

react-slider

Filter by name

ReactSlider

ReactSlider



```
import ReactSlider from 'react-slider' 
```

PROPS & METHODS

Prop name	Type	Default	Description
ariaLabel	union		<p>aria-label for screen-readers to apply to the thumbs. Use an array for more than one thumb. The length of the array must match the number of thumbs in the value array.</p> <p>One of type: <code>string</code>, <code>string[]</code></p>
ariaLabelledby	union		<p>aria-labelledby for screen-readers to apply to the thumbs. Used when slider rendered with separate label. Use an array for more than one thumb. The length of the array must match the number of thumbs in the value array.</p> <p>One of type: <code>string</code>, <code>string[]</code></p>
ariaValuetext	union		<p>aria-valuetext for screen-readers. Can be a static string, or a function that returns a string. The function will be passed a single argument, an object with the following properties:</p> <div><pre>state => `Value: \${state.value}`</pre></div> <ul style="list-style-type: none"><code>state.index</code> {number} the index of the thumb<code>state.value</code> {number array} the current value state<code>state.valueNow</code> {number} the value of the thumb (i.e. <code>aria-valuenow</code>) <p>One of type: <code>string</code>, <code>func</code></p>
className	string	slider	The css class set on the slider node.
defaultValue	union	0	<p>Determines the initial positions of the thumbs and the number of thumbs.</p> <p>If a number is passed a slider with one thumb will be rendered. If an array is passed each value will determine the position of one thumb. The values in the array must be sorted.</p> <p>One of type: <code>number</code>, <code>number[]</code></p>
disabled	bool	false	If true the thumbs can't be moved.
invert	bool	false	Inverts the slider.

react-slider

ReactSlider

<code>markClassName</code>	<code>string</code>	<code>mark</code>	The css class set on the marks.
<code>marks</code>	<code>union</code>	<code>[]</code>	Shows passed marks on the track, if true it shows all the marks, if an array of numbers it shows just the passed marks, if a number is passed it shows just the marks in that steps: like passing 3 shows the marks 3, 6, 9 One of type: <code>number[]</code> , <code>bool</code> , <code>number</code>
<code>max</code>	<code>number</code>	<code>100</code>	The maximum value of the slider.
<code>min</code>	<code>number</code>	<code>0</code>	The minimum value of the slider.
<code>minDistance</code>	<code>number</code>	<code>0</code>	The minimal distance between any pair of thumbs. Must be positive, but zero means they can sit on top of each other.
<code>onAfterChange</code>	<code>func</code>		Callback called only after moving a thumb has ended. The callback will only be called if the action resulted in a change. The function will be called with two arguments, the first being the result value(s) the second being thumb index.
<code>onBeforeChange</code>	<code>func</code>		Callback called before starting to move a thumb. The callback will only be called if the action will result in a change. The function will be called with two arguments, the first being the initial value(s) the second being thumb index.
<code>onChange</code>	<code>func</code>		Callback called on every value change. The function will be called with two arguments, the first being the new value(s) the second being thumb index.
<code>onSliderClick</code>	<code>func</code>		Callback called when the the slider is clicked (thumb or tracks). Receives the value at the clicked position as argument.
<code>orientation</code>	<code>enum</code>	<code>horizontal</code>	Determines whether the slider moves horizontally (from left to right) or vertically (from top to bottom). One of: <code>horizontal</code> , <code>vertical</code>
<code>pageFn</code>	<code>func</code>	<code>Function</code>	The result of the function is the value to be added or subtracted when the Page Up or Page Down keys are pressed. The current step value will be passed as the only argument. By default, paging will modify step by a factor of 10.
<code>pearling</code>	<code>bool</code>	<code>false</code>	If true the active thumb will push other thumbs within the constraints of min, max, step and

