

Theming React With CSS variables

How to theme a React application using CSS custom properties and hooks



Malcolm Laing

Jul 1 · 3 min read ★



Get one more story in your member preview when you sign up. It's free.



Sign up with Google



Sign up with Facebook

Already have an account? [Sign in](#)

• • •

Getting Started

As always, let's start with `create-react-app`. We can bootstrap our new application as follows:

```
npx create-react-app css-theming
```

For this demo, we won't need to install any other dependencies. `React`, `react-dom`, and `react-scripts` will be enough.

Open the new application in your favourite editor.

Let's start by declaring some CSS variables in `App.css`.

```
:root {  
  --color-surface: white;  
  --color-solid: black;  
  --color-primary: teal;  
}
```

Next, we create some basic styles for our app. Notice how we consume our variables with the syntax `var(--variable-name)`. This should be familiar to you if you are used to using PostCSS custom properties.

```
1  html {  
2    background: var(--color-surface);  
3    color: var(--color-solid);
```



Get one more story in your member preview when you sign up. It's free.



Sign up with Google



Sign up with Facebook

Already have an account? [Sign in](#)

Finally, let's modify our `App.js` to include our button.

```
1  const App = () => {  
2    return (  
3      <div>  
4        <h1>Welcome to our themed Application!</h1>  
5        <button>Toggle theme</button>  
6      </div>  
7    );  
8  };
```

app.js hosted with ❤ by GitHub

[view raw](#)

We are ready to use JavaScript to modify our custom properties.

. . .

Modifying Custom Properties With JavaScript

The syntax for modifying custom properties is quite simple.

Getting a variable's value

```
window  
  .getComputedStyle(document.documentElement)  
  .getPropertyValue("--color-surface");
```

Setting a variable's value

```
document  
  .documentElement.style.setProperty("--color-surface", "blue");
```



Get one more story in your member preview when you sign up. It's free.



Sign up with Google



Sign up with Facebook

Already have an account? [Sign in](#)

```
2  "--color-solid": "black",
3  "--color-surface": "white",
4  "--color-primary": "teal"
5  };
6  const darkTheme = {
7    "--color-solid": "white",
8    "--color-surface": "black",
9    "--color-primary": "purple"
10 };
```

theme.js hosted with ❤ by GitHub

[view raw](#)

Next we modify our app's render function to toggle its theme, using `useState`. We keep track of our current theme, and we update it in the `onClick` handler.

```
1  const App = () => {
2    const [currentTheme, setTheme] = React.useState("light");
3    const onClick = () => {
4      const nextTheme = currentTheme === "light" ? "dark" : "light";
5      setTheme(nextTheme);
6    };
7    return (
8      <div className="App">
9        <h1>{currentTheme === "light" ? "Light theme" : "Dark theme"}</h1>
10       <button onClick={onClick}>Toggle theme</button>
11     </div>
12   );
13 };
```

app.js hosted with ❤ by GitHub

[view raw](#)

Then we create a function which updates our CSS variables.

```
1  const applyTheme = nextTheme => {
2    const theme = nextTheme === "dark" ? lightTheme : darkTheme;
```



Get one more story in your member preview when you sign up. It's free.



Sign up with Google



Sign up with Facebook

Already have an account? [Sign in](#)

Finally, we add a call to our `applyTheme` function in our button's `onClick` handler.

```
const onClick = () => {  
  const nextTheme = currentTheme === "light" ? "dark" : "light";  
  setTheme(nextTheme);  
  applyTheme(nextTheme);  
};
```

And there you have it — We've created a theme which can be configured using JavaScript. Let's see it in action.

Light theme



I may not win any awards for the design, but at least it works as expected.

. . .

Adding Tests



Get one more story in your member preview when you sign up. It's free.



Sign up with Google



Sign up with Facebook

Already have an account? [Sign in](#)

```
1    expect(getByText('Light theme')).toBeInTheDocument(),
8    });
9  });
10  describe("when toggling the theme", () => {
11    it("should update theme", () => {
12      const { getByText } = render(<App />);
13      const button = getByText("Toggle theme");
14      button.click();
15      expect(getByText("Dark theme")).toBeInTheDocument();
16    });
17  });
```

testing.library.js hosted with ❤ by GitHub

[view raw](#)

Here, we test that when the app starts, it defaults to the light theme, and displays the text “Light theme” in the document.

Next, we click on our button to toggle the theme, and test to make sure that the text in our app updates.

While these tests are good, and provide some value, they don’t do anything to make sure that our CSS variables have changed.

To test this, let’s mock one of the methods that we used to set the CSS variables.

We take the function `document.documentElement.style.setProperty` and replace it with a `jest.fn()` mock.

```
1  it("should update the css variables", () => {
2    const setProperty = jest.fn();
3    document.documentElement.style.setProperty = setProperty;
4    const { getByText } = render(<App />);
5    const button = getByText("Toggle theme");
6    button.click();
7    expect(setProperty).toHaveBeenCalledTimes(3);
```



Get one more story in your member preview when you sign up. It's free.



Sign up with Google



Sign up with Facebook

Already have an account? [Sign in](#)

• • •

Conclusion

We were able to set up theming in minutes, without adding any extra dependencies, or changing our build setup.

We can use these variables all over our CSS, and our React components don't need to know about them.

And finally, we were able to set up some tests that will give us confidence in our app's functionality.

Check out the code on code sandbox to see it in action.

css-theming - CodeSandbox

This project was bootstrapped with [Create React App](https://github.com/facebook/create-react-...
codesandbox.io

Thanks for reading!

[React](#) [JavaScript](#) [CSS](#) [Design](#) [Programming](#)

[About](#) [Help](#) [Legal](#)

Get one more story in your member preview when you sign up. It's free.



Sign up with Google



Sign up with Facebook

Already have an account? [Sign in](#)