

## 소프의 처음 만나 리액트



## 제 3장 JSX 소개



## Q & A

- 언제라도 질문하세요  
모르면 외우면 되지만 코딩 안되면 꼭 질문
  1. 대면수업에서 질문
  2. 한성e-class 질의응답게시판
- 강사의 1번 선생님은 여러분의 질문





# 목차

1. JSX란
2. JSX의 역할
3. JSX의 장점
4. JSX 사용법
5. 실습 : JSX 코드 작성해보기
6. 마치며



## 3.1 JSX란?

- JSX : A syntax extension to JS
- JS와 다른 새로운 언어 : JS + XML /HTML 이 교재에서는 JS+HTML

학생> 교수님~ React는 JS 라이브러리인데 왜 JSX 언어로 코딩하나요?

교수> 3.3 절에 JSX의 장점이 나옵니다.

1. 코드가 간결해진다  
jsx 사용  
`<div>Hello, {name}</div>`  
jsx 미사용  
`React.createElement('div', null, 'Hello, ${name}');`
2. 가독성이 향상  
코드 파악이 쉬워서 버그 쉽게 발견할 수 있다
3. Injection Attack 해킹을 방어할 수 있다

➔ 이 교재에서는 JS 대신 모두 JSX 로 코딩



## 3.5 JSX 코드 작성해 보기

- 과제6에서 CRA방식으로 my-app을 만들었습니다.
- 공지사항게시판 본문코드에서 chapter\_03폴더를 복사하여 c:\ws\_react\my-app\src\chapter\_03을 만듭니다.
- index.js 파일을 편집하여 만든다

7번 라인 추가

11번 라인 App => Library 변경

```
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import './index.css';
4 import App from './App';
5 import reportWebVitals from './reportWebVitals';
6
7 const root = ReactDOM.createRoot(document.getElementById('root'));
8 root.render(
9   <React.StrictMode>
10     <App />
11   </React.StrictMode>
12 );
13
14 // If you want to start measuring performance in your app,
15 // to log results (for example: reportWebVitals) or send to an analytics endpoint. Learn more:
16 // https://reactjs.org/docs/concentrating-on-the-web.html
17 reportWebVitals();
```

```
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import './index.css';
4 import App from './App';
5 import reportWebVitals from './reportWebVitals';
6
7 import Library from './chapter_03/Library';
8
9 const root = ReactDOM.createRoot(document.getElementById('root'));
10 root.render(
11   <React.StrictMode>
12     <Library />
13   </React.StrictMode>
14 );
15
16 // If you want to start measuring performance in your app,
17 // to log results (for example: reportWebVitals) or send to an analytics endpoint. Learn more:
18 // https://reactjs.org/docs/concentrating-on-the-web.html
19 reportWebVitals();
```

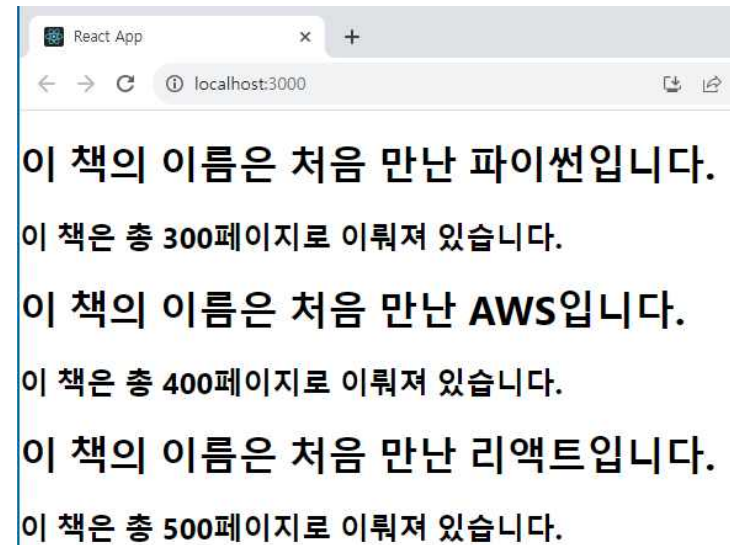


## 3.5 JSX 코드 작성해 보기

- VS code로 chapter\_03의 Book.jsx와 Library.jsx를 봅니다
- 이 코드를 VS code terminal에서 수행해 봅니다.  
C:\ws\_react > cd my-app  
C:\ws\_react\my-app> npm start
- Index.js, Libray.jsx, Book.jsx로 만든 화면입니다

학생> 교수님~ 2장에서는 CRA방식으로 react웹을 수행했는데 3장부터 16장까지 jsx로 코딩하려니 1학년 웹기초 배운것과 전혀 연결이 안되어 코드이해하기 어렵습니다.

