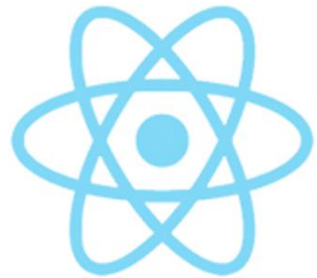


What is React.js

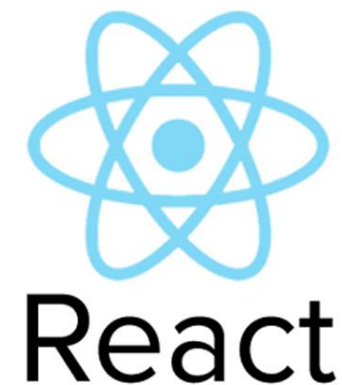
(주)아이와즈 서진규



React

목차

1. 페이스북이 만든 리액트
2. 자바스크립트 UI 라이브러리 리액트
3. 컴포넌트형 구조 리액트
4. 리액트와 JSX
5. 리액트의 Virtual DOM
6. 그렇다면 리액트는 언제 쓰는 것이 올바른가?



1. 페이스북이 만든 리액트

- 현재 16.8.6 버전까지 업데이트 됨

- 버전의 의미

- Semantic Versioning

- 각각의 .에 있는 숫자가 의미가 있음.
 - 맨 앞은 완전 리뉴얼(이전 버전과 호환 안됨)
 - 중간은 기능 추가(이전 버전과 호환은 됨)
 - 세번째는 버그 수정(물론 이전 버전과 호환 됨)

- 오픈소스이며, 사용자의 제안을 기반으로 업데이트됨

- ~기능 추가해주세요
 - 토론
 - 반영 여부 결정
 - 최종적 반영 or 반려



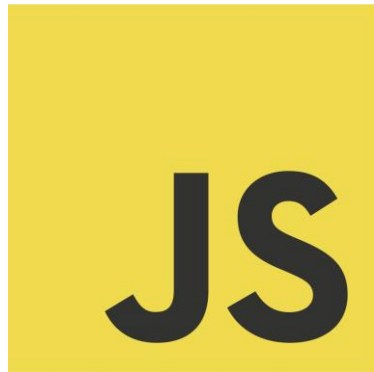
React · license MIT npm v16.8.6 circled passing PRs welcome

React is a JavaScript library for building user interfaces.

- **Declarative:** React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable, simpler to understand, and easier to debug.
- **Component-Based:** Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.
- **Learn Once, Write Anywhere:** We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code. React can also render on the server using Node and power mobile apps using [React Native](#).

2. Javascript UI 라이브러리 React

- React는 라이브러리임!
 - 간혹 프레임워크로 생각하는 사람이 많음 하지만 라이브러리임
 - 둘의 차이
 - 완벽히 기존 Javascript의 Subset이냐
 - Vs 그 외 기타의 것을 강요하는가 (이를테면 특정 백엔드의 사용 등등...)
 - 리엑트는 어느 백엔드에도나 쓸 수 있다. (다만 효율의 차이...)
 - 기존 Javascript의 완벽한 Subset
 - 동등한 특성을 지님
 - 그러므로 This, Bind, Closure 등의 기본 자바스크립트 개념이 매우 중요
 - 또한 ES6 문법을 많이 쓰므로 필수적으로 공부하여 익숙해져야 함
 - Ex) 구조분해할당 (Destructuring Assignment)
 - `const [array1, array2] = someArray;`
 - Ex) 화살표 함수 (Arrow Function)
 - `let a = (num) => num * num;`



3. 컴포넌트형 구조

- 이미 어떠한 비즈니스 로직에 사용된 컴포넌트는 재사용 가능
 - Ex) 한 페이지에 여러 개의 중복된 비즈니스 로직이 있을 때, 활용도 매우 높음
- 컴포넌트는 자체적으로 어떠한 변수(State)를 가질 수 있음
 - 자바스크립트 내부에서 변수 값이 꼬일 일이 없음 (책임의 분리)
 - 각각 컴포넌트는 자신에게 필요한 값만 가짐...
- Hot reloading 지원
 - 값(State)이 바뀌는 순간 UI에 바로 반영됨.
 - Virtual DOM을 사용하여 성능상의 이슈도 최적화 되어있음.
 - Virtual DOM은 추후에 다시 설명...

