

AI 활용 빅데이터분석 풀스택웹서비스 SW 개발자 양성과정

React



부산대학교 소프트웨어교육센터
PUSAN NATIONAL UNIVERSITY SOFTWARE EDUCATION CENTER



실습문제

```
{
  "response": {
    "header": { ...
    },
    "body": {
      "dataType": "JSON",
      "items": {
        "item": [
          {
            "baseDate": "20221022",
            "baseTime": "0600",
            "category": "PTY",
            "nx": 97,
            "ny": 74,
            "obsrValue": "0"
          },
          {
            "baseDate": "20221022",
            "baseTime": "0600",
            "category": "REH",
            "nx": 97,
            "ny": 74,
            "obsrValue": "66"
          },
          {
            "baseDate": "20221022",
            "baseTime": "0600",
            "category": "RN1",
            "nx": 97,
            "ny": 74,
            "obsrValue": "0"
          },
          {
            "baseDate": "20221022",
            "baseTime": "0600",
            "category": "T1H",
            "nx": 97,
            "ny": 74,
            "obsrValue": "17"
          },
          {
            "baseDate": "20221022",
            "baseTime": "0600",
            "category": "UUU",
            "nx": 97,
            "ny": 74,
            "obsrValue": "-0.6"
          },
          {
            "baseDate": "20221022",
            "baseTime": "0600",
            "category": "VEC",
            "nx": 97,
            "ny": 74,
            "obsrValue": "152"
          },
          {
            "baseDate": "20221022",
            "baseTime": "0600",
            "category": "VVV",
            "nx": 97,
            "ny": 74,
            "obsrValue": "1.2"
          }
        ]
      }
    }
  }
}
```

날씨예보 - 단기정보	
강수형태	없음
습도	66%
1시간 강수량	0mm
기온	17°C
풍속(동서성분)	-0.6m/s
풍향	152deg
풍속(남북성분)	1.2m/s
풍속	1.4m/s

과학기술

기상청

활용신청

[승인] 기상청_중기예보 조회서비스

신청일 2022-09-21 만료예정일 2024-09-21

과학기술

기상청

활용신청

[승인] 기상청_단기예보 ((구)_동네예보) 조회서비스

신청일 2022-09-21 만료예정일 2024-09-21

```
{
  "response": {
    "header": { ...
    },
    "body": {
      "dataType": "JSON",
      "items": {
        "item": [
          {
            "regId": "11B00000",
            "rnSt3Am": 20,
            "rnSt3Pm": 20,
            "rnSt4Am": 0,
            "rnSt4Pm": 0,
            "rnSt5Am": 10,
            "rnSt5Pm": 10,
            "rnSt6Am": 0,
            "rnSt6Pm": 0,
            "rnSt7Am": 20,
            "rnSt7Pm": 0,
            "rnSt8": 20,
            "rnSt9": 30,
            "rnSt10": 30,
            "wf3Am": "구름많음",
            "wf3Pm": "구름많음",
            "wf4Am": "맑음",
            "wf4Pm": "맑음",
            "wf5Am": "맑음",
            "wf5Pm": "맑음",
            "wf6Am": "맑음",
            "wf6Pm": "맑음",
            "wf7Am": "구름많음",
            "wf7Pm": "맑음",
            "wf8": "구름많음",
            "wf9": "구름많음",
            "wf10": "구름많음"
          }
        ]
      }
    },
    "pageNo": 1,
    "numOfRows": 10,
    "totalCount": 1
  }
}
```

날씨예보 - 중기예보			
3일 후 날씨	4일 후 날씨	5일 후 날씨	6일 후 날씨
3일 후 오전 강수량 20%			
3일 후 오후 강수량 20%			
3일 후 오전 날씨예보 구름많음			
3일 후 오후 날씨예보 구름많음			

