

Chapter 1. 리액트 소개

교재: 처음만난 리액트 (저자: 이인제, 한빛출판사)



Contents

- CHAPTER 1: 리액트 소개

- 1.1 리액트는 무엇인가?

- 1.2 리액트의 장점

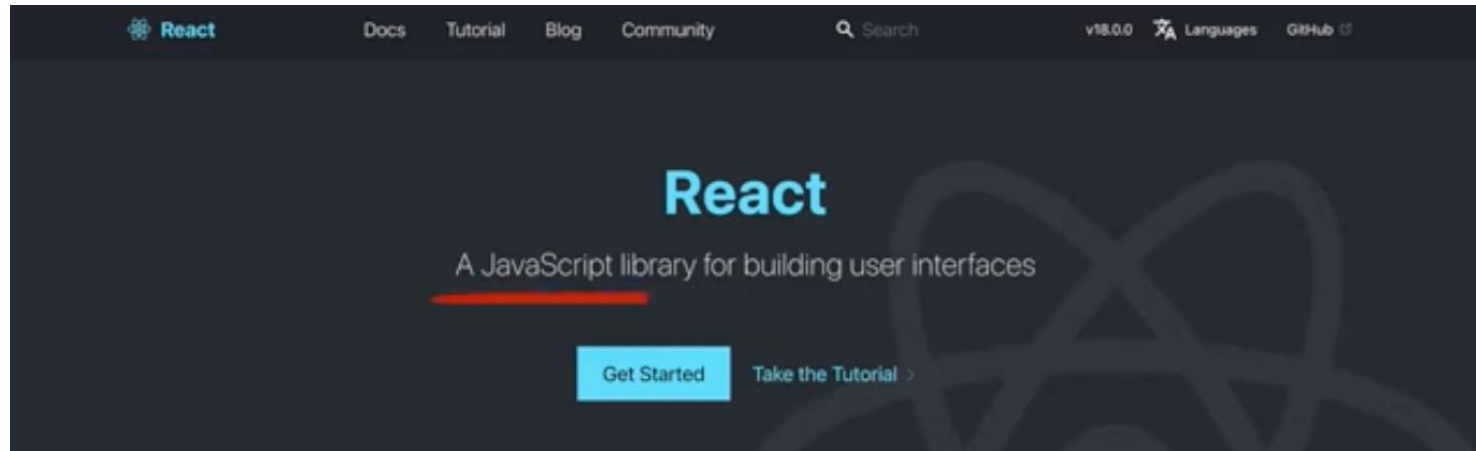
- 1.3 리액트의 단점

- 1.4 개발환경 설정

SECTION 1.1 리엑트는 무엇인가?

- 리엑트의 정의

- A JavaScript library for building user interfaces
 - 사용자 인터페이스를 만들기 위한 JavaScript 라이브러리



SECTION 1.1 리엑트는 무엇인가?

- 리엑트의 정의

- The library for web and native user interfaces
 - 웹과 네이티브 사용자 인터페이스를 만들기 위한 라이브러리

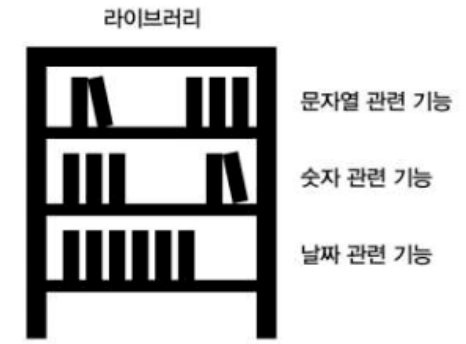


<https://react.dev/>

SECTION 1.1 리엑트는 무엇인가?

◦ 라이브러리

- 자주 사용되는 기능을 정리해 모아 놓은 것
- 특정 프로그래밍 언어에서 자주 사용되는 기능을 잘 모아서 정리해 놓은 모음집



◦ 사용자 인터페이스(User Interface, UI)

- 사용자와 컴퓨터 프로그램이 서로 상호작용을 하기 위해 중간에서 서로 간에 입력과 출력을 제어해주는 것. ex) 텍스트 박스, 버튼

◦ UI 라이브러리

- 사용자 인터페이스를 만들기 위한 기능 모음집(라이브러리)

SECTION 1.1 리엑트는 무엇인가?

