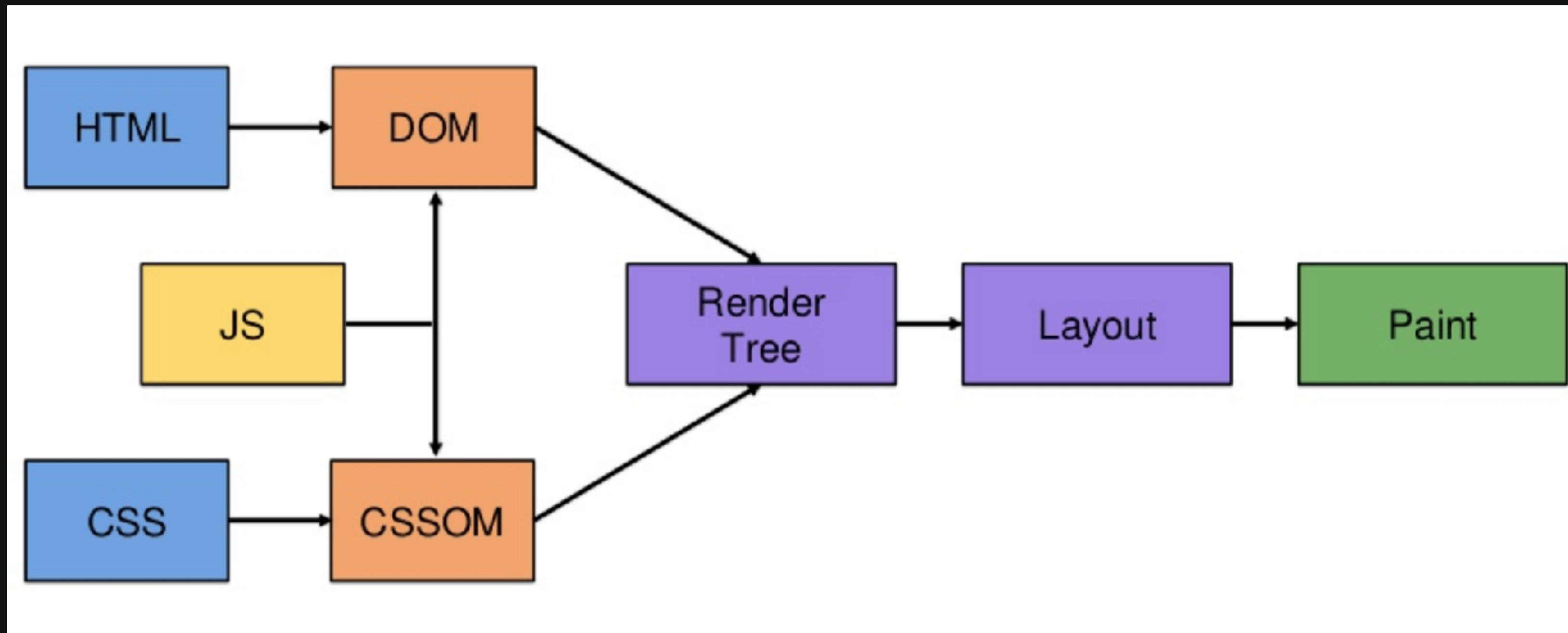


리액트의 렌더링

리액트에서 렌더링은 어떻게 일어나는가?

렌더링이란?

HTML과 CSS 리소스를 기반으로 웹페이지에 필요한 UI를 그리는 과정



렌더링이 발생하는 시점

1. 최초 렌더링: 처음 애플리케이션에 진입했을 때
2. 리렌더링: 그 이후의 모든 렌더링
 - 1) 상태가 변경되는 경우
 - useState의 setter가 실행되는 경우
 - useReducer의 dispatch가 실행되는 경우

렌더링이 발생하는 시점

2) key props가 변경되는 경우

key에 index를 사용하면 안되는 이유!

리액트 fiber는 sibling으로 형제 컴포넌트를 구별

key가 없다면 → 내용이 같아도 구별이 불가능

```
{/* Before */}
<ul>
  <li>first</li>
  <li>second</li>
</ul>
```

```
{/* After */}
<ul>
  <li>first</li>
  <li>second</li>
  <li>third</li>
</ul>
```

```
{/* Before */}
<ul>
  <li>first</li>
  <li>second</li>
</ul>
```

```
{/* After */}
<ul>
  <li>third</li>
  <li>first</li>
  <li>second</li>
</ul>
```

```
{/* Before */}
<ul>
  <li key="1">first</li>
  <li key="2">second</li>
</ul>
```

```
{/* After */}
<ul>
  <li key="3">third</li>
  <li key="1">first</li>
  <li key="2">second</li>
</ul>
```

key에 index를 사용하면 안되는 이유!

