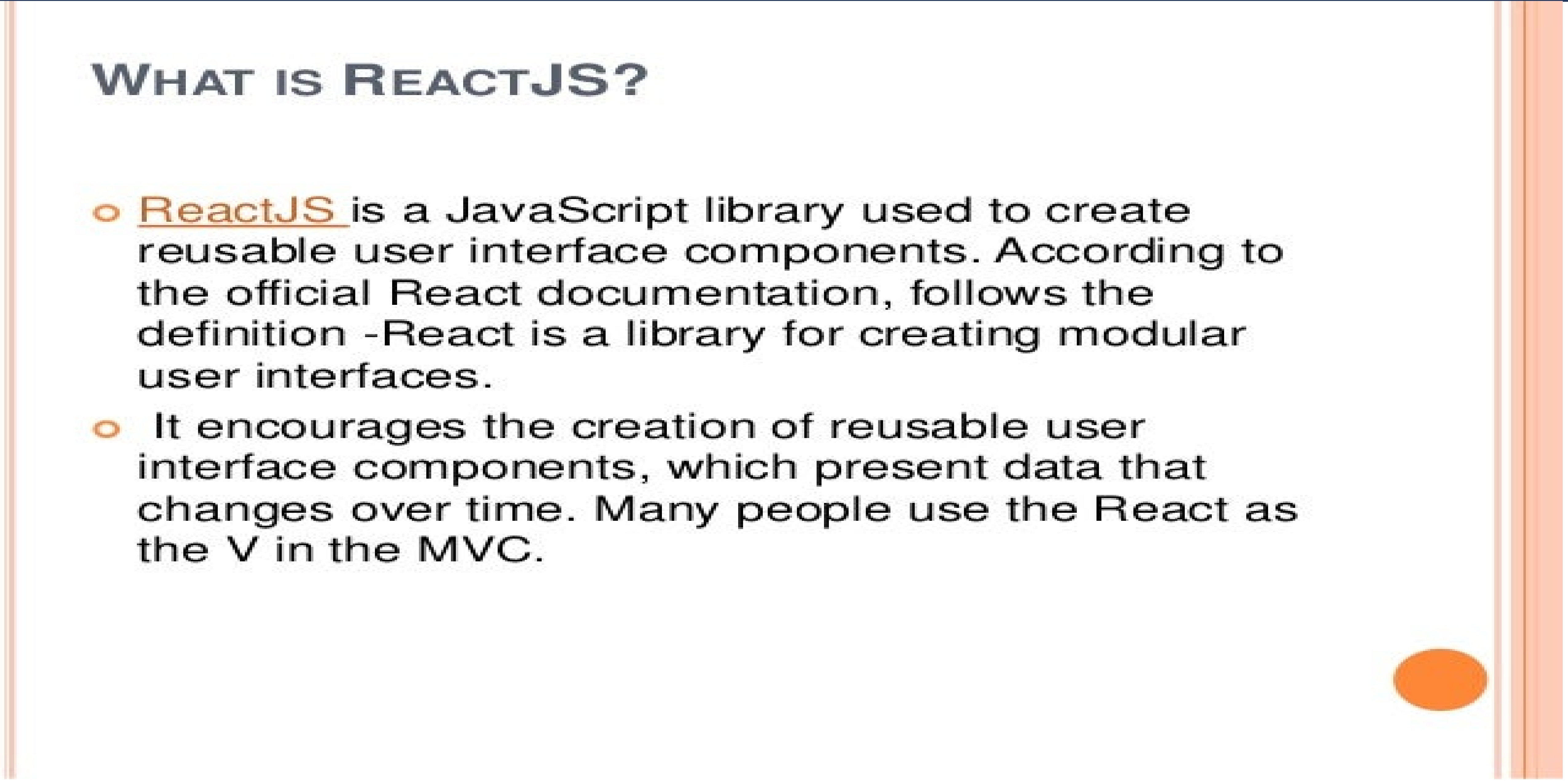
React JS Tutorial



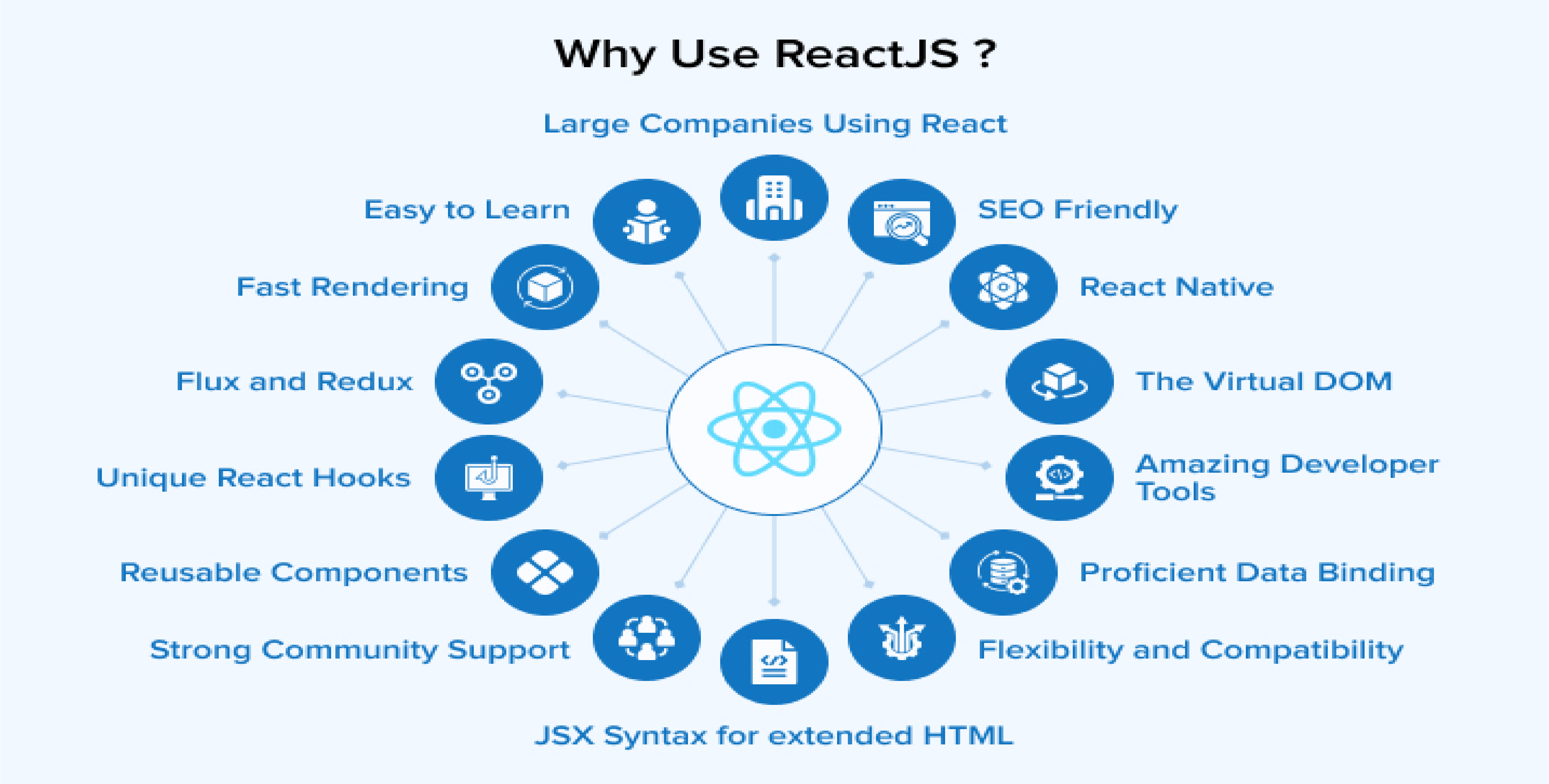
React Js



What is ReactJS?

- ReactJS is a JavaScript library used to create reusable user interface components. According to the official React documentation, follows the definition -React is a library for creating modular user interfaces.
- It encourages the creation of reusable user interface components, which present data that changes over time. Many people use the React as the V in the MVC.

React Js



Why Use ReactJS ?

Large Companies Using React

SEO Friendly

React Native

The Virtual DOM

Amazing Developer Tools

Proficient Data Binding

Flexibility and Compatibility

JSX Syntax for extended HTML

Strong Community Support

Reusable Components

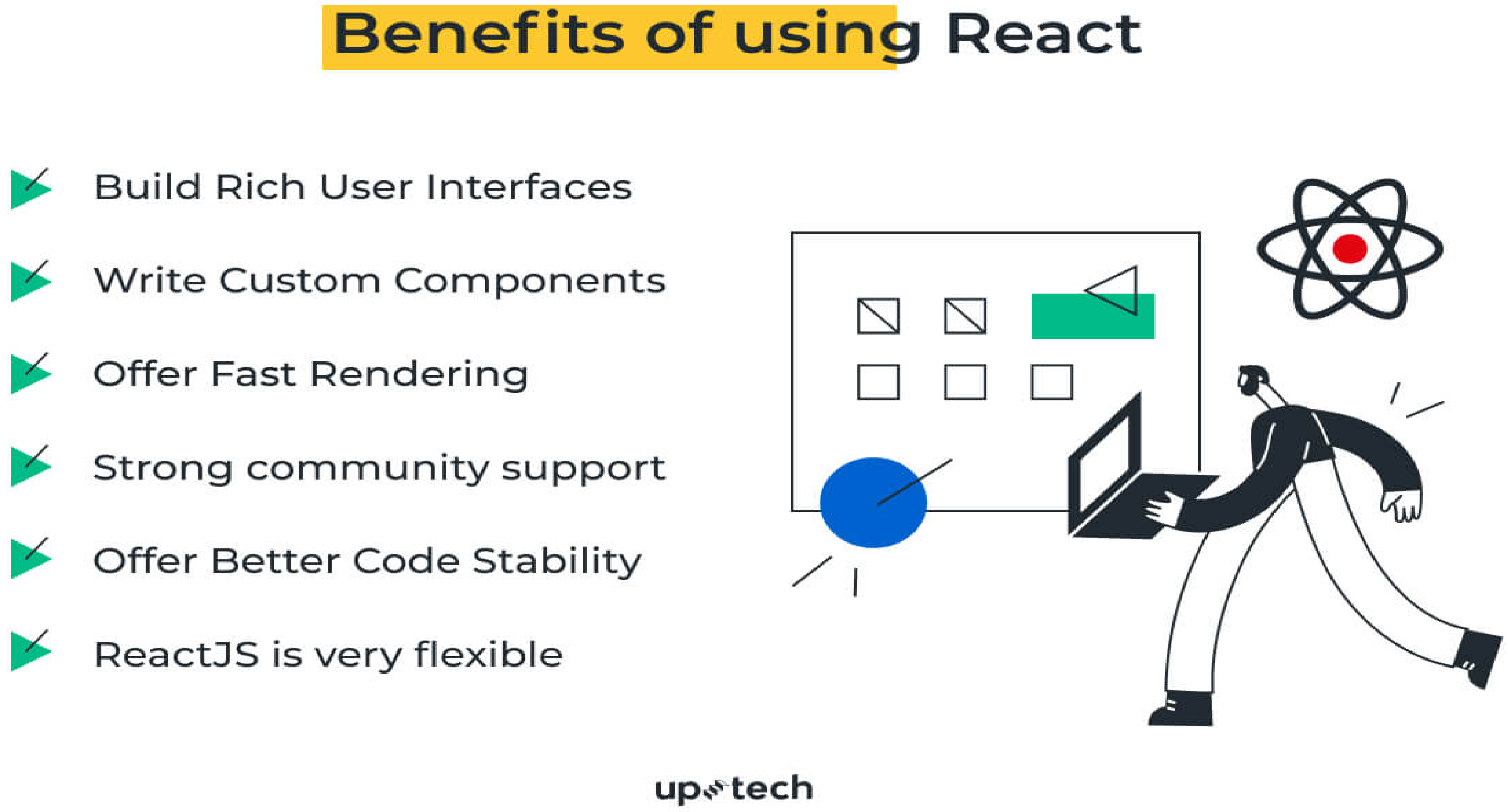
Unique React Hooks

Flux and Redux

Fast Rendering

Easy to Learn

React Js

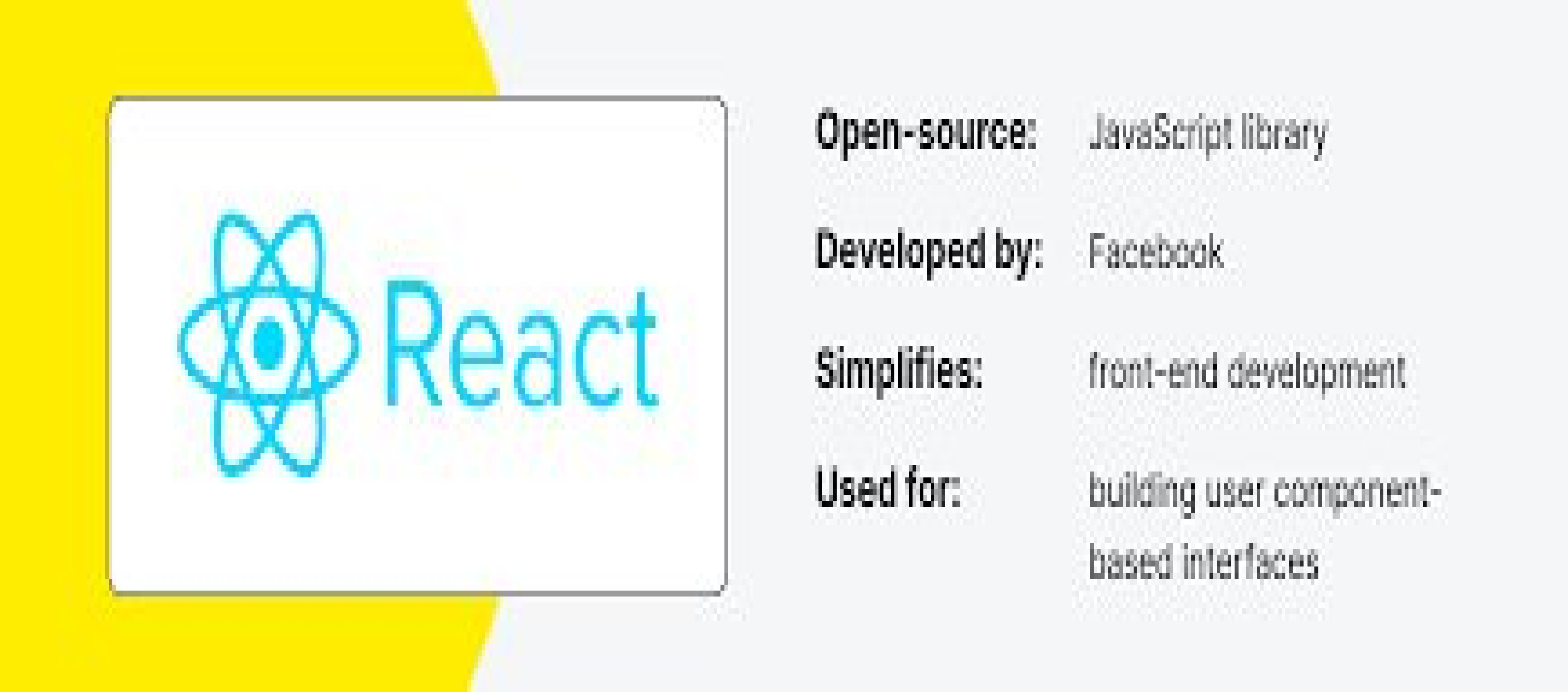


Benefits of using React

- Build Rich User Interfaces
- Write Custom Components
- Offer Fast Rendering
- Strong community support
- Offer Better Code Stability
- ReactJS is very flexible

uptech

React Js



Open-source:	JavaScript library
Developed by:	Facebook
Simplifies:	front-end development
Used for:	building user component-based interfaces

React Js

what you will learn?

