AI 활용 빅데이터분석
풀스택웹서비스 SW 개발자 양성과정

React

부산대학교 소프트웨어교육센터
PUSAN NATIONAL UNIVERSITY SOFTWARE EDUCATION CENTER

리액트 (React)

- 사용자 인터페이스를 만들기 위한 JavaScript 라이브러리
  - 선언형
    - 데이터가 변경됨에 따라 적절한 컴포넌트만 효율적으로 갱신하고 렌더링
  - 컴포넌트 기반
    - 스스로 상태를 관리하는 캡슐화된 컴포넌트를 조합해 복잡한 **UI** 생성
    - 컴포넌트**(component)** : 사용자가 정의한 태그
  - 한 번 배워서 어디서나 사용
    - **Node** 서버에서 렌더링을 할 수도 있고, **React Native**를 이용하면 모바일 앱 개발

리액트 (React)

- SPA(Single-page application)
  - 하나의 HTML 페이지에 애플리케이션 실행에 필요한 JavaScript와 CSS 같은 모든 자산을 로드하는 애플리케이션
  - 웹 사이트 전체 페이지를 하나의 페이지에 담아 동적으로 화면을 바꿔가며 표현하는 것
    - 페이지 또는 후속 페이지의 상호작용은 서버로부터 새로운 페이지를 불러오지 않으므로 페이지가 다시 로드되지 않음
- React를 사용하여 싱글 페이지 애플리케이션을 만들 수 있지만, 필수 사항은 아님

웹 사이트에 React 추가

- 기존 페이지에 리액트 추가하는 방법

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8" />
  <title>Hello World</title>
  <script src="https://unpkg.com/react@18/umd/react.development.js"></script>
  <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

  <!-- Don't use this in production: -->
  <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
</head>

<body>
  <div id="root"></div>

  <script type="text/babel">
    class Hello extends React.Component {
      render() {
        return React.createElement('h1', null, `Hello ${this.props.toWhat}`);
      }
    }

    const root = ReactDOM.createRoot(document.getElementById('root'));
    root.render(React.createElement(Hello, { toWhat: 'World' }, null));
  </script>
</body>
</html>
```

2단계: 스크립트 태그 추가하기(React를 실행 및 만든 컴포넌트)

1단계: HTML 파일에 DOM 컨테이너 설치

3단계: React 컴포넌트 만들기

웹 사이트에 React 추가

- 기존 페이지에 리액트 추가하는 방법(JSX로 추가)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello World</title>
    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

    <!-- Don't use this in production: -->
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  </head>
  <body>
    <div id="root"></div>
    <script type="text/babel">

      function MyApp() {
        return <h1>Hello, world!</h1>;
      }

      const container = document.getElementById('root');
      const root = ReactDOM.createRoot(container);
      root.render(<MyApp />);

    </script>
  </body>
</html>
```

새로운 React 앱 만들기

- Create React App
  - 새로운 싱글 페이지 애플리케이션으로 React의 간편한 환경을 제공
  - 개발 환경을 설정하고, 최신 JavaScript를 사용하게 해주며, 좋은 개발 경험과 프로덕션 앱 최적화
- 설정
  - Node 설치 (Node 14.0.0 이상 , npm 5.6 이상 버전이 필요)
    - https://nodejs.org/en/

