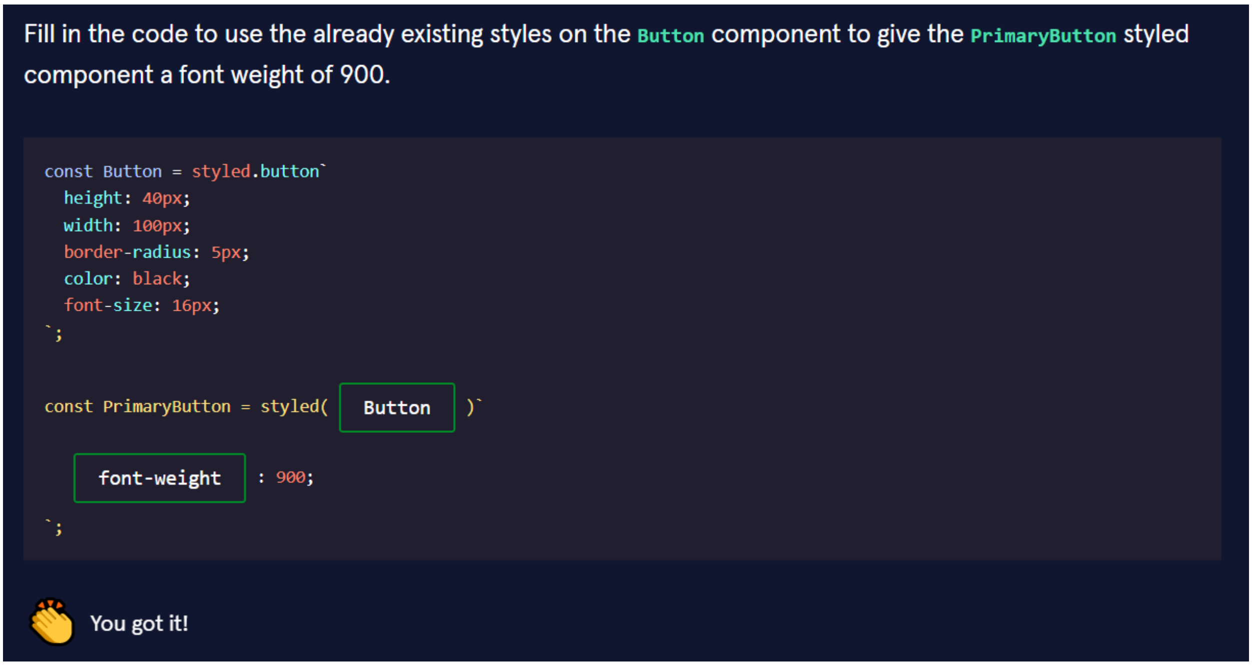
QUIZ



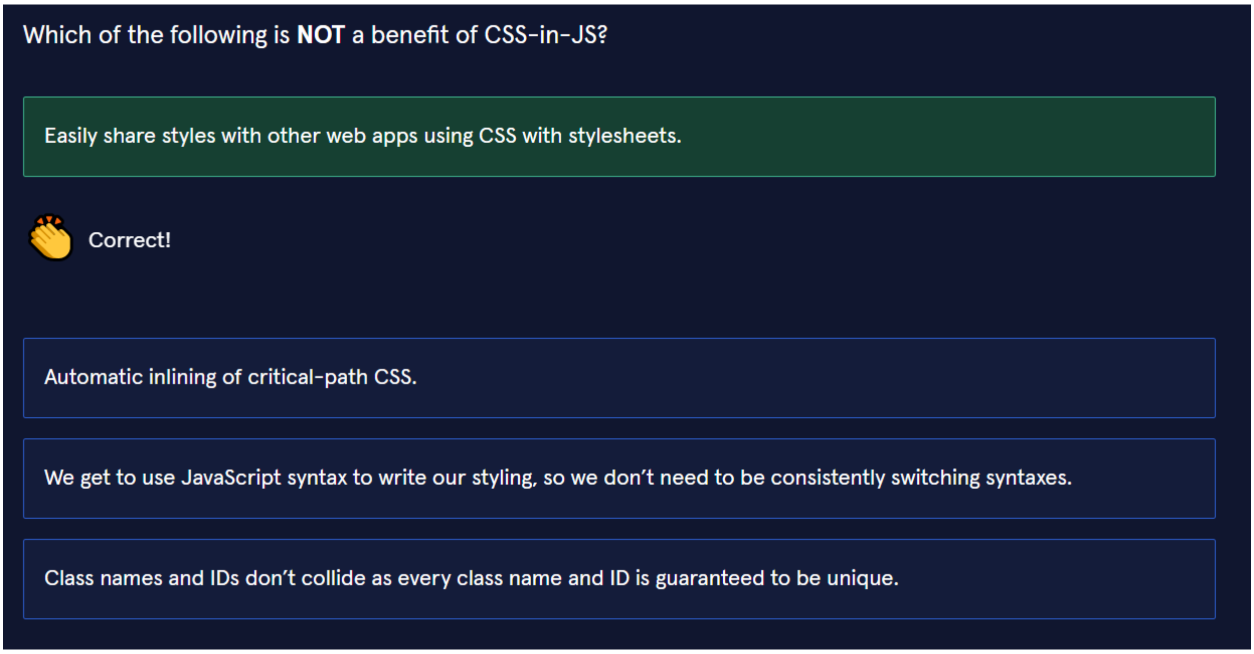
Fill in the code to use the already existing styles on the `Button` component to give the `PrimaryButton` styled component a font weight of 900.

```
const Button = styled.button`
  height: 40px;
  width: 100px;
