

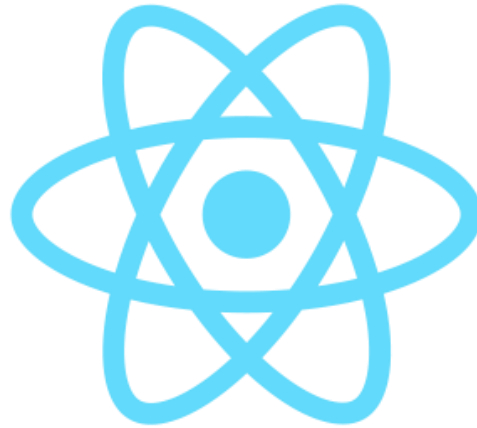
React

React 소개

가장 많이 쓰이는 것은?



ANGULARJS



React JS



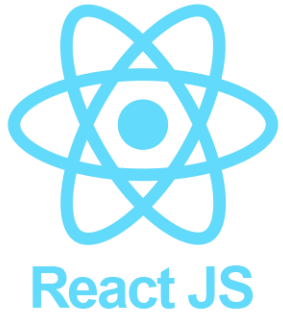
Vue.js

Angular JS



- 구글에서 만든 JavaScript 기반의 오픈 소스 프레임워크
- 양방향 데이터 바인딩으로 양방향 웹 애플리케이션에 적합
- 2016년도 이후 점유율 하락 중
- Ex) 유튜브, 페이스북, 구글, 텔레그램 등등

React JS



- 동적 사용자 인터페이스를 만들기 위해 2011년 페이스북에서 만든 오픈 소스 JavaScript 라이브러리
- 데이터 변경이 잦은 복잡하고, 규모가 큰 라이브러리에 적합
- Angular 보다 배우기 쉽다고 이야기 됨.
- Ex) 페이스북, 인스타그램, 넷플릭스, 야후, 드롭박스 등등

Vue.js



- 2013년 출시된 JavaScript 프레임워크
- Angular와 React의 장점을 수용한 프레임워크
- 중국어 기반으로 Reference가 적음.
- Ex) 샤오미, 알리바바, 깃랩, 어도비 등

React VS. Vue.js

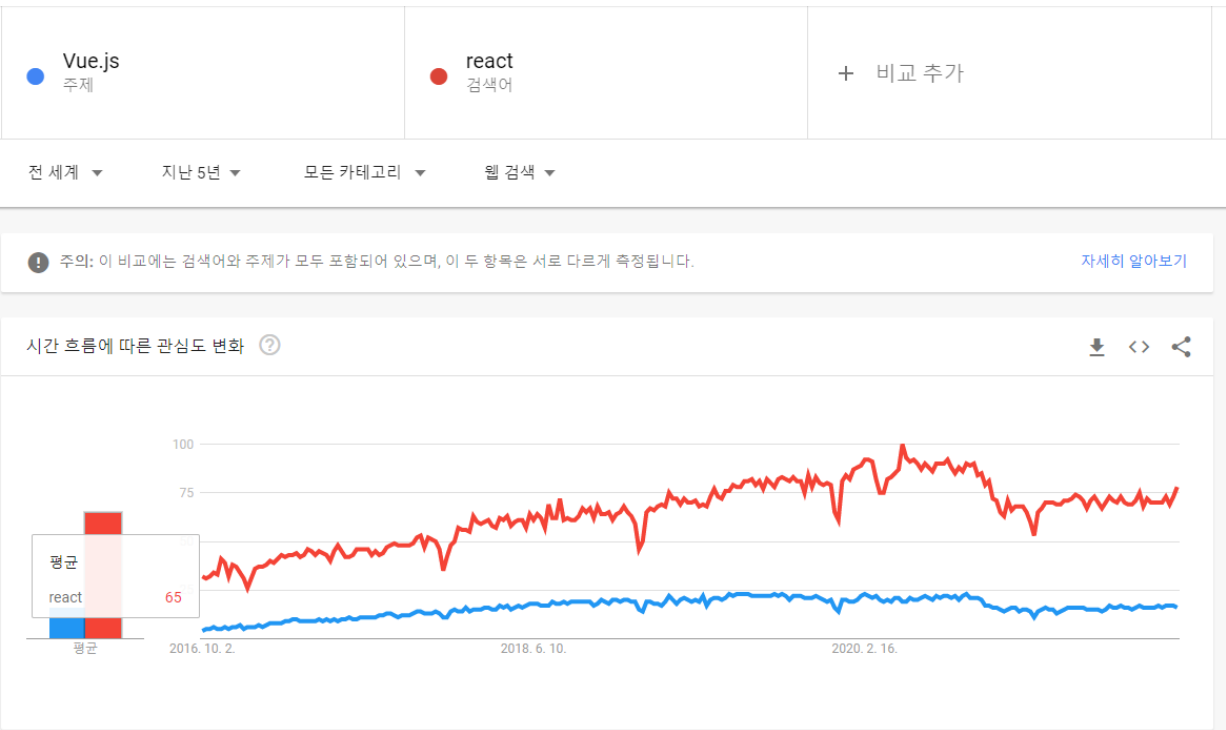
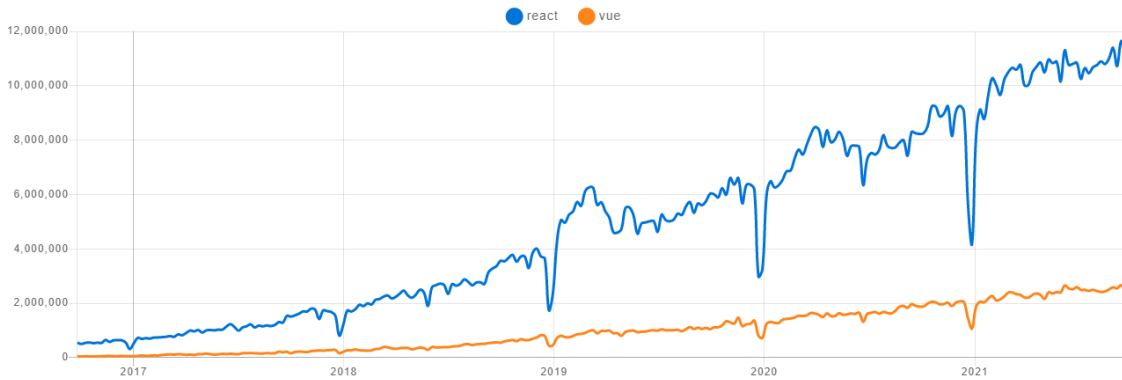


