React.js
렌더링방식
살펴보기

김혜진

목차

React 렌더링

Render Phase

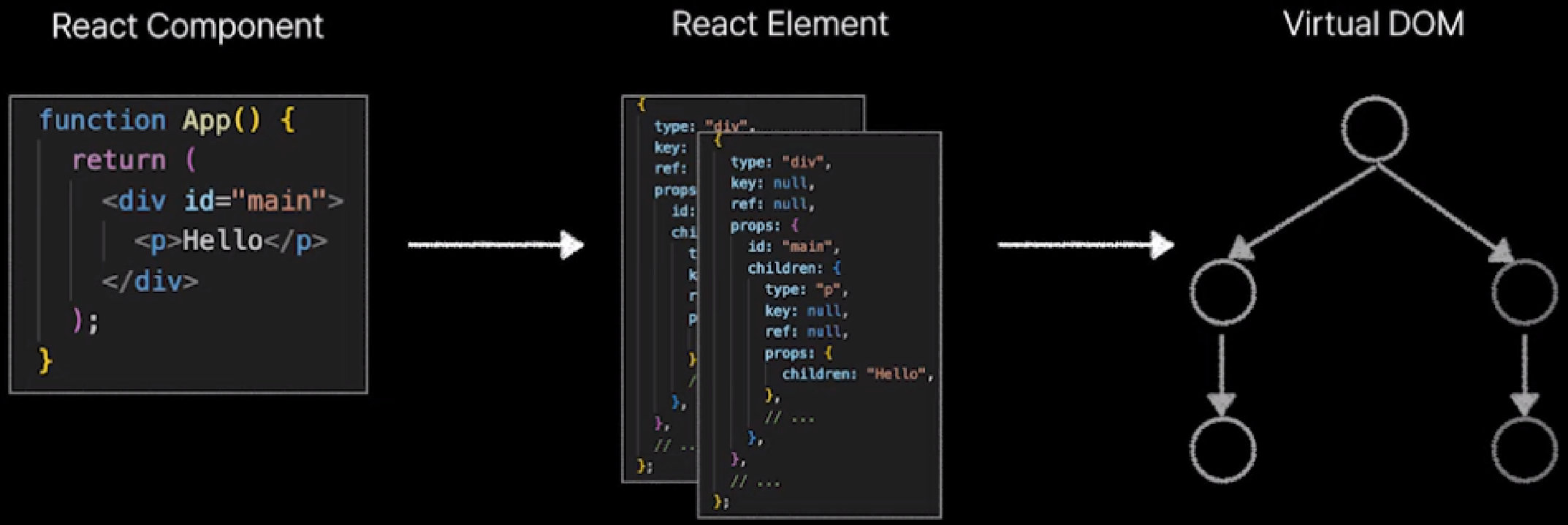
컴포넌트들 계산하고
업데이트 사항을 파악하는 단계

Commit Phase

변경사항을 실제 DOM에
반영하는 단계

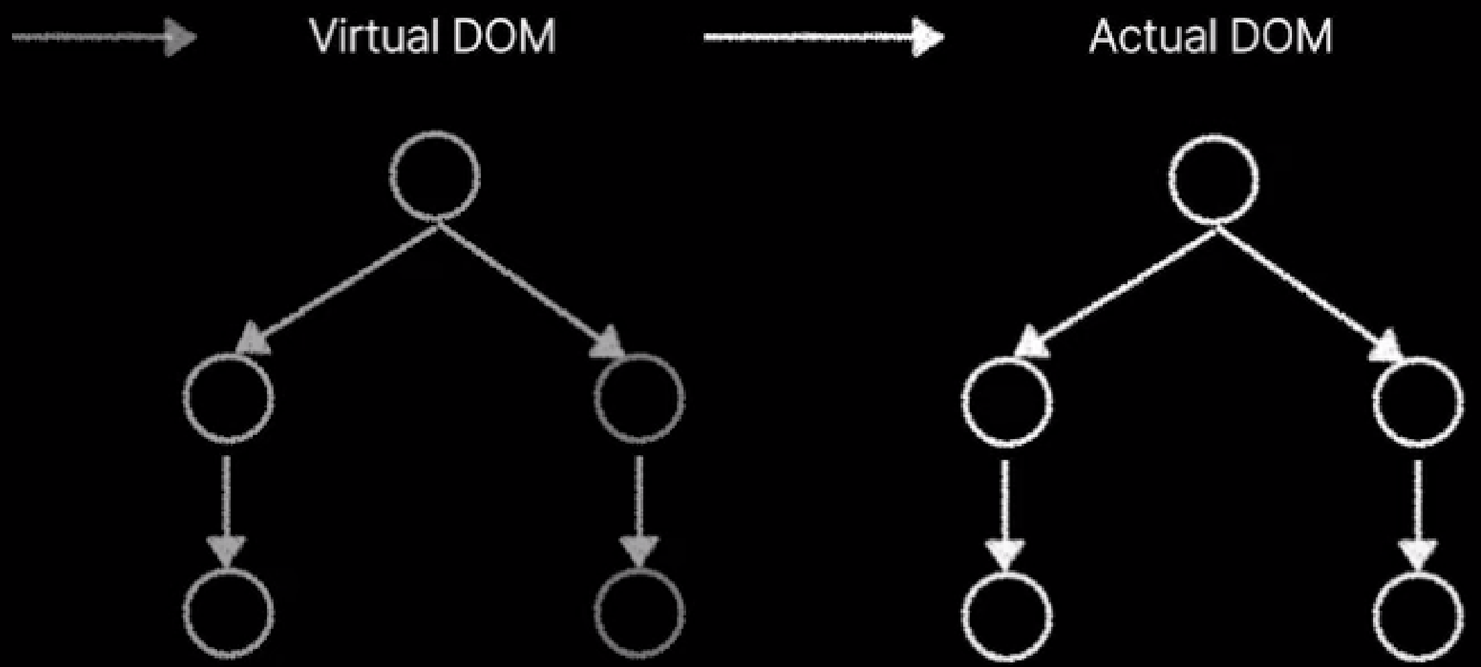
⚙️ Render Phase 정리

- React 컴포넌트가 렌더링 해야 하는 UI를 Virtual DOM이라는 객체 값으로 변환하는 과정

