

Switch 开关组件

开关控制能切换单个设置的开/关两个状态。

开关组件是在移动设备上调整设置的首选方式。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"}}
```

基本的开关

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

标签

得益于 `FormControlLabel` 组件，您可以为 `Switch` 提供标签。

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

Size 大小

使用 `size` 属性来改变开关的大小。

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

Color 颜色

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

Controlled

You can control the switch with the `checked` and `onChange` props:

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

带有 FormGroup 的开关

`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. (参见: [何时使用](#)).

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

自定义样式开关

你可以参考以下一些例子来自定义组件。您可以在 [重写文档页面](#) 中了解更多有关此内容的信息。

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

👉 If you are looking for inspiration, you can check [MUI Treasury's customization examples](#).

标签放置

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

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. 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. 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"}}
```

何时使用

你可以更改标签的位置：

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

无障碍设计

- [多选框 对比 Switches \(开关控件\)](#)

Accessibility

- 它将渲染一个带有 `checkbox` 而不是 `switch` 角色的元素，鉴于该属性尚未得到广泛支持。请首先测试目标受众的辅助技术 (assistive technology) 是否正确支持此 `role` 属性。或者您可以使用 `<Switch inputProps={{ role: 'switch' }}>` 来更改 `role` 属性。
- 所有表单控件都应该带有标签，而这包括了单选按钮，复选框和开关。在大多数情况下，这是通过使用一个 `<label>` 元素 ([FormControlLabel](#)) 实现的。

- 如果无法使用标签，您则必须在输入组件中直接添加属性。在这种情况下，您可以通过 `inputProps` 属性来应用附加的属性（例如 `aria-label`，`aria-labelledby`，`title`）。

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