

range-comp

range-comp is Vue.js (≥ 2.5) web component that wraps around an html input of type 'range' with label and styling options.

range-comp can be installed via with the included `package.json` file for a local installation via the [npm install](#) command. **range-comp** depends on the [vue.js](#) framework. A demo folder is provided that used [Parcel](#) together with its associated `package.json` file to bundle together **range-comp** along with its [vue.js](#) dependency for a simple application. Further details are provided below for running the demo.

Props

A prop in Vue.js is a custom attribute for passing information from a parent component hosting **range-comp** instance(s) to an **range-comp** as a child component.

range-comp has the following props for a parent to bind and send information to:

`heading` -- a heading for the component that can be positioned above, below, or to the left of the input box (string, default: `null`)

`range_value` -- sets the value of **range-comp** (number, default: `null`)

`header_position` -- determines the position of the above `heading` as 'above', 'below', or 'left' (string, default: 'left')

`min, max, step` -- javascript numbers that define **range-comp**'s minimum, maximum and step for selecting range (number, default: 0)

`css_variables` -- defines the css variables for **range-comp** (object, default: `null`)

Styling

The **css_variables** prop is a javascript object that contains any combination of css variable names as keys and associated values for quick, limited styling of **range-comp**. The following list is the css variable names along with their default values:

```
{
  range_comp_font_family: 'Verdana, serif',
  range_comp_heading_color: 'black',
  range_comp_heading_font_weight: 'bold',
  range_comp_thumb_background: 'radial-gradient(ellipse at center, #e4f5fc 0%, #6e6bff 66%)',
  range_comp_track_width: '25rem',
  range_comp_track_background: 'linear-gradient(to bottom, #969696 0, #969696 13%, #5f5f5f 33%, #1e1e1e 64%, #1e1e1e 100%)',
  range_comp_track_focus_background: '#445',
  range_comp_value_color: 'black'
}
```

As an example you could bind the following object to an instance of **range-comp** to set the heading font size and the track width:

```
{range_comp_heading_color: 'gold', range_comp_track_width: '520px'}
```

Note that multiple **range-comp** children of the parent can each be bound to a unique set of `css_variable` prop objects. To enforce the same styling across all **range-comp** children, simply bind just one `css_variable` prop object to all the **input-comp** children.

Events

range-comp has a single one way event (`range_comp_value_changed`) that is emitted to the parent with the current value when the range value changes. The parent can listen for this event and do further processing with the value. Events emitted from a child component back to the parent is explained at [Vue Custom Events](#).

Demonstration

One demonstration of **range-comp** is provided in the folder named `demo/dist` and can be viewed by hosting the `index.html` file. The demo (templated in the `App.vue` file) modifies the layout of the heading, some component styling and listens for changes of the range and show the current value on the browser's console. In addition a button when clicked increases the `range_value` property until it reaches the range maximum, then resets to the range minimum.

As a suggestion, install [http-server](#) locally/globally via [npm](#) then enter the command `http-server` in the **range-comp** `dist` directory. From a browser enter the url: `localhost:8080/` to view the demo.

Here is some example code for using **range-comp** taken from the `App.vue` file:

```
<button v-on:click="set_range_value">Change Value</button>
<range-comp
  heading="Font Size"
  layout="below"
  :range_value="range_value"
  :min="min"
  :max="max"
  :step="step"
  :css_variables="css_variables"
  v-on:range_comp_value_changed="show_value">
</range-comp>
```

And the supporting data references:

```
data() {
  return {
    number_value: 25,
    range_value: null,
    min: 10,
    max: 40,
    step: .5,
    css_variables: {
      range_comp_heading_color: 'white',
      range_comp_input_color: 'white',
      range_comp_input_border_color: 'white',
      range_comp_value_color: 'white'
    }
  }
}
```

```
    }  
  },  
  methods: {  
    //function that responds to change in range  
    show_value: function(value){  
      console.log(value);  
    },  
    set_range_value: function(){  
      this.number_value += .5;  
      if(this.number_value > 39){  
        this.number_value = 10;  
      }  
      this.range_value = this.number_value;  
    }  
  },  
  mounted() {  
    this.range_value = 25;  
  }  
}
```