

NEXT.JS #1

next.js 소개/ 개발 환경 구성

// React //

- JSX
- Virtual DOM
- 뷰 컴포넌트를 위한 라이브러리
- Single Page Application으로 확장 가능

// JSX //

- 리액트를 구성하는 기본 스크립트 형식
- ECMA Script 6 기반의 자바스크립트와 XML이 혼합된 형태
- 초기엔 jsx 확장자를 권장했으나 지금은 js 확장자를 권장

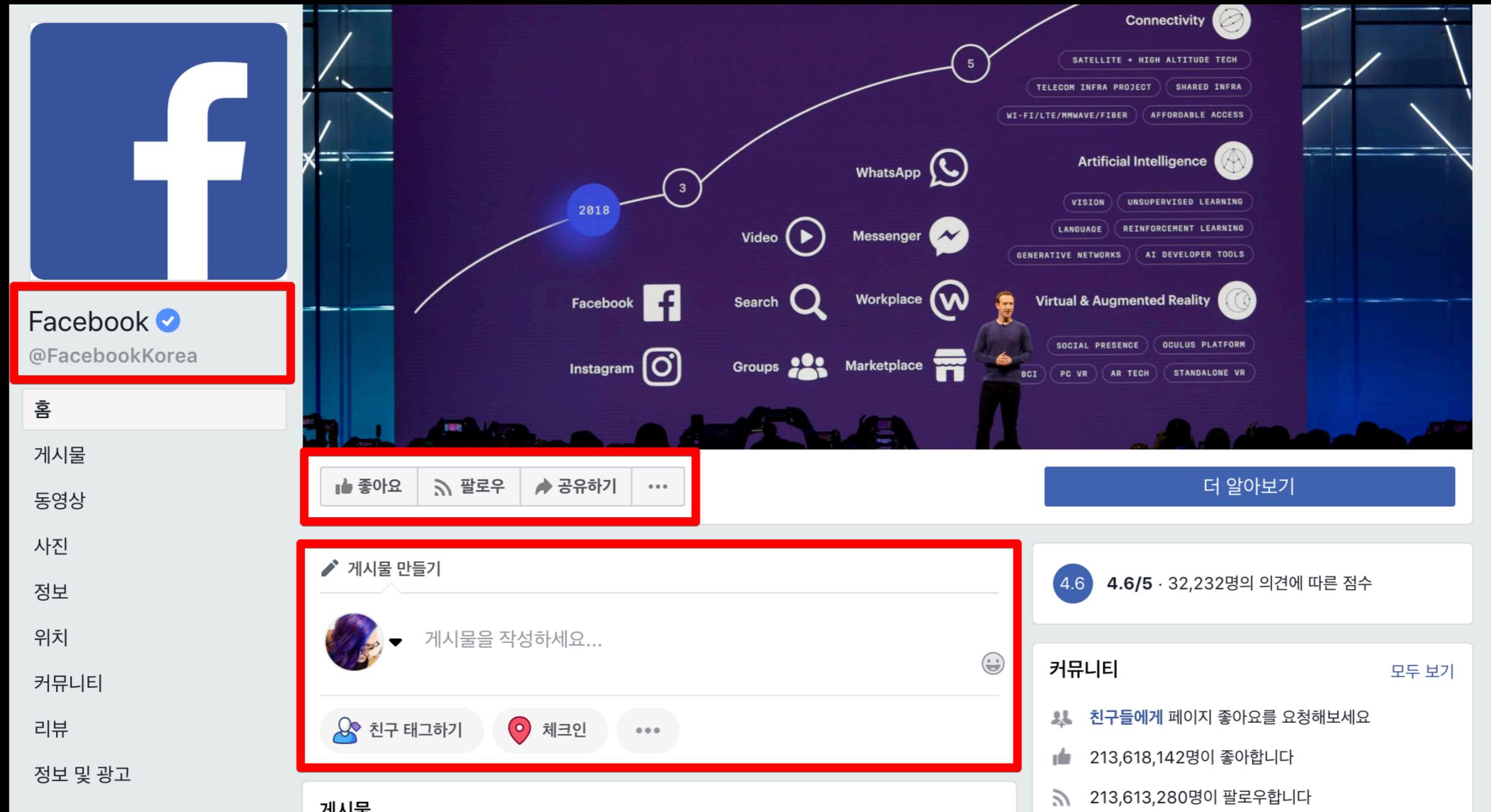
// Virtual DOM //

- JSX를 DOM으로 출력하는 중간 단계
- 리액트가 메모리상에만 보관하고 있는 가상의 DOM
- draw call 최적화

// 뷰 컴포넌트 라이브러리 //

- 리액트는 프레임워크가 아닌 라이브러리임을 강조
- 화면의 일부분만 리액트 컴포넌트 적용 가능
예) 페이스북

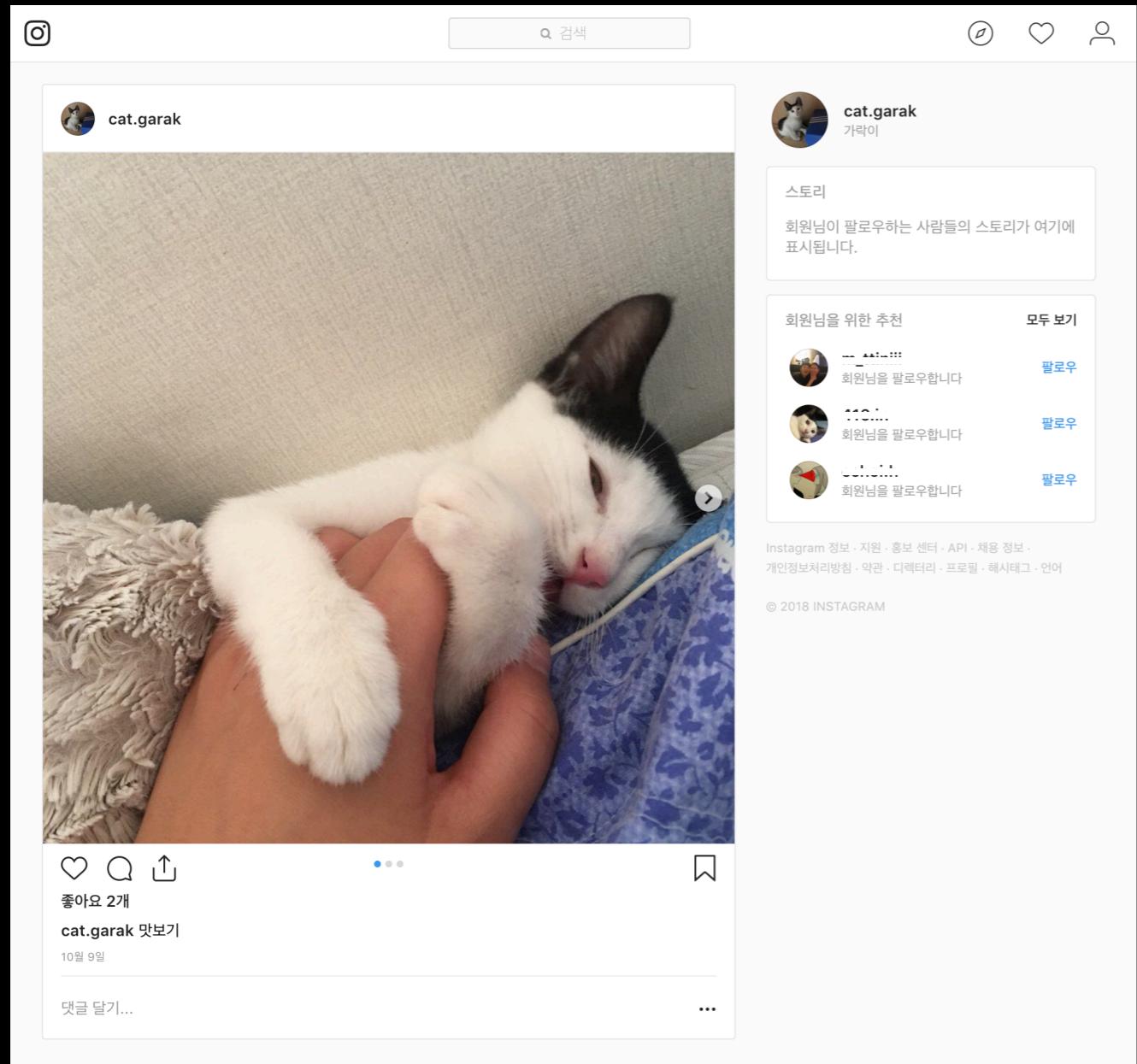
// 예시: 페이스북의 경우 //



// SPA로 확장 //

- Single Page Application
- 사이트 전체가 자바스크립트로 구동되는 웹 애플리케이션 형태
- react-router-dom 등의 서드파티 오픈소스 조합/활용

// 예시: 인스타그램 //



```
▼ <t>
  ▼ <a>
    ▼ <t>
      ▼ <withRouter(Connect(u))>
        ▼ <t>
          ▼ <Connect(u)>
            ▶ <u>...</u> == $r
          </Connect(u)>
        </t>
      </withRouter(Connect(u))>
    </t>
  </a>
</t>
```

React Debugger

Router를 포함하여 SPA 형태로 구성되어 있음

// 검색 엔진 최적화 //

- 개발 이후 운영 단계에서 필수적
- 컨텐츠별로 할당된 고유 URI
- head 태그 내의 meta data
- body 태그 내의 실제 컨텐츠
- 등등...