- Class and function components
- JSX
- Props and state
- Component Life Cycle
- React Router
- React Forms
- Redux
- React Redux Form
- Server Side Communication

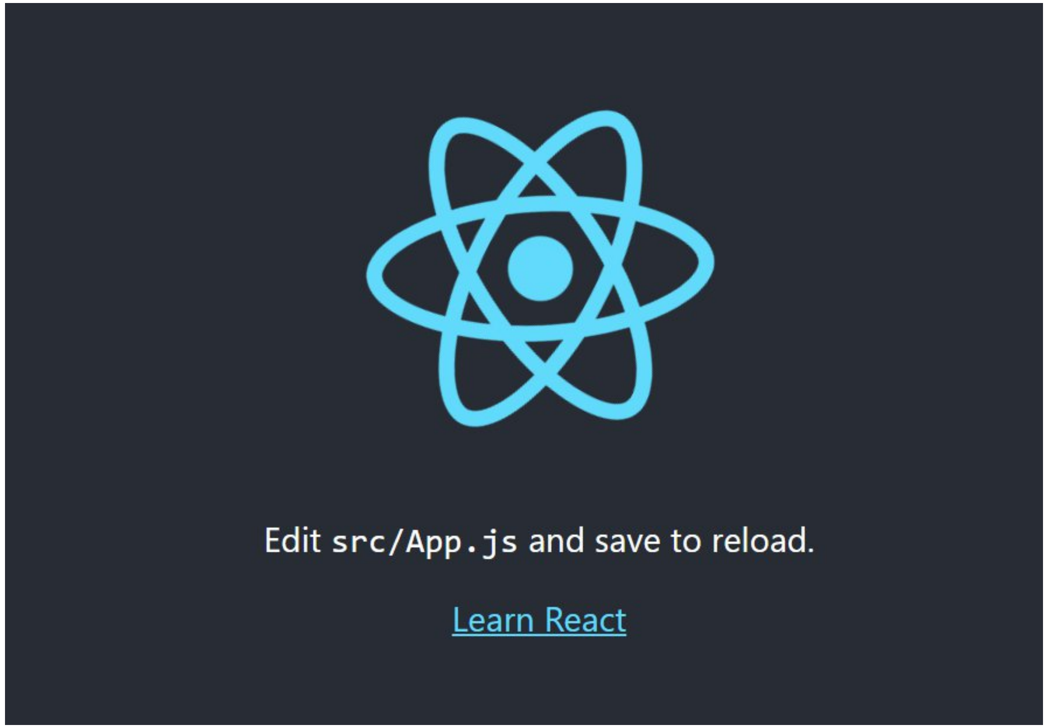
React Js

Install Create-React-App

1. Install nodejs
2. Check node –v
3. Create a folder as react
4. Open with VS code
5. https://create-react-app.dev/docs/getting-started/

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

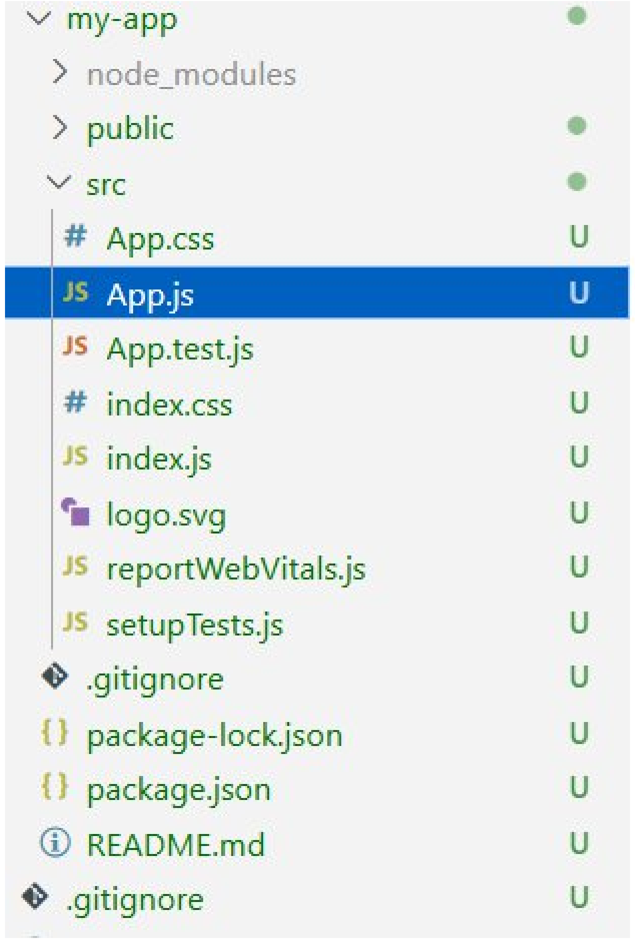
```
git remote add origin https://github.com/your-user-id/react.git
git branch -M main
git push -u origin main
```



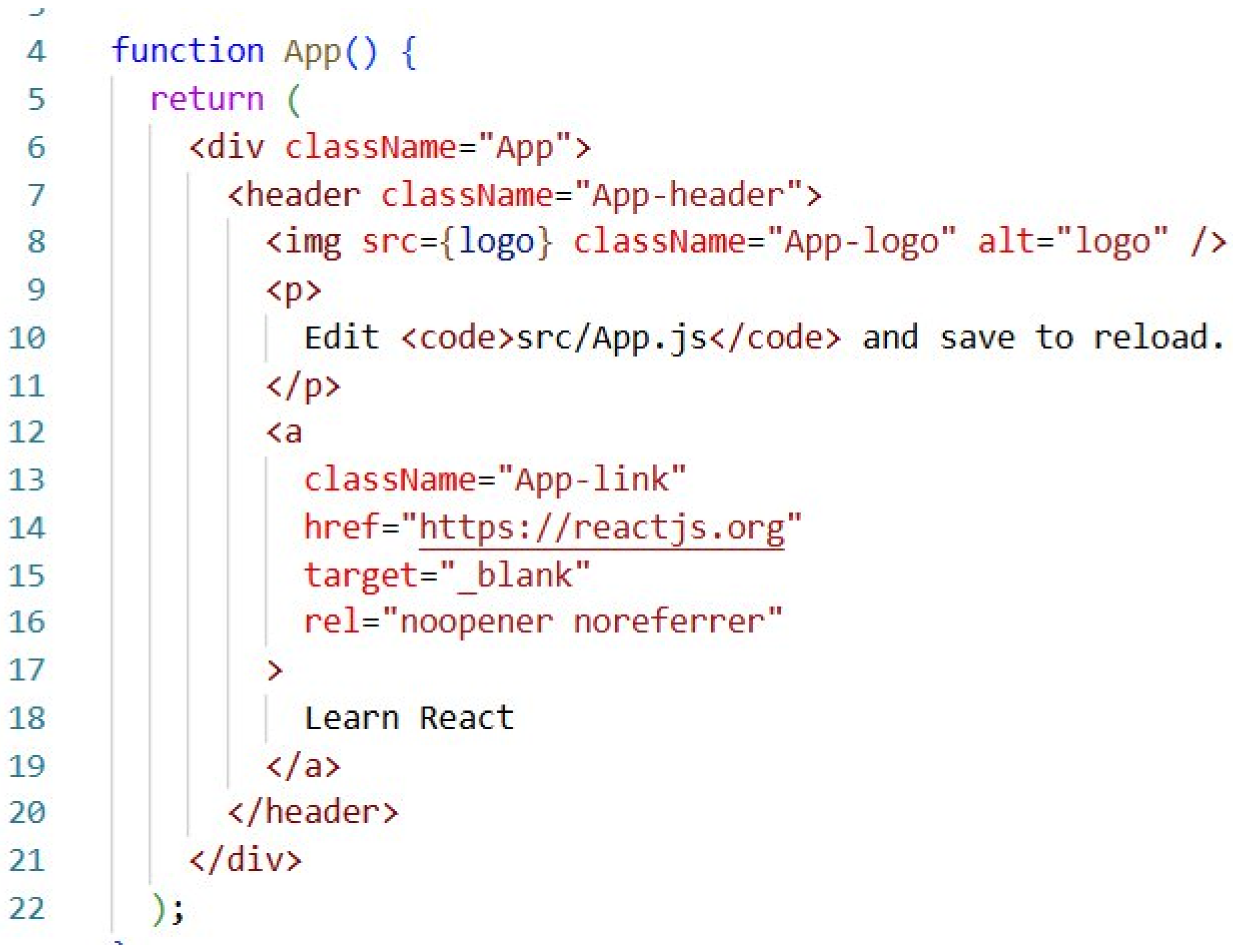
Edit `src/App.js` and save to reload.

Learn React

React Js

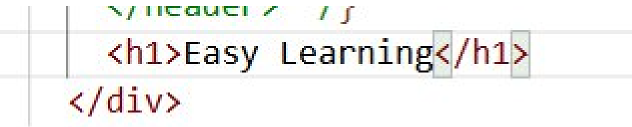


```
⌄ my-app                  ●
  › node_modules
  › public                ●
  ⌄ src                   ●
    # App.css             U
    JS App.js             U
    JS App.test.js        U
    # index.css           U
    JS index.js           U
    logo.svg              U
    JS reportWebVitals.js U
    JS setupTests.js      U
  .gitignore              U
  {} package-lock.json    U
  {} package.json         U
  ⓘ README.md             U
.gitignore                U
```

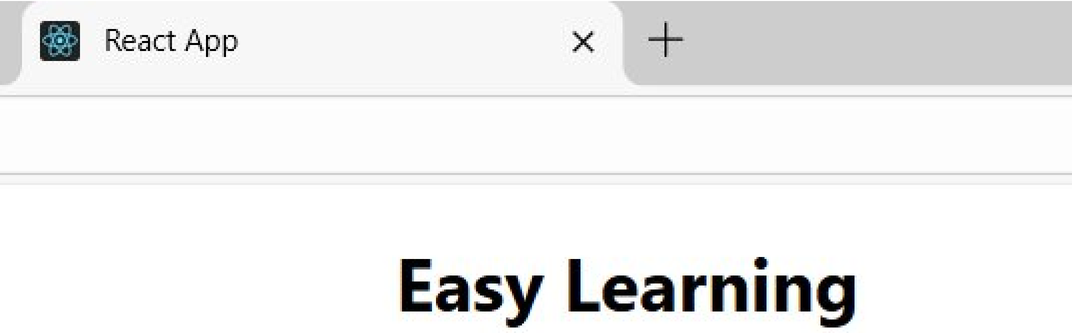


```
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
```

React Js

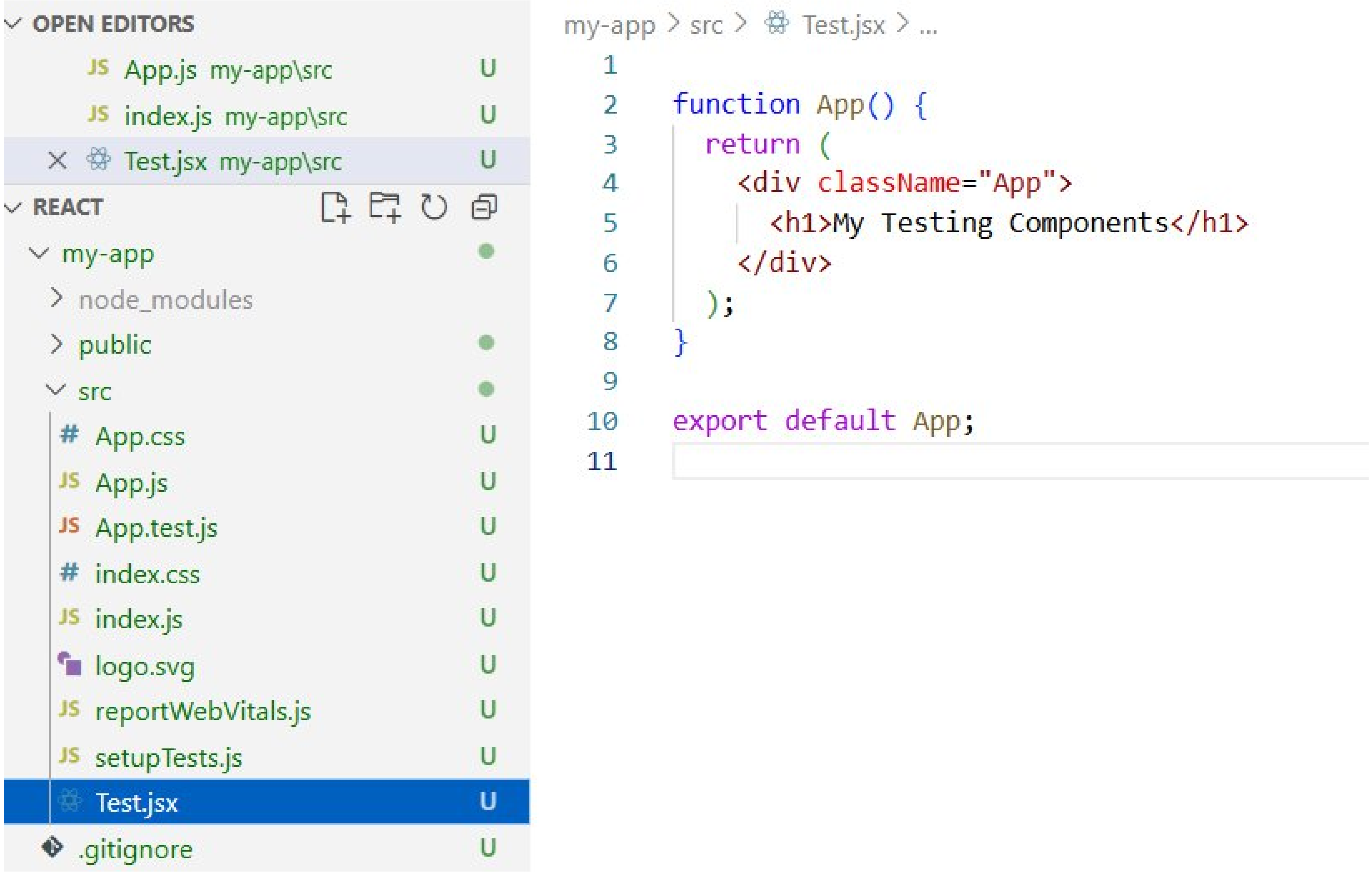


```
  </header> */}
  <h1>Easy Learning</h1>
</div>
```



