

6

2024 GETIT 5기 SW 교육(리액트) 6차시

조건부 렌더링, 라우팅



GETIT 5기 운영진 이은새



참고사항

해당 강의자료는 **MacOS**를 기준으로 제작되었습니다.

도움이 필요하거나 **질의사항**이 있으시다면
이은새 운영진에게 개인적으로 연락하거나
카카오톡 오픈채팅을 이용해주세요!

GETIT 5기 질문방 : <https://open.kakao.com/o/gc7jPrhg>

해당 강의자료는 [이인제-소플의 처음 만난 리액트]를 참고하여 제작되었습니다.

5기 부원 외의 강의자료 및 영상 배포는 원칙적으로 금지합니다.



목차

1. 조건부 렌더링
2. 라우팅



01

조건부 렌더링

1

조건부 렌더링이란?

- 말 그대로, 어떤 조건에 따라 렌더링이 달라지는 것을 말합니다.
- 쉽게 설명하면, 어떤 조건에 따라 화면에 보여지는 게 달라지는 것

조건부 렌더링이란?

App.js

```
import { useState } from "react";
import LoginState from "../components/LoginState";
function App() {
  const [isLogin, setIsLogin] = useState(false);

  const handleLogin = () => {
    setIsLogin(!isLogin);
  };

  return (
    <div>
      <LoginState isLogin={isLogin} />
      <button onClick={handleLogin}>{isLogin ? "로그아웃" : "로그인"}</button>
    </div>
  );
}

export default App;
```

LoginState

```
function LoginState({ isLogin }) {
  return isLogin ? <WelcomeText /> : <LoginText />;
}

function LoginText() {
  return <h1>로그인 해주세요</h1>;
}

function WelcomeText() {
  return <h1>안녕하세요! 어서오세요~</h1>;
}

export default LoginState;
```

- App.js 에 있는 버튼을 누르면 상태변수인 isLogin의 값이 반대로 바뀝니다.(true or false)
- 로그인 상태(isLogin)에 따라 버튼에 있는 텍스트가 바뀝니다.
- 로그인 상태에 따라 LoginState에서 보여주는 텍스트가 달라집니다.

조건부 렌더링이란?

로그인 해주세요

로그인



안녕하세요! 어서오세요~

로그아웃



로그인 해주세요

로그인

- 버튼을 누를 때마다 isLogin의 값이 바뀌면서, 렌더링되는 텍스트나 컴포넌트가 달라집니다.

2

인라인 조건문

- 조건문 자체를 코드에 넣는 것
- 어떤 조건문을 별도로 분리하지 않고, 조건문이 필요한 곳에 직접 넣어주는 방식의 조건문을 말합니다.
- 논리 연산자(`&&` AND)
 - '`&&`' 왼쪽에 있는 조건이 참이면, 오른쪽에 있는 것을 보여주는 방식
- 삼항 연산자(`A ? B : C`)
 - ? 왼쪽에 있는 조건문이 참이면 `->` : 왼쪽에 있는 것을 보여줌
 - ? 왼쪽에 있는 조건문이 거짓이면 `->` : 오른쪽에 있는 것을 보여줌

조건부 렌더링이란?

로그인 해주세요

로그인

안녕하세요! 어서오세요~

로그아웃

마이페이지

App.js

```
import { useState } from "react";
import LoginState from "../components/LoginState";
function App() {
  const [isLogin, setIsLogin] = useState(false);

  const handleLogin = () => {
    setIsLogin(!isLogin);
  };

  const MyPageButton = <button>마이페이지</button>;

  return (
    <div>
      <LoginState isLogin={isLogin} />
      <button onClick={handleLogin}>{isLogin ? "로그아웃" : "로그인"}</button>
      {isLogin && MyPageButton}
    </div>
  );
}

export default App;
```

- isLogin이 true, 즉 로그인이 되어 있는 상태일 때만 MyPageButton을 보여줌



02

라우팅

1

라우팅이란?