- 대표적인 자바스크립트 UI 라이브러리 / 프레임워크



Google

2010



Meta

2013



Evan You

- 프레임워크와 라이브러리
 - 프로그램 흐름에 대한 제어 권한의 차이
 - 프레임워크는 전체적인 흐름을 갖고 있으며, 애플리케이션의 코드는 프레임워크에 의해 사용됨
 - 라이브러리는 흐름에 대한 제어를 하지 않고 개발자가 필요한 부분만 가져다 사용

SECTION 1.1 리엑트는 무엇인가?

- 대표적인 자바스크립트 UI 라이브러리 / 프레임워크



npm trends

angular vs react vs vue

Enter an npm package...

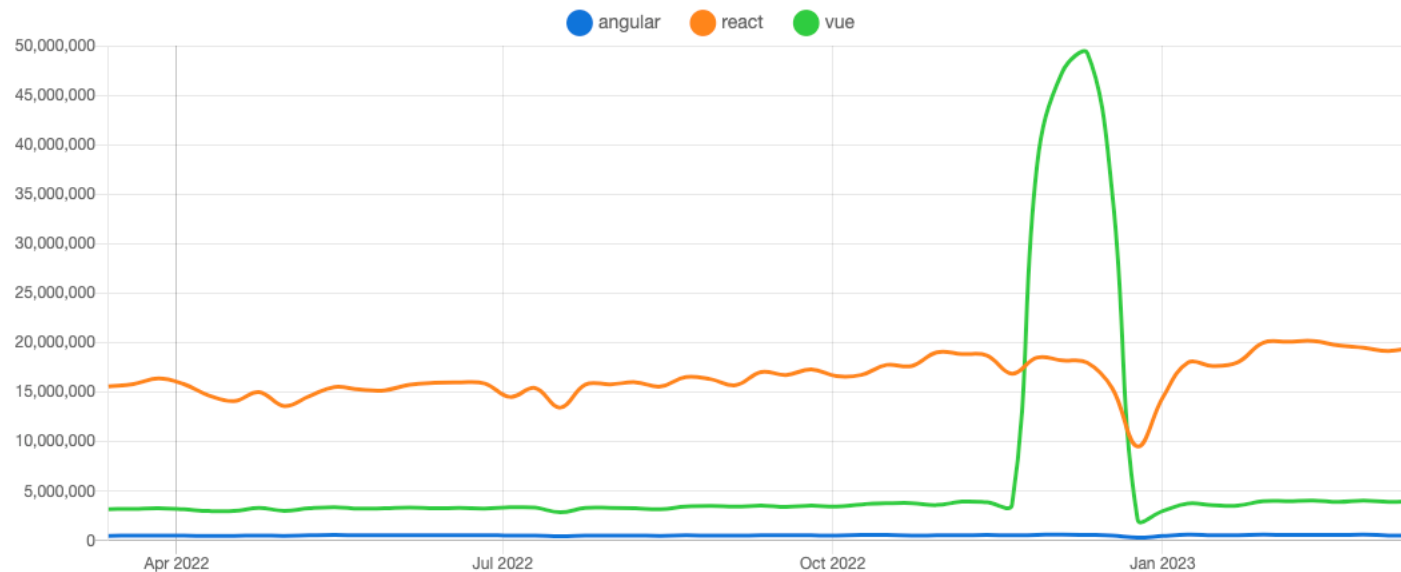
angular x

react x

vue x

+ @angular/core

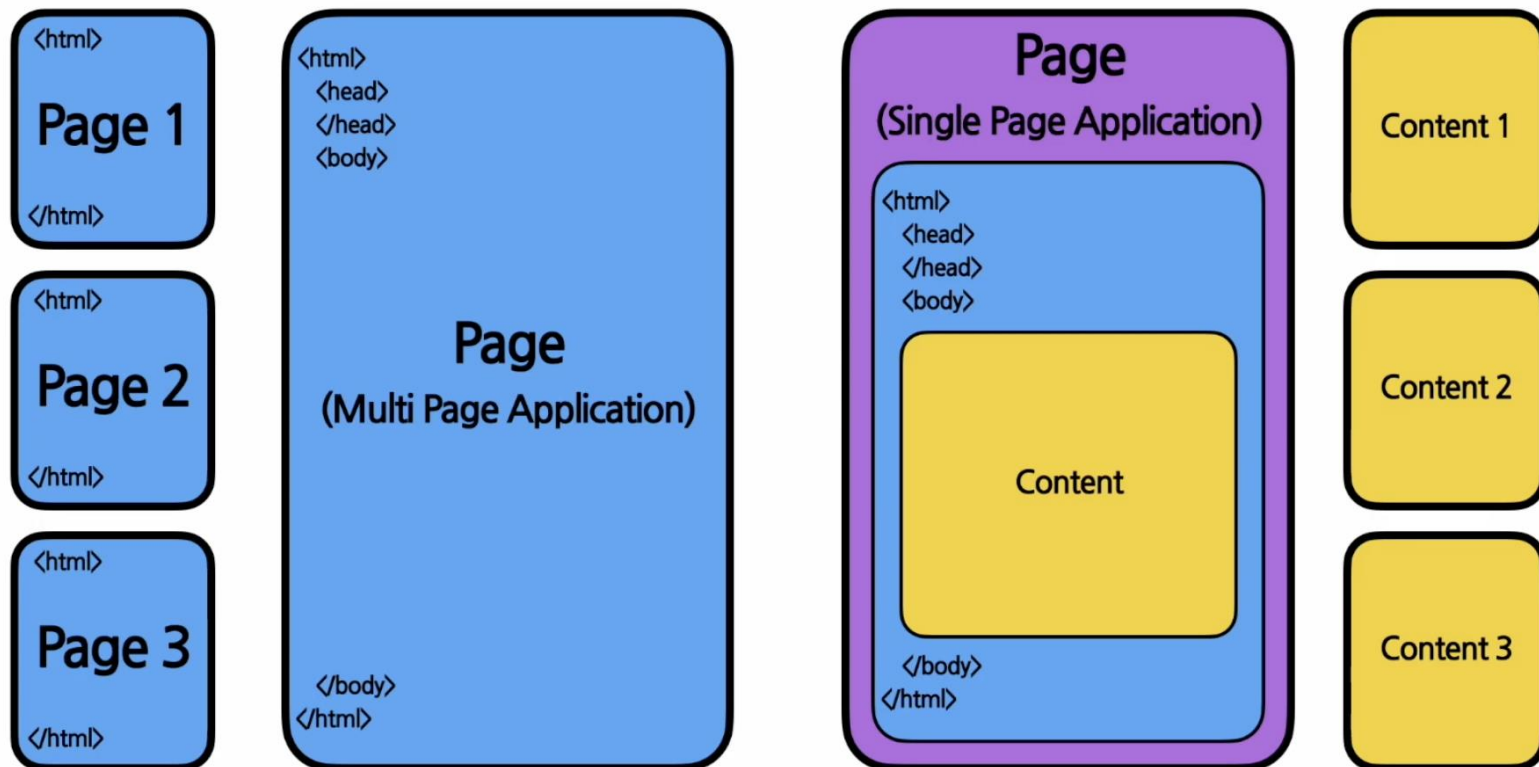
Downloads in past 1 Year



SECTION 1.1 리엑트는 무엇인가?

