

Button 按钮

只需轻点按钮，用户就可以触发动作或做出选择。

按钮可以展示用户能进行的操作。他们通常直接放置在您的用户界面中，例如：

- Modal windows（模态窗口）
- Forms（表单）
- Cards（卡片）
- Toolbars（工具栏）

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

Basic Button

`Button`（按钮）组件有三种形式：文本（默认值）、实心，以及描边。

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

文本按钮

[Text buttons](#) are typically used for less-pronounced actions, including those located: in dialogs, in cards. 在卡片中，文本按钮有助于强调卡片的内容。在卡片中，文本按钮有助于强调卡片的内容。

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

实心按钮

[实心按钮](#) 表示高度的强调，你根据它们的立体效果和填充颜色来区分彼此。它们用于触发应用程序所具有的主要功能。

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

你也可以使用属性 `disableElevation` 属性来消除实心按钮的立体效果。

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

描边按钮

[Outlined buttons](#) are medium-emphasis buttons. They contain actions that are important but aren't the primary action in an app. 它们用于触发应用程序中重要、但并非主要的那些操作。They contain actions that are important but aren't the primary action in an app.

你也可以将描边按钮作为比实心按钮次要一点的替代方案，或者用来作为比文本按钮重要一点的展示。

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

Handling clicks 处理点击

所有组件都接受 `onClick` 处理程序，该处理程序被应用到根 DOM 元素中。

```
<Button
  onClick={() => {
    alert('clicked');
  }}
/>
```

```
>  
  点击我  
</Button>
```

请注意，文档里组件的 API 部分 [避免](#) 提到原生的属性（还是有很多）。

Color 颜色

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

In addition to using the default button colors, you can add custom ones, or disable any you don't need. In addition to using the default button colors, you can add custom ones, or disable any you don't need. See the [Adding new colors](#) example for more info. See the [Adding new colors](#) example for more info.

尺寸

For larger or smaller buttons, use the `size` prop.

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

Upload button 上传按钮

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

带有 icons（图标）和 label（标签）的按钮

因为相比纯文本来说用户对图标更敏感，所以有些时候你可能希望为某些按钮设置图标，以增强应用程序的用户体验。例如，如果您有删除按钮，则可以使用垃圾箱图标对其进行标记。

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

图标按钮

图标按钮通常位于应用栏和工具栏中。

图标也适用于允许选择单个选项的切换按钮或取消选择，例如向项目添加或删除星标。

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

尺寸

如果需要更大或者更小的图标按钮，请使用 `size` 属性。

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

Cursor 鼠标悬浮的禁用

使用 `color` 属性来把调色板应用到组件上。

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

Customized Buttons（自定义按钮）

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": "CustomizedButtons.js", "defaultCodeOpen": false}}
```

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

载入按钮 (Loading buttons)

The loading buttons can show loading state and disable interactions.

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

切换此切换按钮，可以查看不同状态之间的转换。

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

组合按钮

文本按钮，包含按钮，浮动操作按钮和图标按钮构建在同一组件之上：`ButtonBase`。你可以利用这种低级组件来构建自定义交互功能。

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

Third-party routing library (第三方路由库)

One frequent use case is to perform navigation on the client only, without an HTTP round-trip to the server.

`ButtonBase` 组件提供了 `component` 属性来处理此用例。`ButtonBase` 组件提供了 `component` 属性来处理此用例。Here is a [more detailed guide](#).

设计局限

Cursor 鼠标悬浮的禁用

在 disabled 不可用的按钮上，`ButtonBase` 组件会有这个设置：`pointer-events: none;`，这样一来不可用样式的鼠标悬浮就不会出现。

若您希望使用 `not-allowed`，您有以下两种选择：

1. **仅使用 CSS。** 当 `<button>` 元素在处于禁用的状态时，你可以移除其鼠标样式。

```
.MuiButtonBase-root:disabled {  
  cursor: not-allowed;  
  pointer-events: auto;  
}
```

然而：

- 如您要 [在 disabled 不可用元素上显示提示文本](#)，则需添加 `pointer-events: none;`
 - 若您加载除了一个 `button` 元素之外的元素，例如，一个链接 `<a>` 元素，指针是不会改变的。
2. **改变 DOM。** 您可以这样封装按钮：

```
<span style={{ cursor: 'not-allowed' }}>  
  <Button component={Link} disabled>  
    disabled  
  </Button>  
</span>
```

```
</Button>
</span>
```

这个方法能支持任何元素，例如，一个 `<a>` 元素。

Unstyled

The button 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 ButtonUnstyled from '@mui/base/ButtonUnstyled';
```

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

Customizing the root element

By default, the `ButtonUnstyled` renders a native `button` element. You are free to override this by setting the `component` or `components.Root` prop. If a non-interactive element (such as a `span`) is provided this way, the `ButtonUnstyled` will take care of adding accessibility attributes. You are free to override this by setting the `component` or `components.Root` prop. If a non-interactive element (such as a `span`) is provided this way, the `ButtonUnstyled` will take care of adding accessibility attributes.

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

Compare the attributes on the `span` with the `button` from the previous demo.

Complex customization

You are not limited to using HTML elements for the button structure. SVG elements, even with complex structure, are equally acceptable. SVG elements, even with complex structure, are equally acceptable.

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

useButton hook

```
import { useButton } from '@mui/base/ButtonUnstyled';
```

If you need to use Button's functionality in another component, you can use the `useButton` hook. It returns props to be placed on a custom button element and fields representing the internal state of the button. It returns props to be placed on a custom button element and fields representing the internal state of the button.

The `useButton` hook requires the ref of the element it'll be used on. The `useButton` hook requires the ref of the element it'll be used on. Additionally, you need to provide the `component` prop (unless you intend to use the plain `button`).

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