

AI 활용 빅데이터분석 풀스택웹서비스 SW 개발자 양성과정

React



부산대학교 소프트웨어교육센터
PUSAN NATIONAL UNIVERSITY SOFTWARE EDUCATION CENTER



리액트 Hook

- 함수 컴포넌트에서 React state와 생명주기 기능(lifecycle features)을 “연동(hook into)”할 수 있게 해주는 함수
 - React 버전 16.8부터 React 요소로 새로 추가
 - 상태 관련 로직을 추상화해 독립적인 테스트와 재사용이 가능해 레이어 변화 없이 재사용
 - 기존의 라이프사이클 메서드 기반이 아닌 로직 기반으로 나눌 수 있어서 컴포넌트를 함수 단위로 잘게 쪼갤 수 있다는 이점
- Hook 사용 규칙
 - 최상위에서만 Hook을 호출
 - 반복문, 조건문, 중첩된 함수 내에서 Hook을 실행하면 안됨
 - 이 규칙을 따르면 컴포넌트가 렌더링될 때마다 항상 동일한 순서로 Hook이 호출되는 것이 보장
 - 리액트 함수 컴포넌트에서만 Hook을 호출
 - JS함수에서는 Hook을 호출해서는 안됨

리액트 Hook

- 함수 컴포넌트에서 React state와 생명주기 기능(lifecycle features)을 “연동(hook into)”할 수 있게 해주는 함수
 - React 버전 16.8부터 React 요소로 새로 추가
 - 상태 관련 로직을 추상화해 독립적인 테스트와 재사용이 가능해 레이어 변화 없이 재사용
 - 기존의 라이프사이클 메서드 기반이 아닌 로직 기반으로 나눌 수 있어서 컴포넌트를 함수 단위로 잘게 쪼갤 수 있다는 이점

Hook : useState

- 컴포넌트 state

- 컴포넌트가 가지는 상태값으로 상태값이 변경되면 화면이 재 랜더링

- useState

- 동적으로 변경되는 값을 관리할 때 사용하며 상태 유지값과 그값을 갱신하는 함수를 반환
 - `const [state, setState] = useState(initialState);`

```
import { useState } from "react";
```

1. useState import

```
let [cnt, setCnt] = useState(0);
```

2. state 변수 선언

useState()는 배열 반환 => 구조분해 할당

```
const addCnt = () => {  
  setCnt(++cnt);  
  console.log(cnt);  
}
```

3. state 변수 값 변경

버튼을 누르면 함수 호출

```
<div className="divbt">  
  <button onClick={addCnt}>❤️ </button>  
  <span>{cnt}</span>  
</div>
```

4. state 변수 값이 변경되면 재랜더링으로 DOM에 반영

부모 state변수 자식에게 전달

순위	영화명	매출액	관객수	증감율
1	파묘	3,425,663,683원	344,954명	-
2	듀: 파트2	999,638,885원	81,802명	-
3	웬카	205,239,775원		
4	가여운 것들	96,761,307원		
5	브레드이발소: 셸럽 인 베이커리타운	78,874,350원		
6	패스트 라이브즈	63,657,712원		
7	랜드 오브 배드	54,673,073원		
8	용감한 돌고래 벨루와 바닷속 친구들	32,544,720원	4,144명	↑16
9	밥 말리: 원 러브	39,719,054원	4,081명	↓5
10	건국전쟁	36,644,851원	3,812명	-

[20236180] 월카 개봉일 : 2024-01-31 누적관객수 : 3,445,016명

```
<BoxOfficeTbody
  dailyBoxOfficeList={dailyBoxOfficeList}
  setSelectedMv={setSelectedMv} />
```

State의 setSelected함수도 props로 전달

Hook : useEffect

- 컴포넌트 내에서 렌더링이 수행된 이후 실행되는 메서드
- `useEffect(() => {}, dependency값)`

```
//useEffect : dependency가 없을 경우  
useEffect(()=>{  
  console.log('dependency가 없을 경우', pn) ;  
}) ;
```

```
//useEffect : dependency가 빈배열  
useEffect(()=>{  
  console.log('dependency가 빈배열 경우', pn) ;  
}, []) ;
```

```
//useEffect : dependency배열에 값이 있을 경우  
useEffect(()=>{  
  console.log('dependency배열에 값이 있을 경우', pn) ;  
}, [pn]) ;
```