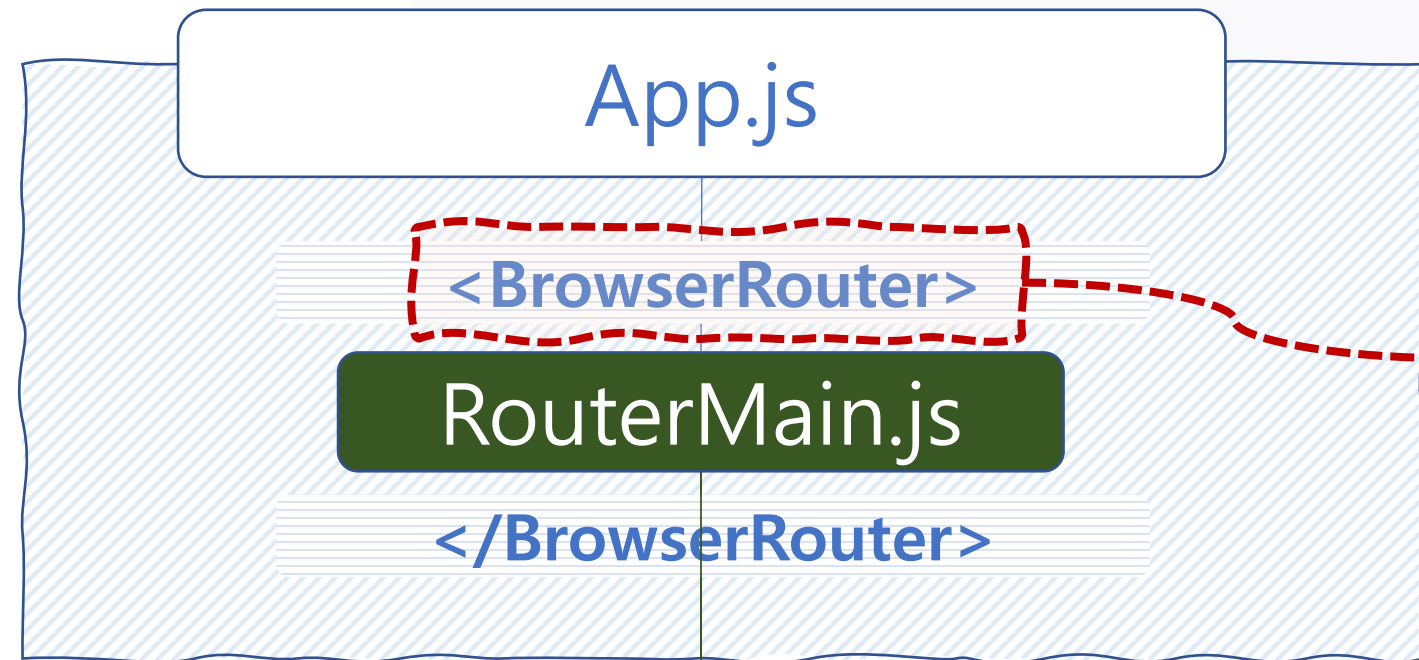
리액트 라우터

- 프로젝트 생성 후 라우터 설치
 - npm install react-router-dom

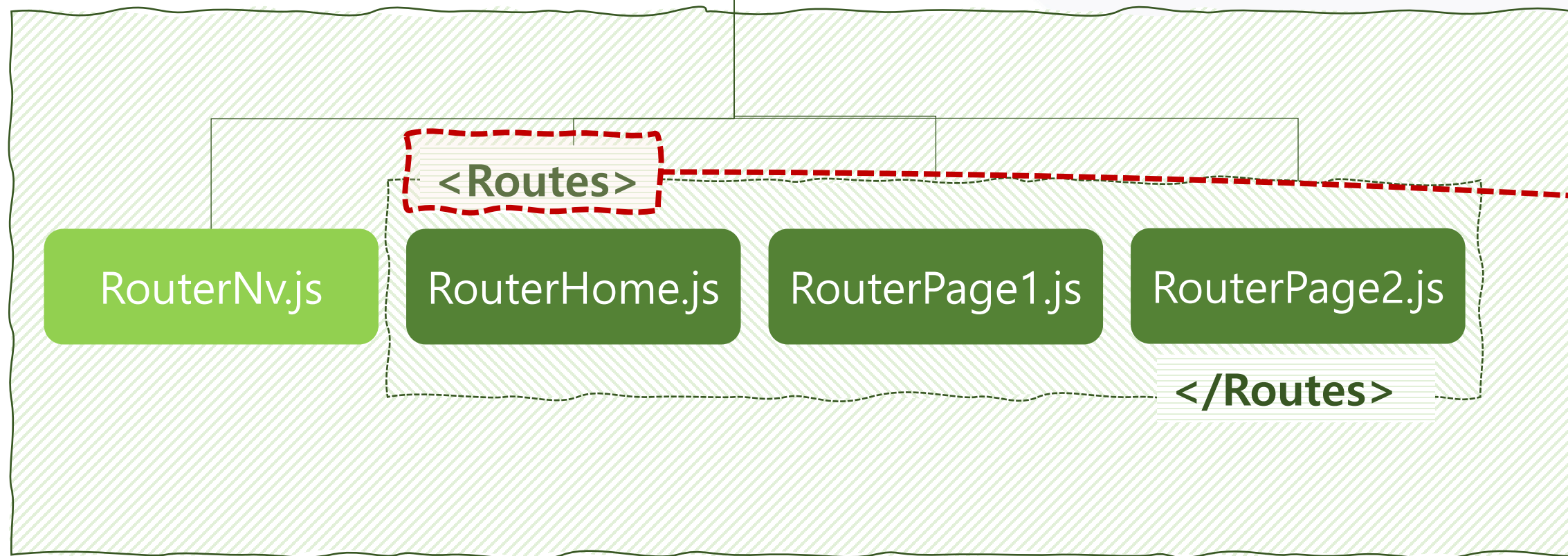
Package.json

```
{
  "name": "react01",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-router-dom": "^6.8.1",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
```

