Congratulations! You passed!

Grade received 100% **To pass** 80% or higher

Go to next item

1. When creating a Provider component, what should you do with the `children` prop that it receives? **1 / 1 point**

- ○ Nothing, the `children` prop is not necessary and can be skipped during the rendering.
- ○ You should clone the `children` inside the component to add the context value to it.
- ◉ You should wrap the JSX that it returns with a Context Provider component and then pass the `children` through.

Correct
That's correct, the children should be passed as a direct child of the Context Provider element.

2. Assuming that the default theme for the application is 'light', what should be the default value passed to the `createContext` call? Select all that apply. **1 / 1 point**

- ☑ undefined

Correct
That's correct, the default value is not relevant and can be any value. It's only useful for testing components in isolation or as a default value when a context consumer does not have a Provider further up in the tree.

- ☑ null

Correct
That's correct, the default value is not relevant and can be any value. It's only useful for testing components in isolation or as a default value when a context consumer does not have a Provider further up in the tree.

- ☑ The string "light"

Correct
That's correct, the default value is not relevant and can be any value. It's only useful for testing components in isolation or as a default value when a context consumer does not have a Provider further up in the tree.

3. One of the parts of the context injected into the application is a function called `toggleTheme`. Assuming that the theme is held in some local state as a string that can be either 'light' or 'dark'. What should be the exact implementation of the `toggleTheme` function? **1 / 1 point**

- ◉ `toggleTheme: () => setTheme(theme === "light" ? "dark" : "light")`
- ○ `toggleTheme: (theme) => setTheme(!theme)`
- ○ `toggleTheme: () => setTheme(theme === "light" ? "light" : "dark")`

Correct
That's correct, that's the right implementation.