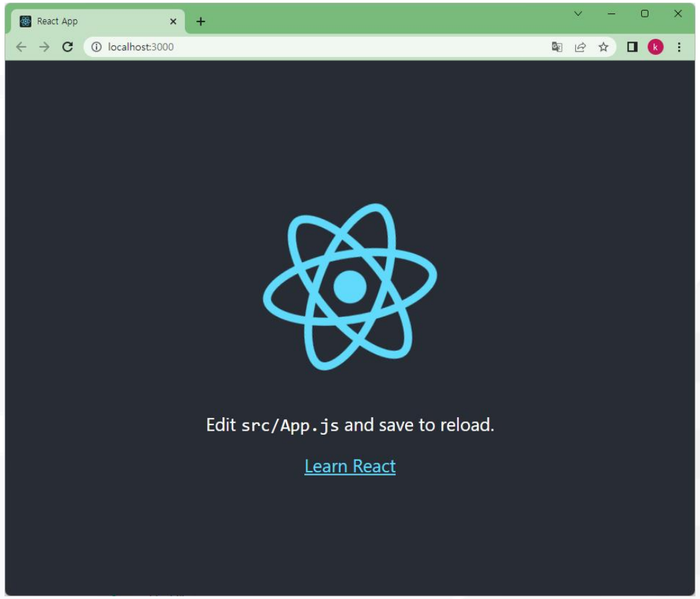
새로운 React 앱 만들기

```
PS C:\minNote\OneDrive - pusan.ac.kr\강의자료\2022_0919_K디지털\React\work> npx create-react-app ./my01

Creating a new React app in C:\minNote\OneDrive - pusan.ac.kr\강의자료\2022_0919_K디지털\React\work\my01.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...


added 1394 packages in 1m

210 packages are looking for funding
  run `npm fund` for details

Initialized a git repository.

Installing template dependencies using npm...

added 56 packages in 9s

210 packages are looking for funding
  run `npm fund` for details
  npm audit fix --force

Run `npm audit` for details.

Created git commit.

Success! Created my01 at C:\minNote\OneDrive - pusan.ac.kr\강의자료\2022_0919_K디지털\React\work\my01
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  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:

    cd my01
    npm start
```

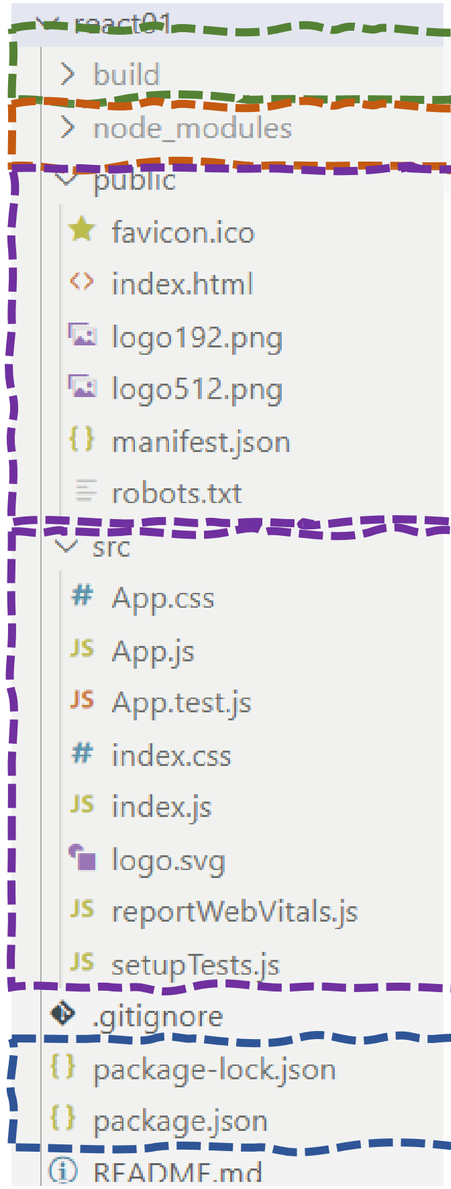
```
Creating a new React app in C:\minNote\OneDrive - pusan.ac.kr\강의자료\2022_0919_K디지털\React\work\my01.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...


added 1394 packages in 1m

210 packages are looking for funding
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
Compiled successfully!

You can now view my01 in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.137.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
```

프로젝트명은 소문자로 작성

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



create-react-app 프로젝트 폴더



- 빌드를 한 경우에 생성 : **npm run build** 후 **npx serve -s build**로 실행
- **node_modules: React** 앱을 실행하기 위해 설치한 **Node** 모듈 저장
- **public:** 웹 브라우저에 실제로 보이는 정적 파일**(static files)** 저장
- **src:** 모든 컴포넌트**(components)** 파일 저장**; UI** 조각 저장
- **package-lock.json: npm**이 **node_modules** 또는 **package.json**을 수정할 때 디펜던시 버전 정보(버전명) 저장
- **package.json:** 프로젝트에 대한 메타데이터**(metadata)** 및 디펜던시 버전 정보(범위) 저장

