**There are many way we can Style React component:** Stylesheet, CSS Modules, Inline styling, CSS in JS,

**Stylesheet method:** import css file in react component using **import ‘./style.css’;**

Importing stylesheet is used to global style applying. Means when we include a CSS file like above importing way. The CSS style written in this file will apply to the all react component. Means it can create conflict if any imported CSS file contain same class name. so we can solved this problem using **CSS modules system.** CSS module system make each file independent for each file. means private file.

**CSS modules:**

* First, Use css file name like **header.module.css**
* Secondly, import styles from ‘./logo.module.css’

Now all the property inside header.module.css will be accessed like javascript property access by **styles** object. All the css we can use like javascript object. **Styles** contains all styles property. So we can access all the style using styles.navbar as dynamic.

We can apply multiple design, using template literal. className={`${styles.padding}, footer-logo`}

Another way: className={[styles.padding, ‘footer-logo’].join(‘ ‘)]}; here we join with space.

**Inline style:** in html we can write directly as string to apply inline css. But react is not html. Jsx. We need to apply css as dynamically like style={{color: ‘red’, marginTop:’bottom’}}… here first {} is dynamic block. Next {} is css object. Note: css pass as object. And always use camelCase.

**Styed component:**

**npm i styled-components**

extension for colorized the CSS inside styled component. Also it give us CSS suggestion. **Vscode-styled-components** this is vs code extension

**note:** if we use styled-components no need to use sass. Since we can do the similar thing in styled component like sass features.

**Styling components:**

**Import:** styled object . inside styled object we have access all the html element

import styled from "styled-components";

**Create component:** using styled object with dot notation. Then use template literal. Write css inside literal

export const Container = styled.div`

  display: flex;

  … entire css

`;

**Props pass to component:** access props using dynamic props render method. **Props** holds the pass props to pass this component. We can dynamically pass like:

<Tag color="#4361ee">ESCLUSIVE</Tag>

// props pass to styled component

export const Tag = styled.span`

  display: inline-block;

  color: #fff;

  background: ${(props) => props.color};

  … others css

`;

Nesting the styled like sass: use ampersand & to access the parent element.

& access the it’s current top element. **& a** means & access div. so div a{…css} this is nesting like sass

So we can styling our component like sass

// nested css like scss or sass

export const ButtonContainer = styled.div`

  …

  margin-top: 10px;

  // here & means it is parent element. that is div.

  // so & a means a inside div. div> a.

  & a {

    display: inline-block;

    ….

  }

`;

**Note:** always use custom element. Do not write jsx element inside styled component

**Custom styled for a component:**

import styled from "styled-components";

import Title from "../Title";

// styled a component not html element. also pass props

// we need to use styled() constructor and pass the component

//Note:we can not get color here directly from props.

// when we pass a props to this Styledtitle, it first go to the "Title" component as props as className.there we need to receive and used into className

export const StyledTitle = styled(Title)`

  color: ${(props) => props.color};

`;

**Theming:**

Create **theme.js** file, inside it, write dart and light theme changes css as JS object like:

// theme for dark and light mode

export const theme = {

  dark: {

    wrapperBg: "#262626",

    card: "#1e1e1e",

    H1: "#fff",

    A: "#aaa",

    aBorder: "#333",

    aAfter: "#262626",

  },

  light: {

    wrapperBg: "#f0f2f5",

    card: "#fff",

    H1: "#111",

    A: "#111",

    aBorder: "#aaa",

    aAfter: "#111",

  },

};

In the entry point of the Project. Wrap **project entry component** with **ThemeProvider** of styled-components. And pass theme object to it. Inside theme object pass the theme.js object’s dark and light based on the state value.

import { theme } from "./theme";

import { ThemeProvider } from "styled-components";

function App() {

  const [isDark, setIsDark] = useState(true);

  return (

    <>

      <GlobalStyle />

      <ThemeProvider theme={isDark ? theme.dark : theme.light}>

        <Container>

          <Card />

        </Container>

      </ThemeProvider>

    </>

  );

}

**SASS Tutorials: learn with sumit**

SASS: <https://www.youtube.com/watch?v=N3vG6Yt-e6k&t=352s>

**Generate Dynamically stars based on rating:**

{

Array(book.rating).fill().map((\_, index) => (

// ... SVG for full star

))

}

**Resources**

**Styled Components Documentation** - [https://styled-components.com/docs](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbHpSTm5nZWlCTDljaFF5eFZpSVdQd2lEOHl6Z3xBQ3Jtc0tsWm15SmNWTE5DVVZwUm9hUEk0aW5XTFV3dm1qLUdncmhibG40bWFVMlNEUHRYX3dUbk12TWtHbDNLSHVVenBBZW5waTZIQmYyTUl5NjNMQ1FPX256NWhnSTY4SXhwMkdWWHk2QU9pZTltVi1ZMWlvbw&q=https%3A%2F%2Fstyled-components.com%2Fdocs&v=RqwnR2osQqc)

