

Range-Slider

https://playdocs1.orangeriver-ad055946.westus2.azurecontainerapps.io/play-docs/docs/ui-components/Inputs/Range-Slider

Range Slider

TheRange Slidercomponent provides a simple and elegant way to select a numeric value or range within a defined interval.It features smooth drag interactions, built-in accessibility, and easy integration with Angular forms for both template-driven and reactive use cases.

Thecomponent supports multiple modes such assingle value,multi-range, andinput-integratedsliders, making it adaptable to diverse UI needs.

```
<aava-slider>
```

How to use

```
import { AavaSliderComponent } from "@aava/play-core";
```

```
import { AavaSliderComponent } from "@aava/play-core";
```

Basic Usage

```
<aava-slider [value]="50" [min]="0" [max]="100" [step]="1" (valueChange)="onSliderChange($event)"></aava-slider>
```

```
<aava-slider [value]="50" [min]="0" [max]="100" [step]="1" (valueChange)="onSliderChange($event)"></aava-slider>
```

```
onSliderChange(value: number) { console.log('Single slider value:', value); }
```

```
onSliderChange(value: number) { console.log('Single slider value:', value); }
```

The simplest version of the slider displays a single draggable handle to choose a numeric value between a default range of 0-100. Ideal for scenarios like volume control, brightness adjustment, or progress selection.

Size Variants

```
<aava-slider size="sm" [value]="30" [min]="0" [max]="100"> </aava-slider><aava-slider size="md" [value]="70" [min]="0" [max]="100"> </aava-slider>
```

```
<aava-slider size="sm" [value]="30" [min]="0" [max]="100"> </aava-slider><aava-slider size="md" [value]="70" [min]="0" [max]="100"> </aava-slider>
```

The slider supports multiple size options to fit different design requirements and layouts. Smaller sliders suit compact UIs, while medium sizes provide comfortable interaction for most use cases.

Available Size

- sm (Small): Compact slider ideal for dense layouts and mobile interfaces
- md (Medium): Standard size slider for most common use cases (default)

States

```
<aava-slider [value]="50" [min]="0" [max]="100"> </aava-slider><h1>Normal State</h1><aava-slider [value]="50" [min]="0" [max]="100" [disabled]="true"></aava-slider><h1>Disabled State</h1><aava-slider type="input" [value]="75" [min]="0" [max]="100"> </aava-slider><h1>Input Variant</h1><aava-slider [value]="50" [min]="0" [max]="100" [showTooltip]="false"></aava-slider><h1>Without Tooltip</h1>
```

```
<aava-slider [value]="50" [min]="0" [max]="100"> </aava-slider><h1>Normal State</h1><aava-slider [value]="50" [min]="0" [max]="100" [disabled]="true"></aava-slider><h1>Disabled State</h1><aava-slider type="input" [value]="75" [min]="0" [max]="100"> </aava-slider><h1>Input Variant</h1><aava-slider [value]="50" [min]="0" [max]="100" [showTooltip]="false"></aava-slider><h1>Without Tooltip</h1>
```

Demonstrates the slider’s different states, including disabled, active, and focused. These states help communicate interactivity and status changes to users clearly.

Multi Range Slider

```
<aava-slider [multiRange]="true" [min]="0" [max]="100" [minValue]="minValue" [maxValue]="maxValue" (minValueChange)="onMinChange($event)" (maxValueChange)="onMaxChange($event)"></aava-slider>
```

```
<aava-slider [multiRange]="true" [min]="0" [max]="100" [minValue]="minValue" [maxValue]="maxValue" (minValueChange)="onMinChange($event)" (maxValueChange)="onMaxChange($event)"></aava-slider>
```

```
    minValue = 20; maxValue = 80; onMinChange(value: number) {    this.minValue = value;    console.log('Min value changed:', value);  } onMaxChange(value: number) {    this.maxValue = value;    console.log('Max value changed:', value);  }
```

```
    minValue = 20; maxValue = 80; onMinChange(value: number) {    this.minValue = value;    console.log('Min value changed:', value);  } onMaxChange(value: number) {    this.maxValue = value;    console.log('Max value changed:', value);  }
```

The multi-range version allows selection of both minimum and maximum values, offering a more flexible range selection experience. It’s ideal for use in filters, price sliders, or any range-based data input scenarios.