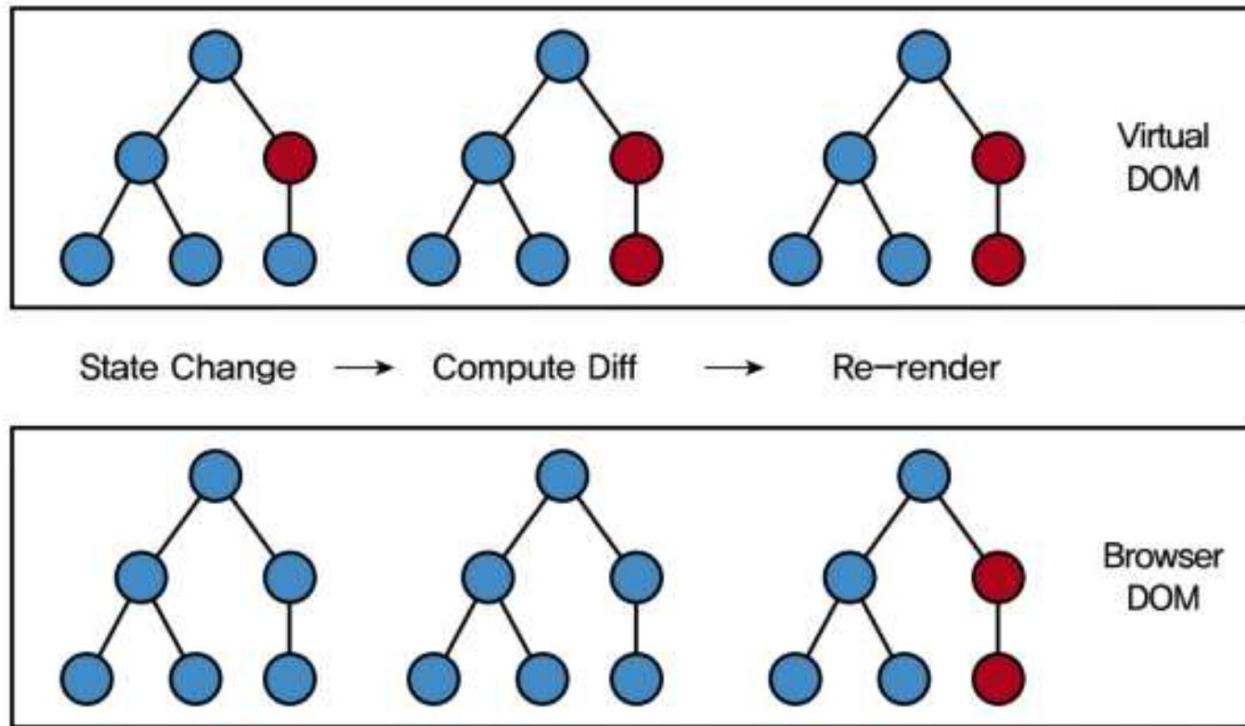
- SPA (Single Page Application)

- 단일 페이지로 구성된 웹 어플리케이션
- 하나의 HTML 틀을 만들어 놓고, 사용자가 특정 페이지를 요청할 때, 그 페이지에 해당하는 콘텐츠를 가져와서 동적으로 페이지의 내용을 채워서 보여주는 방식



