

Hook

React Hook 이란?

리액트 혹은 리액트 클래스형 컴포넌트에서 이용하던 코드를 작성할 필요없이 함수형 컴포넌트에서 다양한 기능알 사용할 수 있게 만들어준 라이브러리라고 할 수 있는데 React 16.8 버전에 새로 추가된 기능이다. 이는 함수형 컴포넌트에 맞게 만들어진 것으로 함수형 컴포넌에서만 사용 가능하다.

Hook 규칙

최상위에서만 Hook 을 호출해야 한다.

- 반복문이나 조건문 호은 중첩된 함수 내에서 Hook 을 호출하면 안된다.
- 리액트 혹은 호출되는 순서에 의존하기 때문에 조건문이나 반복문 안에서 실행하게 될 경우 해당 부분을 건너뛰는 일이 발생할 수 있기 때문에 순서가 꼬여 버그가 발생할 수 있다.
- 그렇기 때문에 이 규칙을 따르면 useState 와 useEffect 가 여러번 호출되는 경우에도 Hook 의 상태를 올바르게 유지할 수 있게 된다.

리액트 함수 내에서만 Hook 을 호출해야 한다.

- Hoo 은 일반적인 js 함수에서는 호출하면 안된다.
- 함수형 컴포넌트나 costum hook 에서는 호출 가능하다.

useState

useState 는 상태를 관리하는 훅으로 다음과 같은 특징을 가진다.

- 함수형 컴포넌트 안에 state 를 추가하여 사용한다.
 - 현재 상태를 나타내는 state 값과 이 상태를 변경하는 setState 값을 한쌍으로 제공한다.
 - state 는 초기값을 설정할 수 있으며, 초기값은 첫 렌더링 때 한번 사용된다.
 - state 는 객체일 필요 없이 문자열, 숫자 boolean, 배열, null, 객체 등의 여러가지 다양한 값을 넣을 수 있다.
 - 반환값을 확인해보면 Object 로 넘어온다.
-

예제

index.js

- 아래 예제들로 App/ FirstApp 등등 차례대로 출력 할 예정이다.
- const root 의 경우 아래 index.html 에 있다.
- 결국 index.html 의 형태로 출력되는 것이다. index.js 는 그 틀을 가져와서 localhost port 로 전달해주는 역할인 느낌이다.

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import FirstApp from './components/FirstApp';
import SecondApp from './components/SecondApp';
import ThirdApp from './components/ThirdApp';
import FourthApp from './components/FourthApp';
import FifthApp from './components/FifthApp';

// const root = ReactDOM.createRoot(document.getElementById('root'));
root 라는 곳을 찾아보자 public 폴더의 index.html 에 있다. 즉, 결국
디자인은 html 파일에서 기입하고. idnex.js 에서 import 받아서 한번에
출력하는 것이다.
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
```

```

    <React.StrictMode>
      {/* <App /> */}
      {/* <App /> 주석처리 하면 화면 안나온다. export 한 것을 받아와서 결국
화면에 띄워주는 곳은 여기이다. */}
      {/* <FirstApp/> */}
      {/* <SecondApp></SecondApp> */}
      {/*<ThirdApp/>*/}
      {/*<FourthApp/>*/}
      <FifthApp/>
    </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();

```

index.html

- index.js 에서 호출한 root 가 여기있다.
- bootstrap 을 사용하려면 여기서 cdn 을 추가해줘야 한다.

```

<!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" />
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstra
p.min.css" rel="stylesheet">
    <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" />
    <!--
      manifest.json provides metadata used when your web app is
      installed on a
      user's mobile device or desktop. See
      https://developers.google.com/web/fundamentals/web-app-manifest/

```

```
-->
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
<!--
  Notice the use of %PUBLIC_URL% in the tags above.
  It will be replaced with the URL of the `public` folder during
the build.
  Only files inside the `public` folder can be referenced from
the HTML.

  Unlike "/favicon.ico" or "favicon.ico",
"%PUBLIC_URL%/favicon.ico" will
  work correctly both with client-side routing and a non-root
public URL.
  Learn how to configure a non-root public URL by running `npm
run build`.
-->
<title>React App</title>
</head>
<body>
  <noscript>You need to enable JavaScript to run this
app.</noscript>
  <div id="root"></div> <!-- 여기에 root 가 있네요 -->
  <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty
page.

