**1. REALDOM VS VIRTUALDOM 각각 설명하시오**

DOM은 웹페이지를 핸들링 하기 위한 프로그래밍 인터페이스이고 자바스크립트를 통해서 DOM을 제어해서 웹페이지를 제어할 수 있다.

Virtual DOM은 실제 DOM에서 수행해야 할 모든 변경 사항을 Virtual DOM에서 수행한 다음 실제 DOM에 전달한다. DOM 관리를 자동화하고 추상화하여 개발자가 작업을 보다 쉽게 할 수 있도록 도와준다.

**출처 :** https://code-masterjung.tistory.com/33 [Jacob's Development Blog]

**2. 버츄얼돔은 왜 좋은가. 어떻게 동작하는가. JQUERY의 돔 직접참조에 비해서 무엇이 개선되었는가.**

웹 어플리케이션의 복잡성 증가로 현재 웹 개발 트렌드는 DOM 조작이 느려지기 쉽다. DOM은 트리 구조로써, 업데이트 후에 변경된 요소와 하위 요소를 다시 렌더링 하여 웹브라우저 UI를 업데이트해야 한다. 따라서 UI 구성 요소가 많을수록 DOM 트리를 다시 렌더링 해야 되므로 성능이 저하된다.

Virtual DOM은 DOM 조작에 있어 비효율적인 이유에서 나오게 되었다. Virtual DOM은 DOM 조작을 훨씬 추상화하는 것에 집중되어있다. Virtual DOM 라이브러리들은 거대한 UI 전체를 항상 다시 렌더링 하는 것을 원칙으로 둔다.

프레임워크에 의하여 별도로 정의된 컴포넌트 객체 덩어리의 전/후 변화를 비교(diff)하여, 그리고 반드시 필요할 때에만 변화 내용을 DOM에 기록한다. 이런 접근법을 통하여 Virtual DOM 접근법은 성능뿐 아니라 유지보수성까지 향상시킨다.

**출처 :** <https://brunch.co.kr/@cadenzah/3>

https://pgg-dev.tistory.com/14

**3. 버츄얼돔이 동작하는 DIFF알고리즘에 대해 설명하시오.**

두 개의 트리를 비교할 때, React는 두 엘리먼트의 루트(root) 엘리먼트부터 비교한다. 이후의 동작은 루트 엘리먼트의 타입에 따라 달라진다.

1. **다른 타입의 요소일 때**

두 루트 엘리먼트의 타입이 다르면, React는 이전 트리를 버리고 완전히 새로운 트리를 구축한다. 루트 엘리먼트 아래의 모든 컴포넌트도 언마운트되고 그 state도 사라지고 새로 다시 마운트가 된다.

1. **DOM 엘리먼트의 타입이 같은 경우**

같은 타입의 두 React DOM 엘리먼트를 비교할 때, React는 두 엘리먼트의 속성을 확인하여, 동일한 내역은 유지하고 변경된 속성들만 갱신한다. DOM 노드의 처리가 끝나면, React는 이어서 해당 노드의 자식들을 재귀적으로 처리한다.

1. **같은 타입의 컴포넌트 엘리먼트**

컴포넌트가 갱신되면 인스턴스는 동일하게 유지되어 렌더링 간 state가 유지된다. React는 새로운 엘리먼트의 내용을 반영하기 위해 현재 컴포넌트 인스턴스의 props를 갱신한다. 이때 해당 인스턴스의 componentWillReceiveProps()와 componentWillUpdate()를 호출한다. 다음으로 render() 메소드가 호출되고 비교 알고리즘이 이전 결과와 새로운 결과를 재귀적으로 처리한다.

**출처 :** https://ko.reactjs.org/docs/reconciliation.html