Menu 菜单

菜单在临时出现的位置上显示了一系列的选项。

菜单在临时的表面上显示选择列表。 当用户和一个按钮、或者其他控制元件交互的时候，菜单会出现。

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

基础菜单

默认情况下，基础菜单会在描点元素上打开（这种方式可以使用属性来 改变）。 当靠近屏幕边缘时，简单菜单会在垂直方向上重新对齐，以确保所有菜单子项（menu items）都完全可见。

理想状态下，选择一个选项是会立刻执行此选项并且关闭整个菜单。

解疑：与简单菜单不同，一个简单的对话框可以提供与列表项可用选项相关的额外细节，或提供与主要任务相关的导航或正交操作。 虽然它们都可以显示相同的内容，但是菜单组件比对话框组件更受欢迎，因为相比之下菜单组件对用户的干扰更小。

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

选择菜单

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

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

菜单定位

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}}

MenuList 组合

If used for item selection, when opened, simple menus places the initial focus on the selected menu item. If used for item selection, when opened, simple menus places the initial focus on the selected menu item. If used for item selection, when opened, simple menus places the initial focus on the selected menu item. 通过 `selected` 属性（在ListItem中），您能够设置当前被选中的选项。 To use a selected menu item without impacting the initial focus, set the `variant` prop to "menu".

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

自定义菜单

因为 `Menu` 组件是基于 `Popover` 组件来进行定位的，所以你也可以使用与之相同的 定位属性 来对它的位置进行改变。 例如，你可以在描点的下方显示菜单：

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

MenuList 组合

在 菜单 组件的内部，其实使用了 `Popover` 组件。 但是，您可能想要使用不同的元素定位的方式，或者您不想禁止页面的滚动。 为了满足这些需求，我们公开了一个 `MenuList` 组件，您可以像下面例子中这样结合 `Popper` 来编写自己的菜单组件。

`MenuList` 组件的主要任务是处理焦点。

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

设计局限

如果最大高度的菜单仍无法显示所有菜单项，则菜单可以在内部滚动。

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

更改过渡动画

以下是自定义组件的一个示例。 您可以在 重写文档页面 中了解更多有关此内容的信息。

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

`MenuItem` 的原理是用额外的样式包装了 `ListItem` 组件。 你可以将同样的列表合成功能来装饰 `MenuItem` 组件：

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

快捷菜单 Context menu

如果最大高度的菜单仍无法显示所有菜单项，则菜单可以在内部滚动。

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

补充项目

有 一个 flexbox 的错误，在 flexbox 的布局中，无法正常使用 `text-overflow: ellipsis`。 您可以使用 `Typography` 组件和 `noWrap` 来解决此问题：

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

更改过渡动画

使用不同的过渡动画。

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

快捷菜单 Context menu

这是一个快捷菜单的示例。 （单击右键就可以打开。）

{{"demo": "ContextMenu.js"}}

Complementary projects

对于更高级的用例，您可以利用：

PopupState helper

这里有一个第三方包 `material-ui-popup-state` 在大部分情况下，它都能帮你处理好菜单组件的状态。

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