

Unstyled React Slider component and hook

Unstyled slider

The `SliderUnstyled` component lets users make selections from a range of values along a horizontal or vertical bar.

Users may need to select a single value or a range of values on a slider. They are ideal for interface controls that benefit from a visual representation of adjustable content, such as volume or brightness settings, or for applying image filters.

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

```
import SliderUnstyled from '@mui/base/SliderUnstyled';
```

```
{{"demo": "UnstyledSlider.js", "defaultCodeOpen": false}}
```

Discrete sliders

The most basic slider is *continuous*, which means it does not have pre-defined (*discrete*) values for the user to select from. This is suitable for situations in which an approximate value is good enough for the user, such as brightness or volume.

But if your users need more precise options, you can create a discrete slider that snaps the thumb to pre-defined stops along the bar.

To generate a mark for each stop, use `marks={true}`:

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

Custom marks

You can create custom marks by providing a rich array to the `marks` prop:

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

Restricted values

If the user should only be able to select from the values provided with the `marks` prop, add `step=null` to disable all other options:

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

Range slider

To let users set the start and end of a range on a slider, provide an array of values to the `value` or `defaultValue` prop:

```
{{“demo”: “RangeSlider.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.