

Chapter 27: Keys in react

Keys in react are used to identify a list of DOM elements from the same hierarchy internally.

So if you are iterating over an array to show a list of li elements, each of the li elements needs a unique identifier specified by the key property. This usually can be the id of your database item or the index of the array.

Section 27.1: Using the id of an element

Here we are having a list of todo items that is passed to the props of our component.

Each todo item has a text and id property. Imagine that the id property comes from a backend datastore and is a unique numeric value:

```
todos = [  
  {  
    id: 1,  
    text: 'value 1'  
  },  
  {  
    id: 2,  
    text: 'value 2'  
  },  
  {  
    id: 3,  
    text: 'value 3'  
  },  
  {  
    id: 4,  
    text: 'value 4'  
  },  
];
```

We set the key attribute of each iterated list element to `todo-${todo.id}` so that react can identify it internally:

```
render() {  
  const { todos } = this.props;  
  return (  
    <ul>  
      { todos.map((todo) =>  
        <li key={`todo-${todo.id}`}>  
          { todo.text }  
        </li>  
      ) }  
    </ul>  
  );  
}
```

Section 27.2: Using the array index

If you don't have unique database ids at hand, you could also use the numeric index of your array like this:

```
render() {  
  const { todos } = this.props;  
  return (  
    <ul>  
      { todos.map((todo, index) =>
```

```
    <li key={ `todo-${index}` }>
      { todo.text }
    </li>
  }
</ul>
);
}
```