1. CSS Stylesheets (external):

This is the most traditional way of styling components, where you define styles in separate CSS files and then import those styles into your components.

Eg:

MyComponent.js

import React from 'react';

import './styles.css';

const MyComponent = () => {

return (

<div>

<h1 className="my-heading">This is an H1 Heading</h1>

</div>

);

};

export default MyComponent;

styles.css

.my-heading {

color: blue;

font-size: 24px;

}

2. Inline CSS: In inline styling basically we create objects of style. And render it inside the components in style attribute using the React technique to incorporate JavaScript variable inside the JSX (Using ‘{ }’ )

Eg :

import React from 'react';

const MyComponent = () => {

const headingStyles = {

color: 'blue',

fontSize: '24px',

};

return (

<div>

<h1 style={headingStyles}>Hello</h1>

</div>

);

};

export default MyComponent;

3.Styled Components: The styled-components allows us to style the CSS under the variable created in JavaScript. style components is a third party package using which we can create a component as a JavaScript variable that is already styled with CSS code and used that styled component in our main component. styled-components allow us to create custom reusable components which can be less of a hassle to maintain.

Command to install styled-components package

npm install --save styled-components

Eg:

import React from 'react';

import styled from 'styled-components';

const StyledButton = styled.button`

background-color: #007bff;

color: #fff;

padding: 10px 20px;

border: none;

cursor: pointer;

`;

const Button = () => {

return <StyledButton>Click me</StyledButton>;

};

4. CSS Modules:

CSS Modules allow you to scope CSS locally to a component. The styles are automatically scoped to the component, preventing naming conflicts.

Eg:

// Button.module.css

.my-button {

background-color: white;

color: black;

}

// Button.js

import React from 'react';

import styles from './Button.module.css';

const Button = () => {

return (

<button className={styles.myButton}>

Click

</button>

);

};

export default Button;

5. Emotions

Emotion is a popular CSS-in-JS library that allows you to style React components using JavaScript or TypeScript. It provides a way to write CSS styles in a more programmatic and component-centric manner.

Installation:

npm install @emotion/react @emotion/styled

Eg:

import './App.css';

import React from 'react';

/\*\* @jsxImportSource @emotion/react \*/

import { css } from '@emotion/react';

const styles = css`

color: orange;

font-size: 22px;

`;

function App() {

return (

<div css={styles}>

Hello

</div>

);

}

export default App;