

Chapter 3. JSX

교재: 처음만난 리액트 (저자: 이인제, 한빛출판사)



Contents

- CHAPTER 3: JSX

3.1 JSX란?

3.2 JSX의 역할

3.3 JSX의 장점

3.4 JSX 사용법

3.5 (실습) JSX 코드 작성해 보기

SECTION 3.1 JSX란?

- JSX 개념

- A syntax extension to JavaScript (자바스크립트 확장 문법)

- JavaScript +XML / HTML

- JSX 코드

```
1 const element = <h1>Hello, world!</h1>;
```

SECTION 3.2 JSX의 역할

- JSX는 내부적으로 XML/HTML 코드를 자바스크립트로 변환
- createElement() - JSX 코드를 자바스크립트 코드로 변환하는 역할
- JSX를 사용한 코드

```
1 class Hello extends React.Component {  
2   render() {  
3     return <div>Hello {this.props.toWhat}</div>;  
4   }  
5 }  
6  
7 ReactDOM.render(  
8   <Hello toWhat="World" />,  
9   document.getElementById( 'root' )  
10 );
```

SECTION 3.2 JSX의 역할

- JSX를 사용하지 않은 코드

```
1 class Hello extends React.Component {  
2   render() {  
3     return React.createElement('div', null, `Hello ${this.props.toWhat}`);  
4   }  
5 }  
6  
7 ReactDOM.render(  
8   React.createElement(Hello, { toWhat: 'World' }, null),  
9   document.getElementById('root')  
10 );
```

- JSX 문법을 사용하면 리액트에서는 내부적으로 createElement 함수를 사용하도록 변환

SECTION 3.2 JSX의 역할

- JSX 코드 비교

```
1 const element = (  
2   <h1 className="greeting">  
3     Hello, world!  
4   </h1>  
5 )  
6  
7 const element = React.createElement(  
8   'h1',  
9   { className: 'greeting' },  
10  'Hello, world!',  
11 )
```

JSX를 사용한 코드

JSX를 사용하지 않은 코드

- ✓ JSX를 사용하면 코드가 간결해지고 생산성과 가독성이 향상됨

SECTION 3.2 JSX의 역할

- React.createElement()의 결과로 자바스크립트 객체가 생성됨

```
1 const element = {  
2   type: 'h1',  
3   props: {  
4     className: 'greeting',  
5     children: 'Hello, world!'  
6   }  
7 }
```

'React element'

SECTION 3.2 JSX의 역할

- createElement() 함수

```
1 React.createElement(  
2   type,  
3   [props],  
4   [...children]  
5 )
```

1. type - 엘리먼트의 유형. <div> 같은 HTML 태그나 다른 리액트 컴포넌트
2. props - 속성
3. children - 현재 엘리먼트가 포함하는 자식 엘리먼트

SECTION 3.3 JSX의 장점

1. 간결한 코드

- JSX 사용한 코드

```
1 <div>Hello, {name}</div>
```

- JSX 사용하지 않은 코드

```
1 React.createElement('div', null, `Hello, ${name}`);
```

2. 가독성 향상

- 의미 파악이 쉬움 (유지보수 용이)
- 버그 발견 용이

SECTION 3.3 JSX의 장점

3. Injection Attacks 방어 (보안성 향상)

- Injection Attack
 - 입력창에 소스코드를 입력하여 해당 코드가 실행되도록 만드는 해킹 기법
 - ex) ID 입력 창에 자바스크립트 코드 `<script>alert();</script>`를 입력
- ReactDOM은 렌더링하기 전에 임베딩된 값을 모두 문자열로 변환

```
1 const title = response.potentiallyMaliciousInput;  
2  
3 // 이 코드는 안전합니다.  
4 const element = <h1>{title}</h1>;
```

- XSS(cross-site-scripting) 공격을 방어할 수 있음

SECTION 3.4 JSX 사용법

... XML / HTML
{ JavaScript 코드 }
... XML / HTML

- 중괄호{ }를 이용하여 자바스크립트 변수 참조

```
1 const name = '소플';  
2 const element = <h1>안녕, {name}</h1>;  
3  
4 ReactDOM.render(  
5   element,  
6   document.getElementById('root')  
7 );
```

SECTION 3.4 JSX 사용법

- 중괄호{ }를 이용하여 자바스크립트 함수 호출

```
1 function formatName(user) {  
2   return user.firstName + ' ' + user.lastName;  
3 }  
4  
5 const user = {  
6   firstName: 'Inje',  
7   lastName: 'Lee'  
8 };  
9  
10 const element = (  
11   <h1>  
12     Hello, {formatName(user)}  
13   </h1>  
14 );  
15  
16 ReactDOM.render(  
17   element,  
18   document.getElementById('root')  
19 );
```

SECTION 3.4 JSX 사용법

- 태그의 속성(attribute)에 값을 넣는 방법

```
1 // 큰따옴표 사이에 문자열을 넣거나
2 const element = <div tabIndex="0"></div>;
3
4 // 중괄호 사이에 자바스크립트 코드를 넣으면 됨!
5 const element = <img src={user.avatarUrl}></img>;
```