교수> 그래서 CDN방식으로 먼저하고 동일한 내용을 CRA방식으로 코딩하겠습니다.





# CDN방식 코드 작성해 보기

- 한성e-class에서 4주차 추가코드를 다운로드합니다.
- CDN\chapter\_03\_reactJS\index.html을 Vscode에서 Ctrl+f5로 수행해봅니다  
Vscode에서 Ctrl+f5로 수행하는 것은 index.html을 크롬에 Drag&Drop과 동일한 기능입니다

학생> 크롬에 아무글자도 없습니다.

교수> F12의 console창에 error 메시지가 나옵니다. Index.js가 File 프로토콜을 지원하지 않는다고 하여 LiveServer로 실행합니다.

CRA로 수행한 결과와 동일합니다.

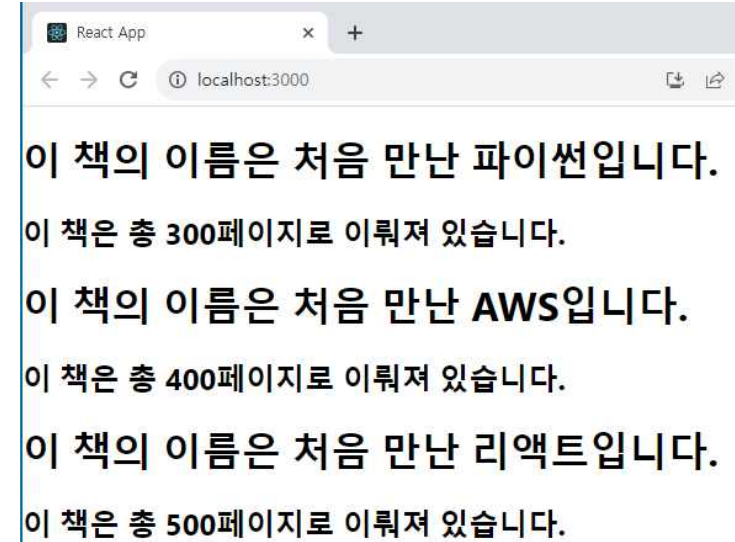
학생> CDN은 파일4개를 사용합니다.

Index.html, index.js, Library.js, Book.js  
하지만 CRA는 파일3개를 사용합니다.

Index.js, Library.jsx, Book.jsx

CRA에서 index.html은 어디있나요?

교수> 16장 미니블로그 맛보기에서 보여주겠습니다.





## CDN 코드 작성해 보기 (index.html, index.js)

Index.html 코드를 보니 index.js를 실행합니다.

Index.js에서

ReactDOM.createRoot()과 React.createElement()는 무슨기능인가요?

ReactDOM.createRoot() : <https://react.dev/reference/react-dom/client/createRoot>

React.createElement() : <https://react.dev/reference/react/createElement>

자세한 설명은 4장 엘리먼트 렌더링에서 합니다





# CDN 코드 작성해 보기 (Library.js)

학생> Library.js 코드를 보니 `React.createElement()` 함수 사용하는 것을 보았습니다.

CRA방식의 `Libray.jsx` 코드와 비교해 보니 혼란스럽습니다.

교수> `jsx`코드는 크롬에서 수행이 되지 않기에 CRA방식에서 크롬에 전달될때는 `js`로 변환되어 전달됩니다.

학생> `jsx`를 `js`로 어떻게 변환하나요?

교수> Babel을 CRA개발 환경에서 내부적으로 사용합니다. (<https://babeljs.io/>)

Babel is javascript compiler.