4. 리액트와 JSX

- JSX는 DOM을 효율적으로 코딩하기 위한 새로운 문법
 - 기존 HTML은 반복문, 제어문 등의 사용이 전혀 불가능
 - 그래서 많은 시도가 있었음...
 - Javascript로 컨트롤
 - JSP의 추가적인 문법
 - 기타 등등.....

4. 리액트와 JSX

- JSX는 Javascript로 컨트롤하는 기존의 방식과는 같으나 자신만의 문법을 만들어냄
- 처음 봤을 때는 약간 괴기할 수 있으나... 익숙해지면 매우 생산성이 높아짐

```
import React, { useState } from 'react';

function CustomInputBox() {
  const [value, setValue] = useState("");

  return (
    <>
      <input value={value} onChange={(e) => { setValue(e.target.value) }}>
      </input>
      <button onClick={() => alert(value)}>
        {value} 값 출력하기!
      </button>
    </>
  )
}

export default CustomInputBox;
```

4. 리액트와 JSX

- 그럼 JSX를 사용해서 장점은?
 - 기존의 Javascript 문법을 위화감 없이 HTML에 접목
 - 반복문(for의 사용)
 - 조건문(if의 사용)

4. 리액트와 JSX

- JSX의 제어 예제
 - If 조건이 맞는 경우, 특정 요소를 렌더링하기

```
{value === '123123' && <p> 123123을 입력했음!!!</p>}
```

123123이란 값을 입력하면 특정 태그가 등장하게 설정

4. 리액트와 JSX

- JSX의 반복 예제
 - For문을 이용하여 특정 요소 반복

```
import React from 'react';

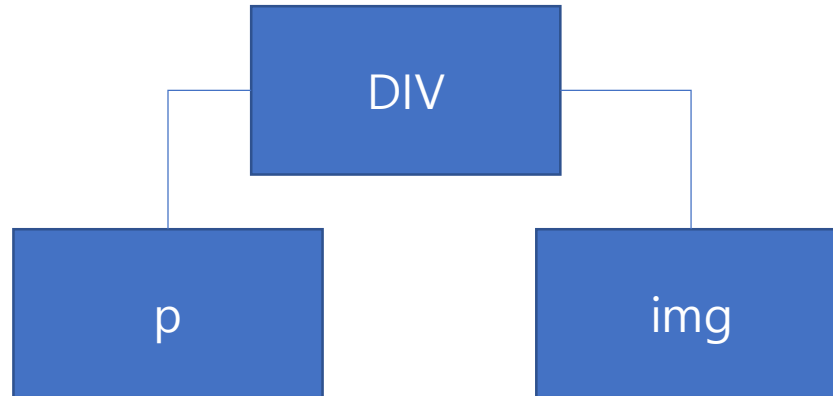
function CustomButton(props) {
  let CustomButton = () => (<<br /><button onClick={() => { alert(`반복될 버튼 ${props.num} 을 클릭하셨습니다.`) }}>반복되는 버튼들 {props.num}</button></>);
  return <CustomButton></CustomButton>
}

function TestFor() {
  let CustomButtons2 = Array(10).fill('').map((ele, idx) => <<CustomButton key={idx} num={idx} />>);
  return CustomButtons2;
}

export default TestFor;
```

5. 리액트의 Virtual DOM

- Virtual DOM이란 무엇인가?
 - DOM을 자바스크립트 오브젝트 형태로 보관 (DOM TREE 형태)
 - 어떤 요소가 변경되어 화면을 다시 렌더링해야 할 때, 변경된 부분만 다시 렌더링
 - 변경하지 않은 부분은 손대지 않음



```
<div>  
  <p>  
    ...  
  </p>  
  <img src='...' />  
</div>
```

6. 그렇다면 리엑트는 언제 사용하는가?

- 매우 동적인 페이지를 만들 때
 - 사용자의 행동 하나하나에 반응해야 할 경우
- 웹을 기반으로 응용 프로그램을 만들고 싶을 때
 - Electron을 이용하여 응용프로그램처럼 만들 수 있음
- 웹을 기반으로 어플리케이션을 만들고 싶을 때
 - React-Native라는 React를 이용한 네이티브 앱 지원 기능이 있음
 - 기존의 앱 업데이트 관련 이슈들 자동 해결 (심사 까다로웠음)
 - OS로부터 자유로움

6. 그렇다면 리액트는 언제 사용하는가?

- 반면에, 정적인 페이지, 상호작용하지 않는 페이지에는 전혀 필요 없다.