

8. Keys

Keys help React identify which items have been changed, added, or removed.

Keys should be given to the elements inside the array to give a stable identity.

```
1  const usersList = [  
2    {  
3      uniqueNo: 1,  
4      name: "Rahul",  
5    },  
6    {  
7      uniqueNo: 2,  
8      name: "Praneetha",  
9    },  
10   {  
11     uniqueNo: 3,  
12     name: "Varakumar",  
13   },  
14 ]  
15 const listItems = usersList.map((user) => (  
16   <li key={user.uniqueNo}>{user.name}</li>  
17 ))
```

JSX

Collapse ^

8.1 Keys in Lists

When the state of a component changes, React updates the virtual DOM tree. Once the virtual DOM has been updated, React then compares the new virtual DOM with the current virtual DOM.

Once React knows which virtual DOM objects have changed, then React updates only those objects, in the HTML DOM.

React compares the virtual DOMs if there is a change in the node, node type, or the attributes passed to the node. But there is a problematic case by only looking at the node or its attributes. i.e. Lists.

Lists will have the same node types and attributes. Hence the list items can't be uniquely identified if they have been changed, added, or removed.

Example-1:

Countries

Add a country to the end of the list

- India
- USA

```
<!DOCTYPE html>
...<html lang="en"> == $0
  ><head>...</head>
  ▼<body data-new-gr-c-s-check-loaded="14.1024.0" data-gr-ext-installed>
    ▼<div id="root">
      ▼<div class="app-container">
        <h1>Countries</h1>
        <button>Add a country to the end of the list</button>
        ▼<ul>
          ▶<li>...</li>
          ▶<li>...</li>
        </ul>
      </div>
    </div>
```

When a new list item is added to the end of the list,

- React creates a new virtual DOM with the three list items.
- React compares the first two list items in the current virtual DOM and the new virtual DOM.
- Since the two list items in both the previous and current virtual DOMs are matched, React creates only the last list item at the end in the HTML DOM.

Example-2:

Countries

Add a country to the start of the list

- India
- USA

```
<!DOCTYPE html>
...<html lang="en"> == $0
  ><head>...</head>
  ▼<body data-new-gr-c-s-check-loaded="14.1024.0" data-gr-ext-installed>
    ▼<div id="root">
      ▼<div class="app-container">
        <h1>Countries</h1>
        <button>Add a country to the start of the list</button>
        ▼<ul>
          ▶<li>...</li>
          ▶<li>...</li>
        </ul>
      </div>
    </div>
```

When a new list item is added to the start of the list,

- React creates a new virtual DOM with the three list items.
- React compares the first list item in the current virtual DOM and the new virtual DOM.
- Since the first list item in both the previous and current virtual DOMs is different, React treats as the whole list is changed in the current virtual DOM.
- React creates all the three list items in the HTML DOM.

So, React will recreate every element, instead of reusing the two elements that remained the same between the previous and current virtual DOMs. It leads to bad performance.

That's where keys are important. Using keys, every item in a list will have a unique identifier (ID).

Countries

Add a country to the start of the list with keys

- India
- USA

```
<!DOCTYPE html>
...<html lang="en"> == $0
  ><head>...</head>
  ▼<body data-new-gr-c-s-check-loaded="14.1024.0" data-gr-ext-installed>
    ▼<div id="root">
      ▼<div class="app-container">
        <h1>Countries</h1>
        <button>Add a country to the start of the list with keys</button>
        ▼<ul>
          ▶<li>...</li>
          ▶<li>...</li>
        </ul>
      </div>
    </div>
```

So, React can easily detect what needs to be changed or not, re-rendering only the ones with changes.