

# Checkbox 选择框

在一个集合内，用户可以通过多选框组件进行一项或者多项选择。

多选框可用于打开或关闭选项。

若一个列表存在多个选择项时，使用多选框替代开关控件，可以节省空间。若只存在一个选择项，请避免使用多选框，而改用开关控件。

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

## 简单的多选框

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

## 标签

使用 `FormControlLabel` 组件，您可以为 `Checkbox` 设置一个标签。

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

## Size 大小

使用 `size` 属性或调整 `svg` 图标字体大小以更改复选框的大小。

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

## Color 颜色

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

## 图标

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

## Controlled

您可以使用 `checked` 和 `onChange` 属性控制复选框：

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

## 不确定的状态

多选框在表单中只能存在两种状态：已选中或未选中。在其状态下提交的值只有存在和空两种形式。Visually, there are **three** states a checkbox can be in: checked, unchecked, or indeterminate.

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

 When `indeterminate` is set, the value of the `checked` prop only impacts the form submitted values. It has no accessibility or UX implications. It has no accessibility or UX implications.

## 表单组

`FormGroup` is a helpful wrapper used to group selection control components.

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

## 标签放置

你可以更改标签的位置:

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

## 自定义的多选框

以下是自定义组件的一个示例。您可以在 [重写文档页面](#) 中了解更多有关此内容的信息。

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

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

## 什么时候使用

- [多选框 对比 单选按钮 \(Radio Buttons\)](#)
- [多选框 对比 Switches \(开关控件\)](#)

## 无障碍设计

(WAI-ARIA: <https://www.w3.org/TR/wai-aria-practices/#checkbox>)

- 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` ）。

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