

K-Digital Training KDT 풀스택 웹 개발자 양성 부트캠프 3기

# React

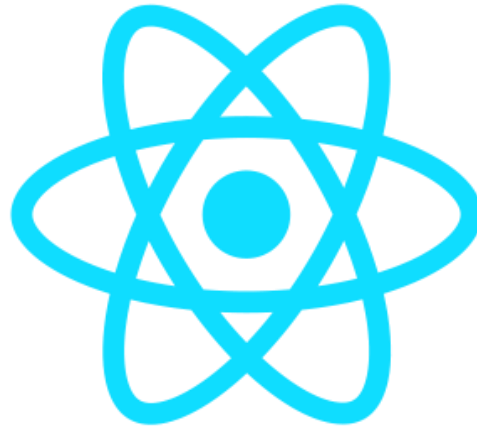
WITH 팀 리처드



# 가장 많이 쓰이는 것은?



ANGULARJS



React JS



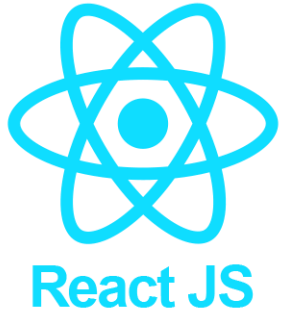
Vue.js

# 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

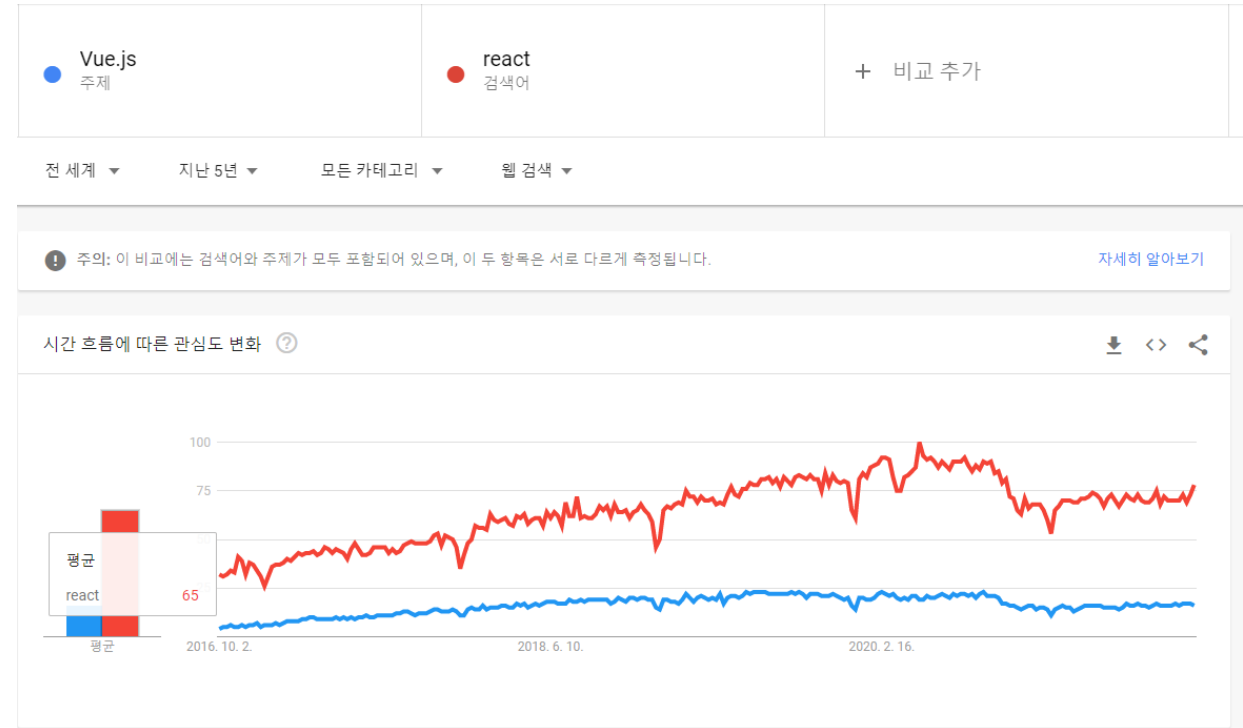
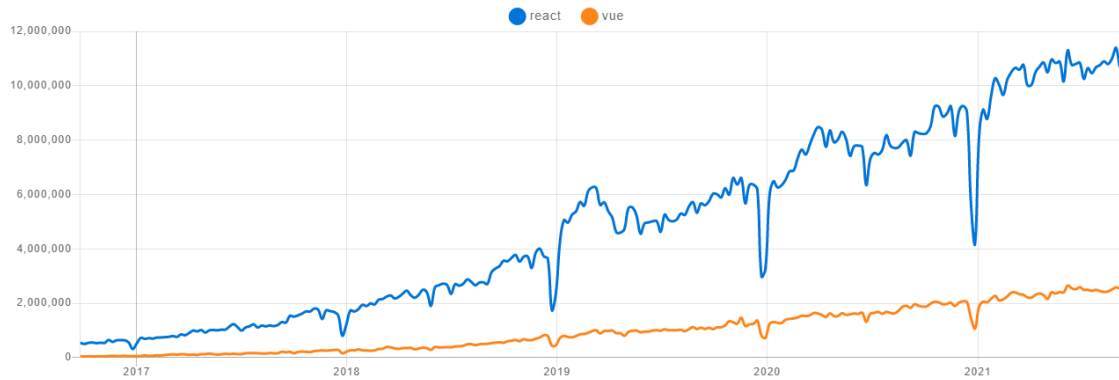