react vs vue

Enter an npm package...

react x vue x + @angular/core + angular + ember-source

Downloads in past 5 Years

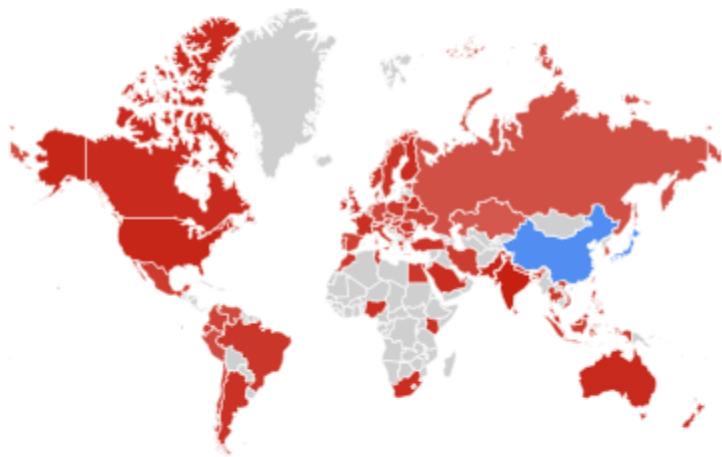


React VS. Vue.js

지역별 비교 분석

지역 ▼ ⬇ <> 🔗

● Vue.js ● react



색상의 농도는 검색어의 비율을 나타냅니다. [자세히 알아보기](#)

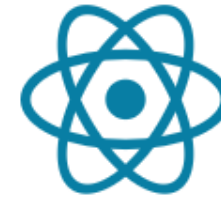
정렬: Vue.js에 대한 관심도 ▼

1	중국	<div><div></div><div></div></div>
2	일본	<div><div></div><div></div></div>
3	대만	<div><div></div><div></div></div>
4	카자흐스탄	<div><div></div><div></div></div>
5	대한민국	<div><div></div><div></div></div>

React

React란?

- <https://reactjs.org/>
- The library for web and native user interfaces



React

The library for web and native user interfaces

Learn React

API Reference

React란?

