



Ashutosh Singh

Adding Dark Mode to a React App with Chakra UI

Ashutosh Singh

Nov 3, 2020 • 6 Min read • 769 Views

Nov 3, 2020 • 6 Min read • 769 Views

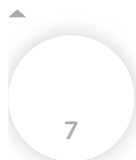
Web Development

Front End Web Dev...

Client-side Framew...

React

Introduction



- [Introduction](#)
- [Installation and Setup](#)
- [Creating Toggle Component](#)
- [Adding Flex and Button Component](#)
- [Adding Dark Mode](#)

Introduction

In this guide, you will learn how to add dark mode to your React app using [Chakra UI](#). Aside from being cool and trendy, dark mode also enhances accessibility for light-sensitive users.

This guide assumes you already know how to install and configure Chakra UI in a React app. You can read [Installing and Using Chakra UI with React](#) to get started.

- [Complete Code](#)
- [Conclusion](#)
- [Top ^](#)

Installation and Setup

In your project's root directory, run the following command.

```
1 npx create-react-app react-dark-mode
2 cd react-dark-mode
```

bash

Install `chakra-ui` by running the following command.

```
1 npm install @chakra-ui/core @emotion/core @emotion/styled emotion-theming
```

bash

Run the following command to start the development server.

```
1 npm start
```

bash

Import and add the `ThemeProvider`, `ColorModeProvider`, and `CSSReset` components in your `App.js`.

Modify your `App.js` like this.

```
import {
  ThemeProvider,
  1    theme,
  2    ColorModeProvider,
  3    CSSReset,
  4  } from "@chakra-ui/core";
  5
  6  function App() {
  7    return (
  8      <ThemeProvider theme={theme}>
  9        <ColorModeProvider>
 10          <CSSReset />
 11        </ColorModeProvider>
 12      </ThemeProvider>
 13    );
 14  }
 15
 16  export default App;
 17
 18
```

Head over to <http://localhost:3000>; your app will be blank now.

Creating Toggle Component

In this section, we will create a new component that will toggle the React app between dark and light modes.

Create a new file named `Toggle.js` in the `src` directory. Run the following commands in your project's root directory to create the file.

```
bash
cd src
touch Toggle.js
1
2
```

Add the following code to `Toggle.js`; this will create an empty functional component in `Toggle.js`.

```
JSX
1  import React from 'react'
2
3  export default function Toggle() {
4    return (
5      <div>
6
7      </div>
8    )
9  }
```

Import and add this `Toggle` component to `App.js`.

```
JSX
1  import {
2    ThemeProvider,
3    theme,
```

```

4      ColorModeProvider,
5      CSSReset,
6    } from "@chakra-ui/core";
7    import Toggle from "../Toggle";
8
9    function App() {
10      return (
11        <ThemeProvider theme={theme}>
12          <ColorModeProvider>
13            <CSSReset />
14            <Toggle />
15          </ColorModeProvider>
16        </ThemeProvider>
17      );
18    }
19
20    export default App;

```

Adding Flex and Button Component

Import `Flex` from `chakra-ui` and use it inside the `Toggle` component.

JSX

```

1    import React from "react";
2    import { Flex } from "@chakra-ui/core";
3    export default function Toggle() {
4      return (
5        <div>

```

```

6      <Flex
7        align="center"
8        justify="center"
9        height="100vh"
10       direction="column"
11     ></Flex>
12   </div>
13 );
14 }

```

You will not see a change, but your app's layout has been changed, which will be apparent after adding more components.

Import the `Button` component from `chakra-ui`.

JSX

```

1    import { Flex, Button } from "@chakra-ui/core";

```

Use this button component inside `Flex`.

JSX

```

1    <Flex align="center" justify="center" height="100vh" direction="column">
2      <Button size="lg">Toggle Mode</Button>
3    </Flex>

```

Here is how your app will look.

Toggle Mode

This button does not do anything yet. In the next section, we will add the logic for changing modes.

