# Style in React

**Estimated time needed:** 3 minutes

Styling in React can be done in various ways, including inline styles, CSS modules, styled components, and more. Let's explore each approach briefly.

## Objective

After completing this reading, you will be able to:

- Explore various styles in React

## Inline styles

```
import React from 'react';
function MyComponent() {
  return (
    <div style={{ backgroundColor: 'lightblue', padding: '20px', borderRadius: '5px' }}>
      <p style={{ color: 'white', fontSize: '18px' }}>This is a paragraph with inline styles.</p>
    </div>
  );
}
export default MyComponent;
```

- The `<div>`element has inline styles applied directly using the style attribute.
- The styles are specified as a JavaScript object where the keys are CSS property names, and the values are the corresponding CSS property values
- Similarly, the `<p>` element inside `<div>` has inline styles applied using the style attribute.
- The above component will render a `<div>` with a light blue background, padding, and border-radius. Inside the `<div>`, there's a `<p>` element with white text color and a font size of 18 pixels, all styled using inline styles.

## CSS modules

`toggleMessage.module.css:`

```
.message {
  display: block;
  color: green;
  font-size: 18px;
  margin-top: 10px;
}
ToggleMessage.js:
import React, { useState } from 'react';
import styles from './toggleMessage.module.css';
function ToggleMessage() {
  const [isVisible, setIsVisible] = useState(true);
  const toggleVisibility = () => {
    setIsVisible(!isVisible);
  };
  return (
    <div>
      <h2>Toggle Message</h2>
      <button onClick={toggleVisibility}>
        {isVisible ? 'Hide Message' : 'Show Message'}
      </button>
      <p className={isVisible ? styles.message : ''}>This is a hidden message.</p>
    </div>
  );
}
export default ToggleMessage;
```

- Import styles from `./toggleMessage.module.css;`: This imports the CSS module `toggleMessage.module.css` as an object named styles. In CSS, a module typically refers to a file (usually with `a.module.css` extension) that contains a set of CSS rules scoped to a particular component or module in your application. These CSS rules are locally scoped to prevent unintended style conflicts between components.
- `const [isVisible, setIsVisible] = useState(true);`: This initializes a state variable `isVisible` using the `useState hook`. It represents whether the message paragraph should be visible or not, and it defaults to true.
- `const toggleVisibility = () => {...};`: This is a function that toggles the visibility of the message paragraph by updating the state variable `isVisible` when the button is clicked.
- `<p className={isVisible ? styles.message : ''}>`This is a hidden message.`</p>`: This paragraph element displays the message. Its class name is conditionally set based on the value of `isVisible`. If `isVisible` is true, it applies the message class from the CSS module, otherwise, it applies an empty string, effectively removing any additional styles.

## Styled components:

CSS can also be applied in React components using JavaScript objects, similar to the messageStyle object in the provided React component. For example, in the following code there is one object with name `messageStyle { color: 'green', fontSize: '18px' }` represents CSS properties for color and font size. It also has a conditional styling. For example, `{ display: isVisible ? 'block' : 'none' }` that dynamically sets the display property based on the value of `isVisible`.

```
import React, { useState } from 'react';
function ToggleMessage() {
  const [isVisible, setIsVisible] = useState(true);
  const toggleVisibility = () => {
    setIsVisible(!isVisible);
  };
  const messageStyle = {
    display: isVisible ? 'block' : 'none',
    color: 'green',
    fontSize: '18px',
    marginTop: '10px'
  };
  return (
    <div>
      <h2>Toggle Message</h2>
      <button onClick={toggleVisibility}>
        {isVisible ? 'Hide Message' : 'Show Message'}
      </button>
      <p style={messageStyle}>This is a hidden message.</p>
    </div>
  );
}
```

## Author(s)

Richa Arora

**© IBM Corporation. All rights reserved.**