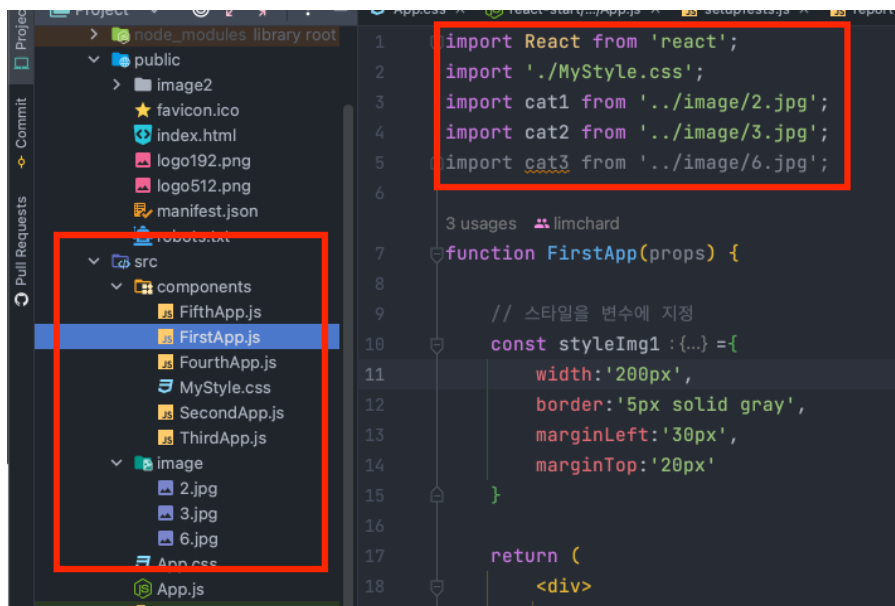
    You can add webfonts, meta tags, or analytics to this file.
    The build step will place the bundled scripts into the <body>
tag.

    To begin the development, run `npm start` or `yarn start`.
    To create a production bundle, use `npm run build` or `yarn
build`.
-->
  </body>
</html>
```

FirstApp.js

- *기본 React form 은 rsf 라는 단축키로 하여 생성해서 사용한다.*
- src 파일안에 있는 image 의 경우 하나하나 import 해서 사용해줘야 한다.

- import 지정이름 from '경로';
- 가져와서 사용할 때에는 {지정이름} 으로 사용 가능하다.
- style 같은 경우도 변수로 지정해서 사용 가능하다.
- style 을 작성할때에는 {width:'200px'} 와 같은 형식으로 기입해야 한다.
- public 폴더의 이미지는 import 없이 바로 경로로 호출 가능하다.
- css 도 import 를 사용해서 가져와 사용 가능하다.



출력문 원문

```
import React from 'react';
import './MyStyle.css';
import cat1 from '../image/2.jpg';
import cat2 from '../image/3.jpg';
import cat3 from '../image/6.jpg';
```

```
function FirstApp(props) {

  // 스타일을 변수에 지정
  const styleImg1={
    width:'200px',
    border:'5px solid gray',
    marginLeft:'30px',
    marginTop:'20px'
  }
}
```

```

    return (
      <div>
        {/* <h2 className='alert alert-warning'>FirstApp
Component!!!</h2> */}
        <h2 className='line2'>FirstApp Component!!!</h2>
        <div
style={{fontSize:'30px',marginLeft:'100px',backgroundColor:'salmon'}}
>
          안녕~~ 오늘은 목요일이야!!
        </div>
        {/* src 의 이미지는 import */}
        <img src={cat1} style={styleImg1}/>
        <img src={cat2} style={{width:'200px',border:'3px dotted
pink',marginLeft:'20px'}}/>

        {/* public image 는 직접 호출 가능합니다. */}

        <hr/>
        <h3> public image </h3>
        <img src='../image2/17.jpg'
style={{width:'250px',marginLeft:'20px'}}/>
        </div>
      </div>
    );
  }
  export default FirstApp;

```

FirstApp Component!!!

안녕~~ 오늘은 목요일이야!!



public image



SecondApp.js

- bootstrap 도 사용이 가능하다.
- 변수로 지정해놓으면 여러번 바로 호출이 가능하다.

