

# Popper

A Popper can be used to display some content on top of another. It's an alternative to react-popper.

Some important features of the `Popper` component:

- 📦 Popper relies on the 3rd party library ([Popper.js](#)) for perfect positioning.
- 🔥 It's an alternative API to react-popper. It aims for simplicity.
- 📄 [8 kB gzipped](#).
- The children is `Portal` to the body of the document to avoid rendering problems. You can disable this behavior with `disablePortal`.
- The scroll isn't blocked like with the `Popover` component. The placement of the popper updates with the available area in the viewport.
- Clicking away does not hide the `Popper` component. If you need this behavior, you can use `ClickAwayListener` - see the example in the [menu documentation section](#).
- The `anchorEl` is passed as the reference object to create a new `Popper.js` instance.

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

## Basic popper

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

## Transitions

The open/close state of the popper can be animated with a render prop child and a transition component. This component should respect the following conditions:

- Be a direct child descendent of the popper.
- Call the `onEnter` callback prop when the enter transition starts.
- Call the `onExited` callback prop when the exit transition is completed. These two callbacks allow the popper to unmount the child content when closed and fully transitioned.

Popper has built-in support for [react-transition-group](#).

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

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

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

## Positioned popper

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

## Scroll playground

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

## Virtual element

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](#) .

Highlight part of the text to see the popper:

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

## Complementary projects

For more advanced use cases you might be able to take advantage of:

### PopupState helper

There is a 3rd party package [material-ui-popup-state](#) that takes care of popper state for you in most cases.

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