SECTION 1.2 리액트의 장점

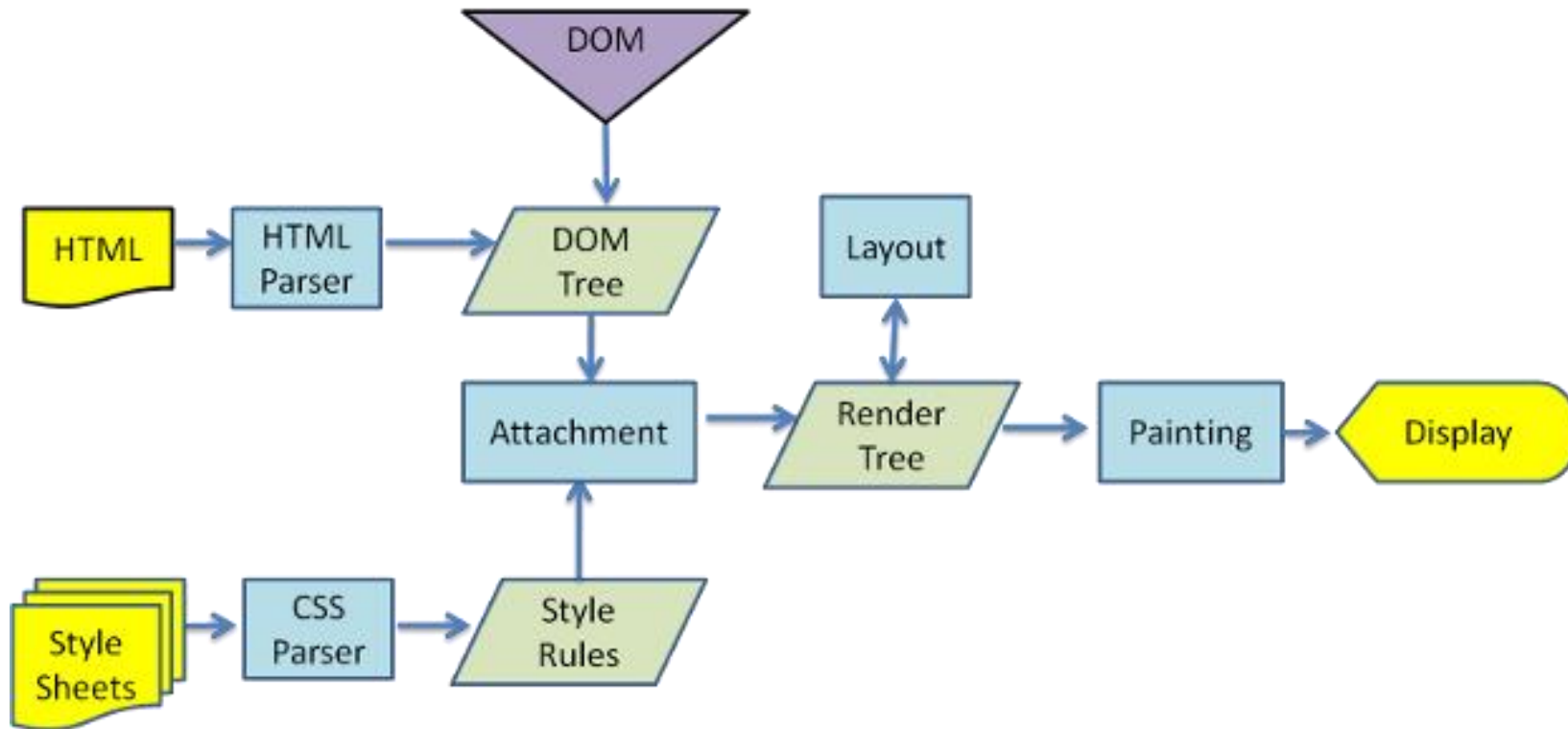
- 빠른 업데이트와 렌더링 속도
 - Virtual DOM을 사용하여 최소한의 부분만을 찾아서 빠르게 업데이트
 - Virtual DOM (가상의 DOM)
 - 웹페이지와 실제 DOM 사이에서 중간 매개체 역할