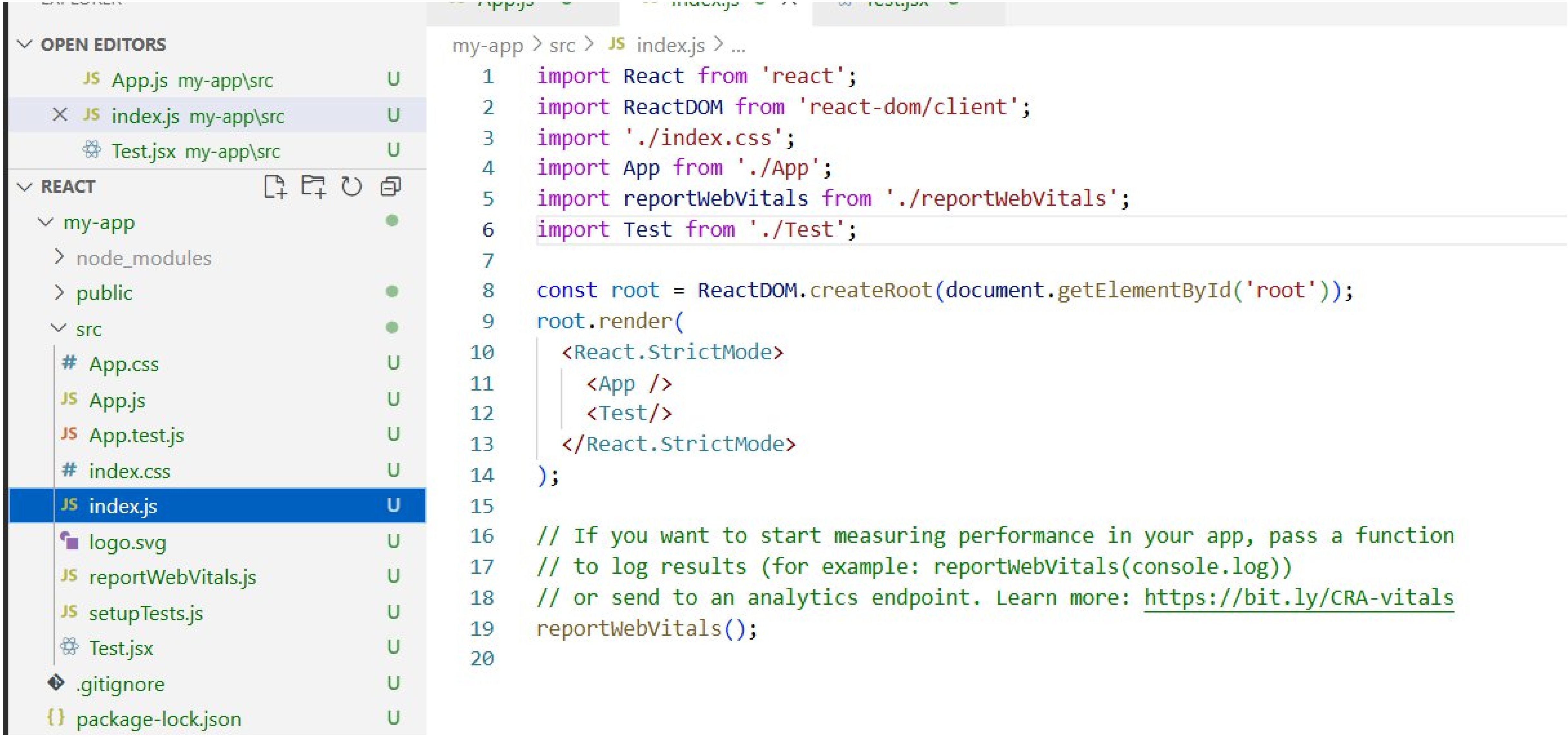
Easy Learning

React Component Basic



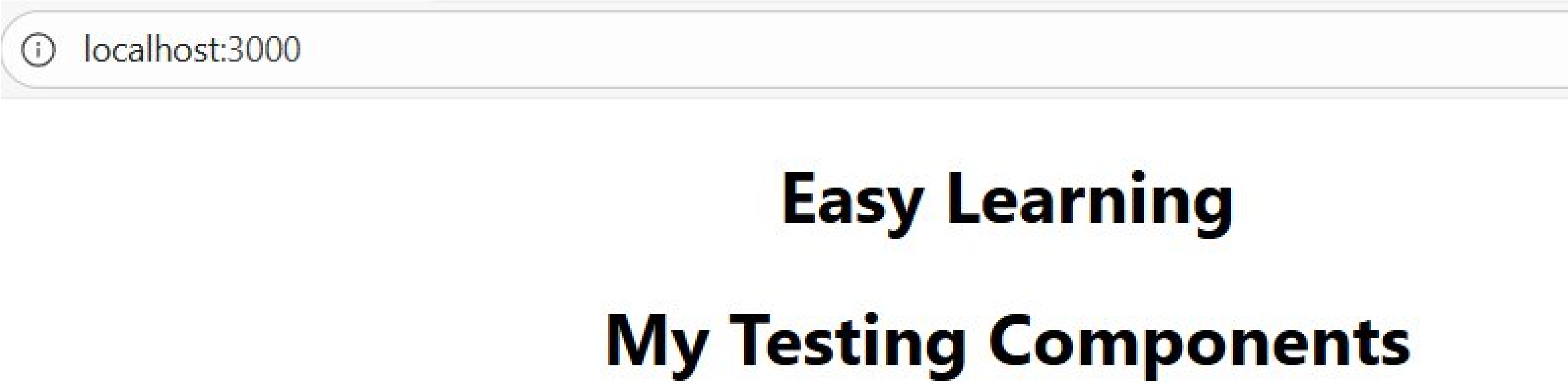
```

function App() {
  return (
    <div className="App">
      <h1>My Testing Components</h1>
    </div>
  );
}

export default App;

```

React Component Basic



```javascript
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import Test from './Test';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
    <Test/>
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

React Component Basic

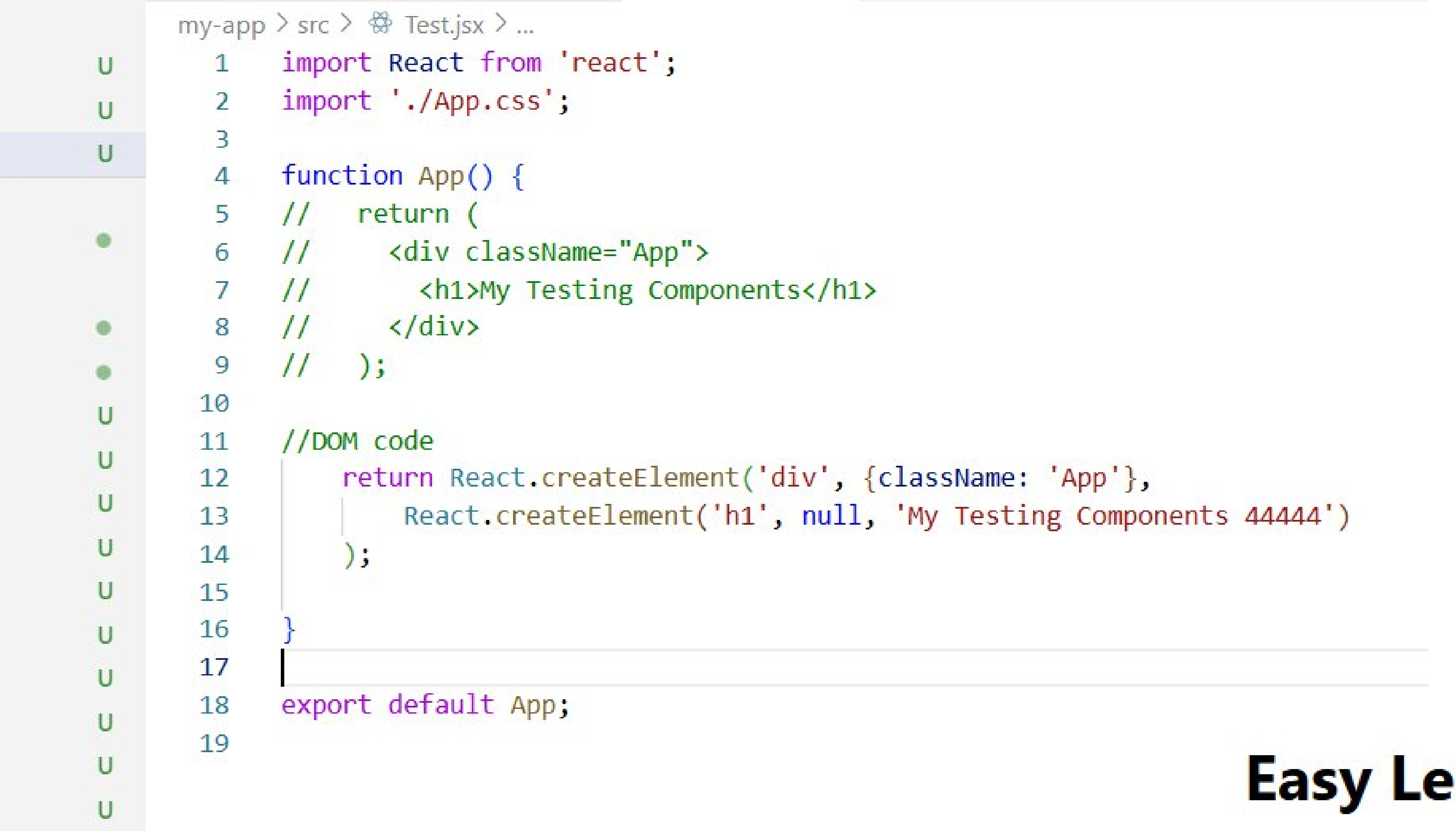


localhost:3000

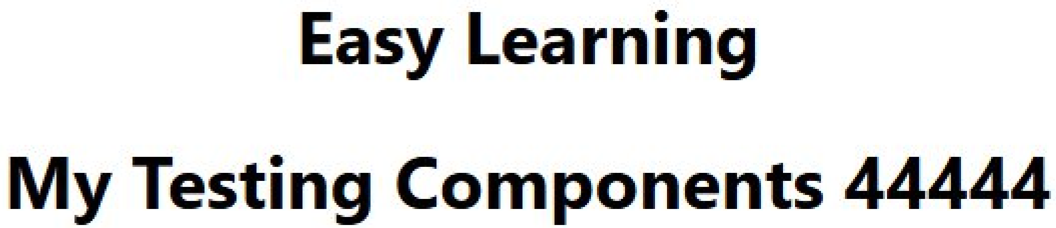
Easy Learning

My Testing Components

JSX syntax



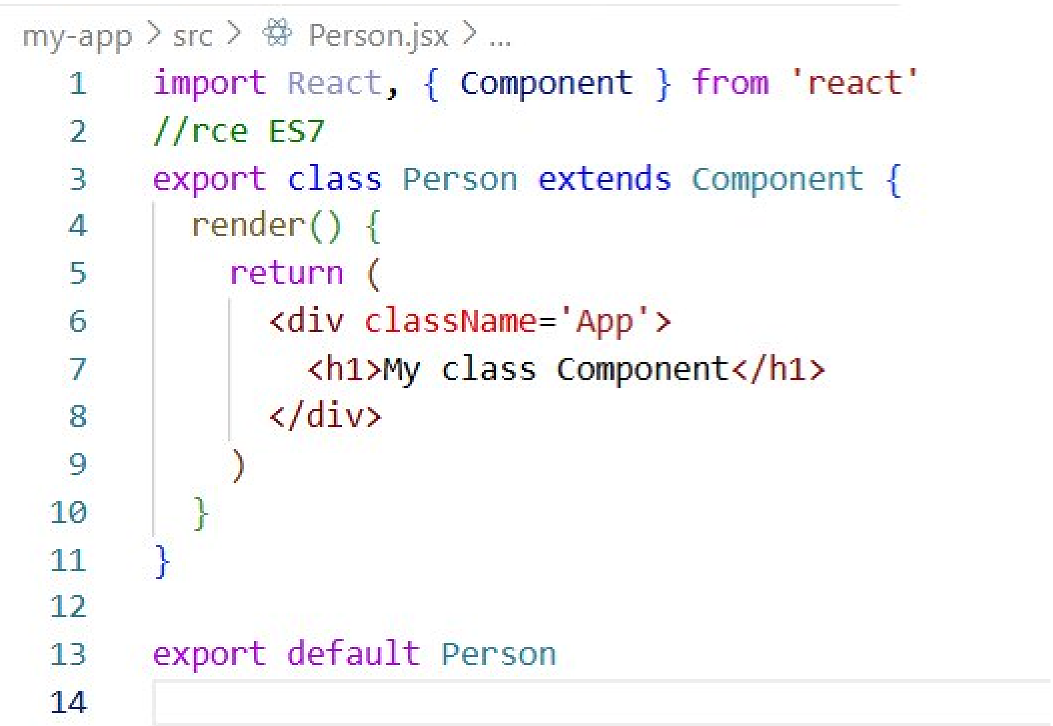
```jsx
import React from 'react';
import './App.css';

function App() {
//   return (
//     <div className="App">
//       <h1>My Testing Components</h1>
//     </div>
//   );

//DOM code
    return React.createElement('div', {className: 'App'},
        React.createElement('h1', null, 'My Testing Components 44444')
    );

}

export default App;
```



Easy Learning

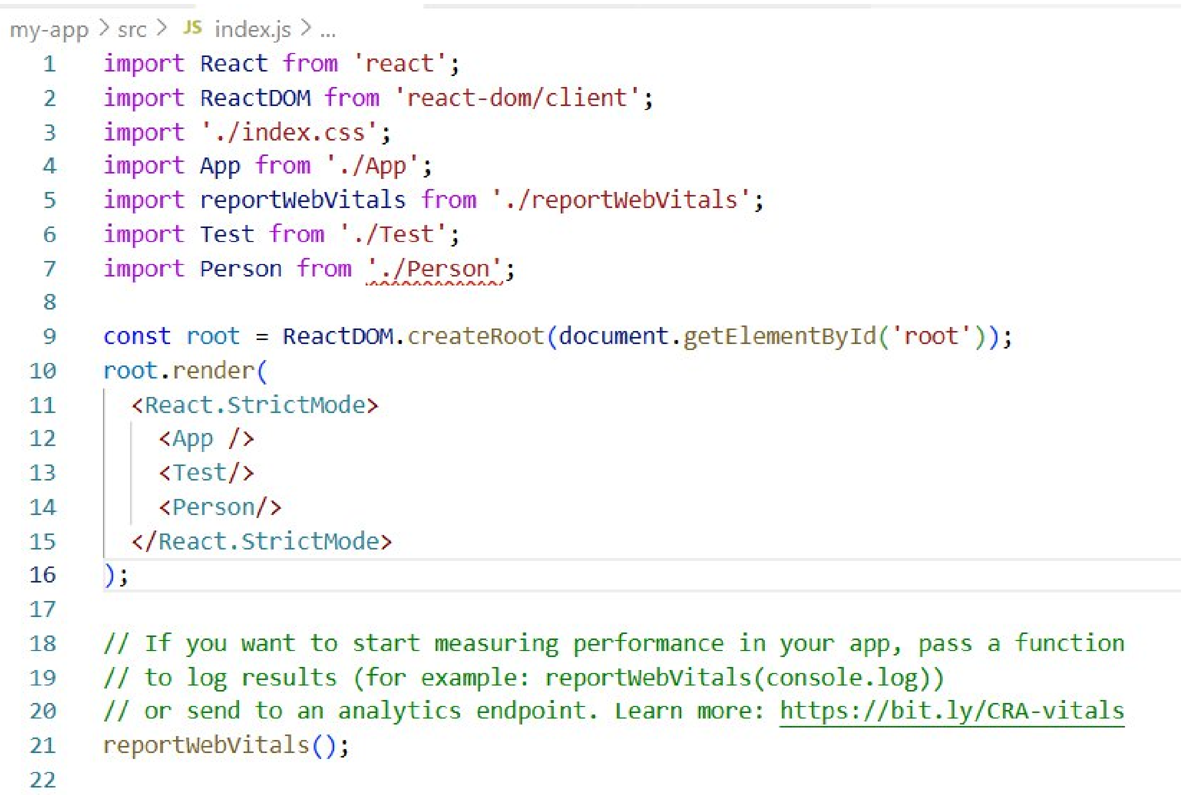
My Testing Components 44444

React Js- Class components



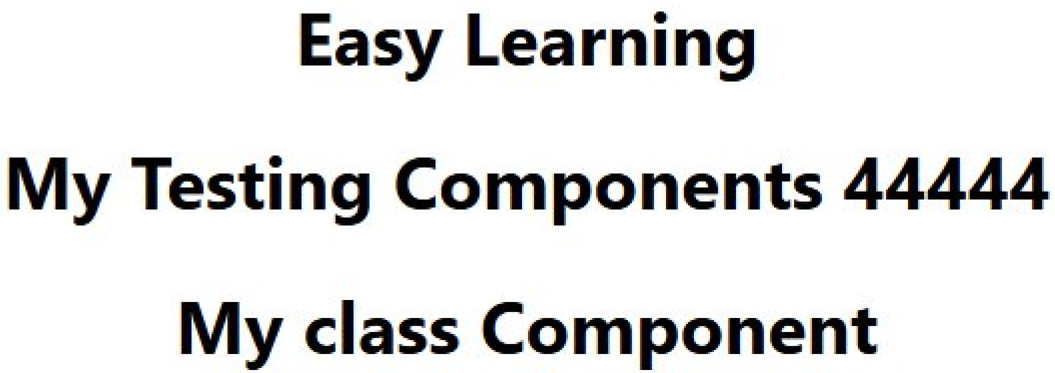
my-app > src > Person.jsx > ...

