

button-comp

button-comp is a Vue.js(version >= 2.5) component that provides a method for CSS styling and responding to a click event.

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

button-comp has a single property for a parent to bind to:

`css_variables` -- defines some css variables for easy styling of **button-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 **button-comp**. The following list shows the css variable names along with their default values:

```
{
  button_comp_font_family: 'Verdana, serif',
  button_comp_font_size: '1rem',
  button_comp_background: 'linear-gradient(to bottom, #969696 0, #969696 13%, #5f5f5f 33%, #1e1e1e 64%, #1e1e1e 100%)',
  button_comp_color: 'rgba(255, 255, 255, 0.901961)',
  button_comp_box_shadow: '4px 4px 6px 1px rgba(0, 0, 0, 0.4), 2px 2px 2px 0 rgba(0, 0, 0, 0.2)',
  button_comp_hover_background: 'linear-gradient(to bottom, #969696 0, #1e1e1e 100%)'
}
```

As an example you could bind from the parent the following object to the `css_variables` prop for setting the font size and color of **button-comp**: `{button_comp_font_size: '24px', button_comp_color: 'purple'}`

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

Events

button-comp emits one event when clicked by sending a native MouseEvent event object :

```
this.$emit('button_comp_clicked', event)
```

The demonstration section below shows how a **button-comp** element can be configured to respond to this event.

Demonstration

A demonstration of **button-comp** is provided in the [ButtonComp](#) repository by hosting the `index.html` file under the `dist` folder. The demo (templated in the `App.vue` file) has three **button-comp** children that are styled differently and are assigned different `button_comp_clicked` event callbacks by the parent.

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

Below is example code of setting **button-comp** properties -- the demo's full template from `App.vue`:

```
<div class="demo_container">
<div class="buttons_box">
  <button-comp
    :css_variables="styled_css_variables"
    v-on:button_comp_clicked="change_font_weight">Change Font Weight
  </button-comp>

  <button-comp
    :css_variables="increase_css_variables"
    v-on:button_comp_clicked="increase_font_size">Increase Font Size
  </button-comp>

  <button-comp
    :css_variables="decrease_css_variables"
    v-on:button_comp_clicked="decrease_font_size">Decrease Font Size
  </button-comp>
</div>

<p :style="{fontSize: target_font_size + 'px', fontWeight:
target_font_weight}">{{target_text}}</p>
</div>
```

And a portion of the supporting data references:

```
data() {
  return
  ...
  ...
  target_text: 'we have no oranges today.',
  ...
  ...
  target_font_weight: 'normal',
  ...
  ...
  styled_css_variables: {
    button_comp_font_size: '1.2rem',
```

```

        button_comp_background: 'linear-gradient(to bottom, #a3ff33 0, #3cc72a
13%, #2eba1e 33%, #4bba37 64%, #4bba37 100%)',
        button_comp_hover_background: 'linear-gradient(to bottom, #a3ff33 0,
#4bba37 100%)'
    }
},
methods: {
    //functions that responds to 'button_comp_clicked' events
    change_font_weight: function(){
        console.log("Button 'font-weight' clicked");
        if(this.target_font_weight === 'normal') {
            this.target_font_weight = 'bold';
        }else {
            this.target_font_weight = 'normal';
        }
    },
    increase_font_size: function(){
        console.log("Button 'increase font-size' clicked");
        this.target_font_size += 12;
    },
    decrease_font_size: function(){
        console.log("Button 'decrease font-size' clicked");
        this.target_font_size -= 12;
    }
}
}

```