프로젝트에 라우터 적용

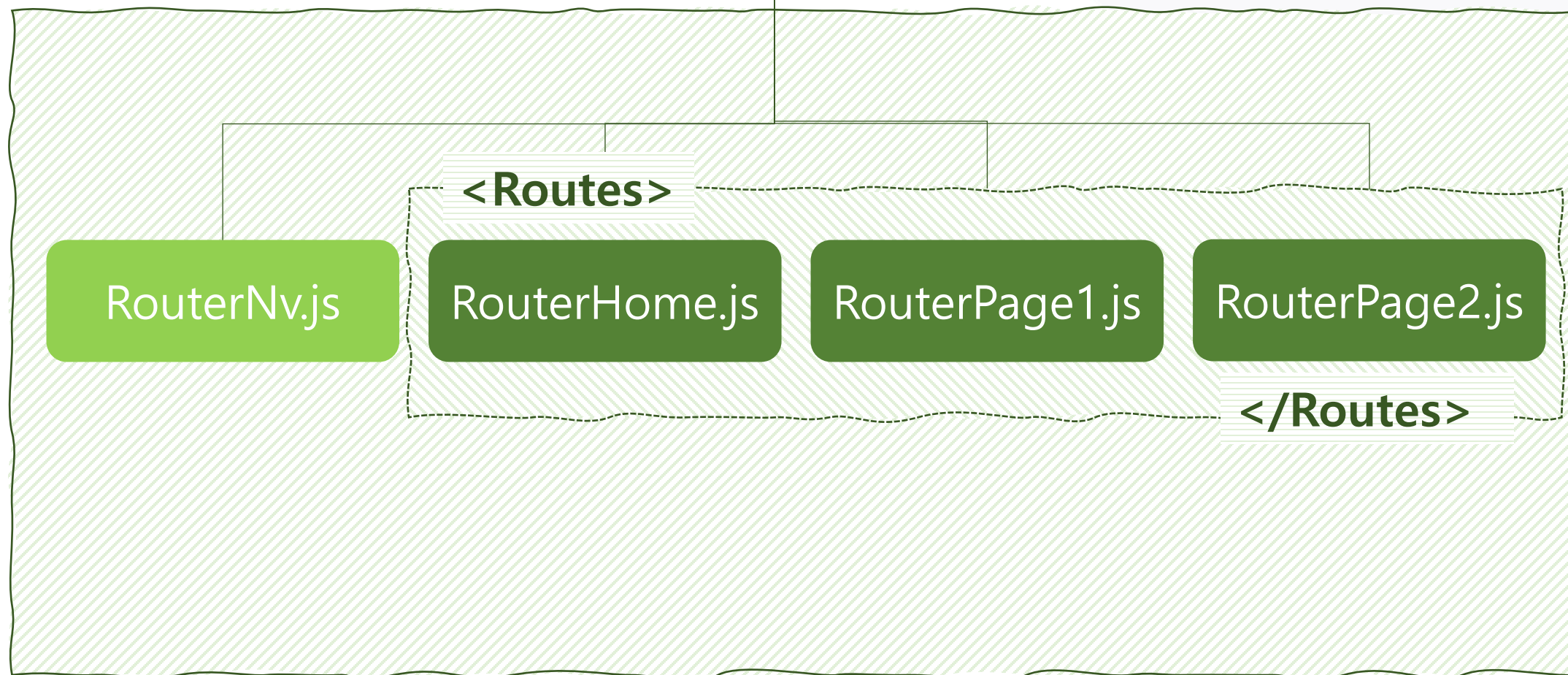
