

# Package: muiMaterial (via r-universe)

July 2, 2026

**Type** Package

**Title** 'Material UI' for 'shiny' Apps and 'Quarto'

**Version** 0.2.2

**Maintainer** Felix Luginbuhl <[felix.luginbuhl@protonmail.ch](mailto:felix.luginbuhl@protonmail.ch)>

**Description** Wraps the 'Material UI' React components <<https://mui.com/>> for use in R, 'shiny' applications and 'quarto' documents, including inputs, layouts, navigation, and surfaces. All inputs come with R usage examples.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** en-US

**Depends** R (>= 4.1)

**Imports** checkmate (>= 2.0.0), htmltools (>= 0.5.0), shiny (>= 1.7.0), shiny.react (>= 0.4.0)

**Suggests** knitr, reactRouter (>= 0.2.0), shiny.router, testthat (>= 3.0.0)

**RoxygenNote** 7.3.3

**URL** <https://felixluginbuhl.com/muiMaterial/>

**BugReports** <https://github.com/lgnbhl/muiMaterial/issues>

**Config/testthat/edition** 3

**Collate** 'aaa-utils.R' 'ThemeProvider.R' 'browse.R' 'components-extra.R' 'components.R' 'docs-templates.R' 'documentation.R' 'inputs.R' 'muiMaterial-package.R' 'muiMaterialDependency.R' 'muiMaterialExample.R' 'muiMaterialPage.R' 'reexports.R' 'static-tabs.R' 'triggerId.R'

**NeedsCompilation** no

**Author** Felix Luginbuhl [aut, cre, cph] (ORCID: <<https://orcid.org/0009-0008-6625-2899>>), MUI [cph] (Copyright holder of the bundled '@mui/material', '@mui/lab', '@mui/system' and '@mui/utils' JavaScript libraries), Emotion

team [cph] (Copyright holder of the bundled '@emotion/react' and '@emotion/styled' JavaScript libraries), Meta Platforms, Inc. and affiliates [cph] (Copyright holder of the bundled 'react-is' JavaScript library; 'react' and 'react-dom' are declared as peer dependencies and provided at runtime by 'shiny.react')

**Repository** <https://cran.r-universe.dev>

**Date/Publication** 2026-07-02 12:10:02 UTC

**RemoteUrl** <https://github.com/cran/muiMaterial>

**RemoteRef** HEAD

**RemoteSha** bca9fc6f1bac433776ea89c8c2076444e4637c49

## Contents

Accordion . . . . .	5
AccordionActions . . . . .	6
AccordionDetails . . . . .	7
AccordionSummary . . . . .	8
Alert . . . . .	9
AlertTitle . . . . .	10
AppBar . . . . .	11
Autocomplete . . . . .	12
Avatar . . . . .	17
AvatarGroup . . . . .	19
Backdrop . . . . .	20
Badge . . . . .	21
BottomNavigation . . . . .	23
BottomNavigationAction . . . . .	24
Box . . . . .	25
Breadcrumbs . . . . .	26
Button . . . . .	27
ButtonBase . . . . .	29
ButtonGroup . . . . .	30
Card . . . . .	32
CardActionArea . . . . .	32
CardActions . . . . .	33
CardContent . . . . .	34
CardHeader . . . . .	34
CardMedia . . . . .	36
Checkbox . . . . .	37
Chip . . . . .	39
CircularProgress . . . . .	40
ClickAwayListener . . . . .	41
Collapse . . . . .	42
Container . . . . .	43
CssBaseline . . . . .	44

Dialog . . . . .	45
Dialog.triggerId . . . . .	47
DialogActions . . . . .	48
DialogContent . . . . .	49
DialogContentText . . . . .	49
DialogTitle . . . . .	50
Divider . . . . .	51
Drawer . . . . .	52
Drawer.triggerId . . . . .	54
Fab . . . . .	54
Fade . . . . .	56
FilledInput . . . . .	57
FormControl . . . . .	59
FormControlLabel . . . . .	61
FormGroup . . . . .	62
FormHelperText . . . . .	63
FormLabel . . . . .	64
GlobalStyles . . . . .	65
Grid . . . . .	66
Grow . . . . .	67
Icon . . . . .	68
IconButton . . . . .	69
ImageList . . . . .	70
ImageListItem . . . . .	71
ImageListItemBar . . . . .	72
Input . . . . .	73
InputAdornment . . . . .	75
InputBase . . . . .	76
InputLabel . . . . .	79
LinearProgress . . . . .	80
Link . . . . .	81
List . . . . .	82
ListItem . . . . .	83
ListItemAvatar . . . . .	84
ListItemButton . . . . .	85
ListItemIcon . . . . .	86
ListItemSecondaryAction . . . . .	87
ListItemText . . . . .	88
ListSubheader . . . . .	89
LoadingButton . . . . .	90
Masonry . . . . .	91
Menu . . . . .	92
Menu.triggerId . . . . .	94
MenuItem . . . . .	94
MenuList . . . . .	96
MobileStepper . . . . .	97
Modal . . . . .	98
Modal.triggerId . . . . .	100

muiMaterialDependency . . . . .	101
muiMaterialExample . . . . .	101
muiMaterialPage . . . . .	102
NativeSelect . . . . .	103
NoSsr . . . . .	104
OutlinedInput . . . . .	105
Pagination . . . . .	108
PaginationItem . . . . .	110
Paper . . . . .	111
Popover . . . . .	112
Popover.triggerId . . . . .	114
Popper . . . . .	115
Portal . . . . .	116
print.muiMaterial . . . . .	117
Radio . . . . .	118
RadioGroup . . . . .	120
Rating . . . . .	121
ScopedCssBaseline . . . . .	123
Select . . . . .	124
Skeleton . . . . .	126
Slide . . . . .	127
Slider . . . . .	128
Snackbar . . . . .	131
SnackbarContent . . . . .	133
SpeedDial . . . . .	134
SpeedDialAction . . . . .	136
SpeedDialIcon . . . . .	137
Stack . . . . .	138
Step . . . . .	139
StepButton . . . . .	140
StepConnector . . . . .	141
StepContent . . . . .	142
StepIcon . . . . .	143
StepLabel . . . . .	144
Stepper . . . . .	145
SvgIcon . . . . .	146
SwipeableDrawer . . . . .	147
SwipeableDrawer.triggerId . . . . .	149
Switch . . . . .	149
Tab . . . . .	151
TabContext . . . . .	152
Table . . . . .	154
TableBody . . . . .	155
TableCell . . . . .	155
TableContainer . . . . .	157
TableFooter . . . . .	157
TableHead . . . . .	158
TablePagination . . . . .	159

TableRow . . . . . 161  
 TableSortLabel . . . . . 162  
 TabList . . . . . 163  
 TabPanel . . . . . 164  
 Tabs . . . . . 165  
 TabScrollButton . . . . . 167  
 TextareaAutosize . . . . . 168  
 TextField . . . . . 169  
 ThemeProvider . . . . . 172  
 Timeline . . . . . 173  
 TimelineConnector . . . . . 174  
 TimelineContent . . . . . 174  
 TimelineDot . . . . . 175  
 TimelineItem . . . . . 176  
 TimelineOppositeContent . . . . . 177  
 TimelineSeparator . . . . . 177  
 ToggleButton . . . . . 178  
 ToggleButtonGroup . . . . . 179  
 Toolbar . . . . . 181  
 Tooltip . . . . . 182  
 Typography . . . . . 184  
 Zoom . . . . . 185

**Index** **187**

Accordion

*Accordion*

**Description**

<https://mui.com/material-ui/api/accordion/>

**Usage**

Accordion(...)

**Arguments**

... Props to pass to the component.

**Details**

- children node  
 Default is - The content of the component.
- classes object  
 Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.

- `defaultExpanded` `bool`  
Default is `FALSE` If true, expands the accordion by default.
- `disabled` `bool`  
Default is `FALSE` If true, the component is disabled.
- `disableGutters` `bool`  
Default is `FALSE` If true, it removes the margin between two expanded accordion items and the increase of height.
- `expanded` `bool`  
Default is - If true, expands the accordion, otherwise collapse it. Setting this prop enables control over the accordion.
- `onChange` `func`  
Default is - Callback fired when the expand/collapse state is changed. Signature: `function(event: React.SyntheticEvent, expanded: boolean) = void` event The event source of the callback. Warning: This is a generic event not a change event. `expanded` The expanded state of the accordion.
- `slotProps` `{ heading?: func | object, root?: func | object, transition?: func | object }`  
Default is The props used for each slot inside.
- `slots` `{ heading?: elementType, root?: elementType, transition?: elementType }`  
Default is The components used for each slot inside.
- `square` `bool`  
Default is `FALSE` If true, rounded corners are disabled.
- `sx` `Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `TransitionComponent` `elementType`  
Default is - The component used for the transition. Follow this guide to learn more about the requirements for this component. **Deprecated** Use `slots.transition` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `TransitionProps` `object`  
Default is - Props applied to the transition element. By default, the element is based on this Transition component. **Deprecated** Use `slotProps.transition` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

AccordionActions

*AccordionActions*

---

**Description**

<https://mui.com/material-ui/api/accordion-actions/>

**Usage**

AccordionActions(...)

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- disableSpacing bool  
Default is FALSE If true, the actions do not have additional margin.
- sx Array func| object| bool | func| object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

AccordionDetails	<i>AccordionDetails</i>
------------------	-------------------------

---

**Description**

<https://mui.com/material-ui/api/accordion-details/>

**Usage**

AccordionDetails(...)

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.

- `sx` Array `func| object| bool | func| object`  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

AccordionSummary	<i>AccordionSummary</i>
------------------	-------------------------

---

**Description**

<https://mui.com/material-ui/api/accordion-summary/>

**Usage**

AccordionSummary(...)

**Arguments**

... Props to pass to the component.

**Details**

- `children` node  
Default is - The content of the component.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `expandIcon` node  
Default is - The icon to display as the expand indicator.
- `focusVisibleClassName` string  
Default is - This prop can help identify which element has keyboard focus. The class name will be applied when the element gains the focus through keyboard interaction. It's a polyfill for the CSS `:focus-visible` selector. The rationale for using this feature is explained here. A polyfill can be used to apply a focus-visible class to other components if needed.
- `slotProps` { `content?: func| object`, `expandIconWrapper?: func| object`, `root?: func| object` }  
Default is The props used for each slot inside.
- `slots` { `content?: elementType`, `expandIconWrapper?: elementType`, `root?: elementType` }  
Default is The components used for each slot inside.
- `sx` Array `func| object| bool | func| object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

Alert

*Alert***Description**

<https://mui.com/material-ui/api/alert/>

**Usage**

```
Alert(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- **action node**  
Default is - The action to display. It renders after the message, at the end of the alert.
- **children node**  
Default is - The content of the component.
- **classes object**  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **closeText string**  
Default is 'Close' Override the default label for the close popup icon button. For localization purposes, you can use the provided translations.
- **color 'error' | 'info' | 'success' | 'warning' | string**  
Default is - The color of the component. Unless provided, the value is taken from the severity prop. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- **components { CloseButton?: elementType, CloseIcon?: elementType }**  
Default is The components used for each slot inside. Deprecated use the slots prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- **componentsProps { closeButton?: object, closeIcon?: object }**  
Default is The extra props for the slot components. You can override the existing props or add new ones. Deprecated use the slotProps prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- **icon node**  
Default is - Override the icon displayed before the children. Unless provided, the icon is mapped to the value of the severity prop. Set to false to remove the icon.
- **iconMapping { error?: node, info?: node, success?: node, warning?: node }**  
Default is - The component maps the severity prop to a range of different icons, for instance success to successoutlined. If you wish to change this mapping, you can provide your own. Alternatively, you can use the icon prop to override the icon displayed.

- `onClose` func  
Default is - Callback fired when the component requests to be closed. When provided and no `action` prop is set, a close icon button is displayed that triggers the callback when clicked. Signature: `function(event: React.SyntheticEvent) = void` event The event source of the callback.
- `role` string  
Default is 'alert' The ARIA role attribute of the element.
- `severity` 'error' | 'info' | 'success' | 'warning' | string  
Default is 'success' The severity of the alert. This defines the color and icon used.
- `slotProps` { `action?: func` | object, `closeButton?: func` | object, `closeIcon?: func` | object, `icon?: func` | object }  
Default is The props used for each slot inside.
- `slots` { `action?: elementType`, `closeButton?: elementType`, `closeIcon?: elementType`, `icon?: elementType` }  
Default is The components used for each slot inside.
- `sx` Array `func` | object | bool | `func` | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant` 'filled' | 'outlined' | 'standard' | string  
Default is 'standard' The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

AlertTitle

*AlertTitle*

---

**Description**

<https://mui.com/material-ui/api/alert-title/>

**Usage**

```
AlertTitle(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children` node  
Default is NA The content of the component.
- `classes` object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- `sx` Array `func` | object | bool | `func` | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 AppBar

*AppBar*


---

**Description**

<https://mui.com/material-ui/api/app-bar/>

**Usage**

```
AppBar(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children` node  
Default is - The content of the component.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` `'default' | 'inherit' | 'primary' | 'secondary' | 'transparent' | 'error' | 'info' | 'success' | 'warning'` | string  
Default is `'primary'` The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- `enableColorOnDark` bool  
Default is `FALSE` If true, the color prop is applied in dark mode.
- `position` `'absolute' | 'fixed' | 'relative' | 'static' | 'sticky'`  
Default is `'fixed'` The positioning type. The behavior of the different options is described in the MDN web docs. Note: `sticky` is not universally supported and will fall back to `static` when unavailable.
- `sx` Array `func | object | bool` | `func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

Autocomplete

*Autocomplete***Description**

<https://mui.com/material-ui/api/autocomplete/>

MUI's Autocomplete requires a `renderInput` function prop, which R cannot express directly. The R wrapper resolves the input in this order:

1. `renderInput` — pass an explicit `JS()` callback for full control.
2. A child element (e.g. `TextField(...)`, `OutlinedInput(...)`) — it is cloned and receives the params from MUI automatically. This is the recommended path: it stays close to the MUI API and survives upstream changes without touching R user code.
3. `inputProps` — a named list of props forwarded to a default `TextField`. Kept for backward compatibility with older examples.

**Usage**

```
Autocomplete(...)
```

```
Autocomplete.shinyInput(inputId, ..., value = NULL)
```

```
updateAutocomplete.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

<code>...</code>	Props to pass to the component.
<code>inputId</code>	ID of the component.
<code>value</code>	Starting value.
<code>session</code>	Object passed as the 'session' argument to Shiny server.

**Details**

- `options` array  
Default is - A list of options that will be shown in the Autocomplete.
- `renderInput` func  
Default is - `Render the input.Signature:function(params: object) = ReactNode`
- `autoComplete` bool  
Default is `FALSE` If true, the portion of the selected suggestion that the user hasn't typed, known as the completion string, appears inline after the input cursor in the textbox. The inline completion string is visually highlighted and has a selected state.

- `autoHighlight` `bool`  
Default is `FALSE` If true, the first option is automatically highlighted.
- `autoSelect` `bool`  
Default is `FALSE` If true, the selected option becomes the value of the input when the Autocomplete loses focus unless the user chooses a different option or changes the character string in the input. When using the `freeSolo` mode, the typed value will be the input value if the Autocomplete loses focus without highlighting an option.
- `blurOnSelect` `'mouse' | 'touch' | bool`  
Default is `FALSE` Control if the input should be blurred when an option is selected: `false` the input is not blurred. `true` the input is always blurred. `touch` the input is blurred after a touch event. `mouse` the input is blurred after a mouse event.
- `ChipProps` `object`  
Default is - Props applied to the Chip element. `Deprecated` Use `slotProps.chip` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `classes` `object`  
Default is - Override or extend the styles applied to the component. See [CSS classes API](#) below for more details.
- `clearIcon` `node`  
Default is `ClearIcon` `fontSize="small"`. The icon to display in place of the default clear icon.
- `clearOnBlur` `bool`  
Default is `!props.freeSolo` If true, the input's text is cleared on blur if no value is selected. Set it to true if you want to help the user enter a new value. Set it to false if you want to help the user resume their search.
- `clearOnEscape` `bool`  
Default is `FALSE` If true, clear all values when the user presses escape and the popup is closed.
- `clearText` `string`  
Default is `'Clear'` Override the default text for the clear icon button. For localization purposes, you can use the provided translations.
- `closeText` `string`  
Default is `'Close'` Override the default text for the close popup icon button. For localization purposes, you can use the provided translations.
- `componentsProps` `{ clearIndicator?: object, paper?: object, popper?: object, popupIndicator?: object`  
Default is - The props used for each slot inside. `Deprecated` Use the `slotProps` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `defaultValue` `any`  
Default is `props.multiple ? [] : null` The default value. Use when the component is not controlled.
- `disableClearable` `bool`  
Default is `FALSE` If true, the input can't be cleared.
- `disableCloseOnSelect` `bool`  
Default is `FALSE` If true, the popup won't close when a value is selected.
- `disabled` `bool`  
Default is `FALSE` If true, the component is disabled.

- `disabledItemsFocusable` `bool`  
Default is `FALSE` If true, will allow focus on disabled items.
- `disableListWrap` `bool`  
Default is `FALSE` If true, the list box in the popup will not wrap focus.
- `disablePortal` `bool`  
Default is `FALSE` If true, the Popper content will be under the DOM hierarchy of the parent component.
- `filterOptions` `func`  
Default is `createFilterOptions()` A function that determines the filtered options to be rendered on search. Signature: `function(options: Array, state: object) = Array<options>` The options to render. `state` The state of the component.
- `filterSelectedOptions` `bool`  
Default is `FALSE` If true, hide the selected options from the list box.
- `forcePopupIcon` `'auto' | bool`  
Default is `'auto'` Force the visibility display of the popup icon.
- `freeSolo` `bool`  
Default is `FALSE` If true, the Autocomplete is free solo, meaning that the user input is not bound to provided options.
- `fullWidth` `bool`  
Default is `FALSE` If true, the input will take up the full width of its container.
- `getLimitTagsText` `func`  
Default is `(more) = +$more` The label to display when the tags are truncated (`limitTags`). Signature: `function(more: number) = ReactNode` The number of truncated tags.
- `getOptionDisabled` `func`  
Default is - Used to determine the disabled state for a given option. Signature: `function(option: Value) = boolean` option The option to test.
- `getOptionKey` `func`  
Default is - Used to determine the key for a given option. This can be useful when the labels of options are not unique (since labels are used as keys by default). Signature: `function(option: Value) = string | number` option The option to get the key for.
- `getOptionLabel` `func`  
Default is `(option) = option.label ?? option` Used to determine the string value for a given option. It's used to fill the input (and the list box options if `renderOption` is not provided). If used in free solo mode, it must accept both the type of the options and a string. Signature: `function(option: Value) = string`
- `groupBy` `func`  
Default is - If provided, the options will be grouped under the returned string. The `groupBy` value is also used as the text for group headings when `renderGroup` is not provided. Signature: `function(option: Value) = string` option The Autocomplete option.
- `handleHomeEndKeys` `bool`  
Default is `!props.freeSolo` If true, the component handles the "Home" and "End" keys when the popup is open. It should move focus to the first option and last option, respectively.
- `id` `string`  
Default is - This prop is used to help implement the accessibility logic. If you don't provide an id it will fall back to a randomly generated one.

- `includeInputInList` `bool`  
Default is `FALSE` If true, the highlight can move to the input.
- `inputValue` `string`  
Default is - The input value.
- `isOptionEqualToValue` `func`  
Default is - Used to determine if the option represents the given value. Uses strict equality by default. Both arguments need to be handled, an option can only match with one value. Signature: `function(option: Value, value: Value) = boolean` option The option to test. value The value to test against.
- `limitTags` `integer`  
Default is `-1` The maximum number of tags that will be visible when not focused. Set `-1` to disable the limit.
- `ListboxComponent` `elementType`  
Default is `'ul'` The component used to render the listbox. **Deprecated** Use `slotProps.listbox.component` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `ListboxProps` `object`  
Default is - Props applied to the Listbox element. **Deprecated** Use `slotProps.listbox` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `loading` `bool`  
Default is `FALSE` If true, the component is in a loading state. This shows the `loadingText` in place of suggestions (only if there are no suggestions to show, for example options are empty).
- `loadingText` `node`  
Default is `'Loading80'` Text to display when in a loading state. For localization purposes, you can use the provided translations.
- `multiple` `bool`  
Default is `FALSE` If true, value must be an array and the menu will support multiple selections.
- `noOptionsText` `node`  
Default is `'No options'` Text to display when there are no options. For localization purposes, you can use the provided translations.
- `onChange` `func`  
Default is - Callback fired when the value changes. Signature: `function(event: React.SyntheticEvent, value: Value | Array, reason: string, details?: string) = void` event The event source of the callback. value The new value of the component. reason One of "createOption", "selectOption", "removeOption", "blur" or "clear".
- `onClose` `func`  
Default is - Callback fired when the popup requests to be closed. Use in controlled mode (see [open](#)). Signature: `function(event: React.SyntheticEvent, reason: string) = void` event The event source of the callback. reason Can be: "toggleInput", "escape", "selectOption", "removeOption", "blur".
- `onHighlightChange` `func`  
Default is - Callback fired when the highlight option changes. Signature: `function(event: React.SyntheticEvent, option: Value, reason: string) = void` event The event source of the callback. option The highlighted option. reason Can be: "keyboard", "mouse", "touch".

- **onInputChange** func  
Default is - Callback fired when the input value changes. Signature: `function(event: React.SyntheticEvent, value: string, reason: string) = void` event The event source of the callback. value The new value of the text input. reason Can be: "input" (user input), "reset" (programmatic change), "clear", "blur", "selectOption", "removeOption"
- **onOpen** func  
Default is - Callback fired when the popup requests to be opened. Use in controlled mode (see `open`). Signature: `function(event: React.SyntheticEvent) = void` event The event source of the callback.
- **open** bool  
Default is - If true, the component is shown.
- **openOnFocus** bool  
Default is FALSE If true, the popup will open on input focus.
- **openText** string  
Default is 'Open' Override the default text for the open popup icon button. For localization purposes, you can use the provided translations.
- **PaperComponent** elementType  
Default is Paper The component used to render the body of the popup. **Deprecated** Use `slots.paper` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- **PopperComponent** elementType  
Default is Popper The component used to position the popup. **Deprecated** Use `slots.popper` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- **popupIcon** node  
Default is ArrowDropDownIcon. The icon to display in place of the default popup icon.
- **readOnly** bool  
Default is FALSE If true, the component becomes readonly. It is also supported for multiple tags where the tag cannot be deleted.
- **renderGroup** func  
Default is - Render the group. Signature: `function(params: AutocompleteRenderGroupParams) = ReactNode` params The group to render.
- **renderOption** func  
Default is - Render the option, use `getOptionLabel` by default. Signature: `function(props: object, option: Value, state: object, ownerState: object) = ReactNode` props The props to apply on the li element. option The option to render. state The state of each option. ownerState The state of the Autocomplete component.
- **renderTags** func  
Default is - Render the selected value. Signature: `function(value: Array, getTagProps: function, ownerState: object) = ReactNode` value The value provided to the component. `getTagProps` A tag props getter. ownerState The state of the Autocomplete component.
- **selectOnFocus** bool  
Default is `!props.freeSolo` If true, the input's text is selected on focus. It helps the user clear the selected value.

- `size` `'small' | 'medium' | string`  
Default is `'medium'` The size of the component.
- `slotProps` `{ chip?: func | object, clearIndicator?: func | object, listBox?: func | object, paper?: func | object }`  
Default is The props used for each slot inside.
- `slots` `{ listBox?: elementType, paper?: elementType, popper?: elementType }`  
Default is The components used for each slot inside.
- `sx` `Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `value` `any`  
Default is - The value of the autocomplete. The value must have reference equality with the option in order to be selected. You can customize the equality behavior with the `isOptionEqualToValue` prop.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

### Examples

```
library(shiny)
library(muiMaterial)

ui <- muiMaterialPage(
  CssBaseline(),
  Autocomplete.shinyInput(
    inputId = "auto",
    options = c("Apple", "Banana", "Cherry"),
    TextField(label = "Fruit")
  ),
  verbatimTextOutput("out")
)

server <- function(input, output, session) {
  output$out <- renderPrint(input$auto)
}

shinyApp(ui, server)
```

### Description

<https://mui.com/material-ui/api/avatar/>

**Usage**

```
Avatar(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `alt` string  
Default is - Used in combination with `src` or `srcSet` to provide an alt attribute for the rendered `img` element.
- `children` node  
Default is - Used to render icon or text elements inside the Avatar if `src` is not set. This can be an element, or just a string.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component` elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `imgProps` object  
Default is - Attributes applied to the `img` element if the component is used to display an image. It can be used to listen for the loading error event. **Deprecated** Use `slotProps.img` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `sizes` string  
Default is - The sizes attribute for the `img` element.
- `slotProps` { `img?: func | object` }
- `slots` { `img?: elementType` }
- `src` string  
Default is - The `src` attribute for the `img` element.
- `srcSet` string  
Default is - The `srcSet` attribute for the `img` element. Use this attribute for responsive image display.
- `sx` Array `func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant` 'circular' | 'rounded' | 'square' | string  
Default is 'circular' The shape of the avatar.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

AvatarGroup

*AvatarGroup***Description**

<https://mui.com/material-ui/api/avatar-group/>

**Usage**

```
AvatarGroup(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- children node  
Default is - The avatars to stack.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- componentsProps { additionalAvatar?: object }  
Default is - The extra props for the slot components. You can override the existing props or add new ones. This prop is an alias for the slotProps prop. **Deprecated** use the slotProps prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- max number  
Default is 5 Max avatars to show before +x.
- renderSurplus func  
Default is - custom renderer of extraAvatars  
Signature: function(surplus: number) = React.ReactNode  
Returns: custom element to display
- slotProps { additionalAvatar?: object, surplus?: func | object }  
Default is The props used for each slot inside.
- slots { surplus?: elementType }  
Default is The components used for each slot inside.
- spacing 'medium' | 'small' | number  
Default is 'medium' Spacing between avatars.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the [sx](#) page for more details.