create-react-app Web Vitals

- 웹 퍼포먼스 측정도구로 web-vitals 라는 라이브러리를 사용

index.js

src > index.js > ...

```
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, pass a function
15 // to log results (for example: reportWebVitals(console.log))
16 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
17 reportWebVitals(console.log);
```

```
// metric(측정도구) 이름
name: 'CLS' | 'FCP' | 'FID' | 'LCP' | 'TTFB';
// 측정된 현재값 (값이 작을수록 빠른성능을 뜻합니다)
value: number;
//  현재 측정값(current value)과 최신 측정값(last-reported value) 차이
//  첫번째 리포트에서 위 둘값은 항상 같습니다.
delta: number;
// 특정 측정도구를 나타대는 유니크한 ID 값으로 중복되는 값들을 관리할 때 사용된다.
id: string;
// 계산된 측정값들의 내용들이 배열로 나열 된다.
// ex) PerformanceNavigationTiming, LargestContentfulPaint
entries: (PerformanceEntry | FirstInputPolyfillEntry | NavigationTimingPolyfillEntry)[];
```

web-vitals.js:1

```
{name: 'FCP', value: 138.5, delta: 138.5, entries: Array(1), id: 'v2-1674642413892-9772824035677'}
  delta: 138.5
▶ entries: [PerformancePaintTiming]
  id: "v2-1674642413892-9772824035677"
  name: "FCP"
  value: 138.5
▶ [[Prototype]]: Object
```

web-vitals.js:1

```
{name: 'TTFB', value: 4.800000011920929, delta: 4.800000011920929, entries: Array(1), id: 'v2-1674642413893-5419766597049'}
  delta: 4.800000011920929
▶ entries: [PerformanceNavigationTiming]
  id: "v2-1674642413893-5419766597049"
  name: "TTFB"
  value: 4.800000011920929
▶ [[Prototype]]: Object
```

web-vitals.js:1

```
▶ {name: 'LCP', value: 190.5, delta: 190.5, entries: Array(2), id: 'v2-1674642413892-5795453229467'}
```

web-vitals.js:1

```
▶ {name: 'CLS', value: 0, delta: 0, entries: Array(0), id: 'v2-1674642413892-4993133043576'}
```

web-vitals.js:1

```
▶ {name: 'FID', value: 3.2000000029802322, delta: 3.2000000029802322, entries: Array(1), id: 'v2-1674642413892-2665784556651'}
```

public > index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>
```

React의 결과물이 들어가는 부분으로
src 폴더의 index.js파일을 수정

- 웹 브라우저가 렌더링하는 HTML 문서
  - 이 문서가 웹 브라우저에서 렌더링된 결과물이 바로 SPA에서 말하는 단 1개의 페이지

React 페이지

Public > index.html

```html
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>
```

src > index.js

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

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

src > App.js

```jsx
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          현재시간 : { new Date().toLocaleTimeString() }
        </p>
      </header>
    </div>
  );
}

export default App;
```

엘리먼트(Element)

- **엘리먼트는 리액트 앱의 가장 작은 단위**
  - **리액트 엘리먼트는 불변객체**
    - 엘리먼트를 생성한 이후에는 해당 엘리먼트의 자식이나 속성을 변경할 수 없음

./src/app.js

```
import './App.css';

function App() {
  return (
    <div>
      현재 시간 : {new Date().toLocaleTimeString()}
    </div>
  );
}

export default App;
```

컴포넌트를 가지고 와서 엘리먼트로 생성

./src/index.js

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

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

reportWebVitals();
```