만약 key에 index를 사용하게되면?

React App

localhost:3000

추가 삭제

Show Problem Example

철수, idx: 0	여기는 철수 데이터!!!!
영희, idx: 1	
민수, idx: 2	

React App

localhost:3000

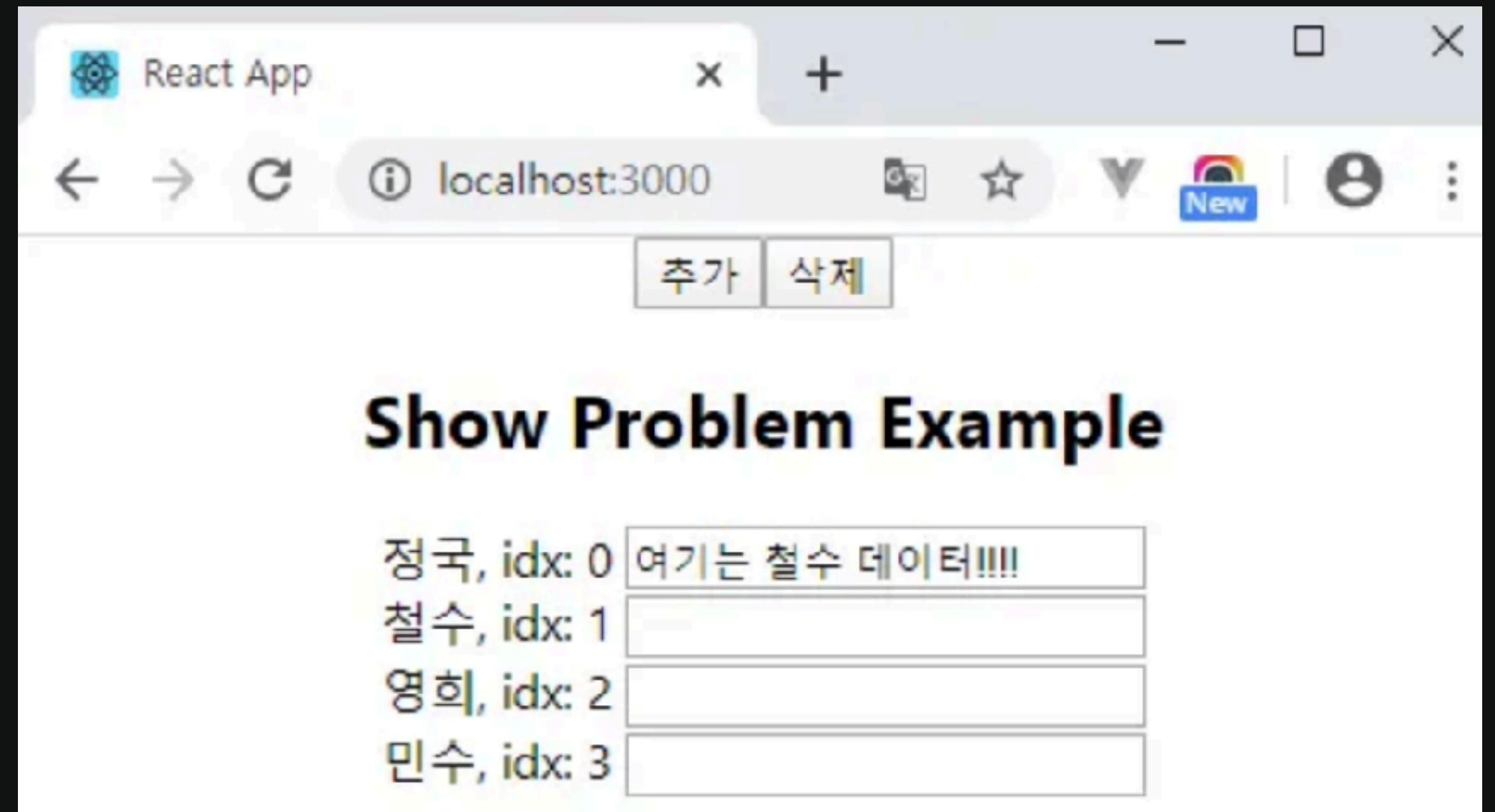
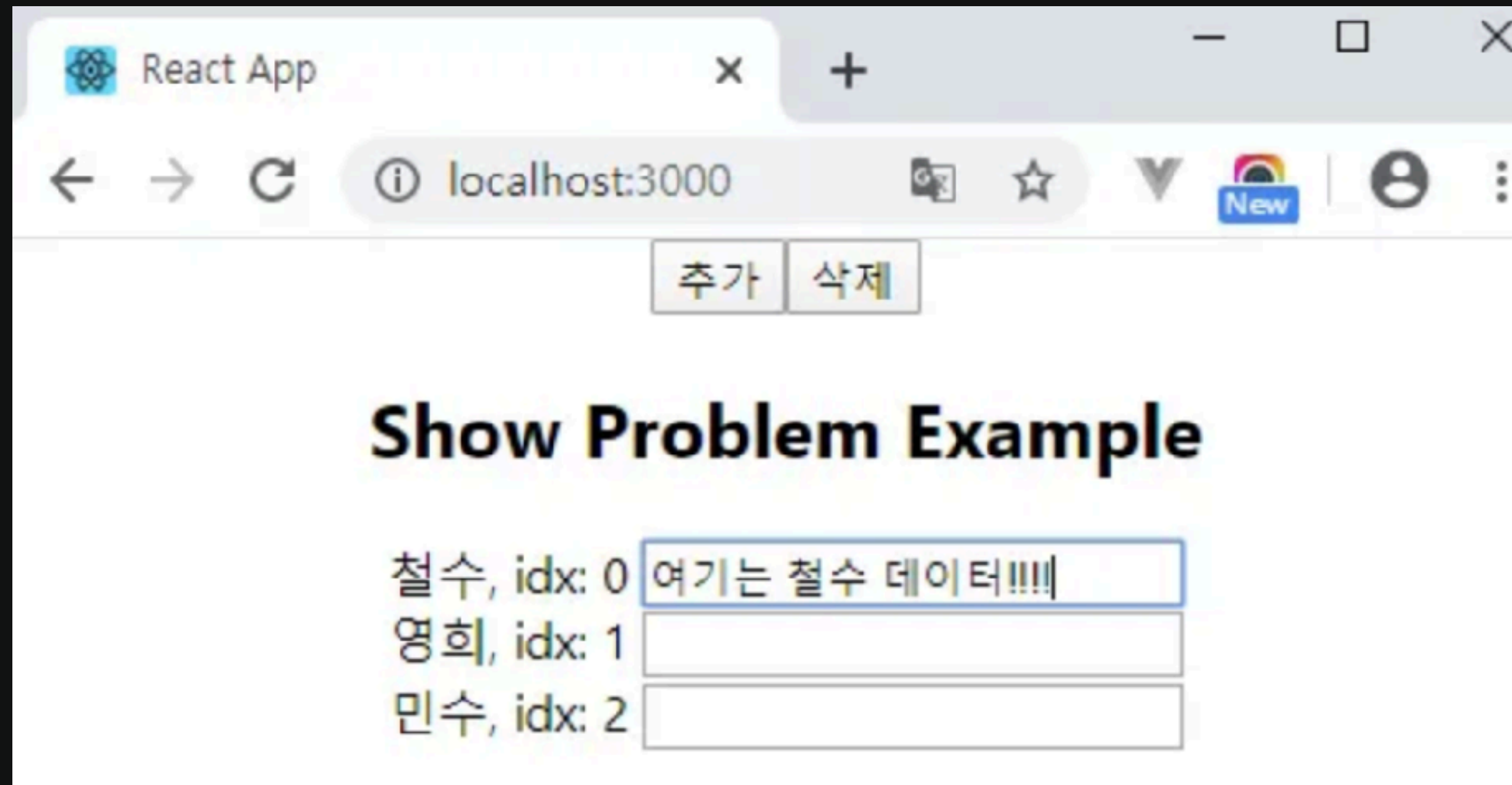
추가 삭제

Show Problem Example

정국, idx: 0	
철수, idx: 1	여기는 철수 데이터!!!!
영희, idx: 2	
민수, idx: 3	

key에 index를 사용하면 안되는 이유!

key가 동일하면, 동일한 DOM element를 렌더링!



렌더링이 발생하는 시점

3) props가 변경되는 경우: 넘겨받는 props가 달라질 때

4) 부모 컴포넌트가 렌더링될 경우

이외에는 리렌더링 되지않음

렌더링의 두 단계

1) 렌더(Render Phase)

: 컴포넌트를 실행(return)한 결과와, 이전 가상 DOM을 비교하여
변경이 필요한 컴포넌트를 체크하는 단계

type, props, key 중 하나라도 변경되었으면 변경이 필요하다고 인식

렌더링의 두 단계

2) 커밋(Commit Phase)

: 렌더 단계에서 체크한 변경 사항을 실제 DOM에 적용하는 단계

만약, 렌더 단계에서 변경사항이 감지되지 않는다면
커밋 단계는 생략 가능하다.