```
import React, { Component } from 'react'
//rce ES7
export class Person extends Component {
  render() {
    return (
      <div className='App'>
        <h1>My class Component</h1>
      </div>
    )
  }
}

export default Person

```



my-app > src > JS index.js > ...

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import Test from './Test';
import Person from './Person';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
    <Test/>
    <Person/>
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();

```

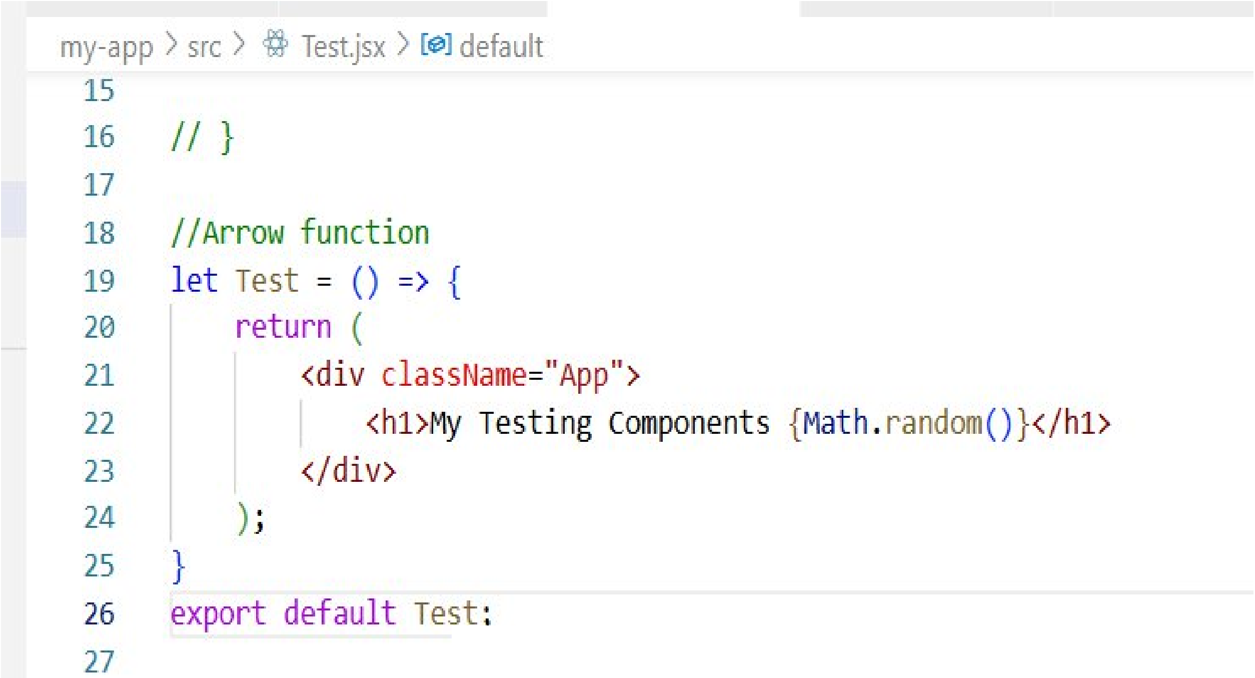


Easy Learning

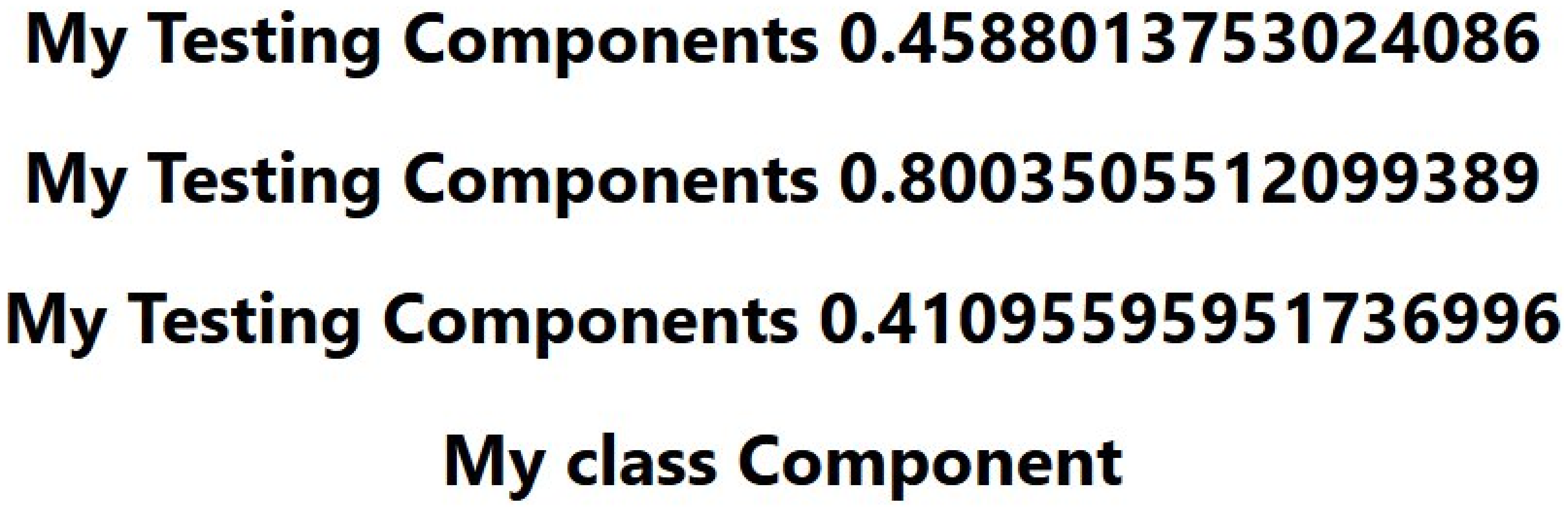
My Testing Components 44444

My class Component

React Js- Dynamic functional Component

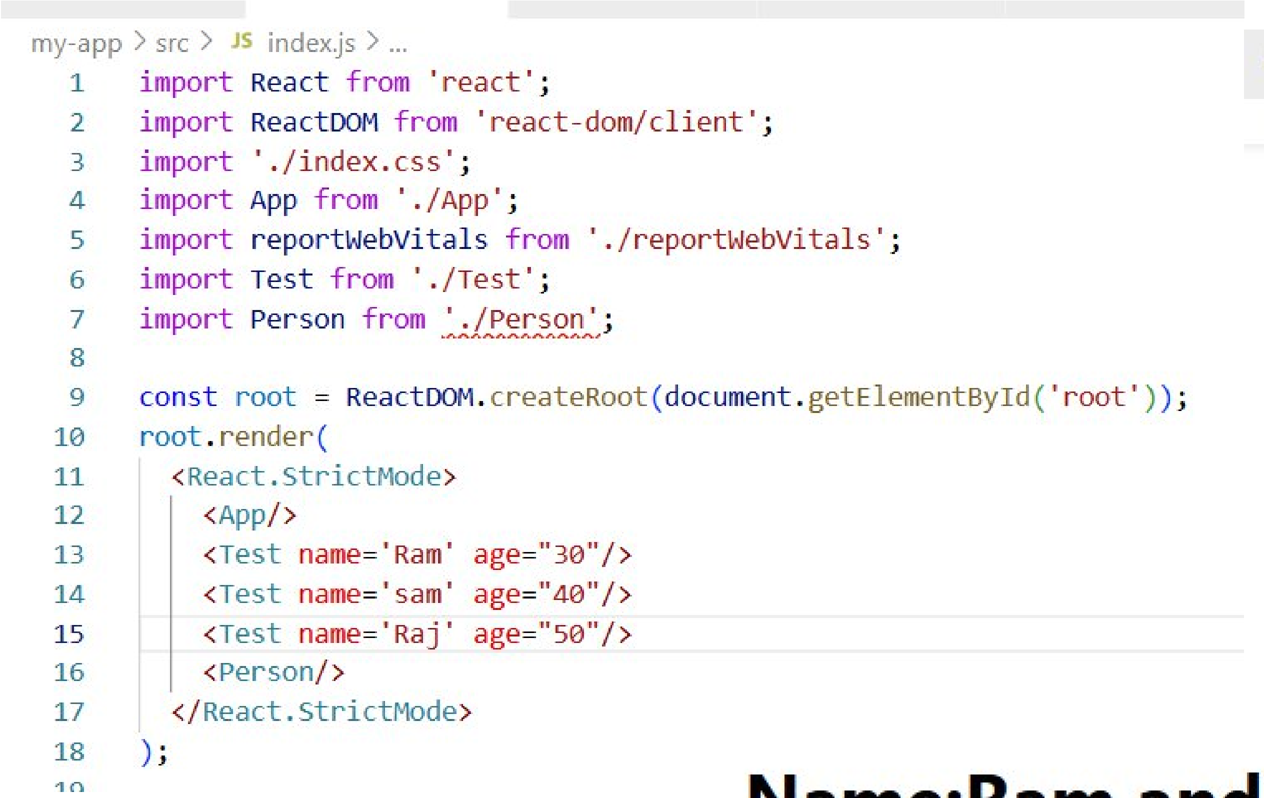


```

// }

//Arrow function
let Test = () => {
    return (
        <div className="App">
            <h1>My Testing Components {Math.random()}</h1>
        </div>
    );
}
export default Test;

```

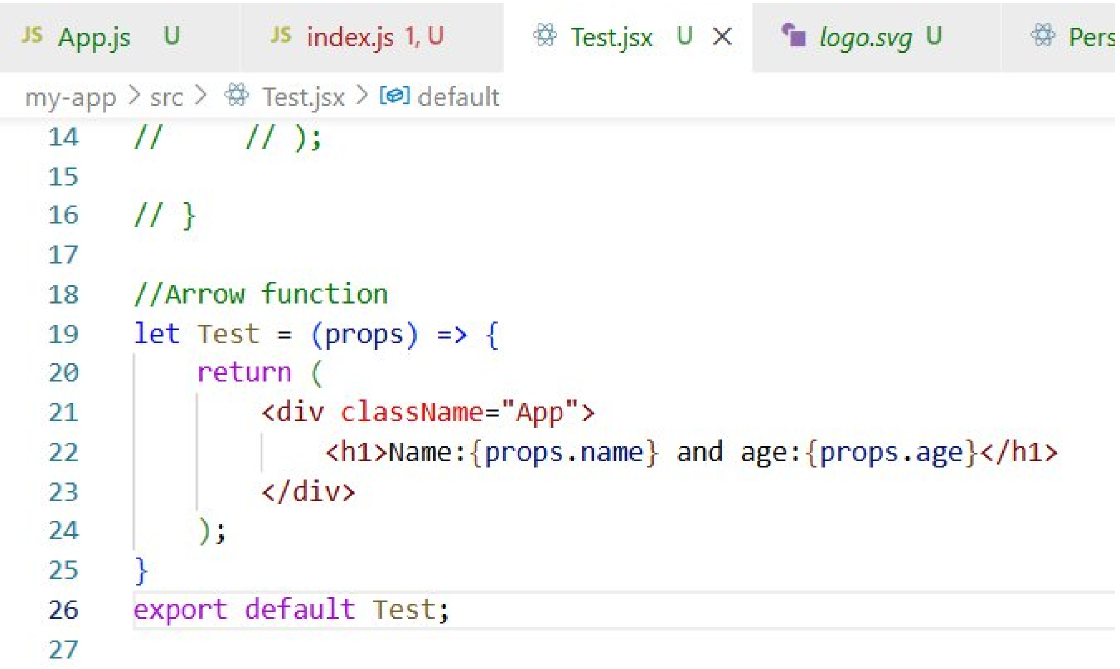


```
My Testing Components 0.4588013753024086

My Testing Components 0.8003505512099389

My Testing Components 0.41095595951736996

My class Component
```

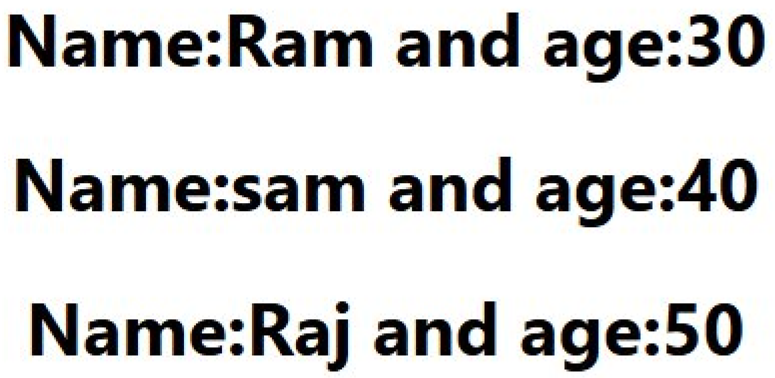
React Js



```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import Test from './Test';
import Person from './Person';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App/>
    <Test name='Ram' age="30"/>
    <Test name='sam' age="40"/>
    <Test name='Raj' age="50"/>
    <Person/>
  </React.StrictMode>
);
```



```
//     // );

// }

//Arrow function
let Test = (props) => {
    return (
        <div className="App">
            <h1>Name:{props.name} and age:{props.age}</h1>
        </div>
    );
}
export default Test;

```

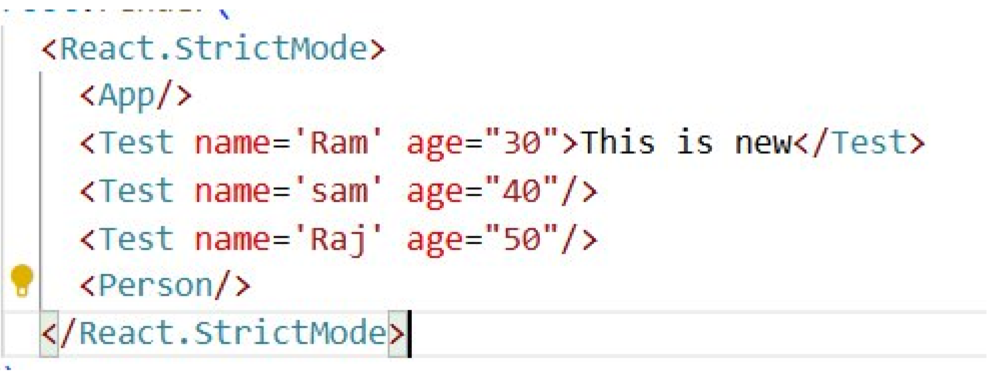


Name:Ram and age:30

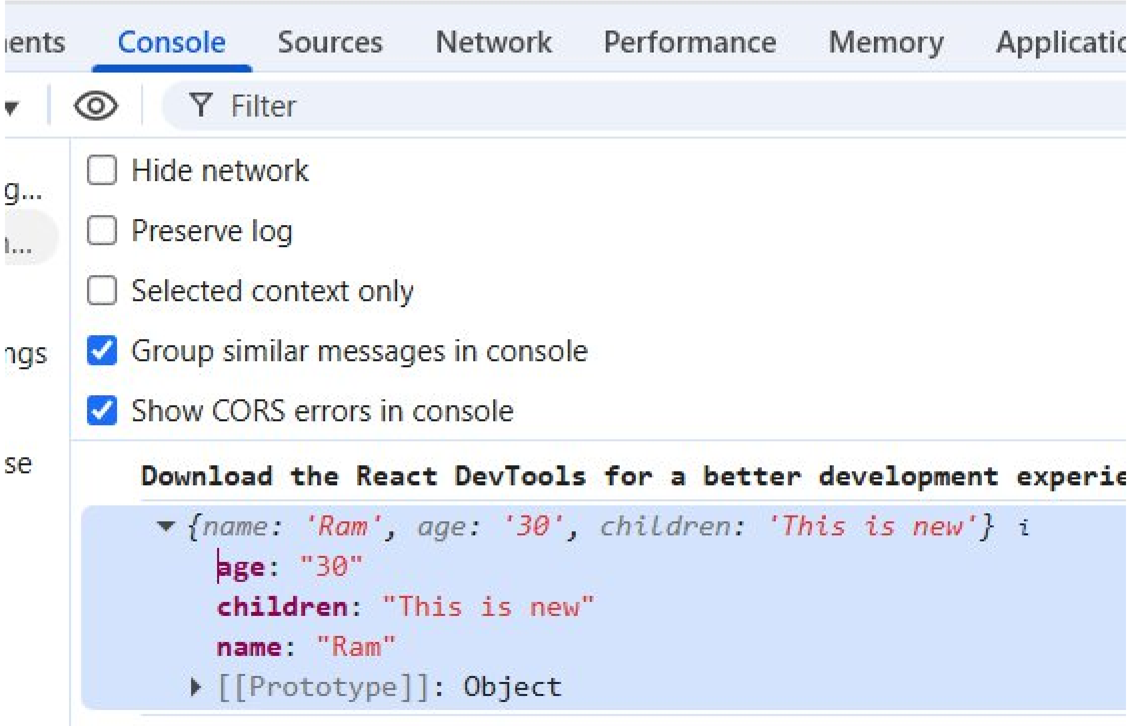
Name:sam and age:40

Name:Raj and age:50

React Js

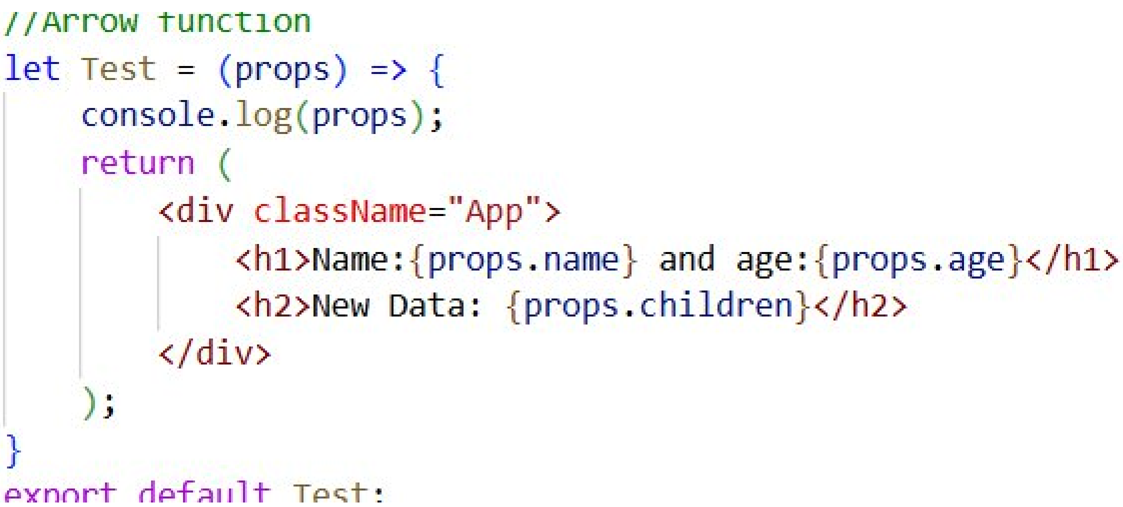


```
  <React.StrictMode>
    <App/>
    <Test name='Ram' age="30">This is new</Test>
    <Test name='sam' age="40"/>
    <Test name='Raj' age="50"/>
    <Person/>
  </React.StrictMode>
```

