

TypeScript

Você pode adicionar tipagem estática para o JavaScript para melhorar a produtividade do desenvolvimento e a qualidade do código graças ao TypeScript.

Uso de `withStyles`

Material-UI requires a minimum version of TypeScript 3.5. Have a look at the [Create React App with TypeScript](#) example.

Dê uma olhada no exemplo [Create React App com TypeScript](#).

```
{
  "compilerOptions": {
    "lib": ["es6", "dom"],
    "noImplicitAny": true,
    "noImplicitThis": true,
    "strictNullChecks": true
  }
}
```

As opções do modo strict são as mesmas que são necessárias para todos os tipos de pacote publicados no namespace `@types/`. Usando uma `tsconfig.json` menos rigorosa ou omitindo algumas das bibliotecas podem causar erros. Para obter a melhor experiência com os tipos, recomendamos configurar `"strict": true`.

Manipulando `value` e manipuladores de eventos

Muitos componentes preocupados com a entrada do usuário oferecem uma propriedade `value` ou manipuladores de eventos que incluem o valor atual em `value`. Na maioria das situações, `value` só é manipulado dentro do React, o que permite que seja de qualquer tipo, como objetos ou matrizes.

No entanto, esse tipo não pode ser verificado em tempo de compilação em situações em que depende de nós filhos do componente, por exemplo, para `Select` ou `RadioGroup`. Isso significa que a opção mais segura é tipando como `unknown` e deixar que o desenvolvedor decida como deseja restringir esse tipo. Não oferecemos a possibilidade de usar um tipo genérico nesses casos, devido [as mesmas razões que `event.target` não é genérico no React](#).

As demonstrações incluem variantes tipadas que usam conversão de tipo. É uma troca aceitável porque os tipos estão todos localizados em um único arquivo e são muito básicos. Você tem que decidir por si mesmo se a mesma troca é aceitável para você. The library types are strict by default and loose via opt-in.

Uso da propriedade `component`

Moved to </customization/theming/#custom-variables>.

Uso da propriedade `component`

No entanto, a ampliação de tipos continuará a causar dores de cabeça se você tentar fazer com que os estilos dependam do tema: