# Different Ways to add CSS in React JS

[#react](https://dev.to/t/react)[#javascript](https://dev.to/t/javascript)[#webdev](https://dev.to/t/webdev)[#beginners](https://dev.to/t/beginners)

Hi guys!.

In this post, we will be looking into the best ways CSS code can be added in React JS or to you React App.  
Obviously, CSS is crucial in making your app user friendly and visually attractive. These are the different ways to add CSS to your react app:

Please check out my new [tutorial](https://discover.hubpages.com/technology/reactCounter).

## 1 - External Stylesheet

You can create a new CSS file in your project directory and add your CSS inside it. You can then import it in your component, class or React JS page.  
The following code is used to import an external CSS stylesheet.

import "./styles.css";

## 2 - Inline CSS

Probably the most common and quickest out of all 3 is inline CSS. However it has many disadvantages and it is generally discouraged to use unless it is a very small application.  
Basically, we create an object that contains different references which are then called using the style{} attribute.  
For example, the CSS is added like this:

const styles = {

section: {

fontSize: "18px",

color: "#292b2c",

backgroundColor: "#fff",

padding: "0 20px"

},

wrapper: {

textAlign: "center",

margin: "0 auto",

marginTop: "50px"

}}

It is then added to an element like this:

<section style={styles.section}>

<div style={styles.wrapper}>

</div></section>

## 3 - Styled Components

Probably the most powerful and useful in my opinion is Styled Components. Styled Components lets you write actual CSS in your JavaScript. The main advantage is that you can add conditional code and use variables and functions within the CSS!.

You can install Styled Components using the following command:

npm install --save styled-components

Next, you need to import it in you component. Then you can create a new variable that will contain the CSS. The same variable name with open and close brackets will render or create an HTML element with the previously added styles on it.

import styled from 'styled-components'// Create a button variable and add CSS

const Button = styled.button`

background: transparent;

border-radius: 3px;

border: 2px solid red;

color:red;

`//display the HTMLrender(

<div>

<Button>Button</Button> </div>);

### **Using a CSS-in-JS library**

There are also several libraries that allow you to write your styles in JavaScript and apply them to your components.

This can be convenient if you want to dynamically generate styles or take advantage of features like theming. Some popular CSS-in-JS libraries include styled-components and emotion.

import React from 'react';

import styled from 'styled-components';

const RedText = styled.div`

color: red;

font-size: 32px;`;

function MyComponent() {

return (

<RedText>

This text is red and 32px

</RedText>

);}

This is a functional React component that uses the "styled-components" library to style its JSX elements. The styled-components library allows you to write actual CSS code to style your components, in a way that is scoped to the component. It also allows you to use JavaScript expressions in your CSS.

Here, the component imports the styled object from the styled-components library and creates a new component called RedText that renders a div element with the styles defined inside the backticks. The styles defined inside the backticks include setting the text color to red and font-size to 32px.

Then the component returns JSX that uses the RedText component, which renders a div with the defined styles.

When this component is rendered by a React application, it will display a red text with font size 32px.

This method of styling allows you to create reusable, self-contained components, with their own styles, that are easy to manage and test.

##### Other than these, there are 3 more useful ways you can add CSS ([credit to lukeshiru](https://dev.to/lukeshiru))

## 4 - CSS Modules

You can also add scoped styles quite easily, you just need to create a file with the extension .module.css, like this:

// ComponentName.module.css

.Red {

color: #f00;}

.Blue {

color: #00f;}

Then you import it like this:

import styles from "./ComponentName.module.css";

and use it like this:

<span className={styles.Red}>This text will be red</span><span className={styles.Blue}>This text will be blue</span>

## 5 - Atomic CSS

One of the most popular atomic CSS frameworks out there is [Tailwind](https://tailwindcss.com/), by just making it part of your project following their [instructions](https://tailwindcss.com/docs/guides/create-react-app) you can just use classNames without even touching CSS.

<button className="font-bold bg-blue-600 px-6 py-3 text-white rounded-md">Blue button</button>

## 6 - Emotion

Styled-components is not the only library that allows you to create components with styles embeded on them, you have great alternatives out there such as Emotion. The best thing about Emotion is that it's framework agnostic, so you can take your knowledge to other libraries and frameworks besides React, while being pretty similar to styled-components (so you can in several scenarios just change the import).

Note :

<https://dev.to/salehmubashar/3-ways-to-add-css-in-react-js-336f#:~:text=Different%20Ways%20to%20add%20CSS%20in%20React%20JS,Atomic%20CSS%20...%206%206%20-%20Emotion%20>

Note : React small task

[How to Create a Counter in React JS - Owlcation](https://owlcation.com/stem/reactCounter)

<https://owlcation.com/stem/reactCounter>