Slider

Sliders allow users to make selections from a range of values.

Sliders reflect a range of values along a bar, from which users may select a single value. They are ideal for adjusting settings such as volume, brightness, or applying image filters.

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

Continuous sliders

Continuous sliders allow users to select a value along a subjective range.

{{"demo": "ContinuousSlider.js"}}

Sizes

For smaller slider, use the prop `size="small"` .

{{"demo": "SliderSizes.js"}}

Discrete sliders

Discrete sliders can be adjusted to a specific value by referencing its value indicator. You can generate a mark for each step with `marks={true}` .

{{"demo": "DiscreteSlider.js"}}

Small steps

You can change the default step increment.

{{"demo": "DiscreteSliderSteps.js"}}

Custom marks

You can have custom marks by providing a rich array to the `marks` prop.

{{"demo": "DiscreteSliderMarks.js"}}

Restricted values

You can restrict the selectable values to those provided with the `marks` prop with `step={null}` .

{{"demo": "DiscreteSliderValues.js"}}

Label always visible

You can force the thumb label to be always visible with `valueLabelDisplay="on"` .

{{"demo": "DiscreteSliderLabel.js"}}

Range slider

The slider can be used to set the start and end of a range by supplying an array of values to the `value` prop.

{{"demo": "RangeSlider.js"}}

Minimum distance

You can enforce a minimum distance between values in the `onChange` event handler. By default, when you move the pointer over a thumb while dragging another thumb, the active thumb will swap to the hovered thumb. You can disable this behavior with the `disableSwap` prop. If you want the range to shift when reaching minimum distance, you can utilize the `activeThumb` parameter in `onChange`.

{{"demo": "MinimumDistanceSlider.js"}}

Slider with input field

In this example, an input allows a discrete value to be set.

{{"demo": "InputSlider.js"}}

Color

{{"demo": "ColorSlider.js"}}

Customization

Here are some examples of customizing the component. You can learn more about this in the overrides documentation page.

{{"demo": "CustomizedSlider.js"}}

Music player

{{"demo": "MusicPlayerSlider.js"}}

Vertical sliders

{{"demo": "VerticalSlider.js"}}

WARNING: Chrome, Safari and newer Edge versions i.e. any browser based on WebKit exposes `<Slider orientation="vertical" />` as horizontal (chromium issue #1158217). By applying `-webkit-appearance: slider-vertical;` the slider is exposed as vertical.

However, by applying `-webkit-appearance: slider-vertical;` keyboard navigation for horizontal keys (`Arrow Left`, `Arrow Right`) is reversed (chromium issue #1162640). Usually, up and right should increase and left and down should decrease the value. If you apply `-webkit-appearance` you could prevent keyboard navigation for horizontal arrow keys for a truly vertical slider. This might be less confusing to users compared to a change in direction.

{{"demo": "VerticalAccessibleSlider.js"}}

Track

The track shows the range available for user selection.

Removed track

The track can be turned off with `track={false}` .

{{"demo": "TrackFalseSlider.js"}}

Inverted track

The track can be inverted with `track="inverted"` .

{{"demo": "TrackInvertedSlider.js"}}

Non-linear scale

You can use the `scale` prop to represent the `value` on a different scale.

In the following demo, the value *x* represents the value *2^x*. Increasing *x* by one increases the represented value by factor *2*.

{{"demo": "NonLinearSlider.js"}}

Accessibility

(WAI-ARIA: https://www.w3.org/TR/wai-aria-practices/#slider)

The component handles most of the work necessary to make it accessible. However, you need to make sure that:

- Each thumb has a user-friendly label (`aria-label` , `aria-labelledby` or `getAriaLabel` prop).
- Each thumb has a user-friendly text for its current value. This is not required if the value matches the semantics of the label. You can change the name with the `getAriaValueText` or `aria-valuetext` prop.

Limitations

IE 11

The slider's value label is not centered in IE 11. The alignement is not handled to make customizations easier with the lastest browsers. You can solve the issue with:

```css
.MuiSlider-valueLabel {
  left: calc(-50% - 4px);
}
```