- `total number`  
Default is `children.length` The total number of avatars. Used for calculating the number of extra avatars.
- `variant 'circular' | 'rounded' | 'square' | string`  
Default is `'circular'` The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Backdrop

*Backdrop*

---

**Description**

<https://mui.com/material-ui/api/backdrop/>

**Usage**

`Backdrop(...)`

**Arguments**

`...` Props to pass to the component.

**Details**

- `open bool`  
Default is - If true, the component is shown.
- `children node`  
Default is - The content of the component.
- `classes object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `components { Root?: elementType }`  
Default is The components used for each slot inside. **Deprecated** Use the `slots` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `componentsProps { root?: object }`  
Default is The extra props for the slot components. You can override the existing props or add new ones. **Deprecated** Use the `slotProps` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.

- `invisible` `bool`  
Default is `FALSE` If true, the backdrop is invisible. It can be used when rendering a popover or a custom select component.
- `slotProps` `{ root?: func| object, transition?: func| object }`  
Default is The props used for each slot inside.
- `slots` `{ root?: elementType, transition?: elementType }`  
Default is The components used for each slot inside.
- `sx` `Array func| object| bool | func| object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `TransitionComponent` `elementType`  
Default is `Fade` The component used for the transition. Follow this guide to learn more about the requirements for this component. **Deprecated** Use `slots.transition` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `transitionDuration` `number| { appear?: number, enter?: number, exit?: number }`  
Default is - The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Badge

*Badge*

---

**Description**

<https://mui.com/material-ui/api/badge/>

**Usage**

```
Badge(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- `anchorOrigin` `{ horizontal?: 'left'| 'right', vertical?: 'bottom'| 'top' }`  
Default is `vertical: 'top', horizontal: 'right'`, The anchor of the badge.
- `badgeContent` `node`  
Default is - The content rendered within the badge.
- `children` `node`  
Default is - The badge will be added relative to this node.

- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` `'default' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string`  
Default is `'default'` The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `components` `{ badge?: elementType, root?: elementType }`  
Default is The components used for each slot inside. *Deprecated* use the `slots` prop instead. This prop will be removed in a future major release. See *Migrating from deprecated APIs* for more details.
- `componentsProps` `{ badge?: func | object, root?: func | object }`  
Default is The extra props for the slot components. You can override the existing props or add new ones. *Deprecated* use the `slotProps` prop instead. This prop will be removed in a future major release. See *Migrating from deprecated APIs* for more details.
- `invisible` `bool`  
Default is `FALSE` If true, the badge is invisible.
- `max` `number`  
Default is `99` Max count to show.
- `overlap` `'circular' | 'rectangular'`  
Default is `'rectangular'` Wrapped shape the badge should overlap.
- `showZero` `bool`  
Default is `FALSE` Controls whether the badge is hidden when `badgeContent` is zero.
- `slotProps` `{ badge?: func | object, root?: func | object }`  
Default is The props used for each slot inside the Badge.
- `slots` `{ badge?: elementType, root?: elementType }`  
Default is The components used for each slot inside the Badge. Either a string to use a HTML element or a component.
- `sx` `Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant` `'dot' | 'standard' | string`  
Default is `'standard'` The variant to use.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

BottomNavigation      *BottomNavigation*

---

## Description

<https://mui.com/material-ui/api/bottom-navigation/>

## Usage

```
BottomNavigation(...)
```

```
BottomNavigation.shinyInput(inputId, ..., value = defaultValue)
```

```
updateBottomNavigation.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

## Arguments

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

## Details

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- onChange func  
Default is - Callback fired when the value changes. Signature: function(event: React.SyntheticEvent, value: any) = void event The event source of the callback. Warning: This is a generic event not a change event. value We default to the index of the child.
- showLabels bool  
Default is FALSE If true, all BottomNavigationActions will show their labels. By default, only the selected BottomNavigationAction will show its label.

- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `value` any  
Default is - The value of the currently selected `BottomNavigationAction`.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

**Note**

Give each child `BottomNavigationAction` a `value`; the selected action's `value` is reported to `input[[inputId]]`. Pass an initial value matching one of them to pre-select it; when omitted, the component mounts with nothing selected (`value = FALSE`).

---

`BottomNavigationAction`

*BottomNavigationAction*

---

**Description**

<https://mui.com/material-ui/api/bottom-navigation-action/>

**Usage**

```
BottomNavigationAction(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children` unsupportedProp  
Default is NA This prop isn't supported. Use the component prop if you need to change the children structure.
- `classes` object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- `icon` node  
Default is NA The icon to display.
- `label` node  
Default is NA The label element.

- `showLabel` `bool`  
Default is `NA` If true, the `BottomNavigationAction` will show its label. By default, only the selected `BottomNavigationAction` inside `BottomNavigation` will show its label. The prop defaults to the value (`false`) inherited from the parent `BottomNavigation` component.
- `sx` `Array` `func` | `object` | `bool` | `func` | `object`  
Default is `NA` The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `value` `any`  
Default is `NA` You can provide your own value. Otherwise, we fallback to the child position index.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Box

*Box*

---

**Description**

<https://mui.com/material-ui/api/box/>

**Usage**

`Box(...)`

**Arguments**

`...` Props to pass to the component.

**Details**

- `component` `elementType`  
Default is `NA` The component used for the root node. Either a string to use a HTML element or a component.
- `sx` `Array` `func` | `object` | `bool` | `func` | `object`  
Default is `NA` The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

Breadcrumbs

*Breadcrumbs***Description**

<https://mui.com/material-ui/api/breadcrumbs/>

**Usage**

```
Breadcrumbs(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- expandText string  
Default is 'Show path' Override the default label for the expand button. For localization purposes, you can use the provided translations.
- itemsAfterCollapse integer  
Default is 1 If max items is exceeded, the number of items to show after the ellipsis.
- itemsBeforeCollapse integer  
Default is 1 If max items is exceeded, the number of items to show before the ellipsis.
- maxItems integer  
Default is 8 Specifies the maximum number of breadcrumbs to display. When there are more than the maximum number, only the first itemsBeforeCollapse and last itemsAfterCollapse will be shown, with an ellipsis in between.
- separator node  
Default is '/' Custom separator node.
- slotProps { collapsedIcon?: func | object }  
Default is The props used for each slot inside the Breadcrumbs.
- slots { CollapsedIcon?: elementType }  
Default is The components used for each slot inside the Breadcrumbs. Either a string to use a HTML element or a component.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 Button

*Button*


---

**Description**

<https://mui.com/material-ui/api/button/>

**Usage**

```
Button(...)
```

```
Button.shinyInput(inputId, ...)
```

```
updateButton.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

<code>...</code>	Props to pass to the component.
<code>inputId</code>	ID of the component.
<code>session</code>	Object passed as the ‘session’ argument to Shiny server.

**Details**

- `children node`  
Default is - The content of the component.
- `classes object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color 'inherit' | 'primary' | 'secondary' | 'success' | 'error' | 'info' | 'warning' | string`  
Default is ‘primary’ The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- `component elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `disabled bool`  
Default is FALSE If true, the component is disabled.

- `disableElevation` `bool`  
Default is `FALSE` If true, no elevation is used.
- `disableFocusRipple` `bool`  
Default is `FALSE` If true, the keyboard focus ripple is disabled.
- `disableRipple` `bool`  
Default is `FALSE` If true, the ripple effect is disabled. Without a ripple there is no styling for `:focus-visible` by default. Be sure to highlight the element by applying separate styles with the `.Mui-focusVisible` class.
- `endIcon` `node`  
Default is - Element placed after the children.
- `fullWidth` `bool`  
Default is `FALSE` If true, the button will take up the full width of its container.
- `href` `string`  
Default is - The URL to link to when the button is clicked. If defined, an a element will be used as the root node.
- `loading` `bool`  
Default is `null` If true, the loading indicator is visible and the button is disabled. If true | false, the loading wrapper is always rendered before the children to prevent Google Translation Crash.
- `loadingIndicator` `node`  
Default is `CircularProgress color="inherit" size=16` / Element placed before the children if the button is in loading state. The node should contain an element with `role="progressbar"` with an accessible name. By default, it renders a `CircularProgress` that is labeled by the button itself.
- `loadingPosition` `'center' | 'end' | 'start'`  
Default is `'center'` The loading indicator can be positioned on the start, end, or the center of the button.
- `size` `'small' | 'medium' | 'large' | string`  
Default is `'medium'` The size of the component. `small` is equivalent to the dense button styling.
- `startIcon` `node`  
Default is - Element placed before the children.
- `sx` `Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant` `'contained' | 'outlined' | 'text' | string`  
Default is `'text'` The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

**Examples**

```
library(shiny)
library(muiMaterial)
```

```
ui <- muiMaterialPage(  
  Button.shinyInput("btn", "Click me", variant = "contained"),  
  verbatimTextOutput("count")  
)  
  
server <- function(input, output, session) {  
  output$count <- renderPrint(input$btn)  
}  
  
shinyApp(ui, server)
```

---

ButtonBase

*ButtonBase*

---

## Description

<https://mui.com/material-ui/api/button-base/>

## Usage

```
ButtonBase(...)
```

## Arguments

... Props to pass to the component.

## Details

- `action ref`  
Default is - A ref for imperative actions. It currently only supports `focusVisible()` action.
- `centerRipple` bool  
Default is `FALSE` If true, the ripples are centered. They won't start at the cursor interaction position.
- `children` node  
Default is - The content of the component.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component element` type  
Default is - The component used for the root node. Either a string to use a HTML element or a component. This needs to be able to hold a ref.
- `disabled` bool  
Default is `FALSE` If true, the component is disabled.

- `disableRipple` `bool`  
Default is `FALSE` If true, the ripple effect is disabled. Without a ripple there is no styling for `:focus-visible` by default. Be sure to highlight the element by applying separate styles with the `.Mui-focusVisible` class.
- `disableTouchRipple` `bool`  
Default is `FALSE` If true, the touch ripple effect is disabled.
- `focusRipple` `bool`  
Default is `FALSE` If true, the base button will have a keyboard focus ripple.
- `focusVisibleClassName` `string`  
Default is - This prop can help identify which element has keyboard focus. The class name will be applied when the element gains the focus through keyboard interaction. It's a polyfill for the CSS `:focus-visible` selector. The rationale for using this feature is explained here. A polyfill can be used to apply a focus-visible class to other components if needed.
- `LinkComponent` `elementType`  
Default is `'a'` The component used to render a link when the `href` prop is provided.
- `onFocusVisible` `func`  
Default is - Callback fired when the component is focused with a keyboard. We trigger a `onFocus` callback too.
- `sx` `Array` `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `TouchRippleProps` `object`  
Default is - Props applied to the `TouchRipple` element.
- `touchRippleRef` `func` | `{ current?: { pulsate: func, start: func, stop: func } }`  
Default is - A ref that points to the `TouchRipple` element.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

ButtonGroup

*ButtonGroup*

---

**Description**

<https://mui.com/material-ui/api/button-group/>

**Usage**

ButtonGroup(...)

**Arguments**

...                    Props to pass to the component.

## Details

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- color 'inherit' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'primary' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- disabled bool  
Default is FALSE If true, the component is disabled.
- disableElevation bool  
Default is FALSE If true, no elevation is used.
- disableFocusRipple bool  
Default is FALSE If true, the button keyboard focus ripple is disabled.
- disableRipple bool  
Default is FALSE If true, the button ripple effect is disabled.
- fullWidth bool  
Default is FALSE If true, the buttons will take up the full width of its container.
- orientation 'horizontal' | 'vertical'  
Default is 'horizontal' The component orientation (layout flow direction).
- size 'small' | 'medium' | 'large' | string  
Default is 'medium' The size of the component. small is equivalent to the dense button styling.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- variant 'contained' | 'outlined' | 'text' | string  
Default is 'outlined' The variant to use.

## Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

Card

*Card*

---

**Description**

<https://mui.com/material-ui/api/card/>

**Usage**

```
Card(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- raised bool  
Default is FALSE If true, the card will use raised styling.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

CardActionArea

*CardActionArea*

---

**Description**

<https://mui.com/material-ui/api/card-action-area/>

**Usage**

```
CardActionArea(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func| object| bool | func| object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny . tag class suitable for use in the UI of a Shiny app.

---

CardActions

*CardActions*

---

**Description**

<https://mui.com/material-ui/api/card-actions/>

**Usage**

CardActions(...)

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- disableSpacing bool  
Default is FALSE If true, the actions do not have additional margin.
- sx Array func| object| bool | func| object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny . tag class suitable for use in the UI of a Shiny app.

---

 CardContent

*CardContent*


---

### Description

<https://mui.com/material-ui/api/card-content/>

### Usage

```
CardContent(...)
```

### Arguments

... Props to pass to the component.

### Details

- children node  
Default is NA The content of the component.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is NA The component used for the root node. Either a string to use a HTML element or a component.
- sx Array func | object | bool | func | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

### Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

 CardHeader

*CardHeader*


---

### Description

<https://mui.com/material-ui/api/card-header/>

### Usage

```
CardHeader(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- action node  
Default is - The action to display in the card header.
- avatar node  
Default is - The Avatar element to display.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- disableTypography bool  
Default is FALSE If true, subheader and title won't be wrapped by a Typography component. This can be useful to render an alternative Typography variant by wrapping the title text, and optional subheader text with the Typography component.
- slotProps { action?: func| object, avatar?: func| object, content?: func| object, root?: func| object }  
Default is The props used for each slot inside.
- slots { action?: elementType, avatar?: elementType, content?: elementType, root?: elementType, subheader?: elementType }  
Default is The components used for each slot inside.
- subheader node  
Default is - The content of the component.
- subheaderTypographyProps object  
Default is - These props will be forwarded to the subheader (as long as disableTypography is not true). Deprecated Use slotProps.subheader instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- sx Array func| object| bool | func| object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- title node  
Default is - The content of the component.
- titleTypographyProps object  
Default is - These props will be forwarded to the title (as long as disableTypography is not true). Deprecated Use slotProps.title instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

CardMedia

*CardMedia*

---

## Description

<https://mui.com/material-ui/api/card-media/>

## Usage

```
CardMedia(...)
```

## Arguments

... Props to pass to the component.

## Details

- `children` node  
Default is NA The content of the component.
- `classes` object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component` elementType  
Default is NA The component used for the root node. Either a string to use a HTML element or a component.
- `image` string  
Default is NA Image to be displayed as a background image. Either `image` or `src` prop must be specified. Note that caller must specify height otherwise the image will not be visible.
- `src` string  
Default is NA An alias for image property. Available only with media components. Media components: video, audio, picture, iframe, img.
- `sx` Array func | object | bool | func | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

## Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 Checkbox

*Checkbox*


---

**Description**

<https://mui.com/material-ui/api/checkbox/>

**Usage**

```
Checkbox(...)

Checkbox.shinyInput(inputId, ..., value = defaultValue)

updateCheckbox.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- checked bool  
Default is - If true, the component is checked.
- checkedIcon node  
Default is CheckBoxIcon / The icon to display when the component is checked.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- color 'default' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'primary' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- defaultChecked bool  
Default is - The default checked state. Use when the component is not controlled.
- disabled bool  
Default is FALSE If true, the component is disabled.
- disableRipple bool  
Default is FALSE If true, the ripple effect is disabled.

- `icon` node  
Default is `CheckBoxOutlineBlankIcon` / The icon to display when the component is unchecked.
- `id` string  
Default is - The id of the input element.
- `indeterminate` bool  
Default is `FALSE` If true, the component appears indeterminate. This does not set the native input element to indeterminate due to inconsistent behavior across browsers. However, we set a `data-indeterminate` attribute on the input.
- `indeterminateIcon` node  
Default is `IndeterminateCheckBoxIcon` / The icon to display when the component is indeterminate.
- `inputProps` object  
Default is - Attributes applied to the input element. `Deprecated` Use `slotProps.input` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `onChange` func  
Default is - Callback fired when the state is changed. Signature: `function(event: React.ChangeEvent) = void` event The event source of the callback. You can pull out the new checked state by accessing `event.target.checked` (boolean).
- `required` bool  
Default is `FALSE` If true, the input element is required.
- `size` `'medium' | 'small' | string`  
Default is `'medium'` The size of the component. `small` is equivalent to the dense checkbox styling.
- `slotProps` { `input?: func | object`, `root?: func | object` }  
Default is The props used for each slot inside.
- `slots` { `input?: elementType`, `root?: elementType` }  
Default is The components used for each slot inside.
- `sx` `Array` `func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `value` any  
Default is - The value of the component. The DOM API casts this to a string. The browser uses "on" as the default value.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Chip

*Chip*

---

## Description

<https://mui.com/material-ui/api/chip/>

## Usage

```
Chip(...)
```

## Arguments

... Props to pass to the component.

## Details

- avatar element  
Default is - The Avatar element to display.
- children unsupportedProp  
Default is - This prop isn't supported. Use the component prop if you need to change the children structure.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- clickable bool  
Default is - If true, the chip will appear clickable, and will raise when pressed, even if the onClick prop is not defined. If false, the chip will not appear clickable, even if onClick prop is defined. This can be used, for example, along with the component prop to indicate an anchor Chip is clickable. Note: this controls the UI and does not affect the onClick event.
- color 'default' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'default' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- deleteIcon element  
Default is - Override the default delete icon element. Shown only if onDelete is set.
- disabled bool  
Default is FALSE If true, the component is disabled.
- icon element  
Default is - Icon element.
- label node  
Default is - The content of the component.

- `onDelete` func  
Default is - Callback fired when the delete icon is clicked. If set, the delete icon will be shown.
- `size` 'medium' | 'small' | string  
Default is 'medium' The size of the component.
- `skipFocusWhenDisabled` bool  
Default is FALSE If true, allows the disabled chip to escape focus. If false, allows the disabled chip to receive focus.
- `sx` Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- `variant` 'filled' | 'outlined' | string  
Default is 'filled' The variant to use.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

CircularProgress	<i>CircularProgress</i>
------------------	-------------------------

---

### Description

<https://mui.com/material-ui/api/circular-progress/>

### Usage

```
CircularProgress(...)
```

### Arguments

... Props to pass to the component.

### Details

- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` 'inherit' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'primary' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- `disableShrink` bool  
Default is FALSE If true, the shrink animation is disabled. This only works if variant is indeterminate.

- `size` `number | string`  
Default is 40 The size of the component. If using a number, the pixel unit is assumed. If using a string, you need to provide the CSS unit, for example '3rem'.
- `sx` `Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `thickness` `number`  
Default is 3.6 The thickness of the circle.
- `value` `number`  
Default is 0 The value of the progress indicator for the determinate variant. Value between 0 and 100.
- `variant` `'determinate' | 'indeterminate'`  
Default is 'indeterminate' The variant to use. Use `indeterminate` when there is no progress value.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

ClickAwayListener      *ClickAwayListener*

---

**Description**

<https://mui.com/material-ui/api/click-away-listener/>

**Usage**

```
ClickAwayListener(...)
```

**Arguments**

...      Props to pass to the component.

**Details**

- `children` `element`  
Default is - The wrapped element. This needs to be able to hold a ref.
- `onClickAway` `func`  
Default is - Callback fired when a "click away" event is detected.
- `disableReactTree` `bool`  
Default is FALSE If true, the React tree is ignored and only the DOM tree is considered. This prop changes how portaled elements are handled.

- `mouseEvent` `'onClick' | 'onMouseDown' | 'onMouseUp' | 'onPointerDown' | 'onPointerUp' | false`  
Default is `'onClick'` The mouse event to listen to. You can disable the listener by providing `false`.
- `touchEvent` `'onTouchEnd' | 'onTouchStart' | false`  
Default is `'onTouchEnd'` The touch event to listen to. You can disable the listener by providing `false`.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Collapse

*Collapse*

---

**Description**

<https://mui.com/material-ui/api/collapse/>

**Usage**

`Collapse(...)`

**Arguments**

`...` Props to pass to the component.

**Details**

- `addEndListener` `func`  
Default is - Add a custom transition end trigger. Called with the transitioning DOM node and a done callback. Allows for more fine grained transition end logic. Note: Timeouts are still used as a fallback if provided.
- `children` `node`  
Default is - The content node to be collapsed.
- `classes` `object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `collapsedSize` `number | string`  
Default is `'0px'` The width (horizontal) or height (vertical) of the container when collapsed.
- `componentElement` `type`  
Default is - The component used for the root node. Either a string to use a HTML element or a component. This needs to be able to hold a ref.
- `easing` `{ enter?: string, exit?: string } | string`  
Default is - The transition timing function. You may specify a single easing or a object containing enter and exit values.

- `in` `bool`  
Default is - If true, the component will transition in.
- `orientation` `'horizontal' | 'vertical'`  
Default is `'vertical'` The transition orientation.
- `sx` `Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `timeout` `'auto' | number | { appear?: number, enter?: number, exit?: number }`  
Default is `duration.standard` The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object. Set to `'auto'` to automatically calculate transition time based on height.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Container	<i>Container</i>
-----------	------------------

---

**Description**

<https://mui.com/material-ui/api/container/>

**Usage**

```
Container(...)
```

**Arguments**

`...` Props to pass to the component.

**Details**

- `classes` `object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `disableGutters` `bool`  
Default is `FALSE` If true, the left and right padding is removed.
- `fixed` `bool`  
Default is `FALSE` Set the max-width to match the min-width of the current breakpoint. This is useful if you'd prefer to design for a fixed set of sizes instead of trying to accommodate a fully fluid viewport. It's fluid by default.

- `maxWidth` 'xs' | 'sm' | 'md' | 'lg' | 'xl' | `false` | `string`  
Default is 'lg' Determine the max-width of the container. The container width grows with the size of the screen. Set to false to disable `maxWidth`.
- `sx` `Array` `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

CssBaseline

*CssBaseline*

---

**Description**

<https://mui.com/material-ui/api/css-baseline/>

**Usage**

`CssBaseline(...)`

**Arguments**

... Props to pass to the component.

**Details**

- `children` `node`  
Default is - You can wrap a node.
- `enableColorScheme` `bool`  
Default is `FALSE` Enable color-scheme CSS property to use `theme.palette.mode`. For more details, check out <https://developer.mozilla.org/en-US/docs/Web/CSS/color-scheme> For browser support, check out <https://caniuse.com/?search=color-scheme>

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 Dialog

*Dialog*


---

## Description

<https://mui.com/material-ui/api/dialog/>

## Usage

```
Dialog(...)
```

```
Dialog.shinyInput(inputId, ...)
```

```
updateDialog.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

## Arguments

...	Props to pass to the component.
inputId	ID of the component.
session	Object passed as the 'session' argument to Shiny server.

## Details

- `open` bool  
Default is - If true, the component is shown.
- `aria-describedby` string  
Default is - The id(s) of the element(s) that describe the dialog.
- `aria-labelledby` string  
Default is - The id(s) of the element(s) that label the dialog.
- `aria-modal` 'false' | 'true' | bool  
Default is TRUE Informs assistive technologies that the element is modal. It's added on the element with role="dialog".
- `BackdropComponent` elementType  
Default is styled(Backdrop, name: 'MuiModal', slot: 'Backdrop', overridesResolver: (props, styles) = return styles.backdrop; , )( zIndex: -1, ) A backdrop component. This prop enables custom backdrop rendering. **Deprecated** Use slots.backdrop instead. While this prop currently works, it will be removed in the next major version.
- `children` node  
Default is - Dialog children, usually the included sub-components.

- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `disableEscapeKeyDown` bool  
Default is FALSE If true, hitting escape will not fire the `onClose` callback.
- `fullScreen` bool  
Default is FALSE If true, the dialog is full-screen.
- `fullWidth` bool  
Default is FALSE If true, the dialog stretches to `maxWidth`. Notice that the dialog width grow is limited by the default margin.
- `maxWidth` `'xs' | 'sm' | 'md' | 'lg' | 'xl' | false | string`  
Default is `'sm'` Determine the max-width of the dialog. The dialog width grows with the size of the screen. Set to `false` to disable `maxWidth`.
- `onClose` func  
Default is - Callback fired when the component requests to be closed. Signature: `function(event: object, reason: string) = void` event The event source of the callback. reason Can be: `"escapeKeyDown"`, `"backdropClick"`.
- `PaperComponent` elementType  
Default is `Paper` The component used to render the body of the dialog.
- `PaperProps` object  
Default is `Props` applied to the `Paper` element. **Deprecated** Use `slotProps.paper` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `scroll` `'body' | 'paper'`  
Default is `'paper'` Determine the container for scrolling the dialog.
- `slotProps` { `backdrop?: func | object`, `container?: func | object`, `paper?: func | object`, `root?: func | object` }  
Default is The props used for each slot inside.
- `slots` { `backdrop?: elementType`, `container?: elementType`, `paper?: elementType`, `root?: elementType`, `tr` }  
Default is The components used for each slot inside.
- `sx` `Array` `func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `TransitionComponent` elementType  
Default is `Fade` The component used for the transition. Follow this guide to learn more about the requirements for this component. **Deprecated** Use `slots.transition` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `transitionDuration` `number | { appear?: number, enter?: number, exit?: number }`  
Default is `enter: theme.transitions.duration.enteringScreen, exit: theme.transitions.duration.leavingScreen`, The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object.
- `TransitionProps` object  
Default is - Props applied to the transition element. By default, the element is based on this `Transition` component. **Deprecated** Use `slotProps.transition` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

**Note**

This is an overlay surface wired as a click-reporter: `input[[inputId]]` holds a click count (incremented on every click inside the surface), and the wrapper does **not** manage the open state. Render it with `open = TRUE/FALSE` and toggle visibility from the server with `updateDialog.shinyInput(session, inputId, open = TRUE)`. For the common "open on click of a button" pattern, [Dialog.triggerId](#) is simpler (open/close handled entirely client-side, no server logic).