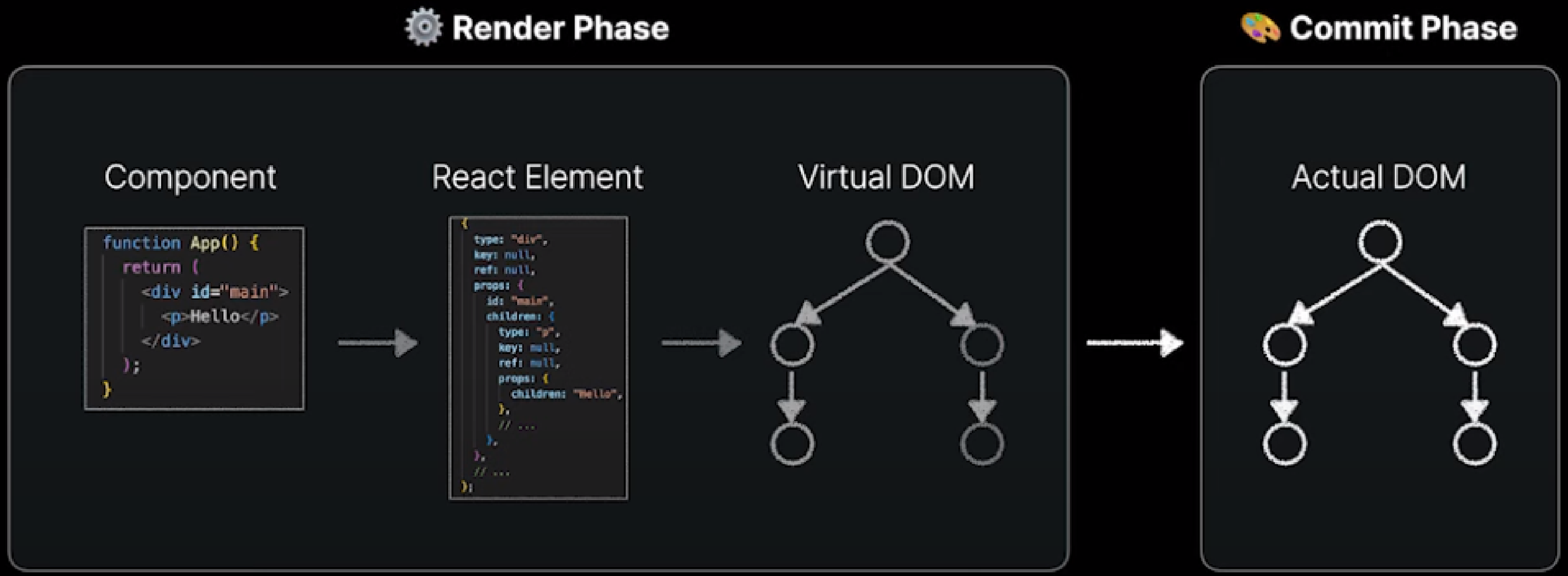


🎨 Commit Phase

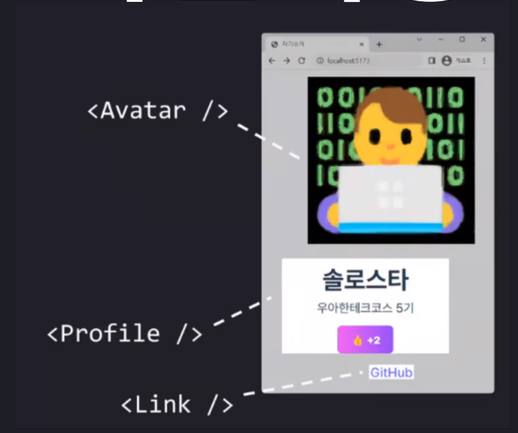
- Virtual DOM을 Actual DOM에 반영함



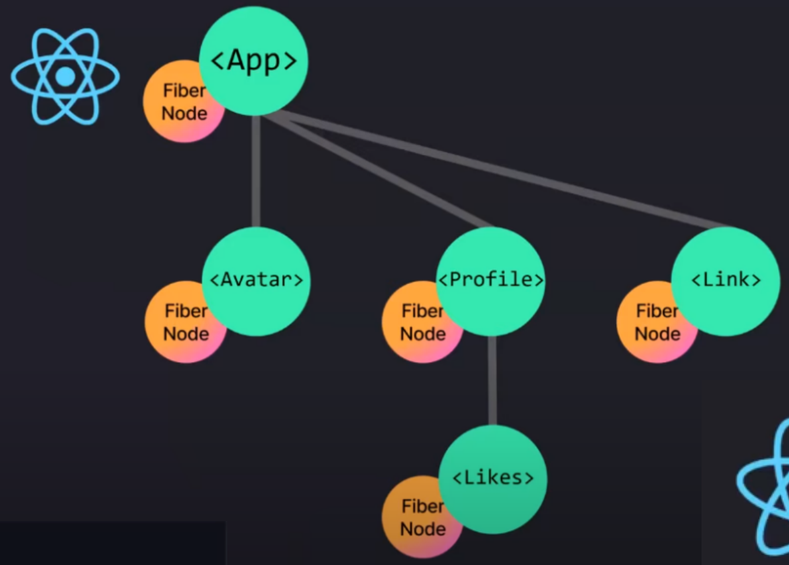
React 렌더링 프로세스

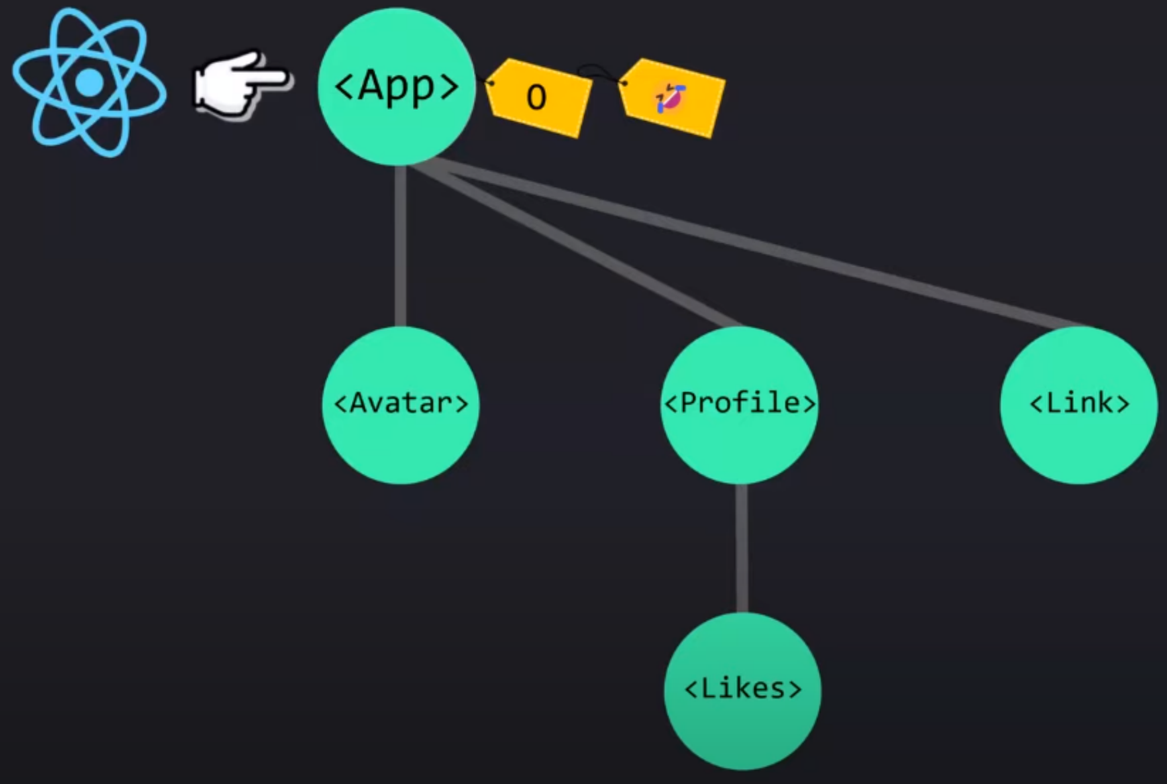


리렌더링




```jsx