Multi Range Features

- Dual Handles: Independent control of minimum and maximum values
- Range Selection: Visual indication of selected range between handles
- Collision Prevention: Handles cannot cross over each other
- Synchronized Tooltips: Both handles show their respective values

Input Type Variant

```
<aava-slider type="default" [value]="50" [min]="0" [max]="100" [showTooltip]="true"></aava-slider><h1>Default Type</h1><aava-slider type="input" [value]="75" [min]="0" [max]="100"> </aava-slider><h1>Input Type</h1>
```

```
<aava-slider type="default" [value]="50" [min]="0" [max]="100" [showTooltip]="true"></aava-slider><h1>Default Type</h1><aava-slider type="input" [value]="75" [min]="0" [max]="100"> </aava-slider><h1>Input Type</h1>
```

```
    currentValue = 50; onSliderChange(value: number) {    this.currentValue = value;    console.log('Slider value:', value);  }
```

```
    currentValue = 50; onSliderChange(value: number) {    this.currentValue = value;    console.log('Slider value:', value);  }
```

This variant combines a slider handle with a numeric input field, enabling precise manual entry alongside drag interaction. It’s especially useful in cases where exact numeric control is needed, such as filtering or budget ranges.

Input Type Features

- Dual Input Methods: Users can drag the slider or type directly in the input field
- Real-time Sync: Input field and slider stay synchronized
- Validation: Input respects min/max boundaries and step values
- Accessibility: Input field provides keyboard navigation alternative
- Responsive Design: Input field adapts to slider size variants

Icon Slider Variants

```
<aava-slider [min]="0" [max]="100" [value]="50" [iconStart]='volume-x' [iconEnd]='volume-2'></aava-slider><h4>Volume Control with Start/End Icons</h4><aava-slider [min]="0" [max]="100" [value]="75" [handleIcon]='sun' [iconStart]='moon' [iconEnd]='sun'></aava-slider><h4>Brightness with Handle Icon</h4>
```

```
<aava-slider [min]="0" [max]="100" [value]="50" [iconStart]='volume-x' [iconEnd]='volume-2'></aava-slider><h4>Volume Control with Start/End Icons</h4><aava-slider [min]="0" [max]="100" [value]="75" [handleIcon]='sun' [iconStart]='moon' [iconEnd]='sun'></aava-slider><h4>Brightness with Handle Icon</h4>
```

```
currentValue = 50; onSliderChange(value: number) { this.currentValue = value; console.log('Slider value:', value); }
```

```
currentValue = 50; onSliderChange(value: number) { this.currentValue = value; console.log('Slider value:', value); }
```

Customizable slider with icon-based thumbs for enhanced visual feedback and thematic consistency.

Icon Thumb Features

- Custom Icons: Replace default handle with Lucide icons
- Thematic Consistency: Icons that match your content context
- Multiple Variants: Various icon styles for different use cases
- Responsive Sizing: Icons scale appropriately with slider size
- Color Theming: Icons inherit slider theme colors
- Hover Effects: Enhanced visual feedback on interaction

Icon Thumb Variants

- Volume Control: Speaker/volume icons for audio controls
- Brightness: Sun/brightness icons for display settings
- Temperature: Thermometer icons for climate controls
- Speed: Gauge/speedometer icons for rate adjustments
- Rating: Star icons for rating and review systems
- Progress: Arrow or progress icons for completion tracking

Orientation

```
<aava-slider [value]="100" [min]="0" [max]="200" [step]="10"> </aava-slider><h1>Custom Range (0-200)</h1><aava-slider [value]="0.5" [min]="0" [max]="1" [step]="0.1"> </aava-slider><h1>Decimal Steps (0.1)</h1>
```

```
<aava-slider [value]="100" [min]="0" [max]="200" [step]="10"> </aava-slider><h1>Custom Range (0-200)</h1><aava-slider [value]="0.5" [min]="0" [max]="1" [step]="0.1"> </aava-slider><h1>Decimal Steps (0.1)</h1>
```

```
currentValue = 50; onSliderChange(value: number) { this.currentValue = value; console.log('Slider value:', value); }
```

```
currentValue = 50; onSliderChange(value: number) { this.currentValue = value; console.log('Slider value:', value); }
```

Accessibility

Built-in accessibility features ensuring WCAG compliance and inclusive user experience.