렌더링 후 생성된 엘리먼트는 변경안됨

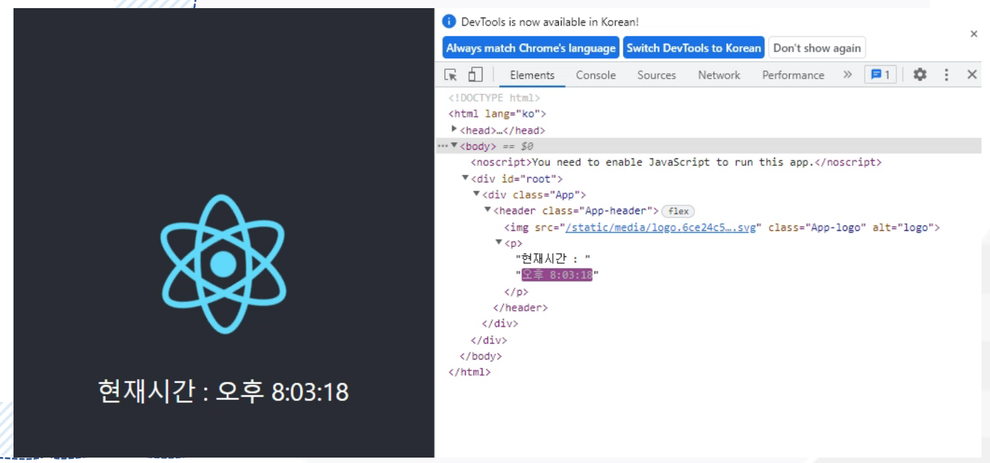
```
<!DOCTYPE html>
<html lang="ko">
▶ <head>…</head>
▼ <body>
    <noscript>You need to enable JavaScript to run this app.
    </noscript>
  ▼ <div id="root">
    ▼ <div>
        "현재 시간 : "
        "오후 12:51:08" == $0
      </div>
    </div>
  </body>
</html>
```

엘리먼트 랜더링

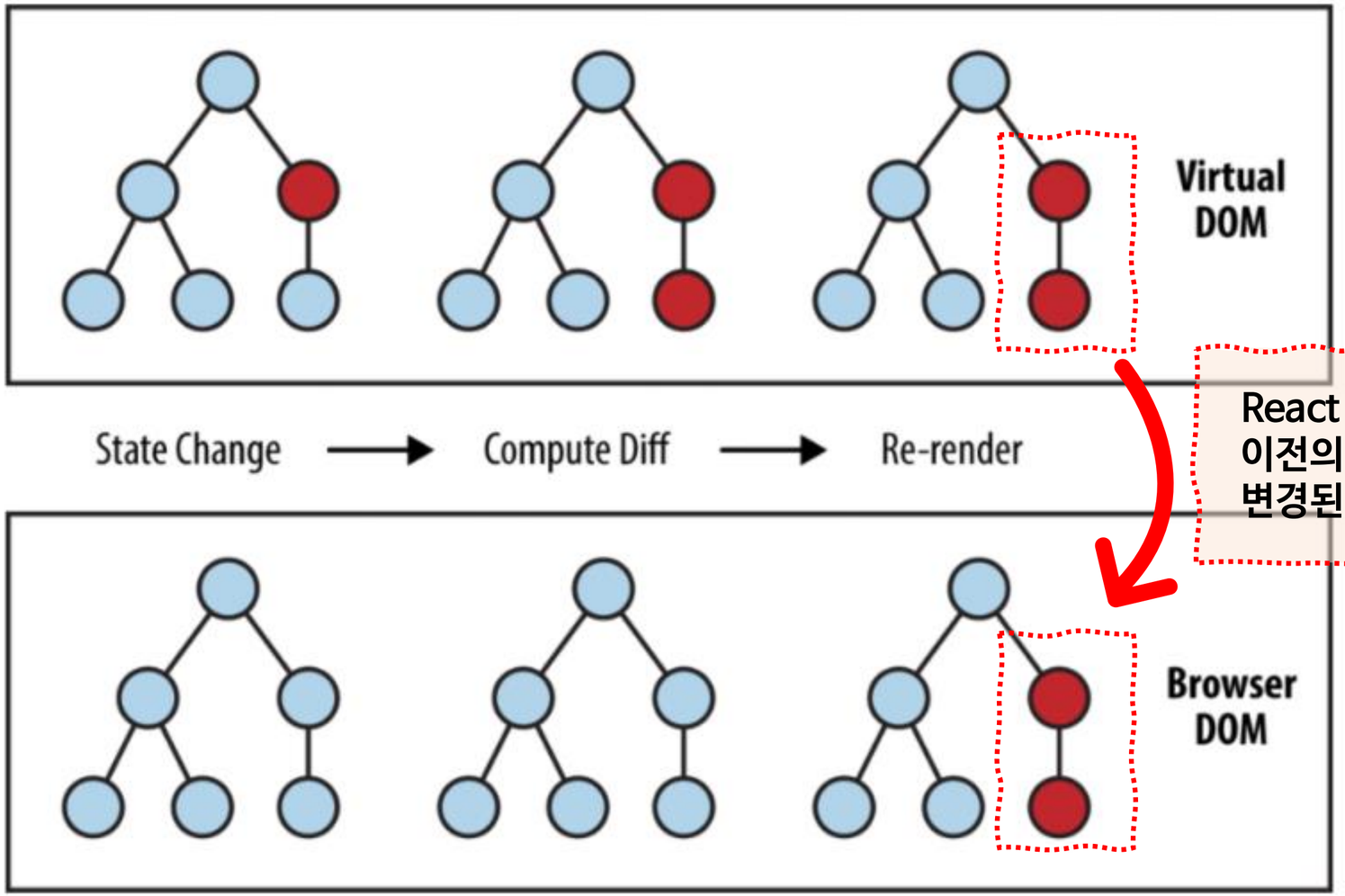
- 랜더링 된 엘리먼트 업데이트하기
  - 새로운 엘리먼트를 생성하고 root.render()로 전달

