Встановлення

npm install --save styled-components

<https://www.styled-components.com/docs/basics#installation>

<https://medium.com/@stasonmars/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE-%D0%BF%D0%BE-%D1%82%D1%80%D1%91%D0%BC-%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%B0%D0%BC-%D1%81%D1%82%D0%B8%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8-%D0%B2-react-2ca5c0c7464b>

|  |  |
| --- | --- |
| Загальна форма | Приклад |
| **const новий\_стилізовний\_компонент** = *styled*. стандарний компонент **`**  **опис стильових правил**  **`** | **const** OuterDiv = *styled*.div**`  position: fixed;  left: 0%;  top:0%;  width:100%;  height: 100%;  background: gray;  opacity: 0.5;  z-index:990 `** |
|  |  |

Перевагою стилізованих компонентів є те, що можна динамічно змінювти стилі у залежності від властивостей у компоненті, де він використовується

Приклад

|  |  |
| --- | --- |
| Опис компонента | **const** MyStyledSpan = *styled*.**span`  background-color:** ${ props => props.**value** < props.minValue ? **'red'** : **'green'**} |
| Використання | **class** App **extends** Component {  render() {  **return** (  <**label**>  Розмір заробітної плати: <**MyStyledSpan minValue=**{3200} **value=**{5000}>5000</**MyStyledSpan**>  </**label**>  )  }  } |
| На екрані |  |

Приклад

|  |  |
| --- | --- |
|  |  |
|  | const Button = styled.button`  */\* Adapt the colors based on primary prop \*/*  background: ${props => props.primary ? "palevioletred" : "white"};  color: ${props => props.primary ? "white" : "palevioletred"};  font-size: 1em;  margin: 1em;  padding: 0.25em 1em;  border: 2px solid palevioletred;  border-radius: 3px;  `;  render(  <div>  <Button>Normal</Button>  <Button primary>Primary</Button>  </div>  ); |

**Полиморфное свойство as**

Есть ещё одна новая возможность styled-components v4, к которой мы относимся с большим энтузиазмом. Речь идёт о нативной поддержке свойства as для любых стилизованных компонентов, что позволяет динамически, во время выполнения программы, влиять на рендеринг. Рассмотрим пример:

import styled from "styled-components"

import { Link } from "react-router-dom"

// <Component /> рендерит в DOM элемент div

const Component = styled.div`

color: red;

`

<Component>Hello World!</Component>

// Но мы можем сделать так, чтобы он рендерил любой другой HTML-тег или компонент!

<Component as="span">Hello World!</Component>

<Component as={Link} to="home">Hello World!</Component>

**Theming**

styled-components has full theming support by exporting a <ThemeProvider> wrapper component. This component provides a theme to all React components underneath itself via the context API. In the render tree all styled-components will have access to the provided theme, even when they are multiple levels deep.

|  |  |
| --- | --- |
| Опис стильових правил  const назва\_об’єкта\_з\_стильовими\_правилами = {  опис стильових правил  }; | const theme = {  main: "mediumseagreen"  }; |
| Опис теми  <ThemeProvider theme={ об’єкт\_з\_стильовими\_правилами }>  компонети, до яких будуть застосовані стильові правила  </ThemeProvider> | <ThemeProvider theme={theme}>  <Button>Themed</Button>  </ThemeProvider> |
| Зчитування даних з теми у внутрішніх компонентах  const Мій\_компонент = styled. базовий\_комопнент `  . . . . . . . . . . . . . . . . .  властивість : ${props => props.theme. значення\_з\_теми };  . . . . . . . . . . . . . . . . .  `; | const Button = styled.button`  font-size: 1em;  margin: 1em;  padding: 0.25em 1em;  border-radius: 3px;  */\* Color the border and text with theme.main \*/*  color: ${props => props.theme.main};  border: 2px solid ${props => props.theme.main};  `; |

*// Define our button, but with the use of props.theme this time*

const Button = styled.button`

font-size: 1em;

margin: 1em;

padding: 0.25em 1em;

border-radius: 3px;

*/\* Color the border and text with theme.main \*/*

color: ${props => props.theme.main};

border: 2px solid ${props => props.theme.main};

`;

*// We are passing a default theme for Buttons that arent wrapped in the ThemeProvider*

Button.defaultProps = {

theme: {

main: "palevioletred"

}

}

*// Define what props.theme will look like*

const theme = {

main: "mediumseagreen"

};

render(

<div>

<Button>Normal</Button>

<ThemeProvider theme={theme}>

<Button>Themed</Button>

</ThemeProvider>

</div>

);