function App() {
  const [count, setCount] = useState(0)
  const [emotion, setEmotion] = useState('🤣')

  return (
    <div>
      <Avatar />
      <Profile />
      <Link />
    </div>
  )
}
```





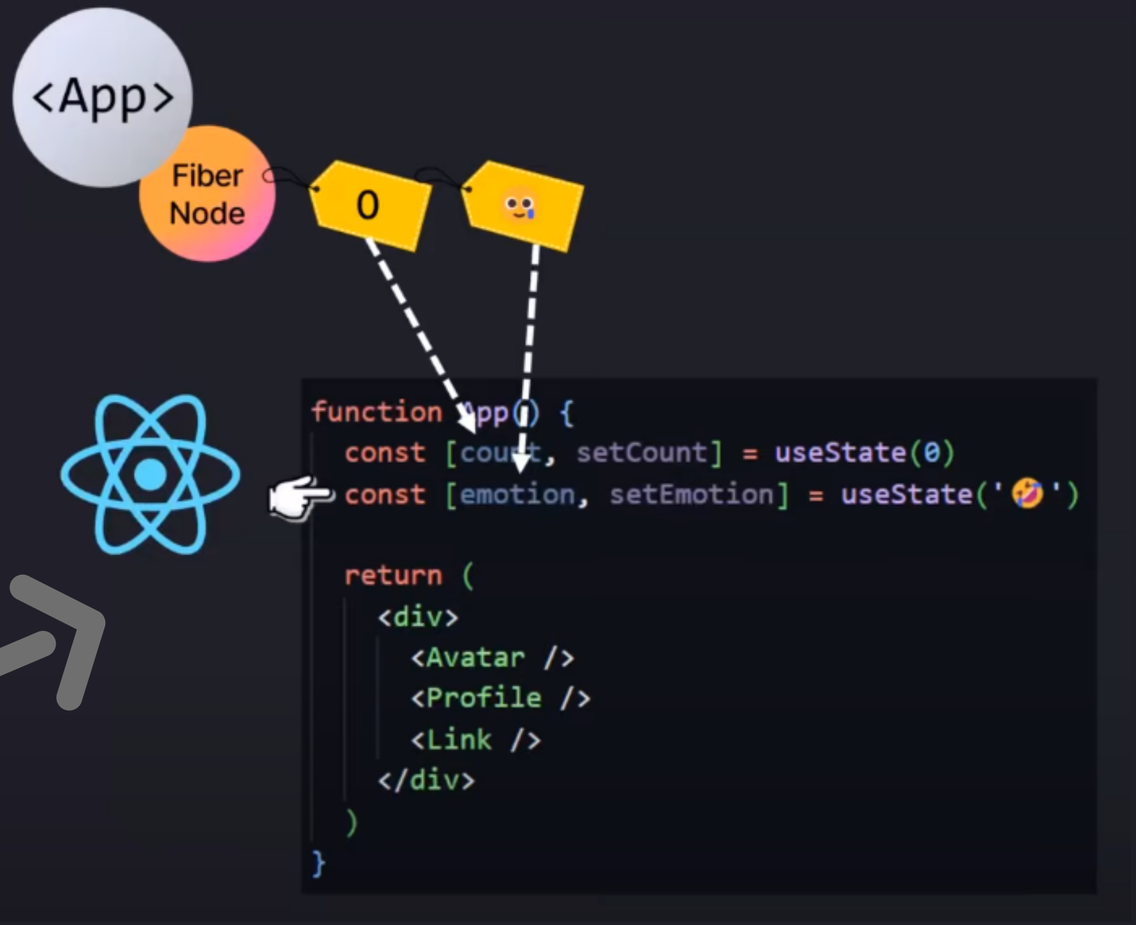
리렌더링

```jsx
function App() {
  const [count, setCount] = useState(0);
  const [emotion, setEmotion] = useState('🤣');

  return (
    <div>
      <Avatar />
      <Profile />
      <h1>나의 상태: {emotion}</h1>
      <button onClick={() => setEmotion('😭')}>슬퍼요!</button>
      <Link />
    </div>
  )
}
```