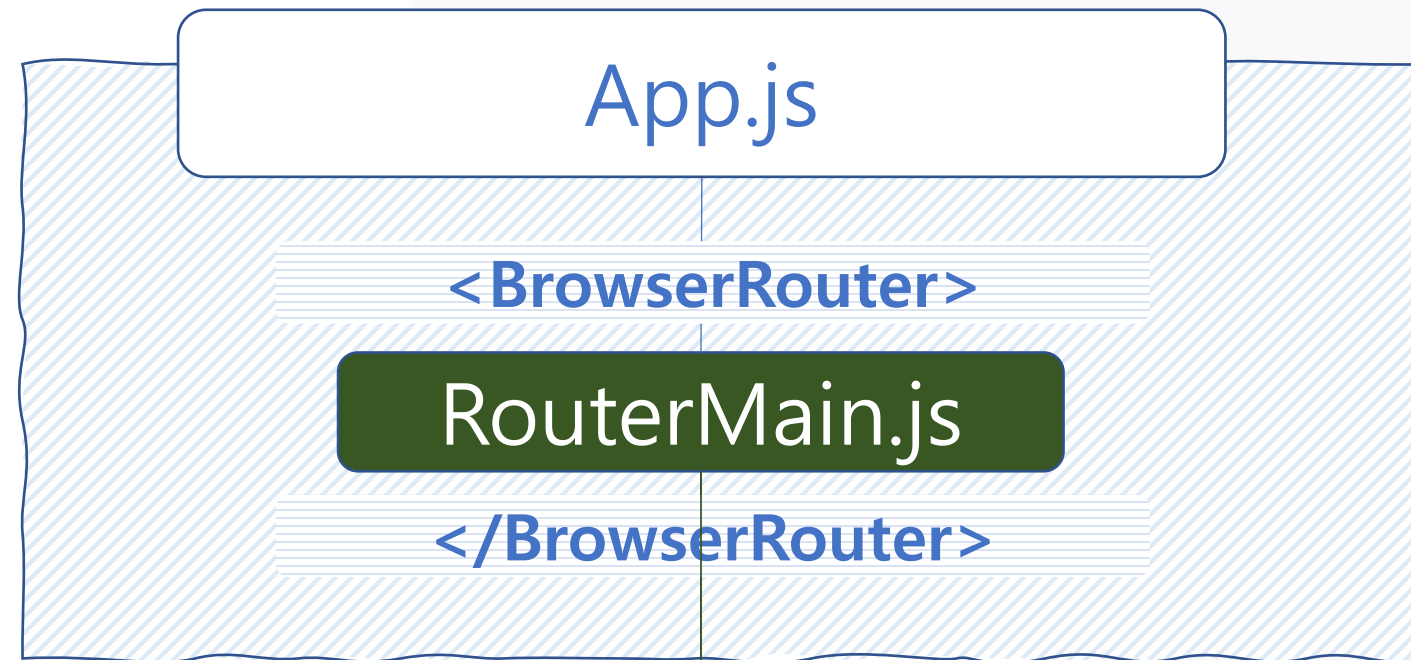


