



# VUETIFY FUNDAMENTALS

## CHEAT SHEET

### GET STARTED

Install @vue/cli by running the following command:

```
yarn global add @vue/cli
```

Then create a new Vue application:

```
vue create vuetify --preset vuetifyjs/vue-  
cli-preset-vuetify
```

### DEFAULT APP MARKUP

```
<v-app>  
  <v-navigation-drawer app>  
    ...  
  </v-navigation-drawer>  
  <v-app-bar app>...</v-app-bar>  
  <v-content>  
    <v-container fluid>  
      <!-- If using vue-router -->  
      <router-view />  
    </v-container>  
  </v-content>  
  <v-footer app>...</v-footer>  
</v-app>
```

### CHANGING FONT

Out of the box, Vuetify supports MDI, Font Awesome 4 & 5, and Material Icons. Change it by updating the icons option:

```
// src/plugins/vuetify.js  
  
export default new Vuetify({  
  icons: { iconfont: 'fa' }}  
})
```

### CHANGING TAGS

```
<v-toolbar tag="nav">...
```

sets html tag

```
<nav class="v-toolbar">...
```

### BASIC FORM

```
<v-form @submit.prevent="onSubmit">  
  <v-text-field  
    v-model="name"  
    label="Name"  
  />  
  <v-text-field  
    v-model="email"  
    label="Email"  
  />  
  <v-btn type="submit">Submit</v-btn>  
</v-form>
```

### PROGRAMMATIC BREAKPOINTS

When the window is resized, the \$vuetify object will update:

```
<v-btn :block="$vuetify.breakpoint.smAndDown">...
```

### MODIFYING SASS VARIABLES

The Vuetify CLI plugin will automatically search for a variables file and apply it during compilation:

```
// src/sass/variables.scss
```

```
$body-font-family: 'Work Sans', serif;  
$border-radius-root: 6px;  
$font-size-root: 14px;
```

### GRID SYSTEM

Vuetify utilizes a flexbox grid-system based off of Bootstrap 4.

```
<v-container full-width>  
  <v-row align="center" justify="center">  
    <v-col cols="12" md="6">  
      <!-- Content -->  
    </v-col>  
  </v-row>  
</v-container>
```

Want to learn more about Vuetify?

Checkout the official documentation at [vuetifyjs.com](https://vuetifyjs.com)





# VUETIFY FUNDAMENTALS CHEAT SHEET

## ACTIVATOR SLOTS

Binds listeners and accessibility attributes.

```
<template v-slot:activator="props">
  <v-btn
    v-bind="props.attrs"
    v-on="props.on" ← binds @click listener
  >
    Click me
  </v-btn>
</template>
```

## DYNAMIC THEME

The Vuetify theme can be modified at run-time:

```
mounted () {
  this.$vuetify.theme.light = {
    primary: '#1976D2',
    secondary: '#424242',
    accent: '#82B1FF',
    error: '#FF5252',
    info: '#2196F3',
    success: '#4CAF50',
    warning: '#FFC107'
  }
}
```

## INTERSECT DIRECTIVE

Under the hood, v-intersect uses the **Intersection Observer API** to call a method when an element enters or leaves the viewport:

```
<v-card v-intersect="onIntersect">...
```

Whenever <v-card> is observed, **onIntersect** will be invoked:

```
onIntersect: function (
  entries,
  observer,
  isIntersecting
) {
  // Logic → boolean value
}
```

## LAZY LOADING

Defers the rendering of default slot until visible:

```
<v-lazy min-height="200">...deferred content
```

## INPUT ACTIONS

```
<v-autocomplete
  v-model="val"
  @click:clear="onClear"
  @click:prepend="onPrependClick"
/>
```

If the bound listener is emitted, the callback is invoked:

```
methods: {
  onClear: function () {
    // Logic
  },
  onPrependClick: function () {
    // Logic
  }
}
```

## SMOOTH SCROLLING

The **goTo** method can scroll to a querySelector, element or Vue component:

```
<v-btn @click="$vuetify.goTo('#loc')">...
```

## ECOSYSTEM

Join and/or follow us on one of our social platforms:

### Github

View the repository @ <https://github.com/vuetifyjs>

### Discord Community

Join us on Discord @ <https://community.vuetifyjs.com>

### Twitter

Follow us on twitter @[@vuetifyjs](https://twitter.com/vuetifyjs)

### Reddit

Join us on Reddit @ <https://reddit.com/r/vuetifyjs>

Want to learn more about Vuetify?

Checkout the official documentation at [vuetifyjs.com](https://vuetifyjs.com)