▶ Virtual DOM 개념도

SECTION 1.2 리액트의 장점

- 빠른 업데이트와 렌더링 속도
 - [참고] 브라우저의 workflow

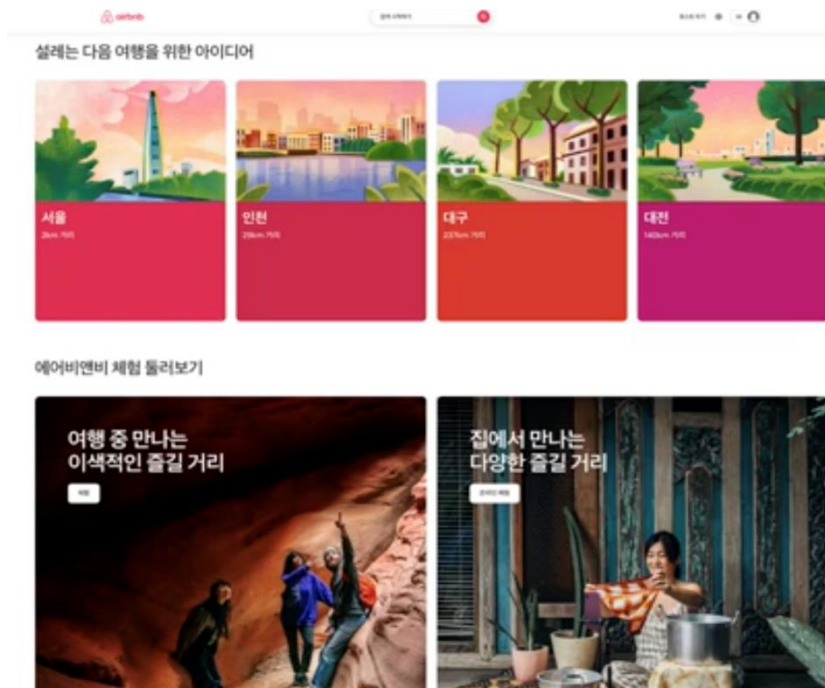


출처 - <https://velopert.com/3236>

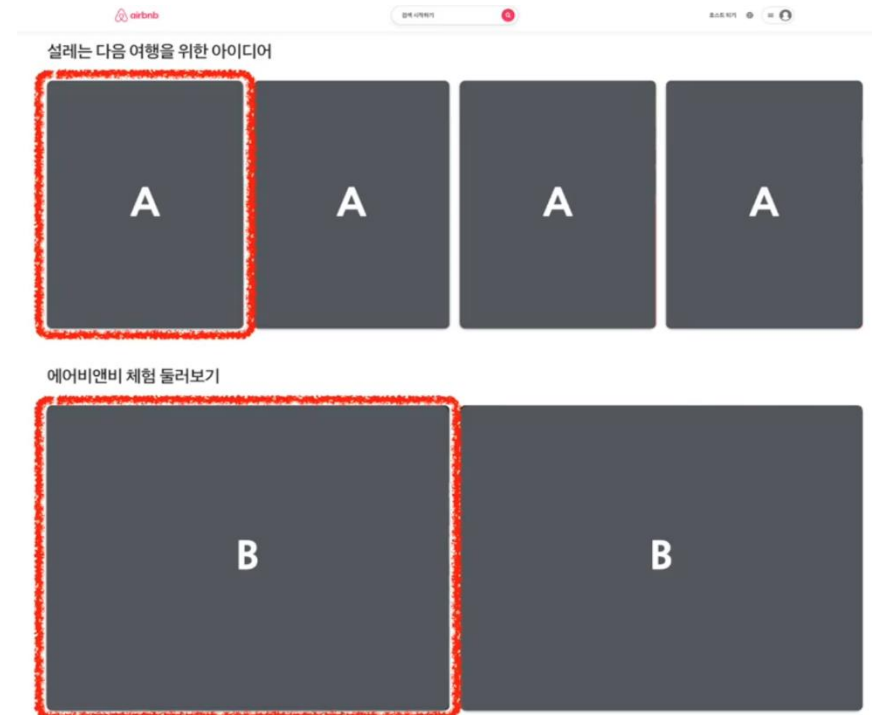
SECTION 1.2 리액트의 장점

◦ 컴포넌트 기반 구조(Component-Based)

- 컴포넌트 - 독립적인 기능을 수행하는 작은 기능 단위 모듈
- 리액트에서는 모든 페이지가 컴포넌트로 구성됨
- 하나의 컴포넌트는 또 다른 여러 개의 컴포넌트의 조합으로 구성될 수 있음
 - 레고 블록을 조립하는 것처럼 컴포넌트를 조합해서 사용