## react vs vue

Enter an npm package...

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

Downloads in past 5 Years

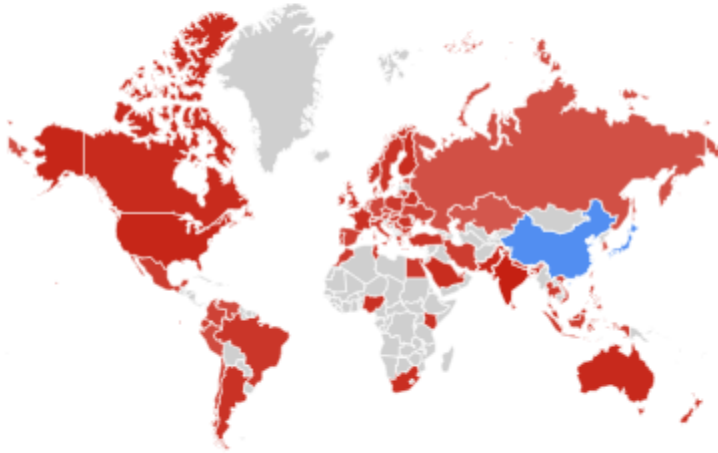


# React VS. Vue.js

지역별 비교 분석

지역 ▼ ⬇ <> 🔗

● Vue.js ● react



색상의 농도는 검색어의 비율을 나타냅니다. [자세히 알아보기](#)

