

면접 질문에 대답하며 정리하는 학습 내용

Q. 상태 관리가 무엇이고, Redux를 사용하는 이유가 무엇일까요?

상태관리란?

상태는 애플리케이션 내에서 데이터를 의미합니다.
상태 관리는 상태를 효율적으로 관리하는 것을 말합니다.

Redux를 사용하는 이유

애플리케이션의 모든 상태를 하나의 중앙 저장소인 store에서 관리합니다.

이를 통해 모든 컴포넌트가 동일한 상태에 접근하고, 변경될때 일관되게 반영합니다.

Redux는 모든 상태 변화가 Action을 통해서 발생하기 때문에 상태 변화가 예측 가능하고 추적하기 용이합니다.

Q. 컴포넌트 설계 시 재사용성을 높이는 방법은 무엇인가요?

단일 책임 원칙(SRP)

```
const Button = ({
  label,
  onClick,
  type = 'button',
  disabled = false
}) => {
  return (
    <button
      type={type}
      onClick={onClick}
      disabled={disabled}
    >
      {label}
    </button>
  )
}
```

GOOD

```
const Button = ({
  label,
  onClick,
  type = 'button',
  disabled = false
}) => {
  const fn = () => { ... }
  const handleClick = () => {
    fn()
    onClick();
  }
  return (
    <button
      type={type}
      onClick={handleClick}
      disabled={disabled}
    >
      {label}
    </button>
  )
}
```

BAD

합성

```
const Card = ({ imageURL, title, content, footer
}) => {
  return (
    <div className="card">
      {imageURL &&
        <Thumbnail src={imageURL} round />
      }
      <Title>{title}</Title>
      <Content lines={2}>{content}</Content>
      {footer &&
        <Footer>
          {footer}
        </Footer>
      }
    </div>
  );
};
```

GOOD

```
const Card = ({ imageURL, title, content, footer
}) => {
  return (
    <div className="card">
      {imageURL && <img src={imageURL} />}
      <h2>{title}</h2>
      <div>{content}</div>
      {footer &&
        <div className="card-footer">
          {footer}
        </div>
      }
    </div>
  );
};
```

BAD

의존성 역전(DIP)

```
const DataList = ({ data, renderItem }) => {  
  return (  
    <ul>  
      {data.map((item, index) => (  
        <li key={index}>{renderItem(item)}</li>  
      ))}  
    </ul>  
  );  
};
```

GOOD

```
const DataList = ({data}) => {  
  return (  
    <ul>  
      {data.map((item, index) => (  
        <li key={index}>{item.name}</li>  
      ))}  
    </ul>  
  );  
};
```

BAD