

Unstyled menu

The menu components provide your users with a list of options on temporary surfaces.

MenuUnstyled and MenuItemUnstyled components

```
import MenuUnstyled from '@mui/base/MenuUnstyled';
import MenuItemUnstyled from '@mui/base/MenuItemUnstyled';
```

You can use the `MenuUnstyled` component to create custom menus. It renders a list of items in a popup that users can navigate through with a mouse or keyboard.

When not customized, the `MenuUnstyled` component renders a plain `` element.

Basic usage

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

Wrapping MenuItems

`MenuItemUnstyled` components don't have to be direct children of a `MenuUnstyled` component. You can wrap them in any component needed to achieve the desired appearance.

In addition to `MenuItemUnstyled` components, the `MenuUnstyled` component can also contain non-interactive children, such as help text.

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

Customization

Slots

`MenuUnstyled` has two slots:

- **Root** - represents the popup container the menu is placed in; set to `PopperUnstyled` by default. Can be customized by setting the `component` or `components.Root` props.
- **Listbox** - set to `` by default

The `MenuItemUnstyled` component has one slot: the root, which renders an `` element by default. Just like `MenuUnstyled`, it can be customized by setting the `component` or `components.Root` props.

CSS classes

`MenuUnstyled` can set the following class:

- `Mui-expanded` - set when the menu is open; this class is set on both Root and Popper slots

`MenuItemUnstyled` can set the following classes:

- `Mui-disabled` - set when the MenuItem has the `disabled` prop
- `Mui-focusVisible` - set when the MenuItem is highlighted via keyboard navigation. This is a polyfill for the native `:focus-visible` pseudoclass as it's not available in Safari.

The useMenu and useMenuItem hooks

```
import { useMenu } from '@mui/base/MenuUnstyled';  
import { useMenuItem } from '@mui/base/MenuItemUnstyled';
```

You can use the `useMenu` and `useMenuItem` hooks to apply the functionality of the unstyled menu components to a different component. These hooks gives you the most customization options, but require more work to implement. Using the hooks allows you to take full control over the rendered components, their props and CSS classes.

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

Unstyled components and their corresponding hooks work interchangeably with one another—for example, you can create a `MenuUnstyled` component that contains menu items built with the `useMenuItem` hook.