

input-comp

input-comp is a Vue.js (version ≥ 2.5) component similar to an html input element with alternate layouts for its labeling and a number of css variables for quick styling. In addition, an event is provided to let the component's parent be notified of a change in value.

input-comp can be installed via with the included `package.json` file for a local installation via the `npm install` command. **input-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 **input-comp's** [github location](#) 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 **input-comp** instance(s) to a **input-comp** as a child component.

input-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`)

`input_type` -- a string that sets the type of input--currently supports `text`, `number`, `date`, `checkbox`, `time`, `month`, `color`, `email` (string, default: `text`)

`input_value` -- a property to set the component's initial value if `input_type` is not `checkbox` (string, default: `null`)

`input_checked` -- a property to set the component's initial `checkbox` value if `input_type` is `checkbox` (boolean, `false`)

`placeholder` -- a placeholder for the input box (string, default: 'Enter a value')

`input_size` -- the maximum number of characters for input (string, default: '40') If `input_type` is `number` then this property value sets the width of the component in pixels

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

`single_border` -- determines either full rectangular border or single bottom border for the input box (boolean, default: `false`)

`css_variables` -- defines the css variables for this instance (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 **input-comp**. The following list is the css variable names along with their default values:

```
{
  input_comp_font_family: 'verdana, serif',
  input_comp_heading_color: 'black',
```

```

    input_comp_heading_font_size: '1rem',
    input_comp_heading_font_weight: 'bold',

    input_comp_input_font_size: '.8rem',
    input_comp_input_font_weight: 'normal',
    input_comp_input_color: 'black',
    input_comp_input_background: 'transparent',
    input_comp_input_border_color: 'black',
    input_comp_input_placeholder_color: 'black',
    input_comp_input_focus_outline: 'yellow',
    input_comp_input_focus_background: 'white'
  }

```

As an example you could bind the following object to an instance of **input-comp** to set the heading font size and input color: `{input_comp_heading_font_size: '24px', input_comp_input_color: 'purple'}`

Note that multiple **input-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 **input-comp** children, simply bind just one `css_variable` prop object to all the **input-comp** children.

Events

input-comp has one event that notifies the parent component of the current input value.

input-comp emits the following single named event to its parent:

```
'input_comp_value_changed' -- returns the current value for a specific input component
```

The parent component can listen to the above event and get the current input value for further processing. Events emitted from a child component back to the parent is explained at [Vue Custom Events](#). A typical pattern as shown in the demonstration is to have the parent respond by setting one of its data variables to the event's emitted value.

Demonstration

A demonstration of **input-comp** is provided in the [InputComp](#) repository by hosting the `index.html` file under the `dist` folder. The demo (templated in the `App.vue` file) hosts **input-comp**'s different layouts and types along with component styling and event handling by the parent.

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

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

```

<button v-on:click="set_job">Set Job to janitor</button>
...
...
<input-comp
  heading="Enter your job:"
  placeholder="Enter job here"
  input_size="40"
  :input_value="current_job"
  :css_variables="css_variables"
  v-on:input_comp_value_changed="value => {this.current_job = value}">
</input-comp>

```

Note that the parent responds to the `input_comp_value_changed` event by assigning the emitted value to the parent's `current_job` data variable.

And the supporting data references:

```

data() {
  return {
    single_border: true,

    current_job: "Pipe Fitter",
    ...
    ...
    ...
    css_variables: {
      input_comp_heading_color: 'white',
      input_comp_input_color: 'white',
      input_comp_input_border_color: 'white',
      input_comp_input_placeholder_color: 'white',
      input_comp_input_focus_background: 'black'
    }
  }
},
methods: {
  ...
  ...
  ...

  // set the value of InputComp for job to 'Janitor'
  initialize_job: function(){
    this.current_job = "Janitor";
  }
}

```