

Styling React Using CSS

There are many ways to style React with CSS, this tutorial will take a closer look at three common ways:

- Inline styling
- CSS stylesheets
- CSS Modules

Inline Styling

To style an element with the inline style attribute, the value must be a JavaScript object:

Example:

Insert an object with the styling information:

```
const Header = () => {  
  return (  
    <>  
      <h1 style={{color: "red"}}>Hello Style!</h1>  
      <p>Add a little style!</p>  
    </>  
  );  
}
```

Note: In JSX, JavaScript expressions are written inside curly braces, and since JavaScript objects also use curly braces, the styling in the example above is written inside two sets of curly braces `{{}}`.

camelCased Property Names

Since the inline CSS is written in a JavaScript object, properties with hyphen separators, like `background-color`, must be written with camel case syntax:

Example:

Use `backgroundColor` instead of `background-color`:

```
const Header = () => {  
  return (  
    <>  
    <h1 style={{backgroundColor: "lightblue"}}>Hello Style!</h1>  
    <p>Add a little style!</p>  
    </>  
  );  
}
```

JavaScript Object

You can also create an object with styling information, and refer to it in the style attribute:

Example:

Create a style object named `myStyle`:

```
const Header = () => {  
  const myStyle = {  
    color: "white",  
    backgroundColor: "DodgerBlue",
```

```
padding: "10px",  
fontFamily: "Sans-Serif"  
};  
return (  
  <>  
    <h1 style={myStyle}>Hello Style!</h1>  
    <p>Add a little style!</p>  
  </>  
);  
}
```

CSS Stylesheet

You can write your CSS styling in a separate file, just save the file with the `.css` file extension, and import it in your application.

App.css:

Create a new file called "App.css" and insert some CSS code in it:

```
body {  
  background-color: #282c34;  
  color: white;  
  padding: 40px;  
  font-family: Sans-Serif;  
  text-align: center;  
}
```

Import the stylesheet in your application:

Header.js:

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './App.css';

const Header = () => {
  return (
    <>
      <h1>Hello Style!</h1>
      <p>Add a little style!</p>
    </>
  );
}
```

CSS Modules

Another way of adding styles to your application is to use CSS Modules.

CSS Modules are convenient for components that are placed in separate files.

The CSS inside a module is available only for the component that imported it, and you do not have to worry about name conflicts.

Create the CSS module with the `.module.css` extension, example: `my-style.module.css`.

Create a new file called "my-style.module.css" and insert some CSS code in it:

my-style.module.css:

```
.bigblue {  
  color: DodgerBlue;  
  padding: 40px;  
  font-family: Sans-Serif;  
  text-align: center;  
}
```

Import the stylesheet in your component:

Car.js:

```
import styles from './my-style.module.css';  
  
const Car = () => {  
  return <h1 className={styles.bigblue}>Hello Car!</h1>;  
}  
  
export default Car;
```