

Next.js 기본 3

1 Pages / Layout / Image

Pages

pages/index.js => /

pages/ssg.js => /ssg

pages/products/[slug].js => /products/*

(js, jsx, ts, tsx)

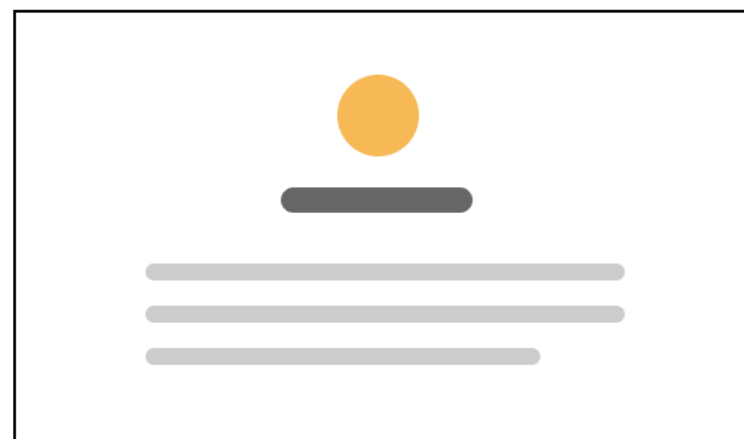
Pre-renders 와 SEO

Next.js는 모든 페이지를 기본적으로 pre-render 한다.

Pre-rendering (Using Next.js)

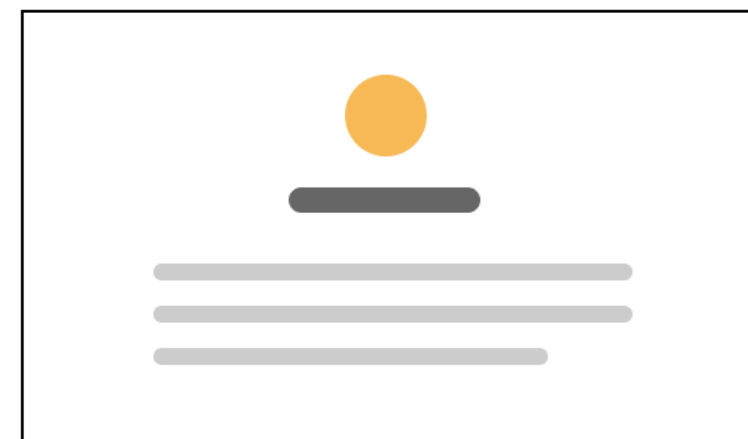
Initial Load:

Pre-rendered HTML is displayed



JS loads
→

Hydration: React components are initialized and App becomes interactive



If your app has interactive components like `<Link />`, they'll be active after JS loads

No Pre-rendering (Plain React.js app)

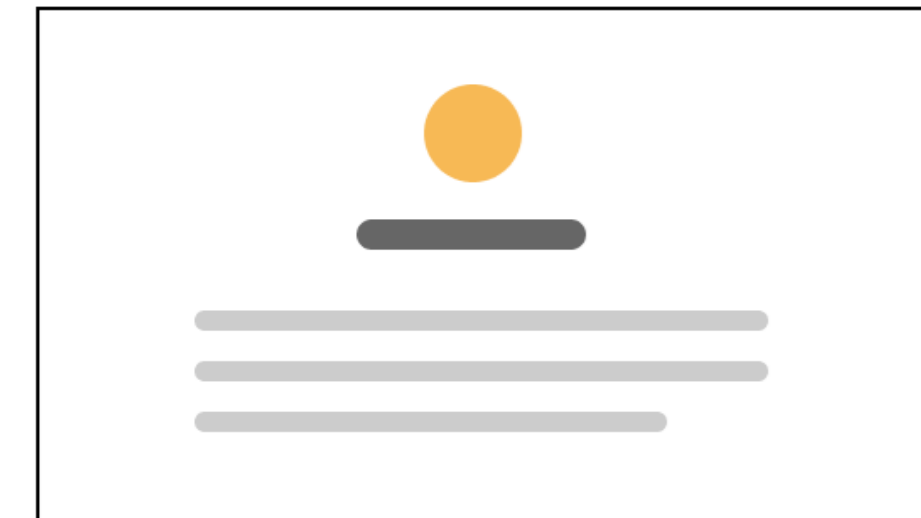
Initial Load:

App is not rendered



JS loads
→

Hydration: React components are initialized and App becomes interactive



Pre-rendering과 SEO의 상관관계

CSR만 제공한다면,
Client(브라우저) 처럼 동작하지 않는 검색엔진의 경우
아무런 데이터도 조회해갈 수 없다.

Pre-render를 해두면 Client 처럼 동작하지 않는 검색엔진에게
필요한 데이터를 제공 할 수 있다.

Next.js 의 Pre-rendering 방식?

SSG(recommended) & SSR

SSG는 빌드 타임에 pre-render

SSR은 요청 타임에 pre-render

SSG 2가지 상황

- Page의 내용물이 외부 데이터에 의존적인 상황
- Page Paths 까지 외부 데이터에 의존적인 상황

첫 번째는 `getStaticProps`만 가지고도 가능

두 번째는 `getStaticPaths` 도 함께 활용해야 가능

Layouts

여러 Page들의 공통 처리

하나의 공통된 Layout을 쓰는 경우

`components/Layout.js`

컴포넌트 하나를 `pages/_app.js` 에서 활용하면 됨

여러개의 Layouts을 활용하고 싶은 경우

components/SubLayout.js

Page.getLayout에 getLayout 함수를 제공

Images

Next.js 가 제공하는 최적화 Image Component

- Improved Performance
- Visual Stability(CLS- Cumulative Layout Shift 방지)
- Faster Page Loads(viewport 진입시 로드 / blur 처리)
- Asset Flexibility(리사이징)

```
```bash
```

```
npx create-next-app --example image-component image-app
```

```
```
```

responsive / color 살펴보기

정리

Pages / Layout / Image

Pre-rendering

SEO 와 초기 로딩 속도

SSG vs SSR

SSG은 빌드 시 / SSR은 요청 시

Layout

pages/_app.js 활용해서 페이지 공통화

Images

최적화된 이미지 활용 util