**1. 컴포넌트 설계가 갖는 장점은 무엇인가. 컴포넌트란 무엇인가. 컴포넌트 설계는**

**어떻게 해야 하는가. 컴포넌트의 리랜더링은 언제 이루어지는가.**

**Component**

- 사용자 정의 태그

- 한 번 정의해 놓으면 여러 곳에서 재 사용할 수 있다

- 실시간으로 변경된 내용을 업데이트하여 유지/보수가 쉽다

- 다른 곳에서도 재사용 할 수 있도록 필요한 최소한의 기능을 넣어 설계하는 것이 유리하다.

**React 컴포넌트가 렌더링을 수행하는 시점**

* Props가 변경되었을 때
* State가 변경되었을 때
* [forceUpdate()](https://reactjs.org/docs/react-component.html#forceupdate)를 실행하였을 때
* **부모 컴포넌트가 렌더링되었을 때**

**참고**

유튜브: 생활코딩 youtu.be/XMb0w3KMw00

칼럼:

<https://medium.com/vingle-tech-blog/react-%EB%A0%8C%EB%8D%94%EB%A7%81-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0-f255d6569849>

**2. 리액트와 뷰의 차이점과 공통점은 무엇인가.**

**React vs Vue**

**React**

-

|  |  |  |
| --- | --- | --- |
| **React** |  | **Vue** |
| **공통점** | | |
| **Virtual Dom 사용** →Real Dom을 얕은 복사한 Virtual Dom 내의 변경사항을 추적하여 전체 트리의 다른 부분에 영향을 주지 않고, 매핑된 실제 DOM상의 특정요소만 업데이트. Real Dom을 직접 조작하는 것에 비해 성능이 뛰어나다.  **Component 기반의 개발** → 웹 UI를 작은 Component 단위로 구성 → Component는 다른 프로젝트에서도 재사용할 수 있고, Component \*캡슐화와 확장이 가능해 개발이 유연해진다. | | |
| **차이점** | | |
| **JSX 문법 (Javascript XML) 사용** → XML 포맷의 템플릿을 직접 자바스크립트에 내장시키는 형태 → 다른 프레임워크에 비해 더 JavaScript에 집중된 형태의 Component | **문법** | **HTML 마크업 기반** → 웹의 뷰(view)를 구성하는 요소인 HTML, CSS, JavaScript 코드를 .vue 확장자를 가진 하나의 파일에 모두 정의 (HTML 마크업 후 그 위에 템플릿 문법을 추가해 컴포넌트의 구조와 구성요소를 정의) → Component의 JavaScript 집중도가 낮다 |
| **단방향** → Model이 변경되면 View가 변경되는 방법만을 허용 → View가 변경된 경우 Model을 변경하는 코드를 작성. → 예측 가능성이 높기에 코드가 더 안정적 → 양방향 binding에 비해 코드 작성시간이 더 필요 | **\*Data binding** | **양방향** → View가 변경되면 Model도 변경, Model이 변경되면 View도 변경되는 처리방식 |
| - 다른 프레임워크에 비해 방대한 에코시스템 - 사용자가 자유롭게 개발에 필요한 모듈을 결정하고 선택할 수 있지만, 사용자에게 책임이라는 부담을 지우기도 한다. - 그래서 Create React App이라는 툴을 제공, 초심자들의 진입 장벽을 낮추었다. | **module** | - CLI 툴을 제공 - 가이드 문서를 통해 특정모듈 권장 |

\* **Data binding**

Model(비즈니스 로직)과 View(사용자에게 보이는 UI) 간에 데이터를 동기화하는 프로세스.

\* **캡슐화 (Encapsulation)**

- 은닉성이 있다: 클래스에 담는 내용 중 중요한 데이터나 기능을 외부에서 접근하지 못하게 할 수 있다.

-  프로퍼티 이름 앞에 \_(underBar) 를 붙힌다: 외부에서 \_(underBar)를 붙힌 프로퍼티나 메서드에 접근하여 사용하지 말라는 암묵적인 약속.

