

Styled Components

Styled components est un outil permettant la création de CSS grâce à des composants react. Il permet de fusionner react et css pour une meilleure maintenabilité de votre design.

[Lien vers la documentation officielle](#)

Installation

Pour installer styled-components rien de compliqué :

```
npm i styled-components
npm i -D @types/styled-components
# Pour react native et typescript :
npm i -D @types/styled-components-react-native
```

Styliser un composant

Styled component permet de créer des composants à l'aide de CSS :

```
import styled from 'styled-components'

/**
 * Utiliser `styled` depuis styled components
 * afin de créer un composant stylisé.
 *
 * Vous pouvez spécifier ce que vous souhaitez
 * comme balise HTML à styled, ici nous stylisons un h1 :
 */
export const Title = styled.h1`
  font-size: 2rem;
  font-weight: bold;
  text-align: center;
`
```

Pour utiliser ce composant :

```
// on importe le title
import { Title } from '../Style/App'

/**
 * Prenons un composant lambda
 */
export default function App() {
  return (
    <>
      <Title>Hello !</Title>
    </>
  )
}
```

Pour React Native

Styled components marche très bien avec React Native, pour cela il faut utiliser `styled` depuis `styled-components/native` :

```
import styled from 'styled-components/native'

/**
 * On stylise un text. Vous pouvez ici styliser tout les composants
 * React Native !
 */
export const MyText = styled.Text`
  font-size: 2.6rem;
  text-align: center;
`
```

Créer un style "Global" (attention, uniquement pour le web, pas en native !)

Il est aussi possible de créer du css Global :

```
import { createGlobalStyle } from 'styled-components'

/**
 * Création d'un style global pour notre application
 * web :
 */
export const GlobalStyle = createGlobalStyle`
  html, body {
    margin: 0;
    padding: 0;
  }

  * {
    box-sizing: border-box;
  }
`
```

Pour appliquer ce style global rien de plus simple, il suffit d'utiliser le composant :

```
// on importe le title
import { GlobalStyle, Title } from '../Style/App'

/**
 * Prenons un composant lambda
 */
export default function App() {
  return (
    <>
      <GlobalStyle />
      <Title>Hello !</Title>
    </>
  )
}
```

```
}
```

Hérité d'un autre style

Il est tout à fait possible d'hériter un composant stylisé d'un autre :

```
import styled from 'styled-components'

/**
 * Création d'un boîte
 */
export const Box = styled.div`
  display: flex;
`

/**
 * Création d'une boîte à alignement en colonne qui
 * reprend toutes les propriétés de la Box
 */
export const ColumnBox = styled(Box)`
  flex-direction: column;
`
```

Utiliser un thème :

Il est recommandé et très simple d'utiliser un "thème". De centraliser certaines informations de notre css comme les couleurs, tailles, fonts etc ..

```
import styled from 'styled-components'

/**
 * Création d'un thème
 */
export const Theme = {
  colors: {
    dark: '#e7e7e7',
    white: '#F0F0f0',
  },
}

/**
 * On peut maintenant appliquer ce theme un peu partout très simplement :
 */
export const Box = styled.div`
  background-color: ${Theme.colors.dark};
  color: ${Theme.colors.white};
`
```

Rendre des composants dynamique avec des Props :

Il est aussi possible d'utiliser styled components avec des props personnalisé :

```
import styled from 'styled-components'
```

```
/**
 * Création d'une box qui accepte des props :
 */
export const Box = styled.div<BoxProps>`
  display: flex;
  // Création d'une propriété dynamique grâce à l'utilisation
  // d'une fonction acceptant les props :
  flex-direction: ${({ direction = 'row' }) => direction};
`;

/**
 * Création du type des props pour notre boite :
 */
export type BoxProps = {
  direction?: 'row' | 'row-reverse' | 'column' | 'column-reverse'
}
```