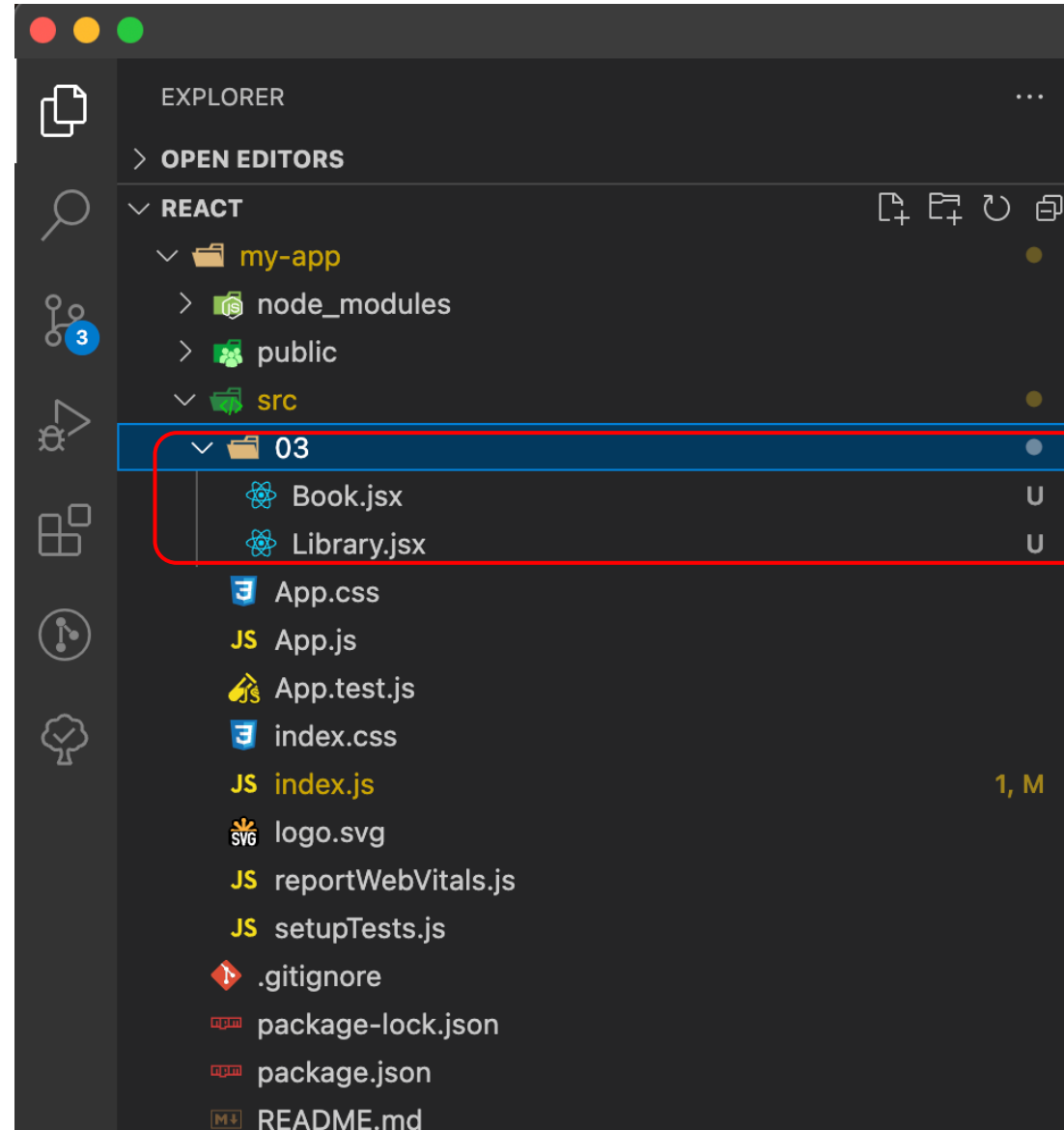
SECTION 3.4 JSX 사용법

- 자식(children)을 정의하는 방법

```
1 const element = (  
2   <div>  
3     <h1>안녕하세요</h1>  
4     <h2>열심히 리액트를 공부해 봅시다!</h2>  
5   </div>  
6 );
```


PRACTICE 3.5 JSX 코드 작성해보기

- Source 구조



PRACTICE 3.5 JSX 코드 작성해보기


- Book.jsx

```
my-app > src > 03 >  Book.jsx > ...
```

```
1  import React from 'react';
2
3  function Book(props) {
4    return (
5      <div>
6        <h1>{'이 책의 이름은 ${props.name}입니다.'}</h1>
7        <h2>{'이 책은 총 ${props.numOfPage}페이지로 이뤄져 있습니다.'}</h2>
8      </div>
9    );
10 }
11
12 export default Book;
```


PRACTICE 3.5 JSX 코드 작성해보기

- Library.jsx

```
my-app > src > 03 >  Library.jsx > ...  
1   import React from 'react';  
2   import Book from './Book';  
3  
4   function Library(props) {  
5     return (  
6       <div>  
7         <Book name="처음 만난 파이썬" numOfPage={300} />  
8         <Book name="처음 만난 AWS" numOfPage={400} />  
9         <Book name="처음 만난 리액트" numOfPage={500} />  
10      </div>  
11    )  
12  }  
13  
14  export default Library;
```

PRACTICE 3.5 JSX 코드 작성해보기

- index.js

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

import Library from './03/Library';

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

PRACTICE 3.5 JSX 코드 작성해보기

◦ 실행

```
// VS Code - Terminal  
$ npm start
```