```
import React from 'react';  
import cat4 from '../image/6.jpg'
```

```
function SecondApp(props) {

  const catStyle={
    width:'200px',
    marginLeft:'30px'
  }

  let helloEle=<h2 className='alert alert-info'>Hello~~</h2>

  return (
    <div>
      <h2 className='alert alert-danger'>SecondApp 입니다.</h2>
      <img src='../image2/19.jpg' style={catStyle}/>
      <img src={cat4} style={catStyle}/>
      {helloEle}
      {helloEle}
      {helloEle}
      {helloEle}
      {helloEle}
    </div>
  );
}

export default SecondApp;
```


SecondApp 입니다.



Hello~~

Hello~~

Hello~~

Hello~~

ThirdApp.js

- useState 라는 훅스를 쓰는 순간 위에 import React, { useState } from 'react';로 import 됨

엔터 이벤트 함수

```
const enterEvent=(e)=>{ // 엔터 이벤트 함수
  if(e.key==='Enter'){
    setMessage('');
    e.target.value='';
  }
}
// const enterEvent=(e)=>{
//   if(e.keyCode===13){
//     setMessage('');
//     e.target.value='';
//   }
// }
```

onChange 이벤트

- onChange 가 바로 변환되는 이벤트
- 이렇게 이벤트를 줄 수도 있고, 위에 함수 만들어서 호출하는 방법도 있음
- setMessage(e.target.value) : e 라는 인자값에 target 을 주는데 value 라는 것을 준다는 이벤트(메소드아님)
- 위에 message 를 setMessage 로 바꿔줄 수 있는 이벤트
- event.target.value
onChange 에 해당하는 인자값 e 를 target 향한다 value 로..결국 value 값이 다시 onChange 에 영향을 주어 무한루프를 도는 느낌이 된다. 그래서 입력 할때마다 실시간으로 반응한다.

```
onChange={e=>{
  console.log(e.target.value); // 인터넷 콘솔창 확인용
}}
```

```

    // message 변수에 입력값 넣기
    setMessage(e.target.value); // 실제 event
  }}

```

출력문 원문

```

import React, { useState } from 'react';

function ThirdApp(props) {

  // 상태를관리 위한 변수설정
  const [message,setMessage]=useState('기본값');

  // 엔터 이벤트 함수 (Enter 도 가능하고, 13 도
  가능하다)!!](https://velog.velcdn.com/images/limchard/post/fde9f424-
  d73b-4919-898b-ac0ddda308fd/image.gif)

  const enterEvent=(e)=>{ // 엔터 이벤트 함수
    if(e.key==='Enter'){
      setMessage('');
      e.target.value='';
    }
  }
  // const enterEvent=(e)=>{
  //   if(e.keyCode===13){
  //     setMessage('');
  //     e.target.value='';
  //   }
  // }

  return (
    <div>
      <h3 className='alert alert-dark'>ThirdApp 입니다.</h3>
      <h3 style={{color:'red'}}>{message}</h3>
      <h4>메시지를 입력해 주세요</h4>
      <input className='form-control' type='text'
        style={{width:'300px',fontSize:'2em'}}
        defaultValue={message}
        onChange={(e)=>{
          console.log(e.target.value);
          // message 변수에 입력값 넣기
          setMessage(e.target.value);
        }}
        onKeyDown={enterEvent}
      />
    </div>
  )
}

```

```
export default ThirdApp;
```

ThirdApp 입니다.

기본값

메시지를 입력해 주세요

기본값

FourthApp.js

- 클릭 이벤트 추가

