React

Router

웹 렌더링

SPA

- **S**ingle **P**age **A**pplication
- **단일 웹페이지로** 돌아가는 애플리케이션 (index.html)
- 브라우저에서 자바스크립트를 이용해 단일 웹페이지 상의 HTML 요소를 동적으로 생성 및 조작
- 검색 엔진 최적화(SEO)에 적합하진 않다.
- React, Svelte, Vue.js와 같은 라이브러리로 개발 가능

참고! 2차 팀 프로젝트와 같은 사이트는 **MPA (Multi Page Application)** 이라고 부름!

: 여러 페이지를 만들고, 각 요청에 따라 적절한 페이지를 보여주는 방식

React Router

라우팅(Routing)이란?

=== 경로를 지정하는 행위!

- 사용자가 요청한 URL 에 따라 해당 URL 에 맞는 페이지를 보여주는 것
- 가장 많이 쓰이는 라이브러리? React Router

리액트 라우터 (React Router)

- 개발자가 **주소별로 다른 컴포넌트**를 보여주기 위해 사용하는 라이브러리
- 여러 환경에서 동작할 수 있도록 여러 종류의 라우터 컴포넌트 제공
- 주요 컴포넌트
  - **BrowserRouter**
  - **Routes**
  - **Route**
  - **Link**
  - **createBrowserRouter**

```
npm i react-router-dom@6 # 버전6 지정 설치
```

BrowserRouter

- 〈BrowserRouter〉
- HTML5를 지원하는 브라우저의 주소 감지
- Router의 역할
- 새로고침을 하지 않아도 새로운 컴포넌트를 렌더링 해주는 기능 담당
- URL마다 컴포넌트가 바뀔 때 이때 바뀌는 부분의 최상단에 위치해야 한다.

BrowserRouter

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

```
const ReactRouter = () => {
    return (
        <div>
            <BrowserRouter>
            </BrowserRouter>
        </div>
    );
};
```

Routes, Route

- 〈Routes〉, 〈Route〉
- 경로가 일치하는 컴포넌트를 렌더링해주도록 경로 매칭
- Route에서는 구체적으로 어떤 컴포넌트를 렌더링할지 결정

```
<Routes>
    <Route path="/" element={<RouteMain />}></Route>
    <Route path="/product/:id" element={<RouteProduct />}></Route>
    <Route path="*" element={<RouteNotFound />}></Route>
</Routes>
```

- **path** : 경로
- **element** : 연결할 컴포넌트

Link

- 〈Link〉
- 경로를 변경한다.
- 기존 HTML 의 a 태그가 새로고침 해 렌더링을 수행했다면 〈Link〉 컴포넌트는 페이지 전환을 방지한다.

```
<Link to="/">
    <h1>헤더입니다.</h1>
</Link>
```

1. URL 파라미터

- useParams()
- /product/:id
- 경로에 : 를 사용해 설정되는 것
- URL 파라미터가 여러 개인 경우 /product/:id/:name 처럼 설정

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

const { 파라미터명 } = useParams();
// const 변수명 = useParams().파라미터명;
```

2. URL 쿼리스트링

- **useSearchParams()**

```
localhost:3000/?mode=Dark
```

```
const [searchParams, setSearchParams] = useSearchParams();
```

```
searchParams.get('mode')
```

```
setSearchParams({ mode: 'Dark' });
```

3. 페이지 이동

- useNavigate()
  - **Link 컴포넌트를 사용하지 않고** 다른 페이지로 이동해야 하는 경우, 뒤로가기 등에 사용한다.

```
const navigate = useNavigate();
```

```
<li><button onClick={() => navigate(-2)}>Go 2 pages back</button></li>
<li><button onClick={() => navigate(-1)}>Go back</button></li>
<li><button onClick={() => navigate(1)}>Go forward</button></li>
<li><button onClick={() => navigate(2)}>Go 2 pages forward</button></li>
<li><button onClick={() => navigate('/')}>Go Root</button></li>
```