Typography

Use a tipografia para apresentar seu design e conteúdo da forma mais clara e eficiente possível.

O uso de diferentes tamanhos e estilos de uma só vez pode estragar qualquer leiaute. Uma escala tipográfica tem um conjunto limitado de tipos de tamanhos que funcionam bem em conjunto com o leiaute de grade.

{{"component": "modules/components/ComponentLinkHeader.js"}}

Geral

A fonte *Roboto* **não** será carregada automaticamente pelo Material-UI. Você é responsável por carregar quaisquer fontes usadas em sua aplicação. A fonte Roboto possui algumas maneiras fáceis de ser carregada. Para uma abordagem mais avançada, dê uma olhada na seção de customização de temas.

Fonte Roboto via CDN

Temos abaixo um exemplo de markup de link usado para carregar a fonte Roboto de uma CDN:

```
<link
  rel="stylesheet"
  href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
/>
```

Instalar via npm

Você pode instalá-la digitando o comando a seguir em um terminal:

```
npm install @fontsource/roboto
```

Então, você pode importá-la no seu ponto de entrada (entry-point).

```
import '@fontsource/roboto/300.css';
import '@fontsource/roboto/400.css';
import '@fontsource/roboto/500.css';
import '@fontsource/roboto/700.css';
```

Para maiores informações, confira em Fontsource.

Fontsource pode ser configurado para carregar subconjuntos, pesos e estilos específicos. A configuração de tipografia padrão do Material-UI depende apenas dos pesos de fonte de 300, 400, 500 e 700.

Componente

O componente tipografia facilita a aplicação de um conjunto padrão de pesos e tamanhos de fonte na sua aplicação.

{{"demo": "Types.js"}}

Tema

Em algumas situações, talvez você não consiga usar o componente `Typography`. Felizmente, você pode tirar vantagem com o uso das chaves de `tipografia` do tema.

{{"demo": "TypographyTheme.js"}}

Alterando o elemento semântico

O componente de Tipografia (Typography) usa a propriedade `variantMapping` para associar a variação da UI com um elemento semântico. It's important to realize that the style of a typography component is independent from the semantic underlying element.

- Você pode alterar o elemento subjacente para uma ocasião em específico com a propriedade `component`:

```
{/ * Já existe um h1 na página, não vamos duplicá-lo. */
}
<Typography variant="h1" component="h2">
  h1. Heading
</Typography>;
```

- Você pode alterar o mapeamento globalmente usando o tema:

```
const theme = createTheme({
  components: {
    MuiTypography: {
      defaultProps: {
        variantMapping: {
          h1: 'h2',
          h2: 'h2',
          h3: 'h2',
          h4: 'h2',
          h5: 'h2',
          h6: 'h2',
          subtitle1: 'h2',
          subtitle2: 'h2',
          body1: 'span',
          body2: 'span',
        },
      },
    },
  },
});
```

Adding & disabling variants

In addition to using the default typography variants, you can add custom ones, or disable any you don't need. See the Adding & disabling variants example for more info.

System props

As a CSS utility component, the `Typography` supports all `system` properties. You can use them as prop directly on the component. For instance, a margin-top:

```
<Typography mt={2}>
```

Acessibilidade

Alguns fatores chave a seguir para uma tipografia acessível:

- **Cor**. Forneça contraste suficiente entre o texto e seu fundo, confira a razão de contraste de cor WCAG 2.0 mínima recomendada (4.5:1).
- **Tamanho da fonte**. Use unidades relativas (rem) para acomodar as configurações do usuário.
- **Hierarquia de títulos**. Não pule níveis de título. Para resolver esse problema, você precisa separar a semântica do estilo.