Menu

Menus display a list of choices on temporary surfaces.

A menu displays a list of choices on a temporary surface. It appears when the user interacts with a button, or other control.

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

Basic menu

A basic menu opens over the anchor element by default (this option can be changed via props). When close to a screen edge, a basic menu vertically realigns to make sure that all menu items are completely visible.

Choosing an option should immediately ideally commit the option and close the menu.

Disambiguation: In contrast to simple menus, simple dialogs can present additional detail related to the options available for a list item or provide navigational or orthogonal actions related to the primary task. Although they can display the same content, simple menus are preferred over simple dialogs because simple menus are less disruptive to the user's current context.

{{"demo": "BasicMenu.js"}}

Icon menu

In desktop viewport, padding is increased to give more space to the menu.

{{"demo": "IconMenu.js", "bg": true}}

Dense menu

For the menu that has long list and long text, you can use the `dense` prop to reduce the padding and text size.

{{"demo": "DenseMenu.js", "bg": true}}

Selected menu

If used for item selection, when opened, simple menus places the initial focus on the selected menu item. The currently selected menu item is set using the `selected` prop (from ListItem). To use a selected menu item without impacting the initial focus, set the `variant` prop to "menu".

{{"demo": "SimpleListMenu.js"}}

Positioned menu

Because the `Menu` component uses the `Popover` component to position itself, you can use the same positioning props to position it. For instance, you can display the menu on top of the anchor:

{{"demo": "PositionedMenu.js"}}

MenuList composition

The `Menu` component uses the `Popover` component internally. However, you might want to use a different positioning strategy, or not blocking the scroll. For answering those needs, we expose a `MenuList` component that you can compose, with `Popper` in this example.

The primary responsibility of the `MenuList` component is to handle the focus.

{{"demo": "MenuListComposition.js", "bg": true}}

Account menu

`Menu` content can be mixed with other components like `Avatar`.

{{"demo": "AccountMenu.js"}}

Customization

Here is an example of customizing the component. You can learn more about this in the overrides documentation page.

{{"demo": "CustomizedMenus.js"}}

The `MenuItem` is a wrapper around `ListItem` with some additional styles. You can use the same list composition features with the `MenuItem` component:

🎨 If you are looking for inspiration, you can check MUI Treasury's customization examples.

Max height menu

If the height of a menu prevents all menu items from being displayed, the menu can scroll internally.

{{"demo": "LongMenu.js"}}

Limitations

There is a flexbox bug that prevents `text-overflow: ellipsis` from working in a flexbox layout. You can use the `Typography` component with `noWrap` to workaround this issue:

{{"demo": "TypographyMenu.js", "bg": true}}

Change transition

Use a different transition.

{{"demo": "FadeMenu.js"}}

Context menu

Here is an example of a context menu. (Right click to open.)

{{"demo": "ContextMenu.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 menu state for you in most cases.

{{"demo": "MenuPopupState.js"}}