

Toggle Button React component

Toggle Buttons

Toggle buttons can be used to group related options.

To emphasize groups of related Toggle buttons, a group should share a common container. The `ToggleButtonGroup` controls the selected state of its child buttons when given its own `value` prop.

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

Exclusive selection

With exclusive selection, selecting one option deselects any other.

In this example, text justification toggle buttons present options for left, center, right, and fully justified text (disabled), with only one item available for selection at a time.

Note: Exclusive selection does not enforce that a button must be active. For that effect see `enforceValue` prop.

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

Multiple selection

Multiple selection allows for logically-grouped options, like bold, italic, and underline, to have multiple options selected.

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

Size

For larger or smaller buttons, use the `size` prop.

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

Color

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

Vertical buttons

The buttons can be stacked vertically with the `orientation` prop set to “vertical”.

```
{{“demo”: “VerticalToggleButtons.js”}}
```

Enforce value set

If you want to enforce that at least one button must be active, you can adapt your `handleChange` function.

```
const handleAlignment = (event, newAlignment) => {  
  if (newAlignment !== null) {  
    setAlignment(newAlignment);  
  }  
};
```

```
const handleDevices = (event, newDevices) => {  
  if (newDevices.length) {  
    setDevices(newDevices);  
  }  
};
```

```
{{“demo”: “ToggleButtonNotEmpty.js”}}
```

Standalone toggle button

```
{{“demo”: “StandaloneToggleButton.js”}}
```

Customization

Here is an example of customizing the component. You can learn more about this in the [overrides](#) documentation page.

```
{{“demo”: “CustomizedDividers.js”, “bg”: true}}
```

Accessibility

ARIA

- `ToggleButtonGroup` has `role="group"`. You should provide an accessible label with `aria-label="label"`, `aria-labelledby="id"` or `<label>`.
- `ToggleButton` sets `aria-pressed="<bool>"` according to the button state. You should label each button with `aria-label`.

Keyboard

At present, toggle buttons are in DOM order. Navigate between them with the tab key. The button behavior follows standard keyboard semantics.