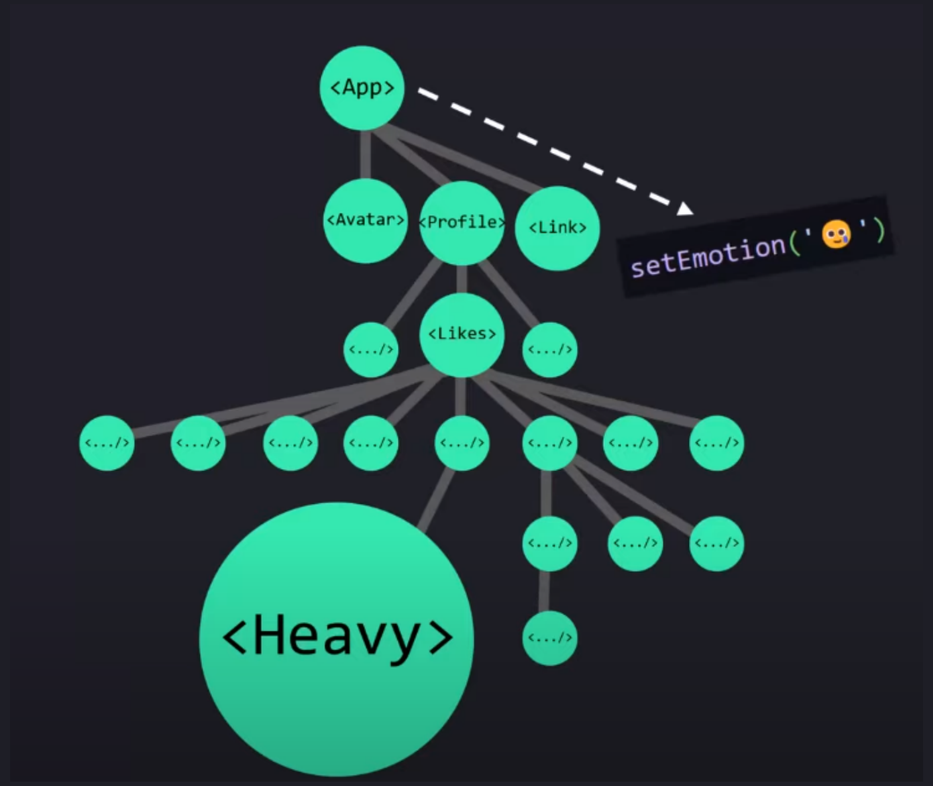
```
setEmotion('😭')
```

1) 상태를 변경

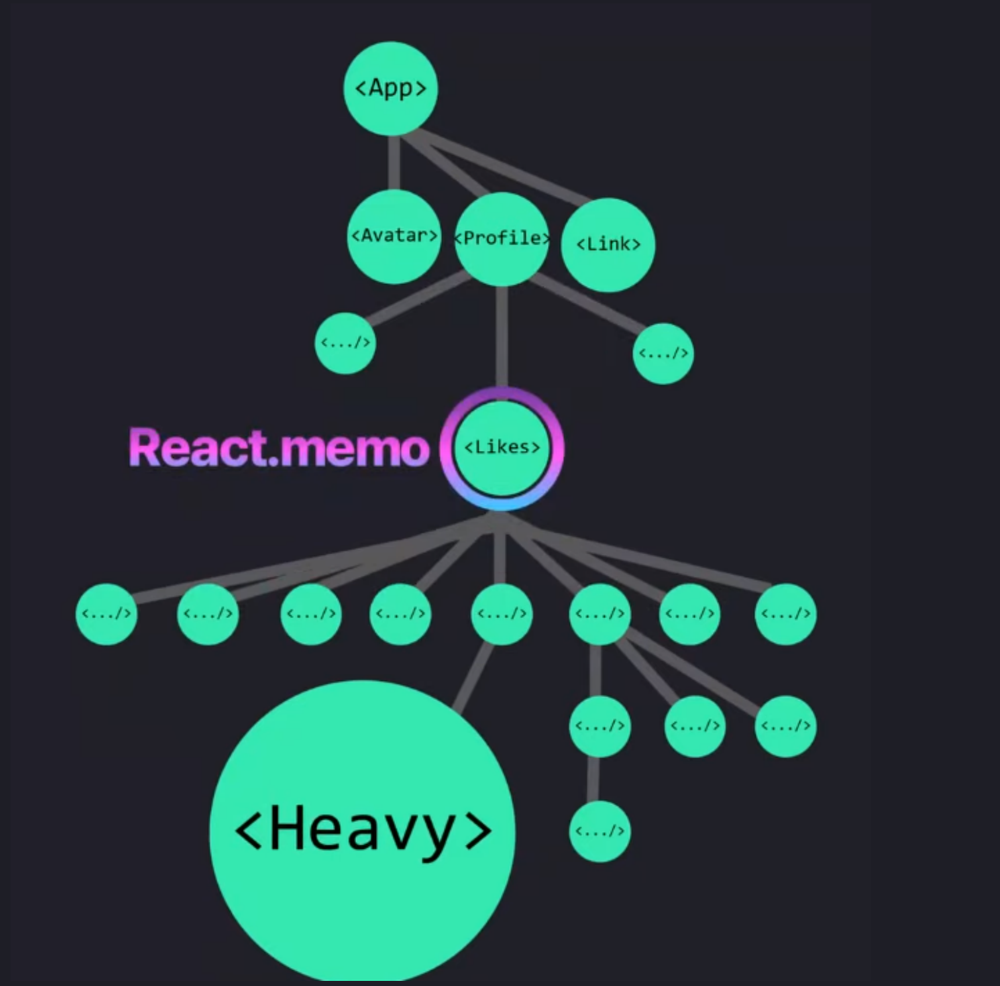
2) 리-렌더링



리렌더링 최적화



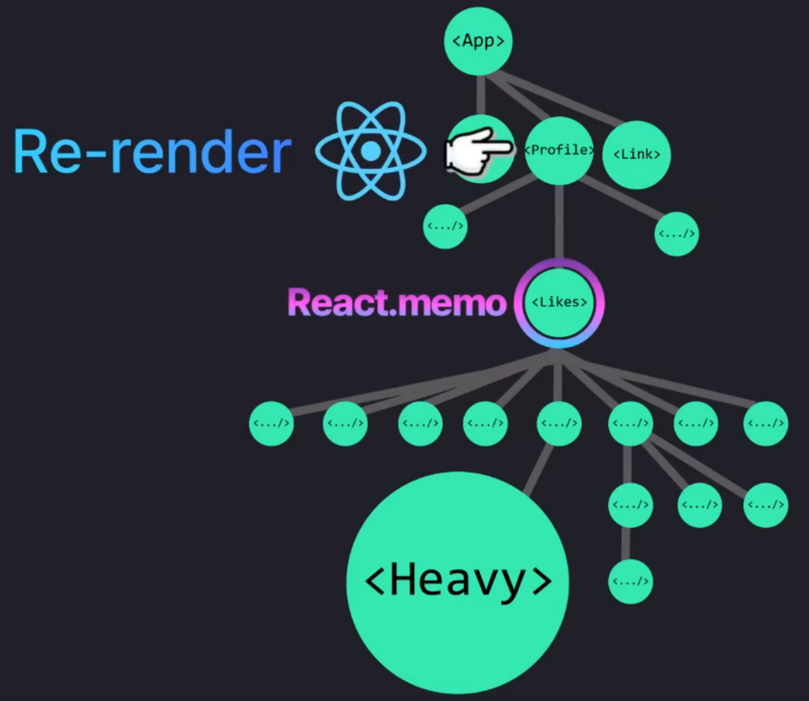
setEmotion으로 상태 변경
전체를 다시 리렌더링

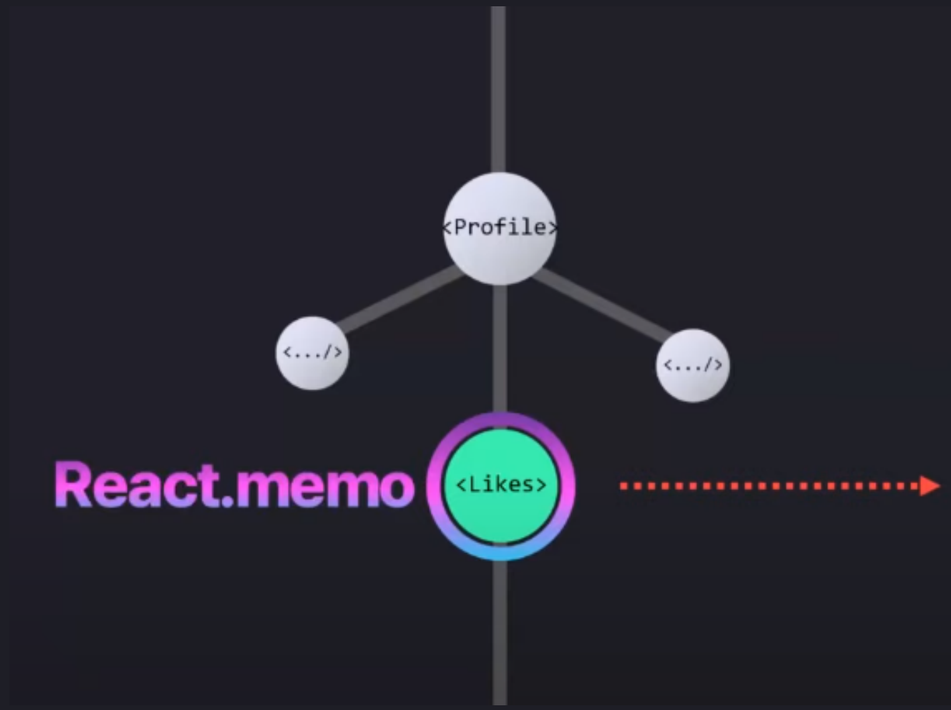