react-slider

ReactSlider

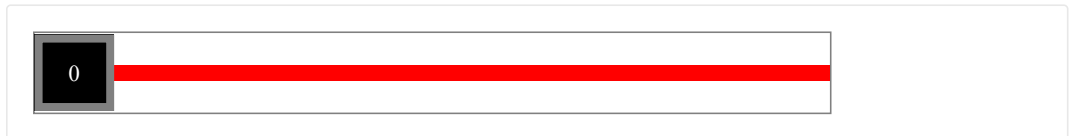
<code>renderMark</code>	<code>func</code>	Function	<p>Provide a custom render function for the mark node. The render function will be passed one argument, an object with props that should be added to your handle node</p> <pre>(props) => </pre> <ul style="list-style-type: none"> • <code>props {object}</code> props to be spread into your track node
<code>renderThumb</code>	<code>func</code>	Function	<p>Provide a custom render function for dynamic thumb content. The render function will be passed two arguments, an object with props that should be added to your thumb node, and an object with thumb and slider state:</p> <pre>(props, state) => <div {...props} /></pre> <ul style="list-style-type: none"> • <code>props {object}</code> props to be spread into your thumb node • <code>state.index {number}</code> the index of the thumb • <code>state.value {number array}</code> the current value state • <code>state.valueNow {number}</code> the value of the thumb (i.e. <code>aria-valuenow</code>)
<code>renderTrack</code>	<code>func</code>	Function	<p>Provide a custom render function for the track node. The render function will be passed two arguments, an object with props that should be added to your handle node, and an object with track and slider state:</p> <pre>(props, state) => <div {...props} /></pre> <ul style="list-style-type: none"> • <code>props {object}</code> props to be spread into your track node • <code>state.index {number}</code> the index of the track • <code>state.value {number array}</code> the current value state
<code>snapDragDisabled</code>	<code>bool</code>	false	Disables thumb move when clicking the slider track
<code>step</code>	<code>number</code>	1	Value to be added or subtracted on each step the slider makes. Must be greater than zero. <code>max - min</code> should be evenly divisible by the step value.
<code>thumbActiveClassName</code>	<code>string</code>	active	The css class set on the thumb that is currently being moved.
<code>thumbClassName</code>	<code>string</code>	thumb	<p>The css class set on each thumb node.</p> <p>In addition each thumb will receive a numbered css class of the form <code>`\${thumbClassName}-\${i}</code>, e.g. <code>thumb-0</code>, <code>thumb-1</code>, ...</p>

react-slider

ReactSlider

<code>trackClassName</code>	<code>string</code>	track	The css class set on the tracks between the thumbs. In addition track fragment will receive a numbered css class of the form <code>\${trackClassName}-\${i}</code> , e.g. <code>track-0</code> , <code>track-1</code> , ...
<code>value</code>	<code>union</code>		Like <code>defaultValue</code> but for controlled components . One of type: <code>number</code> , <code>number[]</code>
<code>withTracks</code>	<code>bool</code>	<code>true</code>	If true tracks between the thumbs will be rendered.

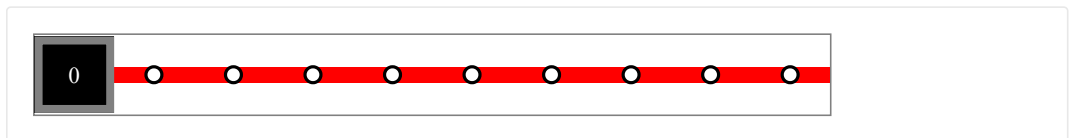
Single slider, similar to `<input type="range" defaultValue={0} />`