정렬: Vue.js에 대한 관심도 ▼

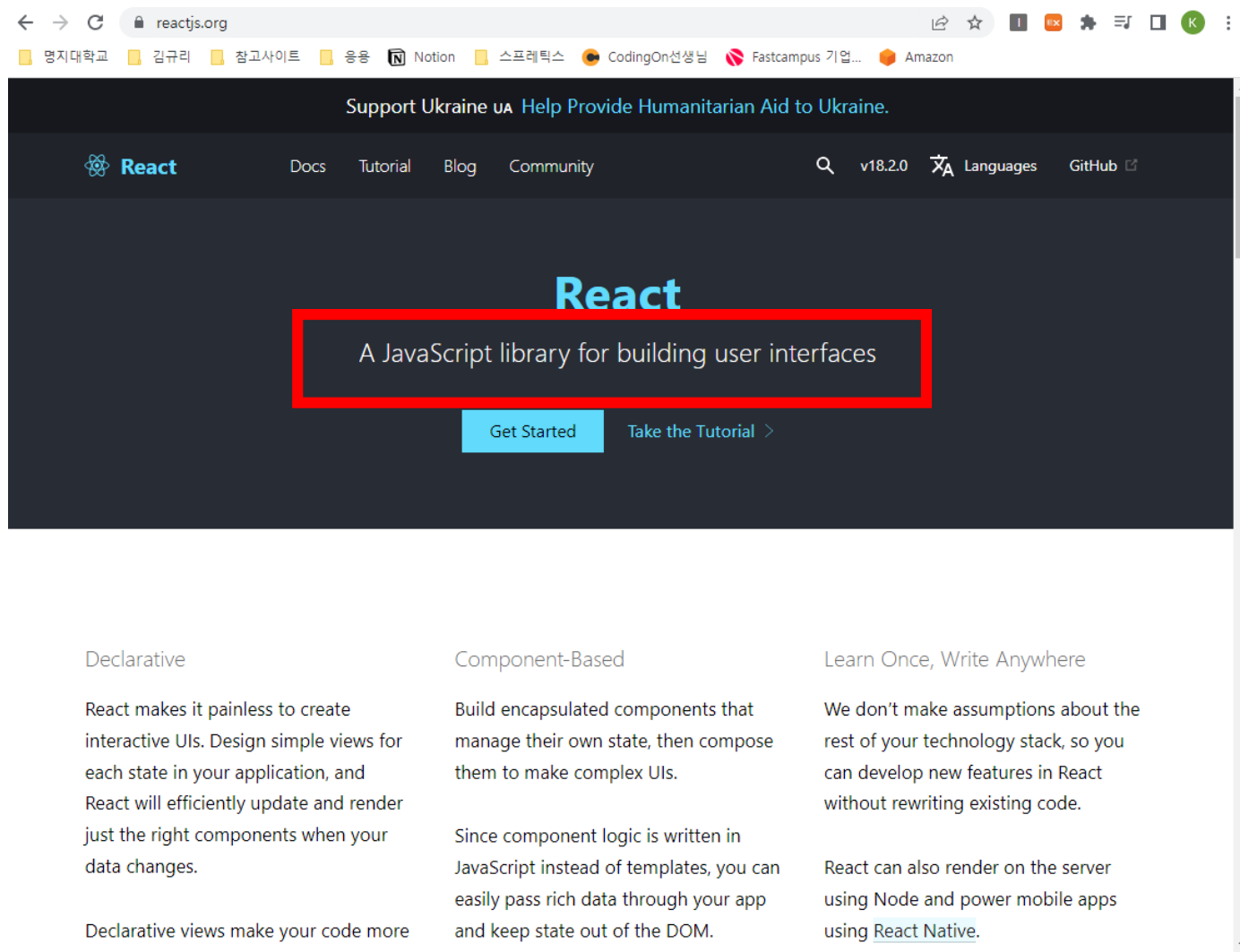
1	중국	<div><div></div><div></div></div>
2	일본	<div><div></div><div></div></div>
3	대만	<div><div></div><div></div></div>
4	카자흐스탄	<div><div></div><div></div></div>
5	대한민국	<div><div></div><div></div></div>