React.memo 사용

리렌더링 최적화

React.memo

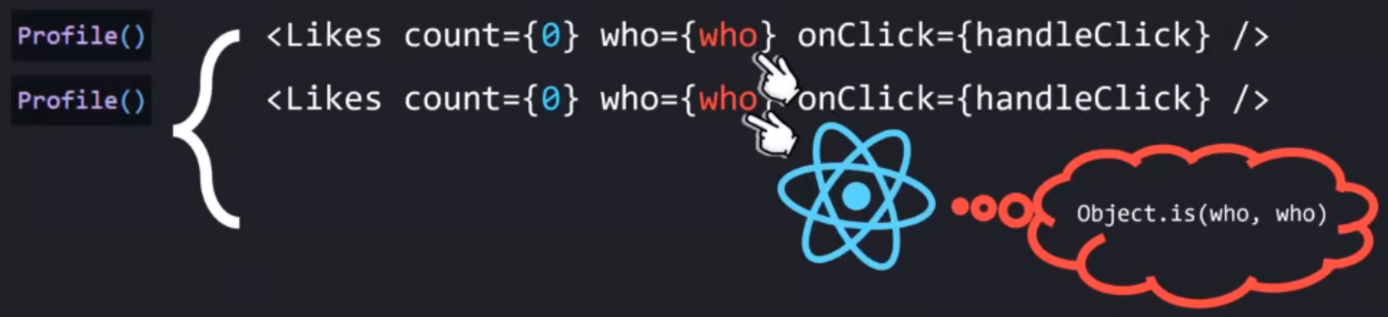





```jsx
const Likes = React.memo(LikesImpl);

function Profile() {
  const [count, setCount] = useState(0)

  return (
    <section>