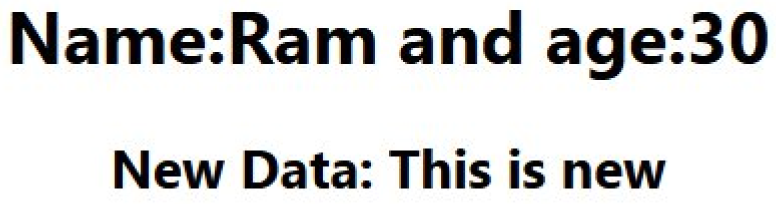


☐ Hide network
☐ Preserve log
☐ Selected context only
☑ Group similar messages in console
☑ Show CORS errors in console

```
Download the React DevTools for a better development experie
▼ {name: 'Ram', age: '30', children: 'This is new'} i
    age: "30"
    children: "This is new"
    name: "Ram"
  ▶ [[Prototype]]: Object
```



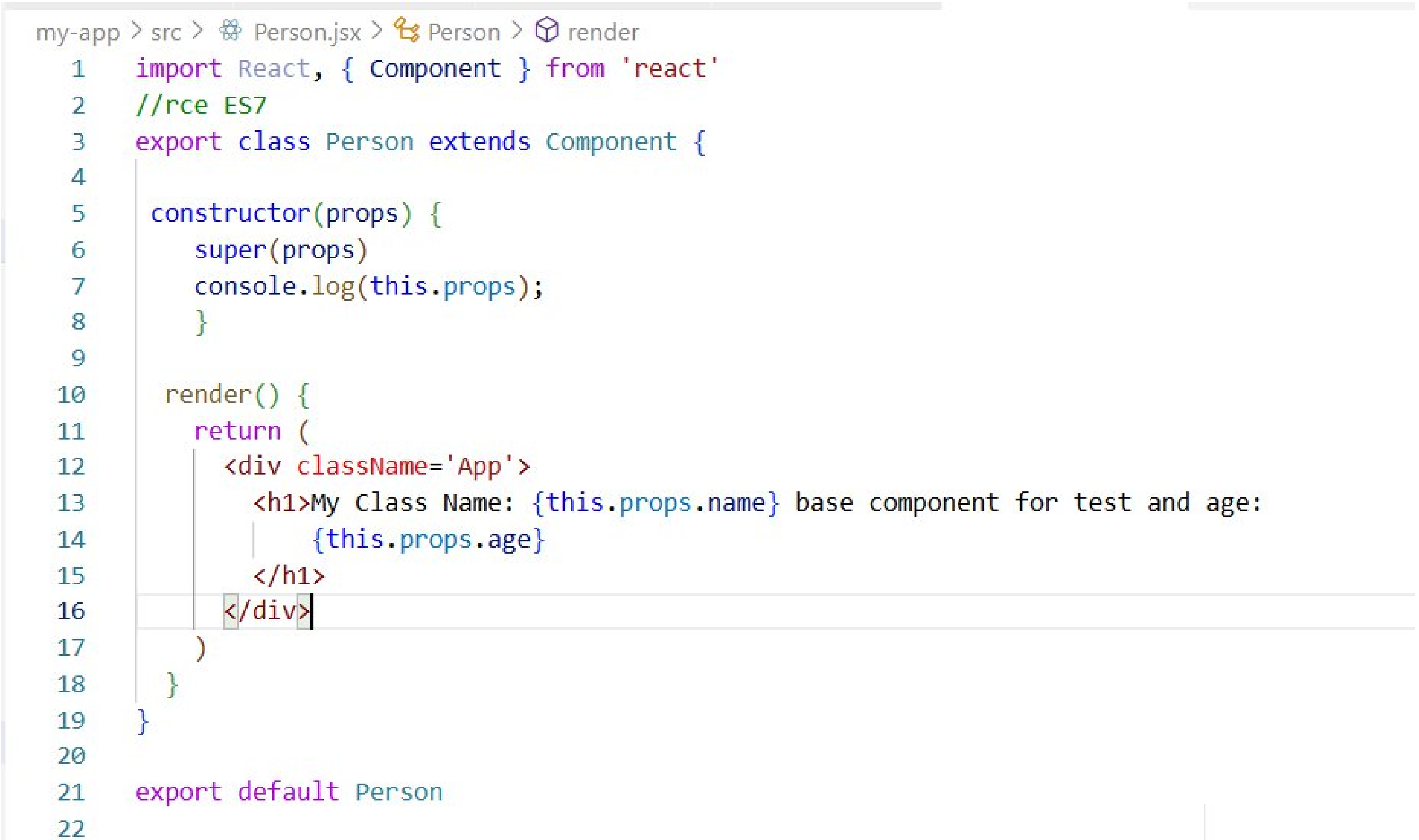
```
//Arrow function
let Test = (props) => {
    console.log(props);
    return (
        <div className="App">
            <h1>Name:{props.name} and age:{props.age}</h1>
            <h2>New Data: {props.children}</h2>
        </div>
    );
}
export default Test;
```



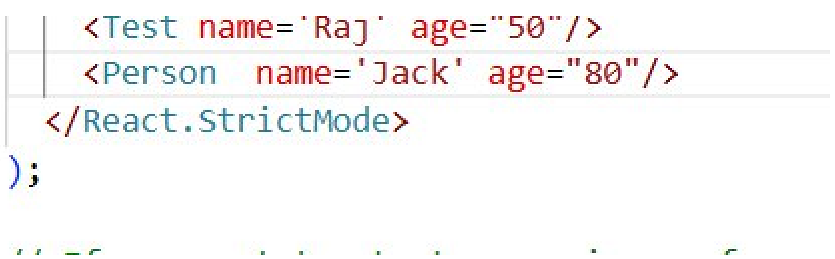
Name:Ram and age:30

New Data: This is new

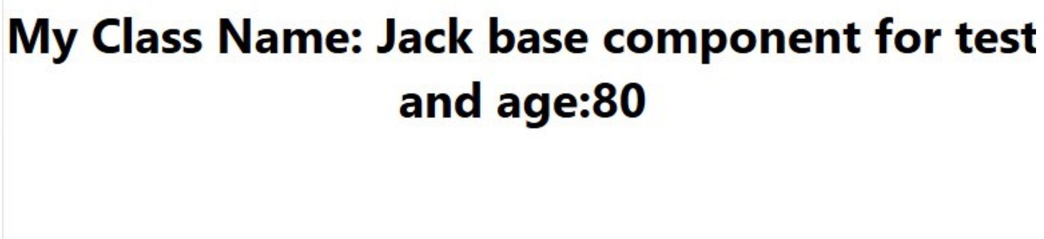
React Js – class based



```jsx
import React, { Component } from 'react'
//rce ES7
export class Person extends Component {

 constructor(props) {
    super(props)
    console.log(this.props);
    }

  render() {
    return (
      <div className='App'>
        <h1>My Class Name: {this.props.name} base component for test and age:
            {this.props.age}
        </h1>
      </div>
    )
  }
}

export default Person

```

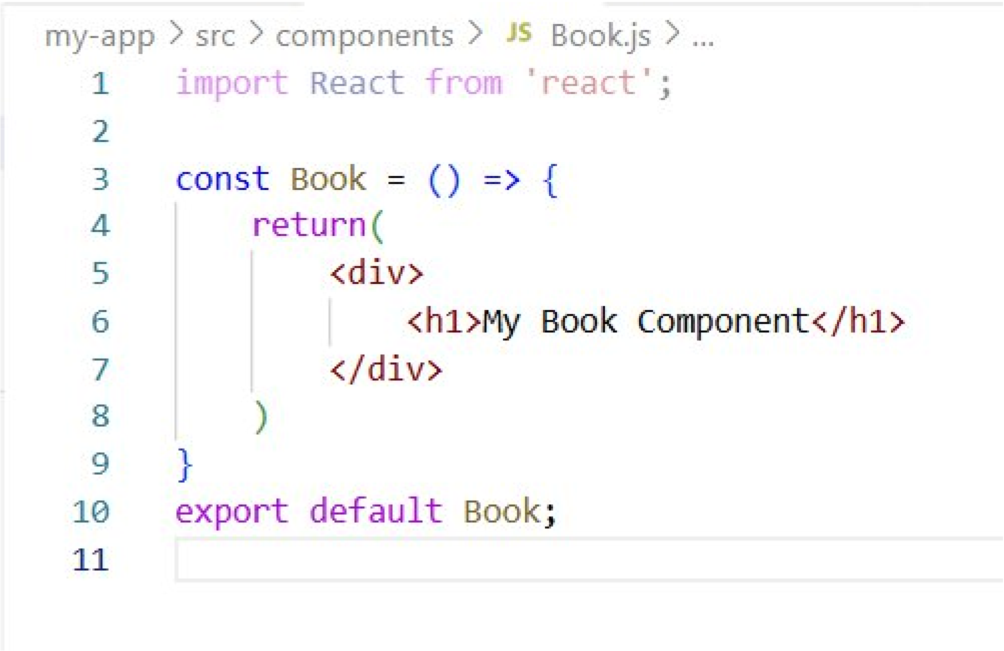


```jsx
    <Test name='Raj' age="50"/>
    <Person  name='Jack' age="80"/>
  </React.StrictMode>
);
```



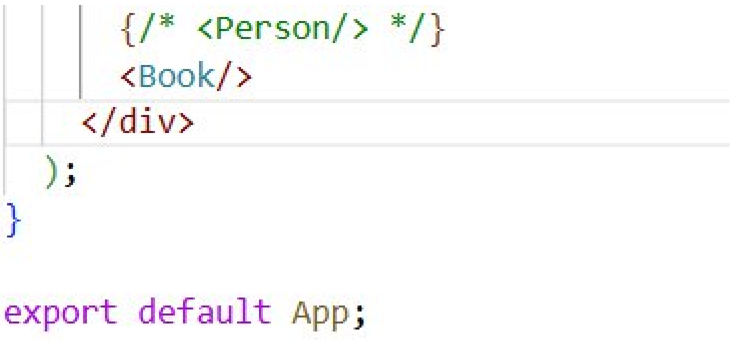
My Class Name: Jack base component for test and age:80

React Js- new component

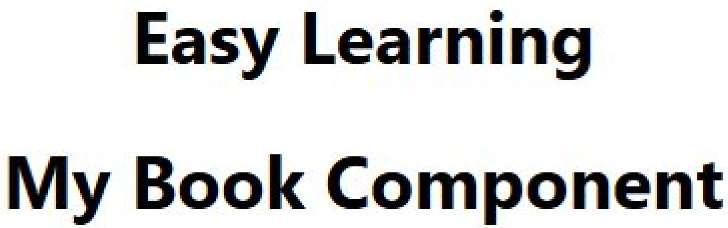


my-app > src > components > JS Book.js > ...

```
import React from 'react';

const Book = () => {
    return(
        <div>
            <h1>My Book Component</h1>
        </div>
    )
}
export default Book;

```



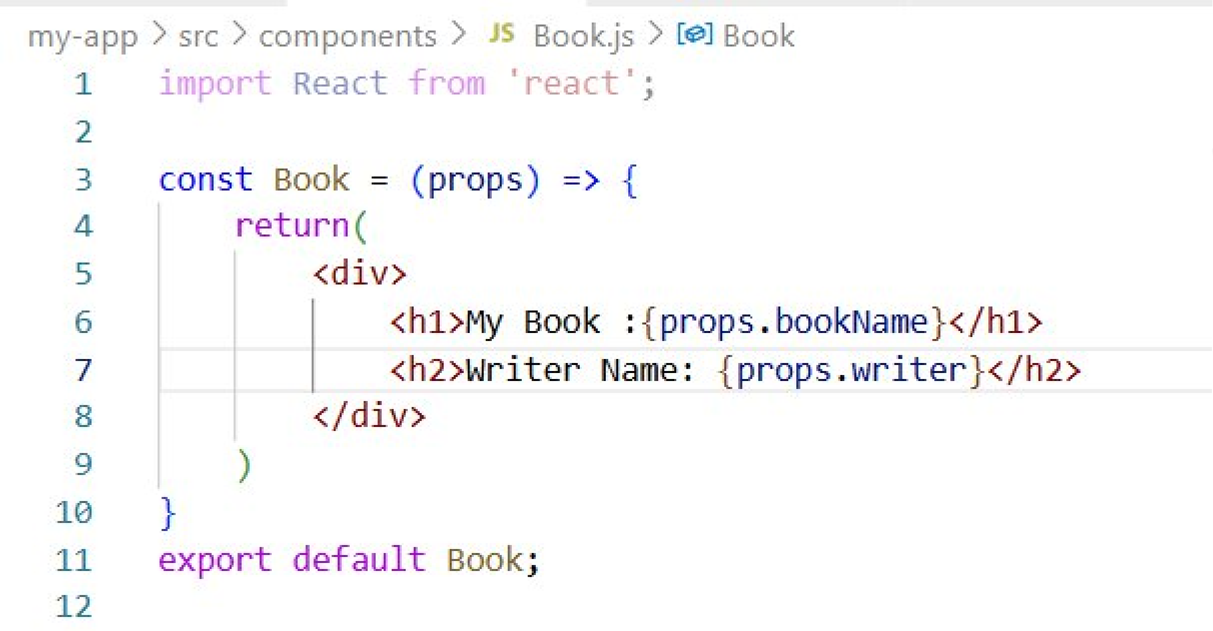
```
            {/* <Person/> */}
            <Book/>
        </div>
    );
}

export default App;
```



