Slider

Drag the slider within a fixed range.

Basic usage

The current value is displayed when the slider is being dragged.

:::demo Customize the initial value of the slider by setting the binding value.

```
<template>
  <div class="block">
    <span class="demonstration">Default value</span>
    <el-slider v-model="value1"></el-slider>
  </div>
  <div class="block">
    <span class="demonstration">Customized initial value</span>
    <el-slider v-model="value2"></el-slider>
  </div>
  <div class="block">
    <span class="demonstration">Hide Tooltip</span>
    <el-slider v-model="value3" :show-tooltip="false"></el-slider>
  </div>
  <div class="block">
    <span class="demonstration">Format Tooltip</span>
    <el-slider v-model="value4" :format-tooltip="formatTooltip"></el-slider>
  </div>
  <div class="block">
    <span class="demonstration">Disabled</span>
    <el-slider v-model="value5" disabled></el-slider>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        value1: 0,
        value2: 50,
        value3: 36,
        value4: 48,
        value5: 42
      }
    },
    methods: {
      formatTooltip(val) {
        return val / 100;
      }
    }
  }
</script>
```

:::

Discrete values

The options can be discrete.

:::demo Set step size with the `step` attribute. You can display breakpoints by setting the `show-stops` attribute.

```
<template>
  <div class="block">
    <span class="demonstration">Breakpoints not displayed</span>
    <el-slider
      v-model="value1"
      :step="10">
    </el-slider>
  </div>
  <div class="block">
    <span class="demonstration">Breakpoints displayed</span>
    <el-slider
      v-model="value2"
      :step="10"
      show-stops>
    </el-slider>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        value1: 0,
        value2: 0
      }
    }
  }
</script>
```

:::

Slider with input box

Set value via a input box.

:::demo Set the `show-input` attribute to display an input box on the right.

```
<template>
  <div class="block">
    <el-slider
      v-model="value"
      show-input>
    </el-slider>
  </div>
```

```
</template>

<script>
  export default {
    data() {
      return {
        value: 0
      }
    }
  }
</script>
```

:::

Range selection

Selecting a range of values is supported.

:::demo Setting the `range` attribute activates range mode, where the binding value is an array made up of two boundary values.

```
<template>
  <div class="block">
    <el-slider
      v-model="value"
      range
      show-stops
      :max="10">
    </el-slider>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        value: [4, 8]
      }
    }
  }
</script>
```

:::

Vertical mode

:::demo Setting the `vertical` attribute to `true` enables vertical mode. In vertical mode, the `height` attribute is required.

```
<template>
  <div class="block">
    <el-slider
```

```
      v-model="value"
      vertical
      height="200px">
    </el-slider>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        value: 0
      }
    }
  }
</script>
```

:::

Show marks

:::demo Setting this `marks` attribute can show mark on slider.

```
<template>
  <div class="block">
    <el-slider
      v-model="value"
      range
      :marks="marks">
    </el-slider>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        value: [30, 60],
        marks: {
          0: '0°C',
          8: '8°C',
          37: '37°C',
          50: {
            style: {
              color: '#1989FA'
            },
            label: this.$createElement('strong', '50%')
          }
        }
      }
    }
  }
</script>
```

:::

Attributes

Attribute	Description	Type	Accepted Values	Default
value / v-model	binding value	number	—	0
min	minimum value	number	—	0
max	maximum value	number	—	100
disabled	whether Slider is disabled	boolean	—	false
step	step size	number	—	1
show-input	whether to display an input box, works when `range` is false	boolean	—	false
show-input-controls	whether to display control buttons when `show-input` is true	boolean	—	true
input-size	size of the input box	string	large / medium / small / mini	small
show-stops	whether to display breakpoints	boolean	—	false
show-tooltip	whether to display tooltip value	boolean	—	true
format-tooltip	format to display tooltip value	function(value)	—	—
range	whether to select a range	boolean	—	false
vertical	vertical mode	boolean	—	false
height	Slider height, required in vertical mode	string	—	—
label	label for screen reader	string	—	—
debounce	debounce delay when typing, in milliseconds, works when `show-input` is true	number	—	300
tooltip-class	custom class name for the tooltip	string	—	—
marks	marks， type of key must be `number` and must in	object	—	—

	closed interval `[min, max]`, each mark can custom style			

Events

Event Name	Description	Parameters
change	triggers when the value changes (if the mouse is being dragged, this event only fires when the mouse is released)	value after changing
input	triggers when the data changes (It'll be emitted in real time during sliding)	value after changing