      <h1>솔로스타</h1>
      <h2>우아한테크코스 5기</h2>
      <Likes
        count={count}
      />
    </section>
  )
}
```

동일한 count 값 넘겨주면
Likes 이전의 렌더링 결과 재사용

리렌더링 최적화



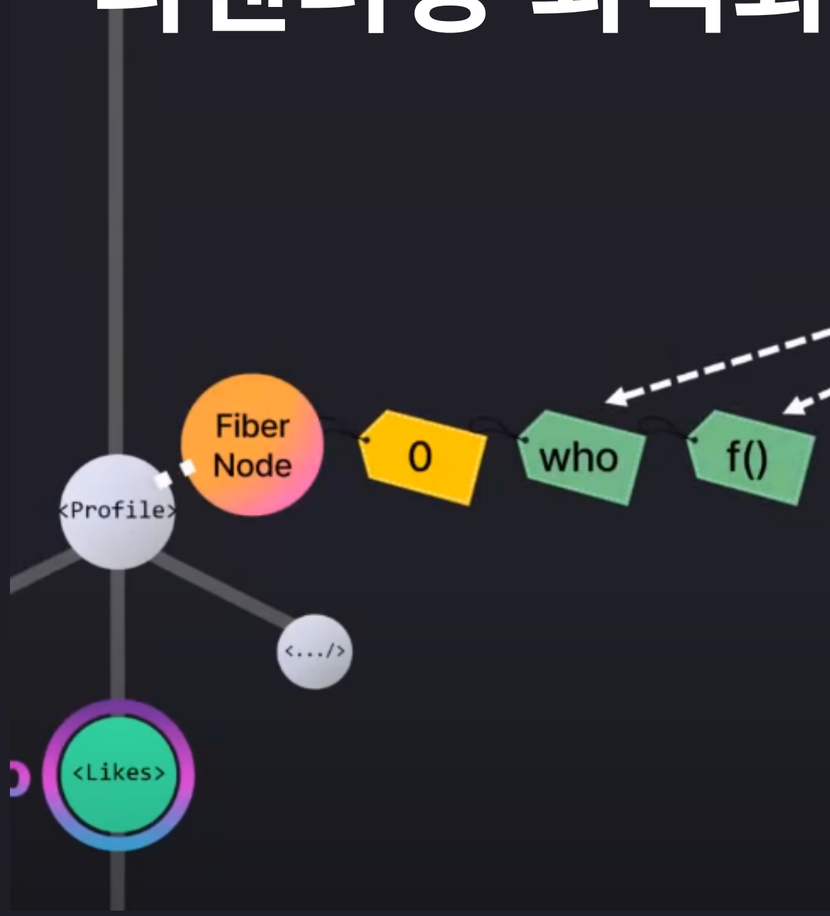
who 객체와
handleClick 함수를
prop으로 넘겨주면?