# React



# React란?

- <https://reactjs.org/>
- A JavaScript library for building user interfaces



# 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
- 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

# 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>  
    );  
  }  
}
```

# 특징 2) Component 기반 구조

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

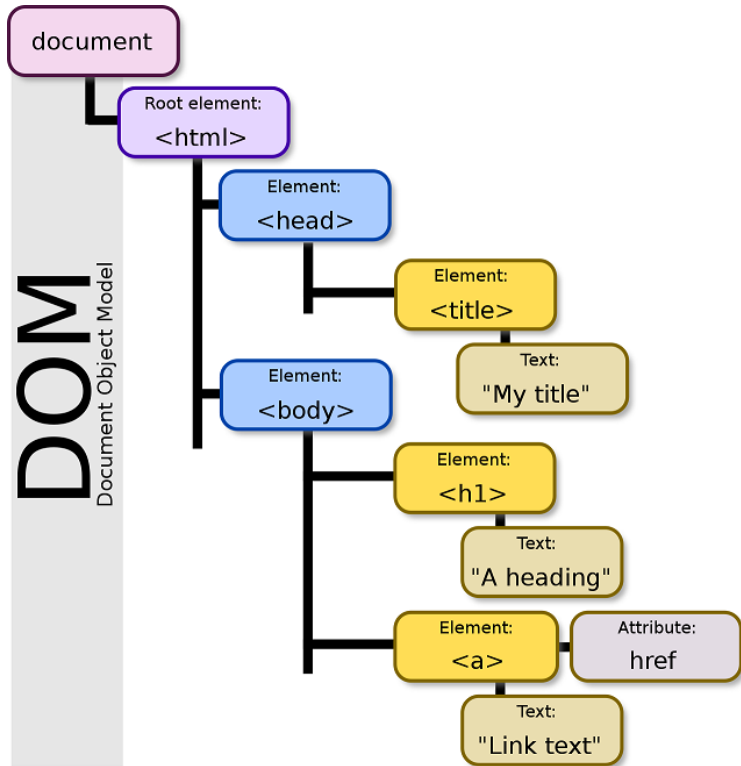
Header, Footer 같은 구조를 컴포넌트로 제작

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



# 특징 3) Virtual DOM

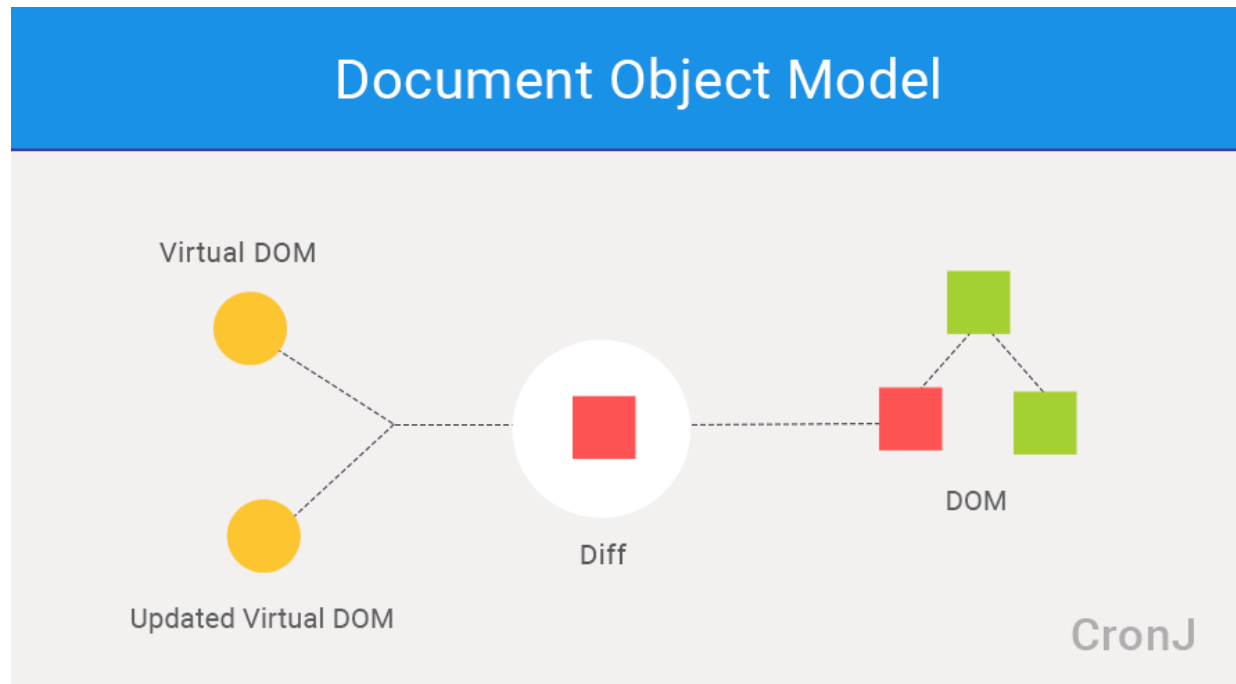
- DOM : Document Object Model ( 문서 객체 모델 )



- 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 -> 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 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 많이 사용된다!
- JSX = Javascript + XML



더 자세한 건 React 프로젝트를  
생성한 후에 배워보자!

# React 프로젝트 생성

# React 프로젝트

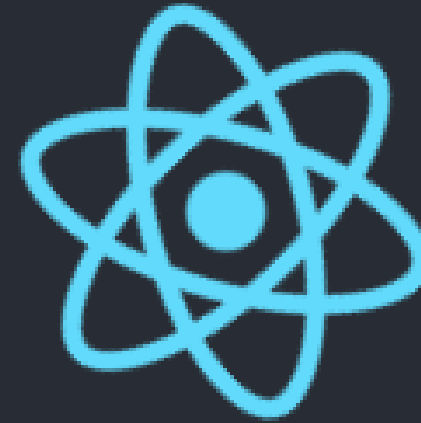
```
npx create-react-app {앱이름}
```

- 앱 이름에는 대문자 사용 불가! 대문자 대신 대시(-) 사용하기
- npx란?
  - npm의 자식 명령어로 npm보다 가볍게 패키지를 구성한다.
  - npm 버전이 5.2 이상일 때, node 버전이 14 이상일 때 사용 가능

# React 프로젝트

i localhost:3000