Adding Dark Mode

Import `useColorMode` from `chakra-ui`.

```
1 import { Flex, Button, useColorMode } from "@chakra-ui/core";
```

JSX

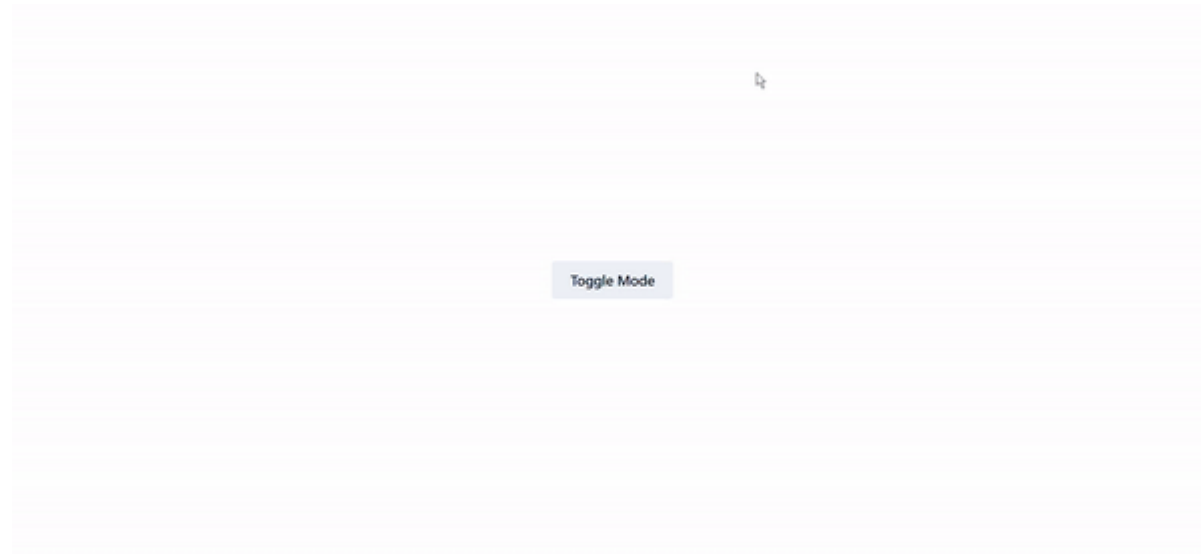
Extract `colorMode` and `toggleColorMode` from `useColorMode` using destructuring.

```
const Toggle = () => {  
  1      const { colorMode, toggleColorMode } = useColorMode();  
  2  
  3      return(  
  4          ...  
  5      )  
  6  };  
  7
```

Add the `toggleColorMode()` function to the button's `onClick` event.

```
1      <Button size="lg" onClick={() => toggleColorMode()}>  
2          Toggle Mode  
3      </Button>
```

And it's done; you have created and added a dark mode toggle button to your React app.



You can use `colorMode` to display the current mode. For example, instead of `Toggle Mode` as the button text, you can use `{colorMode}`, and it will show you the current mode.

JSX

```
1   <Button size="lg" onClick={() => toggleColorMode()}>
2     {colorMode}
3   </Button>
```

Here is how this will look.



Complete Code

Here is the complete code for the example. You can also see a [live example](#) and code on [GitHub](#).

JSX

```
1   import React from "react";
2   import { Button, Flex, useColorMode } from "@chakra-ui/core";
3
4   const Toggle = () => {
5     const { colorMode, toggleColorMode } = useColorMode();
6
7     return (
8       <div>
9         <Flex align="center" justify="center" height="100vh" direction="col">
10           <Button size="lg" onClick={() => toggleColorMode()}>
11             Toggle Mode {colorMode}
```

```
12         </Button>
13     </Flex>
14 </div>
15 );
16 };
17
18 export default Toggle;
```

Conclusion

In this guide, we discussed the step-by-step process of adding a dark mode toggle button to a React app. You can take this toggle button a step further by using icons for light and dark mode to change the theme.

Here are some additional resources that can be helpful.

- [Chakra UI Docs](#)
- [Chakra UI GitHub](#)

Happy coding!

