1 creo mi proyecto

2 cambio a jsx al app y modifico el indes. Para que no de problema

3 elimino todo lo del return y pongo <h1>Iniciando proyecto</h1> y ejecuto

4 creo la carpeta componetes y dentro un archivo Navbar.jsx con su estructura sfc

const Navbar = () => {

    return (

        <div>

            <h1>Nabvar</h1>

            <input

                type="color"

                onChange={e => console.log(e.target.value)}

            />

        </div>

     );

}

5 Regreso al app quito la etiqueta e importo mi componente <Navbar />

6 ahora vamos a hacer lo siguiente en el navbar vamos a crear otro input para color y con ellos vamos a cambiar el color de fondo y de fuente del componente para eso usaremos dos consantes y el hook useState y también style de css

const Navbar = () => {

    const [color, setColor] = useState('')

    const [colorTexto, setColorTexto] = useState('')

    return (

        <div style={

            {

                background: color,

                color: colorTexto

            }

        }>

            <h1>Nabvar</h1>

            <input

                type="color"

                onChange={e => setColor(e.target.value)}

            />

            <input

                type="color"

                onChange={e => setColorTexto(e.target.value)}

            />

        </div>

     );

}

7 ahora trabajaremos con contex creamos una carpeta llamada contex y creamos un archivo que se llame ThemeProvider.js y lo creamos como cualquier otro componente sfc

Los context permiten envolver a muchos componentes y usar variables globales

Con el siguiente código

import React from 'react'

export const ThemeContext = React.createContext()

const ThemeProvider = (props) => {

    const themes = {

        color: '#000',

        background: '#eee'

    }

    const [theme, setTheme] = React.useState(themes)

    return (

        <ThemeContext.Provider value={{theme, setTheme}}>

            {props.children}

        </ThemeContext.Provider>

    )

}

export default ThemeProvider

8 ahora debemos envolver al componente Navbar que usamos en el app importamos nuestro ThemeProvider y envolvemos a Navbar

import Navbar from './componentes/Navbar';

import logo from './logo.svg';

import ThemeProvider from './context/ThemeProvider'

function App() {

  return (

    <ThemeProvider>

      <Navbar />

    </ThemeProvider>

  );

}

export default App;

9 Por ultimo editamos el Navbar para que ahora los parámetros se tomen desde el Theme y no conlas constantes que definimos pero aquí imprtamos el ThemeContext

import React, { useState } from "react";

import { ThemeContext } from "../context/ThemeProvider";

const Navbar = () => {

    //const [color, setColor] = useState('')

    //const [colorTexto, setColorTexto] = useState('')

    const {theme, setTheme} = React.useContext(ThemeContext)

    return (

        <div style={

            {

                background: theme.background,

                color: theme.color

            }

        }>

            <h1>Nabvar</h1>

            <input

                type="color"

                onChange={e => setTheme({...theme, color: e.target.value})}

            />

            <input

                type="color"

                onChange={e => setTheme({...theme, background: e.target.value})}

            />

        </div>

     );

}

export default Navbar;