```
npm start
```



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

[Learn React](#)

# JSX 사용하기

1) 최상위 요소는 형제가 없는 고유한 요소! 반드시 부모 요소 하나가 감싸는 형태를 가져야 한다.

```
function App() {
  return (
    <div>
      <h1>안녕</h1>
      <h2>바보</h2>
    </div>
  );
}
```



```
function App() {
  return (
    <h1>안녕</h1>
    <h2>바보</h2>
  );
}
```





# JSX 사용하기

2) JSX 안에서 JavaScript 사용이 가능하다.

```
function App() {
  const name = "김소연";
  return (
    <>
      <div>{name} 안녕?</div>
      <div>만가워!</div>
    </>
  );
}
```

JSX 안에서 JavaScript 문법을 사용할 때는

**{ }**

로 감싸서 사용해야 한다.

# JSX 사용하기

3) 연산자를 사용할 때는 아래의 방법을 이용해야 한다.

- ① 계산을 다 한 후, 변수에 담아서 JSX 문법에서 보여주기
- ② {} 안에서 삼항 연산자 이용하기 ( if X, for X )

# JSX 사용하기

① 계산을 다 한 후, 변수에 담아서 JSX 문법에서 보여주기

```
function App() {
  let flag = true;
  let txt = "";

  if ( flag ) txt = "True 입니다.";
  else txt = "False 입니다.";

  return (
    <div>{txt}</div>
  );
}
```

② {} 안에서 삼항 연산자 이용하기 ( if X, for X )

```
function App() {
  let flag = true;
  return (
    <div>
      { flag ? (<h1>True입니다.</h1>) : (<h1>False입니다.</h1>) }
    </div>
  );
}
```


```
function App() {
  let flag = true;
  return (
    <div>
      { flag && (<h1>True일 때만 보이는 True 입니다.</h1>) }
    </div>
  );
}
```

# JSX 사용하기

4) 요소에 스타일을 적용할 때는 문자열이 아닌, 객체 형태로 사용

```
function App() {
  return (
    <>
      <div>안녕?</div>
      <div style="background-color:yellow;">반가워!</div>
    </>
  );
}

export default App;
```



```
function App() {
  const styles = {
    backgroundColor: "yellow",
    color: "blue",
    fontSize: "48px",
  };
  return (
    <>
      <div>안녕?</div>
      <div style={styles}>반가워!</div>
    </>
  );
}
```

# JSX 사용하기

5) Style을 이용할 때는 camelCase를 이용!

background-color ❌

backgroundColor ○

font-size ❌

fontSize ○

HTML의 attribute를 쓸 때도 camelCase를 이용

class ❌

className ○

onclick ❌

onClick ○

# JSX 사용하기

## 6) 종료태그는 필수!

기존의 종료 태그가 없는 태그를 사용하더라도 종료 태그를 작성 or Self-closing

잘못된 코드

```
function App() {
  return (
    <>
    <div>안녕?</div>
    <input>
    <br>
    </>
  );
}
```

올바른 코드 1

```
function App() {
  return (
    <>
    <div>안녕?</div>
    <input></input>
    <br></br>
    </>
  );
}
```

올바른 코드 2

```
function App() {
  return (
    <>
    <div>안녕?</div>
    <input />
    <br />
    </>
  );
}
```

# JSX 사용하기

## 7) 주석의 사용

```
function App() {  
  return (  
    // jsx 밖에서의 주석 -> 일반 자바스크립트 주석 문법을 따름  
    <>  
      {/* 여기는 주석입니다 */}  
      <div>안녕?</div> {/* 여기도 주석입니다 */}  
    </>  
  );  
}
```

JSX 안에서 주석을 사용할 때는 **{/\* \*/}** 형태로 사용