- 웹 애플리케이션에서 사용자의 요청에 따라 서로 다른 페이지나 뷰를 표시하는 것
- 브라우저에서 URL을 입력하거나 링크를 클릭하면 URL의 경로와 쿼리를 바탕으로 이에 해당하는 컴포넌트를 렌더링해줌
- 리액트에서는 React-Router를 사용해서 라우팅을 구현

2

React-Router, 왜 쓰지?

- SPA에서 사용자 경험을 향상시키기 위한 것
 - 뒤로가기 기능 : URL이 하나이면, 이전 페이지로 이동 불가
 - 새로고침 기능 : 새로고침 하면 처음 페이지 대신 현재 페이지에서
 - 페이지 특정 가능 : URL이 하나로 고정되면, 특정 페이지를 공유하거나 저장할 수 없음

3

react-router-dom 사용하기

1. 설치하기

- 터미널에 `npm install react-router-dom --save`

2. 불러오기

- `import { BrowserRouter, Routes, Route } from 'react-router-dom'`

react-router-dom 사용하기

3. 사용하기

- **BrowserRouter**
 - 라우팅에 대한 큰 틀. URL 기록들을 관리.
- **Routes**
 - Route의 상위 경로에 있음. 현재 location에 맞는 Route를 찾아줌
- **Route**
 - URL과 컴포넌트를 매핑하여 라우팅을 처리함
 - 어떤 URL 경로가 입력되면, 해당 경로에 맞는 컴포넌트를 렌더링

react-router-dom 사용하기

1. BrowerRouter

- index.js에서 App을 BrowserRouter로 감싸기

```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";
import reportWebVitals from "./reportWebVitals";
import { BrowserRouter } from "react-router-dom";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>
);

// 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();
```

react-router-dom 사용하기

2. Routes, Route

- 경로(path)와 그에 대해 렌더링할 컴포넌트(element)

```
import { Routes, Route } from "react-router-dom";
```

```
<div>  
  <Routes>  
    <Route path="/" element={<HomePage />} />  
    <Route path="/login" element={<LoginPage />} />  
    <Route path="/contents" element={<MyPage />} />  
  </Routes>  
</div>
```