React.memo

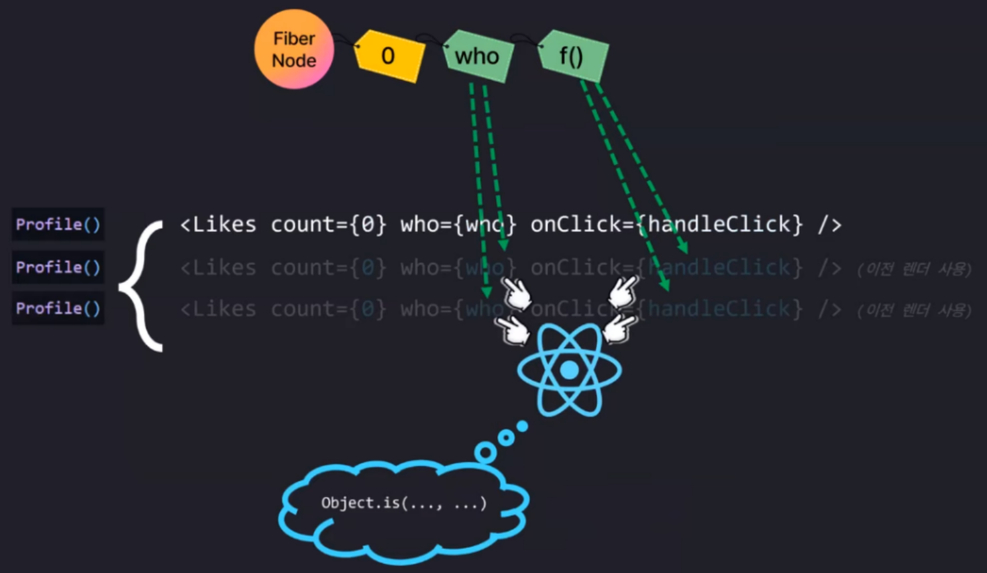
```jsx
const Likes = React.memo(LikesImpl);

function Profile() {
  const [count, setCount] = useState(0)

  const who = { name: 'solo5star' }
  const handleClick = () => setCount((count) => count + 1)

  return (
    <section>
      <h1>솔로스타</h1>
      <h2>우아한테크코스 5기</h2>
      <Likes
        count={count}
        who={who}
        onClick={handleClick}
      />
    </section>
  )
}
```

리렌더링 최적화

useMemo와 useCallback




```jsx
const Likes = React.memo(LikesImpl);

function Profile() {
  const [count, setCount] = useState(0)

  const who = useMemo(() => ({ name: 'solo5star' }), [])
  const handleClick = useCallback(() => setCount((count) => count + 1), [])

  return (
    <section>
      <h1>솔로스타</h1>
      <h2>우아한테크코스 5기</h2>
      <Likes
        count={count}
        who={who}
        onClick={handleClick}
      />
    </section>
  )
}
```



who 객체와 handleClick 함수를
Fiber Node에 저장

참고자료

When to useMemo and useCallback (번역)

https://goongoguma.github.io/2021/04/26/When-to-useMemo-and-useCallback/

React.js의 렌더링 방식 살펴보기 - 이정환 | 2023 NE(O)RDINARY CONFERENCE

https://www.youtube.com/watch?v=N7qlk_GQRJU

[10분 테코톡] 솔로스타의 React 렌더링

https://www.youtube.com/watch?v=eBDj0B0HbEQ