

'23. 05. 14. (SUN) 신현호 (@SWARUVY)

Item 18. 맵핑된 타입을 사용하여 값을 동기화하기



어떻게하면 이상적으로 최적화를 할 수 있을까요?

```
interface ScatterProps {  
  // The data  
  xs: number[];  
  ys: number[];  
  
  // Display  
  xRange: [number, number];  
  yRange: [number, number];  
  color: string;  
  
  // Events  
  onClick: (x: number, y: number, index: number) => void;  
}
```

여시 코드는 위와 같습니다

보수적 접근법

```
function shouldUpdate(
  oldProps: ScatterProps,
  newProps: ScatterProps
) {
  let k: keyof ScatterProps;
  for (k in oldProps) {
    if (oldProps[k] !== newProps[k]) {
      if (k !== 'onClick') return true;
    }
  }
  return false;
}
```

새로운 속성이 추가되면 **shouldUpdate**는
값이 변경될 때 마다 차트를 다시 그릴 것

실패에 열린 접근법

```
function shouldUpdate(
  oldProps: ScatterProps,
  newProps: ScatterProps
) {
  return (
    oldProps.xs !== newProps.xs ||
    oldProps.y !== newProps.y ||
    oldProps.xRange !== newProps.xRange ||
    oldProps.yRange !== newProps.yRange ||
    oldProps.color !== newProps.color
    // (no check for onClick)
  );
}
```

차트를 불필요하게 다시 그리는 단점은 해결,
하지만 실제로 차트를 다시 그리는 경우를 누락

사실 둘 모두 이상적이지 않아요

```
interface ScatterProps {  
  xs: number[];  
  ys: number[];  
  // ...  
  onClick: (x: number, y: number, index: number) => void;  
  
  // 참고: 여기에 속성을 추가하려면, shouldUpdate를 고치세요!  
}
```

새로운 속성이 추가될 때 직접 `shouldUpdate`를 고치는게 나아요
하지만 이 방법또한 최선은 아니고, 이를 타입 체커가 대신 하는게 좋아요

```
const REQUIRES_UPDATE: {[k in keyof ScatterProps]: boolean} = {
  xs: true,
  ys: true,
  xRange: true,
  yRange: true,
  color: true,
  onClick: false,
};

function shouldUpdate(
  oldProps: ScatterProps,
  newProps: ScatterProps
) {
  let k: keyof ScatterProps;
  for (k in oldProps) {
    if (oldProps[k] !== newProps[k] && REQUIRES_UPDATE[k]) {
      return true;
    }
  }
  return false;
}
```

핵심은 매핑된 타입 & 객체!

[k in keyof ScatterProps]는 타입 체커에게 정보를 제공합니다.
(REQUIRES_UPDATE가 ScatterProps와 동일한 속성을 가져야 함)

```
interface ScatterProps {
  // ...
  onDoubleClick: () => void;
}
```

이후 새로운 속성은
이런식으로 추가될겁니다

```
const REQUIRES_UPDATE: {[k in keyof ScatterProps]: boolean} = {
  // ~~~~~ 'onDoubleClick' 속성이 타입에 없습니다.
  // ...
};
```

그렇게 되면 REQUIRES_UPDATE의 정의에는 오류가 생길거예요
(타입 체커 동작)

요약

- 매핑된 타입을 사용하여 관련된 값과 타입을 동기화하는걸 추천합니다
- 인터페이스에 새로운 속성을 추가할 때, 매핑된 타입을 고려하세요