에어비앤비 사이트

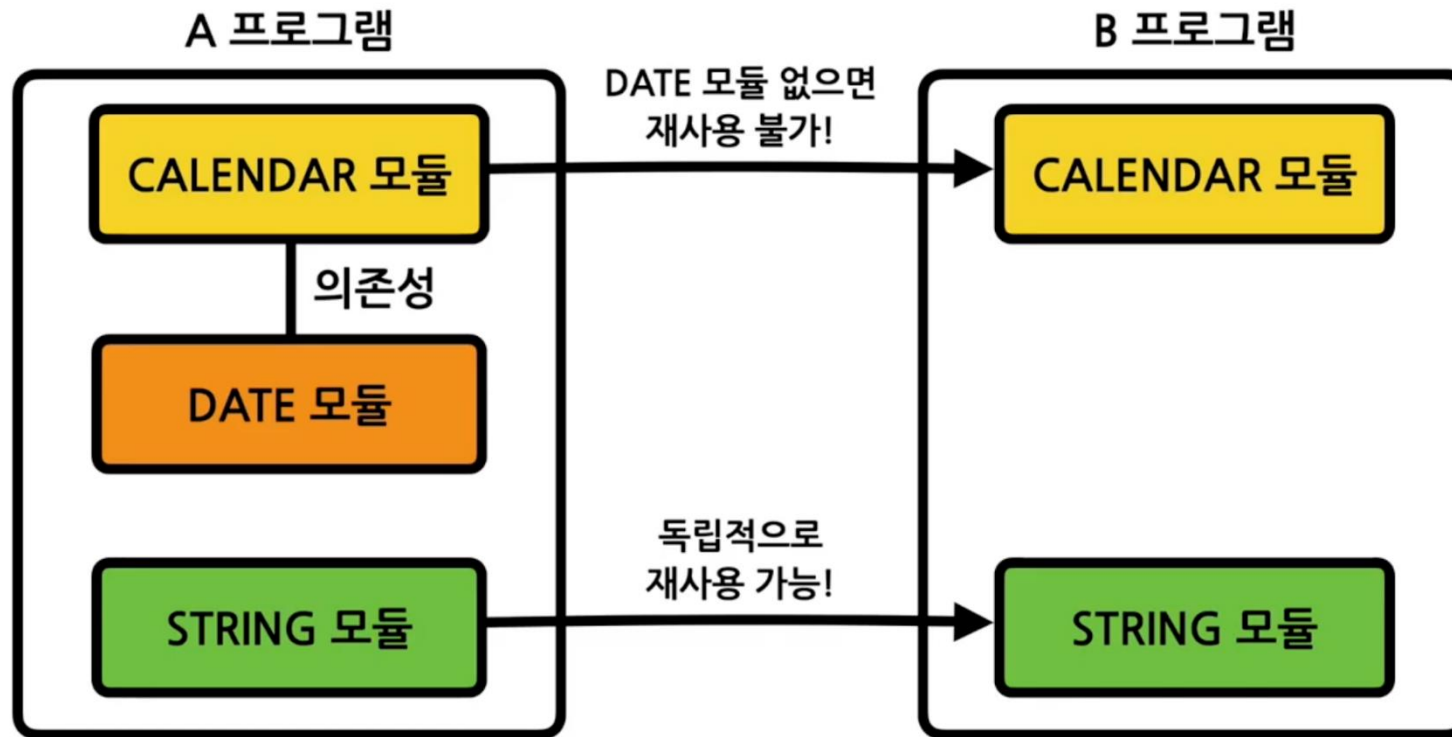


컴포넌트 구조

SECTION 1.2 리액트의 장점

◦ 재사용성(Reusability)

- 객체지향 프로그래밍의 주요 특징으로 소프트웨어 개발에 있어서 매우 중요
- 다시 사용이 가능한 성질
- 다른 모듈과 의존성이 낮을수록 재사용성은 증가함



SECTION 1.2 리액트의 장점

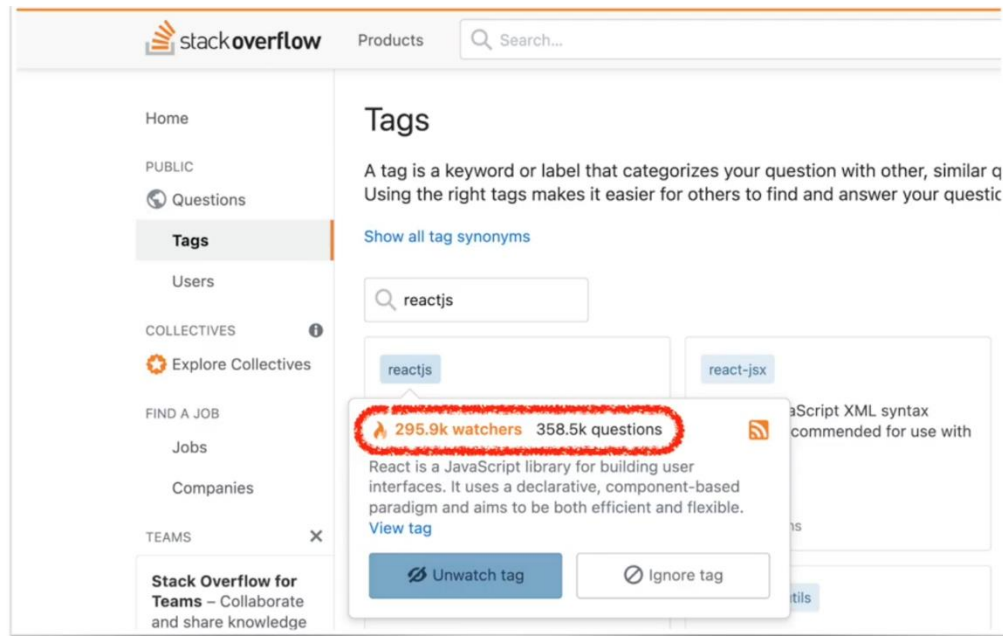
- 재사용성이 높아지면 좋은 점
 - 개발 기간 단축(생산성 향상)
 - 유지보수 용이
- 리액트와 재사용성
 - 리액트는 컴포넌트 기반 구조
 - 컴포넌트들은 웹 사이트의 여러 곳에서 반복적으로 사용 가능
 - 리액트 컴포넌트 개발 시 쉽고 재사용 가능한 형태로 개발