- `react-router-dom`에 내장되어 있는 `BrowserRouter`라는 컴포넌트를 사용하여 감싸줌
- 사용자가 입력한 주소를 감지하는 역할
- 이 컴포넌트는 웹 애플리케이션에 HTML5의 History API를 사용하여 페이지를 새로 불러오지 않고도 주소를 변경하고 현재 주소의 경로에 관련된 정보를 리액트 컴포넌트에서 사용할 수 있도록 함



- `Routes` 컴포넌트 내부에 `Route` 컴포넌트를 이용하여 사용자의 브라우저 주소 경로에 따라 보여주기 원하는 컴포넌트 설정

프로젝트에 라우터 적용



```
function App() {  
  return (  
    <BrowserRouter>  
      <RouterMain />  
    </BrowserRouter>  
  );  
}
```

```
const RouterMain = () => {  
  return (  
    <>  
      <RouterNv />  
      <Routes>  
        <Route path="/" element={<RouterHome />} />  
        <Route path="/page1" element={<RouterPage1 />} />  
        <Route path="/page2" element={<RouterPage2 />} />  
      </Routes>  
    </>  
  );  
}
```

Router 컴포넌트

• Route 컴포넌트로 특정 경로에 원하는 컴포넌트 보여주기

```
import { Routes, Route } from "react-router-dom";
import RouterNav from "../RouterNav";
import RouterHome from "../RouterHome";
import RouterPage1 from "../RouterPage1";
import RouterPage2 from "../RouterPage2";
```

```
const RouterMain = () => {
  return (
    <>
      <RouterNav />
      <Routes>
        <Route path="/" element={<RouterHome />} />
        <Route path="/page1" element={<RouterPage1 />} />
        <Route path="/page2" element={<RouterPage2 />} />
      </Routes>
    </>
  );
};
```

```
export default RouterMain;
```

Nav.js

공통으로 사용되는 영역은 Route
컴포넌트에 넣지 않음

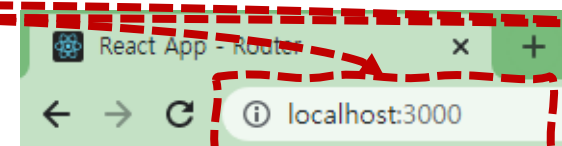
```
import { Link } from 'react-router-dom' ;

function Nav() {
  return (
    <nav>
      <ul>
        <li><Link to="/">홈이동</Link></li>
        <li><Link to="/page1">page1이동</Link></li>
        <li><Link to="/page2">page2이동</Link></li>
      </ul>
    </nav>
  );
}

export default Nav ;
```

