

Tooltip

当用户鼠标悬停，聚焦或者轻触一个元素时，工具提示组件会显示一段有意义的文本。

当它触发时，`Tooltip` 会显示标识一个元素的文本标签，比如对该功能的描述。

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

简单的工具提示

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

文字提示的位置

`Tooltip` 有 12 个位置选项。它们没有方向箭头，而是依靠源头发出的运动来传递方向。

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

自定义文字提示

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": "CustomizedToolips.js"}}
```

文字提示的箭头

您可以通过添加 `arrow` 属性向提示标签增加箭头指示器，从而可以凸显所指示的元素。

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

自定义子元素

文字提示组件需要将 DOM 事件监听器应用到其子元素当中。如果子元素是自定义的 React 组件，你需要确保它能够将其属性传播到底部的 DOM 元素。

```
const MyComponent = React.forwardRef(function MyComponent(props, ref) {
  // 将属性传递给底部的 DOM 元素。
  return <div {...props} ref={ref}>Bin</div>
});

// ...

<Tooltip title="删除">
  <MyComponent>
</Tooltip>
```

您可以在[包装的组件](#)指南中找到类似的概念。

触发器

你可以定义各种类型的事件来触发显示工具提示组件。

The touch action requires a long press due to the `enterTouchDelay` prop being set to `700` ms by default.

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

可控的工具提示

You can use the `open` , `onOpen` and `onClose` props to control the behavior of the tooltip.

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

变量宽度

`Tooltip` 为了保证较好的阅读性，会自动将较长的文字折行。

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

交互式

Tooltips are interactive by default (to pass [WCAG 2.1 success criterion 1.4.13](#)). 若用户在 `leaveDelay` 过期之前将鼠标悬停在工具提示上时，它则不会被关闭。 It won't close when the user hovers over the tooltip before the `leaveDelay` is expired. You can disable this behavior (thus failing the success criterion which is required to reach level AA) by passing `disableInteractive` .

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

禁用元素

默认情况下，`<Button>` 等 disabled 的元素不会触发用户交互，因此 `Tooltip` 不会在 hover 等正常事件上激活显示。若想容纳已禁用的元素激活工具提示，请添加一个简单的包装元素，如 `span` 。

 为了在 Safari 中正常显示，在文字提示的包装组件中，您至少需要一个 `display` 为 `block` 或 `flex` 的元素。

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

如果你没有包装从 `ButtonBase` 继承的 Material-UI 组件，譬如一个原生的 `<button>` 元素，当禁用元素的时候，你应该将 `pointer-events: none`; 这个 CSS 属性添加到您的元素中：

```
<Tooltip title="您没有足够的操作权限">
  <span>
    <button disabled={disabled} style={disabled ? { pointerEvents: 'none' } : {}}>
      A disabled button
    </button>
  </span>
</Tooltip>
```

过渡动画

Use a different transition.

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

跟踪光标

You can enable the tooltip to follow the cursor by setting `followCursor={true}` .

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

虚拟元素

In the event you need to implement a custom placement, you can use the `anchorEl` prop: The value of the `anchorEl` prop can be a reference to a fake DOM element. You need to create an object shaped like the [VirtualElement](#) .

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

显示和隐藏组件

当用户的鼠标悬浮在该元素时工具提示会立即显示，并且当用户鼠标离开时立即隐藏。A delay in showing or hiding the tooltip can be added through the `enterDelay` and `leaveDelay` props, as shown in the Controlled Tooltips demo above.

在移动设备上使用时，用户长按元素就会显示出文字提示，并且持续 1500ms 之后就会自动隐藏。You can disable this feature with the `disableTouchListener` prop.

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

无障碍设计

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

By default, the tooltip only labels its child element. This is notably different from `title` which can either label **or** describe its child depending on whether the child already has a label. For example, in:

```
<button title="some more information">一个按钮</button>
```

the `title` acts as an accessible description. If you want the tooltip to act as an accessible description you can pass `describeChild` . Note that you shouldn't use `describeChild` if the tooltip provides the only visual label.

Otherwise, the child would have no accessible name and the tooltip would violate [success criterion 2.5.3 in WCAG 2.1](#).

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