Styling Your Applications

The css Prop

In Emotion, the `css` prop is used to style elements. It can be used with object styles and tagged template literals.

```
// object styles
<div
  css={{
   backgroundColor: 'rebeccapurple',
  }}
>
  Content to Display
</div>


// tagged template literals
<div
  css={css`
   background-color: rebeccapurple;
  `}
>
  Content to Display
</div>
```

React Styled Components

Styled components are React components that have styles attached to them.

```
const ContentWrapper = styled.div`
  width: 100vw;
  height: 100vh;
  display: grid;
`

function App() {
  return (
    <ContentWrapper>Content</ContentWrapper>
  )
}
```

Composition

Compositions can be used to group styles together and be used in other style blocks.

```
const Button = styled.button`
  color: black;
  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid black;
  border-radius: 3px;
`;

// A new component based on Button, but
with some override styles
const CornflowerButton = styled(Button)`
  color: cornflowerblue;
  border-color: cornflowerblue;
`;
```

Theming

In Emotion, themes are used to specify styles that are used throughout the whole application. Styles defined in the theme can be used as variables in other style blocks.

```
import { ThemeProvider } from
'@emotion/react'
import styled from '@emotion/styled'

const theme = {
  colors: {
    primary: 'tomato'
  }
}

const AlertText = styled.div`
  color: ${props =>
props.theme.colors.primary};
`

render(
  <ThemeProvider theme={theme}>
    <AlertText>alert text</AlertText>
  </ThemeProvider>
)
```

keyframes Helper

In Emotion, the `keyframes` helper can be used to define animations and returns an object that can be used in other style blocks.

```
import { jsx, css, keyframes } from
'@emotion/react'

const partyText = keyframes`
  0% {
      background-color: red;
  }
  20% {
      background-color :yellow;
  }
  40% {
      background-color: green;
  }
  60% {
      background-color: blue;
  }
  80% {
      background-color: red;
  }
`

render(
  <div
    css={css`
      animation: ${partyText} 1s ease
infinite;
    `}
  >
    party text!
  </div>
)
```