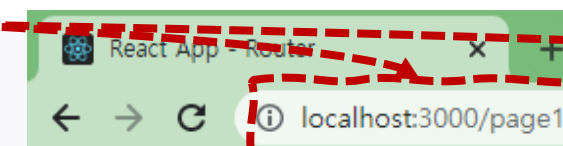
Link 컴포넌트를 사용하여 다른 페이지로 이동

Routes 컴포넌트 내부에 Route 컴포넌트를
이용하여 사용자의 브라우저 주소 경로에 따라
보여주기 원하는 컴포넌트 설정



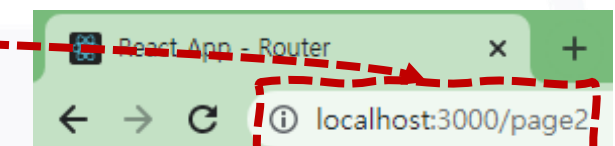
Home

RouterHome.js



Page1

RouterPage1.js



Page2

RouterPage2.js

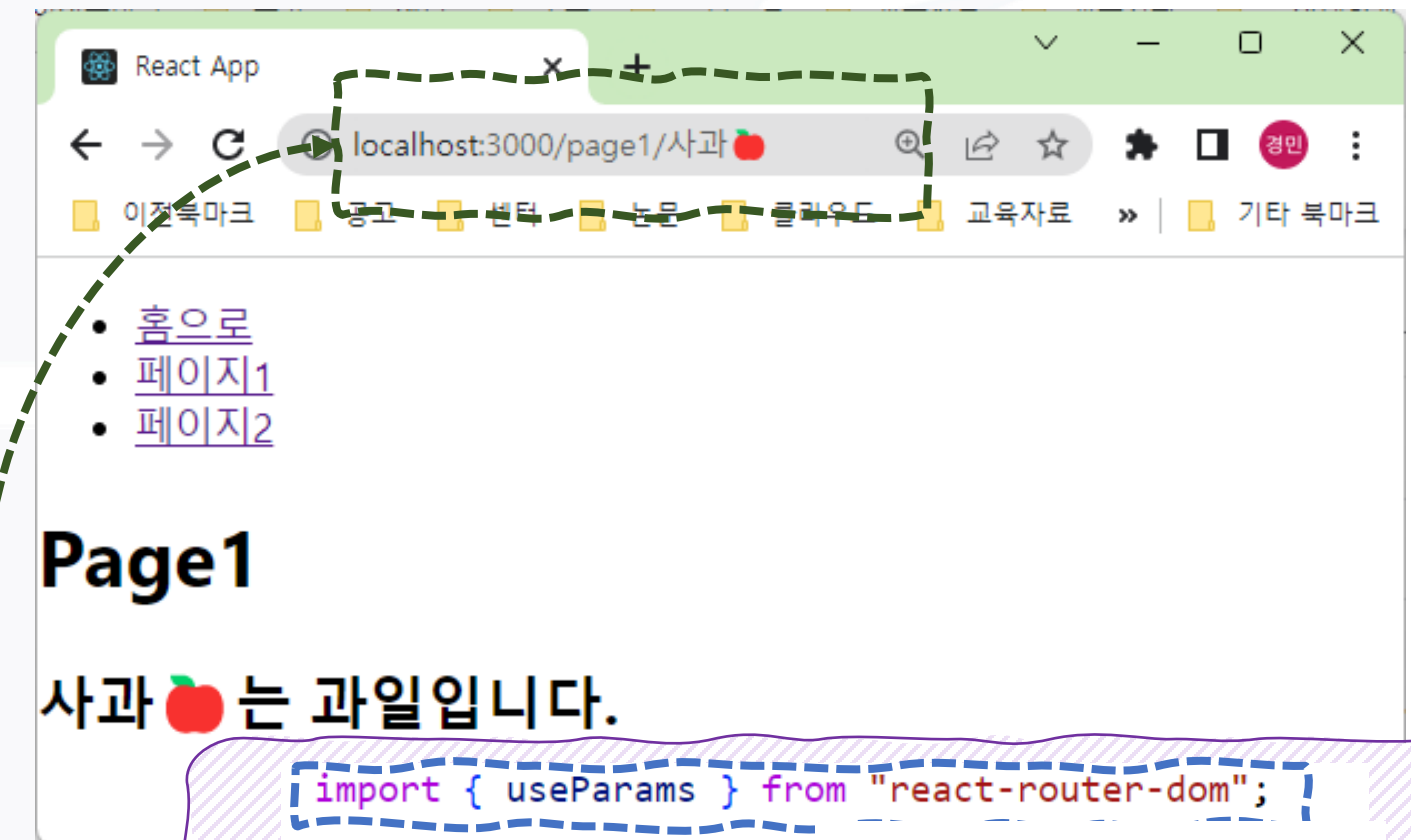
URL 파라미터

• 주소의 경로에 값을 넣어 전송

```
<Routes>
  <Route path="/" element={<RouterPage1 />} />
  <Route path="/page1/:item" element={<RouterPage1 />} />
  <Route path="/page2" element={<RouterPage2 />} />
</Routes>
```

