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

* 컴포넌트란 리액트에서 가장 중요한 요소로 리액트로 작성한 화면을 구성하는 기본 단위이다. UI를 재사용 가능한 개별적인 조각, 즉 개별 컴포넌트로 나누면 비슷한 요소가 반복되는 웹 사이트의 화면 구성을 더욱 효율적으로 할 수 있다.
* 컴포넌트를 설계할 때 가장 기본적인 원칙은 단일책임원칙(single responsibility principle)이다. 이는 하나의 컴포넌트가 한 가지의 일을 하는 게 이상적이라는 뜻으로 하나의 컴포넌트가 커져서 여러 기능을 수행하게 되면 더 작은 하위 컴포넌트로 나누는 게 좋다.
* 컴포넌트는 개념적으로 Javascript 함수와 비슷하며 작성하는 방식도 유사하다. 컴포넌트는 크게 프로퍼티, state, 컨텍스트로 구성되어 있다. 프로퍼티(properties)는 props라고도 하는데 상위 컴포넌트에서 하위 컴포넌트로 전달되는 읽기 전용 데이터이다. State는 컴포넌트의 상태를 저장하고 변경할 수 있는 데이터 이다. 컨텍스트는 부모 컴포넌트에서 생성하여 모든 자식 컴포넌트에 전달하는 데이터이다.
* React 컴포넌트가 렌더링을 수행하는 시점은 다음과 같습니다.
  1. Props가 변경되었을 때
  2. State가 변경되었을 때
  3. forceUpdate() 를 실행하였을 때
  4. 부모 컴포넌트가 렌더링되었을 때
* 1~3번의 과정을 통해 컴포넌트가 렌더링될 때, 자식 컴포넌트 또한 같은 과정으로 렌더링이 진행됩니다. 하지만 컴포넌트에서 렌더링 결과에 전혀 영향을 미치지 않는 변경사항이 발생하게 된다면, 불필요한 렌더링이 발생 하므로 성능 손실이 발생합니다. 이는 렌더링에서 수행하는 로직이 많을 수록, 많은 컴포넌트를 출력할 수록 손실은 배가 됩니다.

출처 1) DO it! 리액트 프로그래밍(박호준)

출처 2) <https://ko.reactjs.org/docs/components-and-props.html>

출처 3) <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.

# 3. STATE와 PROPS를 설명하시오.

* props란 리액트에서 상위 컴포넌트가 하위 컴포넌트에 값을 전달할 때 사용하는 읽기 전용 데이터이다. props에서는 JS의 모든 자료형을 사용할 수 있으며 미리 자료형을 선언해줌으로써 리액트 엔진이 props로 전달하는 값의 변화를 효율적으로 감지할 수 있고 실수로 지정되지 않은 자료형을 props에 전달하려고 할 때 경고로 알려준다.
* State란 값을 저장하거나 변경할 수 있는 객체로 버튼을 클릭하거나 값을 입력하는 등의 이벤트와 함께 사용하며 이를 통해 컴포넌트의 상태를 관리할 수 있다. State를 사용할 때는 주의해야 할 부분이 몇 가지 있는데

1. State에 저장되는 객체는 반드시 초기화해야 한다. 이렇게 하지 않으면 내부 함수에서 state 값에 접근할 수 없으므로 만약에 마땅한 초기값이 없을 경우 빈 객체라도 넣어야 한다.
2. state 값을 변경할 때는 반드시 setState() (상태관리함수)를 사용해야 한다. 이렇게 해야만 리액트 엔진이 render() 함수를 호출하여 변경된 state값을 출력할 수 있다.
3. setState() 함수는 비동기로 처리되며, setState() 코드 이후로 연결된 함수의 실행이 완료된 시점에 화면 동기화 과정을 거친다.

출처: DO it! 리액트 프로그래밍(박호준)