---

<code>Dialog.triggerId</code>	<i>Dialog.triggerId</i>
-------------------------------	-------------------------

---

**Description**

Custom Dialog bound to a DOM element by id. See `'js/src/MuiDialogTriggerId.jsx'`. Open/close state is managed entirely client-side, so this works in Shiny apps, Quarto documents, and static HTML without server logic.

**Usage**

```
Dialog.triggerId(triggerId, ...)
```

**Arguments**

<code>triggerId</code>	HTML id of an existing DOM element that acts as the trigger to open the Dialog.
<code>...</code>	Named arguments forwarded as React props, plus children to render inside the component.

**Value**

Object with `'shiny.tag'` class suitable for use in the UI of a Shiny app.

**Examples**

```
library(shiny)
library(muiMaterial)

ui <- muiMaterialPage(
  Button(id = "openDialog", "Open dialog"),
  Dialog.triggerId(
    "openDialog",
    DialogTitle("Hello"),
    DialogContent("Open/close managed entirely client-side.")
  )
)
```

```
shinyApp(ui, function(input, output, session) {})
```

---

DialogActions

*DialogActions*

---

### Description

<https://mui.com/material-ui/api/dialog-actions/>

### Usage

```
DialogActions(...)
```

### Arguments

... Props to pass to the component.

### Details

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- disableSpacing bool  
Default is FALSE If true, the actions do not have additional margin.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

### Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

DialogContent	<i>DialogContent</i>
---------------	----------------------

---

**Description**

<https://mui.com/material-ui/api/dialog-content/>

**Usage**

```
DialogContent(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- dividers bool  
Default is FALSE Display the top and bottom dividers.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

DialogContentText	<i>DialogContentText</i>
-------------------	--------------------------

---

**Description**

<https://mui.com/material-ui/api/dialog-content-text/>

**Usage**

```
DialogContentText(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func| object| bool | func| object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

DialogTitle

*DialogTitle*

---

**Description**

<https://mui.com/material-ui/api/dialog-title/>

**Usage**

DialogTitle(...)

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func| object| bool | func| object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

 Divider

*Divider*


---

**Description**

<https://mui.com/material-ui/api/divider/>

**Usage**

```
Divider(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- **absolute** bool  
Default is FALSE Absolutely position the element.
- **children** node  
Default is - The content of the component.
- **classes** object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **component** elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- **flexItem** bool  
Default is FALSE If true, a vertical divider will have the correct height when used in flex container. (By default, a vertical divider will have a calculated height of 0px if it is the child of a flex container.)
- **light** bool  
Default is FALSE If true, the divider will have a lighter color. **Deprecated** Use `Divider sx={opacity: 0.6}` / (or any opacity or color) instead. See [Migrating from deprecated APIs](#) for more details.
- **orientation** 'horizontal' | 'vertical'  
Default is 'horizontal' The component orientation.
- **sx** Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the [sx](#) page for more details.
- **textAlign** 'center' | 'left' | 'right'  
Default is 'center' The text alignment.
- **variant** 'fullWidth' | 'inset' | 'middle' | string  
Default is 'fullWidth' The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Drawer

*Drawer*

---

**Description**

<https://mui.com/material-ui/api/drawer/>

**Usage**

```
Drawer(...)
```

```
Drawer.shinyInput(inputId, ...)
```

```
updateDrawer.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

<code>...</code>	Props to pass to the component.
<code>inputId</code>	ID of the component.
<code>session</code>	Object passed as the 'session' argument to Shiny server.

**Details**

- `anchor` 'bottom' | 'left' | 'right' | 'top'  
Default is 'left' Side from which the drawer will appear.
- `children` node  
Default is - The content of the component.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `elevation` integer  
Default is 16 The elevation of the drawer.
- `hideBackdrop` bool  
Default is FALSE If true, the backdrop is not rendered.
- `ModalProps` object  
Default is Props applied to the Modal element.

- `onClose` func  
Default is - Callback fired when the component requests to be closed. The reason parameter can optionally be used to control the response to `onClose`. Signature: `function(event: object, reason: string) = void` event The event source of the callback. Can be: "escapeKeyDown", "backdropClick".
- `open` bool  
Default is FALSE If true, the component is shown.
- `PaperProps` object  
Default is Props applied to the Paper element. Deprecated use the `slotProps.paper` prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- `SlideProps` object  
Default is - Props applied to the Slide element. Deprecated use the `slotProps.transition` prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- `slotProps` { `backdrop?: func| object`, `docked?: func| object`, `paper?: func| object`, `root?: func| object` }  
Default is The props used for each slot inside.
- `slots` { `backdrop?: elementType`, `docked?: elementType`, `paper?: elementType`, `root?: elementType`, `transition?: elementType` }  
Default is The components used for each slot inside.
- `sx` Array `func| object| bool | func| object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `transitionDuration` number | { `appear?: number`, `enter?: number`, `exit?: number` }  
Default is `enter: theme.transitions.duration.enteringScreen`, `exit: theme.transitions.duration.leavingScreen`,  
The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object.
- `variant` 'permanent' | 'persistent' | 'temporary'  
Default is 'temporary' The variant to use.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

### Note

This is an overlay surface wired as a click-reporter: `input[[inputId]]` holds a click count, and the wrapper does **not** manage the open state. Render it with `open = TRUE/FALSE` and toggle visibility from the server with `updateDrawer.shinyInput(session, inputId, open = TRUE)`. For the common "open on click of a button" pattern, `Drawer.triggerId` is simpler (open/close handled entirely client-side, no server logic).

---

<code>Drawer.triggerId</code>	<i>Drawer.triggerId</i>
-------------------------------	-------------------------

---

### Description

Custom Drawer bound to a DOM element by id. See `'js/src/MuiDrawerTriggerId.jsx'`.

### Usage

```
Drawer.triggerId(triggerId, ...)
```

### Arguments

<code>triggerId</code>	HTML id of an existing DOM element that acts as the trigger (button, link, etc.) to open the Drawer.
<code>...</code>	Named arguments forwarded as React props, plus children to render inside the component. Pass <code>closeOnLinkClick = FALSE</code> to keep the Drawer open when any link inside it is clicked. The default (TRUE) closes the Drawer on any <code>&lt;a&gt;</code> click, including external links with <code>target = "_blank"</code> .

### Value

Object with `'shiny.tag'` class suitable for use in the UI of a Shiny app.

### Examples

```
library(shiny)
library(muiMaterial)

ui <- muiMaterialPage(
  Button(id = "openDrawer", "Open drawer"),
  Drawer.triggerId("openDrawer", anchor = "left", "Drawer content here")
)

shinyApp(ui, function(input, output, session) {})
```

---

Fab	<i>Fab</i>
-----	------------

---

### Description

<https://mui.com/material-ui/api/fab/>

**Usage**

```
Fab(...)
```

```
Fab.shinyInput(inputId, ...)
```

```
updateFab.shinyInput(session = shiny::getDefaultReactiveDomain(), inputId, ...)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- color 'default' | 'error' | 'info' | 'inherit' | 'primary' | 'secondary' | 'success' | 'warning' | string  
Default is 'default' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- disabled bool  
Default is FALSE If true, the component is disabled.
- disableFocusRipple bool  
Default is FALSE If true, the keyboard focus ripple is disabled.
- disableRipple bool  
Default is - If true, the ripple effect is disabled.
- href string  
Default is - The URL to link to when the button is clicked. If defined, an element will be used as the root node.
- size 'small' | 'medium' | 'large' | string  
Default is 'large' The size of the component. small is equivalent to the dense button styling.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- variant 'circular' | 'extended' | string  
Default is 'circular' The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Fade	<i>Fade</i>
------	-------------

---

**Description**

<https://mui.com/material-ui/api/fade/>

**Usage**

```
Fade(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children` element  
Default is - A single child content element. This needs to be able to hold a ref.
- `addEndListener` func  
Default is - Add a custom transition end trigger. Called with the transitioning DOM node and a done callback. Allows for more fine grained transition end logic. Note: Timeouts are still used as a fallback if provided.
- `appear` bool  
Default is TRUE Perform the enter transition when it first mounts if `in` is also true. Set this to false to disable this behavior.
- `easing` { `enter?: string`, `exit?: string` } | string  
Default is - The transition timing function. You may specify a single easing or a object containing enter and exit values.
- `in` bool  
Default is - If true, the component will transition in.
- `timeout` number | { `appear?: number`, `enter?: number`, `exit?: number` }  
Default is `enter: theme.transitions.duration.enteringScreen`, `exit: theme.transitions.duration.leavingScreen`,  
The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

FilledInput

*FilledInput***Description**

<https://mui.com/material-ui/api/filled-input/>

**Usage**

```
FilledInput(...)
```

```
FilledInput.shinyInput(inputId, ..., value = defaultValue)
```

```
updateFilledInput.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- **autoComplete** string  
Default is - This prop helps users to fill forms faster, especially on mobile devices. The name can be confusing, as it's more like an autofill. You can learn more about it following the specification.
- **autoFocus** bool  
Default is - If true, the input element is focused during the first mount.
- **classes** object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **color** 'primary' | 'secondary' | string  
Default is - The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide. The prop defaults to the value ('primary') inherited from the parent FormControl component.
- **components** { Input?: elementType, Root?: elementType }  
Default is The components used for each slot inside. **Deprecated** use the slots prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.

- `componentsProps { input?: object, root?: object }`  
Default is The extra props for the slot components. You can override the existing props or add new ones. Deprecated use the `slotProps` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `defaultValue any`  
Default is - The default value. Use when the component is not controlled.
- `disabled bool`  
Default is - If true, the component is disabled. The prop defaults to the value (false) inherited from the parent `FormControl` component.
- `disableUnderline bool`  
Default is FALSE If true, the input will not have an underline.
- `endAdornment node`  
Default is - End `InputAdornment` for this component.
- `error bool`  
Default is - If true, the input will indicate an error. The prop defaults to the value (false) inherited from the parent `FormControl` component.
- `fullWidth bool`  
Default is FALSE If true, the input will take up the full width of its container.
- `hiddenLabel bool`  
Default is FALSE If true, the label is hidden. This is used to increase density for a `FilledInput`. Be sure to add `aria-label` to the input element.
- `id string`  
Default is - The id of the input element.
- `inputComponent elementType`  
Default is 'input' The component used for the input element. Either a string to use a HTML element or a component.
- `inputProps object`  
Default is Attributes applied to the input element.
- `inputRef ref`  
Default is - Pass a ref to the input element.
- `margin 'dense' | 'none'`  
Default is - If dense, will adjust vertical spacing. This is normally obtained via context from `FormControl`. The prop defaults to the value ('none') inherited from the parent `FormControl` component.
- `maxRows number | string`  
Default is - Maximum number of rows to display when multiline option is set to true.
- `minRows number | string`  
Default is - Minimum number of rows to display when multiline option is set to true.
- `multiline bool`  
Default is FALSE If true, a `TextareaAutosize` element is rendered.
- `name string`  
Default is - Name attribute of the input element.

- `onChange` func  
Default is - Callback fired when the value is changed. Signature: `function(event: React.ChangeEvent) = void`. The event source of the callback. You can pull out the new value by accessing `event.target.value` (string).
- `placeholder` string  
Default is - The short hint displayed in the input before the user enters a value.
- `readOnly` bool  
Default is - It prevents the user from changing the value of the field (not from interacting with the field).
- `required` bool  
Default is - If true, the input element is required. The prop defaults to the value (false) inherited from the parent `FormControl` component.
- `rows` number | string  
Default is - Number of rows to display when multiline option is set to true.
- `slotProps` { `input?: object`, `root?: object` }  
Default is The extra props for the slot components. You can override the existing props or add new ones. This prop is an alias for the `componentsProps` prop, which will be deprecated in the future.
- `slots` { `input?: elementType`, `root?: elementType` }  
Default is The components used for each slot inside. This prop is an alias for the `components` prop, which will be deprecated in the future.
- `startAdornment` node  
Default is - Start `InputAdornment` for this component.
- `sx` Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `type` string  
Default is 'text' Type of the input element. It should be a valid HTML5 input type.
- `value` any  
Default is - The value of the input element, required for a controlled component.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

FormControl

*FormControl*

---

**Description**

<https://mui.com/material-ui/api/form-control/>

**Usage**

`FormControl(...)`

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- color 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'primary' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- disabled bool  
Default is FALSE If true, the label, input and helper text should be displayed in a disabled state.
- error bool  
Default is FALSE If true, the label is displayed in an error state.
- focused bool  
Default is - If true, the component is displayed in focused state.
- fullWidth bool  
Default is FALSE If true, the component will take up the full width of its container.
- hiddenLabel bool  
Default is FALSE If true, the label is hidden. This is used to increase density for a FilledInput. Be sure to add aria-label to the input element.
- margin 'dense' | 'none' | 'normal'  
Default is 'none' If dense or normal, will adjust vertical spacing of this and contained components.
- required bool  
Default is FALSE If true, the label will indicate that the input is required.
- size 'medium' | 'small' | string  
Default is 'medium' The size of the component.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- variant 'filled' | 'outlined' | 'standard'  
Default is 'outlined' The variant to use.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

FormControlLabel      *FormControlLabel*

---

## Description

<https://mui.com/material-ui/api/form-control-label/>

## Usage

```
FormControlLabel(...)
```

```
FormControlLabel.shinyInput(inputId, ..., value = defaultValue)
```

```
updateFormControlLabel.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

## Arguments

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

## Details

- control element  
Default is - A control element. For instance, it can be a Radio, a Switch or a Checkbox.
- checked bool  
Default is - If true, the component appears selected.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- componentsProps { typography?: object }  
Default is The props used for each slot inside. Deprecated use the slotProps prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- disabled bool  
Default is - If true, the control is disabled.
- disableTypography bool  
Default is - If true, the label is rendered as it is passed without an additional typography node.
- inputRef ref  
Default is - Pass a ref to the input element.

- `label node`  
Default is - A text or an element to be used in an enclosing label element.
- `labelPlacement 'bottom' | 'end' | 'start' | 'top'`  
Default is 'end' The position of the label.
- `onChange func`  
Default is - Callback fired when the state is changed. Signature: `function(event: React.SyntheticEvent) = void` event The event source of the callback. You can pull out the new checked state by accessing `event.target.checked` (boolean).
- `required bool`  
Default is - If true, the label will indicate that the input is required.
- `slotProps { typography?: func | object }`  
Default is The props used for each slot inside.
- `slots { typography?: elementType }`  
Default is The components used for each slot inside.
- `sx Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `value any`  
Default is - The value of the component.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 FormGroup

*FormGroup*


---

**Description**

<https://mui.com/material-ui/api/form-group/>

**Usage**

```
FormGroup(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- row bool  
Default is FALSE Display group of elements in a compact row.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

FormHelperText	<i>FormHelperText</i>
----------------	-----------------------

---

**Description**

<https://mui.com/material-ui/api/form-helper-text/>

**Usage**

```
FormHelperText(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component. If ' ' is provided, the component reserves one line height for displaying a future message.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is NA The component used for the root node. Either a string to use a HTML element or a component.
- disabled bool  
Default is NA If true, the helper text should be displayed in a disabled state.

- `error` `bool`  
Default is `NA` If true, helper text should be displayed in an error state.
- `filled` `bool`  
Default is `NA` If true, the helper text should use filled classes key.
- `focused` `bool`  
Default is `NA` If true, the helper text should use focused classes key.
- `margin` `'dense'`  
Default is `NA` If dense, will adjust vertical spacing. This is normally obtained via context from `FormControl`.
- `required` `bool`  
Default is `NA` If true, the helper text should use required classes key.
- `sx` `Array` `func` | `object` | `bool` | `func` | `object`  
Default is `NA` The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant` `'filled'` | `'outlined'` | `'standard'` | `string`  
Default is `NA` The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

FormLabel

*FormLabel*

---

**Description**

<https://mui.com/material-ui/api/form-label/>

**Usage**

```
FormLabel(...)
```

**Arguments**

`...` Props to pass to the component.

**Details**

- `children` `node`  
Default is `NA` The content of the component.
- `classes` `object`  
Default is `NA` Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` `'error'` | `'info'` | `'primary'` | `'secondary'` | `'success'` | `'warning'` | `string`  
Default is `NA` The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.

- `elementType`  
Default is NA The component used for the root node. Either a string to use a HTML element or a component.
- `disabled` bool  
Default is NA If true, the label should be displayed in a disabled state.
- `error` bool  
Default is NA If true, the label is displayed in an error state.
- `filled` bool  
Default is NA If true, the label should use filled classes key.
- `focused` bool  
Default is NA If true, the input of this label is focused (used by FormGroup components).
- `required` bool  
Default is NA If true, the label will indicate that the input is required.
- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

GlobalStyles

*GlobalStyles*

---

**Description**

<https://mui.com/material-ui/api/global-styles/>

**Usage**

```
GlobalStyles(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `styles` array | `func` | `number` | `object` | `string` | `bool`  
Default is NA The styles you want to apply globally.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 Grid

*Grid*


---

### Description

<https://mui.com/material-ui/api/grid/>

### Usage

```
Grid(...)
```

### Arguments

... Props to pass to the component.

### Details

- children node  
Default is - The content of the component.
- columns Array number | number | object  
Default is 12 The number of columns.
- columnSpacing Array number | string | number | object | string  
Default is - Defines the horizontal space between the type item components. It overrides the value of the spacing prop.
- container bool  
Default is FALSE If true, the component will have the flex container behavior. You should be wrapping items with a container.
- direction 'column-reverse' | 'column' | 'row-reverse' | 'row' | Array 'column-reverse' | 'column' | 'row-reverse' | 'row'  
Default is 'row' Defines the flex-direction style property. It is applied for all screen sizes.
- offset string | number | Array string | number | object  
Default is - Defines the offset value for the type item components.
- rowSpacing Array number | string | number | object | string  
Default is - Defines the vertical space between the type item components. It overrides the value of the spacing prop.
- size string | bool | number | Array string | bool | number | object  
Default is - Defines the size of the the type item components.
- spacing Array number | string | number | object | string  
Default is 0 Defines the space between the type item components. It can only be used on a type container component.
- wrap 'nowrap' | 'wrap-reverse' | 'wrap'  
Default is 'wrap' Defines the flex-wrap style property. It's applied for all screen sizes.

### Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

Grow

*Grow*

---

## Description

<https://mui.com/material-ui/api/grow/>

## Usage

```
Grow(...)
```

## Arguments

... Props to pass to the component.

## Details

- children element  
Default is - A single child content element. This needs to be able to hold a ref.
- addEndListener func  
Default is - Add a custom transition end trigger. Called with the transitioning DOM node and a done callback. Allows for more fine grained transition end logic. Note: Timeouts are still used as a fallback if provided.
- appear bool  
Default is TRUE Perform the enter transition when it first mounts if in is also true. Set this to false to disable this behavior.
- easing { enter?: string, exit?: string } | string  
Default is - The transition timing function. You may specify a single easing or a object containing enter and exit values.
- in bool  
Default is - If true, the component will transition in.
- timeout 'auto' | number | { appear?: number, enter?: number, exit?: number }  
Default is 'auto' The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object. Set to 'auto' to automatically calculate transition time based on height.

## Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

 Icon
*Icon***Description**

<https://mui.com/material-ui/api/icon/>

**Usage**

```
Icon(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- **baseClassName** string  
Default is 'material-icons' The base class applied to the icon. Defaults to 'material-icons', but can be changed to any other base class that suits the icon font you're using (for example material-icons-rounded, fas, etc).
- **children** node  
Default is - The name of the icon font ligature.
- **classes** object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **color** 'inherit' | 'action' | 'disabled' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'inherit' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- **component** elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- **fontSize** 'inherit' | 'large' | 'medium' | 'small' | string  
Default is 'medium' The fontSize applied to the icon. Defaults to 24px, but can be configured to inherit font size.
- **sx** Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

 IconButton
*IconButton***Description**

<https://mui.com/material-ui/api/icon-button/>

**Usage**

```
IconButton(...)
```

```
IconButton.shinyInput(inputId, ...)
```

```
updateIconButton.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- children node  
Default is - The icon to display.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- color 'inherit' | 'default' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'default' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- disabled bool  
Default is FALSE If true, the component is disabled.
- disableFocusRipple bool  
Default is FALSE If true, the keyboard focus ripple is disabled.
- disableRipple bool  
Default is FALSE If true, the ripple effect is disabled. Without a ripple there is no styling for :focus-visible by default. Be sure to highlight the element by applying separate styles with the .Mui-focusVisible class.

- `edge 'end' | 'start' | false`  
Default is FALSE If given, uses a negative margin to counteract the padding on one side (this is often helpful for aligning the left or right side of the icon with content above or below, without ruining the border size and shape).
- `loading bool`  
Default is null If true, the loading indicator is visible and the button is disabled. If true | false, the loading wrapper is always rendered before the children to prevent Google Translation Crash.
- `loadingIndicator node`  
Default is `CircularProgress color="inherit" size=16` / Element placed before the children if the button is in loading state. The node should contain an element with `role="progressbar"` with an accessible name. By default, it renders a `CircularProgress` that is labeled by the button itself.
- `size 'small' | 'medium' | 'large' | string`  
Default is 'medium' The size of the component. small is equivalent to the dense button styling.
- `sx Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

ImageList

*ImageList*

---

**Description**

<https://mui.com/material-ui/api/image-list/>

**Usage**

```
ImageList(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children node`  
Default is - The content of the component, normally `ImageListItems`.
- `classes object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.

- `cols` integer  
Default is 2 Number of columns.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `gap` number  
Default is 4 The gap between items in px.
- `rowHeight` 'auto' | number  
Default is 'auto' The height of one row in px.
- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant` 'masonry' | 'quilted' | 'standard' | 'woven' | `string`  
Default is 'standard' The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

ImageListItem

*ImageListItem*

---

**Description**

<https://mui.com/material-ui/api/image-list-item/>

**Usage**

```
ImageListItem(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children` node  
Default is - The content of the component, normally an `img`.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `cols` integer  
Default is 1 Width of the item in number of grid columns.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.

- `rows` integer  
Default is 1 Height of the item in number of grid rows.
- `sx` Array `func| object| bool | func| object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

ImageListItemBar	<i>ImageListItemBar</i>
------------------	-------------------------

---

**Description**

<https://mui.com/material-ui/api/image-list-item-bar/>

**Usage**

```
ImageListItemBar(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- `actionIcon` node  
Default is - An `IconButton` element to be used as secondary action target (primary action target is the item itself).
- `actionPosition` `'left' | 'right'`  
Default is `'right'` Position of secondary action `IconButton`.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `position` `'below' | 'bottom' | 'top'`  
Default is `'bottom'` Position of the title bar.
- `subtitle` node  
Default is - String or element serving as subtitle (support text).
- `sx` Array `func| object| bool | func| object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `title` node  
Default is - Title to be displayed.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 Input
*Input***Description**

<https://mui.com/material-ui/api/input/>

**Usage**

```
Input(...)
```

```
Input.shinyInput(inputId, ..., value = defaultValue)
```

```
updateInput.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- **autoComplete** string  
Default is - This prop helps users to fill forms faster, especially on mobile devices. The name can be confusing, as it's more like an autofill. You can learn more about it following the specification.
- **autoFocus** bool  
Default is - If true, the input element is focused during the first mount.
- **classes** object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **color** 'primary' | 'secondary' | string  
Default is - The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide. The prop defaults to the value ('primary') inherited from the parent FormControl component.
- **components** { Input?: elementType, Root?: elementType }  
Default is The components used for each slot inside. **Deprecated** use the slots prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.

- `componentsProps { input?: object, root?: object }`  
Default is The extra props for the slot components. You can override the existing props or add new ones. Deprecated use the `slotProps` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `defaultValue any`  
Default is - The default value. Use when the component is not controlled.
- `disabled bool`  
Default is - If true, the component is disabled. The prop defaults to the value (false) inherited from the parent `FormControl` component.
- `disableUnderline bool`  
Default is FALSE If true, the input will not have an underline.
- `endAdornment node`  
Default is - End `InputAdornment` for this component.
- `error bool`  
Default is - If true, the input will indicate an error. The prop defaults to the value (false) inherited from the parent `FormControl` component.
- `fullWidth bool`  
Default is FALSE If true, the input will take up the full width of its container.
- `id string`  
Default is - The id of the input element.
- `inputComponent elementType`  
Default is 'input' The component used for the input element. Either a string to use a HTML element or a component.
- `inputProps object`  
Default is Attributes applied to the input element.
- `inputRef ref`  
Default is - Pass a ref to the input element.
- `margin 'dense' | 'none'`  
Default is - If dense, will adjust vertical spacing. This is normally obtained via context from `FormControl`. The prop defaults to the value ('none') inherited from the parent `FormControl` component.
- `maxRows number | string`  
Default is - Maximum number of rows to display when multiline option is set to true.
- `minRows number | string`  
Default is - Minimum number of rows to display when multiline option is set to true.
- `multiline bool`  
Default is FALSE If true, a `TextareaAutosize` element is rendered.
- `name string`  
Default is - Name attribute of the input element.
- `onChange func`  
Default is - Callback fired when the value is changed. Signature: `function(event: React.ChangeEvent) = void` event The event source of the callback. You can pull out the new value by accessing `event.target.value` (string).

- `placeholder` string  
Default is - The short hint displayed in the input before the user enters a value.
- `readOnly` bool  
Default is - It prevents the user from changing the value of the field (not from interacting with the field).
- `required` bool  
Default is - If true, the input element is required. The prop defaults to the value (false) inherited from the parent `FormControl` component.
- `rows` number | string  
Default is - Number of rows to display when multiline option is set to true.
- `slotProps` { `input?: object`, `root?: object` }  
Default is The extra props for the slot components. You can override the existing props or add new ones. This prop is an alias for the `componentsProps` prop, which will be deprecated in the future.
- `slots` { `input?: elementType`, `root?: elementType` }  
Default is The components used for each slot inside. This prop is an alias for the `components` prop, which will be deprecated in the future.
- `startAdornment` node  
Default is - Start `InputAdornment` for this component.
- `sx` Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `type` string  
Default is 'text' Type of the input element. It should be a valid HTML5 input type.
- `value` any  
Default is - The value of the input element, required for a controlled component.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

InputAdornment	<i>InputAdornment</i>
----------------	-----------------------

---

**Description**

<https://mui.com/material-ui/api/input-adornment/>

**Usage**

```
InputAdornment(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- position 'end' | 'start'  
Default is - The position this adornment should appear relative to the Input.
- children node  
Default is - The content of the component, normally an IconButton or string.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- disablePointerEvents bool  
Default is FALSE Disable pointer events on the root. This allows for the content of the adornment to focus the input on click.
- disableTypography bool  
Default is FALSE If children is a string then disable wrapping in a Typography component.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- variant 'filled' | 'outlined' | 'standard'  
Default is - The variant to use. Note: If you are using the TextField component or the FormControl component you do not have to set this manually.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

InputBase

*InputBase*

---

**Description**

<https://mui.com/material-ui/api/input-base/>

**Usage**

```
InputBase(...)
```

**Arguments**

... Props to pass to the component.

