1. **컴포넌트 설계가 갖는 장점은 무엇인가. 컴포넌트란 무엇인가. 컴포넌트 설계는 어떻게 해야 하는가. 컴포넌트의 리랜더링은 언제 이루어지는가.**

컴포넌트란 props를 input으로 하고 UI가 어떻게 보여야 하는지 정의하는 React Element를 output으로 하는 함수이다.  
컴포넌트 설계가 갖는 장점은 컴포넌트의 가독성이 매우 높고 간단하여 유지보수하기가 쉬우며, 간편한 UI 수정 및 재사용에 용이하다. 요구 사항이 수시로 바뀌는 현대의 서비스 개발에 특화되어 빠르게 서비스를 피벗하거나 사용자가 급등할 수 있는 환경을 고려해 유연한 환경을 조성하는데 도움이 된다. 여기서 피벗의 의미는 초기에 수립한 사업의 목표나 서비스 운영방식 등을 중간에 바꿔 다른 성격의 사업으로 이전하는 것을 의미한다.

컴포넌트 설계는 대규모의 사용자 인터페이스를 독립적인 마이크로시스템으로 캡슐화하는 것이다. 하나의 컴포넌트와 관련된 모든 메소드 및 API가 컴포넌트의 구조 내에 존재하도록 요구한다.

리액트가 렌더링을 할 때

1. Props가 변경되었을 때
2. State가 변경되었을 때
3. forceUpdate() 를 실행하였을 때
4. 부모 컴포넌트가 렌더링되었을 때

출처 : <https://medium.com/little-big-programming/react의-기본-컴포넌트를-알아보자-92c923011818>

<https://velog.io/@rjs1197/프론트엔드-React를-중심으로-컴포넌트-기반-개발-방법을-알아보자2019>

<https://brunch.co.kr/@kyungwookmin/30>

1. 리액트와 뷰의 차이점과 공통점은 무엇인가.

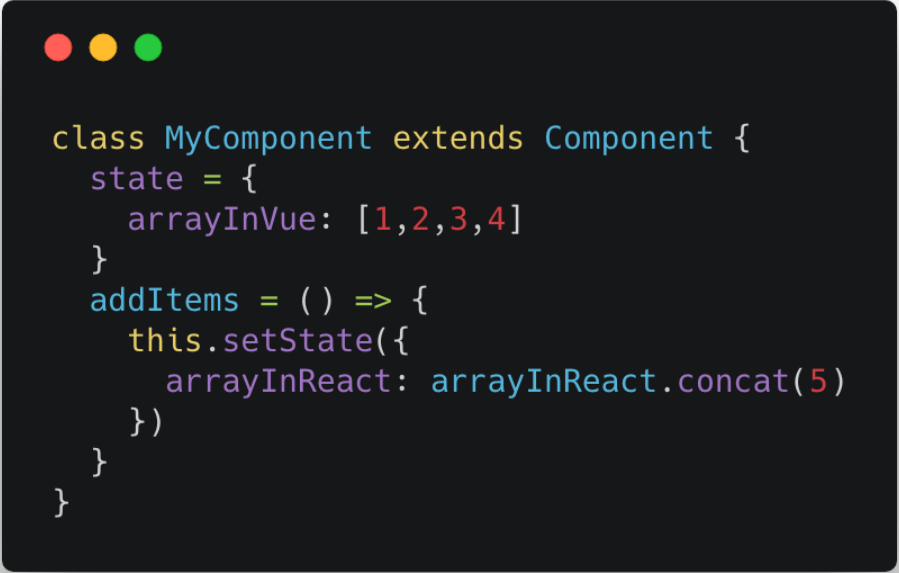
리액트와 뷰의 공통점은 컴포넌트 기반 SPA(Single Page Application) 라이브러리라는 점이다. 컴포넌트 단위로 변화사항을 감지하고 렌더링이 필요한 부분만 바꾸어주는 기능은 양쪽의 프레임워크가 목적론적으로나 방법론적으로나 동일하다. 둘다 virtual DOM을 기반으로 하며, tree알고리즘을 통해 변화사항을 찾아낸다. 둘 다 props, state, lifecycle을 가지고 있다.

리액트와 뷰의 차이점은 먼저 데이터를 핸들링할 때 알 수 있다. 아래의 예제는 Array 타입의 데이터를 핸들링 하는 코드이다.

만약 Array 타입의 데이터에 값을 추가하는 경우, Vue를 이용할 때는 아래와 같이 array 의 내장 메소드인 push 메소드를 이용하면 쉽게 추가할 수 있다.



하지만 React 의 경우에는 기존의 배열의 깊은 복사를 이용하여 setState 에서 값을 변경 해주어야 한다.