- 화면을 만들기 위한 자바스크립트 라이브러리
- 현재 가장 많이 활용되고 있는 인기 있는 라이브러리
- 사용자와 상호 작용이 가능한 동적 UI 제작 가능

React의 특징

1. Data Flow
2. Component 기반 구조
3. Virtual Dom
4. Props and State
5. JSX

특징 1) Data Flow

- 양방향 X, **단방향 O 데이터 흐름**
- 부모에서 자식으로의 데이터 전달
- Angular JS 처럼 양방향 데이터 바인딩은 규모가 커질수록 데이터의 흐름을 추적하기 힘들고 복잡해지는 경향이 있다.

특징 2) Component 기반 구조

- **Component** : 독립적인 단위의 소프트웨어 모듈로 소프트웨어를 독립적인 **하나의 부품**으로 만드는 방법
- React는 UI(View)를 여러 Component를 쪼개서 만든다.
- 한 페이지 내에서 여러 부분을 Component로 만들고 이를 조립해 화면을 구성

(컴포넌트는 마치 하나의 태그처럼 사용된다)

특징 2) Component 기반 구조

장점

1. Component 단위로 쪼개져 있기 때문에, 전체 코드를 파악하기 쉽다.
2. 기능 단위, UI 단위로 캡슐화시켜 코드를 관리하기 때문에 **재사용성**이 높다.
3. 코드를 반복할 필요 없이 Component만 import 해서 사용하면 된다는 **간편함**이 있다.
4. 애플리케이션이 복잡해지더라도 **코드의 유지보수, 관리가 용이**하다.

캡슐화란?

데이터와, 데이터를 처리하는 행위를 묶고, 외부에는 그 행위를 보여주지 않는 것

특징 2) Component 기반 구조

```
class App extends Component {  
  render() {  
    return (  
      <Layout>  
        <Header />  
        <Navigation />  
        <Content>  
          <Sidebar></Sidebar>  
          <Router />  
        </Content>  
        <Footer></Footer>  
      </Layout>  
    );  
  }  
}
```

개발자가 만든 tag는 닫는 tag도 있어야 컴파일러가 시작과 끝을 인식한다.

기준에 있는 태그들은 닫는 태그 없어도 된다



특징 2) Component 기반 구조

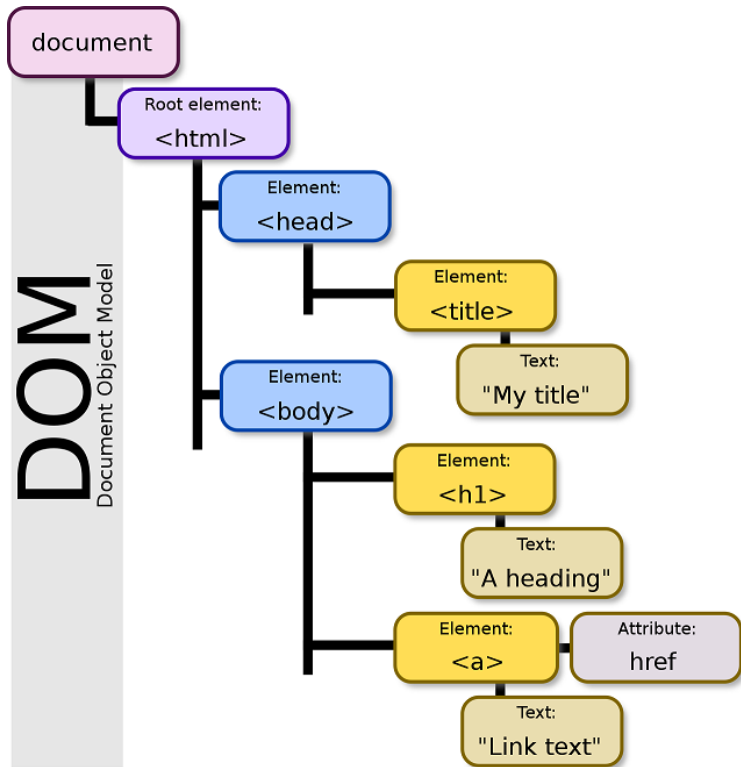
```
class App extends Component {  
  render() {  
    return (  
      <Layout>  
        <Header />  
        <Navigation />  
        <Content>  
          <Sidebar></Sidebar>  
          <Router />  
        </Content>  
        <Footer></Footer>  
      </Layout>  
    );  
  }  
}
```