## Details

- `autoComplete` `string`  
Default is - This prop helps users to fill forms faster, especially on mobile devices. The name can be confusing, as it's more like an autofill. You can learn more about it following the specification.
- `autoFocus` `bool`  
Default is - If true, the input element is focused during the first mount.
- `classes` `object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` `'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string`  
Default is - The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide. The prop defaults to the value ('primary') inherited from the parent FormControl component.
- `components` `{ Input?: elementType, Root?: elementType }`  
Default is The components used for each slot inside. ~~Deprecated~~ use the `slots` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `componentsProps` `{ input?: object, root?: object }`  
Default is The extra props for the slot components. You can override the existing props or add new ones. ~~Deprecated~~ use the `slotProps` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `defaultValue` `any`  
Default is - The default value. Use when the component is not controlled.
- `disabled` `bool`  
Default is - If true, the component is disabled. The prop defaults to the value (false) inherited from the parent FormControl component.
- `disableInjectingGlobalStyles` `bool`  
Default is FALSE If true, GlobalStyles for the auto-fill keyframes will not be injected/removed on mount/unmount. Make sure to inject them at the top of your application. This option is intended to help with boosting the initial rendering performance if you are loading a big amount of Input components at once.
- `endAdornment` `node`  
Default is - End InputAdornment for this component.
- `error` `bool`  
Default is - If true, the input will indicate an error. The prop defaults to the value (false) inherited from the parent FormControl component.
- `fullWidth` `bool`  
Default is FALSE If true, the input will take up the full width of its container.
- `id` `string`  
Default is - The id of the input element.
- `inputComponent` `element type`  
Default is 'input' The component used for the input element. Either a string to use a HTML element or a component. This needs to be able to hold a ref.

- `inputProps` object  
Default is Attributes applied to the input element.
- `inputRef` `ref`  
Default is - Pass a ref to the input element.
- `margin` `'dense' | 'none'`  
Default is - If dense, will adjust vertical spacing. This is normally obtained via context from `FormControl`. The prop defaults to the value (`'none'`) inherited from the parent `FormControl` component.
- `maxRows` `number | string`  
Default is - Maximum number of rows to display when multiline option is set to true.
- `minRows` `number | string`  
Default is - Minimum number of rows to display when multiline option is set to true.
- `multiline` `bool`  
Default is `FALSE` If true, a `TextareaAutosize` element is rendered.
- `name` `string`  
Default is - Name attribute of the input element.
- `onBlur` `func`  
Default is - Callback fired when the input is blurred. Notice that the first argument (event) might be undefined.
- `onChange` `func`  
Default is - Callback fired when the value is changed. Signature: `function(event: React.ChangeEvent) = void` event The event source of the callback. You can pull out the new value by accessing `event.target.value` (string).
- `onInvalid` `func`  
Default is - Callback fired when the input doesn't satisfy its constraints.
- `placeholder` `string`  
Default is - The short hint displayed in the input before the user enters a value.
- `readOnly` `bool`  
Default is - It prevents the user from changing the value of the field (not from interacting with the field).
- `required` `bool`  
Default is - If true, the input element is required. The prop defaults to the value (`false`) inherited from the parent `FormControl` component.
- `rows` `number | string`  
Default is - Number of rows to display when multiline option is set to true.
- `size` `'medium' | 'small' | string`  
Default is - The size of the component.
- `slotProps` { `input?: object`, `root?: object` }  
Default is The extra props for the slot components. You can override the existing props or add new ones. This prop is an alias for the `componentsProps` prop, which will be deprecated in the future.
- `slots` { `input?: elementType`, `root?: elementType` }  
Default is The components used for each slot inside. This prop is an alias for the `components` prop, which will be deprecated in the future.

- `startAdornment` node  
Default is - Start InputAdornment for this component.
- `sx` Array `func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `type` string  
Default is 'text' Type of the input element. It should be a valid HTML5 input type.
- `value` any  
Default is - The value of the input element, required for a controlled component.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

InputLabel

*InputLabel*

---

**Description**

<https://mui.com/material-ui/api/input-label/>

**Usage**

```
InputLabel(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children` node  
Default is - The content of the component.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` 'error' | 'info' | 'primary' | 'secondary' | 'success' | 'warning' | string  
Default is - The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- `disableAnimation` bool  
Default is FALSE If true, the transition animation is disabled.
- `disabled` bool  
Default is - If true, the component is disabled.
- `error` bool  
Default is - If true, the label is displayed in an error state.

- `focused` `bool`  
Default is - If true, the input of this label is focused.
- `margin` `'dense'`  
Default is - If dense, will adjust vertical spacing. This is normally obtained via context from `FormControl`.
- `required` `bool`  
Default is - if true, the label will indicate that the input is required.
- `shrink` `bool`  
Default is - If true, the label is shrunk.
- `size` `'medium'` | `'small'` | `string`  
Default is `'medium'` The size of the component.
- `sx` `Array` `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant` `'filled'` | `'outlined'` | `'standard'`  
Default is - The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

LinearProgress

*LinearProgress*

---

**Description**

<https://mui.com/material-ui/api/linear-progress/>

**Usage**

```
LinearProgress(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `classes` `object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` `'inherit'` | `'primary'` | `'secondary'` | `string`  
Default is `'primary'` The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.

- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `value` number  
Default is - The value of the progress indicator for the determinate and buffer variants. Value between 0 and 100.
- `valueBuffer` number  
Default is - The value for the buffer variant. Value between 0 and 100.
- `variant` `'buffer'` | `'determinate'` | `'indeterminate'` | `'query'`  
Default is `'indeterminate'` The variant to use. Use `indeterminate` or `query` when there is no progress value.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Link

*Link*

---

**Description**

<https://mui.com/material-ui/api/link/>

**Usage**

`Link(...)`

**Arguments**

`...` Props to pass to the component.

**Details**

- `children` node  
Default is - The content of the component.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` `'primary'` | `'secondary'` | `'success'` | `'error'` | `'info'` | `'warning'` | `'textPrimary'` | `'textSecondary'` | `'textDisabled'` | `string`  
Default is `'primary'` The color of the link.
- `componentElement` type  
Default is - The component used for the root node. Either a string to use a HTML element or a component. This needs to be able to hold a ref.

- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `TypographyClasses` `object`  
Default is - classes prop applied to the `Typography` element.
- `underline` `'always'` | `'hover'` | `'none'`  
Default is `'always'` Controls when the link should have an underline.
- `variant` `'body1'` | `'body2'` | `'button'` | `'caption'` | `'h1'` | `'h2'` | `'h3'` | `'h4'` | `'h5'` | `'h6'` | `'inherit'` | `'overline'` | `'subtitle1'` | `'subtitle2'` | `string`  
Default is `'inherit'` Applies the theme typography styles.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

List

*List*

---

**Description**

<https://mui.com/material-ui/api/list/>

**Usage**

```
List(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children` `node`  
Default is - The content of the component.
- `classes` `object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `dense` `bool`  
Default is `FALSE` If true, compact vertical padding designed for keyboard and mouse input is used for the list and list items. The prop is available to descendant components as the `dense` context.
- `disablePadding` `bool`  
Default is `FALSE` If true, vertical padding is removed from the list.

- subheader node  
Default is - The content of the subheader, normally ListSubheader.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

ListItem	<i>ListItem</i>
----------	-----------------

---

**Description**

<https://mui.com/material-ui/api/list-item/>

**Usage**

ListItem(...)

**Arguments**

... Props to pass to the component.

**Details**

- alignItems 'center' | 'flex-start'  
Default is 'center' Defines the align-items style property.
- children node  
Default is - The content of the component if a ListItemSecondaryAction is used it must be the last child.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- components { Root?: elementType }  
Default is The components used for each slot inside. ~~Deprecated Use the slots prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.~~
- componentsProps { root?: object }  
Default is The extra props for the slot components. You can override the existing props or add new ones. ~~Deprecated Use the slotProps prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.~~

- **ContainerComponent** element type  
Default is 'li' The container component used when a ListItemSecondaryAction is the last child. This needs to be able to hold a ref. **Deprecated** Use the component or slots.root prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- **ContainerProps** object  
Default is Props applied to the container component if used. **Deprecated** Use the slotProps.root prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- **dense** bool  
Default is FALSE If true, compact vertical padding designed for keyboard and mouse input is used. The prop defaults to the value inherited from the parent List component.
- **disableGutters** bool  
Default is FALSE If true, the left and right padding is removed.
- **disablePadding** bool  
Default is FALSE If true, all padding is removed.
- **divider** bool  
Default is FALSE If true, a 1px light border is added to the bottom of the list item.
- **secondaryAction** node  
Default is - The element to display at the end of ListItem.
- **slotProps** { root?: object }  
Default is The extra props for the slot components. You can override the existing props or add new ones.
- **slots** { root?: elementType }  
Default is The components used for each slot inside.
- **sx** Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

ListItemAvatar

*ListItemAvatar*

---

**Description**

<https://mui.com/material-ui/api/list-item-avatar/>

**Usage**

ListItemAvatar(...)

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component, normally an Avatar.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func | object | bool | func | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

ListItemButton	<i>ListItemButton</i>
----------------	-----------------------

---

**Description**

<https://mui.com/material-ui/api/list-item-button/>

**Usage**

ListItemButton(...)

**Arguments**

... Props to pass to the component.

**Details**

- alignItems 'center' | 'flex-start'  
Default is 'center' Defines the align-items style property.
- autoFocus bool  
Default is FALSE If true, the list item is focused during the first mount. Focus will also be triggered if the value changes from false to true.
- children node  
Default is - The content of the component if a ListItemSecondaryAction is used it must be the last child.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.

- `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `dense` `bool`  
Default is `FALSE` If true, compact vertical padding designed for keyboard and mouse input is used. The prop defaults to the value inherited from the parent List component.
- `disabled` `bool`  
Default is `FALSE` If true, the component is disabled.
- `disableGutters` `bool`  
Default is `FALSE` If true, the left and right padding is removed.
- `divider` `bool`  
Default is `FALSE` If true, a 1px light border is added to the bottom of the list item.
- `focusVisibleClassName` `string`  
Default is - This prop can help identify which element has keyboard focus. The class name will be applied when the element gains the focus through keyboard interaction. It's a polyfill for the CSS `:focus-visible` selector. The rationale for using this feature is explained here. A polyfill can be used to apply a focus-visible class to other components if needed.
- `selected` `bool`  
Default is `FALSE` Use to apply selected styling.
- `sx` `Array` `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

ListItemIcon

*ListItemIcon*

---

**Description**

<https://mui.com/material-ui/api/list-item-icon/>

**Usage**

```
ListItemIcon(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component, normally Icon, SvgIcon, or a @mui/icons-material SVG icon element.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func| object| bool | func| object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

ListItemSecondaryAction

*ListItemSecondaryAction*

---

**Description**

<https://mui.com/material-ui/api/list-item-secondary-action/>

**Usage**

```
ListItemSecondaryAction(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component, normally an IconButton or selection control.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func| object| bool | func| object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

 ListItemText

*ListItemText*


---

## Description

<https://mui.com/material-ui/api/list-item-text/>

## Usage

```
ListItemText(...)
```

## Arguments

... Props to pass to the component.

## Details

- children node  
Default is - Alias for the primary prop.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- disableTypography bool  
Default is FALSE If true, the children won't be wrapped by a Typography component. This can be useful to render an alternative Typography variant by wrapping the children (or primary) text, and optional secondary text with the Typography component.
- inset bool  
Default is FALSE If true, the children are indented. This should be used if there is no left avatar or left icon.
- primary node  
Default is - The main content element.
- primaryTypographyProps object  
Default is - These props will be forwarded to the primary typography component (as long as disableTypography is not true). Deprecated Use slotProps.primary instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- secondary node  
Default is - The secondary content element.
- secondaryTypographyProps object  
Default is - These props will be forwarded to the secondary typography component (as long as disableTypography is not true). Deprecated Use slotProps.secondary instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- slotProps { primary?: func| object, root?: func| object, secondary?: func| object }  
Default is The props used for each slot inside.
- slots { primary?: elementType, root?: elementType, secondary?: elementType }  
Default is The components used for each slot inside.

- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

ListSubheader

*ListSubheader*

---

**Description**

<https://mui.com/material-ui/api/list-subheader/>

**Usage**

ListSubheader(...)

**Arguments**

...                    Props to pass to the component.

**Details**

- `children` node  
Default is - The content of the component.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` 'default' | 'inherit' | 'primary'  
Default is 'default' The color of the component. It supports those theme colors that make sense for this component.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `disableGutters` `bool`  
Default is FALSE If true, the List Subheader will not have gutters.
- `disableSticky` `bool`  
Default is FALSE If true, the List Subheader will not stick to the top during scroll.
- `inset` `bool`  
Default is FALSE If true, the List Subheader is indented.
- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

LoadingButton	<i>LoadingButton</i>
---------------	----------------------

---

**Description**

<https://mui.com/material-ui/react-button/>

**Usage**

```
LoadingButton(...)  
  
LoadingButton.shinyInput(inputId, ...)  
  
updateLoadingButton.shinyInput(  
  session = shiny::getDefaultReactiveDomain(),  
  inputId,  
  ...  
)
```

**Arguments**

<code>...</code>	Props to pass to the component.
<code>inputId</code>	ID of the component.
<code>session</code>	Object passed as the ‘session’ argument to Shiny server.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

**Note**

LoadingButton is part of `@mui/lab`, which is published on the MUI beta channel. Lab APIs may change in future minor releases.

---

Masonry

*Masonry*


---

**Description**

<https://mui.com/material-ui/api/masonry/>

**Usage**

```
Masonry(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- columns Array number | string | number | object | string  
Default is 4 Number of columns.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- defaultColumns number  
Default is - The default number of columns of the component. This is provided for server-side rendering.
- defaultHeight number  
Default is - The default height of the component in px. This is provided for server-side rendering.
- defaultSpacing number  
Default is - The default spacing of the component. Like spacing, it is a factor of the theme's spacing. This is provided for server-side rendering.
- sequential bool  
Default is FALSE Allows using sequential order rather than adding to shortest column
- spacing Array number | string | number | object | string  
Default is 1 Defines the space between children. It is a factor of the theme's spacing.
- sx Array func | object | bool | func | object  
Default is - Allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Menu

*Menu*

---

**Description**

<https://mui.com/material-ui/api/menu/>

**Usage**

```
Menu(...)
```

```
Menu.shinyInput(inputId, ...)
```

```
updateMenu.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

<code>...</code>	Props to pass to the component.
<code>inputId</code>	ID of the component.
<code>session</code>	Object passed as the ‘session’ argument to Shiny server.

**Details**

- `open` bool  
Default is - If true, the component is shown.
- `anchorEl` HTML element | func  
Default is - An HTML element, or a function that returns one. It’s used to set the position of the menu.
- `autoFocus` bool  
Default is TRUE If true (Default) will focus the role="menu" if no focusable child is found. Disabled children are not focusable. If you set this prop to false focus will be placed on the parent modal container. This has severe accessibility implications and should only be considered if you manage focus otherwise.
- `children` node  
Default is - Menu contents, normally MenuItems.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.

- `disableAutoFocusItem` bool  
Default is FALSE When opening the menu will not focus the active item but the `role="menu"` unless `autoFocus` is also set to false. Not using the default means not following WAI-ARIA authoring practices. Please be considerate about possible accessibility implications.
- `MenuListProps` object  
Default is Props applied to the `MenuList` element. Deprecated use the `slotProps.list` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `onClose` func  
Default is - Callback fired when the component requests to be closed. Signature: `function(event: object, reason: string) = void` event The event source of the callback. reason Can be: "escapeKeyDown", "backdropClick", "tabKeyDown".
- `PopoverClasses` object  
Default is - classes prop applied to the Popover element.
- `slotProps` { `backdrop?: func | object`, `list?: func | object`, `paper?: func | object`, `root?: func | object`, ... }  
Default is The props used for each slot inside.
- `slots` { `backdrop?: elementType`, `list?: elementType`, `paper?: elementType`, `root?: elementType`, `transition?: elementType`, ... }  
Default is The components used for each slot inside.
- `sx` Array `func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the [sx](#) page for more details.
- `transitionDuration` 'auto' | number | { `appear?: number`, `enter?: number`, `exit?: number` }  
Default is 'auto' The length of the transition in ms, or 'auto'
- `TransitionProps` object  
Default is Props applied to the transition element. By default, the element is based on this `Transition` component. Deprecated use the `slotProps.transition` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `variant` 'menu' | 'selectedMenu'  
Default is 'selectedMenu' The variant to use. Use `menu` to prevent selected items from impacting the initial focus.

## Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

## Note

This is an overlay surface wired as a click-reporter: `input[[inputId]]` holds a click count (it does not tell you *which* `MenuItem` was clicked), and the wrapper does **not** manage the open state. Render it with `open = TRUE/FALSE` and toggle visibility from the server with `updateMenu.shinyInput(session, inputId, open = TRUE)`. For the common "open on click of a button" pattern, [Menu.triggerId](#) is simpler (open/close handled entirely client-side, no server logic).

---

Menu.triggerId	<i>Menu.triggerId</i>
----------------	-----------------------

---

**Description**

Custom Menu bound to a DOM element by id. See `'js/src/MuiMenuTriggerId.jsx'`.

**Usage**

```
Menu.triggerId(triggerId, ...)
```

**Arguments**

triggerId	HTML id of an existing DOM element that acts as the trigger (button, link, etc.) to open the Menu.
...	Named arguments forwarded as React props, plus children to render inside the component. Pass <code>'closeOnItemClick = FALSE'</code> to keep the menu open after a click.

**Details**

Pass `'closeOnItemClick = FALSE'` to disable auto-close on click (useful when the menu contains interactive children like checkboxes).

**Value**

Object with `'shiny.tag'` class suitable for use in the UI of a Shiny app.

---

MenuItem	<i>MenuItem</i>
----------	-----------------

---

**Description**

<https://mui.com/material-ui/api/menu-item/>

**Usage**

```
MenuItem(...)

MenuItem.shinyInput(inputId, ...)

updateMenuItem.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

## Arguments

...	Props to pass to the component.
inputId	ID of the component.
session	Object passed as the 'session' argument to Shiny server.

## Details

- **autoFocus** bool  
Default is FALSE If true, the list item is focused during the first mount. Focus will also be triggered if the value changes from false to true.
- **children** node  
Default is - The content of the component.
- **classes** object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **component elementType**  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- **dense** bool  
Default is FALSE If true, compact vertical padding designed for keyboard and mouse input is used. The prop defaults to the value inherited from the parent Menu component.
- **disableGutters** bool  
Default is FALSE If true, the left and right padding is removed.
- **divider** bool  
Default is FALSE If true, a 1px light border is added to the bottom of the menu item.
- **focusVisibleClassName** string  
Default is - This prop can help identify which element has keyboard focus. The class name will be applied when the element gains the focus through keyboard interaction. It's a polyfill for the CSS `:focus-visible` selector. The rationale for using this feature is explained here. A polyfill can be used to apply a focus-visible class to other components if needed.
- **selected** bool  
Default is FALSE If true, the component is selected.
- **sx** Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

## Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

MenuList

*MenuList*

---

## Description

<https://mui.com/material-ui/api/menu-list/>

## Usage

```
MenuList(...)
```

## Arguments

...                    Props to pass to the component.

## Details

- `autoFocus` bool  
Default is FALSE If true, will focus the role="menu" container and move into tab order.
- `autoFocusItem` bool  
Default is FALSE If true, will focus the first menuitem if variant="menu" or selected item if variant="selectedMenu".
- `children` node  
Default is - MenuList contents, normally MenuItem.
- `disabledItemsFocusable` bool  
Default is FALSE If true, will allow focus on disabled items.
- `disableListWrap` bool  
Default is FALSE If true, the menu items will not wrap focus.
- `variant` 'menu' | 'selectedMenu'  
Default is 'selectedMenu' The variant to use. Use menu to prevent selected items from impacting the initial focus and the vertical alignment relative to the anchor element.

## Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 MobileStepper

*MobileStepper*


---

## Description

<https://mui.com/material-ui/api/mobile-stepper/>

## Usage

```
MobileStepper(...)
```

## Arguments

... Props to pass to the component.

## Details

- `steps` integer  
Default is - The total steps.
- `activeStep` integer  
Default is 0 Set the active step (zero based index). Defines which dot is highlighted when the variant is 'dots'.
- `backButton` node  
Default is - A back button element. For instance, it can be a Button or an IconButton.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `LinearProgressProps` object  
Default is - Props applied to the LinearProgress element. **Deprecated** Use `slotProps.progress` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `nextButton` node  
Default is - A next button element. For instance, it can be a Button or an IconButton.
- `position` 'bottom' | 'static' | 'top'  
Default is 'bottom' Set the positioning type.
- `slotProps` { `dot?: func | object`, `dots?: func | object`, `progress?: func | object`, `root?: func | object` }  
Default is The props used for each slot inside.
- `slots` { `dot?: elementType`, `dots?: elementType`, `progress?: elementType`, `root?: elementType` }  
Default is The components used for each slot inside.
- `sx` Array `func | object | bool` | `func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the [sx page](#) for more details.
- `variant` 'dots' | 'progress' | 'text'  
Default is 'dots' The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 Modal

*Modal*


---

**Description**

<https://mui.com/material-ui/api/modal/>

**Usage**

```
Modal(...)
```

```
Modal.shinyInput(inputId, ...)
```

```
updateModal.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

<code>...</code>	Props to pass to the component.
<code>inputId</code>	ID of the component.
<code>session</code>	Object passed as the 'session' argument to Shiny server.

**Details**

- `children` element  
Default is - A single child content element. This needs to be able to hold a ref.
- `open` bool  
Default is - If true, the component is shown.
- `BackdropComponent` elementType  
Default is `styled(Backdrop, name: 'MuiModal', slot: 'Backdrop', overridesResolver: (props, styles) => return styles.backdrop; , )( zIndex: -1, )` A backdrop component. This prop enables custom backdrop rendering. **Deprecated** Use `slots.backdrop` instead. While this prop currently works, it will be removed in the next major version.
- `BackdropProps` object  
Default is - Props applied to the Backdrop element. **Deprecated** Use `slotProps.backdrop` instead.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.

- `closeAfterTransition` `bool`  
Default is `FALSE` When set to `true` the Modal waits until a nested Transition is completed before closing.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `components` `{ Backdrop?: elementType, Root?: elementType }`  
Default is The components used for each slot inside. `Deprecated` Use the `slots` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `componentsProps` `{ backdrop?: func | object, root?: func | object }`  
Default is The extra props for the slot components. You can override the existing props or add new ones. `Deprecated` Use the `slotProps` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `container` `HTML element | func`  
Default is - An HTML element or function that returns one. The container will have the portal children appended to it. You can also provide a callback, which is called in a React layout effect. This lets you set the container from a ref, and also makes server-side rendering possible. By default, it uses the body of the top-level document object, so it's simply `document.body` most of the time.
- `disableAutoFocus` `bool`  
Default is `FALSE` If `true`, the modal will not automatically shift focus to itself when it opens, and replace it to the last focused element when it closes. This also works correctly with any modal children that have the `disableAutoFocus` prop. Generally this should never be set to `true` as it makes the modal less accessible to assistive technologies, like screen readers.
- `disableEnforceFocus` `bool`  
Default is `FALSE` If `true`, the modal will not prevent focus from leaving the modal while open. Generally this should never be set to `true` as it makes the modal less accessible to assistive technologies, like screen readers.
- `disableEscapeKeyDown` `bool`  
Default is `FALSE` If `true`, hitting escape will not fire the `onClose` callback.
- `disablePortal` `bool`  
Default is `FALSE` The children will be under the DOM hierarchy of the parent component.
- `disableRestoreFocus` `bool`  
Default is `FALSE` If `true`, the modal will not restore focus to previously focused element once modal is hidden or unmounted.
- `disableScrollLock` `bool`  
Default is `FALSE` Disable the scroll lock behavior.
- `hideBackdrop` `bool`  
Default is `FALSE` If `true`, the backdrop is not rendered.
- `keepMounted` `bool`  
Default is `FALSE` Always keep the children in the DOM. This prop can be useful in SEO situation or when you want to maximize the responsiveness of the Modal.
- `onClose` `func`  
Default is - Callback fired when the component requests to be closed. The `reason` parameter

can optionally be used to control the response to onClose. Signature: function(event: object, reason: string) = void. event The event source of the callback. reason Can be: "escapeKeyDown", "backdropClick".

- onTransitionEnter func  
Default is - A function called when a transition enters.
- onTransitionExited func  
Default is - A function called when a transition has exited.
- slotProps { backdrop?: func| object, root?: func| object }  
Default is The props used for each slot inside the Modal.
- slots { backdrop?: elementType, root?: elementType }  
Default is The components used for each slot inside the Modal. Either a string to use a HTML element or a component.
- sx Array func| object| bool | func| object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

### Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

### Note

This is an overlay surface wired as a click-reporter: input[[inputId]] holds a click count, and the wrapper does **not** manage the open state. Render it with open = TRUE/FALSE and toggle visibility from the server with updateModal.shinyInput(session, inputId, open = TRUE). For the common "open on click of a button" pattern, Modal.triggerId is simpler (open/close handled entirely client-side, no server logic).

---

Modal.triggerId	<i>Modal.triggerId</i>
-----------------	------------------------

---

### Description

Custom Modal bound to a DOM element by id. See 'js/src/MuiModalTriggerId.jsx'. Open/close state is managed entirely client-side.

### Usage

