



# Click-away listener

Detect if a click event happened outside of an element. It listens for clicks that occur somewhere in the document.

-  [981 B gzipped](#).
-  Supports portals

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

## Example

For instance, if you need to hide a menu dropdown when people click anywhere else on your page:

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

Notice that the component only accepts one child element. You can find a more advanced demo on the [Menu documentation section](#).

## Portal


The following demo uses `Portal` to render the dropdown into a new "subtree" outside of current DOM hierarchy.

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

## Leading edge

By default, the component responds to the trailing events (click + touch end). However, you can configure it to respond to the leading events (mouse down + touch start).

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

 *In this mode, only interactions on the scrollbar of the document is ignored.*

## Accessibility

By default `<ClickAwayListener />` will add an `onClick` handler to its children. This can result in e.g. screen readers announcing the children as clickable. However, the purpose of the `onClick` handler is not to make children interactive.

In order to prevent screen readers from marking non-interactive children as "clickable" add `role="presentation"` to the immediate children:

```
<ClickAwayListener>
  <div role="presentation">
    <h1>non-interactive heading</h1>
  </div>
</ClickAwayListern>
```

This is also required to fix a quirk in NVDA when using Firefox that prevents announcement of alert messages (see [mui/material-ui#29080](#)).