  border-radius: 5px;
  color: black;
  font-size: 16px;
`;

const PrimaryButton = styled( Button )`

    font-weight : 900;
`;
```

👏 You got it!



Which of the following is **NOT** a benefit of CSS-in-JS?

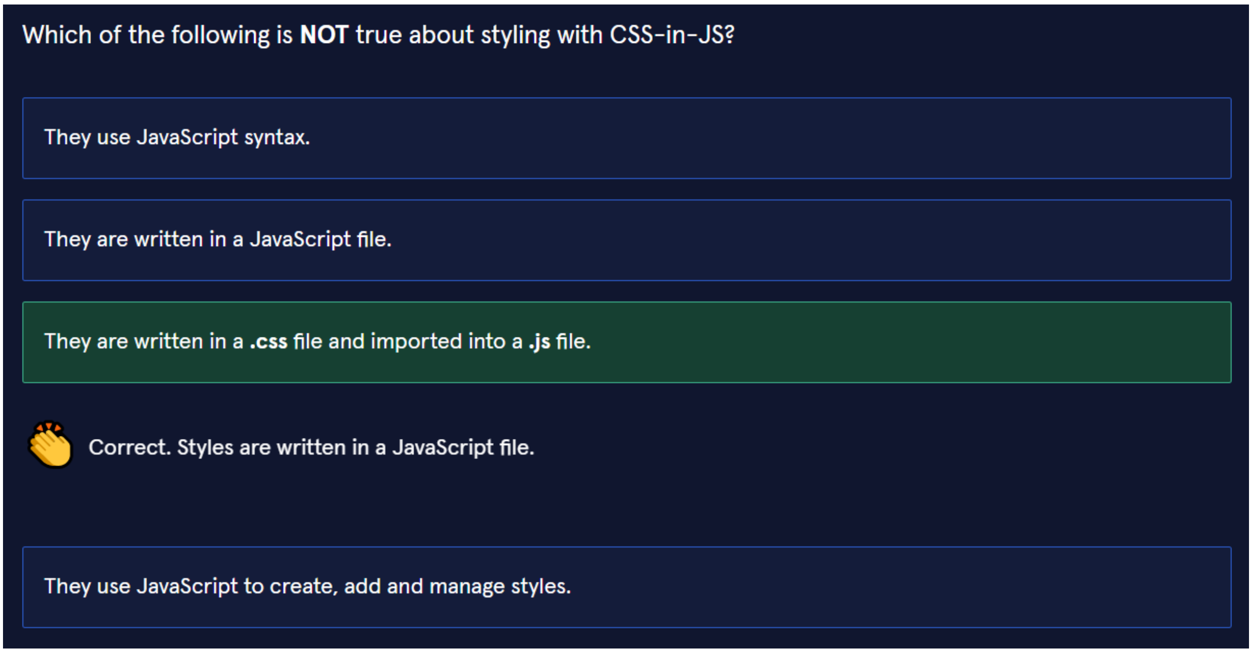
Easily share styles with other web apps using CSS with stylesheets.