URL 파라미터로 사용할 파라미터를 :뒤에 작성

```
<ul>
  <li><Link to="/page1/사과🍏">사과🍏</Link></li>
  <li><Link to="/page1/바나나🍌">바나나🍌</Link></li>
  <li><Link to="/page1/당근🥕">당근🥕</Link></li>
</ul>
```



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

```
const RouterPage1 = () : useParams 훅을 사용
```

```
const item = useParams().item ;
console.log(
```

URL 파라미터로 전달되는 값 받음

```
const fruits = ['사과🍏', '바나나🍌']
return (
  <>
    <h1>Page1</h1>
    {(fruits.includes(item))
      ? <h2>{item}는 과일입니다.</h2>
      : <h2>{item}는 과일이 아닙니다.</h2>}
  </>
);
```

쿼리스트링

- Route 컴포넌트에 별도 설정할 필요없이 `useLocation` 훅을 사용하면 전달된 쿼리스트링을 확인할 수 있음
- 쿼리스트링을 오브젝트로 변환하기 위해서는 `query-string` 패키지 설치
 - `npm install query-string`

localhost:3000/page2?item=사과🍏

- [홈으로](#)
- [페이지1](#)
- [페이지2](#)

Page2

사과🍏는 과일입니다.

```
import { useLocation } from "react-router-dom";
import qs from "query-string";

const RouterPage2 = () => {
  const location = useLocation().search;
  const item = qs.parse(location).item;
  console.log("page2 location", location);
  console.log("page2 item", item);

  const fruits = ['사과', '바나나', '딸기'];
  return (
    <>
    <h1>Page2</h1>
    {(fruits.includes(item))
      ? <h2>{item}는 과일입니다.</h2>
      : <h2>{item}는 과일이 아닙니다.</h2>}
    </>
  );
};
```

URL주소로 전달되는 값은 `useLocation` 훅을 이용하여 전달된 쿼리스트링을 받음

page2 location ?item=%EC%82%AC%EA%B3%BC%F0%9F%8D%8E
page2 item 사과🍏

이벤트 발생시 URL 이동

- useNavigate

–event가 발생할 때, url을 조작할 수 있는 interface

[page2 useNavigate 사용] 과일을 선택해 주세요.

사과 🍏 바나나 🍌 당근 🥕

← → ↻

localhost:3000/page2?item=사과 🍏

이전북마크

광고

센터

문헌

클라우드

표

- [홈으로](#)
- [페이지1](#)
- [페이지2](#)

Page2

사과 🍏 는 과일입니다.

```
import { Link } from 'react-router-dom';
import { useNavigate } from 'react-router-dom';

const RouterHome = () => {
  const navigate = useNavigate();
  const url = {
    '사과': '/page2?item=사과 🍏',
    '바나나': '/page2?item=바나나 🍌',
    '당근': '/page2?item=당근 🥕'
  };
  const goUrl = (item) => {
    navigate(url[item]);
  };
  return (
    <>
      <h1>Home</h1>
      <div>...
    </div>
      <div>...
    </div>
      <div>
        <h2>[page2 useNavigate 사용] 과일을 선택해 주세요.</h2>
        <button onClick={() => goUrl('사과')}>사과 🍏 </button>
        <button onClick={() => goUrl('바나나')}>바나나 🍌 </button>
        <button onClick={() => goUrl('당근')}>당근 🥕 </button>
      </div>
    </>
  );
};
```

해결문제