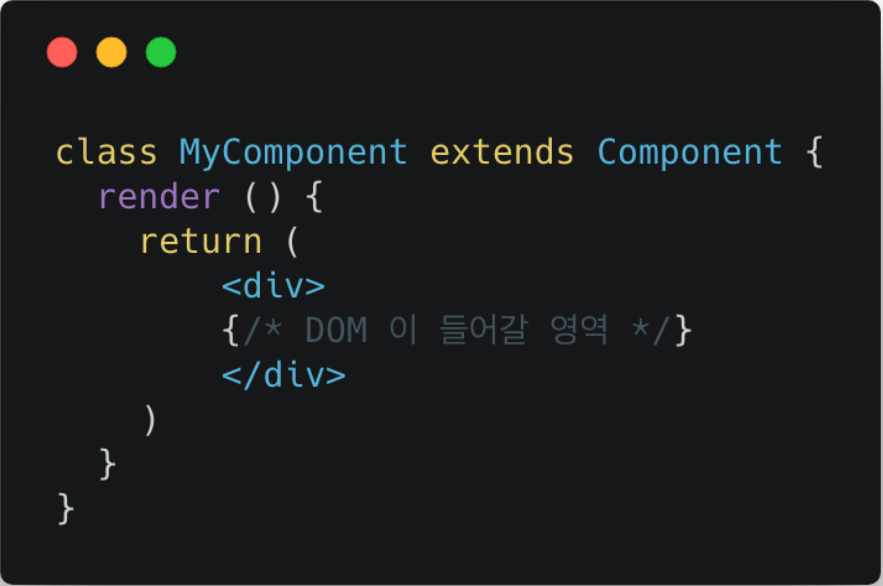


두번째는 가독성에 있다.

일단 Vue 컴포넌트의 가장 큰 특징으로는 SFC(Single File Component) 라는 점이다. SFC는 하나의 파일 안에 DOM, Script, Style을 포함하고 있다는 특징이 있다.



하지만 반대로 React 의 경우는 다음과 같다.



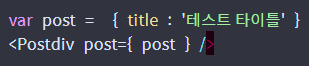
출처 : <https://blog.martinwork.co.kr/review/2018/11/01/between-vuejs-and-reactjs.html>

<https://velog.io/@vraimentres/react-vs-vue-1>

1. STATE와 PROPS를 설명하시오.

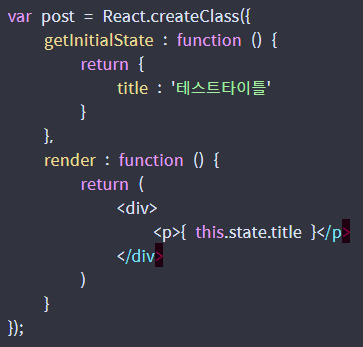
Props는 개발에서 많이들 사용되는 용어인 프로퍼티(properties)의 줄임말입니다. Props는 React에서는 사용자가 컴포넌트에 전달해서 보관하길 원하는 데이터입니다. 즉, 컴포넌트 내에서 데이터가 보관되면, 이 데이터는 수정되지 않고 보존되어야 하는 법칙이 성립됩니다. 만약 props의 값을 변경하고자 할 때에는 컴포넌트 내부가 아닌, 부모 컴포넌트에서 이에 대한 부분이 변경되어야 한다.

간단하게 예제를 살펴보겠다.



간략하게 부모객체에서 자식객체에 props를 넘겨주는 예시입니다. 이처럼, 부모 객체에서는 자식객체에 post라는 데이터를 props 형태로 전달 해 줄수 있습니다. 이는 부모 객체에서 넘겨주는 데이터이기 때문에, 실제 사용하는 컴포넌트 내에서 props의 변경은 원칙적으로 금지되어 있습니다.

State는 컴포넌트 내부에 존재하고 있기 때문에, 상태값 변경이 가능하다는 것입니다.



state에 관련된 간략한 예시입니다. 우리는 post라는 컴포넌트에서 getInitialState를 통해서 컴포넌트 내에서 쓰일 state값을 리턴받습니다. getInitialState는 return을 통해 state 초기값을 반환해 주게 되는데, 이렇게 함으로써 컴포넌트 내부에서는 this.state를 통해 상태값을 제어할 수 있게 됩니다.

render 함수에서는 바로 getInitialState에서 받은 state값 중, title을 출력하고 있음을 알 수 있습니다. 이렇게 초기화된 state값을 출력할 수 있지만, 중간에 state값을 바꿔줄 수도 있습니다.

출처 : <https://trustyoo86.github.io/react/2017/11/18/props-state-react.html>