

列表 & Key

一个基础的列表：

```
function NumberList(props) {
  const numbers = props.numbers;
  const listItems = numbers.map((number) =>
    <li key={number.toString()}>{number}</li>
  );
  return (
    <ul>{listItems}</ul>
  );
}

const numbers = [1, 2, 3, 4, 5];
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<NumberList numbers={numbers} />);
```

Key

****key 帮助 React 识别哪些元素改变了，比如被添加或删除。因此应当给数组中的每一个元素赋予一个确定的标识。**

一个元素的 key 最好是这个元素在列表中拥有的一个独一无二的字符串。

如果没有显示的指定 key 值，React 将默认使用索引来作为列表的 key 值。

如果项目列表的顺序可能会变化，就不建议使用 key 值，因为这样做会导致性能变差，还可能引起组件的状态问题。

在 map() 方法中的元素需要设置 key 属性。

```
function ListItem(props) {
  // 正确！这里不需要指定 key:
  return <li>{props.value}</li>;
}

function NumberList(props) {
  const numbers = props.numbers;
  const listItems = numbers.map((number) =>
    // 正确！key 应该在数组的上下文中被指定
    <ListItem key={number.toString()} value={number} />
  );
  return (
    <ul>
      {listItems}
    </ul>
  );
}
```

```
    );  
  }
```

key 会传递信息给 React，但不会传递给你的组件。如果你的组件中需要使用 key 属性的值，请用其他属性名显式传递这个值：

```
const content = posts.map((post) =>  
  <Post  
    key={post.id}  
    id={post.id}  
    title={post.title} />  
);
```

上面代码中，Post 组件可以读出 props.id，但是不能读出 props.key。

JSX 中嵌套 map()

```
function NumberList(props) {  
  const numbers = props.numbers;  
  const listItems = numbers.map((number) =>  
    <ListItem key={number.toString()}  
              value={number} />  
  );  
  return (  
    <ul>  
      {listItems}  
    </ul>  
  );  
}
```

等价于：

```
function NumberList(props) {  
  const numbers = props.numbers;  
  return (  
    <ul>  
      {numbers.map((number) =>  
        <ListItem key={number.toString()} value={number} />  
      )}  
    </ul>  
  );  
}
```

如果嵌套组件太多层级或比较复杂，就需要提取出来。