```
import React, { useState } from 'react';
import cat from '../image/3.jpg'
```

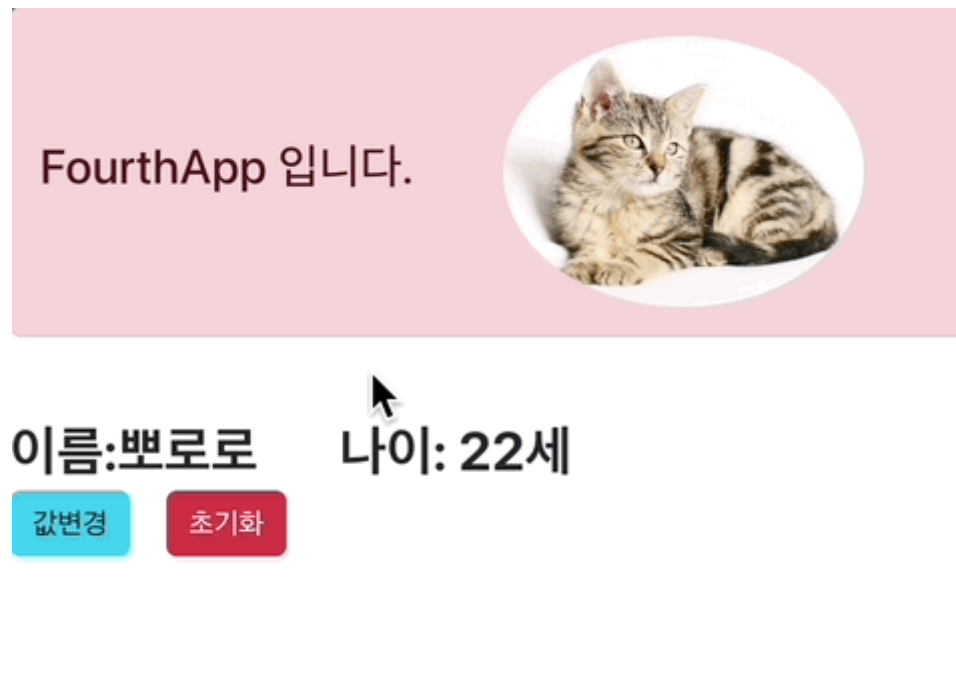
[illegible]

```

        <button type='button' className='btn btn-info'
onClick={()=>{
            setName("리처드");
            setAge(25);
        }}>값변경</button>
        <button type='button' className='btn btn-danger'
style={{marginLeft:"20px"}}
onClick={()=>{
            setName("");
            setAge('');
        }}>초기화</button>
    </div>
    );
}

export default FourthApp;

```



FifthApp.js (종합예제)

```

import React, {useState} from 'react';

function FifthApp(props) {

    const [name,setName]=useState('');
    const [resultName,setResultName]=useState('');

```

```

const [java,setJava]=useState(0);
const [react,setReact]=useState(0);
const [total,setTotal]=useState(0);
const [avg,setAvg]=useState(0);

return (
  <div>
    <h2 className='alert alert-danger' >FifthApp 입니다.</h2>
    <div className={'inp'}>
      이름: <input type={"text"} style={{width:"150px"}}
onChange={(event)=>{
        setName(event.target.value);
      }}/>
      <span style={{marginLeft:"10px"}}>{name}</span>
      <br/><br/>
      java 점수: <input type={"text"} style={{width:"150px"}}
onChange={(event)=>{
        setJava(event.target.value);
      }}/>
      <span style={{marginLeft:"10px"}}>{java}</span>
      <br/><br/>
      react 점수: <input type={"text"}
style={{width:"150px"}} onChange={(event)=>{
        setReact(event.target.value);
      }}/>
      <span style={{marginLeft:"10px"}}>{react}</span>
      <br/><br/>
      <button type={"button"} className={'btn btn-outline-
info'} onClick={()=>{
        // setTotal(parseInt({react})+parseInt({java}));
        setResultName(name);
        setTotal(parseInt(react)+parseInt(java));
        setAvg((parseInt(react)+parseInt(java))/2);
      }}>결과보기</button>
    </div>
    <div className='result'>
      이름 : {resultName} <br/>
      총합 : {total} 점 <br/>
      평균 : {avg} 점
    </div>
  </div>
);
}

```

```
export default FifthApp;
```

FifthApp 입니다.

이름:

java점수: 0

react점수: 0

결과보기

이름 :

총합 : 0 점

평균 : 0 점