Hook : useEffect cleanup

- 컴포넌트가 unmount되어 DOM에서 제거될 경우 실행되는 메소드
`useEffect(() => { ... return ()=>{} }, dependency값)`

```
import { useState, useEffect } from "react";

function MyClockTime() {
  const [currentTime, setCurrentTime] = useState(new Date());

  useEffect(() => {
    const tm = setInterval(() => {
      setCurrentTime(new Date());
    }, 1000);

    return () => {
      clearInterval(tm);
    };
  }, []);

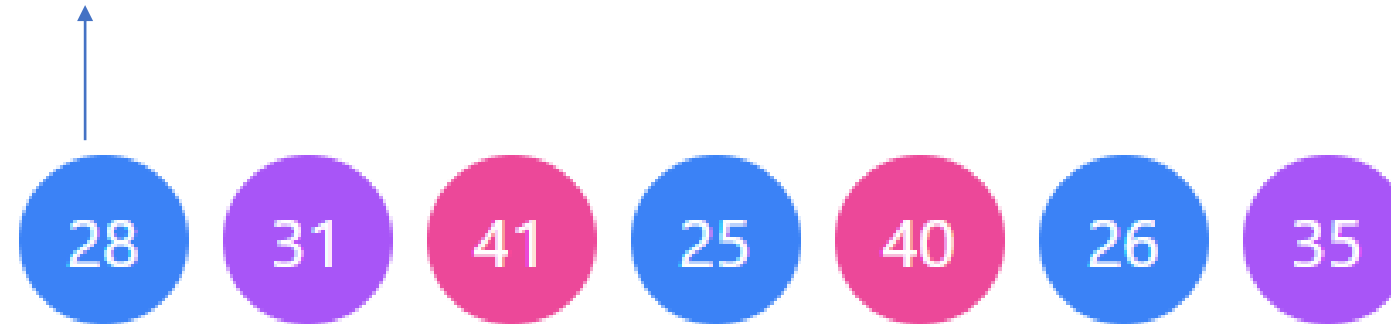
  return (
    <h1>
      현재 시각 : {currentTime.toLocaleTimeString()}
    </h1>
  )
}

export default MyClockTime ;
```

컴포넌트 제거 시 cleanup 처리

해결문제

```
<TailBall n={n} />
```

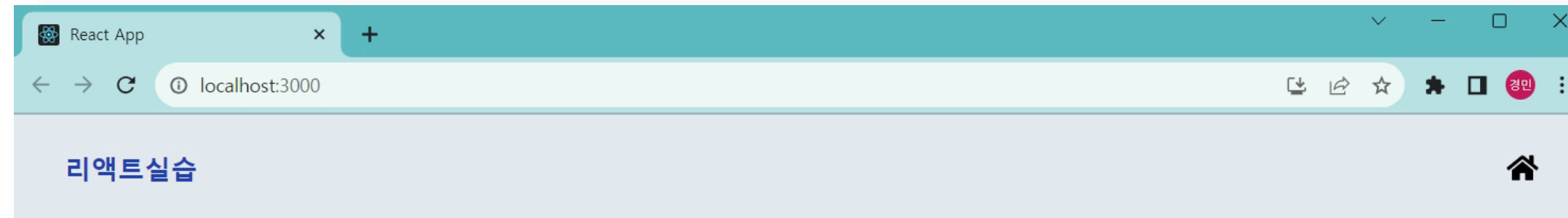


로또번호생성

```
<TailButton color='blue' caption='로또번호생성' onClick={handleClick} />
```

<Lotto />

해결문제



1. 사회복지법인

2. 재단법인

2. 재단법인

3. 사단법인

8. 비영리 민간단체

Fetch API

- JavaScript에서 HTTP 요청을 보내고 응답을 받는 기능을 제공하는 API
 - Promise 기반으로 동작
 - fetch 함수를 사용하여 HTTP 요청을 보내고, 응답이 완료되면 Promise 객체를 반환
 - 응답은 Response 객체로 반환되며, 이 객체를 통해 응답의 상태와 데이터를 처리

```
let promise = fetch(url, [options])
```

- url - 접근하고자 하는 URL
- options - 선택 매개변수, method나 header 등을 지정할 수 있음
options에 아무것도 넘기지 않으면 요청은 GET 메서드로 진행

Fetch API

```
fetch(url)
  .then((resp) => resp.json())
  .then((data) => {
    const dailyBoxOfficeList = data.boxOfficeResult.dailyBoxOfficeList
    console.log(dailyBoxOfficeList)
  })
  .catch((err) => console.log(err));
```

성공

성공

실패

해결문제

- 공공데이터 포털 도로교통공단_사고유형별 교통사고 통계 데이터를 이용하여 다음과 같이 해당 자료를 출력

교통사고 대분류

차대사람

차대차

차량단독

철길건널목

교통사고 중분류

횡단중

차도통행중

길가장자리구역통행중

보도통행중

기타