*// Public 프로퍼티*

this.프로퍼티이름 = 값;

*//private/protected 프로퍼티*

this.\_프로퍼티이름 = 값;

**참고**

칼럼:

<https://www.samsungsds.com/global/ko/support/insights/frameworks.html>

[웹 프론트엔드 프레임워크, 무엇을 쓸까?](https://www.samsungsds.com/global/ko/support/insights/frameworks.html" \t "_blank)

[웹 프론트엔드 프레임워크, 무엇을 쓸까?](https://www.samsungsds.com/global/ko/support/insights/frameworks.html" \t "_blank)

[www.samsungsds.com](https://www.samsungsds.com/global/ko/support/insights/frameworks.html" \t "_blank)

<http://library.gabia.com/contents/infrahosting/8284>

[가비아 라이브러리](http://library.gabia.com/contents/infrahosting/8284" \t "_blank)

[IT 콘텐츠 허브](http://library.gabia.com/contents/infrahosting/8284" \t "_blank)

[library.gabia.com](http://library.gabia.com/contents/infrahosting/8284" \t "_blank)

<https://webclub.tistory.com/156>

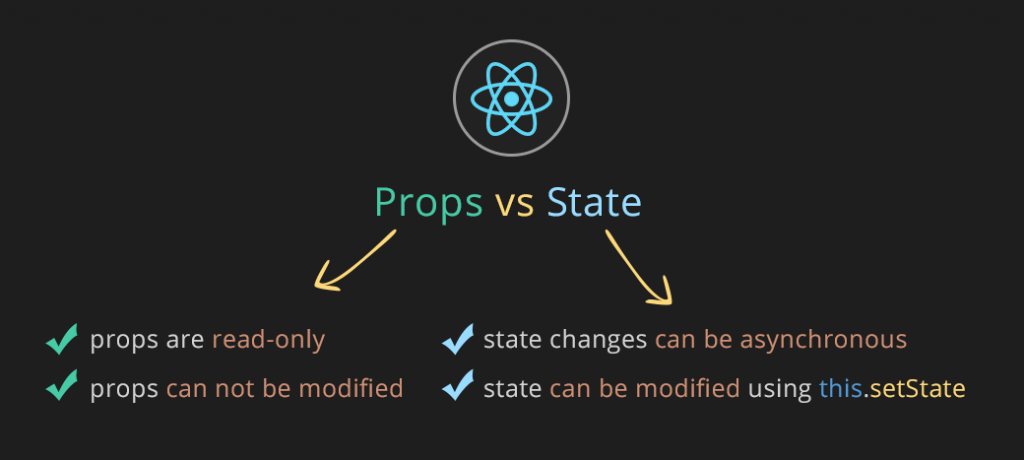
[객체지향 프로그래밍의 캡슐화, 상속, 다형성](https://webclub.tistory.com/156" \t "_blank)

[캡슐화란? 은닉성이다!!! 캡슐화는 중요한 데이터를 보존,보호하는 것입니다. 캡슐화(encapsulation)는 일반적으로 연관 있는 변수와 함수를 클래스로 묶는 작업을 말합니다. 그런데 이 작업은 클래�](https://webclub.tistory.com/156" \t "_blank)

[webclub.tistory.com](https://webclub.tistory.com/156" \t "_blank)

#### 3. STATE와 PROPS를 설명하시오.

#### state / props

https://blogreact.com/   props vs state

**state**

- 사용자가 알아서는 안되는 내부적으로 사용하는 정보

- 부모인 app에서 state라는 내부정보 사용. 그것을 자식에게 전달할 때는 props 사용

- state의 값이 바뀌면 그 state를 가지고 있는 component의 render 함수가 다시 호출됨.

그 render 함수가 다시 호출됨에 따라서 그 render함수 하위에 있는 component들도 각자의 render함수가 있는데,

그 함수들도 싹 호출 됨 -> 화면이 다시 그려진다

1-1

class App extends Component {

constructor(props){

super(props);

this.state={

mode: 'read',

selected\_content\_id:2,

subject:{title:'WEB', sub:'world wide web'},

welcome:{title:'Welcome',desc:'Hello, React!!'},

contents: [

{id:1, title:'HTML', desc:'HTML is for information'},

{id:2, title:'CSS', desc:'CSS is for design'},

{id:3, title:'JavaScript', desc:'HTML is for interactive'}

]

}

}

1-2

<Subject

title={this.state.subject.title}

sub={this.state.subject.sub}

onChangePage={function(){

this.setState({mode:'welcome'});

}.bind(this)}>

</Subject>

- state는직접 변경하면 안되고 function 형태를 빌려 변경해야 한다. App이라고 하는 component가 생성될 때 최초로 실행되는 constructor 함수에서는 1-1과 같이 state값을 정의하지만, 이미 component 생성이 끝난 다음에 동적으로 state값을 바꿀 때는 1-2처럼 함수의 형태를 빌어야한다.(setState)

**props**

- 사용자에게 중요한 정보

- props를 통해 component 조작

**참고**

유튜브: 생활코딩 youtu.be/XMb0w3KMw00