Accessibility Features

- Keyboard Navigation: Arrow keys, Home, and End key support
- ARIA Attributes: Properrole="slider",aria-valuemin,aria-valuemax,aria-valuenow
- Focus Management: Clear focus indicators and outline
- Touch Support: Optimized for touch devices
- Input Integration: Numeric input field provides alternative input method

| |
|---------------|
| role="slider" |
| aria-valuemin |
| aria-valuemax |
| aria-valuenow |

Keyboard Controls

- Arrow Right/Up: Increase value by step amount
- Arrow Left/Down: Decrease value by step amount
- Home: Jump to minimum value
- End: Jump to maximum value

API Reference

Inputs

| Property | Type | Default | Description |
|-----------------|---------------------------------|-----------|--|
| min | number | 0 | Minimum value of the slider range |
| max | number | 100 | Maximum value of the slider range |
| value | number | 0 | Current value of the slider |
| step | number | 1 | Step increment for value changes |
| showTooltip | boolean | true | Whether to display the value tooltip |
| size | 'sm' 'md' 'md' | 'md' | Size variant of the slider |
| type | 'default' 'input' 'default' | 'default' | Display type with or without input field |
| multiRange | boolean | false | Enable multi-range (two-handle) slider |
| minValue | number | 20 | Minimum selected value in multi-range mode |
| maxValue | number | 80 | Maximum selected value in multi-range mode |
| iconStart | string | '' | Icon displayed at the start of the slider track |
| iconEnd | string | '' | Icon displayed at the end of the slider track |
| handleIcon | string | '' | Icon displayed on the slider handle |
| handleIconStart | string | '' | Icon displayed on the start handle (multi-range) |
| handleIconEnd | string | '' | Icon displayed on the end handle (multi-range) |
| customStyles | Record | {} | CSS custom properties override |
| disabled | boolean | false | Disable the slider |

| |
|--------|
| min |
| number |
| 0 |
| max |
| number |
| 100 |
| value |

| |
|---------------------|
| number |
| 0 |
| step |
| number |
| 1 |
| showTooltip |
| boolean |
| true |
| size |
| 'sm' 'md' |
| 'md' |
| type |
| 'default' 'input' |
| 'default' |
| multiRange |
| boolean |
| false |
| minValue |
| number |
| 20 |
| maxValue |
| number |
| 80 |
| iconStart |
| string |
| ' ' |

| |
|------------------------|
| iconEnd |
| string |
| '' |
| handleIcon |
| string |
| '' |
| handleIconStart |
| string |
| '' |
| handleIconEnd |
| string |
| '' |
| customStyles |
| Record<string, string> |
| {} |
| disabled |
| boolean |
| false |

Outputs

| Event | Type | Description |
|----------------------|--------------|--|
| valueChange | EventEmitter | Emitted when the main slider value changes |
| minValueChange | EventEmitter | Emitted when the minimum value changes |
| maxValueChange | EventEmitter | Emitted when the maximum value changes |
| valueChange | | |
| EventEmitter<number> | | |
| minValueChange | | |
| EventEmitter<number> | | |
| maxValueChange | | |

EventEmitter<number>

Methods

The component implements `ControlValueAccessor` for form integration:

`ControlValueAccessor`

| Method | Parameters | Description |
|--------------------------------|----------------------------|----------------------------|
| <code>writeValue</code> | <code>value: number</code> | Set value programmatically |
| <code>registerOnChange</code> | <code>fn: Function</code> | Register change callback |
| <code>registerOnTouched</code> | <code>fn: Function</code> | Register touched callback |

`writeValue`

`value: number`

`registerOnChange`

`fn: Function`

`registerOnTouched`

`fn: Function`

CSS Custom Properties

The slider supports a wide range of CSS custom properties for theming and customization:

