

React: Keys, Children, and Inline Style Guide

A Comprehensive Overview for Developers and Interview Preparation

August 27, 2025

Contents

1 Keys in React	2
1.1 Good Practice: Using Unique IDs	2
1.2 Acceptable for Static Lists: Using Index	2
1.3 Why Keys Matter	2
2 Children in React	2
2.1 Example	3
3 Inline Style in React	3
3.1 Example	3

1 Keys in React

Keys are special string attributes used in React to identify which items in a list have changed, been added, or been removed. They help React efficiently update the user interface when working with dynamic lists of elements.

1.1 Good Practice: Using Unique IDs

Using unique IDs as keys ensures React can track list items accurately.

```
const todoItems = todos.map((todo) => (
  <li key={todo.id}>
    {todo.text}
  </li>
));
```

1.2 Acceptable for Static Lists: Using Index

Using the index as a key is acceptable for static lists but should be used with caution, as it may cause issues with dynamic lists.

```
const items = data.map((item, index) => (
  <div key={index}>
    {item.name}
  </div>
));
```

1.3 Why Keys Matter

Without proper keys, React may:

- Re-render all list items, even if only one changed.
- Lose focus in input fields when the list order changes.
- Cause performance issues with large lists.

With proper keys, React can:

- Efficiently update only the changed items.
- Preserve component state correctly.
- Maintain better performance.

2 Children in React

In React, `children` is a special prop automatically passed to every component, representing the content (JSX) placed between the components opening and closing tags. It is useful for creating reusable wrapper components like modals, layouts, and cards.

2.1 Example

```
function Card({ children }) {
  return (
    <div style={{ padding: "20px", border: "1px solid gray", borderRadius: "10px" }}>
      {children} /* Renders content inside <Card> ... </Card> */
    </div>
  );
}

function App() {
  return (
    <div>
      <Card>
        <h2>Hello World</h2>
        <p>This is inside the Card component</p>
      </Card>
    </div>
  );
}
```

Output: The Card component wraps around the content passed inside <Card>...</Card>, making it a flexible wrapper for reusable UI patterns.

3 Inline Style in React

React allows styling elements using the `style` prop, which accepts a JavaScript object (not a CSS string). Property names are camelCase (e.g., `backgroundColor` instead of `background-color`).

3.1 Example

```
function App() {
  return (
    <div>
      <h1 style={{ color: "blue", fontSize: "30px", textAlign: "center" }}>
        Inline Styled Heading
      </h1>
      <button style={{ backgroundColor: "green", color: "white", padding: "10px" }}>
        Click Me
      </button>
    </div>
  );
}
```