```
Modal.triggerId(triggerId, ...)
```

### Arguments

triggerId	HTML id of an existing DOM element that acts as the trigger to open the Modal.
...	Named arguments forwarded as React props, plus children to render inside the component.

**Value**

Object with ‘shiny.tag’ class suitable for use in the UI of a Shiny app.

---

`muiMaterialDependency` *muiMaterial JS dependency*

---

**Description**

muiMaterial JS dependency

**Usage**

```
muiMaterialDependency()
```

**Value**

HTML dependency object.

---

`muiMaterialExample` *Run muiMaterial example*

---

**Description**

Launch a Shiny example app or list the available examples. Use ‘`muiMaterial::muiMaterialExample("showcase")`’ to run a showcase app with all the components.

**Usage**

```
muiMaterialExample(example = NULL, ...)
```

**Arguments**

<code>example</code>	The name of the example to run, or ‘NULL’ to retrieve the list of examples.
<code>...</code>	Additional arguments to pass to ‘ <code>shiny::runApp()</code> ’.

**Details**

This function is adapted from ‘`runExample()`’ in the `shiny.blueprint` package.

**Value**

When ‘`example`’ is ‘NULL’, a character vector of the available example names. Otherwise the function launches a Shiny app and does not return; interrupt R to stop it (usually by pressing Ctrl+C or Esc).

---

muiMaterialPage

*HTML body without Bootstrap and margins*


---

### Description

Creates a Material UI page without Bootstrap and with 0 margin in body by default. You can choose to use Google Roboto font as well as Google icons fonts with the 'Icon()' component.

### Usage

```
muiMaterialPage(
  ...,
  useFontRoboto = FALSE,
  useMaterialIconsFilled = FALSE,
  useMaterialIconsOutlined = FALSE,
  useMaterialIconsRounded = FALSE,
  useMaterialIconsTwoTones = FALSE,
  suppressBootstrap = TRUE,
  styleBody = "margin:0",
  debugReact = FALSE
)
```

### Arguments

...	The contents of the document body.
useFontRoboto	Use Google Roboto font CDN in head, FALSE by default.
useMaterialIconsFilled	Use Google icons CDN in head to use 'Icon()' component, FALSE by default.
useMaterialIconsOutlined	Use Google icons CDN in head to use 'Icon()' component, FALSE by default.
useMaterialIconsRounded	Use Google icons CDN in head to use 'Icon()' component, FALSE by default.
useMaterialIconsTwoTones	Use Google icons CDN in head to use 'Icon()' component, FALSE by default.
suppressBootstrap	Whether to suppress Bootstrap. TRUE by default.
styleBody	CSS style in body, using 'margin:0' by default.
debugReact	Whether to enable react debug mode. FALSE by default.

### Details

The Bootstrap library is suppressed by default, as it doesn't work well with Material UI in general. The full set of available Material Icon names is at <<https://fonts.google.com/icons?icon.set=Material+Icons>>.

**Value**

html object with 'margin:0' which can be passed as the UI of a Shiny app.

**Examples**

```
library(shiny)
library(muiMaterial)

ui <- muiMaterialPage(
  useFontRoboto = TRUE,
  useMaterialIconsFilled = TRUE,
  Box(
    sx = list(p = 2),
    Typography("Hello Material UI!", variant = "h4"),
    Icon("home")
  )
)

shinyApp(ui, function(input, output, session) {})
```

---

NativeSelect

*NativeSelect*

---

**Description**

<https://mui.com/material-ui/api/native-select/>

**Usage**

```
NativeSelect(...)

NativeSelect.shinyInput(inputId, ..., value = defaultValue)

updateNativeSelect.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- children node  
Default is - The option elements to populate the select with. Can be some option elements.
- classes object  
Default is Override or extend the styles applied to the component. See CSS classes API below for more details.
- IconComponent elementType  
Default is ArrowDropDownIcon The icon that displays the arrow.
- input element  
Default is Input / An Input element; does not have to be a material-ui specific Input.
- inputProps object  
Default is - Attributes applied to the select element.
- onChange func  
Default is - Callback fired when a menu item is selected. Signature: function(event: React.ChangeEvent) = void event The event source of the callback. You can pull out the new value by accessing event.target.value (string).
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- value any  
Default is - The input value. The DOM API casts this to a string.
- variant 'filled' | 'outlined' | 'standard'  
Default is - The variant to use.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

NoSsr

*NoSsr*

---

**Description**

<https://mui.com/material-ui/api/no-ssr/>

**Usage**

NoSsr(...)

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - You can wrap a node.
- defer bool  
Default is FALSE If true, the component will not only prevent server-side rendering. It will also defer the rendering of the children into a different screen frame.
- fallback node  
Default is null The fallback content to display.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

OutlinedInput

*OutlinedInput*

---

**Description**

<https://mui.com/material-ui/api/outlined-input/>

**Usage**

```
OutlinedInput(...)
```

```
OutlinedInput.shinyInput(inputId, ..., value = defaultValue)
```

```
updateOutlinedInput.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- autoComplete string  
Default is - This prop helps users to fill forms faster, especially on mobile devices. The name can be confusing, as it's more like an autofill. You can learn more about it following the specification.

- `autoFocus` `bool`  
Default is - If true, the input element is focused during the first mount.
- `classes` `object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` `'primary' | 'secondary' | string`  
Default is - The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide. The prop defaults to the value ('primary') inherited from the parent FormControl component.
- `components` `{ Input?: elementType, Root?: elementType }`  
Default is The components used for each slot inside. Deprecated use the slots prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- `defaultValue` `any`  
Default is - The default value. Use when the component is not controlled.
- `disabled` `bool`  
Default is - If true, the component is disabled. The prop defaults to the value (false) inherited from the parent FormControl component.
- `endAdornment` `node`  
Default is - End InputAdornment for this component.
- `error` `bool`  
Default is - If true, the input will indicate an error. The prop defaults to the value (false) inherited from the parent FormControl component.
- `fullWidth` `bool`  
Default is FALSE If true, the input will take up the full width of its container.
- `id` `string`  
Default is - The id of the input element.
- `inputComponent` `elementType`  
Default is 'input' The component used for the input element. Either a string to use a HTML element or a component.
- `inputProps` `object`  
Default is Attributes applied to the input element.
- `inputRef` `ref`  
Default is - Pass a ref to the input element.
- `label` `node`  
Default is - The label of the input. It is only used for layout. The actual labelling is handled by InputLabel.
- `margin` `'dense' | 'none'`  
Default is - If dense, will adjust vertical spacing. This is normally obtained via context from FormControl. The prop defaults to the value ('none') inherited from the parent FormControl component.
- `maxRows` `number | string`  
Default is - Maximum number of rows to display when multiline option is set to true.

- `minRows` `number | string`  
Default is - Minimum number of rows to display when multiline option is set to true.
- `multiline` `bool`  
Default is `FALSE` If true, a `TextareaAutosize` element is rendered.
- `name` `string`  
Default is - Name attribute of the input element.
- `notched` `bool`  
Default is - If true, the outline is notched to accommodate the label.
- `onChange` `func`  
Default is - Callback fired when the value is changed. Signature: `function(event: React.ChangeEvent) = void` event The event source of the callback. You can pull out the new value by accessing `event.target.value` (string).
- `placeholder` `string`  
Default is - The short hint displayed in the input before the user enters a value.
- `readOnly` `bool`  
Default is - It prevents the user from changing the value of the field (not from interacting with the field).
- `required` `bool`  
Default is - If true, the input element is required. The prop defaults to the value (false) inherited from the parent `FormControl` component.
- `rows` `number | string`  
Default is - Number of rows to display when multiline option is set to true.
- `slots` { `input?: elementType`, `root?: elementType` }  
Default is The components used for each slot inside. This prop is an alias for the components prop, which will be deprecated in the future.
- `startAdornment` `node`  
Default is - Start `InputAdornment` for this component.
- `sx` `Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `type` `string`  
Default is `'text'` Type of the input element. It should be a valid HTML5 input type.
- `value` `any`  
Default is - The value of the input element, required for a controlled component.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 Pagination

*Pagination*


---

## Description

<https://mui.com/material-ui/api/pagination/>

## Usage

```
Pagination(...)
```

```
Pagination.shinyInput(inputId, ..., value = defaultValue)
```

```
updatePagination.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

## Arguments

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

## Details

- `boundaryCount` integer  
Default is 1 Number of always visible pages at the beginning and end.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` 'primary' | 'secondary' | 'standard' | string  
Default is 'standard' The active color. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- `count` integer  
Default is 1 The total number of pages.
- `defaultPage` integer  
Default is 1 The page selected by default when the component is uncontrolled.
- `disabled` bool  
Default is FALSE If true, the component is disabled.

- `getItemAriaLabel` func  
 Default is - Accepts a function which returns a string value that provides a user-friendly name for the current page. This is important for screen reader users. For localization purposes, you can use the provided translations. Signature: `function(type: string, page: number | null, selected: boolean) = string` The link or button type to format ('page' | 'first' | 'last' | 'next' | 'previous' | 'start-ellipsis' | 'end-ellipsis'). Defaults to 'page'. `page` The page number to format. `selected` If true, the current page is selected.
- `hideNextButton` bool  
 Default is FALSE If true, hide the next-page button.
- `hidePrevButton` bool  
 Default is FALSE If true, hide the previous-page button.
- `onChange` func  
 Default is - Callback fired when the page is changed. Signature: `function(event: React.ChangeEvent, page: number) = void` `event` The event source of the callback. `page` The page selected.
- `page` integer  
 Default is - The current page. Unlike `TablePagination`, which starts numbering from 0, this pagination starts from 1.
- `renderItem` func  
 Default is `(item) = PaginationItem ...item / Render the item`. Signature: `function(params: PaginationRenderItemParams) = ReactNode` `params` The props to spread on a `PaginationItem`.
- `shape` 'circular' | 'rounded'  
 Default is 'circular' The shape of the pagination items.
- `showFirstButton` bool  
 Default is FALSE If true, show the first-page button.
- `showLastButton` bool  
 Default is FALSE If true, show the last-page button.
- `siblingCount` integer  
 Default is 1 Number of always visible pages before and after the current page.
- `size` 'small' | 'medium' | 'large' | string  
 Default is 'medium' The size of the component.
- `sx` Array func | object | bool | func | object  
 Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant` 'outlined' | 'text' | string  
 Default is 'text' The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 PaginationItem

*PaginationItem*


---

## Description

<https://mui.com/material-ui/api/pagination-item/>

## Usage

```
PaginationItem(...)
```

## Arguments

... Props to pass to the component.

## Details

- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` `'primary' | 'secondary' | 'standard' | string`  
Default is `'standard'` The active color. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `components` `{ first?: elementType, last?: elementType, next?: elementType, previous?: elementType }`  
Default is The components used for each slot inside. This prop is an alias for the `slots` prop. It's recommended to use the `slots` prop instead. **Deprecated** use the `slots` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `disabled` `bool`  
Default is `FALSE` If true, the component is disabled.
- `page` `node`  
Default is - The current page number.
- `selected` `bool`  
Default is `FALSE` If true the pagination item is selected.
- `shape` `'circular' | 'rounded'`  
Default is `'circular'` The shape of the pagination item.
- `size` `'small' | 'medium' | 'large' | string`  
Default is `'medium'` The size of the component.
- `slotProps` `{ first?: func | object, last?: func | object, next?: func | object, previous?: func | object }`  
Default is The props used for each slot inside.

- `slots { first?: elementType, last?: elementType, next?: elementType, previous?: elementType }`  
Default is The components used for each slot inside.
- `sx Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `type 'end-ellipsis' | 'first' | 'last' | 'next' | 'page' | 'previous' | 'start-ellipsis'`  
Default is 'page' The type of pagination item.
- `variant 'outlined' | 'text' | string`  
Default is 'text' The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Paper

*Paper*

---

**Description**

<https://mui.com/material-ui/api/paper/>

**Usage**

`Paper(...)`

**Arguments**

`...` Props to pass to the component.

**Details**

- `children` node  
Default is - The content of the component.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `elevation` `integer`  
Default is 1 Shadow depth, corresponds to `dp` in the spec. It accepts values between 0 and 24 inclusive.
- `square` `bool`  
Default is `FALSE` If true, rounded corners are disabled.

- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant` `'elevation'` | `'outlined'` | `string`  
Default is `'elevation'` The variant to use.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Popover	<i>Popover</i>
---------	----------------

---

### Description

<https://mui.com/material-ui/api/popover/>

### Usage

```
Popover(...)
```

### Arguments

... Props to pass to the component.

### Details

- `open` `bool`  
Default is - If true, the component is shown.
- `action` `ref`  
Default is - A ref for imperative actions. It currently only supports `updatePosition()` action.
- `anchorEl` `HTML element` | `func`  
Default is - An HTML element, `PopoverVirtualElement`, or a function that returns either. It's used to set the position of the popover.
- `anchorOrigin` { `horizontal`: `'center'` | `'left'` | `'right'` | `number`, `vertical`: `'bottom'` | `'center'` | `'top'` | `number` }  
Default is `vertical`: `'top'`, `horizontal`: `'left'`, This is the point on the anchor where the popover's `anchorEl` will attach to. This is not used when the `anchorReference` is `'anchorPosition'`. Options: `vertical`: `top`, `center`, `bottom`; `horizontal`: `left`, `center`, `right`.
- `anchorPosition` { `left`: `number`, `top`: `number` }  
Default is - This is the position that may be used to set the position of the popover. The coordinates are relative to the application's client area.
- `anchorReference` `'anchorEl'` | `'anchorPosition'` | `'none'`  
Default is `'anchorEl'` This determines which anchor prop to refer to when setting the position of the popover.

- **BackdropComponent** elementType  
Default is styled(Backdrop, name: 'MuiModal', slot: 'Backdrop', overridesResolver: (props, styles) => return styles.backdrop; , )( zIndex: -1, ) A backdrop component. This prop enables custom backdrop rendering. Deprecated Use slots.backdrop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- **BackdropProps** object  
Default is - Props applied to the Backdrop element. Deprecated Use slotProps.backdrop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- **children** node  
Default is - The content of the component.
- **classes** object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **container** HTML element | func  
Default is - An HTML element, component instance, or function that returns either. The container will be passed to the Modal component. By default, it uses the body of the anchorElement's top-level document object, so it's simply document.body most of the time.
- **disableScrollLock** bool  
Default is FALSE Disable the scroll lock behavior.
- **elevation** integer  
Default is 8 The elevation of the popover.
- **marginThreshold** number  
Default is 16 Specifies how close to the edge of the window the popover can appear. If null, the popover will not be constrained by the window.
- **onClose** func  
Default is - Callback fired when the component requests to be closed. The reason parameter can optionally be used to control the response to onClose.
- **PaperProps** { component?: element type }  
Default is Props applied to the Paper element. This prop is an alias for slotProps.paper and will be overridden by it if both are used. Deprecated Use slotProps.paper instead.
- **slotProps** { backdrop?: func | object, paper?: func | object, root?: func | object, transition?: func | object }  
Default is The props used for each slot inside.
- **slots** { backdrop?: elementType, paper?: elementType, root?: elementType, transition?: elementType }  
Default is The components used for each slot inside.
- **sx** Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- **transformOrigin** { horizontal: 'center' | 'left' | 'right' | number, vertical: 'bottom' | 'center' | 'top' }  
Default is vertical: 'top', horizontal: 'left', This is the point on the popover which will attach to the anchor's origin. Options: vertical: top, center, bottom, x(px); horizontal: left, center, right, x(px).
- **TransitionComponent** elementType  
Default is Grow The component used for the transition. Follow this guide to learn more about

the requirements for this component. Deprecated use the `slots.transition` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.

- `transitionDuration` 'auto' | number | { appear?: number, enter?: number, exit?: number }  
Default is 'auto' Set to 'auto' to automatically calculate transition time based on height.
- `TransitionProps` object  
Default is `Props` applied to the transition element. By default, the element is based on this `Transition` component. Deprecated use the `slotProps.transition` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

<code>Popover.triggerId</code>	<i>Popover.triggerId</i>
--------------------------------	--------------------------

---

### Description

Custom Popover bound to a DOM element by id. See `'js/src/MuiPopoverTriggerId.jsx'`. The trigger element acts as the anchor; the Popover opens on click and closes on clickaway.

### Usage

```
Popover.triggerId(triggerId, ...)
```

### Arguments

<code>triggerId</code>	HTML id of an existing DOM element that acts as the anchor/trigger for the Popover.
<code>...</code>	Named arguments forwarded as React props, plus children to render inside the component.

### Value

Object with `'shiny.tag'` class suitable for use in the UI of a Shiny app.

Popper

*Popper*

**Description**

<https://mui.com/material-ui/api/popper/>

**Usage**

Popper(...)

**Arguments**

... Props to pass to the component.

**Details**

- open bool  
Default is - If true, the component is shown.
- anchorEl HTML element | object | func  
Default is - An HTML element, virtualElement, or a function that returns either. It's used to set the position of the popper. The return value will be passed as the reference object of the Popper instance.
- children node | func  
Default is - Popper render function or node.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- components { Root?: elementType }  
Default is The components used for each slot inside the Popper. Either a string to use a HTML element or a component. *Deprecated* use the slots prop instead. This prop will be removed in a future major release. *How to migrate.*
- componentsProps { root?: func | object }  
Default is The props used for each slot inside the Popper. *Deprecated* use the slotProps prop instead. This prop will be removed in a future major release. *How to migrate.*
- container HTML element | func  
Default is - An HTML element or function that returns one. The container will have the portal children appended to it. You can also provide a callback, which is called in a React layout effect. This lets you set the container from a ref, and also makes server-side rendering possible. By default, it uses the body of the top-level document object, so it's simply document.body most of the time.
- disablePortal bool  
Default is FALSE The children will be under the DOM hierarchy of the parent component.

- `keepMounted` `bool`  
Default is `FALSE` Always keep the children in the DOM. This prop can be useful in SEO situation or when you want to maximize the responsiveness of the Popper.
- `modifiers` `Array { data?: object, effect?: func, enabled?: bool, fn?: func, name?: any, options?: object }`  
Default is - Popper.js is based on a "plugin-like" architecture, most of its features are fully encapsulated "modifiers". A modifier is a function that is called each time Popper.js needs to compute the position of the popper. For this reason, modifiers should be very performant to avoid bottlenecks. To learn how to create a modifier, read the modifiers documentation.
- `placement` `'auto-end' | 'auto-start' | 'auto' | 'bottom-end' | 'bottom-start' | 'bottom' | 'left-end' | 'left-start' | 'left' | 'right-end' | 'right-start' | 'right' | 'top-end' | 'top-start' | 'top'`  
Default is `'bottom'` Popper placement.
- `popperOptions` `{ modifiers?: array, onFirstUpdate?: func, placement?: 'auto-end' | 'auto-start' | 'auto' }`  
Default is Options provided to the Popper.js instance.
- `popperRef` `ref`  
Default is - A ref that points to the used popper instance.
- `slotProps` `{ root?: func | object }`  
Default is The props used for each slot inside the Popper.
- `slots` `{ root?: elementType }`  
Default is The components used for each slot inside the Popper. Either a string to use a HTML element or a component.
- `sx` `Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `transition` `bool`  
Default is `FALSE` Help supporting a `react-transition-group/Transition` component.

## Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Portal

*Portal*

---

## Description

<https://mui.com/material-ui/api/portal/>

## Usage

```
Portal(...)
```

## Arguments

... Props to pass to the component.

**Details**

- children node  
Default is - The children to render into the container.
- container HTML element| func  
Default is - An HTML element or function that returns one. The container will have the portal children appended to it. You can also provide a callback, which is called in a React layout effect. This lets you set the container from a ref, and also makes server-side rendering possible. By default, it uses the body of the top-level document object, so it's simply document.body most of the time.
- disablePortal bool  
Default is FALSE The children will be under the DOM hierarchy of the parent component.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

print.muiMaterial      *Print muiMaterial components*

---

**Description**

When called interactively, renders the component in the IDE viewer panel. Otherwise, falls back to standard shiny.tag printing (raw HTML text).

**Usage**

```
## S3 method for class 'muiMaterial'
print(x, browse = interactive(), ...)
```

**Arguments**

x	A muiMaterial object (also inherits shiny.tag).
browse	Whether to render in viewer. Defaults to TRUE in interactive sessions.
...	Additional arguments passed to print.

**Value**

Invisibly returns x.

Radio

*Radio***Description**

<https://mui.com/material-ui/api/radio/>

**Usage**

```
Radio(...)
```

```
Radio.shinyInput(inputId, ..., value = defaultValue)
```

```
updateRadio.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- checked bool  
Default is - If true, the component is checked.
- checkedIcon node  
Default is `RadioButtonIcon checked` / The icon to display when the component is checked.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- color 'default' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'primary' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- disabled bool  
Default is - If true, the component is disabled.
- disableRipple bool  
Default is `FALSE` If true, the ripple effect is disabled.
- icon node  
Default is `RadioButtonIcon` / The icon to display when the component is unchecked.

- `id` string  
Default is - The id of the input element.
- `inputProps` object  
Default is - Attributes applied to the input element. **Deprecated** Use `slotProps.input` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `inputRef` `ref`  
Default is - Pass a ref to the input element. **Deprecated** Use `slotProps.input.ref` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `name` string  
Default is - Name attribute of the input element.
- `onChange` func  
Default is - Callback fired when the state is changed. Signature: `function(event: React.ChangeEvent) = void` event The event source of the callback. You can pull out the new value by accessing `event.target.value` (string). You can pull out the new checked state by accessing `event.target.checked` (boolean).
- `required` bool  
Default is `FALSE` If true, the input element is required.
- `size` `'medium'` | `'small'` | string  
Default is `'medium'` The size of the component. `small` is equivalent to the dense radio styling.
- `slotProps` { `input?: func` | object, `root?: func` | object }  
Default is The props used for each slot inside.
- `slots` { `input?: elementType`, `root?: elementType` }  
Default is The components used for each slot inside.
- `sx` Array `func` | object | bool | `func` | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the [sx](#) page for more details.
- `value` any  
Default is - The value of the component. The DOM API casts this to a string.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

**Note**

A standalone `Radio.shinyInput` reports a boolean (checked) to the server, not the selected option string. For mutually-exclusive option groups use [RadioGroup.shinyInput](#) instead, which reports the string value of the selected radio.

**See Also**

[RadioGroup.shinyInput](#)

RadioGroup

*RadioGroup***Description**

<https://mui.com/material-ui/api/radio-group/>

**Usage**

```
RadioGroup(...)