Easy Learning

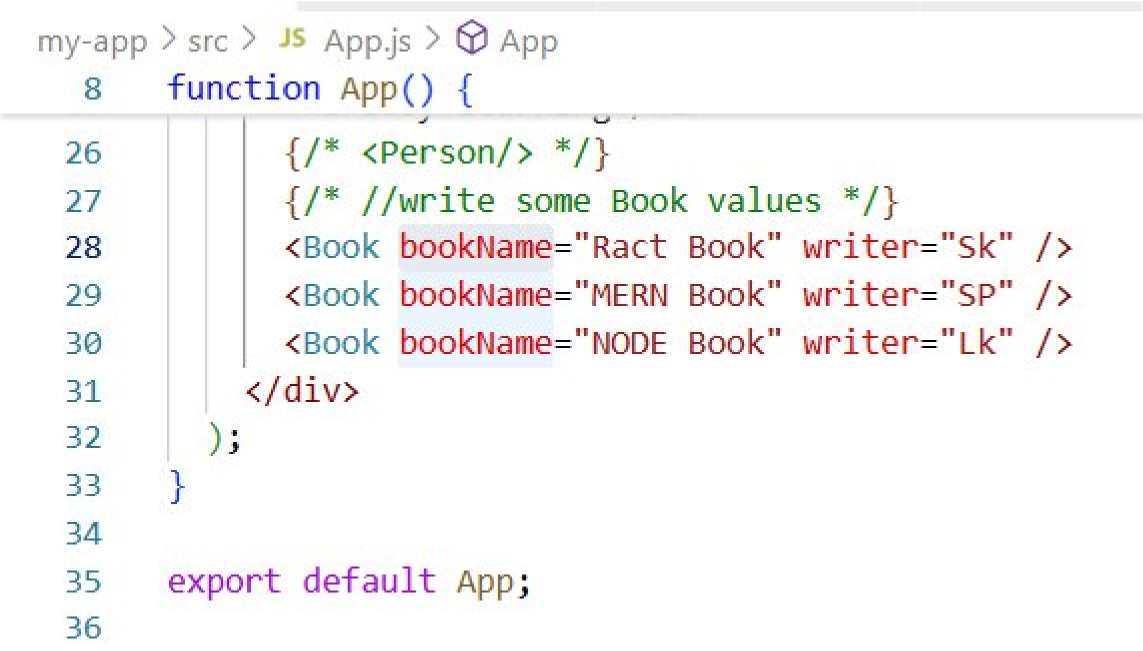
My Book Component

React Js



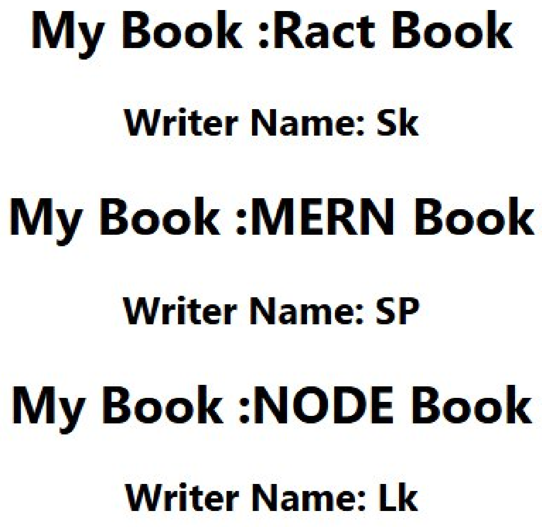
my-app > src > components > JS Book.js > Book

```
import React from 'react';

const Book = (props) => {
    return(
        <div>
            <h1>My Book :{props.bookName}</h1>
            <h2>Writer Name: {props.writer}</h2>
        </div>
    )
}
export default Book;
```



my-app > src > JS App.js > App

```
function App() {
      {/* <Person/> */}
      {/* //write some Book values */}
      <Book bookName="Ract Book" writer="Sk" />
      <Book bookName="MERN Book" writer="SP" />
      <Book bookName="NODE Book" writer="Lk" />
    </div>
  );
}

export default App;
```



My Book :Ract Book

Writer Name: Sk

My Book :MERN Book

Writer Name: SP

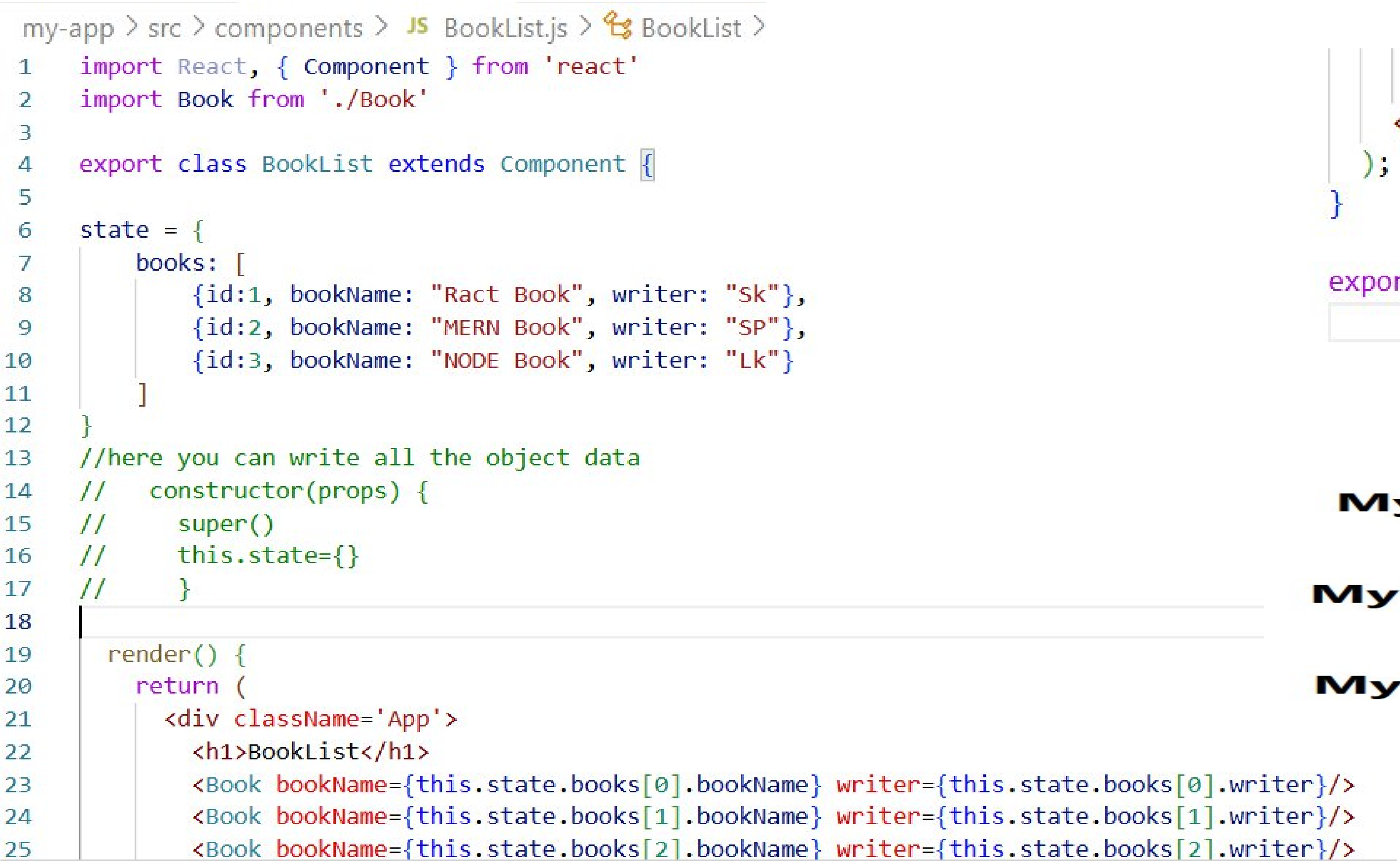
My Book :NODE Book

Writer Name: Lk

React Js- State of a component

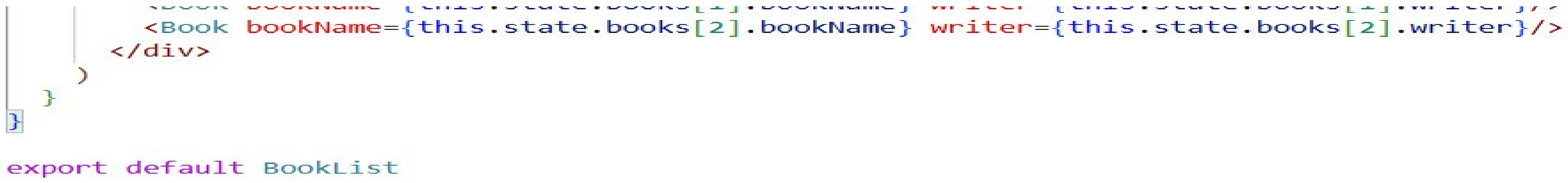
state- is a plain JS object used by React to represent an information about the component
same data same components
used for only class components
state is mutable and can be changed
state can be changed using this.setState() method
setState()- is a method that is used to update the state object
props-Props are read-only components. It is an object which stores the value of attributes of a tag and work similar to the HTML attributes.
Props are passed to the component in a way similar to the HTML attributes.
Props are used to pass data from one component to another.
Props are immutable so we cannot modify the props from inside the component.
Props can be accessed by the child component.

React Js- state of a components

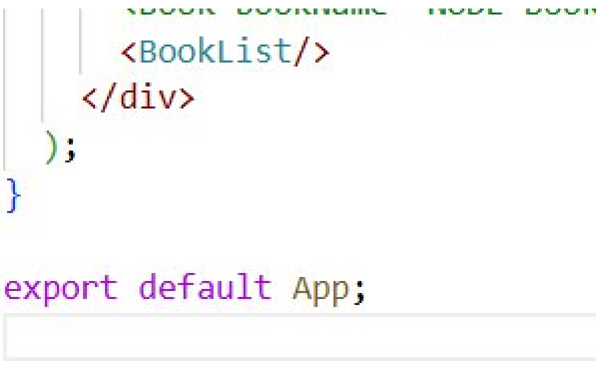


my-app > src > components > JS BookList.js > BookList >

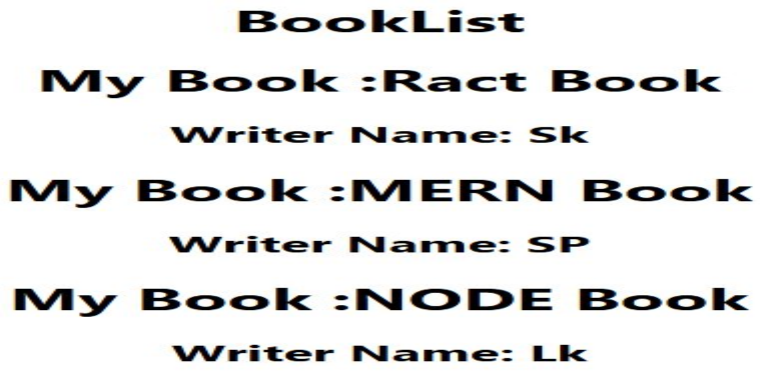
```
import React, { Component } from 'react'
import Book from './Book'

export class BookList extends Component {

state = {
    books: [
        {id:1, bookName: "Ract Book", writer: "Sk"},
        {id:2, bookName: "MERN Book", writer: "SP"},
        {id:3, bookName: "NODE Book", writer: "Lk"}
    ]
}
//here you can write all the object data
//   constructor(props) {
//     super()
//     this.state={}
//     }

  render() {
    return (
      <div className='App'>
        <h1>BookList</h1>
        <Book bookName={this.state.books[0].bookName} writer={this.state.books[0].writer}/>
        <Book bookName={this.state.books[1].bookName} writer={this.state.books[1].writer}/>
        <Book bookName={this.state.books[2].bookName} writer={this.state.books[2].writer}/>
```



```
      <Book bookName={this.state.books[2].bookName} writer={this.state.books[2].writer}/>
    </div>
   )
  }
}

export default BookList
```



```
        <BookList/>
      </div>
    );
}

export default App;
```