예) JSX를 JS로 변환, TS를 JS로 변환

React에서는 JSX를 사용하고 Angular에서는 TS를 사용하는데 크롬은 JS만 실행이 되기에 Babel compiler가 필요합니다.

통상 compiler는 HL에서 ML(기계어)로 변환시키는데 Babel은 HL에서 HL로 변환시키기에 transpiler 라고도 합니다



# CDN 코드 작성해 보기 (Babel<sup>사용</sup>)

Library.jsx를 Library.js로 변환

Babeljs.io의 Try it out 창에서

PRESETS에서 react 선택(Angular에서는 typescript선택)

Options의 React Runtime은 classic 선택

3장(JSX소개), 4장(Element), 5장(Component)까지 JS로 된 React를 하고 그 다음 JSX로 된 React를 하겠습니다.

3장, 4장, 5장 코드가 JSX로 되어 있기에 Babel을 이용하여 JS로 변환후 코드 내용 설명하겠습니다.

즉 JS로 된 React를 설명하고 그 다음에 JSX로 된 React를 설명합니다



## 4가지 코딩 방식

3장, 4장, 5장 코드가 **JSX**로 되어 있습니다.

이해도를 높이기 위해 같은 기능을 4가지 코딩방식으로 하겠습니다.

1. **reactJSX**

2. **reactJS** : **JSX**코드를 **Babel**을 사용하여 **JS**로 변환

3. **vanillaJS**

[https://en.wikipedia.org/wiki/JavaScript#Libraries\\_and\\_frameworks](https://en.wikipedia.org/wiki/JavaScript#Libraries_and_frameworks)

4. **noJS** : **JS**사용하지 않고 **HTML**만 사용

학생> 교수님~ 3장,4장,5장은 왜 4가지 코딩 방식을 사용하나요?

교수>

1. 6장~16장은 **JSX**로만 코딩 예정입니다.

2. **JSX**방식은 새로운 언어라서 **reactJS**로 **react** 라이브러리를 이해합니다. ( **react doc**이 **reactJS**설명)

3. **vanillaJS**는 웹기초에서는 **vanillaJS**로 코딩했기에 **vanillaJS**와 **reactJS** 비교를 통해 1학년 웹기초에서 배운 내용을 **reactJS**로 확장하고자 합니다

4. **reactJS**는 **html**파일 1개만 사용(**SPA**:교재31P)하고 **HTML** 태그도 **<div>** 1개만 사용하고 나머지 **HTML** 태그는 **JS**파일에서 만들어 냅니다.

**noJS** 즉 **HTML**만 사용으로 **SPA**방식(**html**파일 1개만 사용하고 **html**파일에는 **<div>** 태그 1개만 사용)과 비교하고자 합니다



# 4가지 코딩 방식 중 1.reactJSX

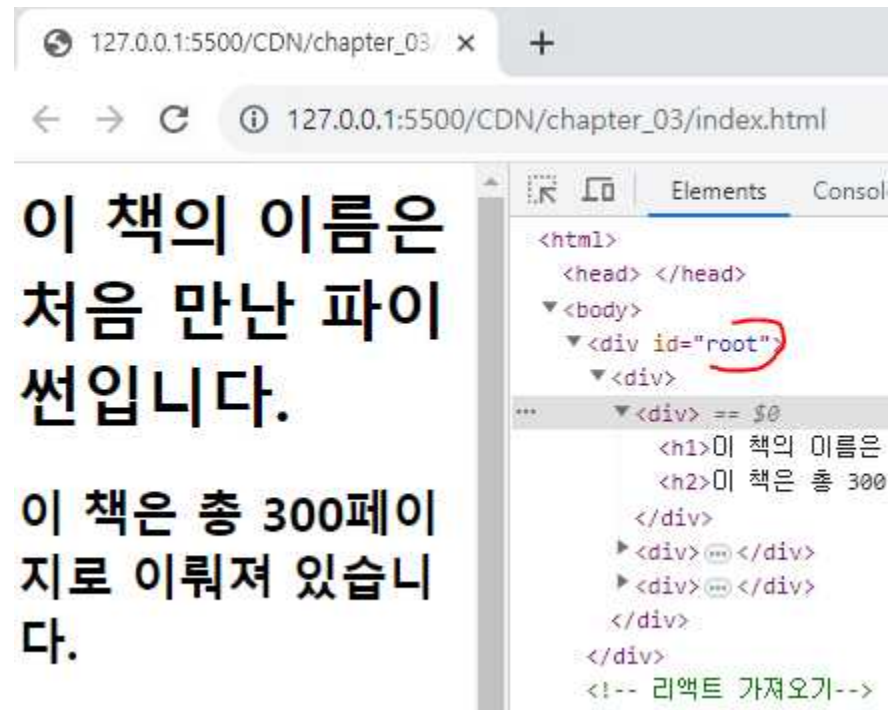
코드위치: c:\ws\_react\my-app\src\chapter\_03