RadioGroup.shinyInput(inputId, ..., value = defaultValue)

updateRadioGroup.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- children node  
Default is NA The content of the component.
- defaultValue any  
Default is NA The default value. Use when the component is not controlled.
- name string  
Default is NA The name used to reference the value of the control. If you don't provide this prop, it falls back to a randomly generated name.
- onChange func  
Default is NA Callback fired when a radio button is selected. Signature: function(event: React.ChangeEvent, value: string) = void event The event source of the callback. value The value of the selected radio button. You can pull out the new value by accessing event.target.value (string).
- value any  
Default is NA Value of the selected radio button. The DOM API casts this to a string.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

Rating

*Rating*


---

**Description**

<https://mui.com/material-ui/api/rating/>

**Usage**

```
Rating(...)
```

```
Rating.shinyInput(inputId, ..., value = defaultValue)
```

```
updateRating.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- **classes** object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **component** elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- **defaultValue** number  
Default is null The default value. Use when the component is not controlled.
- **disabled** bool  
Default is FALSE If true, the component is disabled.
- **emptyIcon** node  
Default is StarBorder fontSize="inherit" / The icon to display when empty.
- **emptyLabelText** node  
Default is 'Empty' The label read when the rating input is empty.

- `getLabelText` func  
Default is function `defaultLabelText(value) return $value || '0' Star$value !== 1 ? 's' : ''`;  
Accepts a function which returns a string value that provides a user-friendly name for the current value of the rating. This is important for screen reader users. For localization purposes, you can use the provided translations. Signature: `function(value: number) = string` value The rating label's value to format.
- `highlightSelectedOnly` bool  
Default is `FALSE` If true, only the selected icon will be highlighted.
- `icon` node  
Default is `Star` `fontSize="inherit"` / The icon to display.
- `IconContainerComponent` elementType  
Default is function `IconContainer(props) const value, ...other = props; return span ...other /` ;  
The component containing the icon. **Deprecated** Use `slotProps.icon.component` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `max` number  
Default is `5` Maximum rating.
- `name` string  
Default is - The name attribute of the radio input elements. This input name should be unique within the page. Being unique within a form is insufficient since the name is used to generate IDs.
- `onChange` func  
Default is - Callback fired when the value changes. Signature: `function(event: React.SyntheticEvent, value: number | null) = void` event The event source of the callback. value The new value.
- `onChangeActive` func  
Default is - Callback function that is fired when the hover state changes. Signature: `function(event: React.SyntheticEvent, value: number) = void` event The event source of the callback. value The new value.
- `precision` number  
Default is `1` The minimum increment value change allowed.
- `readOnly` bool  
Default is `FALSE` Removes all hover effects and pointer events.
- `size` 'small' | 'medium' | 'large' | string  
Default is 'medium' The size of the component.
- `slotProps` { decimal?: func | object, icon?: func | object, label?: func | object, root?: func | object }  
Default is The props used for each slot inside.
- `slots` { decimal?: elementType, icon?: elementType, label?: elementType, root?: elementType }  
Default is The components used for each slot inside.
- `sx` Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the [sx](#) page for more details.
- `value` number  
Default is - The rating value.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

ScopedCssBaseline	<i>ScopedCssBaseline</i>
-------------------	--------------------------

---

**Description**

<https://mui.com/material-ui/api/scoped-css-baseline/>

**Usage**

```
ScopedCssBaseline(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children` node  
Default is NA The content of the component.
- `classes` object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component` `elementType`  
Default is NA The component used for the root node. Either a string to use a HTML element or a component.
- `enableColorScheme` `bool`  
Default is NA Enable color-scheme CSS property to use `theme.palette.mode`. For more details, check out <https://developer.mozilla.org/en-US/docs/Web/CSS/color-scheme> For browser support, check out <https://caniuse.com/?search=color-scheme>
- `sx` `Array` `func` | `object` | `bool` | `func` | `object`  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 Select
 

---

*Select***Description**

<https://mui.com/material-ui/api/select/>

**Usage**

```
Select(...)
```

```
Select.shinyInput(inputId, ..., value = defaultValue)
```

```
updateSelect.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- `autoWidth` `bool`  
Default is `FALSE` If true, the width of the popover will automatically be set according to the items inside the menu, otherwise it will be at least the width of the select input.
- `children` `node`  
Default is - The option elements to populate the select with. Can be some `MenuItem` when `native` is false and `option` when `native` is true. The `MenuItem` elements must be direct descendants when `native` is false.
- `classes` `object`  
Default is `Override` or `extend` the styles applied to the component. See `CSS classes API` below for more details.
- `defaultOpen` `bool`  
Default is `FALSE` If true, the component is initially open. Use when the component open state is not controlled (i.e. the `open` prop is not defined). You can only use it when the `native` prop is false (default).
- `defaultValue` `any`  
Default is - The default value. Use when the component is not controlled.

- `displayEmpty` `bool`  
Default is `FALSE` If true, a value is displayed even if no items are selected. In order to display a meaningful value, a function can be passed to the `renderValue` prop which returns the value to be displayed when no items are selected. When using this prop, make sure the label doesn't overlap with the empty displayed value. The label should either be hidden or forced to a shrunk state.
- `IconComponent` `elementType`  
Default is `ArrowDropDownIcon` The icon that displays the arrow.
- `id` `string`  
Default is - The id of the wrapper element or the select element when native.
- `input` `element`  
Default is - An Input element; does not have to be a material-ui specific Input.
- `inputProps` `object`  
Default is - Attributes applied to the input element. When native is true, the attributes are applied on the select element.
- `label` `node`  
Default is - See `OutlinedInput#label`
- `labelId` `string`  
Default is - The ID of an element that acts as an additional label. The Select will be labelled by the additional label and the selected value.
- `MenuProps` `object`  
Default is - Props applied to the Menu element.
- `multiple` `bool`  
Default is `FALSE` If true, value must be an array and the menu will support multiple selections.
- `native` `bool`  
Default is `FALSE` If true, the component uses a native select element.
- `onChange` `func`  
Default is - Callback fired when a menu item is selected. Signature: `function(event: SelectChangeEvent, child?: object) = void` event The event source of the callback. You can pull out the new value by accessing `event.target.value` (any). Warning: This is a generic event, not a change event, unless the change event is caused by browser autofill. `child` The react element that was selected when native is false (default).
- `onClose` `func`  
Default is - Callback fired when the component requests to be closed. Use it in either controlled (see the open prop), or uncontrolled mode (to detect when the Select collapses). Signature: `function(event: object) = void` event The event source of the callback.
- `onOpen` `func`  
Default is - Callback fired when the component requests to be opened. Use it in either controlled (see the open prop), or uncontrolled mode (to detect when the Select expands). Signature: `function(event: object) = void` event The event source of the callback.
- `open` `bool`  
Default is - If true, the component is shown. You can only use it when the native prop is false (default).

- `renderValue` func  
Default is - Render the selected value. You can only use it when the `native` prop is false (default). Signature: `function(value: any) = ReactNode` The value provided to the component.
- `SelectDisplayProps` object  
Default is - Props applied to the clickable div element.
- `sx` Array  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `value` '' | any  
Default is - The input value. Providing an empty string will select no options. Set to an empty string '' if you don't want any of the available options to be selected. If the value is an object it must have reference equality with the option in order to be selected. If the value is not an object, the string representation must match with the string representation of the option in order to be selected.
- `variant` 'filled' | 'outlined' | 'standard'  
Default is 'outlined' The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Skeleton

*Skeleton*

---

**Description**

<https://mui.com/material-ui/api/skeleton/>

**Usage**

`Skeleton(...)`

**Arguments**

... Props to pass to the component.

**Details**

- `animation` 'pulse' | 'wave' | false  
Default is 'pulse' The animation. If false the animation effect is disabled.
- `children` node  
Default is - Optional children to infer width and height from.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.

- `component elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `height number | string`  
Default is - Height of the skeleton. Useful when you don't want to adapt the skeleton to a text element but for instance a card.
- `sx Array`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant 'circular' | 'rectangular' | 'rounded' | 'text' | string`  
Default is 'text' The type of content that will be rendered.
- `width number | string`  
Default is - Width of the skeleton. Useful when the skeleton is inside an inline element with no width of its own.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Slide

*Slide*

---

**Description**

<https://mui.com/material-ui/api/slide/>

**Usage**

`Slide(...)`

**Arguments**

`...` Props to pass to the component.

**Details**

- `children element`  
Default is - A single child content element. This needs to be able to hold a ref.
- `addEndListener func`  
Default is - Add a custom transition end trigger. Called with the transitioning DOM node and a done callback. Allows for more fine grained transition end logic. Note: Timeouts are still used as a fallback if provided.
- `appear bool`  
Default is TRUE Perform the enter transition when it first mounts if `in` is also true. Set this to false to disable this behavior.

- `container HTML element | func`  
Default is - An HTML element, or a function that returns one. It's used to set the container the Slide is transitioning from.
- `direction 'down' | 'left' | 'right' | 'up'`  
Default is 'down' Direction the child node will enter from.
- `easing { enter?: string, exit?: string } | string`  
Default is enter: `theme.transitions.easing.easeOut`, exit: `theme.transitions.easing.sharp`, The transition timing function. You may specify a single easing or a object containing enter and exit values.
- `in bool`  
Default is - If true, the component will transition in.
- `timeout number | { appear?: number, enter?: number, exit?: number }`  
Default is enter: `theme.transitions.duration.enteringScreen`, exit: `theme.transitions.duration.leavingScreen`, The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 Slider

*Slider*


---

**Description**

<https://mui.com/material-ui/api/slider/>

**Usage**

```
Slider(...)
```

```
Slider.shinyInput(inputId, ..., value = defaultValue)
```

```
updateSlider.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

<code>...</code>	Props to pass to the component.
<code>inputId</code>	ID of the component.
<code>value</code>	Starting value.
<code>session</code>	Object passed as the 'session' argument to Shiny server.

## Details

- `aria-label` string  
Default is - The label of the slider.
- `aria-labelledby` string  
Default is - The id of the element containing a label for the slider.
- `aria-valuetext` string  
Default is - A string value that provides a user-friendly name for the current value of the slider.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'primary' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- `components` { `Input?: elementType`, `Mark?: elementType`, `MarkLabel?: elementType`, `Rail?: elementType`,  
Default is The components used for each slot inside. Deprecated use the `slots` prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- `componentsProps` { `input?: func | object`, `mark?: func | object`, `markLabel?: func | object`, `rail?: func |`  
Default is The extra props for the slot components. You can override the existing props or add new ones. Deprecated use the `slotProps` prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- `defaultValue` Array  
Default is - The default value. Use when the component is not controlled.
- `disabled` bool  
Default is FALSE If true, the component is disabled.
- `disableSwap` bool  
Default is FALSE If true, the active thumb doesn't swap when moving pointer over a thumb while dragging another thumb.
- `getAriaLabel` func  
Default is - Accepts a function which returns a string value that provides a user-friendly name for the thumb labels of the slider. This is important for screen reader users. Signature: `function(index: number) = string` index The thumb label's index to format.
- `getAriaValueText` func  
Default is - Accepts a function which returns a string value that provides a user-friendly name for the current value of the slider. This is important for screen reader users. Signature: `function(value: number, index: number) = string` value The thumb label's value to format. index The thumb label's index to format.
- `marks` Array { `label?: node`, `value: number` } | bool  
Default is FALSE Marks indicate predetermined values to which the user can move the slider. If true the marks are spaced according to the value of the `step` prop. If an array, it should contain objects with `value` and an optional `label` key.
- `max` number  
Default is 100 The maximum allowed value of the slider. Should not be equal to min.

- **min number**  
Default is 0 The minimum allowed value of the slider. Should not be equal to max.
- **name string**  
Default is - Name attribute of the hidden input element.
- **onChange func**  
Default is - Callback function that is fired when the slider's value changed. Signature: `function(event: Event, value: Value, activeThumb: number) = void` event The event source of the callback. You can pull out the new value by accessing `event.target.value` (any). Warning: This is a generic event not a change event. value The new value. activeThumb Index of the currently moved thumb.
- **onChangeCommitted func**  
Default is - Callback function that is fired when the mouseup is triggered. Signature: `function(event: React.SyntheticEvent | Event, value: Value) = void` event The event source of the callback. Warning: This is a generic event not a change event. value The new value.
- **orientation 'horizontal' | 'vertical'**  
Default is 'horizontal' The component orientation.
- **scale func**  
Default is `function Identity(x) return x`; A transformation function, to change the scale of the slider. Signature: `function(x: any) = any`
- **shiftStep number**  
Default is 10 The granularity with which the slider can step through values when using Page Up/Page Down or Shift + Arrow Up/Arrow Down.
- **size 'small' | 'medium' | string**  
Default is 'medium' The size of the slider.
- **slotProps { input?: func | object, mark?: func | object, markLabel?: func | object, rail?: func | object, ... }**  
Default is The props used for each slot inside the Slider.
- **slots { input?: elementType, mark?: elementType, markLabel?: elementType, rail?: elementType, root?: ... }**  
Default is The components used for each slot inside the Slider. Either a string to use a HTML element or a component.
- **step number**  
Default is 1 The granularity with which the slider can step through values. (A "discrete" slider.) The min prop serves as the origin for the valid values. We recommend  $(\text{max} - \text{min})$  to be evenly divisible by the step. When step is null, the thumb can only be slid onto marks provided with the marks prop.
- **sx Array**  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- **tabIndex number**  
Default is - Tab index attribute of the hidden input element.
- **track 'inverted' | 'normal' | false**  
Default is 'normal' The track presentation: normal the track will render a bar representing the slider value. inverted the track will render a bar representing the remaining slider value. false the track will render without a bar.
- **value Array**  
Default is - The value of the slider. For ranged sliders, provide an array with two values.

- `valueLabelDisplay` 'auto' | 'off' | 'on'  
Default is 'off' Controls when the value label is displayed: auto the value label will display when the thumb is hovered or focused. on will display persistently. off will never display.
- `valueLabelFormat` `func` | `string`  
Default is function `Identity(x)` return `x`; The format function the value label's value. When a function is provided, it should have the following signature:- `number value` The value label's value to format - `number index` The value label's index to format

## Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

## Examples

```
library(shiny)
library(muiMaterial)

ui <- muiMaterialPage(
  Slider.shinyInput("s", value = 30, min = 0, max = 100),
  verbatimTextOutput("out")
)

server <- function(input, output, session) {
  output$out <- renderPrint(input$s)
}

shinyApp(ui, server)
```

---

Snackbar

*Snackbar*

---

## Description

<https://mui.com/material-ui/api/snackbar/>

## Usage

```
Snackbar(...)

Snackbar.shinyInput(inputId, ...)

updateSnackbar.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- **action node**  
Default is - The action to display. It renders after the message, at the end of the snackbar.
- **anchorOrigin { horizontal: 'center' | 'left' | 'right', vertical: 'bottom' | 'top' }**  
Default is vertical: 'bottom', horizontal: 'left' The anchor of the Snackbar. On smaller screens, the component grows to occupy all the available width, the horizontal alignment is ignored.
- **autoHideDuration number**  
Default is null The number of milliseconds to wait before automatically calling the onClose function. onClose should then set the state of the open prop to hide the Snackbar. This behavior is disabled by default with the null value.
- **children element**  
Default is - Replace the SnackbarContent component.
- **classes object**  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **ClickAwayListenerProps object**  
Default is - Props applied to the ClickAwayListener element. **Deprecated** Use slotProps.clickAwayListener instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- **ContentProps object**  
Default is - Props applied to the SnackbarContent element. **Deprecated** Use slotProps.content instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- **disableWindowBlurListener bool**  
Default is FALSE If true, the autoHideDuration timer will expire even if the window is not focused.
- **key any**  
Default is - When displaying multiple consecutive snackbars using a single parent-rendered Snackbar, add the key prop to ensure independent treatment of each message. For instance, use Snackbar key=message. Otherwise, messages might update in place, and features like autoHideDuration could be affected.
- **message node**  
Default is - The message to display.
- **onClose func**  
Default is - Callback fired when the component requests to be closed. Typically onClose is used to set state in the parent component, which is used to control the Snackbar open prop. The reason parameter can optionally be used to control the response to onClose, for example ignoring clickaway. Signature: function(event: React.SyntheticEvent | Event, reason: string)

= voidevent The event source of the callback.reason Can be: "timeout" (autoHideDuration expired), "clickaway", or "escapeKeyDown".

- open bool  
Default is - If true, the component is shown.
- resumeHideDuration number  
Default is - The number of milliseconds to wait before dismissing after user interaction. If autoHideDuration prop isn't specified, it does nothing. If autoHideDuration prop is specified but resumeHideDuration isn't, we default to autoHideDuration / 2 ms.
- slotProps { clickAwayListener?: func | { children: element, disableReactTree?: bool, mouseEvent?: 'on...'  
Default is The props used for each slot inside.
- slots { clickAwayListener?: elementType, content?: elementType, root?: elementType, transition?: el...  
Default is The components used for each slot inside.
- sx Array  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- TransitionComponent elementType  
Default is Grow The component used for the transition. Follow this guide to learn more about the requirements for this component. **Deprecated** Use slots.transition instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- transitionDuration number | { appear?: number, enter?: number, exit?: number }  
Default is enter: theme.transitions.duration.enteringScreen, exit: theme.transitions.duration.leavingScreen, The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object.
- TransitionProps object  
Default is Props applied to the transition element. By default, the element is based on this Transition component. **Deprecated** Use slotProps.transition instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.

## Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

## Note

This is an overlay surface wired as a click-reporter: input[[inputId]] holds a click count, and the wrapper does **not** manage the open state. Render it with open = TRUE/FALSE and toggle visibility from the server with updateSnackbar.shinyInput(session, inputId, open = TRUE) (e.g. open it in response to another event, then close it from the Snackbar's autoHideDuration/onClose).

---

SnackbarContent

*SnackbarContent*

---

## Description

<https://mui.com/material-ui/api/snackbar-content/>

**Usage**

```
SnackbarContent(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- **action node**  
Default is - The action to display. It renders after the message, at the end of the snackbar.
- **classes object**  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **message node**  
Default is - The message to display.
- **role string**  
Default is 'alert' The ARIA role attribute of the element.
- **sx Array**  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

SpeedDial

*SpeedDial*

---

**Description**

<https://mui.com/material-ui/api/speed-dial/>

**Usage**

```
SpeedDial(...)
```

**Arguments**

... Props to pass to the component.

## Details

- `ariaLabel` string  
Default is - The aria-label of the button element. Also used to provide the id for the SpeedDial element and its children.
- `children` node  
Default is - SpeedDialActions to display when the SpeedDial is open.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `direction` 'down' | 'left' | 'right' | 'up'  
Default is 'up' The direction the actions open relative to the floating action button.
- `FabProps` object  
Default is Props applied to the Fab element.
- `hidden` bool  
Default is FALSE If true, the SpeedDial is hidden.
- `icon` node  
Default is - The icon to display in the SpeedDial Fab. The SpeedDialIcon component provides a default Icon with animation.
- `onClose` func  
Default is - Callback fired when the component requests to be closed. Signature: `function(event: object, reason: string) = void` event The event source of the callback. reason Can be: "toggle", "blur", "mouseLeave", "escapeKeyDown".
- `onOpen` func  
Default is - Callback fired when the component requests to be open. Signature: `function(event: object, reason: string) = void` event The event source of the callback. reason Can be: "toggle", "focus", "mouseEnter".
- `open` bool  
Default is - If true, the component is shown.
- `openIcon` node  
Default is - The icon to display in the SpeedDial Fab when the SpeedDial is open.
- `slotProps` { `root?: func | object`, `transition?: func | object` }  
Default is The props used for each slot inside.
- `slots` { `root?: elementType`, `transition?: elementType` }  
Default is The components used for each slot inside.
- `sx` Array  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- `TransitionComponent` elementType  
Default is Zoom \* @deprecated Use `slots.transition` instead. This prop will be removed in a future major release. The component used for the transition. Follow this guide to learn more about the requirements for this component. @deprecated Use `slots.transition` instead. This prop will be removed in a future major release. How to migrate

- `transitionDuration` `number` | `{ appear?: number, enter?: number, exit?: number }`  
Default is `enter: theme.transitions.duration.enteringScreen, exit: theme.transitions.duration.leavingScreen`,  
The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object.
- `TransitionProps` object  
Default is - Props applied to the transition element. By default, the element is based on this Transition component. **Deprecated** Use `slotProps.transition` instead. This prop will be removed in a future major release. [How to migrate](#)

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

SpeedDialAction	<i>SpeedDialAction</i>
-----------------	------------------------

---

**Description**

<https://mui.com/material-ui/api/speed-dial-action/>

**Usage**

```
SpeedDialAction(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `delay` `number`  
Default is 0 Adds a transition delay, to allow a series of SpeedDialActions to be animated.
- `FabProps` object  
Default is Props applied to the Fab component. **Deprecated** Use `slotProps.fab` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `icon` `node`  
Default is - The icon to display in the SpeedDial Fab.
- `id` `string`  
Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.
- `open` `bool`  
Default is - If true, the component is shown.

- `slotProps` { `fab?: func | object`, `staticTooltip?: func | object`, `staticTooltipLabel?: func | object`, `tooltips?: func | object` }  
Default is The props used for each slot inside.
- `slots` { `fab?: elementType`, `staticTooltip?: elementType`, `staticTooltipLabel?: elementType`, `tooltips?: elementType` }  
Default is The components used for each slot inside.
- `sx` Array  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `TooltipClasses` object  
Default is - classes prop applied to the Tooltip element. **Deprecated** Use `slotProps.tooltip.classes` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `tooltipOpen` bool  
Default is `FALSE` Make the tooltip always visible when the SpeedDial is open. **Deprecated** Use `slotProps.tooltip.open` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `tooltipPlacement` `'auto-end' | 'auto-start' | 'auto' | 'bottom-end' | 'bottom-start' | 'bottom' | 'left-end' | 'left-start' | 'left' | 'right-end' | 'right-start' | 'right' | 'top-end' | 'top-start' | 'top'`  
Default is `'left'` Placement of the tooltip. **Deprecated** Use `slotProps.tooltip.placement` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `tooltipTitle` node  
Default is - Label to display in the tooltip. **Deprecated** Use `slotProps.tooltip.title` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

SpeedDialIcon

*SpeedDialIcon*

---

**Description**

<https://mui.com/material-ui/api/speed-dial-icon/>

**Usage**

```
SpeedDialIcon(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- **classes** object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- **icon** node  
Default is NA The icon to display.
- **openIcon** node  
Default is NA The icon to display in the SpeedDial Floating Action Button when the SpeedDial is open.
- **sx** Array func| object| bool | func| object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

Stack

*Stack*


---

**Description**

<https://mui.com/material-ui/api/stack/>

**Usage**

```
Stack(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- **children** node  
Default is - The content of the component.
- **component** elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- **direction** 'column-reverse'| 'column'| 'row-reverse'| 'row'| Array 'column-reverse'| 'column'| 'row-reverse'| 'row'  
Default is 'column' Defines the flex-direction style property. It is applied for all screen sizes.
- **divider** node  
Default is - Add an element between each child.
- **spacing** Array number| string | number| object| string  
Default is 0 Defines the space between immediate children.

- `sx` Array `func| object| bool | func| object`  
Default is - The system prop, which allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `useFlexGap` bool  
Default is `FALSE` If true, the CSS flexbox `gap` is used instead of applying margin to children. While CSS `gap` removes the known limitations, it is not fully supported in some browsers. We recommend checking <https://caniuse.com/?search=flex%20gap> before using this flag. To enable this flag globally, follow the theme's default props configuration.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Step	<i>Step</i>
------	-------------

---

**Description**

<https://mui.com/material-ui/api/step/>

**Usage**

`Step(...)`

**Arguments**

`...` Props to pass to the component.

**Details**

- `active` bool  
Default is - Sets the step as active. Is passed to child components.
- `children` node  
Default is - Should be Step sub-components such as `StepLabel`, `StepContent`.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `completed` bool  
Default is - Mark the step as completed. Is passed to child components.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `disabled` bool  
Default is - If true, the step is disabled, will also disable the button if `StepButton` is a child of `Step`. Is passed to child components.

- `expanded` bool  
Default is `FALSE` Expand the step.
- `index` integer  
Default is - The position of the step. The prop defaults to the value inherited from the parent Stepper component.
- `last` bool  
Default is - If true, the Step is displayed as rendered last. The prop defaults to the value inherited from the parent Stepper component.
- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

StepButton

*StepButton*

---

### Description

<https://mui.com/material-ui/api/step-button/>

### Usage

```
StepButton(...)
```

```
StepButton.shinyInput(inputId, ...)
```

```
updateStepButton.shinyInput(  
  session = shiny::getDefaultReactiveDomain(),  
  inputId,  
  ...  
)
```

### Arguments

<code>...</code>	Props to pass to the component.
<code>inputId</code>	ID of the component.
<code>session</code>	Object passed as the 'session' argument to Shiny server.

**Details**

- children node  
Default is NA Can be a StepLabel or a node to place inside StepLabel as children.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- icon node  
Default is NA The icon displayed by the step label.
- optional node  
Default is NA The optional node to display.
- sx Array func| object| bool | func| object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

StepConnector

*StepConnector*

---

**Description**

<https://mui.com/material-ui/api/step-connector/>

**Usage**

StepConnector(...)

**Arguments**

... Props to pass to the component.

**Details**

- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func| object| bool | func| object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

StepContent

*StepContent***Description**

<https://mui.com/material-ui/api/step-content/>

**Usage**

```
StepContent(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- slotProps { transition?: func | object }  
Default is The props used for each slot inside.
- slots { transition?: elementType }  
Default is The components used for each slot inside.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- TransitionComponent elementType  
Default is Collapse The component used for the transition. Follow this guide to learn more about the requirements for this component. **Deprecated** Use slots.transition instead. This prop will be removed in a future major release. How to migrate.
- transitionDuration 'auto' | number | { appear?: number, enter?: number, exit?: number }  
Default is 'auto' Adjust the duration of the content expand transition. Passed as a prop to the transition component. Set to 'auto' to automatically calculate transition time based on height.
- TransitionProps object  
Default is - Props applied to the transition element. By default, the element is based on this Transition component. **Deprecated** Use slotProps.transition instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

StepIcon

*StepIcon*

---

## Description

<https://mui.com/material-ui/api/step-icon/>

## Usage

```
StepIcon(...)
```

## Arguments

... Props to pass to the component.

## Details

- active bool  
Default is FALSE Whether this step is active.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- completed bool  
Default is FALSE Mark the step as completed. Is passed to child components.
- error bool  
Default is FALSE If true, the step is marked as failed.
- icon node  
Default is - The label displayed in the step icon.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

## Value

Object with shiny . tag class suitable for use in the UI of a Shiny app.

