



## create a code for to-do list app with react hooks

Here is a simple React to-do list app using hooks. This code example uses `useState` for managing to-do items and handles adding and deleting tasks.

### To-Do List App with React Hooks

```
import React, { useState } from 'react';

function TodoApp() {
  const [todos, setTodos] = useState([]);
  const [input, setInput] = useState('');

  const addTodo = () => {
    if (input.trim() !== "") {
      setTodos([...todos, { text: input, id: Date.now() }]);
      setInput('');
    }
  };

  const removeTodo = (id) => {
    setTodos(todos.filter(todo => todo.id !== id));
  };

  return (
    <div style={{ maxWidth: '400px', margin: 'auto', padding: '20px' }}>
      <h2>To-Do List</h2>
      <input
        type="text"
        value={input}
        onChange={e => setInput(e.target.value)}
        placeholder="Add a new task..."
        style={{ width: '70%', marginRight: '10px' }}
      />
      <button onClick={addTodo}>Add</button>
      <ul style={{ listStyleType: 'none', padding: 0 }}>
        {todos.map(todo => (
          <li key={todo.id} style={{ display: 'flex', justifyContent: 'space-between', padding: '5px 0' }}>
            <span>{todo.text}</span>
            <button onClick={() => removeTodo(todo.id)}>Delete</button>
          </li>
        ))}
      </ul>
    </div>
  );
}
```

```
export default TodoApp;
```

- This app allows users to add new to-dos and remove them from the list with simple buttons.
- The styling is inline for readability and simplicity, but can be customized further.
- To use, save as `TodoApp.js` and import in your main React project file.

This code provides all the basic features for a beginner-friendly React Hook-driven to-do app.