Header, Footer 같은 구조를 컴포넌트로 제작

이것들을 조합해서 root component (최상위 컴포넌트) 로 만들기 - Layout

특징 3) Virtual DOM

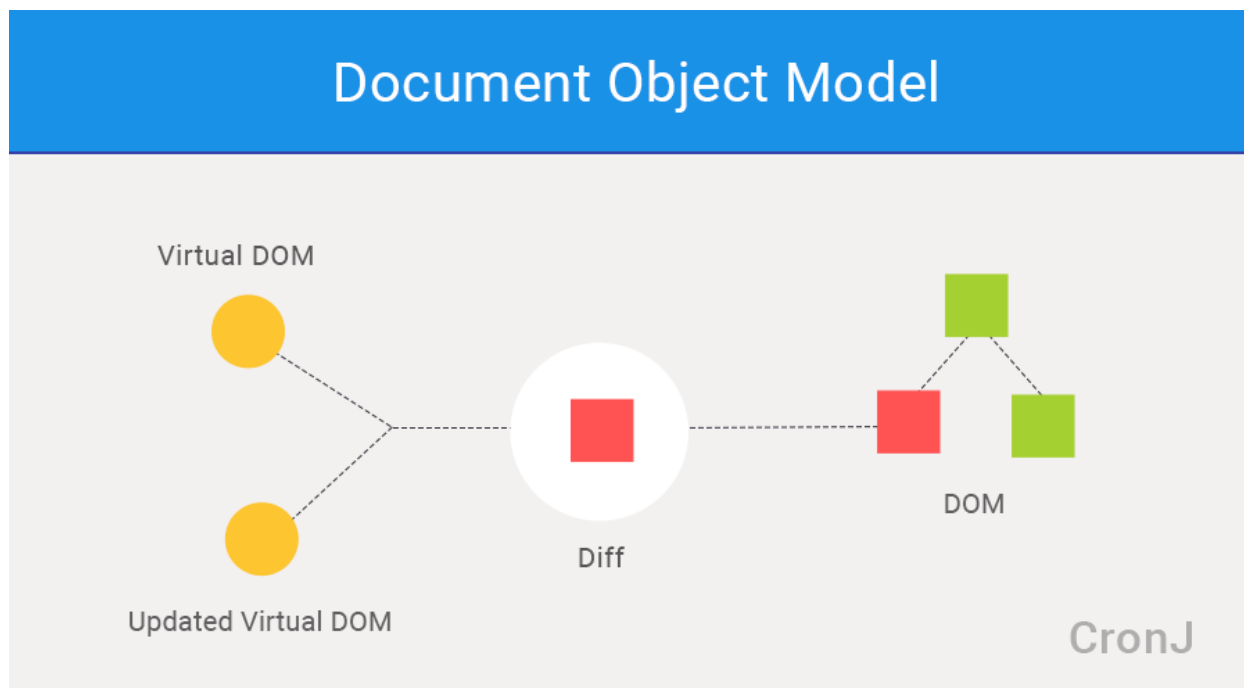
- DOM : Document Object Model (문서 객체 모델)



- React는 DOM Tree 구조와 같은 구조를 Virtual DOM 으로 가지고 있다.
- 이벤트가 발생할 때마다 Virtual DOM을 만들고 다시 그릴 때 실제와 전후 상태를 계속 비교, 변경이 필요한 최소한의 변경사항만 실제 DOM에 반영
→ 앱의 효율성과 속도 개선

특징 3) Virtual DOM

- 이벤트가 발생할 때마다 Virtual DOM을 만들고 다시 그릴 때 실제와 전후 상태를 계속 비교 -> 앱의 효율성과 속도 개선



특징 4) Props and State

- Props 부모 comp (데이터) to 자식 component
 - 부모 컴포넌트에서 자식 컴포넌트로 전달해주는 데이터
 - 자식에서는 props 변경 불가능, props를 전달한 최상위에서만 변경 가능
- State
 - 컴포넌트 내부에서 선언되고 내부에서 값을 변경
 - 클래스형 컴포넌트에서만 사용 가능, 각각의 state는 독립적 (함수형에서는 NO!)

특징 5) JSX

- React 에서 JSX 사용이 필수는 아니지만 편의성을 위해 대부분의 프로젝트에서 많이 사용된다!
- JSX = Javascript + XML
- JavaScript를 확장한 문법으로 React Element를 생성



더 자세한 건 React 프로젝트를

생성한 후에 배워보자!