BookList

My Book :Ract Book

Writer Name: Sk

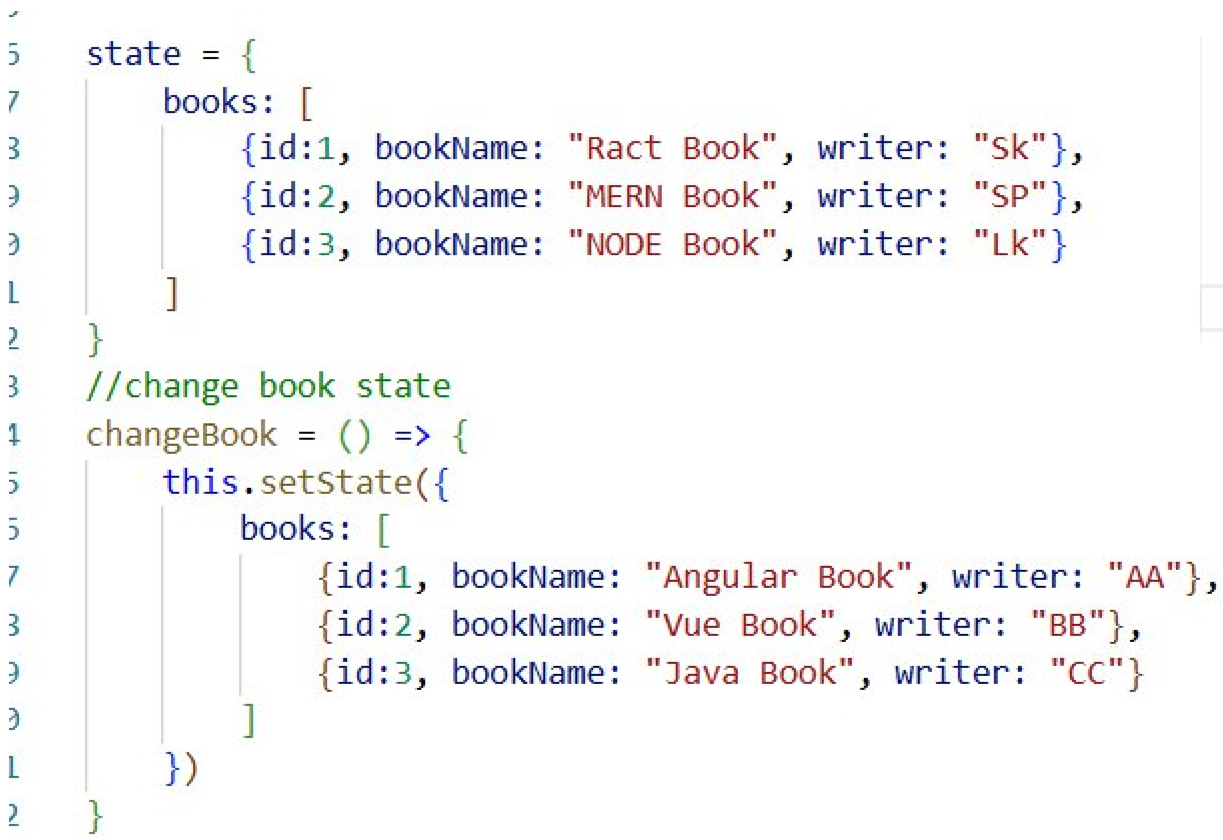
My Book :MERN Book

Writer Name: SP

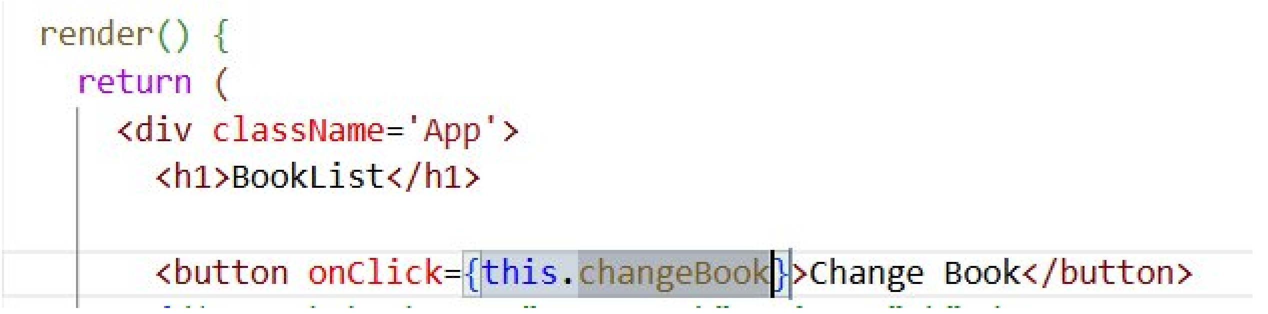
My Book :NODE Book

Writer Name: Lk

React Js- changing state with Js events

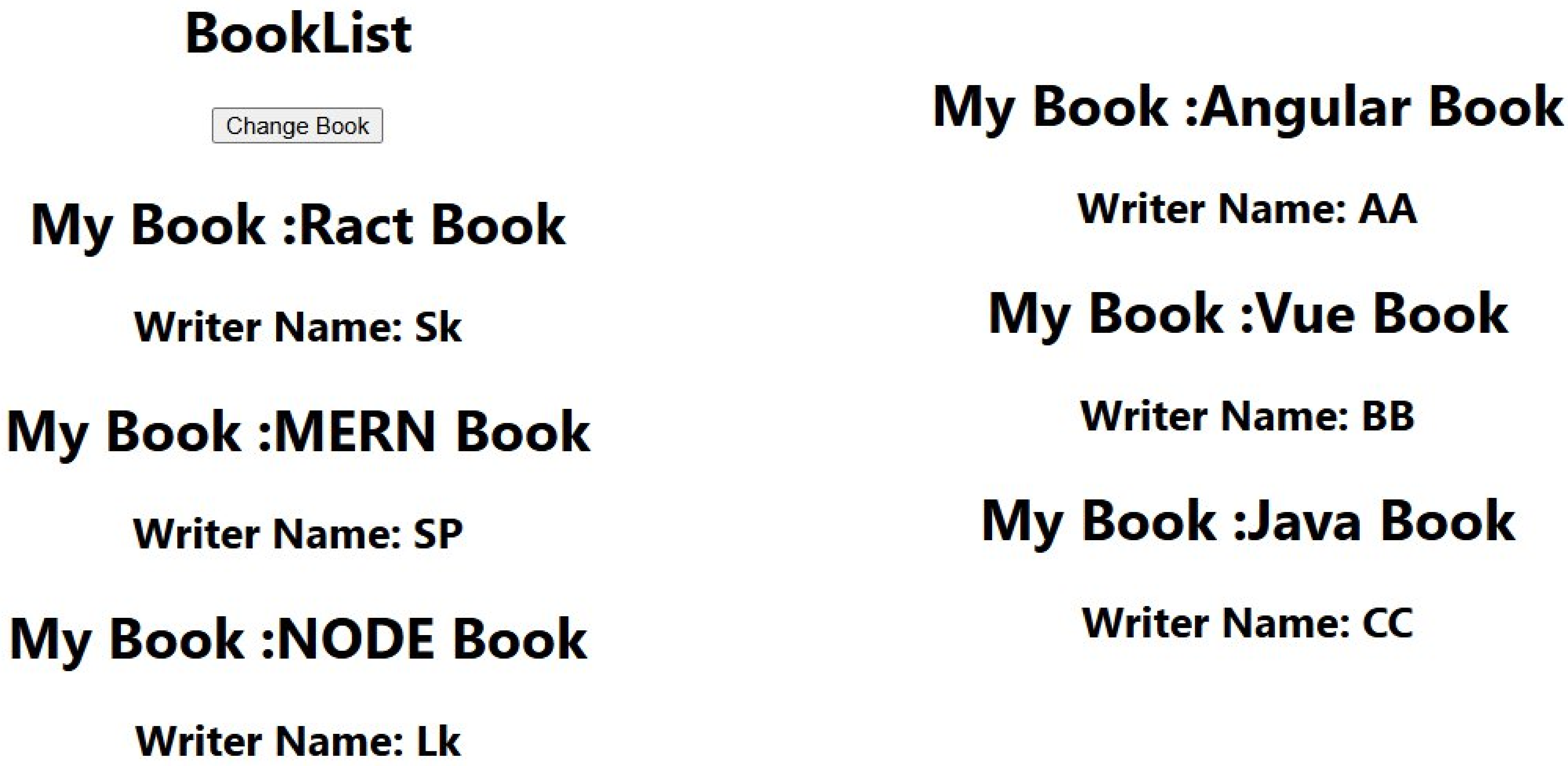


```
state = {
    books: [
        {id:1, bookName: "Ract Book", writer: "Sk"},
        {id:2, bookName: "MERN Book", writer: "SP"},
        {id:3, bookName: "NODE Book", writer: "Lk"}
    ]
}
//change book state
changeBook = () => {
    this.setState({
        books: [
            {id:1, bookName: "Angular Book", writer: "AA"},
            {id:2, bookName: "Vue Book", writer: "BB"},
            {id:3, bookName: "Java Book", writer: "CC"}
        ]
    })
}
```



```
render() {
  return (
    <div className='App'>
      <h1>BookList</h1>

      <button onClick={this.changeBook}>Change Book</button>
```

React Js



BookList

My Book :Ract Book

Writer Name: Sk

My Book :MERN Book

Writer Name: SP

My Book :NODE Book

Writer Name: Lk

My Book :Angular Book

Writer Name: AA

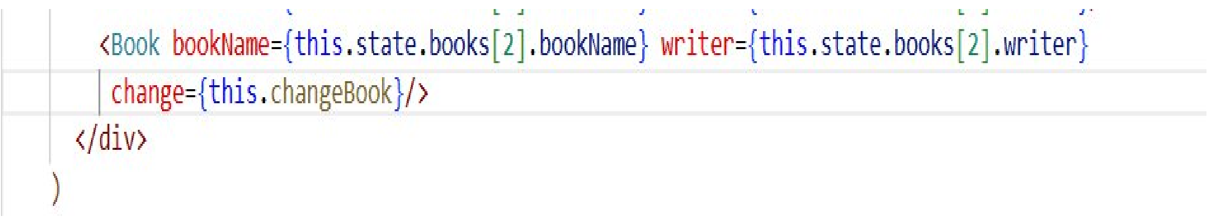
My Book :Vue Book

Writer Name: BB

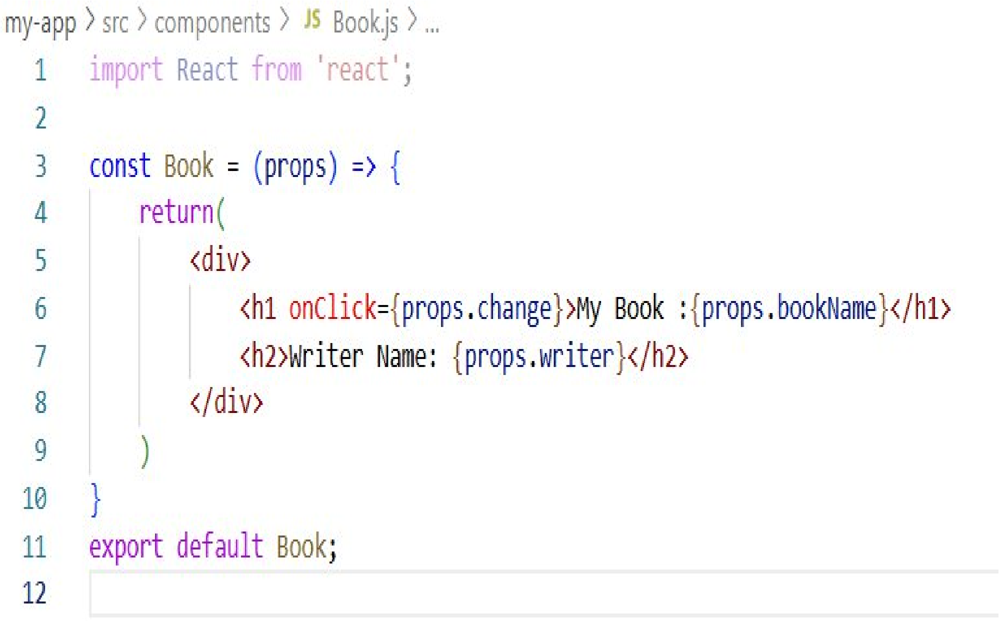
My Book :Java Book

Writer Name: CC

React Js- pass methods in props



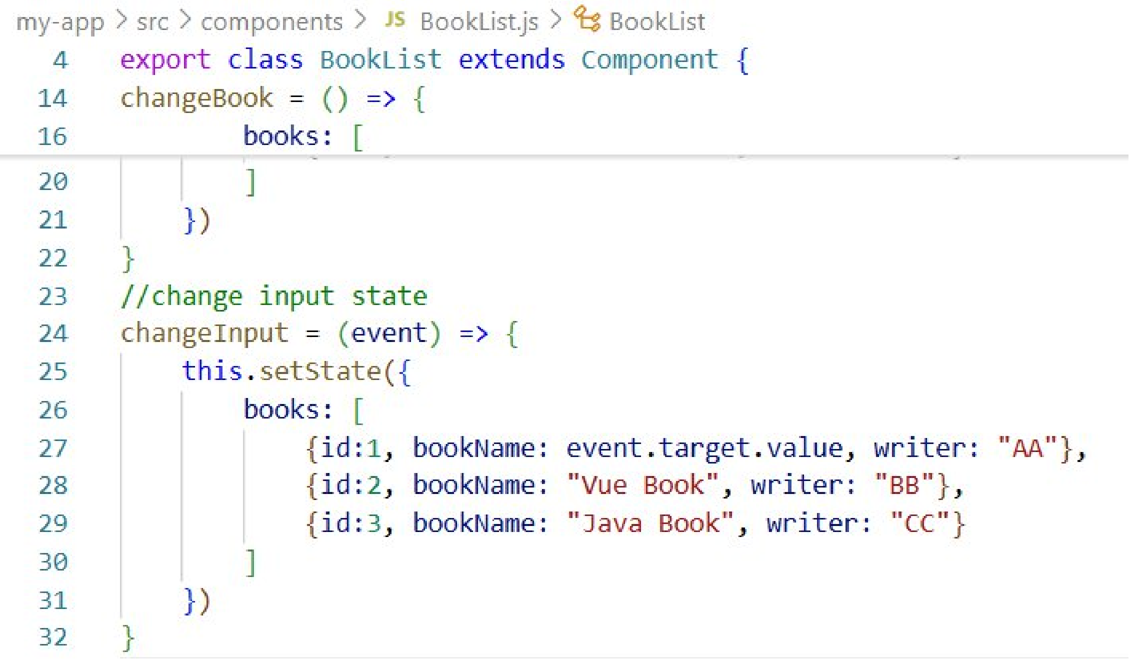
```
      <Book bookName={this.state.books[2].bookName} writer={this.state.books[2].writer}
       change={this.changeBook}/>
    </div>
  )
```



my-app > src > components > JS Book.js > ...

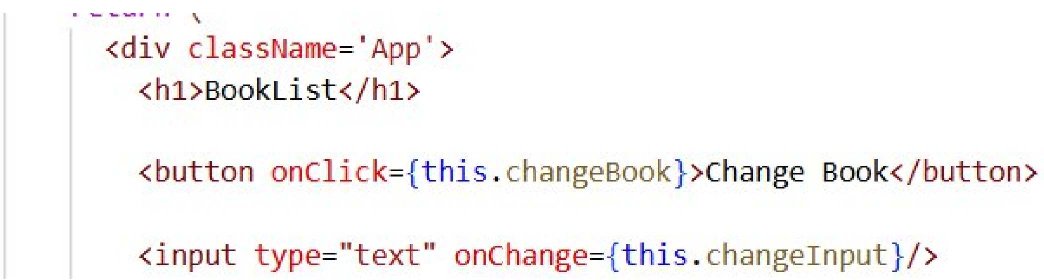
```
import React from 'react';

const Book = (props) => {
    return(
        <div>
            <h1 onClick={props.change}>My Book :{props.bookName}</h1>
            <h2>Writer Name: {props.writer}</h2>
        </div>
    )
}
export default Book;

```

React Js

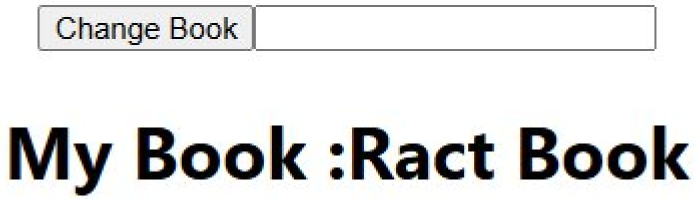


my-app > src > components > JS BookList.js > BookList

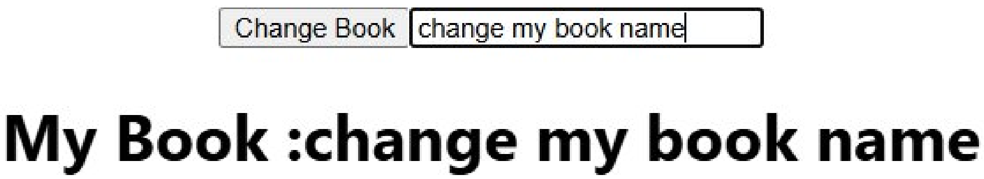
```
export class BookList extends Component {
changeBook = () => {
        books: [
        ]
    })
}
//change input state
changeInput = (event) => {
    this.setState({
        books: [
            {id:1, bookName: event.target.value, writer: "AA"},
            {id:2, bookName: "Vue Book", writer: "BB"},
            {id:3, bookName: "Java Book", writer: "CC"}
        ]
    })
}
```



```
<div className='App'>
  <h1>BookList</h1>

  <button onClick={this.changeBook}>Change Book</button>

  <input type="text" onChange={this.changeInput}/>
```

