

조건부 렌더링

흑함수 review

1. 입력값이 맞으면 '환영메시지' 출력, 틀리면 경고창 출력: useState
2. 입력창 나타날 때 아이디 입력창 포커싱되도록 할 것: useRef

로그인

아이디 비밀번호 로그인

환영합니다. [admin]님

로

localhost:3000 내용:

아이디 또는 비밀번호가 잘못되었습니다.

aa

확인

훅 함수 review

```
function Login() {  
  const [username, setUsername] = useState(""); // 사용자 이름 입력  
  const [password, setPassword] = useState(""); // 비밀번호 입력  
  const [isLoggedIn, setIsLoggedIn] = useState(false); // 로그인 여부  
  const inputRef = useRef(""); // 포커싱 위해서 입력요소 저장  
  
  useEffect(() => {  
    inputRef.current.focus();  
  }, []);  
  
  const handleLogin = () => {  
    if (username === "admin" && password === "1234") {  
      setIsLoggedIn(true);  
    } else {  
      alert("아이디 또는 비밀번호가 잘못되었습니다.");  
      setUsername("");  
      setPassword("");  
      inputRef.current.focus();  
      setIsLoggedIn(false);  
    }  
  };  
}
```

후함수 review

```
return (  
  <div>  
    {isLoggedIn ? (  
      <h2>{`환영합니다. [{username}]님`}</h2>  
    ) : (  
      <div>  
        <h2>로그인</h2>  
        <input ref={inputRef} type="text" placeholder="아이디" value={username} onChange={(e) => setUsername(e.target.value)} />  
        <input type="password" placeholder="비밀번호" value={password} onChange={(e) => setPassword(e.target.value)} />  
        <button onClick={handleLogin}>로그인</button>  
      </div>  
    )}  
  </div>  
)  
);  
}
```

조건부 렌더링이란

- 조건에 따라서 렌더링이 달라지는 것

```
01 function UserGreeting(props) {  
02   return <h1>다시 오셨군요!</h1>;  
03 }
```

} 회원에게 보여주는 코드

```
04  
05 function GuestGreeting(props) {  
06   return <h1>회원가입을 해 주세요.</h1>;  
07 }
```

} 비회원에게 보여주는 코드



props로 전달되는 값에 따라 화면 출력이 달라짐

```
01 function Greeting(props) {  
02   const isLoggedIn = props.isLoggedIn;  
03   if (isLoggedIn) {  
04     return <UserGreeting />;  
05   }  
06   return <GuestGreeting />;  
07 }
```

조건부 렌더링이란

index.js에서 Greetings 컴포넌트 호출

```
root.render(  
  <React.StrictMode>  
    <Greetings isLoggedIn={true} />  
  </React.StrictMode>  
);
```

조건부 렌더링

- 엘리먼트 변수

- 렌더링 해야하는 컴포넌트를 변수처럼 다루고자 할 때 사용하는 변수

회원가입해주세요

로그인

다시 오셨군요.

로그아웃

버튼 클릭하였을 때, 상태 변수가 true/false 로 변경 되고, 이에 따른 버튼 엘리먼트를 리턴하고자 할 때 엘리먼트 변수에 저장하여 다룸

```
function LoginControl(props) {  
  const [isLoggedIn, setIsLoggedIn] =  
    useState(false);
```

```
  const handleLoginClick = () => {  
    setIsLoggedIn(true);  
  };
```

```
  const handleLogoutClick = () => {  
    setIsLoggedIn(false);  
  };
```

let button; 엘리먼트 변수

```
  if (isLoggedIn) {  
    button = <LogoutButton  
      onClick={handleLogoutClick} />;  
  } else {  
    button = <LoginButton  
      onClick={handleLoginClick} />;  
  }
```

```
  return (  
    <div>  
      <Greetings isLoggedIn={isLoggedIn} />  
      {button}  
    </div>  
  );  
}
```

조건부 렌더링

```
01  function LoginButton(props) {  
02      return (  
03          <button onClick={props.onClick}>  
04              로그인  
05          </button>  
06      );  
07  }  
08  
09  function LogoutButton(props) {  
10      return (  
11          <button onClick={props.onClick}>  
12              로그아웃  
13          </button>  
14      );  
15  }
```


조건부 렌더링

- 인라인 조건문
 - If문을 필요한 곳에 직접 집어 넣어서 사용하는 방법
 - 논리 연산자 AND 또는 OR 사용

```
true && expression -> expression
```

```
false && expression -> false
```

조건부 렌더링

- JSX 코드 안에서 `&&` 연산자를 사용.
 - `{}` 안에 넣는다.

```
01 function Mailbox(props) {  
02     const unreadMessages = props.unreadMessages;  
03  
04     return (  
05         <div>  
06             <h1>안녕하세요!</h1>  
07             {unreadMessages.length > 0 &&  
08                 <h2>  
09                     현재 {unreadMessages.length}개의 읽지 않은 메시지가 있습니다.  
10                 </h2>  
11             }  
12         </div>  
13     );  
14 }
```

조건부 렌더링

- &&연산자 사용시 주의사항
 - Falsy expression의 결과값이 그대로 리턴된다.
 - 예) count가 0이므로 false가 되고, 그 결과 <div>0</div> 리턴됨.

```
01  function Counter(props) {  
02      const count = 0;  
03  
04      return (  
05          <div>  
06              {count && <h1>현재 카운트: {count}</h1>}  
07          </div>  
08      );  
09  }
```

조건부 렌더링

- Falsy(교재 273 참조)

- falsy

- false
- 0, -0 (zero, minus zero)
- 0n (BigInt zero)
- '', '', `` (empty string)
- null
- undefined
- NaN (not a number)

