Interruptor

Interruptores alternam o estado de uma única configuração ligado ou desligado.

Interruptores são a forma preferida de ajustes de configuração em mobile. The option that the switch controls, as well as the state it's in, should be made clear from the corresponding inline label.

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

Interruptores básicos

{{"demo": "BasicSwitches.js"}}

Caixas de seleção com FormGroup

Você pode fornecer um rótulo para o `Switch` graças ao componente `FormControlLabel`.

{{"demo": "SwitchLabels.js"}}

Tamanho

Use a propriedade `size` para alterar o tamanho do interruptor.

{{"demo": "SwitchesSize.js"}}

Cor

{{"demo": "ColorSwitches.js"}}

Controlled

Você pode controlar o interruptor com as propriedades `checked` e `onChange`:

{{"demo": "ControlledSwitches.js"}}

Interruptores com FormGroup

`FormGroup` is a helpful wrapper used to group selection controls components that provides an easier API. However, you are encouraged to use Checkboxes instead if multiple related controls are required. (Veja: Quando usar).

{{"demo": "SwitchesGroup.js"}}

Interruptores customizados

Aqui estão alguns exemplos de customização do componente. Você pode aprender mais sobre isso na página de documentação de sobrescritas.

{{"demo": "CustomizedSwitches.js"}}

🎨 If you are looking for inspiration, you can check MUI Treasury's customization examples.

Posicionamento do rótulo

The switch also comes with an unstyled version. It's ideal for doing heavy customizations and minimizing bundle size.

Unstyled component

```
import SwitchUnstyled from '@mui/base/SwitchUnstyled';
```

The `SwitchUnstyled` component provides default components and assigns CSS classes you can style entirely on your own. You are free to choose any styling solution - plain CSS classes, a CSS framework, Emotion, etc. It is also possible to replace these default components by other HTML elements or custom components. It is also possible to replace these default components by other HTML elements or custom components.

There are three components you can override by the `components` prop: `Root`, `Thumb` and `Input`. Each one's props can be set using the `componentsProps` object.

{{"demo": "UnstyledSwitches.js"}}

useSwitch hook

For the ultimate customizability, a `useSwitch` hook is available. It accepts almost the same options as the SwitchUnstyled component minus the `component`, `components`, and `componentsProps` props.

```
import { useSwitch } from '@mui/base/SwitchUnstyled';
```

Basic example

{{"demo": "UseSwitchesBasic.js"}}

Customized look and feel

{{"demo": "UseSwitchesCustom.js"}}

Quando usar

Você pode alterar o posicionamento do rótulo:

{{"demo": "FormControlLabelPosition.js"}}

Acessibilidade

- Caixas de seleção vs. interruptores

Accessibility

- Ele irá renderizar um elemento com a regra de `checkbox` e não `switch`, pois esta regra não é amplamente suportada ainda. Por favor, teste primeiro se a tecnologia assistiva do seu público-alvo suporta essa regra corretamente. Em seguida, você pode alterar a regra com `<Switch inputProps={{ role: 'switch' }}>`
- Todos os controles de formulário devem ter rótulos, e isso inclui os botões de opção, caixas de seleção e interruptores. Na maioria dos casos, isso é feito usando o elemento `<label>` (FormControlLabel).

- Quando um rótulo não pode ser usado, é necessário adicionar um atributo diretamente no componente de entrada. Nesse caso você pode aplicar um atributo adicional (por exemplo, `aria-label`, `aria-labelledby`, `title`) através da propriedade `inputProps`.

```
<code>
  <Switch value="checkedA" inputProps={{ 'aria-label': 'Switch A' }} />
</code>
```