



[JavaScript,HTML,React]페이지 이동

≡ 작성자	FE 박효빈
≡ 주제	JS 페이지이동

JavaScript

window.location

- window.location.href

→ 현재 페이지에서 이동할 페이지로 이동(뒤로가기 가능)

→ 객체의 속성으로 작동

→ 주소 히스토리 기록O

```
function goToNewPage() {  
  window.location.href = "https://www.example.com";  
}
```

- window.location.replace :

→ 현재 페이지를 새로운 페이지로 변경시킨다. 즉 덮어씌운다.(뒤로가기 불가능)

→ 메서드(함수)로 작동

→ 주소 히스토리 기록X

```
function replacePage() {  
  window.location.replace("https://www.example.com");  
}
```

추가적으로 이러한 형태로도 사용 가능

```
<button onClick="window.location.href='https://www.example.com'">Go to Example.com</button>
<button onClick="window.location.replace('https://www.example.com')">Go to Example.com</button>
```

window.open

→ 새창에 열린다.

```
function replacePage() {
  window.location.open("https://www.example.com");
}
```

결론

- `window.location.href`: 사용자가 이전 페이지로 돌아갈 수 있어야 하고, 페이지를 이동할 때 자바스크립트 상태를 유지해야 할 때 적합합니다.
- `window.location.replace`: 사용자가 이전 페이지로 돌아가지 않아야 할 때 적합하며, 페이지를 이동할 때 자바스크립트 상태를 유지할 필요가 없을 때 사용됩니다.
- `window.open`: 새로운 창이나 탭을 열어서 별도의 작업을 수행할 때 사용하며, 현재 페이지를 유지한 채로 새로운 창에서 작업해야 할 때 적합합니다.

HTML

a태그

- 새로그침 되며 페이지 이동함.
- 열리는 방식에 대한 설정은 `target` 속성을 사용하여 지정할 수 있다. (자세한 건 mdn문서 참고 <https://developer.mozilla.org/ko/docs/Web/HTML/Element/a>)

```
<a href="https://www.example.com">Go to Example.com</a>
```

meta태그

- 페이지를 자동으로 이동시킬 때 사용. 사용자의 상호작용 없이 페이지 이동이 가능하다. (예를들어 페이지 점검중일때 meta태그로 설정해두면 자동으로 점검중입니다.를 보여주는 페이지를 보여줄 수 있다.)
- 사용방법 : content 속성에 이동할 url을 작성해주면 된다.

```
<meta http-equiv="refresh" content="5; url=https://www.example.com">
```

React

Link태그

- React 기반의 단일 페이지 애플리케이션에서 페이지 간의 이동을 관리하며, 새로고침 없는 페이지 전환을 제공한다.
- 주소 히스토리 저장
- React Router 라이브러리와 통합되어 사용됨.

```
import { Link } from 'react-router-dom';  
  
<Link to="/target-page">Go to Target Page</Link>
```

- 그러나 replace 속성을 이용하면 주소 히스토리를 저장하지 않게 할 수 있다.

```
<Link to="/" replace={true} />
```

▼ Link 자세한 내용

`href` : 이동할 URL을 값을 넣음

`replace` : 속성값 불리언

- `true` : 현재 페이지를 새로운 URL로 대체하여 브라우저 history를 변경합니다. 즉 뒤로가기 버튼을 눌러도 이전 페이지로 돌아가지 않고 현재 페이지가 유지된다. (주소 히스토리 저장x)
- `false` : 새로운 URL을 history에 추가하여 페이지를 이동한다. 사용자가 뒤로가기 버튼을 누르면 이전 페이지로 이동한다. (기본값)

`target` : 링크를 열 위치를 지정한다.

NavLink태그

- Link의 special version이다.
- `activeStyle`과 `activeClassName` 속성을 통해 특정 링크에 스타일을 넣어 줄 수 있다.

useNavigate 훅

- 태그가 아닌 훅의 형태로 함수 안에 적어 특정 조건을 충족할 경우에 페이지 이동을 할 수 있도록 설정할 수 있다.

정리

- 특정 조건에 충족해야만 페이지 이동을 가능하도록 설정 할 경우

`window.location` , `window.open` , `useNavigate`

- 별다른 것 없이 바로 페이지 이동만 할 경우

`Link태그` , `a태그`

▼ Link vs a

둘은 새로고침 유무에 따라서 달라진다.

- 페이지를 새로고침 할 경우 : SEO(검색 엔진 최적화)에 좋지만 초기 로딩 속도가 느릴 수 있다.
- 페이지를 새로고침 안 할 경우 : 초기 로딩 속도가 빠른 반면 일부 검색엔진은 크롤링하지 않을 수 있기 때문에 SEO에 좋지 않은 영향을 줄 수 있다. 그러나 SPA(Single Page Application)일 경우에는 처음에 필요한 모든 자원을 불러오게 되는데 이때 Link태그와 같이 새로 고침을 안 하는 것들을 사용했을 시 로딩 시간이 길어질 수 있다.

따라서 새로고침이 필요할 때는 a태그를 필요하지 않을때는 Link 태그를 이용하면 된다.