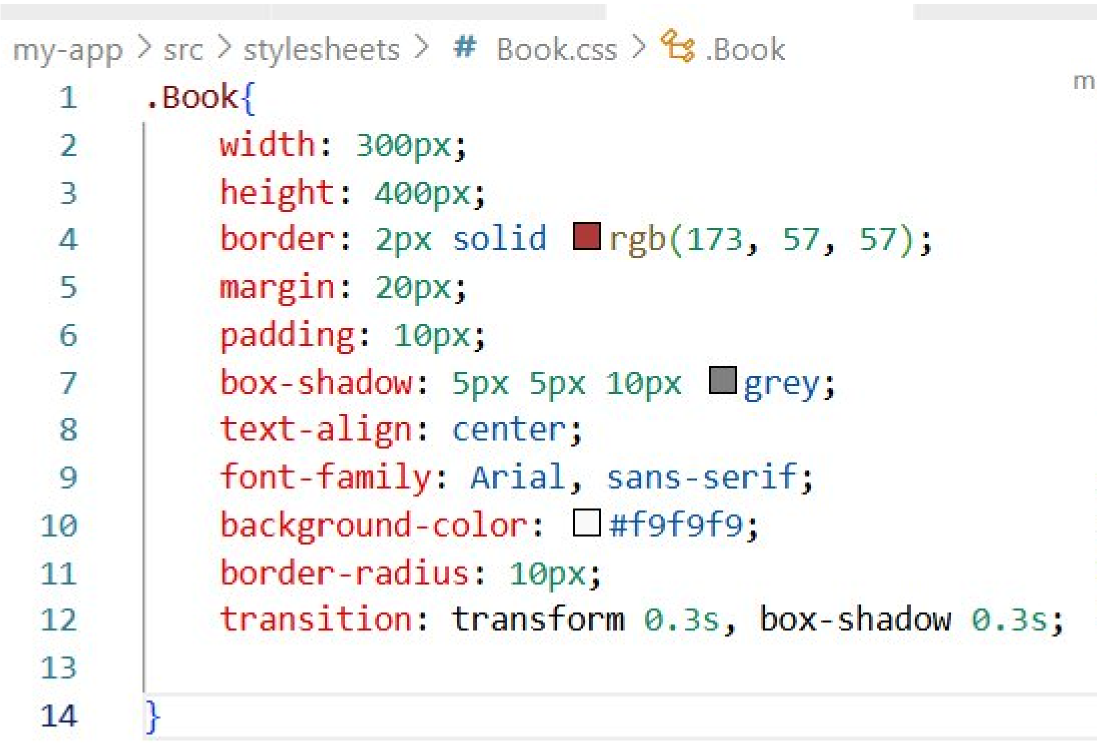


My Book :Ract Book



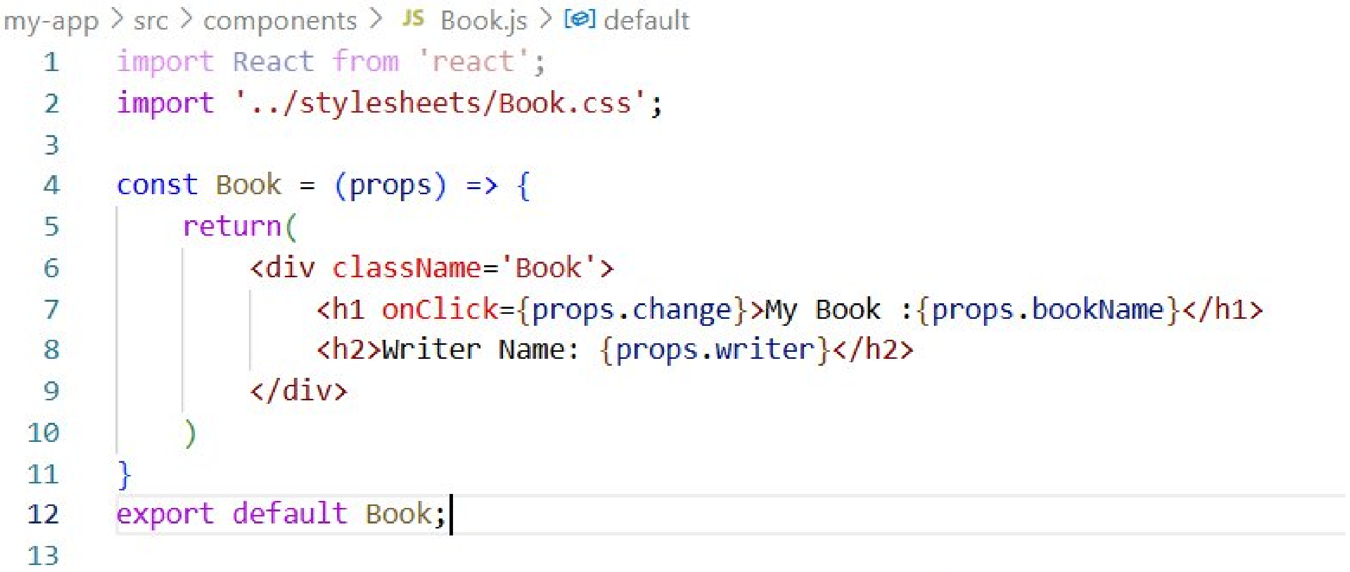
My Book :change my book name

React Js styling components



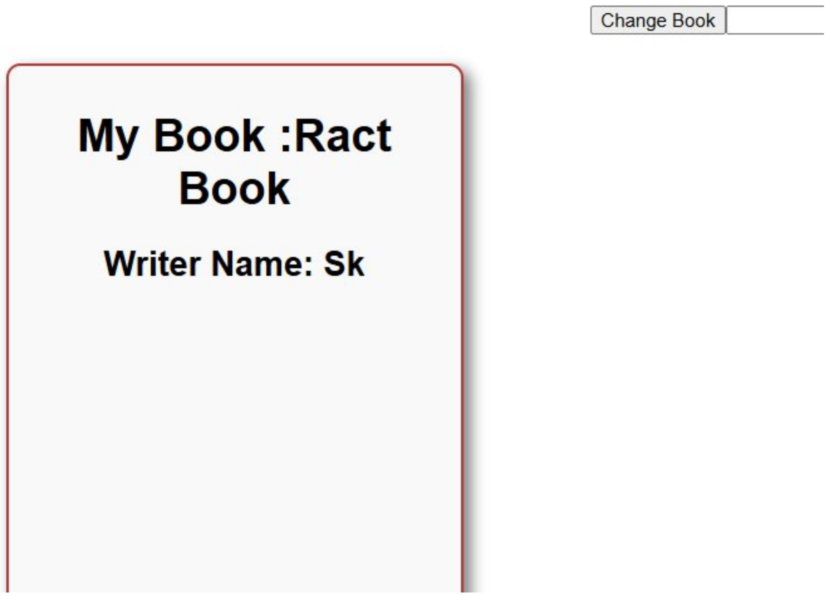
my-app > src > stylesheets > # Book.css > .Book

```css
.Book{
    width: 300px;
    height: 400px;
    border: 2px solid rgb(173, 57, 57);
    margin: 20px;
    padding: 10px;
    box-shadow: 5px 5px 10px grey;
    text-align: center;
    font-family: Arial, sans-serif;
    background-color: #f9f9f9;
    border-radius: 10px;
    transition: transform 0.3s, box-shadow 0.3s;

}
```



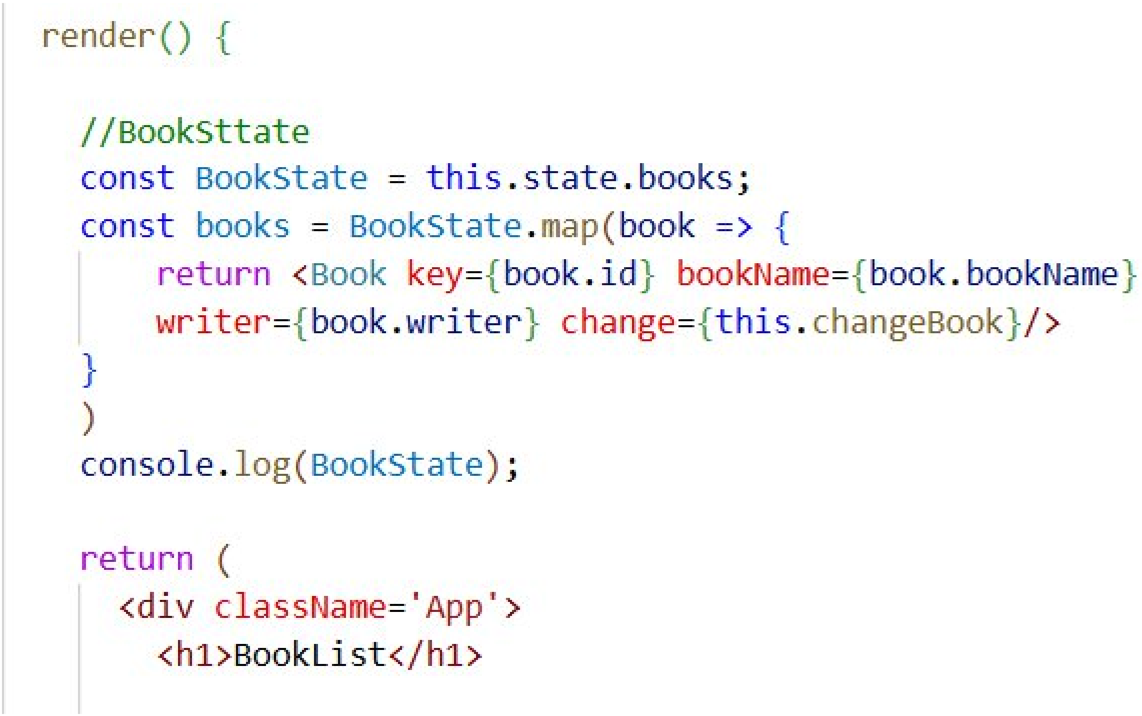
my-app > src > components > JS Book.js > [e] default

```jsx
import React from 'react';
import '../stylesheets/Book.css';

const Book = (props) => {
    return(
        <div className='Book'>
            <h1 onClick={props.change}>My Book :{props.bookName}</h1>
            <h2>Writer Name: {props.writer}</h2>
        </div>
    )
}
export default Book;

```

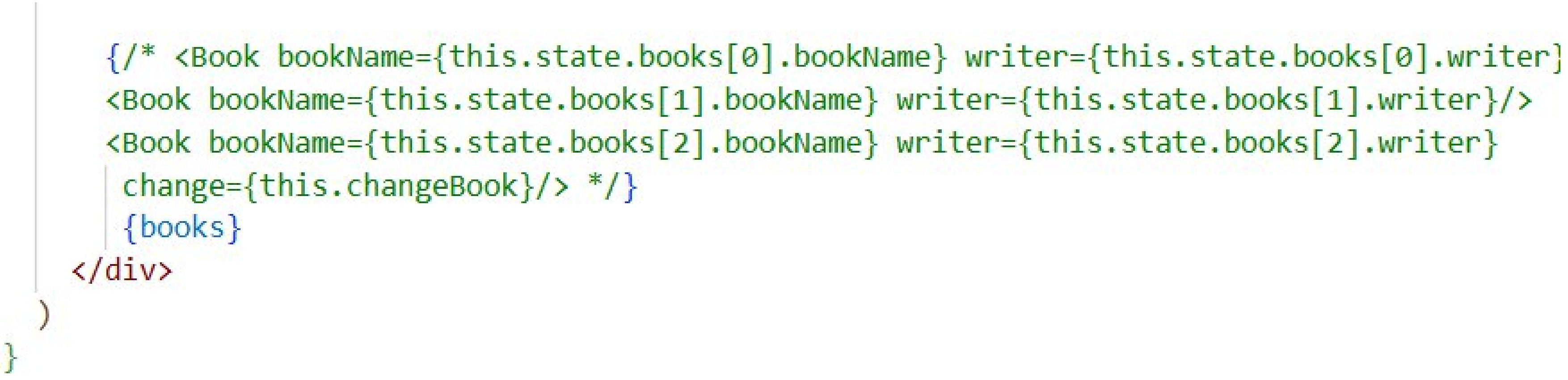


```
My Book :Ract
Book

Writer Name: Sk
```

React Js- creating component list

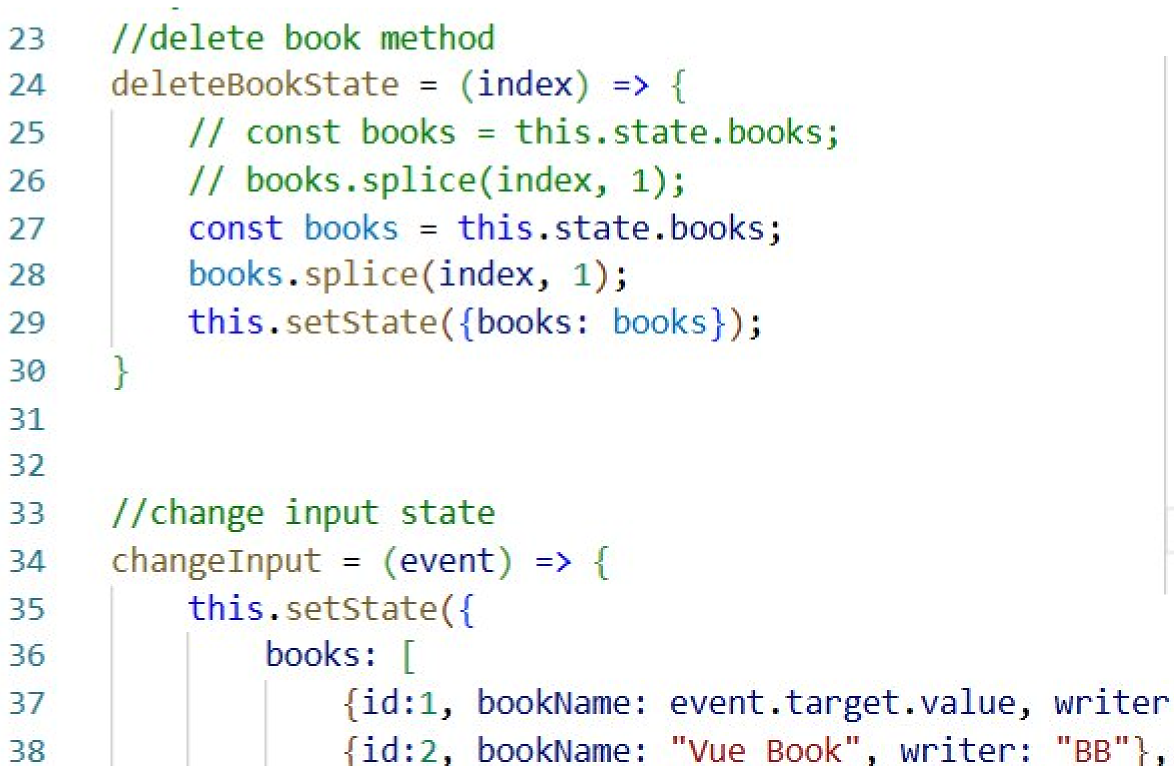


```
 render() {

   //BookSttate
   const BookState = this.state.books;
   const books = BookState.map(book => {
       return <Book key={book.id} bookName={book.bookName}
       writer={book.writer} change={this.changeBook}/>
   }
   )
   console.log(BookState);

   return (
     <div className='App'>
       <h1>BookList</h1>
```

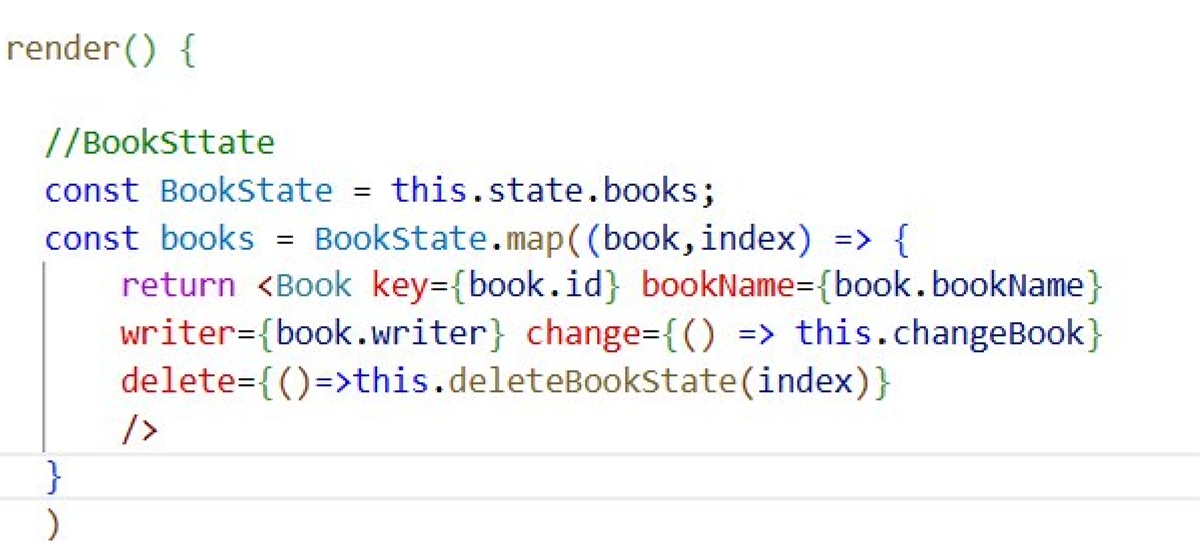


```
      {/* <Book bookName={this.state.books[0].bookName} writer={this.state.books[0].writer}
      <Book bookName={this.state.books[1].bookName} writer={this.state.books[1].writer}/>
      <Book bookName={this.state.books[2].bookName} writer={this.state.books[2].writer}
       change={this.changeBook}/> */}
       {books}
    </div>
  )
}
```

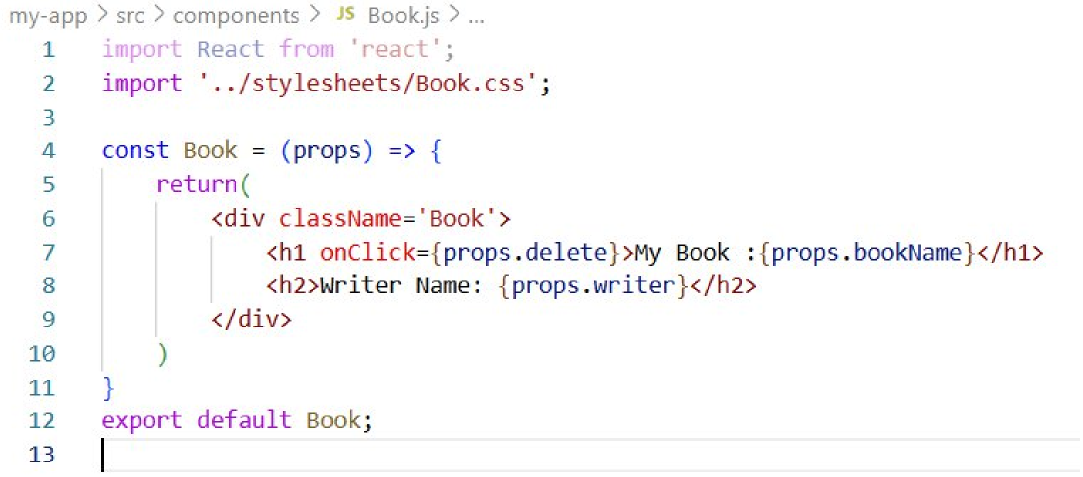
React Js- change state



```
//delete book method
deleteBookState = (index) => {
    // const books = this.state.books;
    // books.splice(index, 1);
    const books = this.state.books;
    books.splice(index, 1);
    this.setState({books: books});
}


//change input state
changeInput = (event) => {
    this.setState({
        books: [
            {id:1, bookName: event.target.value, writer:
            {id:2, bookName: "Vue Book", writer: "BB"},
```



```
render() {

  //BookSttate
  const BookState = this.state.books;
  const books = BookState.map((book,index) => {
      return <Book key={book.id} bookName={book.bookName}
      writer={book.writer} change={() => this.changeBook}
      delete={()=>this.deleteBookState(index)}
      />
  }
  )
```



my-app > src > components > Book.js > ...

```
import React from 'react';
import '../stylesheets/Book.css';

const Book = (props) => {
    return(
        <div className='Book'>
            <h1 onClick={props.delete}>My Book :{props.bookName}</h1>
            <h2>Writer Name: {props.writer}</h2>
        </div>
    )
}
export default Book;

```

Thank You!!