React 프로젝트 생성

React 프로젝트

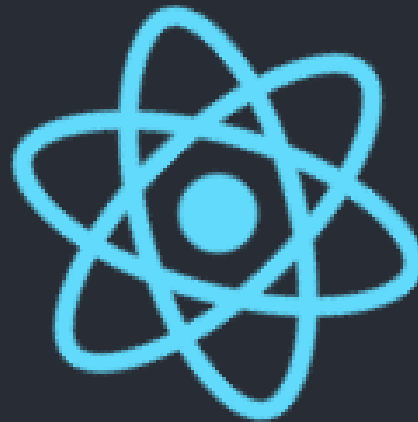
```
npx create-react-app [앱이름]
```

- 리액트 앱(프로젝트) 이름 규칙
 - 대문자 사용 불가!
 - 단어 여러 개 사용시 하이픈(-) 기호로 구분
- npx란?
 - npm의 자식 명령어로 npm보다 가볍게 패키지를 구성한다.
 - npm 버전이 5.2 이상일 때, node 버전이 14 이상일 때 사용 가능
- CRA 방식으로 쉽게 프로젝트 시작하기
 - 페이스북에서 만든 react 웹 개발용 boilerplate이다.

React 프로젝트

localhost:3000

```
npm start
```



Edit `src/App.js` and save to reload.

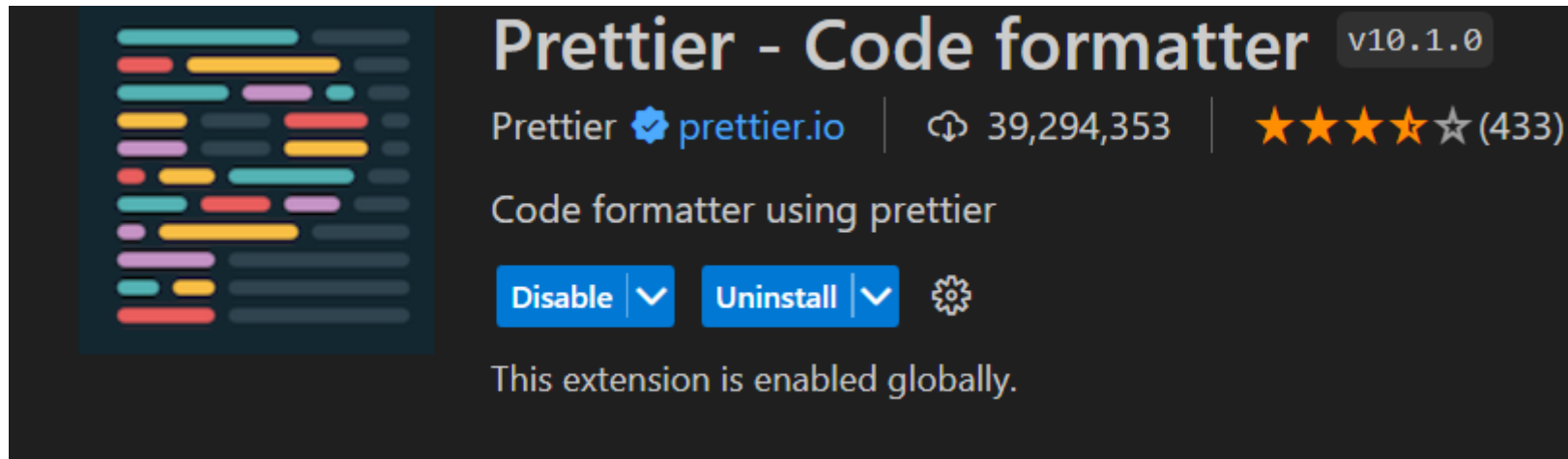
[Learn React](#)

프로젝트 폴더 설정 추가

- .js 파일에서 html emmet 사용 가능하도록 .vscode/settings.json에 추가하기

```
"emmet.includeLanguages": {  
  "javascript": "javascriptreact"  
},
```

- Prettier 확장 설치하기



(추가) Prettier란?

- Prettier는 규칙을 정하고 파일을 저장하면 그 규칙에 따라 코드 스타일을 자동으로 정리한다.
- 코드 스타일 통일 가능!
- 설정
 - **Default Formatter** 검색 > **Prettier - Code formatter** 선택
 - **Format on Save** 검색 > **체크박스 on**

(추가) Prettier란?

- 프로젝트 별로 다른 규칙 적용 가능
 - 프로젝트 최상단 폴더에 `.prettierrc` 파일 생성
 - 아래와 같이 json 형태로 규칙 입력하기

```
{
  "trailingComma": "es5", // 객체, 배열, 함수 등의 후행에 쉼표
  "semi": true,           // 세미콜론
  "singleQuote": true,    // 단따옴표를 기본값으로
  "tabWidth": 2           // 탭 간격
}
```

- 제대로 적용이 안된다면?
 - 껴다 켜고 테스트 or 해당 내용 추가