StepLabel

*StepLabel***Description**

<https://mui.com/material-ui/api/step-label/>

**Usage**

```
StepLabel(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - In most cases will simply be a string containing a title for the label.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- componentsProps { label?: object }  
Default is The props used for each slot inside. Deprecated use the slotProps prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- error bool  
Default is FALSE If true, the step is marked as failed.
- icon node  
Default is - Override the default label of the step icon.
- optional node  
Default is - The optional node to display.
- slotProps { label?: func| object, root?: func| object, stepIcon?: func| object }  
Default is The props used for each slot inside.
- slots { label?: elementType, root?: elementType, stepIcon?: elementType }  
Default is The components used for each slot inside.
- StepIconComponent elementType  
Default is - The component to render in place of the StepIcon. Deprecated Use slots.stepIcon instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- StepIconProps object  
Default is - Props applied to the StepIcon element. Deprecated Use slotProps.stepIcon instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.

- `sx` Array `func| object| bool | func| object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny` tag class suitable for use in the UI of a Shiny app.

---

Stepper

*Stepper*

---

**Description**

<https://mui.com/material-ui/api/stepper/>

**Usage**

```
Stepper(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `activeStep` integer  
Default is 0 Set the active step (zero based index). Set to -1 to disable all the steps.
- `alternativeLabel` bool  
Default is FALSE If set to 'true' and orientation is horizontal, then the step label will be positioned under the icon.
- `children` node  
Default is - Two or more Step / components.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component` elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `connector` element  
Default is StepConnector / An element to be placed between each step.
- `nonLinear` bool  
Default is FALSE If set the Stepper will not assist in controlling steps for linear flow.
- `orientation` 'horizontal' | 'vertical'  
Default is 'horizontal' The component orientation (layout flow direction).
- `sx` Array `func| object| bool | func| object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

SvgIcon

*SvgIcon*

---

**Description**

<https://mui.com/material-ui/api/svg-icon/>

**Usage**

```
SvgIcon(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children` node  
Default is - Node passed into the SVG element.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` `'inherit' | 'action' | 'disabled' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string`  
Default is `'inherit'` The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide. You can use the `htmlColor` prop to apply a color attribute to the SVG element.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `fontSize` `'inherit' | 'large' | 'medium' | 'small' | string`  
Default is `'medium'` The `fontSize` applied to the icon. Defaults to 24px, but can be configured to inherit font size.
- `htmlColor` `string`  
Default is - Applies a color attribute to the SVG element.
- `inheritViewBox` `bool`  
Default is `FALSE` If true, the root node will inherit the custom component's `viewBox` and the `viewBox` prop will be ignored. Useful when you want to reference a custom component and have `SvgIcon` pass that component's `viewBox` to the root node.
- `shapeRendering` `string`  
Default is - The shape-rendering attribute. The behavior of the different options is described on the MDN Web Docs. If you are having issues with blurry icons you should investigate this prop.

- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `titleAccess` string  
Default is - Provides a human-readable title for the element that contains it. <https://www.w3.org/TR/SVG-access/#Equivalent>
- `viewBox` string  
Default is `'0 0 24 24'` Allows you to redefine what the coordinates without units mean inside an SVG element. For example, if the SVG element is 500 (width) by 200 (height), and you pass `viewBox="0 0 50 20"`, this means that the coordinates inside the SVG will go from the top left corner (0,0) to bottom right (50,20) and each unit will be worth 10px.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

SwipeableDrawer	<i>SwipeableDrawer</i>
-----------------	------------------------

---

**Description**

<https://mui.com/material-ui/api/swipeable-drawer/>

**Usage**

```
SwipeableDrawer(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- `onClose` func  
Default is - Callback fired when the component requests to be closed. Signature: `function(event: React.SyntheticEvent) = void` event The event source of the callback.
- `onOpen` func  
Default is - Callback fired when the component requests to be opened. Signature: `function(event: React.SyntheticEvent) = void` event The event source of the callback.
- `allowSwipeInChildren` func | `bool`  
Default is `FALSE` If set to true, the swipe event will open the drawer even if the user begins the swipe on one of the drawer's children. This can be useful in scenarios where the drawer is partially visible. You can customize it further with a callback that determines which children the user can drag over to open the drawer (for example, to ignore other elements that handle touch move events, like sliders).

- **children** node  
Default is - The content of the component.
- **disableBackdropTransition** bool  
Default is FALSE Disable the backdrop transition. This can improve the FPS on low-end devices.
- **disableDiscovery** bool  
Default is FALSE If true, touching the screen near the edge of the drawer will not slide in the drawer a bit to promote accidental discovery of the swipe gesture.
- **disableSwipeToOpen** bool  
Default is `typeof navigator !== 'undefined' && /iPad|iPhone|iPod/.test(navigator.userAgent)`  
If true, swipe to open is disabled. This is useful in browsers where swiping triggers navigation actions. Swipe to open is disabled on iOS browsers by default.
- **hysteresis** number  
Default is 0.52 Affects how far the drawer must be opened/closed to change its state. Specified as percent (0-1) of the width of the drawer
- **minFlingVelocity** number  
Default is 450 Defines, from which (average) velocity on, the swipe is defined as complete although hysteresis isn't reached. Good threshold is between 250 - 1000 px/s
- **open** bool  
Default is FALSE If true, the component is shown.
- **slotProps** { backdrop?: func| object, docked?: func| object, paper?: func| object, root?: func| object }  
Default is The props used for each slot inside.
- **slots** { backdrop?: elementType, docked?: elementType, paper?: elementType, root?: elementType, swipe }  
Default is The components used for each slot inside.
- **SwipeAreaProps** object  
Default is - The element is used to intercept the touch events on the edge. Deprecated use the `slotProps.swipeArea` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- **swipeAreaWidth** number  
Default is 20 The width of the left most (or right most) area in px that the drawer can be swiped open from.
- **transitionDuration** number | { appear?: number, enter?: number, exit?: number }  
Default is `enter: theme.transitions.duration.enteringScreen, exit: theme.transitions.duration.leavingScreen`,  
The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

```
SwipeableDrawer.triggerId
  SwipeableDrawer.triggerId
```

---

**Description**

Custom SwipeableDrawer bound to a DOM element by id. See `'js/src/MuiSwipeableDrawerTriggerId.jsx'`.

**Usage**

```
SwipeableDrawer.triggerId(triggerId, ...)
```

**Arguments**

<code>triggerId</code>	HTML id of an existing DOM element that acts as the trigger (button, link, etc.) to open the Drawer.
<code>...</code>	Named arguments forwarded as React props, plus children to render inside the component. Pass <code>closeOnLinkClick = FALSE</code> to keep the Drawer open when any link inside it is clicked. The default (TRUE) closes the Drawer on any <code>&lt;a&gt;</code> click, including external links with <code>target = "_blank"</code> .

**Value**

Object with `'shiny.tag'` class suitable for use in the UI of a Shiny app.

---

```
Switch          Switch
```

---

**Description**

<https://mui.com/material-ui/api/switch/>

**Usage**

```
Switch(...)

Switch.shinyInput(inputId, ..., value = defaultValue)

updateSwitch.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- checked bool  
Default is - If true, the component is checked.
- checkedIcon node  
Default is - The icon to display when the component is checked.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- color 'default' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'primary' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- defaultChecked bool  
Default is - The default checked state. Use when the component is not controlled.
- disabled bool  
Default is - If true, the component is disabled.
- disableRipple bool  
Default is FALSE If true, the ripple effect is disabled.
- edge 'end' | 'start' | false  
Default is FALSE If given, uses a negative margin to counteract the padding on one side (this is often helpful for aligning the left or right side of the icon with content above or below, without ruining the border size and shape).
- icon node  
Default is - The icon to display when the component is unchecked.
- id string  
Default is - The id of the input element.
- inputProps object  
Default is - Attributes applied to the input element. *Deprecated* Use slotProps.input instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- inputRef ref  
Default is - Pass a ref to the input element. *Deprecated* Use slotProps.input.ref instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- onChange func  
Default is - Callback fired when the state is changed. Signature: function(event: React.ChangeEvent)

= voidevent The event source of the callback. You can pull out the new value by accessing event.target.value (string). You can pull out the new checked state by accessing event.target.checked (boolean).

- **required** bool  
Default is FALSE If true, the input element is required.
- **size** 'medium' | 'small' | string  
Default is 'medium' The size of the component. small is equivalent to the dense switch styling.
- **slotProps** { input?: func| object, root?: func| object, switchBase?: func| object, thumb?: func| object }  
Default is The props used for each slot inside.
- **slots** { input?: elementType, root?: elementType, switchBase?: elementType, thumb?: elementType, track?: elementType }  
Default is The components used for each slot inside.
- **sx** Array func| object| bool | func| object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- **value** any  
Default is - The value of the component. The DOM API casts this to a string. The browser uses "on" as the default value.

### Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

Tab	<i>Tab</i>
-----	------------

---

### Description

<https://mui.com/material-ui/api/tab/>

### Usage

```
Tab(...)
```

### Arguments

... Props to pass to the component.

### Details

- **children** unsupportedProp  
Default is - This prop isn't supported. Use the component prop if you need to change the children structure.
- **classes** object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.

- `disabled` `bool`  
Default is `FALSE` If true, the component is disabled.
- `disableFocusRipple` `bool`  
Default is `FALSE` If true, the keyboard focus ripple is disabled.
- `disableRipple` `bool`  
Default is `FALSE` If true, the ripple effect is disabled. Without a ripple there is no styling for `:focus-visible` by default. Be sure to highlight the element by applying separate styles with the `.Mui-focusVisible` class.
- `icon` `element | string`  
Default is - The icon to display.
- `iconPosition` `'bottom' | 'end' | 'start' | 'top'`  
Default is `'top'` The position of the icon relative to the label.
- `label` `node`  
Default is - The label element.
- `sx` `Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `value` `any`  
Default is - You can provide your own value. Otherwise, we fallback to the child position index.
- `wrapped` `bool`  
Default is `FALSE` Tab labels appear in a single row. They can use a second line if needed.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

TabContext

*TabContext*

---

**Description**

<https://mui.com/material-ui/api/tab-context/>

**Usage**

```
TabContext(...)
```

```
TabContext.shinyInput(inputId, ..., value = defaultValue)
```

```
updateTabContext.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

```
TabContext.static(..., value = NULL, defaultValue = NULL)
```

**Arguments**

...	Child elements (typically a Box wrapping <code>TabList.static</code> and <code>TabPanel</code> components). <code>value</code> and <code>defaultValue</code> are accepted by name only – putting ... first guarantees that an unnamed child element is never matched positionally into <code>value</code> / <code>defaultValue</code> .
<code>inputId</code>	ID of the component.
<code>value</code>	Controlled selected tab value. When supplied, the caller is the source of truth: the value is honored on every render and the wrapper never mutates it. Combine with an <code>onChange</code> (on <code>TabList.static</code> or on <code>TabContext.static</code> ) that writes the new value back to wherever it lives. The tidiest binding needs no JavaScript: make the tab a URL path segment, link each <code>Tab</code> with <code>href</code> (e.g. <code>"/overview"</code> ), and read it back with <code>reactRouter::useParams(as = "value", selector = "tab")</code> . Use <code>useParams</code> (a scalar), not <code>useSearchParams</code> , which returns <code>getAll()</code> , an array that never matches a <code>TabPanel</code> 's string value. Other sources (a parent's state, a Shiny input) work too. Use either <code>value</code> or <code>defaultValue</code> , not both; the mode is fixed at the first render.
<code>session</code>	Object passed as the 'session' argument to Shiny server.
<code>defaultValue</code>	Uncontrolled initial selected tab value. The wrapper owns the active-tab state and updates it on user clicks. This is the right choice for Quarto documents and static HTML, where no external state is involved.

**Details**

- `value` `number` | `string`  
Default is `NA` The value of the currently selected `Tab`.
- `children` `node`  
Default is `NA` The content of the component.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

**Note**

`TabContext` is part of `@mui/lab`, which is published on the MUI beta channel. Lab APIs may change in future minor releases.

Pass an initial value matching one of the `TabPanel` values to pre-select that panel; when omitted, the component mounts with no panel selected (`value = ""`) rather than uncontrolled, so a later server-driven update does not trigger MUI's controlled/uncontrolled warning. `TabContext` itself has no change event, so `input[[inputId]]` reports only this initial value. To react to tab clicks on the server, read `input[[inputId]]` from the `TabList.shinyInput` instead.

---

 Table

*Table*


---

### Description

<https://mui.com/material-ui/api/table/>

### Usage

```
Table(...)
```

### Arguments

... Props to pass to the component.

### Details

- children node  
Default is - The content of the table, normally TableHead and TableBody.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- padding 'checkbox' | 'none' | 'normal'  
Default is 'normal' Allows TableCell to inherit padding of the Table.
- size 'medium' | 'small' | string  
Default is 'medium' Allows TableCell to inherit size of the Table.
- stickyHeader bool  
Default is FALSE Set the header sticky.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

### Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

TableBody	<i>TableBody</i>
-----------	------------------

---

**Description**

<https://mui.com/material-ui/api/table-body/>

**Usage**

```
TableBody(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component, normally TableRow.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is NA The component used for the root node. Either a string to use a HTML element or a component.
- sx Array func | object | bool | func | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

TableCell	<i>TableCell</i>
-----------	------------------

---

**Description**

<https://mui.com/material-ui/api/table-cell/>

**Usage**

```
TableCell(...)
```

## Arguments

... Props to pass to the component.

## Details

- align 'center' | 'inherit' | 'justify' | 'left' | 'right'  
Default is 'inherit' Set the text-align on the table cell content. Monetary or generally number fields should be right aligned as that allows you to add them up quickly in your head without having to worry about decimals.
- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- padding 'checkbox' | 'none' | 'normal'  
Default is - Sets the padding applied to the cell. The prop defaults to the value ('default') inherited from the parent Table component.
- scope string  
Default is - Set scope attribute.
- size 'medium' | 'small' | string  
Default is - Specify the size of the cell. The prop defaults to the value ('medium') inherited from the parent Table component.
- sortDirection 'asc' | 'desc' | false  
Default is - Set aria-sort direction.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- variant 'body' | 'footer' | 'head' | string  
Default is - Specify the cell type. The prop defaults to the value inherited from the parent TableHead, TableBody, or TableFooter components.

## Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

TableContainer	<i>TableContainer</i>
----------------	-----------------------

---

**Description**

<https://mui.com/material-ui/api/table-container/>

**Usage**

```
TableContainer(...)
```

**Arguments**

...                    Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component, normally Table.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is NA The component used for the root node. Either a string to use a HTML element or a component.
- sx Array func | object | bool | func | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

TableFooter	<i>TableFooter</i>
-------------	--------------------

---

**Description**

<https://mui.com/material-ui/api/table-footer/>

**Usage**

```
TableFooter(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component, normally TableRow.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is NA The component used for the root node. Either a string to use a HTML element or a component.
- sx Array func | object | bool | func | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

TableHead

*TableHead*

---

**Description**

<https://mui.com/material-ui/api/table-head/>

**Usage**

TableHead(...)

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component, normally TableRow.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is NA The component used for the root node. Either a string to use a HTML element or a component.

- `sx` Array `func` | `object` | `bool` | `func` | `object`  
Default is `NA` The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

TablePagination	<i>TablePagination</i>
-----------------	------------------------

---

**Description**

<https://mui.com/material-ui/api/table-pagination/>

**Usage**

TablePagination(...)

**Arguments**

... Props to pass to the component.

**Details**

- `count` integer  
Default is - The total number of rows. To enable server side pagination for an unknown number of items, provide `-1`.
- `onPageChange` `func`  
Default is - Callback fired when the page is changed. Signature: `function(event: React.MouseEvent | null, page: number) = void` event The event source of the callback. page The page selected.
- `page` integer  
Default is - The zero-based index of the current page.
- `rowsPerPage` integer  
Default is - The number of rows per page. Set `-1` to display all the rows.
- `ActionsComponent` `elementType`  
Default is `TablePaginationActions` The component used for displaying the actions. Either a string to use a HTML element or a component.
- `backIconButtonProps` `object`  
Default is - Props applied to the back arrow `IconButton` component. This prop is an alias for `slotProps.actions.previousButton` and will be overridden by it if both are used. **Deprecated** Use `slotProps.actions.previousButton` instead.
- `classes` `object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.

- `componentElementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `disabled` `bool`  
Default is `FALSE` If true, the component is disabled.
- `getItemAriaLabel` `func`  
Default is function `defaultGetAriaLabel(type) return Go to $type page`; Accepts a function which returns a string value that provides a user-friendly name for the current page. This is important for screen reader users. For localization purposes, you can use the provided translations. Signature: `function(type: string) = stringtype` The link or button type to format ('first' | 'last' | 'next' | 'previous').
- `labelDisplayedRows` `func`  
Default is function `defaultLabelDisplayedRows( from, to, count ) return $from to of $count` !`== -1 ? count : more than $to`; Customize the displayed rows label. Invoked with a from, to, count, page object. For localization purposes, you can use the provided translations.
- `labelRowsPerPage` `node`  
Default is 'Rows per page:' Customize the rows per page label. For localization purposes, you can use the provided translations.
- `nextIconButtonProps` 'object'  
Default is - Props applied to the next arrow `IconButton` element. This prop is an alias for `slotProps.actions.nextButton` and will be overridden by it if both are used. `Deprecated` Use `slotProps.actions.nextButton` instead.
- `onRowsPerPageChange` 'func'  
Default is - Callback fired when the number of rows per page is changed. Signature: `function(event: React.ChangeEvent) = void` event The event source of the callback.
- `rowsPerPageOptions`  
Default is 10, 25, 50, 100 Customizes the options of the rows per page select field. If less than two options are available, no select field will be displayed. Use -1 for the value with a custom label to show all the rows.
- `SelectProps` 'object'  
Default is Props applied to the rows per page `Select` element. This prop is an alias for `slotProps.select` and will be overridden by it if both are used. `Deprecated` Use `slotProps.select` instead.
- `showFirstButton` 'bool'  
Default is `FALSE` If true, show the first-page button.
- `showLastButton` 'bool'  
Default is `FALSE` If true, show the last-page button.
- `slotProps`  
Default is The props used for each slot inside.
- `slots`  
Default is The components used for each slot inside.
- `sx`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the 'sx' page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

TableRow	<i>TableRow</i>
----------	-----------------

---

**Description**

<https://mui.com/material-ui/api/table-row/>

**Usage**

TableRow(...)

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - Should be valid tr children such as TableCell.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- component elementType  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- hover bool  
Default is FALSE If true, the table row will shade on hover.
- selected bool  
Default is FALSE If true, the table row will have the selected shading.
- sx Array func| object| bool | func| object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

 TableSortLabel

*TableSortLabel*


---

### Description

<https://mui.com/material-ui/api/table-sort-label/>

### Usage

```
TableSortLabel(...)
```

### Arguments

... Props to pass to the component.

### Details

- **active** bool  
Default is FALSE If true, the label will have the active styling (should be true for the sorted column).
- **children** node  
Default is - Label contents, the arrow will be appended automatically.
- **classes** object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **direction** 'asc' | 'desc'  
Default is 'asc' The current sort direction.
- **hideSortIcon** bool  
Default is FALSE Hide sort icon when active is false.
- **IconComponent** elementType  
Default is ArrowDownwardIcon Sort icon to use.
- **slotProps** { icon?: func | object, root?: func | object }  
Default is The props used for each slot inside.
- **slots** { icon?: elementType, root?: elementType }  
Default is The components used for each slot inside.
- **sx** Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

`TabList`*TabList*

---

**Description**

<https://mui.com/material-ui/api/tab-list/>

**Usage**

```
TabList(...)
```

```
TabList.shinyInput(inputId, ..., value = defaultValue)
```

```
updateTabList.shinyInput(  
  session = shiny::getDefaultReactiveDomain(),  
  inputId,  
  ...  
)
```

```
TabList.static(...)
```

**Arguments**

<code>...</code>	Child Tab elements and other props.
<code>inputId</code>	ID of the component.
<code>value</code>	Starting value.
<code>session</code>	Object passed as the ‘session’ argument to Shiny server.

**Details**

- `children` node  
Default is NA A list of Tab / elements.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

**Note**

`TabList` is part of `@mui/lab`, which is published on the MUI beta channel. Lab APIs may change in future minor releases.

---

 TabPanel
*TabPanel***Description**

'TabPanel.static()' is an alias for 'TabPanel()', kept for backward compatibility.

<https://mui.com/material-ui/api/tab-panel/>

**Usage**

```

TabPanel(...)

TabPanel.static(...)

TabPanel.shinyInput(inputId, ..., value = defaultValue)

updateTabPanel.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)

```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- value number | string  
Default is - The value of the corresponding Tab. Must use the index of the Tab when no value was passed to Tab.
- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- keepMounted bool  
Default is FALSE Always keep the children in the DOM.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

**Note**

`TabPanel` is part of `@mui/lab`, which is published on the MUI beta channel. Lab APIs may change in future minor releases.

`TabPanel` is a display-only component: it shows or hides its content based on the active tab value held by the surrounding `TabContext`, but never fires `onChange` from user interaction. As a result `input$<inputId>` will always remain `NULL`. To react to tab changes on the server, read `input$<inputId>` from the `TabList.shinyInput` instead.

---

 Tabs

*Tab*s
 

---

**Description**

<https://mui.com/material-ui/api/tabs/>

**Usage**

```
Tabs(...)
```

```
Tabs.shinyInput(inputId, ..., value = defaultValue)
```

```
updateTabs.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

<code>...</code>	Props to pass to the component.
<code>inputId</code>	ID of the component.
<code>value</code>	Starting value.
<code>session</code>	Object passed as the ‘session’ argument to Shiny server.

**Details**

- `action ref`  
Default is - Callback fired when the component mounts. This is useful when you want to trigger an action programmatically. It supports two actions: `updateIndicator()` and `updateScrollButtons()`

- `allowScrollButtonsMobile` `bool`  
Default is `FALSE` If true, the scroll buttons aren't forced hidden on mobile. By default the scroll buttons are hidden on mobile and takes precedence over `scrollButtons`.
- `aria-label` `string`  
Default is - The label for the Tabs as a string.
- `aria-labelledby` `string`  
Default is - An id or list of ids separated by a space that label the Tabs.
- `centered` `bool`  
Default is `FALSE` If true, the tabs are centered. This prop is intended for large views.
- `children` `node`  
Default is - The content of the component.
- `classes` `object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `indicatorColor` `'primary' | 'secondary' | string`  
Default is `'primary'` Determines the color of the indicator.
- `onChange` `func`  
Default is - Callback fired when the value changes. Signature: `function(event: React.SyntheticEvent, value: any) = void` event The event source of the callback. Warning: This is a generic event not a change event. value We default to the index of the child (number)
- `orientation` `'horizontal' | 'vertical'`  
Default is `'horizontal'` The component orientation (layout flow direction).
- `ScrollButtonComponent` `elementType`  
Default is `TabScrollButton` The component used to render the scroll buttons. Deprecated use the `slots.scrollButtons` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `scrollButtons` `'auto' | false | true`  
Default is `'auto'` Determine behavior of scroll buttons when tabs are set to scroll: `auto` will only present them when not all the items are visible. `true` will always present them. `false` will never present them. By default the scroll buttons are hidden on mobile. This behavior can be disabled with `allowScrollButtonsMobile`.
- `selectionFollowsFocus` `bool`  
Default is - If true the selected tab changes on focus. Otherwise it only changes on activation.
- `slotProps` `{ endScrollButtonIcon?: func | object, indicator?: func | object, list?: func | object, root?: func | object }`  
Default is The props used for each slot inside.
- `slots` `{ endScrollButtonIcon?: elementType, EndScrollButtonIcon?: elementType, indicator?: elementType }`  
Default is The components used for each slot inside.
- `sx` `Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the [sx](#) page for more details.

- **TabIndicatorProps** object  
Default is Props applied to the tab indicator element. Deprecated use the slotProps.indicator prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- **TabScrollButtonProps** object  
Default is Props applied to the TabScrollButton element. Deprecated use the slotProps.scrollButtons prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- **textColor** 'inherit' | 'primary' | 'secondary'  
Default is 'primary' Determines the color of the Tab.
- **value** any  
Default is - The value of the currently selected Tab. If you don't want any selected Tab, you can set this prop to false.
- **variant** 'fullWidth' | 'scrollable' | 'standard'  
Default is 'standard' Determines additional display behavior of the tabs: scrollable will invoke scrolling properties and allow for horizontally scrolling (or swiping) of the tab bar. fullWidth will make the tabs grow to use all the available space, which should be used for small views, like on mobile. standard will render the default state.
- **visibleScrollbar** bool  
Default is FALSE If true, the scrollbar is visible. It can be useful when displaying a long vertical list of tabs.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

**Note**

Pass an initial value matching one of the child Tab values to pre-select that tab. When omitted, the component mounts with no tab selected (value = FALSE) rather than uncontrolled, so a later server-driven update does not trigger MUI's controlled/uncontrolled warning.

---

TabScrollButton	<i>TabScrollButton</i>
-----------------	------------------------

---

**Description**

<https://mui.com/material-ui/api/tab-scroll-button/>

**Usage**

```
TabScrollButton(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `direction` `'left' | 'right'`  
Default is - The direction the button should indicate.
- `orientation` `'horizontal' | 'vertical'`  
Default is - The component orientation (layout flow direction).
- `children` node  
Default is - The content of the component.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `disabled` bool  
Default is `FALSE` If true, the component is disabled.
- `slotProps` { `endScrollButtonIcon?: func | object`, `startScrollButtonIcon?: func | object` }  
Default is The extra props for the slot components. You can override the existing props or add new ones.
- `slots` { `EndScrollButtonIcon?: elementType`, `StartScrollButtonIcon?: elementType` }  
Default is The components used for each slot inside.
- `sx` Array `func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

<code>TextareaAutosize</code>	<i>TextareaAutosize</i>
-------------------------------	-------------------------

---

**Description**

<https://mui.com/material-ui/api/textarea-autosize/>

**Usage**

```
TextareaAutosize(...)
```

**Arguments**

...                      Props to pass to the component.