👏 Correct!

Automatic inlining of critical-path CSS.

We get to use JavaScript syntax to write our styling, so we don't need to be consistently switching syntaxes.

Class names and IDs don't collide as every class name and ID is guaranteed to be unique.



Which of the following is **NOT** true about styling with CSS-in-JS?

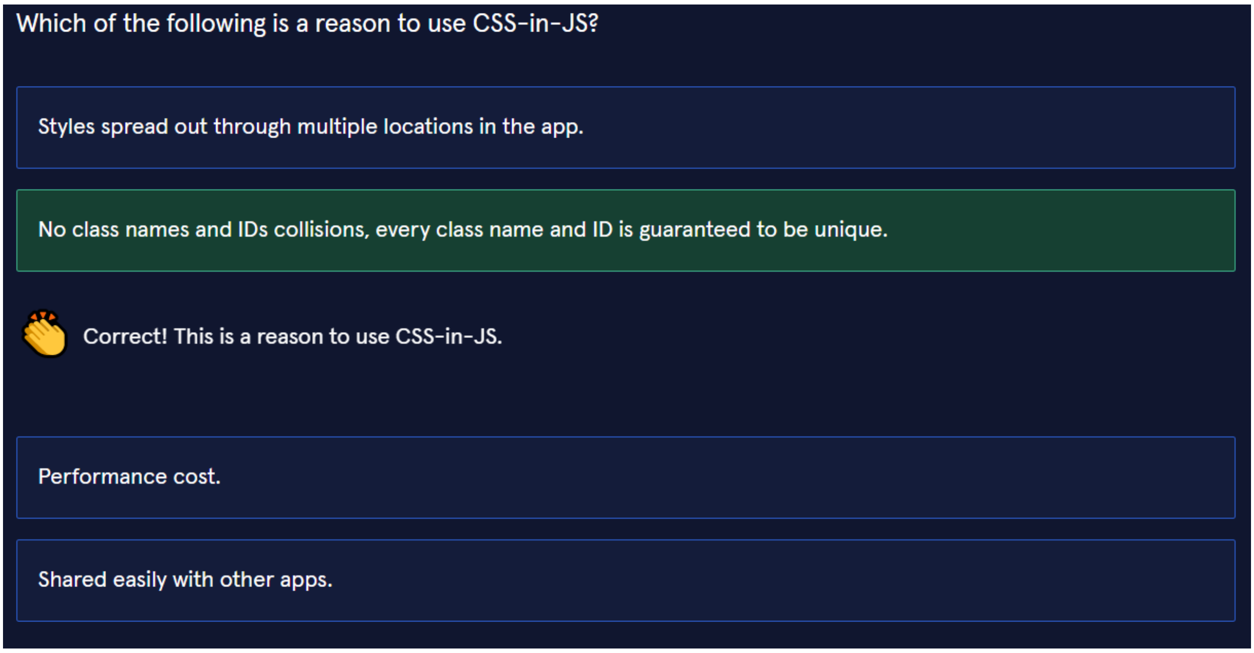
They use JavaScript syntax.

They are written in a JavaScript file.

[✓ They are written in a **.css** file and imported into a **.js** file.]

👏 Correct. Styles are written in a JavaScript file.

They use JavaScript to create, add and manage styles.



Which of the following is a reason to use CSS-in-JS?

