

Emotion Package

One of the better examples of implementing styled components in React is through the use of the [emotion](#) package. It is a node package that allows you to generate encapsulated components and even bundle components together. The basic syntax (as per the docs) is as follows:

```
import styled from "@emotion/styled";

export const Button = styled.button`
  color: turquoise;
`;
```

This looks a little weird at first, but ultimately we're basically just saying that whenever we use the `<Button>` `</Button>` element, it's actually going to render a normal html `<button></button>` element with text color turquoise. Ok so that's not really that crazy, but what about this as a more complex example:

```
import styled from "@emotion/styled";

const Button = styled.button`
  color: ${(props) => (props.primary ? "hotpink" : "turquoise")};
`;

const Container = styled.div((props) => ({
  display: "flex",
  flexDirection: props.column && "column",
})));

render(
  <Container column>
    <Button>This is a regular button.</Button>
    <Button primary>This is a primary button.</Button>
  </Container>
);
```

It might look weirder at first, but let's break it down:

1. The Button is basically taking in any props it is passed and saying "Oh is there a primary prop? Ok well then make the color hotpink, OTHERWISE make it turquoise". This can be expanded upon and you can even change the border, size, background-color or anything else with just that ONE prop
2. The container is going to be a div that is flex and direction column IF you pass the column prop. That's what the `props.column && 'column'` is doing. If we didn't pass the `column` prop it would ignore that property and it would go to the default (row for this instance)
3. The render shows us being able to use it all together.

Let's take a minute to look at some more examples on the docs linked to above.