**Details**

- `maxRows` number | string  
Default is - Maximum number of rows to display.
- `minRows` number | string  
Default is 1 Minimum number of rows to display.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

 TextField

*TextField*


---

**Description**

<https://mui.com/material-ui/api/text-field/>

**Usage**

```
TextField(...)
```

```
TextField.shinyInput(inputId, ..., value = defaultValue)
```

```
updateTextField.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

<code>...</code>	Props to pass to the component.
<code>inputId</code>	ID of the component.
<code>value</code>	Starting value.
<code>session</code>	Object passed as the 'session' argument to Shiny server.

**Details**

- `autoComplete` string  
Default is - This prop helps users to fill forms faster, especially on mobile devices. The name can be confusing, as it's more like an autofill. You can learn more about it following the specification.
- `autoFocus` bool  
Default is FALSE If true, the input element is focused during the first mount.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `color` 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'primary' The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- `defaultValue` any  
Default is - The default value. Use when the component is not controlled.

- `disabled` `bool`  
Default is `FALSE` If true, the component is disabled.
- `error` `bool`  
Default is `FALSE` If true, the label is displayed in an error state.
- `FormHelperTextProps` object  
Default is - Props applied to the `FormHelperText` element. `Deprecated` Use `slotProps.formHelperText` instead. This prop will be removed in a future major release. See `Migrating from deprecated APIs` for more details.
- `fullWidth` `bool`  
Default is `FALSE` If true, the input will take up the full width of its container.
- `helperText` node  
Default is - The helper text content.
- `id` `string`  
Default is - The id of the input element. Use this prop to make label and `helperText` accessible for screen readers.
- `InputLabelProps` object  
Default is - Props applied to the `InputLabel` element. Pointer events like `onClick` are enabled if and only if `shrink` is true. `Deprecated` Use `slotProps.inputLabel` instead. This prop will be removed in a future major release. See `Migrating from deprecated APIs` for more details.
- `inputProps` object  
Default is - Attributes applied to the input element. `Deprecated` Use `slotProps.htmlInput` instead. This prop will be removed in a future major release. See `Migrating from deprecated APIs` for more details.
- `InputProps` object  
Default is - Props applied to the Input element. It will be a `FilledInput`, `OutlinedInput` or `Input` component depending on the variant prop value. `Deprecated` Use `slotProps.input` instead. This prop will be removed in a future major release. See `Migrating from deprecated APIs` for more details.
- `inputRef` `ref`  
Default is - Pass a ref to the input element.
- `label` node  
Default is - The label content.
- `margin` `'dense' | 'none' | 'normal'`  
Default is `'none'` If `dense` or `normal`, will adjust vertical spacing of this and contained components.
- `maxRows` `number | string`  
Default is - Maximum number of rows to display when multiline option is set to true.
- `minRows` `number | string`  
Default is - Minimum number of rows to display when multiline option is set to true.
- `multiline` `bool`  
Default is `FALSE` If true, a `textarea` element is rendered instead of an input.
- `name` `string`  
Default is - Name attribute of the input element.

- `onChange` func  
Default is - Callback fired when the value is changed. Signature: `function(event: object) = void` event The event source of the callback. You can pull out the new value by accessing `event.target.value` (string).
- `placeholder` string  
Default is - The short hint displayed in the input before the user enters a value.
- `required` bool  
Default is FALSE If true, the label is displayed as required and the input element is required.
- `rows` number | string  
Default is - Number of rows to display when multiline option is set to true.
- `select` bool  
Default is FALSE Render a Select element while passing the Input element to Select as input parameter. If this option is set you must pass the options of the select as children.
- `SelectProps` object  
Default is - Props applied to the Select element. **Deprecated** Use `slotProps.select` instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `size` 'medium' | 'small' | string  
Default is 'medium' The size of the component.
- `slotProps` { `formHelperText?: func | object`, `htmlInput?: func | object`, `input?: func | object`, `inputLabel?: func | object` }  
Default is The props used for each slot inside.
- `slots` { `formHelperText?: elementType`, `htmlInput?: elementType`, `input?: elementType`, `inputLabel?: elementType` }  
Default is The components used for each slot inside.
- `sx` Array `func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the [sx](#) page for more details.
- `type` string  
Default is - Type of the input element. It should be a valid HTML5 input type.
- `value` any  
Default is - The value of the input element, required for a controlled component.
- `variant` 'filled' | 'outlined' | 'standard'  
Default is 'outlined' The variant to use.

### Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

### Examples

```
library(shiny)
library(muiMaterial)

ui <- muiMaterialPage(
  TextField.shinyInput("txt", label = "Your name", value = ""),
  verbatimTextOutput("out")
)
```

```

server <- function(input, output, session) {
  output$out <- renderPrint(input$txt)
}

shinyApp(ui, server)

```

---

 ThemeProvider

*ThemeProvider*


---

### Description

Custom ThemeProvider built on top of MUI's 'ThemeProvider'. See 'js/src/ThemeProvider.jsx'. Pass a 'theme' list (as you would pass a JS object to MUI's 'createTheme()') plus any children to render under that theme. See <<https://mui.com/material-ui/customization/theming/>> for the upstream documentation.

### Usage

```
ThemeProvider(...)
```

### Arguments

...                    Named arguments forwarded as React props (notably 'theme', a list mirroring MUI's 'createTheme()' options), plus the children to render under the theme.

### Value

Object with 'shiny.tag' class suitable for use in the UI of a Shiny app.

### Examples

```

library(shiny)
library(muiMaterial)

theme <- list(palette = list(mode = "dark", primary = list(main = "#90caf9")))

ui <- muiMaterialPage(
  ThemeProvider(
    theme = theme,
    CssBaseline(),
    Button("Themed button", variant = "contained")
  )
)

shinyApp(ui, function(input, output, session) {})

```

---

Timeline

*Timeline*

---

## Description

<https://mui.com/material-ui/api/timeline/>

## Usage

```
Timeline(...)
```

## Arguments

... Props to pass to the component.

## Details

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- className string  
Default is - className applied to the root element.
- position 'alternate-reverse' | 'alternate' | 'left' | 'right'  
Default is 'right' The position where the TimelineContent should appear relative to the time axis.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

## Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

## Note

Timeline and its sub-components (TimelineItem, TimelineDot, etc.) are part of [@mui/lab](#), which is published on the MUI beta channel. Lab APIs may change in future minor releases.

---

TimelineConnector      *TimelineConnector*

---

### Description

<https://mui.com/material-ui/api/timeline-connector/>

### Usage

```
TimelineConnector(...)
```

### Arguments

...      Props to pass to the component.

### Details

- children node  
Default is NA The content of the component.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func | object | bool | func | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

### Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

TimelineContent      *TimelineContent*

---

### Description

<https://mui.com/material-ui/api/timeline-content/>

### Usage

```
TimelineContent(...)
```

### Arguments

...      Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func | object | bool | func | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

TimelineDot

*TimelineDot*


---

**Description**

<https://mui.com/material-ui/api/timeline-dot/>

**Usage**

```
TimelineDot(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- color 'error' | 'grey' | 'info' | 'inherit' | 'primary' | 'secondary' | 'success' | 'warning' | string  
Default is 'grey' The dot can have a different colors.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- variant 'filled' | 'outlined' | string  
Default is 'filled' The dot can appear filled or outlined.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

TimelineItem

*TimelineItem*

---

**Description**

<https://mui.com/material-ui/api/timeline-item/>

**Usage**

```
TimelineItem(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- **children** node  
Default is NA The content of the component.
- **classes** object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- **position** 'alternate-reverse' | 'alternate' | 'left' | 'right'  
Default is NA The position where the timeline's item should appear.
- **sx** Array func | object | bool | func | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

TimelineOppositeContent  
*TimelineOppositeContent*

---

### Description

<https://mui.com/material-ui/api/timeline-opposite-content/>

### Usage

```
TimelineOppositeContent(...)
```

### Arguments

... Props to pass to the component.

### Details

- children node  
Default is NA The content of the component.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func | object | bool | func | object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

### Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

TimelineSeparator      *TimelineSeparator*

---

### Description

<https://mui.com/material-ui/api/timeline-separator/>

### Usage

```
TimelineSeparator(...)
```

### Arguments

... Props to pass to the component.

**Details**

- children node  
Default is NA The content of the component.
- classes object  
Default is NA Override or extend the styles applied to the component. See CSS classes API below for more details.
- sx Array func| object| bool | func| object  
Default is NA The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

ToggleButton

*ToggleButton*

---

**Description**

<https://mui.com/material-ui/api/toggle-button/>

**Usage**

```
ToggleButton(...)
```

```
ToggleButton.shinyInput(inputId, ...)
```

```
updateToggleButton.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- value any  
Default is - The value to associate with the button when selected in a ToggleButtonGroup.
- children node  
Default is - The content of the component.

- **classes** object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- **color** 'standard' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'standard' The color of the button when it is in an active state. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- **disabled** bool  
Default is FALSE If true, the component is disabled.
- **disableFocusRipple** bool  
Default is FALSE If true, the keyboard focus ripple is disabled.
- **disableRipple** bool  
Default is FALSE If true, the ripple effect is disabled. Without a ripple there is no styling for :focus-visible by default. Be sure to highlight the element by applying separate styles with the .Mui-focusVisible class.
- **fullWidth** bool  
Default is FALSE If true, the button will take up the full width of its container.
- **onChange** func  
Default is - Callback fired when the state changes. Signature: function(event: React.MouseEvent, value: any) = void event The event source of the callback. value of the selected button.
- **onClick** func  
Default is - Callback fired when the button is clicked. Signature: function(event: React.MouseEvent, value: any) = void event The event source of the callback. value of the selected button.
- **selected** bool  
Default is - If true, the button is rendered in an active state.
- **size** 'small' | 'medium' | 'large' | string  
Default is 'medium' The size of the component. The prop defaults to the value inherited from the parent ToggleButtonGroup component.
- **sx** Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

**Value**

Object with shiny.tag class suitable for use in the UI of a Shiny app.

---

ToggleButtonGroup

*ToggleButtonGroup*

---

**Description**

<https://mui.com/material-ui/api/toggle-button-group/>

**Usage**

```
ToggleButtonGroup(...)

ToggleButtonGroup.shinyInput(inputId, ..., value = defaultValue)

updateToggleButtonGroup.shinyInput(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  ...
)
```

**Arguments**

...	Props to pass to the component.
inputId	ID of the component.
value	Starting value.
session	Object passed as the 'session' argument to Shiny server.

**Details**

- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- color 'standard' | 'primary' | 'secondary' | 'error' | 'info' | 'success' | 'warning' | string  
Default is 'standard' The color of the button when it is selected. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- disabled bool  
Default is FALSE If true, the component is disabled. This implies that all ToggleButton children will be disabled.
- exclusive bool  
Default is FALSE If true, only allow one of the child ToggleButton values to be selected.
- fullWidth bool  
Default is FALSE If true, the button group will take up the full width of its container.
- onChange func  
Default is - Callback fired when the value changes. Signature: function(event: React.MouseEvent, value: any) = void event The event source of the callback. value of the selected buttons. When exclusive is true this is a single value; when false an array of selected values. If no value is selected and exclusive is true the value is null; when false an empty array.
- orientation 'horizontal' | 'vertical'  
Default is 'horizontal' The component orientation (layout flow direction).
- size 'small' | 'medium' | 'large' | string  
Default is 'medium' The size of the component.

- `sx Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `value any`  
Default is - The currently selected value within the group or an array of selected values when `exclusive` is `false`. The value must have reference equality with the option in order to be selected.

## Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Toolbar

*Toolbar*

---

## Description

<https://mui.com/material-ui/api/toolbar/>

## Usage

```
Toolbar(...)
```

## Arguments

... Props to pass to the component.

## Details

- `children node`  
Default is - The Toolbar children, usually a mixture of `IconButton`, `Button` and `Typography`. The Toolbar is a flex container, allowing flex item properties to be used to lay out the children.
- `classes object`  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `component elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `disableGutters bool`  
Default is `FALSE` If true, disables gutter padding.
- `sx Array func | object | bool | func | object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant 'dense' | 'regular' | string`  
Default is `'regular'` The variant to use.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Tooltip	<i>Tooltip</i>
---------	----------------

---

**Description**

<https://mui.com/material-ui/api/tooltip/>

**Usage**

```
Tooltip(...)
```

**Arguments**

... Props to pass to the component.

**Details**

- `children` element  
Default is - Tooltip reference element. This needs to be able to hold a ref.
- `arrow` bool  
Default is FALSE If true, adds an arrow to the tooltip.
- `classes` object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `components` { `Arrow?: elementType`, `Popper?: elementType`, `Tooltip?: elementType`, `Transition?: elementType` }  
Default is The components used for each slot inside. Deprecated use the `slots` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `componentsProps` { `arrow?: object`, `popper?: object`, `tooltip?: object`, `transition?: object` }  
Default is The extra props for the slot components. You can override the existing props or add new ones. Deprecated use the `slotProps` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `describeChild` bool  
Default is FALSE Set to true if the title acts as an accessible description. By default the title acts as an accessible label for the child.
- `disableFocusListener` bool  
Default is FALSE Do not respond to focus-visible events.
- `disableHoverListener` bool  
Default is FALSE Do not respond to hover events.
- `disableInteractive` bool  
Default is FALSE Makes a tooltip not interactive, i.e. it will close when the user hovers over the tooltip before the `leaveDelay` is expired.

- `disableTouchListener` `bool`  
Default is `FALSE` Do not respond to long press touch events.
- `enterDelay` `number`  
Default is `100` The number of milliseconds to wait before showing the tooltip. This prop won't impact the enter touch delay (`enterTouchDelay`).
- `enterNextDelay` `number`  
Default is `0` The number of milliseconds to wait before showing the tooltip when one was already recently opened.
- `enterTouchDelay` `number`  
Default is `700` The number of milliseconds a user must touch the element before showing the tooltip.
- `followCursor` `bool`  
Default is `FALSE` If true, the tooltip follow the cursor over the wrapped element.
- `id` `string`  
Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.
- `leaveDelay` `number`  
Default is `0` The number of milliseconds to wait before hiding the tooltip. This prop won't impact the leave touch delay (`leaveTouchDelay`).
- `leaveTouchDelay` `number`  
Default is `1500` The number of milliseconds after the user stops touching an element before hiding the tooltip.
- `onClose` `func`  
Default is - Callback fired when the component requests to be closed. Signature: `function(event: React.SyntheticEvent) = void` event The event source of the callback.
- `onOpen` `func`  
Default is - Callback fired when the component requests to be open. Signature: `function(event: React.SyntheticEvent) = void` event The event source of the callback.
- `open` `bool`  
Default is - If true, the component is shown.
- `placement` `'auto-end' | 'auto-start' | 'auto' | 'bottom-end' | 'bottom-start' | 'bottom' | 'left-end' | 'left-start' | 'left' | 'right-end' | 'right-start' | 'right' | 'top-end' | 'top-start' | 'top'`  
Default is `'bottom'` Tooltip placement.
- `PopperComponent` `elementType`  
Default is - The component used for the popper. Deprecated use the `slots.popper` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `PopperProps` `object`  
Default is Props applied to the Popper element. Deprecated use the `slotProps.popper` prop instead. This prop will be removed in a future major release. See [Migrating from deprecated APIs](#) for more details.
- `slotProps` `{ arrow?: func | object, popper?: func | object, tooltip?: func | object, transition?: func | object }`  
Default is The props used for each slot inside.

- slots { arrow?: elementType, popper?: elementType, tooltip?: elementType, transition?: elementType }  
Default is The components used for each slot inside.
- sx Array func | object | bool | func | object  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.
- title node  
Default is - Tooltip title. Zero-length titles string, undefined, null and false are never displayed.
- TransitionComponent elementType  
Default is - The component used for the transition. Follow this guide to learn more about the requirements for this component. Deprecated use the slots.transition prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- TransitionProps object  
Default is Props applied to the transition element. By default, the element is based on this Transition component. Deprecated use the slotProps.transition prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.

**Value**

Object with shiny . tag class suitable for use in the UI of a Shiny app.

---

Typography

*Typography*

---

**Description**

<https://mui.com/material-ui/api/typography/>

**Usage**

Typography(...)

**Arguments**

... Props to pass to the component.

**Details**

- align 'center' | 'inherit' | 'justify' | 'left' | 'right'  
Default is 'inherit' Set the text-align on the component.
- children node  
Default is - The content of the component.
- classes object  
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.

- `color` `'primary'` | `'secondary'` | `'success'` | `'error'` | `'info'` | `'warning'` | `'textPrimary'` | `'textSecondary'` | `'textDisabled'` | `string`  
Default is - The color of the component. It supports both default and custom theme colors, which can be added as shown in the palette customization guide.
- `component` `elementType`  
Default is - The component used for the root node. Either a string to use a HTML element or a component.
- `gutterBottom` `bool`  
Default is `FALSE` If true, the text will have a bottom margin.
- `noWrap` `bool`  
Default is `FALSE` If true, the text will not wrap, but instead will truncate with a text overflow ellipsis. Note that text overflow can only happen with block or inline-block level elements (the element needs to have a width in order to overflow).
- `paragraph` `bool`  
Default is `FALSE` If true, the element will be a paragraph element. **Deprecated** Use the component prop instead. This prop will be removed in a future major release. See Migrating from deprecated APIs for more details.
- `sx` `Array` `func` | `object` | `bool` | `func` | `object`  
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.
- `variant` `'body1'` | `'body2'` | `'button'` | `'caption'` | `'h1'` | `'h2'` | `'h3'` | `'h4'` | `'h5'` | `'h6'` | `'inherit'` | `'overline'` | `'subtitle1'` | `'subtitle2'` | `string`  
Default is `'body1'` Applies the theme typography styles.
- `variantMapping` `object`  
Default is `h1: 'h1', h2: 'h2', h3: 'h3', h4: 'h4', h5: 'h5', h6: 'h6', subtitle1: 'h6', subtitle2: 'h6', body1: 'p', body2: 'p', inherit: 'p'`, The component maps the variant prop to a range of different HTML element types. For instance, `subtitle1` to `h6` . If you wish to change that mapping, you can provide your own. Alternatively, you can use the component prop.

**Value**

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

---

Zoom

*Zoom*

---

**Description**

<https://mui.com/material-ui/api/zoom/>

**Usage**

`Zoom(...)`

## Arguments

... Props to pass to the component.

## Details

- children element  
Default is - A single child content element. This needs to be able to hold a ref.
- addEventListener func  
Default is - Add a custom transition end trigger. Called with the transitioning DOM node and a done callback. Allows for more fine grained transition end logic. Note: Timeouts are still used as a fallback if provided.
- appear bool  
Default is TRUE Perform the enter transition when it first mounts if in is also true. Set this to false to disable this behavior.
- easing { enter?: string, exit?: string } | string  
Default is - The transition timing function. You may specify a single easing or a object containing enter and exit values.
- in bool  
Default is - If true, the component will transition in.
- timeout number | { appear?: number, enter?: number, exit?: number }  
Default is enter: theme.transitions.duration.enteringScreen, exit: theme.transitions.duration.leavingScreen, The duration for the transition, in milliseconds. You may specify a single timeout for all transitions, or individually with an object.

## Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

# Index

Accordion, [5](#)  
AccordionActions, [6](#)  
AccordionDetails, [7](#)  
AccordionSummary, [8](#)  
Alert, [9](#)  
AlertTitle, [10](#)  
AppBar, [11](#)  
Autocomplete, [12](#)  
Avatar, [17](#)  
AvatarGroup, [19](#)

Backdrop, [20](#)  
Badge, [21](#)  
BottomNavigation, [23](#)  
BottomNavigationAction, [24](#)  
Box, [25](#)  
Breadcrumbs, [26](#)  
Button, [27](#)  
ButtonBase, [29](#)  
ButtonGroup, [30](#)

Card, [32](#)  
CardActionArea, [32](#)  
CardActions, [33](#)  
CardContent, [34](#)  
CardHeader, [34](#)  
CardMedia, [36](#)  
Checkbox, [37](#)  
Chip, [39](#)  
CircularProgress, [40](#)  
ClickAwayListener, [41](#)  
Collapse, [42](#)  
Container, [43](#)  
CssBaseline, [44](#)

Dialog, [45](#)  
Dialog.triggerId, [47](#), [47](#)  
DialogActions, [48](#)  
DialogContent, [49](#)  
DialogContentText, [49](#)

DialogTitle, [50](#)  
Divider, [51](#)  
Drawer, [52](#)  
Drawer.triggerId, [53](#), [54](#)

Fab, [54](#)  
Fade, [56](#)  
FilledInput, [57](#)  
FormControl, [59](#)  
FormControlLabel, [61](#)  
FormGroup, [62](#)  
FormHelperText, [63](#)  
FormLabel, [64](#)

GlobalStyles, [65](#)  
Grid, [66](#)  
Grow, [67](#)

Icon, [68](#)  
IconButton, [69](#)  
ImageList, [70](#)  
ImageListItem, [71](#)  
ImageListItemBar, [72](#)  
Input, [73](#)  
InputAdornment, [75](#)  
InputBase, [76](#)  
InputLabel, [79](#)

LinearProgress, [80](#)  
Link, [81](#)  
List, [82](#)  
ListItem, [83](#)  
ListItemAvatar, [84](#)  
ListItemButton, [85](#)  
ListItemIcon, [86](#)  
ListItemSecondaryAction, [87](#)  
ListItemText, [88](#)  
ListSubheader, [89](#)  
LoadingButton, [90](#)

Masonry, [91](#)

- Menu, [92](#)
- Menu.triggerId, [93, 94](#)
- MenuItem, [94](#)
- MenuList, [96](#)
- MobileStepper, [97](#)
- Modal, [98](#)
- Modal.triggerId, [100, 100](#)
- muiMaterialDependency, [101](#)
- muiMaterialExample, [101](#)
- muiMaterialPage, [102](#)
  
- NativeSelect, [103](#)
- NoSsr, [104](#)
  
- OutlinedInput, [105](#)
  
- Pagination, [108](#)
- PaginationItem, [110](#)
- Paper, [111](#)
- Popover, [112](#)
- Popover.triggerId, [114](#)
- Popper, [115](#)
- Portal, [116](#)
- print.muiMaterial, [117](#)
  
- Radio, [118](#)
- RadioGroup, [120](#)
- RadioGroup.shinyInput, [119](#)
- Rating, [121](#)
  
- ScopedCssBaseline, [123](#)
- Select, [124](#)
- Skeleton, [126](#)
- Slide, [127](#)
- Slider, [128](#)
- Snackbar, [131](#)
- SnackbarContent, [133](#)
- SpeedDial, [134](#)
- SpeedDialAction, [136](#)
- SpeedDialIcon, [137](#)
- Stack, [138](#)
- Step, [139](#)
- StepButton, [140](#)
- StepConnector, [141](#)
- StepContent, [142](#)
- StepIcon, [143](#)
- StepLabel, [144](#)
- Stepper, [145](#)
- SvgIcon, [146](#)
  
- SwipeableDrawer, [147](#)
- SwipeableDrawer.triggerId, [149](#)
- Switch, [149](#)
  
- Tab, [151](#)
- TabContext, [152](#)
- Table, [154](#)
- TableBody, [155](#)
- TableCell, [155](#)
- TableContainer, [157](#)
- TableFooter, [157](#)
- TableHead, [158](#)
- TablePagination, [159](#)
- TableRow, [161](#)
- TableSortLabel, [162](#)
- TabList, [163](#)
- TabList.shinyInput, [153](#)
- TabPanel, [164](#)
- Tabs, [165](#)
- TabScrollButton, [167](#)
- TextareaAutosize, [168](#)
- TextField, [169](#)
- ThemeProvider, [172](#)
- Timeline, [173](#)
- TimelineConnector, [174](#)
- TimelineContent, [174](#)
- TimelineDot, [175](#)
- TimelineItem, [176](#)
- TimelineOppositeContent, [177](#)
- TimelineSeparator, [177](#)
- ToggleButton, [178](#)
- ToggleButtonGroup, [179](#)
- Toolbar, [181](#)
- Tooltip, [182](#)
- Typography, [184](#)
  
- updateAutocomplete.shinyInput  
  (Autocomplete), [12](#)
- updateBottomNavigation.shinyInput  
  (BottomNavigation), [23](#)
- updateButton.shinyInput (Button), [27](#)
- updateCheckbox.shinyInput (Checkbox), [37](#)
- updateDialog.shinyInput (Dialog), [45](#)
- updateDrawer.shinyInput (Drawer), [52](#)
- updateFab.shinyInput (Fab), [54](#)
- updateFilledInput.shinyInput  
  (FilledInput), [57](#)
- updateFormControlLabel.shinyInput  
  (FormControlLabel), [61](#)

updateIconButton.shinyInput  
(IconButton), [69](#)

updateInput.shinyInput (Input), [73](#)

updateLoadingButton.shinyInput  
(LoadingButton), [90](#)

updateMenu.shinyInput (Menu), [92](#)

updateMenuItem.shinyInput (MenuItem), [94](#)

updateModal.shinyInput (Modal), [98](#)

updateNativeSelect.shinyInput  
(NativeSelect), [103](#)

updateOutlinedInput.shinyInput  
(OutlinedInput), [105](#)

updatePagination.shinyInput  
(Pagination), [108](#)

updateRadio.shinyInput (Radio), [118](#)

updateRadioGroup.shinyInput  
(RadioGroup), [120](#)

updateRating.shinyInput (Rating), [121](#)

updateSelect.shinyInput (Select), [124](#)

updateSlider.shinyInput (Slider), [128](#)

updateSnackbar.shinyInput (Snackbar),  
[131](#)

updateStepButton.shinyInput  
(StepButton), [140](#)

updateSwitch.shinyInput (Switch), [149](#)

updateTabContext.shinyInput  
(TabContext), [152](#)

updateTabList.shinyInput (TabList), [163](#)

updateTabPanel.shinyInput (TabPanel),  
[164](#)

updateTabs.shinyInput (Tabs), [165](#)

updateTextField.shinyInput (TextField),  
[169](#)

updateToggleButton.shinyInput  
(ToggleButton), [178](#)

updateToggleButtonGroup.shinyInput  
(ToggleButtonGroup), [179](#)

Zoom, [185](#)