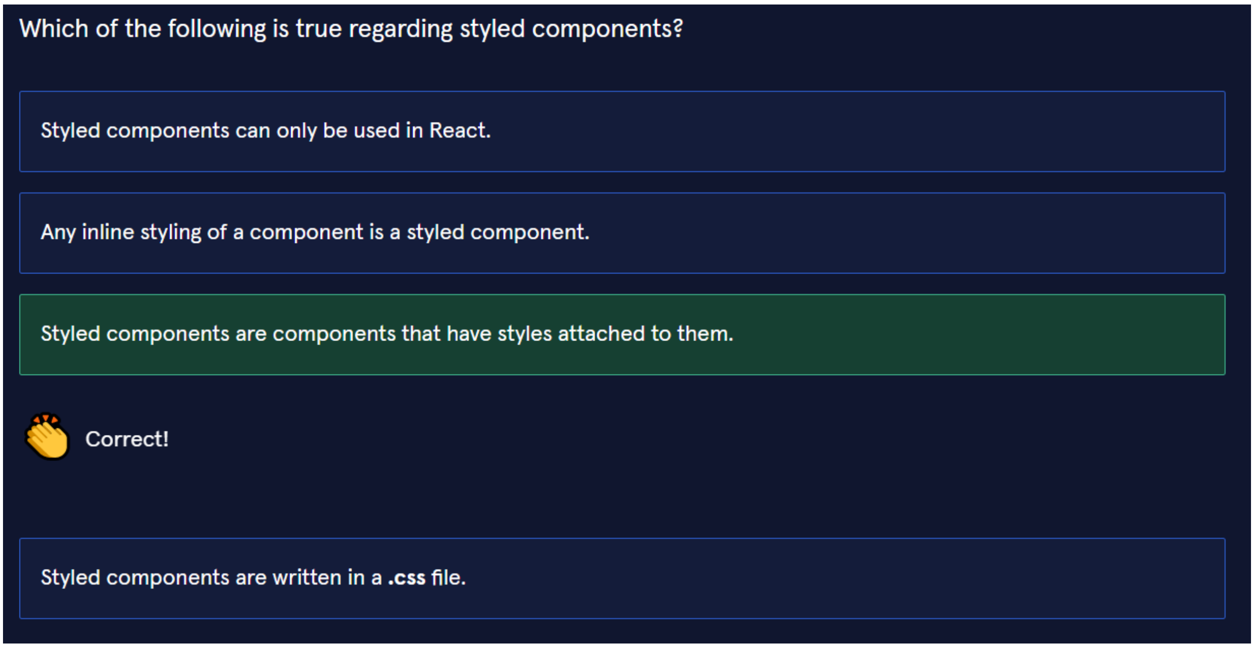
Styles spread out through multiple locations in the app.

[✓ No class names and IDs collisions, every class name and ID is guaranteed to be unique.]

👏 Correct! This is a reason to use CSS-in-JS.

Performance cost.

Shared easily with other apps.



Which of the following is true regarding styled components?

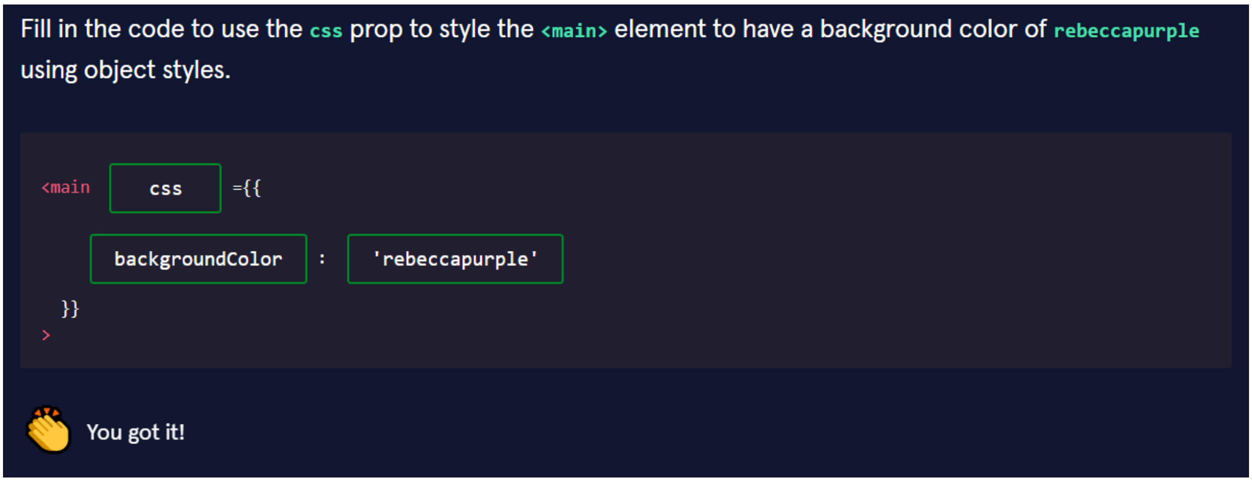
Styled components can only be used in React.

