select-edit-comp

select-edit-comp is a web component (Vue.js >= 2.5) similar to an HTML select element where an item is selected from a list. In addition to selection, the user can interactively add/delete items from the selection list.

select-edit-comp can be installed via with the included `package.json` file for a local installation via the npm install command. **select-edit-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 **select-edit-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 **select-edit-comp** instance(s) to an **select-edit-comp** as a child component.

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

`items` -- an array of strings that sets the items from which to select (array, default: null)

`select_value` -- when assigned by the parent, sets the current selected value if the value is a member of `items` Can be assigned to a static string or bound to a parent's data member. (string, default: null)

`heading` -- a heading to be displayed above the selection box (string,default: null)

`input_size` -- the maximum number of characters for input (string, default: '20')

`placeholder` -- a string placed in the selection box when no selection has been made (string, default: null)

`show_trash` -- to show the trash icon and for the user to delete an item interactively (boolean, default: true)

`drop_panel_height` -- the height of drop down list panel (string, default: '6.5rem')

`blur_panel` -- if false, will not roll up item panel when component is out of focus (boolean, default: true)

`single_border` -- if a single bottom border is displayed or rectangular (boolean, default: false)

`css_variables` -- defines the css variables for **select-edit-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. The following list are the css variable names along with their default values for a quick styling of **select-edit-comp**:

```
{
    select_edit_font_family: 'Verdana,serif',

    select_edit_arrow_icon: '\21D3',
```

```
      select_edit_arrow_color: 'black',

      select_edit_font_size: '1rem',
      select_edit_color: 'darkgray',
      select_edit_background: 'transparent',
      select_edit_border_color: 'black',

      select_edit_heading_font_size: '1rem',
      select_edit_heading_color: 'black',
      select_edit_heading_font_weight: 'bold',

      select_edit_placeholder_color: 'black',

      select_edit_trash_icon: '\2716',
      select_edit_trash_color: 'red',

      select_edit_items_panel_color: 'black',
      select_edit_items_panel_background: 'white',
      select_edit_items_panel_border: '1px solid black',

      select_edit_item_font_size: '.75rem',
      select_edit_item_hover_box_shadow: '0 0 10px gray',
      select_edit_item_hover_color: 'violet'
  }
```

As an example you could bind from the parent the following object to the `css_variables` prop for setting the border color and font-size of **select-edit-comp** :

```
{select_edit_border_color: 'green', select_edit_font_size: '12px'}
```

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

Events

select-edit-comp has one event that notifies the parent component of the current selected item.

select-edit-comp emits the following single named event:

```
'select_edit_comp_value_changed' -- returns the current value of the component
```

The parent component can listen to the above event and provide a callback for further processing. Events emitted from a child component back to the parent is explained at Vue Custom Events.

The demonstration shows how the event can be incorporated.

Demonstration

One demonstration of **select-edit-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) displays two independent **select-edit-comp** components along with buttons which when clicked will make changes to the selection list. Try typing in new values to the selection or delete a selected value

along with monitoring the console.log.

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

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

```
       <div class="buttons_box">
        <button v-on:click="add_honda">Add Honda</button>
        <button v-on:click="delete_straw">Delete Strawberry</button>
        <button v-on:click="set_fruit_value">Set Fruit to Orange'</button>
        <button v-on:click="set_fruit_null">Set Fruit to null'</button>
        <button v-on:click="set_new_fruits">Set New Fruit List</button>
      </div>
      <div class="selects_box">
        <select-edit-comp
          heading="My Favorite Car"
          placeholder="Select a car"
          select_value="Nissan"
          :items="car_items"
          :blur_panel="car_blur_panel"
          :single_border="single_border_car"
          :css_variables="css_variables_cars"
          v-on:select_edit_comp_value_changed="show_car">
        </select-edit-comp>
        <select-edit-comp
          heading="My Favorite Fruit"
          placeholder="Select a fruit"
          :items="fruit_items"
          :select_value="fruit_value"
          :show_trash="fruit_show_trash"
          :css_variables="css_variables"
          v-on:select_edit_comp_value_changed="show_fruit">
        </select-edit-comp>
      </div>
```

And the supporting data references:

```
data() {
   return {
     car_value: null,
     fruit_value: null,
     car_items: ['Ford','Chevy','Toyota','Nissan','Mazda'],
     fruit_items: ['Apple','Orange','Blueberry','Strawberry','Blackberry'],
     single_border_car: true,
     car_blur_panel: false,
     fruit_show_trash: false,
     css_variables_cars: {
       select_edit_items_panel_position: 'absolute',
       select_edit_items_panel_z_index: '100',
       select_edit_border_color: 'green',
       select_edit_placeholder_color: 'blue'
     },
     css_variables: {
       select_edit_border_color: 'green'
     }
```

```
    }
  },
  methods: {
    //respond to change in SelectEditComp car value
    show_car: function(value){
      this.car_value = value;
      console.log(`Favorite Car: ${value}`);
    },
    //respond to change in SelectEditComp fruit value
    show_fruit: function(value){
      this.fruit_value = value;
      console.log(`Favorite Fruit: ${value}`);
    },
    add_honda: function(){
      this.car_items.push('Honda');
    },
    delete_straw: function(){
      const idx = this.fruit_items.indexOf('Strawberry');
      this.fruit_items.splice(idx, 1);
    },
    set_fruit_value: function(){
      this.fruit_value = 'Orange';
    },
    set_fruit_null: function(){
      this.fruit_value = null;
    },
    set_new_fruits: function(){
      this.fruit_items = ['Mango','Pear','Raspberry','Pineapple'];
    }
  }
```