createContext - объявление контекста (можете воспринимать это как store из redux для большего начального понимания).

Context.Provider - компонент, оборачивающий наши компоненты, которые должны иметь доступ к данным.

useContext - хук, позволяющий получить данные из context.

import { FC, createContext, useContext, useState, ReactNode } from 'react';

type Theme = 'light' | 'dark';

interface ThemeContextType {

theme: Theme;

toggleTheme: () => void;

}

export const ThemeContext = createContext<ThemeContextType | undefined>(undefined);

export const ThemeProvider: FC<{ children: ReactNode }> = ({ children }) => {

const [theme, setTheme] = useState<Theme>('light');

const toggleTheme = () => {

setTheme((prevTheme) => (prevTheme === 'light' ? 'dark' : 'light'));

};

return (

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

{children}

</ThemeContext.Provider>

);

};

export const Layout = () => {

return(

<ThemeProvider>

<Header />

<MainContent />

<Footer />

</ThemeProvider>

)

}

**import** {useContext} **from** "react";  
**import** {ThemeContext} **from** "какой-то путь"  
  
**export** **const** Header = () => {  
 **const** {theme, toggleTheme} = useContext(ThemeContext)  
 **return**(  
 <div>  
 <p>Сейчас включена тема: {theme === 'light' ? "Светлая" : "Тёмная"}</p>  
 <button onClick={toggleTheme}>Изменить тему</button>  
 </div>  
 )  
}