Any inline styling of a component is a styled component.

Styled components are components that have styles attached to them.

👏 Correct!

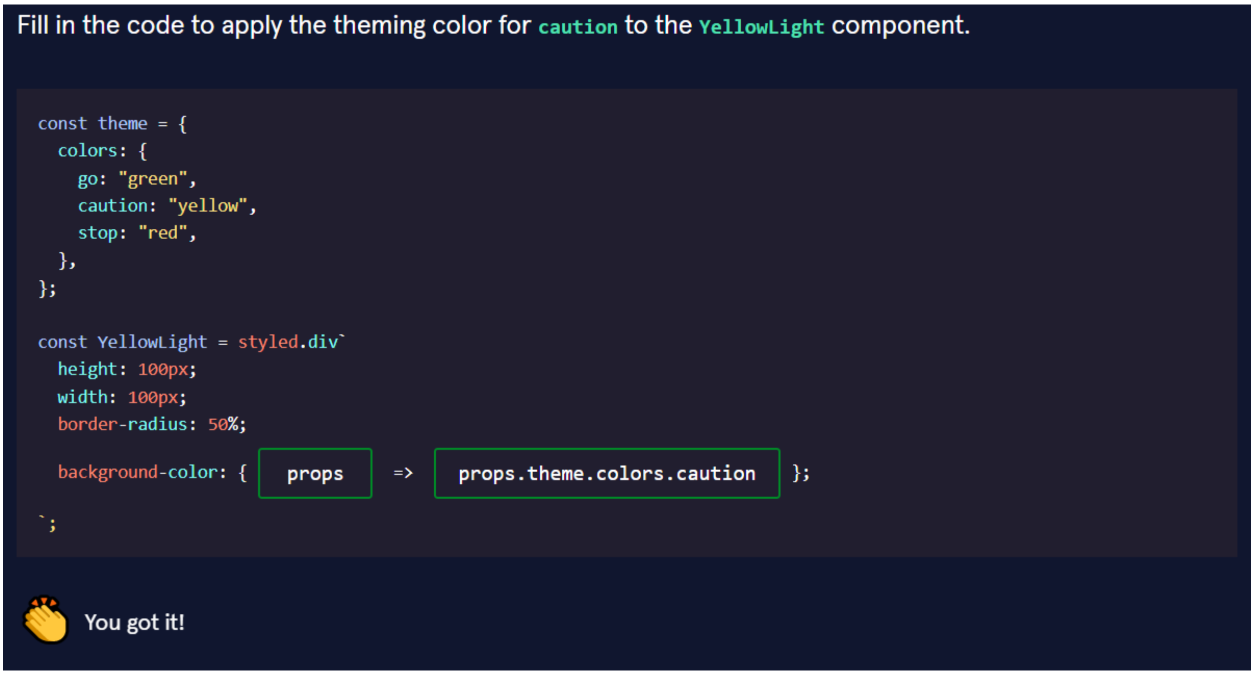
Styled components are written in a **.css** file.



Fill in the code to use the `css` prop to style the `<main>` element to have a background color of `rebeccapurple` using object styles.

```
<main css ={{
    backgroundColor : 'rebeccapurple'
  }}
>
```

👏 You got it!



