

Next.js 연습 1

1 Link Component / Client-side Navigate

프레임워크의 기능들을 익히고
그 장점을 온전히 이해하기 위해서는
실제로 서비스를 만들어 봐야한다.

Next.js 가 제공하는 여러 기능들

- page-based routing system(with dynamic routes)
- Pre-rendering SSG / SSR
- Automatic code splitting for faster page loads
- Client-side routing with optimized prefetching
- API Routes(with Serverless Functions)
- Development environment(with Fast Refresh)

연습 프로젝트(블로그)

<https://next-learn-starter.vercel.app/>

Next.js 가 제공하는 blog 예제를 확장 시켜보자

프로젝트 생성 - 메인 페이지(블로그 글 리스트) - 블로그 글 상세 페이지 - 배포

블로그

개발자들은 자기만의 블로그를 하나씩(?) 가지고 있다.

꾸준히 공부하고 고민한 부분에 대한 기록

다른 사람들에게 지식을 공유 + 본인 스스로 기록 + 나에 대해서 홍보

프로젝트 만들기

```
```bash
```

```
yarn create next-app blog --example "https://github.com/vercel/next-learn/tree/master/basics/learn-starter"
```

```
```
```

prettier 설정도

Pages 만들기

pages/posts/first-post.js

첫번째 글로 이동하기

Link Component

- `첫 번째 글`
- `<Link href="/posts/first-post"><a>첫 번째 글</Link>`

두 방식의 차이

Client Side Navigate

browser에서 url을 직접 쳐서 이동하는 것과 달리
JS 상에서 page 컴포넌트를 교체하는 것

확인(background-color를 body에 주고 navigate 해보기)

Code Splitting

Next.js는 Automatic Code Splitting을 제공

- 특정 페이지에 접근 할 때는 해당 페이지를 그릴때 **필요한** chunk만 로드
- 페이지 이동을 할 땐 목적지 페이지에 **필요한** chunk만 **추가** 로드

이를 통해 성능이 **최적화** 된다.

Prefetching

<Link> 컴포넌트를 이용하면, Viewport에 Link 컴포넌트가 노출되었을 때

href 로 연결된 페이지의 chunk를 로드한다.

이를 통해 성능을 **최적화**한다.

Link Component

- 본 서비스 외부 링크로 연결 할 때는 `<a>` tag 만 쓰면 됨
- Link Component에 스타일을 줄 때는 `<a>` tag 에 줘야 함

정리

Link Component / Client-side Navigate

프레임워크 익히는 최상의
방법

실제 서비스를 프레임워크로 만들기

Link component

Next.js 에서 제공하는 링크 기능

Client side Navigate

url로 접근하는 것과 달리 JS 상에서 변
경

Prefetching

Viewport 내 Link 페이지를 미리 페칭