

React 최적화

2회차 이야기

1. `useState`는 컴포넌트 내에서 사용
 - a. Batch
 - b. Update queue
2. `useReducer`는 가급적 `Context API`와 같이
3. `useMemo`는 1ms이상 걸리는 경우
4. `useCallback`은 가급적 `child`에게 내려주는 `prop`인 경우에
5. `useEffect`는 가급적 쓰지마라
6. 전역상태 툴은 상황에 맞는걸로 사용하면 된다

Lighthouse

1. 구글에서 만든 퍼포먼스 측정 도구
 - a. 다양한 지표들이 있음
 - b. 개인적으로 크게 의미 없다고 생각함
 - c. 하지만 면접에서 물어볼 수 있으니 알아두면 좋음

2. 3가지 타입:

<https://github.com/GoogleChrome/lighthouse/blob/HEAD/docs/user-flows.md>

- a. Navigation
- b. Timespan
- c. Snapshot

Profiler

1. React에서 제공하는 성능 측정도구
 - a. <https://react.dev/reference/react/Profiler>
2. 컴포넌트별로 렌더링 시간을 확인할 수 있음
3. 용어설명
 - a. phase: mount, update, nested-update
 - b. actualDuration: Profiler를 렌더링하는데 걸린 시간
 - c. baseDuration: 아무런 최적화 없이 Profiler를 렌더링할 때 걸릴 시간 (예상치)
 - d. startTime: rendering을 시작한 시간
 - e. endTime: rendering이 commit된 시간

개발자도구 - Performance

1. 개발자도구 탭에서 볼 수 있는 내용들
2. 지표를 보고 개선할 내용들은 크게 없고 개념을 알 수 있음
3. Web Vital
 - a. <https://web.dev/vitals/#core-web-vitals>
 - b. LCP - Largest Contentful Paint
 - c. FID - First Input Delay
 - d. CLS - Cumulative Layout Shift

Next.js - First Load JS

1. 클라이언트 최초 rendering
2. 성능 측정 후 개선사항 도출
3. @next/bundle-analyzer
 - a. <https://github.com/vercel/next.js/tree/canary/packages/next-bundle-analyzer>

[아하!모먼트] 회사를 선택하는 기준

1. 회사 규모에 따라 다름

- a. 초기 스타트업
- b. 어느정도 성장한 스타트업 (시리즈 C ~)
- c. 대기업

2. 인터뷰 하면서 얻는 정보

- a. 개발팀 규모
- b. 개발문화
- c. 리더 검증