src > index.js

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

const root = ReactDOM.createRoot(document.getElementById('root'));


//초에 한번씩 재 랜더링
setInterval(() => root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
), 1000) ;
```



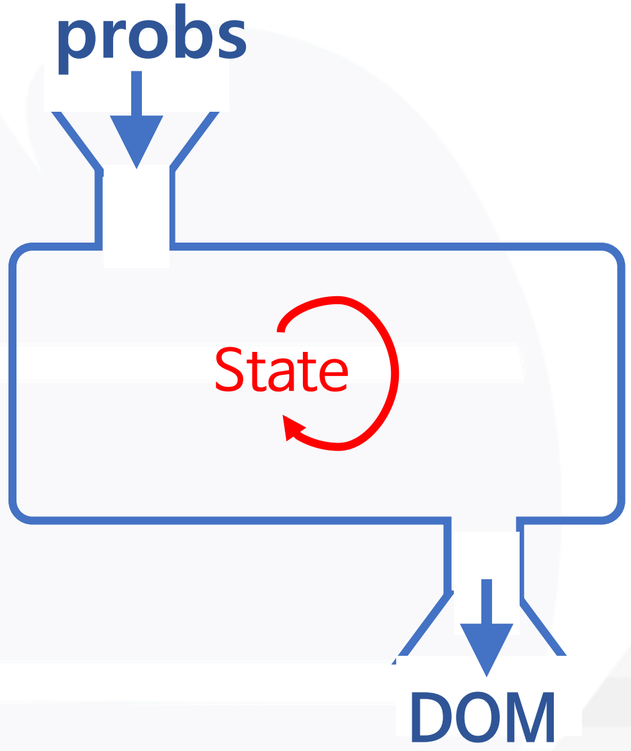
리액트 가상돔(Virtual DOM)



React DOM은 해당 엘리먼트와 그 자식 엘리먼트를
이전의 엘리먼트와 비교하고
변경된 부분만 실제 DOM에 업데이트

컴포넌트(Componet)

- UI를 재사용 가능한 개별적인 조각으로 사용자 정의 태그 생성
  - props라고 하는 임의의 입력을 받은 후, 화면에 어떻게 표시되는지를 기술하는 React 엘리먼트를 반환
  - 컴포넌트는 반드시 하나의 요소를 반환
  - 여러 요소가 있다면 프레그먼트로 감싸서 반환(〈〉…〈/〉)
  - 컴포넌트 명은 반드시 대문자로 시작해야 함
  - JSX 문법으로 작성
    - **JSX**내에 자바스크립트 표현식은 **{}**안에 작성
    - 하이픈은 카멜케이스로 표시해야함
      - background-color => backgroundColor
    - **class** 속성은 **className**으로 사용
    - 태그는 반드시 닫아야 함
    - 주석 : **{/* */}**



JSX(JavaScript XML)

- 자바스크립트에서 XML을 추가한 확장형 문법
  - React “엘리먼트(element)” 를 생성
  - JSX 사용이 필수가 아니지만, 대부분의 사람은 JavaScript 코드 안에서 UI 관련 작업을 할 때 시각적으로 더 도움이 됨
- 표현식 포함
  - JSX 중괄호 안에 유효한 모든 JavaScript 표현식 작성가능
  - const element = ⟨h1⟩Hello, {name}⟨/h1⟩;
- 주입 공격을 방지
  - React DOM은 JSX에 삽입된 모든 값을 렌더링하기 전에 이스케이프 하므로, 애플리케이션에서 명시적으로 작성되지 않은 내용은 주입되지 않음으로 XSS (cross-site-scripting) 공격을 방지
- 객체를 표현
  - Babel은 JSX를 React.createElement() 호출로 컴파일

해결문제

- 영화진흥위원회 일일 박스오피스 API서비스 정보를 이용하여 object변수를 만들고 다음과 같이 일일 박스 오피스 정보가 나타나도록 작성하시오.

박스오피스

1	교섭	(-)
2	더 퍼스트 슬램덩크	(-)
3	아바타: 물의 길	(↑)
4	유령	(↓)
5	영웅	(-)
6	장화신은 고양이: 끝내주는 모험	(-)
7	오늘 밤, 세계에서 이 사랑이 사라진다 해도	(-)
8	라일 라일 크로커다일	(-)
9	캐리와 슈퍼콜라	(-)
10	스위치	(-)


