

Rendering Lists and Using Keys

In this lecture, we will learn how to display lists of data in React. Often, you will need to render multiple items, such as a list of products, users, or tasks. React makes this easy using the JavaScript `map` function.

Rendering a List

Suppose we have an array of fruits. We can render them like this:

```
function App() {
  const fruits = ["Apple", "Banana", "Orange"]

  return (
    <ul>
      {fruits.map(fruit => (
        <li>{fruit}</li>
      ))}
    </ul>
  )
}

export default App
```

Here, we loop through the `fruits` array using `map` and return a `` for each fruit.

The Key Prop

When rendering lists, React requires a special `key` prop. Keys help React identify which items have changed, been added, or removed. Without keys, React may re-render inefficiently.

```
function App() {
  const fruits = ["Apple", "Banana", "Orange"]

  return (
    <ul>
      {fruits.map((fruit, index) => (
        <li key={index}>{fruit}</li>
      )))
    </ul>
  )
}
```

In this example, we use the array index as the key. For real applications, it's better to use a unique ID if available.

Example: Rendering Objects

```
function App() {
  const users = [
    { id: 1, name: "Alice" },
    { id: 2, name: "Bob" },
    { id: 3, name: "Charlie" }
  ]

  return (
    <ul>
      {users.map(user => (
        <li key={user.id}>{user.name}</li>
      )))
    </ul>
  )
}
```

Each user has a unique `id`, which is perfect for the `key` prop.

Recap

- Use the JavaScript `map` function to render lists in React.
- Always add a unique `key` prop to list items.
- Keys help React efficiently update the UI when lists change.