

Chapter 26: How and why to use keys in React

Whenever you are rendering a list of React components, each component needs to have a key attribute. The key can be any value, but it does need to be unique to that list.

When React has to render changes on a list of items, React just iterates over both lists of children at the same time and generates a mutation whenever there's a difference. If there are no keys set for the children, React scans each child. Otherwise, React compares the keys to know which were added or removed from the list

Section 26.1: Basic Example

For a class-less React component:

```
function SomeComponent(props) {  
  
  const ITEMS = ['cat', 'dog', 'rat']  
  function getItemList() {  
    return ITEMS.map(item => <li key={item}>{item}</li>);  
  }  
  
  return (  
    <ul>  
      {getItemList()}  
    </ul>  
  );  
}
```

For this example, the above component resolves to:

```
<ul>  
  <li key='cat'>cat</li>  
  <li key='dog'>dog</li>  
  <li key='rat'>rat</li>  
</ul>
```