조건부 렌더링

- 인라인 조건문
 - if-else: 삼항연산자 적용

조건문 ? 참일 경우 : 거짓일 경우

```
01  function UserStatus(props) {  
02      return (  
03          <div>  
04              이 사용자는 현재 <b>{props.isLoggedIn ? '로  
    그인' : '로그인하지 않은'}</b> 상태입니다.  
05          </div>  
06      )  
07  }
```

조건부 렌더링

- 삼항연산자 적용 예

```
12 let button;
13 if (isLoggedIn) {
14     button = <LogoutButton onClick={handleLogoutClick} />;
15 } else {
16     button = <LoginButton onClick={handleLoginClick} />;
17 }
18
19 return (
20     <div>
21         <Greeting isLoggedIn={isLoggedIn} />
22         {button}
23     </div>
24 )
25 }
```

```
01 function LoginControl(props) {
02     const [isLoggedIn, setIsLoggedIn] = useState(false);
03
04     const handleLoginClick = () => {
05         setIsLoggedIn(true);
06     }
07
08     const handleLogoutClick = () => {
09         setIsLoggedIn(false);
10     }
11
12     return (
13         <div>
14             <Greeting isLoggedIn={isLoggedIn} />
15             {isLoggedIn
16                 ? <LogoutButton onClick={handleLogoutClick} />
17                 : <LoginButton onClick={handleLoginClick} />
18             }
19         </div>
20     )
21 }
```

조건부 렌더링

- 특정 컴포넌트를 렌더링하고 싶지 않을 때에는 null 리턴한다.

```
01  function WarningBanner(props) {  
02      if (!props.warning) {  
03          return null;  
04      }  
05  
06      return (  
07          <div>경고!</div>  
08      );  
09  }
```

조건부 렌더링

- WarningBanner 컴포넌트 렌더링 막기 위해 null 리턴

```
01 function MainPage(props) {  
02     const [showWarning, setShowWarning] = useState(false);  
03  
04     const handleToggleClick = () => {  
05         setShowWarning(prevShowWarning => !prevShowWarning);  
06     }  
07  
08     return (  
09         <div>  
10             <WarningBanner warning={showWarning} />  
11             <button onClick={handleToggleClick}>  
12                 {showWarning ? '감추기' : '보이기'}  
13             </button>  
14         </div>  
15     )  
16 }
```

보이기

경고! 감추기

조건부 렌더링

- WarningBanner 컴포넌트 렌더링 막기 위해 null 리턴

```
const handleToggleClick = () => {  
    setShowWarning((prevShowWarning) => !prevShowWarning);  
};
```

```
(prevShowWarning) => !prevShowWarning)
```

: 콜백함수의 매개변수로 현재 상태값(showWarning)을 최신 상태를 저장하는 변수 prevShowWarning으로 전달, 콜백함수는 네거티브 값을 리턴함

실습: 로그인 여부 나타내는 툴바 만들기

로그인

소플과 함께하는 리액트 공부!

환영합니다!

로그아웃

소플과 함께하는 리액트 공부!

```
function LandingPage(props) {
  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const onClickLogin = () => {
    setIsLoggedIn(true);
  };

  const onClickLogout = () => {
    setIsLoggedIn(false);
  };

  return (
    <div>
      <Toolbar
        isLoggedIn={isLoggedIn}
        onClickLogin={onClickLogin}
        onClickLogout={onClickLogout}
      />
      <div style={{ padding: 16 }}>소플과 함께하는 리액트 공부!</div>
    </div>
  );
}
```

실습: 로그인 여부 나타내는 툴바 만들기

```
function Toolbar(props) {  
  const { isLoggedIn, onClickLogin, onClickLogout } = props;  
  
  return (  
    <div style={styles.wrapper}>  
      {isLoggedIn && <span style={styles.greeting}>환영합니다!</span>}  
  
      {isLoggedIn ? (  
        <button onClick={onClickLogout}>로그아웃</button>  
      ) : (  
        <button onClick={onClickLogin}>로그인</button>  
      )}  
    </div>  
  );  
}
```

실습: 로그인 여부 나타내는 툴바 만들기

```
const styles = {  
  wrapper: {  
    padding: 16,  
    display: "flex",  
    flexDirection: "row",  
    borderBottom: "1px solid grey",  
  },  
  greeting: {  
    marginRight: 8,  
  },  
};
```

실습: id, pwd 입력양식 추가

아이디	비밀번호	로그인
소플과 함께하는 리액트 공부!		

admin님 환영합니다!	로그아웃
소플과 함께하는 리액트 공부!	

aa	localhost:3000 내용: 아이디 또는 비밀번호가 일치하지 않습니다.
소플과 함께하는 리	확인

```
function Toolbar2(props) {
  const { isLoggedIn, onClickLogin, onClickLogout } = props;
  const [id, setId] = useState("");
  const [pwd, setPwd] = useState("");

  //correctId='admin', correctPwd='1234' 예시
  const correctId = "admin";
  const correctPwd = "1234";

  const handleLogin = () => {
  };

  const handleLogout = () => {
  };
}
```

실습: id, pwd 입력양식 추가

```
const handleLogin = () => {  
  if(id와 pwd 맞는가?{ onClickLogin() 호출 }  
  else {  
    “아이디와 비밀번호가 일치하지 않습니다.” 경고창 출력  
    입력창 초기화  
  }  
};  
  
const handleLogout = () => {  
  입력창 초기화  
  onClickLogout() 호출  
};  
return (  
  <div style={styles.wrapper}>  
  
  </div>  
);  
}
```