react-router-dom 사용하기

3. useNavigate

- 버튼을 눌렀을 때 다른 페이지로 이동하게 하고 싶으면?
- useNavigate를 통해 다른 경로의 페이지로 이동가능

```
import { useNavigate } from "react-router-dom";
```

```
const navigate = useNavigate();

const goToHomePage = () => {
  navigate("/");
};

const goToLoginPage = () => {
  navigate("/login");
};

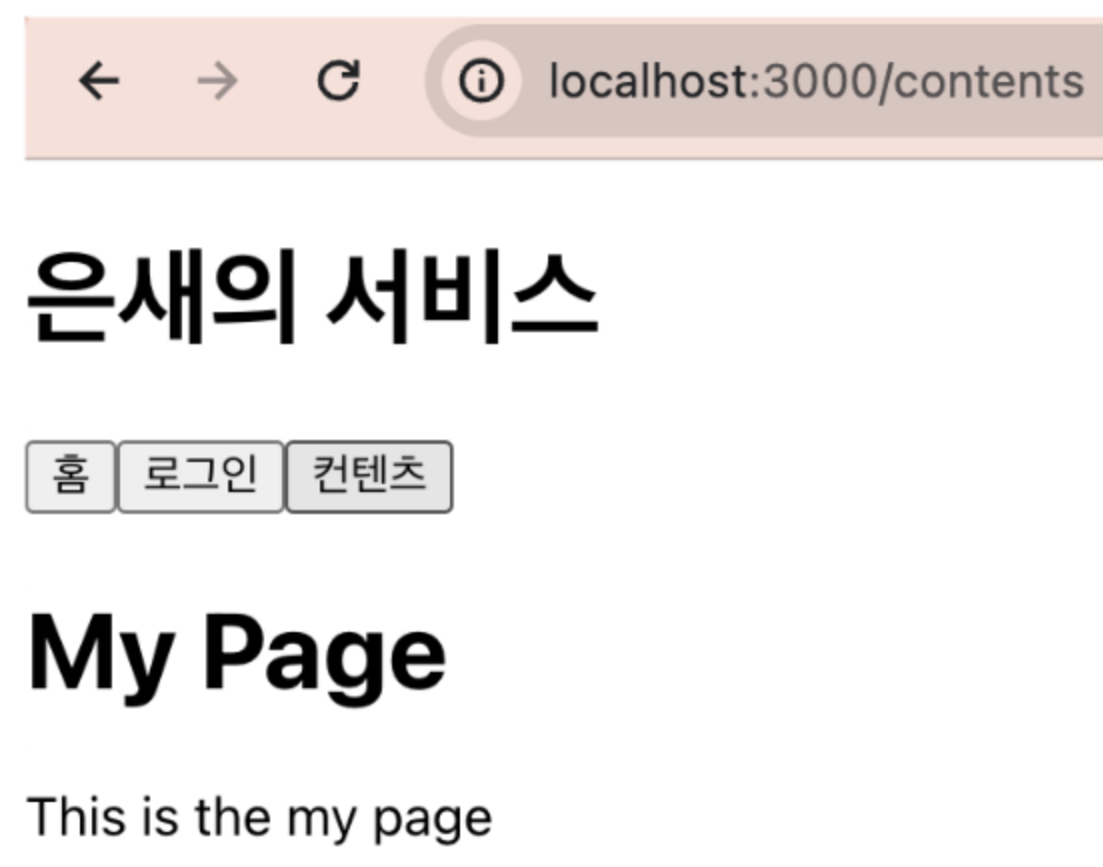
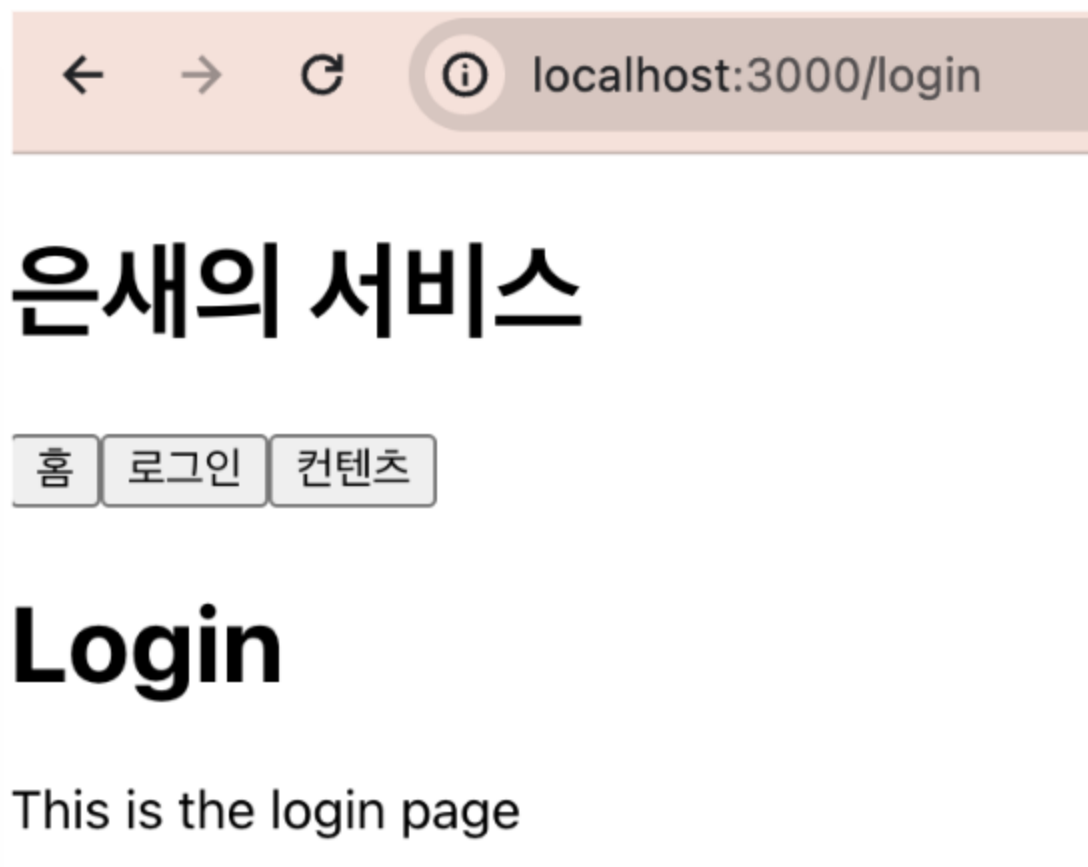
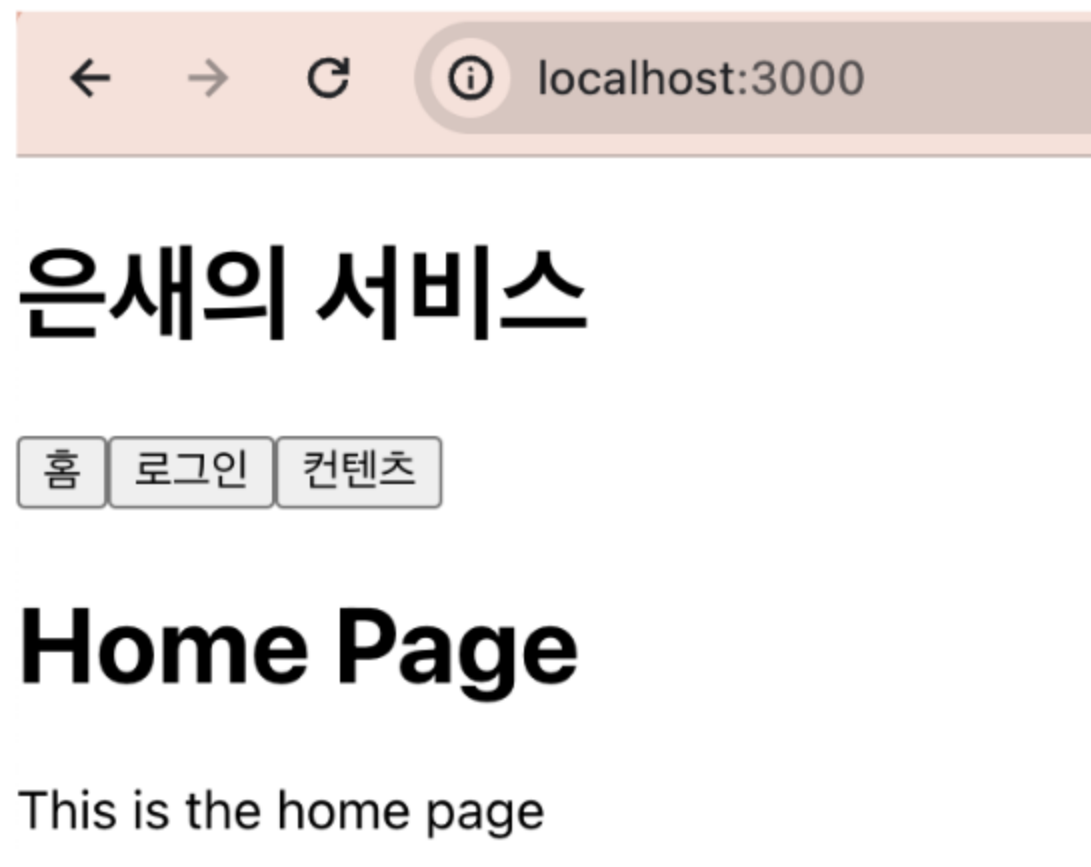
const goToMyPage = () => {
  navigate("/contents");
};
```

react-router-dom 사용하기

4. 버튼에 onClick 이벤트에 대한 이벤트 핸들러로 지정

```
return (
  <div>
    <div>
      <h1>내 서비스</h1>
      <div>
        <button onClick={goToHomePage}>홈</button>
        <button onClick={goToLoginPage}>로그인</button>
        <button onClick={goToMyPage}>컨텐츠</button>
      </div>
    </div>
    <div>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/login" element={<LoginPage />} />
        <Route path="/contents" element={<MyPage />} />
      </Routes>
    </div>
  </div>
);
```

react-router-dom 사용하기





GETIT 5기 SW 교육 6차시 과제

실습 결과 캡처 후 노선에 업로드하기

총 3장의 스크린샷을 Notion 페이지에 올려주세요!

