

# Popper 弹出提示

使用弹出提示组件，您可在另一个元素之上显示一些内容。这可以替代 react-popper。

以下是 弹出提示 组件的一些重要功能：

- 📦 Popper 依赖第三方库 ([Popper.js](#)) 来实现完美的定位。
- 🚒 这是 react-popper 的一个替代性 API。它是为了简单性而设计。
- 📦 [8kB 已压缩的包](#)。
- 为了避免渲染问题，子组件作为页面 body 的 [Portal](#)。您可以使用 `disablePortal` 来禁用此行为。
- 不同于 [Popper](#) 组件，您可以自由实现滚动（scroll）行为。弹出提示的位置会随着视口中的可用面积而更新。
- Clicking away 不会隐藏 弹出提示 组件。若您需要这个功能，请使用 [ClickAwayListener](#) - 可以参照 [menu 文档章节](#) 中的一个样例。
- 创建一个新 `Popper.js` 实例时，`anchorEl` 作为一个参考对象在其中传递。

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

## 基础的弹出提示组件

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

## 过渡动画

通过渲染附属的子元素和一个过渡组件，您可以给弹出提示组件的打开/关闭状态加上动画效果。此组件应遵守以下条件：

- 作为弹出提示的直接子元素。
- 当进入过渡时调用 `onEnter` 回调属性。
- 当退出过渡完成后应该调用 `onExited` 回调属性。这两个回调属性保证了弹出提示组件在关闭并展示完过渡动画时，将会移除子内容。

弹出提示组件已经内嵌支持 [react-transition-group](#)。

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

Alternatively, you can use [react-spring](#).

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

## 弹出提示组件的位置

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

## 滚动 (Scroll) 测试

```
{{"demo": "ScrollPlayground.js", "hideToolbar": true, "bg": true}}
```

## 虚拟元素

而 `anchorEl` 属性的值可以是对一个占位 DOM 元素的引用。您需要创建一个类似 [VirtualElement](#) 的对象。

高亮文本来显示弹出提示组件：

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

## 补充项目

对于更高级的用例，您可以利用：

### PopupState helper

在大多数情况下，这个第三方包 [material-ui-popup-state](#) 可以处理弹出提示组件的 state。

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

## Unstyled

The @mui/base package contains an unstyled version of Popper - PopperUnstyled. It does not have a dependency on @mui/material. The only difference between Popper and PopperUnstyled is the support for theming. Popper can read the `direction` field from the current theme, while PopperUnstyled accepts the `direction` prop instead. It does not have a dependency on @mui/material. The only difference between Popper and PopperUnstyled is the support for theming. Popper can read the `direction` field from the current theme, while PopperUnstyled accepts the `direction` prop instead.

```
import Popper from '@mui/base/PopperUnstyled';
```