3장(JSX소개), 4장(Element), 5장(Component)까지 JS로 된 React를 강의하고 그 다음 JSX로 된 React를 강의하겠습니다.



# 4가지 코딩 방식 중 2.reactJS

- 코드위치: c:\ws\_react\CDN\chapter\_03\_reactJS ( JSX코드를 Babel을 사용하여 JS로 변환 )
- index.js와 Library.js에 있는 ReactDOM.render()과 React.createElement()는 4장(Element), 5장(Component)에서 설명
- F12의 elements 창에서 root div tag 아래 3개의 div tag 있음
- Index.html에 div tag는 root div tag 1개만 있고 index.js, Library.js, Book.js에서 나머지 div tag 3개 만들었음





# 4가지 코딩 방식 중 3.vanillaJS

- 코드위치: c:\ws\_react\CDN\chapter\_03\_vanillaJS ( reactJS와 동일한 tag생성 )
- F12의 elements 창에서  
root div tag 아래 3개의 div tag 있음
- Index.html에 div tag는 root div tag 1개만 있고  
index.js, Library.js 에서  
나머지 div tag 3개 만들었음

학생> div tag 3개는 어떻게 만들었나요?

교수> Libray.js에서 보면

// 웹기초 8.4 HTML 문서의 동적구성

// DOM 객체 동적 생성

// DOM 트리에 삽입

// DOM 객체의 삭제

```
▼ <div id="root"> == $0
  ▶ <div> ... </div>
  ▶ <div> ... </div>
  ▶ <div> ... </div>
  </div>
```



# 4가지 코딩 방식 중 4.noJS

- 코드위치: c:\ws\_react\CDN\chapter\_03\_noJS ( vanillaJS와 동일한 tag생성)
- F12의 elements 창에서  
root div tag 아래 3개의 div tag 있음
- Index.html에 div tag는 root div tag 1개와  
나머지 div tag 3개 만들었음

```
<html>
  <head> </head>
  <body>
    <div id="root"> == $0
      <div> 처음 만난 파이썬 </div>
      <div> 처음 만난 AWS </div>
      <div> 처음 만난 리액트 </div>
    </div>
```



# 이제 나는 할 수 있다

1. 나는 JSX 코드 형태를 보았다. (JS + HTML)
  2. 나는 3장 코드를 가지고 4가지 코딩방식을 해보았다
    1. reactJSX
    2. reactJS
    3. vanillaJS
    4. noJS (HTML파일만 사용)
  3. 나는 Babel을 이용하여 JSX코드를 JS로 변환할 수 있다
  4. 나는 vanillaJS와 reactJS를 비교해보고 웹기초 8.4를 reactJS로 구현방법을 알았다  
웹기초 8.4 HTML문서의 동적생성  
reactJS가 `React.createElement()`를 통해 DOM 객체 동적 생성한다는 것을 알았다  
reactJS가 `root.render()`를 통해 DOM 트리에 삽입한다는 것을 알았다  
DOM 트리가 바뀌면 브라우저 화면이 바로 갱신된다.
  5. 나는 3장(JSX소개), 4장(Element), 5장(Component)을 reactJS로 코딩하고 그 다음 reactJSX로 코딩하겠다
- 유튜브 강의 동영상 : <https://www.youtube.com/c/소플TV>
  - Q&A 사이트 : <https://www.frontoverflow.com>