VIEW CODE



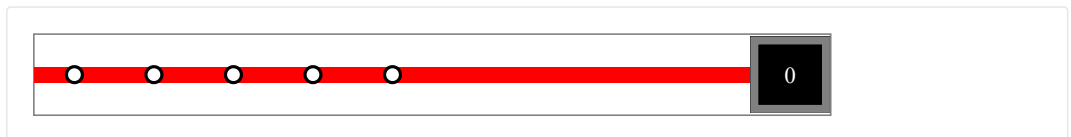
Slider with marks



VIEW CODE



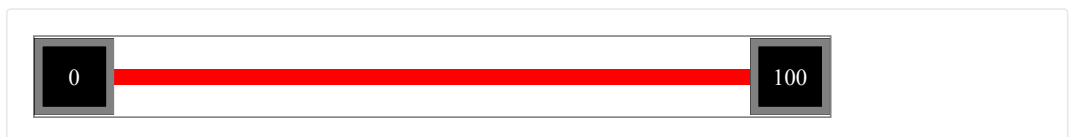
An inverted slider with custom marks



VIEW CODE



Double slider



VIEW CODE

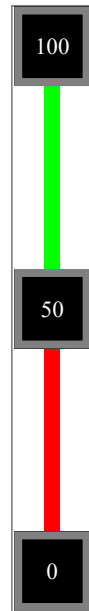


Multi slider

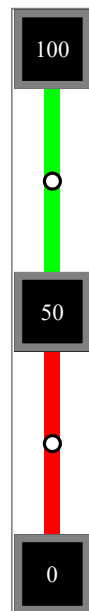
react-slider

[ReactSlider](#)[VIEW CODE](#)

Vertical slider

[VIEW CODE](#)

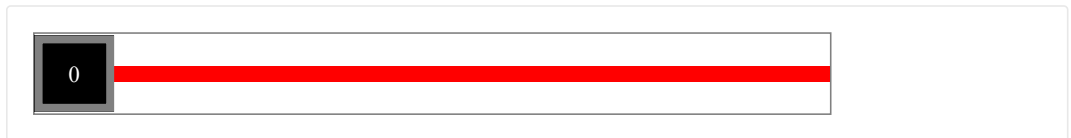
Vertical slider with marks at an interval

[VIEW CODE](#)

react-slider

[ReactSlider](#)

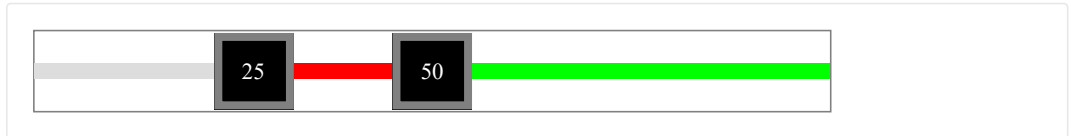
Track changes with `onBeforeChange`, `onChange`, and `onAfterChange` event handlers



VIEW CODE



Using the `onChange` event handler, you can use the slider as a controlled component



VIEW CODE



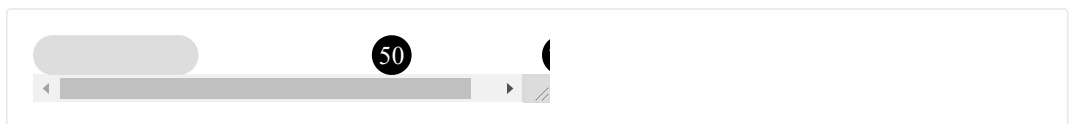
Custom styling using [styled-components](#)



VIEW CODE



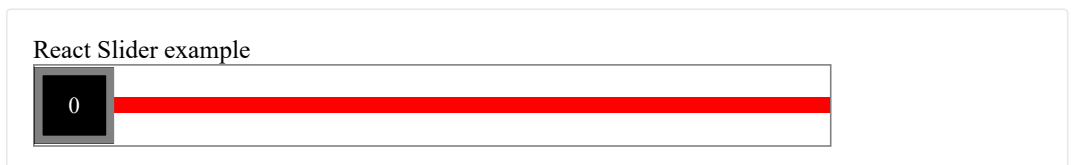
In some case you may need to programmatically tell the slider to resize, for example if the parent container is resizing independently of the window. Since `v2.0.0` this is handled internally by `ResizeObserver`.



VIEW CODE



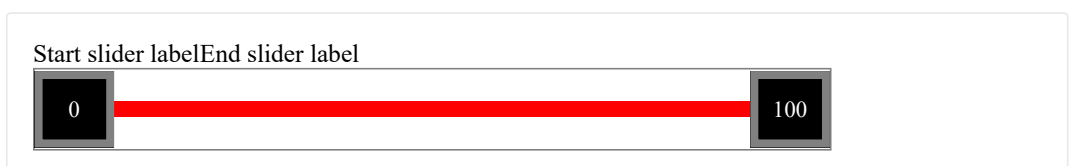
Single slider, applying `ariaLabelledby` to establish association with a label



VIEW CODE



Double slider, applying `ariaLabelledby` as an array to multiple thumb labels



VIEW CODE



react-slider

Created with [React Styleguidist](#)

ReactSlider