**👍 Styling React Component Previous Tutorial** - [https://lwsbd.link/rstyle](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbGZrcXY1R3VnZ0RuZmlTZUxjeEE4OFpQTGNrUXxBQ3Jtc0tsMHpuLUNIZnpaeDJOdW5NRFVYaDVsSENuQW1oeGZTajNjMGxYajlRcFdSLXc2eGIzWlpkeUw0dlhiV0pycFRvU25Ma1VEZ2FRVzV1bVZnVnEwMDBnM1FUOTEzdjVLU3hrN01zbklnUFh1NmROQXIwdw&q=https%3A%2F%2Flwsbd.link%2Frstyle&v=RqwnR2osQqc)

**👍** **Tagged Template Literal Tutorial** - [• #22 - Tagged Template Literals - Mode...](https://www.youtube.com/watch?v=ws_XgqikD6o&list=PLHiZ4m8vCp9MFjMRp9EEHWKArbi0wdgXG&index=1&t=0s)

**👍 VS Code Styled Components Plugin** - [https://marketplace.visualstudio.com/...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqa2NLU0xLWUlVam9ndzRYRVdRMldreHJWSG8xZ3xBQ3Jtc0tsQzVSU01XWGVFWmkwTGZzNFVHVnU2YmtfSURKVWRrUHZFU0FvZ05fVFkyNzJid3Jtd1dqZHlleUJBRHBNVW1OYUQyVTI4QkZRU3BsSXljRDIyZW0yWmdjQ0E5WGNUYktsTkROeHQtdjkwdENQYjg2aw&q=https%3A%2F%2Fmarketplace.visualstudio.com%2Fitems%3FitemName%3Djpoissonnier.vscode-styled-components&v=RqwnR2osQqc)

**👍** **Render Props Tutorial** - [https://lwsbd.link/rp](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbWEzekhGT2dhRk0tMnZ6YzhzSElvTnRRWHlmZ3xBQ3Jtc0tudVJFaF9jV2l3SUtabGZ2cDVXY1hSR2RjWE1Vem01b2J4Nm5YRHJoMmF1LTdoRmZyLVY2YWNyTTBVeGxGUUFZMTdia25obGNfRzUxa3dIQUp0MmVMMk9ZZkJEZGp0MFhCN0hkV2NBOURvZjNlN2lVdw&q=https%3A%2F%2Flwsbd.link%2Frp&v=RqwnR2osQqc)

**👍 SASS Tutorial** - [https://lwsbd.link/sass](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqblVQSDMxTFZUZFRSS3lUbkZTc0w4b1hQY0U3d3xBQ3Jtc0ttV1oyNmNfOVNmc095MmxmXzhqX1ZQR3BteWs2T2RvOEdhVWlvZnV2clVXLTB6SjdCek5BVmJqRWR4cWVoQTJ6aFV4YzBYRWhzTlJTWThmQkQzQk4wNGtDcU0tZ2ZyM2NMeU1OVFZoVkdfWDU3TlBTaw&q=https%3A%2F%2Flwsbd.link%2Fsass&v=RqwnR2osQqc)

**👍 Higher Order Component** - [https://lwsbd.link/hoc](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqa0FDSjRENjBMTklLUUpHa1FpRzVsTFNkci1CZ3xBQ3Jtc0tta24wVWY3ZDUyZFN3MVBFazQ5UmlDQlFVVzE3dU9rWF9tVjFnRzlwdzBpZ18yeDlEakY3djRZZ1RzQlM4Q1drZlk1Y2FDQVhGWVBqYTZ5TGp4cjRmZjdaQTdBOU00WjJOdWZ2cEhUeTZOTTNBXzBMMA&q=https%3A%2F%2Flwsbd.link%2Fhoc&v=RqwnR2osQqc)

**👍 Context API Tutorial** - [https://lwsbd.link/ctx](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqa2k2TDZPZy1YY3BxRTJuNDBnT0Rvc2tuQnRMZ3xBQ3Jtc0ttdDNlRnhSN0dJN3FCVFlzbWstc3NVT29ldDl3dlNYeUoxSERBb1VZR2d5Y1VOakdTcDlpTkRNR2RXZnI2SkVXaFoweXZtU3ZONjdILWNjN2tBaG9FLS00NVJvR3lkb0IzeDNwREVvZTFvT0Z3Tk1DSQ&q=https%3A%2F%2Flwsbd.link%2Fctx&v=RqwnR2osQqc)