

Sure, let's create an example of toggling between dark and light themes using React context and the `useContext` hook.

Step 1: Create a ThemeContext

First, create a context for managing the theme.

```
// ThemeContext.js
import { createContext, useState } from 'react';

const ThemeContext = createContext();

const ThemeProvider = ({ children }) => {
  const [isDarkMode, setDarkMode] = useState(false);

  const toggleTheme = () => {
    setDarkMode((prevMode) => !prevMode);
  };

  const theme = isDarkMode ? 'dark' : 'light';

  return (
    <ThemeContext.Provider value={{ theme, toggleTheme }}>
      {children}
    </ThemeContext.Provider>
  );
};

export { ThemeProvider, ThemeContext };
```

Step 2: Wrap your App with the ThemeProvider

Wrap your application (or part of it) with the `ThemeProvider`.

```
// App.js
import React from 'react';
import { ThemeProvider } from './ThemeContext';
import ThemedComponent from './ThemedComponent';

const App = () => {
  return (
    <ThemeProvider>
      <div>
        <h1>Themed App</h1>
        <ThemedComponent />
      </div>
    </ThemeProvider>
  );
};

export default App;
```

Step 3: Use useContext in a ThemedComponent

Create a component that uses the `useContext` hook to access the theme information.

```
// ThemedComponent.js
import React, { useContext } from 'react';
import { ThemeContext } from './ThemeContext';

const ThemedComponent = () => {
  const { theme, toggleTheme } = useContext(ThemeContext);

  return (
    <div style={{ background: theme === 'dark' ? '#333' : '#fff', color: theme === 'dark'
? '#fff' : '#333' }}>
      <h2>Themed Component</h2>
      <p>Current Theme: {theme}</p>
      <button onClick={toggleTheme}>Toggle Theme</button>
    </div>
  );
};

export default ThemedComponent;
```

Now, when you click the "Toggle Theme" button, it will switch between dark and light themes, and the `ThemedComponent` will reflect the changes in the background and text colors.

Step 4: Render Components

Finally, render your components in the main file.

```
// index.js
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

Now, you have a simple example of toggling themes using React context and the `useContext` hook.