| Property | Description |
|--|--|
| <code>--slider-container-height</code> | Height of the overall slider container |
| <code>--slider-container-gap</code> | Spacing between slider elements |
| <code>--slider-input-gap</code> | Gap between the slider track and input field |
| <code>--slider-size-sm-track-height</code> | Track height for small slider size variant |
| <code>--slider-size-sm-thumb-size</code> | Thumb size for small slider |
| <code>--slider-label-font-size-sm</code> | Label font size for small slider |
| <code>--slider-label-weight-sm</code> | Label font weight for small slider |
| <code>--slider-size-md-track-height</code> | Track height for medium slider size variant |
| <code>--slider-size-md-thumb-size</code> | Thumb size for medium slider |
| <code>--slider-label-font-size-md</code> | Label font size for medium slider |
| <code>--slider-label-weight-md</code> | Label font weight for medium slider |
| <code>--slider-track-height</code> | Height of the slider track |
| <code>--slider-track-background</code> | Background color of the slider track |
| <code>--slider-track-border-radius</code> | Border radius of the track |
| <code>--slider-progress-background</code> | Background color of the filled progress area |
| <code>--slider-progress-border-radius</code> | Border radius of the progress area |
| <code>--slider-thumb-size</code> | Size of the slider thumb |
| <code>--slider-thumb-border-radius</code> | Border radius of the thumb |
| <code>--slider-thumb-inner-background</code> | Background color inside the thumb |
| <code>--slider-thumb-shadow</code> | Shadow of the thumb |
| <code>--slider-thumb-shadow-hover</code> | Thumb shadow on hover |
| <code>--slider-focus-ring</code> | Style of the focus ring |
| <code>--slider-focus-ring-offset</code> | Offset distance of the focus ring |
| <code>--slider-cursor</code> | Cursor style when hovering over slider |
| <code>--slider-tooltip-margin</code> | Margin around the tooltip |
| <code>--slider-tooltip-padding</code> | Padding inside the tooltip |
| <code>--slider-tooltip-border-radius</code> | Border radius of the tooltip |
| <code>--slider-value-color</code> | Text color of the tooltip value |
| <code>--slider-label-font-family</code> | Font family used for labels |
| <code>--slider-label-line-height</code> | Line height for labels |
| <code>--slider-mark-background</code> | Background color of slider marks |
| <code>--slider-handle-icon-width</code> | Width of the handle icon |
| <code>--slider-handle-icon-height</code> | Height of the handle icon |
| <code>--slider-input-width</code> | Width of the input field |
| <code>--slider-input-height</code> | Height of the input field |
| <code>--slider-input-padding</code> | Padding inside the input field |
| <code>--slider-input-border-radius</code> | Border radius of the input field |
| <code>--slider-input-border</code> | Border style of the input field |
| <code>--slider-input-background</code> | Background color of the input field |

--slider-input-font-size | Font size of the input text
--slider-input-font-weight | Font weight of the input text
--slider-input-font-family | Font family of the input text
--slider-input-color | Text color of the input
--slider-input-transition | Transition style for input state changes
--slider-input-focus-border-color | Border color of input when focused
--slider-input-hover-border-color | Border color of input when hovered
--slider-input-disabled-background | Background color of disabled input
--slider-input-disabled-border-color | Border color of disabled input
--slider-value-color-disabled | Text color for disabled value display
--slider-disabled-color | Color used in disabled state
--slider-disabled-rail-background | Background of the slider rail when disabled

--slider-container-height

--slider-container-gap

--slider-input-gap

--slider-size-sm-track-height

--slider-size-sm-thumb-size

--slider-label-font-size-sm

--slider-label-weight-sm

--slider-size-md-track-height

--slider-size-md-thumb-size

--slider-label-font-size-md

--slider-label-weight-md

--slider-track-height

--slider-track-background

--slider-track-border-radius

--slider-progress-background

--slider-progress-border-radius

--slider-thumb-size

--slider-thumb-border-radius

--slider-thumb-inner-background

--slider-thumb-shadow

--slider-thumb-shadow-hover

--slider-focus-ring

--slider-focus-ring-offset

--slider-cursor

--slider-tooltip-margin

--slider-tooltip-padding

--slider-tooltip-border-radius

--slider-value-color

--slider-label-font-family

--slider-label-line-height

--slider-mark-background

--slider-handle-icon-width

--slider-handle-icon-height

--slider-input-width

--slider-input-height

--slider-input-padding

--slider-input-border-radius

--slider-input-border

--slider-input-background

--slider-input-font-size

--slider-input-font-weight

--slider-input-font-family

--slider-input-color

--slider-input-transition

--slider-input-focus-border-color

--slider-input-hover-border-color

--slider-input-disabled-background

--slider-input-disabled-border-color

--slider-value-color-disabled

```
--slider-disabled-color
```

```
--slider-disabled-rail-background
```

Best Practices

Implementation Guidelines

- Use appropriate step values for your use case (1 for integers, 0.1 for decimals)
- Set meaningful min/max boundaries that make sense for your data
- Consider hiding the tooltip for inline sliders in dense layouts
- Always provide proper labels for accessibility
- Choose appropriate size variants based on your layout density
- Use input type for scenarios requiring precise numeric input

Size Selection Guidelines

- Small: Use in compact layouts, mobile interfaces, or when space is limited
- Medium: Default choice for most applications and standard layouts

Input Type Usage

- Default Type: Best for visual-only interactions and quick value selection
- Input Type: Ideal for applications requiring precise numeric input or accessibility compliance

Form Integration

- Use reactive forms for complex validation scenarios
- Implement proper error handling and validation messages
- Consider debouncing frequent value changes for performance
- Leverage input type for better form accessibility and user experience