Fill in the code to apply the theming color for `caution` to the `YellowLight` component.

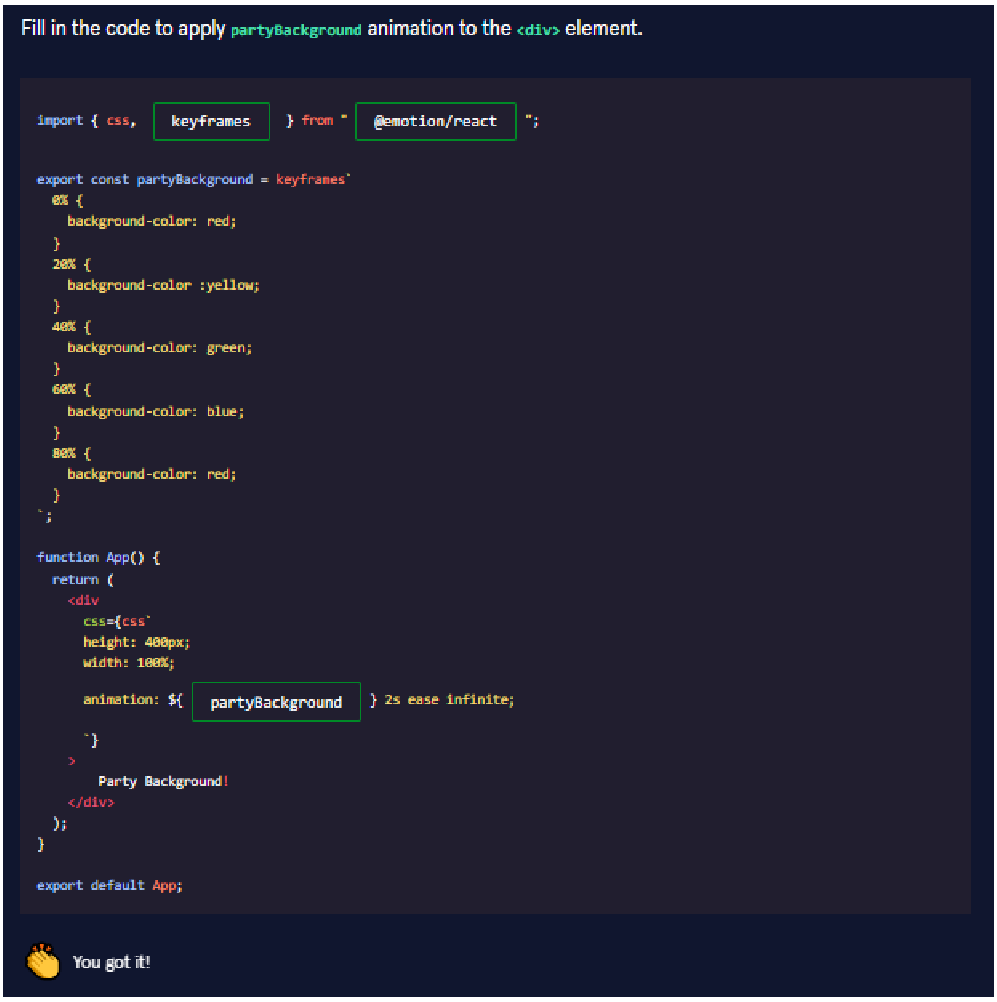
```
const theme = {
  colors: {
    go: "green",
    caution: "yellow",
    stop: "red",
  },
};

const YellowLight = styled.div`
  height: 100px;
  width: 100px;
  border-radius: 50%;

  background-color: { props  =>  props.theme.colors.caution  };

`;
```

You got it!



Fill in the code to apply `partyBackground` animation to the `<div>` element.

```
import { css,  keyframes  } from "  @emotion/react  ";

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

function App() {
  return (
    <div
      css={css`
      height: 400px;
      width: 100%;

      animation: ${  partyBackground  } 2s ease infinite;

      `}
    >
        Party Background!
    </div>
  );
}

export default App;
```

You got it!