SECTION 1.2 리액트의 장점

- 든든한 지원군

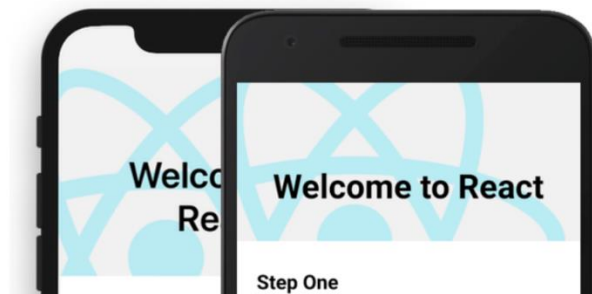


- 꾸준한 버전 업데이트
- 활발한 지식 공유 & 커뮤니티



SECTION 1.2 리액트의 장점

- 모바일 앱 개발 가능
 - 자바스크립트 코딩을 통해 안드로이드/iOS 앱 개발 가능



Create native apps for Android, iOS, and more using React

React Native combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces.

SECTION 1.3 리액트의 단점

- 방대한 학습량

Virtual DOM?
JSX? Component?
State? Props?

SECTION 1.3 리액트의 단점

- 높은 상태관리 복잡도

- 리액트에서 state는 굉장히 중요한 개념
- state - 리액트 컴포넌트의 상태
- Virtual DOM에서 state가 바뀐 컴포넌트만 업데이트
- 성능 최적화를 위해 state 관리 중요
- 컴포넌트의 개수가 많아지면 상태 관리의 복잡도 증가
- 큰 규모의 프로젝트에서는 상태 관리를 위해 Redux, MobX, Recoil 등의 외부 라이브러리 사용

[요약]

- 리액트란?
 - 사용자 인터페이스를 만들기 위한 자바스크립트 라이브러리
 - SPA를 쉽고 빠르게 만들수 있도록 해주는 도구
- 리액트의 장점
 - 빠른 업데이트와 렌더링 속도
 - 재사용성이 높은 컴포넌트 기반 구조
 - 메타의 든든한 지원
 - 활발한 지식 공유와 커뮤니티
 - 리액트 네이티브를 통한 모바일 앱 개발 가능
- 리액트의 단점
 - 방대한 학습량
 - 높은 상태 관리 복잡도

SECTION 1.4 개발환경 설정

- Node.js와 npm 설치하기

Node.js

(Javascript runtime)

- Javascript로 네트워크 애플리케이션을 개발할 수 있게 해주는 개발 환경(플랫폼)

npm

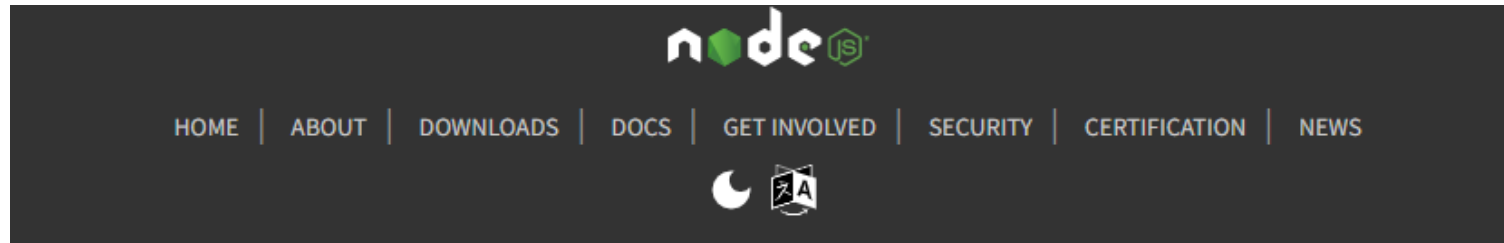
(node package manager)

- 프로젝트에 필요한 다양한 외부 패키지들의 버전과 의존성을 관리
- 패키지를 편하게 설치, 삭제를 할 수 있게 도와주는 역할

SECTION 1.4 개발환경 설정

- Node.js 설치

<https://nodejs.org/en>



Node.js® is an open-source, cross-platform JavaScript runtime environment.

Download for Windows (x64)

18.15.0 LTS

Recommended For Most Users

19.8.1 Current

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

For information about supported releases, see the [release schedule](#).

SECTION 1.4 개발환경 설정

- Node.js 설치 확인 (Windows PowerShell)

```
$ node --version
```

```
v18.15.0
```

```
$ npm --version
```

```
9.5.0
```