```
▼(10) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
  ▼0:
     audiAcc: "265460"
     audiChange: "35.5"
     audiCnt: "91475"
     audiInten: "23961"
     movieCd: "20190808"
     movieNm: "교섭"
     openDt: "2023-01-18"
     rank: "1"
     rankInten: "0"
     rankOldAndNew: "OLD"
     rnum: "1"
     salesAcc: "2605573954"
     salesAmt: "937793286"
     salesChange: "44.1"
     salesInten: "286984810"
     salesShare: "27.8"
     scrnCnt: "1281"
     showCnt: "6064"
   ▶[[Prototype]]: Object
  ▶1: {rnum: '2', rank: '2', rankInten: '0', rankOldAndNew: 'OLD', movieCd: '20228555', …}
  ▶2: {rnum: '3', rank: '3', rankInten: '1', rankOldAndNew: 'OLD', movieCd: '20225061', …}
  ▶3: {rnum: '4', rank: '4', rankInten: '-1', rankOldAndNew: 'OLD', movieCd: '20214823', …}
  ▶4: {rnum: '5', rank: '5', rankInten: '0', rankOldAndNew: 'OLD', movieCd: '20196478', …}
  ▶5: {rnum: '6', rank: '6', rankInten: '0', rankOldAndNew: 'OLD', movieCd: '20224109', …}
  ▶6: {rnum: '7', rank: '7', rankInten: '0', rankOldAndNew: 'OLD', movieCd: '20228313', …}
  ▶7: {rnum: '8', rank: '8', rankInten: '0', rankOldAndNew: 'OLD', movieCd: '20229166', …}
  ▶8: {rnum: '9', rank: '9', rankInten: '0', rankOldAndNew: 'OLD', movieCd: '20228895', …}
  ▶9: {rnum: '10', rank: '10', rankInten: '0', rankOldAndNew: 'OLD', movieCd: '20215315', …}
   length: 10
  ▶[[Prototype]]: Array(0)
```

여러 개의 컴포넌트 렌더링

- 여러 개의 엘리먼트를 배열에 저장하고 중괄호를 이용하여 JSX에 포함할 수 있음
  - 자바스크립트 map()함수를 사용

```
import './Box.css'

const Box = () => {
  const dailyBoxOfficeList = [{"rnum":"1","rank":"1","rankInten":"0","rankOldAndNew":"(
    {"rnum":"10","rank":"10","rankInten":"0","rankOldAndNew":"OLD","movieCd":"20215315'

  console.log(dailyBoxOfficeList)
  const mvs = dailyBoxOfficeList.map((mv) =>
    <div className='divRow' key={mv.movieCd}>
      <span className='divCol1'>{mv.rank}</span>
      <span className='divCol2'>{mv.movieNm}</span>
      <span className='divCol3'>({parseInt(mv.rankInten)===0 ? '🟦' : (parseInt(mv.r
    </div>
  );

  return (
    <>
    <div className='content'>
      <h1>박스오피스</h1>
      <div className='divMv'>
      {mvs}
      </div>
    </div>
    </>
  );
}

export default Box;
```