

면접 질문에 대답하며 정리하는 학습 내용

**Q. React와 그 주요 특징에 대해
설명해주세요.**

React의 주요 특징

01

선언형

프로그래밍

무엇을 어떻게 구현할지 보다 무엇을 할지에 초점

개발을 할때 상태에 대한 뷰만 작성

02

컴포넌트 기반

아키텍처

HTML, CSS, JS를 하나로 묶어 개발

각 컴포넌트는 독립적이고 재사용 가능하여, 여러 곳에서 사용할 수 있음

03

Virtual DOM

실제 DOM을 추상화한 개념, 메모리 내에 존재하는 DOM

Virtual DOM을 통해 변경이 필요한 부분만 실제 DOM에 적용

04

단방향 데이터

바인딩

데이터가 부모 컴포넌트에서 자식 컴포넌트로 한 방향으로 흐름

데이터 흐름이 명확하여 예측 가능성 높아지며 단순하여 이해하기 쉬움

**Q. Virtual DOM이란 무엇이며,
실제 DOM과 어떻게 다른가요?**

Dom과 Virtual DOM의 차이점

Real DOM

DOM은 HTML 마크업을
구조화한 것

Virtual DOM

UI 업데이트를 효율적으로 하기 위해 사용

Virtual DOM은 실제 DOM을 추상화
이전 Virtual DOM과 새로운 Virtual DOM을
비교하여 실제로 변경한 부분을 DOM에 적용

Q. Props와 State의 차이점은 무엇인가요?

Props와 State 차이점

Props

부모에서 자식 컴포넌트로 전달하는
읽기 전용 데이터

Props는 직접 변경이 불가능,
변경하려면 부모 컴포넌트에서 변경 또는
업데이트 함수를 호출

차이점

State

State는 컴포넌트 내부에서 관리하는
동적인 데이터

컴포넌트 내부에서 수정 가능

```
// 부모 컴포넌트
function Parent() {
  return <Child data="Hello world" />
}
// 자식 컴포넌트
function Child({data}) {
  return <div>{data}</div>
}
```

Props

```
function Component() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <h1>{count}</h1>
      <button onClick={
        () => setCount(prev => prev + 1)
      }>Click</button>
    </div>
  )
}
```

State

Q. useEffect는 어떻게 작동하나요?

```
useEffect(setup, dependencies?)
```

```
function setup() {  
  // 1. 컴포넌트가 DOM에 추가 되었을때 실행합니다  
  // 2. 그 이후 dependencies가 변경되었을때  
  //   리렌더링할때 cleanup 함수가 있는 경우 먼저 실행 후  
  //   setup 함수가 다시 실행합니다.  
  // 3. DOM에서 제거가 될때 cleanup 함수가 실행합니다.  
  ...  
  return function cleanup() {  
    ...  
  } //optional  
}
```

Setup

```
const dependencies = [  
  // setup 함수내에서 참조하는 모든 reactive 값의 목록  
  // reactive 값은 Props, State  
  ...  
]
```

Dependencies

useEffect 케이스

```
// 모든 렌더링  
useEffect(setup)  
// 컴포넌트가 렌더링 후 동작  
useEffect(setup, [])  
// 컴포넌트가 렌더링 후 동작 + 의존성이 변경될때 동작  
useEffect(setup, [a,b, ...])
```

**Q. React에서 key는
왜 필요한가요?**

Key

비교 알고리즘을 통해 변화를 감지합니다

배열 아이템이 이동, 삽입, 삭제 할때 중요합니다

변화를 정확하게 추론하여 올바른 업데이트합니다.

```
// Before
<ul>
  <li>Apple</li>
  <li>Banana</li>
  <li>Cherry</li>
</ul>

// After
<ul>
  <li>Date palm</li> // Changed
  <li>Apple</li>      // Changed
  <li>Banana</li>      // Changed
  <li>Cherry</li>      // Changed
</ul>
```