// SPA의 한계 //

- 자바스크립트 없이는 비어있는 DOM 뿐
- 라우팅도 프론트엔드에서 처리하므로 모든 URI가 비어있는 DOM

// SPA의 한계 //



유저가 보는 화면

```
<body>
  <noscript>
    You need to enable Ja
  </noscript>
  <div id="root"></div>
  <!--
    This HTML file is a t
```

검색 봇이 읽을 수 있는 DOM

// Server Side Rendering //

- 서버에서 미리 DOM을 렌더링
- 방법1: 메타태그만 렌더링
- 방법2: 컨텐츠 전체를 렌더링

// 메타태그 렌더링 //

- head 내의 title, description 등의 meta 정보만 서버에서 렌더링
- Open Graph 메타 등을 출력해 소셜 미디어 공유에 대응
- 웹사이트 주요 페이지들을 검색 봇에 노출 가능

// 컨텐츠 렌더링 //

- 서버 내에서 node.js 엔진을 연동해 JSX 내용을 DOM으로 출력
- 운영중인 백엔드가 node.js 가 아니라면 많은 서버 리소스 필요
 - 이 때문에 서버사이드 뷰 템플릿과 JSX를 동시에 관리하기도
-> 운영중 모든 업데이트에 2배의 작업 부하

```
// next.js //
```

- React
- Routing
- Server Side Rendering
- ...이 포함된 프레임워크

// 설치 //

yarn init -y

yarn add next react react-dom

// 명령어 세팅 //

// package.json에 추가

```
{  
  "scripts": {  
    "dev": "next",  
    "build": "next build",  
    "start": "next start"  
  }  
}
```

// 파일 생성 //

// pages/index.js 경로에 생성

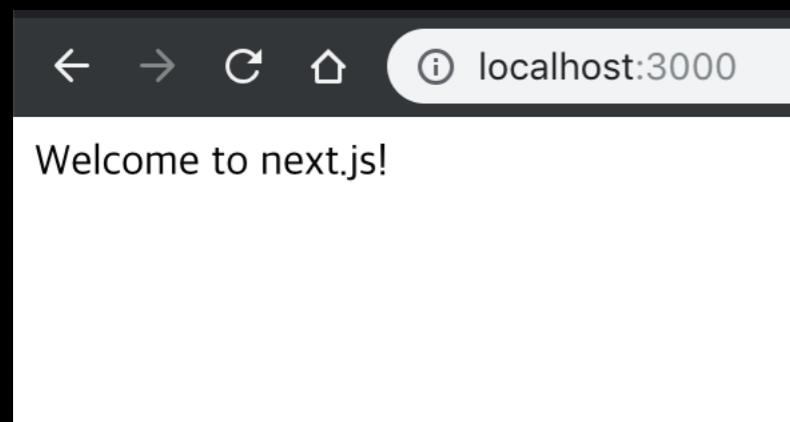
```
export default () => (
  <div>Welcome to next.js!</div>
);
```

// 실행 //

yarn dev

이후 웹 브라우저에서 localhost:3000 접속

// 결과 //



Welcome to next.js!

```
development/pages/index.js" as="script"/><link rel="preload" href="/_next/static/development/pages/_error.js" as="script" data-bbox="108 138 908 188"/>

Welcome to next.js!

</div><script data-bbox="108 188 908 238" pageProps={{ page: '/', query: {}, buildId: 'development' }}>_NEXT_PAGE__/" src="/_next/static/development/pages/index.js"</script><script data-bbox="108 238 908 288" pageProps={{ page: '/_error', query: {}, buildId: 'development' }}>_PAGE__/_error" src="/_next/static/development/pages/_error.js"</script></script></body></html>
```

렌더링 된 화면

검색 봇이 읽을 수 있는 DOM

// 질문 / 답변 / github //

질문 & 답변

코드는: github.com/GrotesQ/CodeLab-Next-JS

감사합니다.