



How to read these codes snippets

Vue components contain in the same place `<script>`, `<template>` and `<style>` data contents. In order to distinguish them:

Cheatography

## Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

How to read these codes snippets (cont)

*Italic code is for script.* Regular code is for template.

**Cheatography**

## Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)



v-if / v-else-if / v-else Condition <section v-show='var == true'>show me</section>

Cheatography

## Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Directives (cont)

**v-show** Condition (no alternative)

```
<section v-if='var == true'>show me</section>
```

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Directives (cont)

**v-for** Loop (iterate dataStore elt) `<li v-for='elt in dataStoreElt'>...</li>`

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Directives (cont)



**v-on:event (@event)** Event listener `<button v-on:click='function'>Click!</button>`

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Directives (cont)

**v-bind:elt (:elt)** Bind `<li v-bind:href='dataStoreElt.prop'>...`  
`</li>`

**Cheatography**

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Directives (cont)

**v-model** Bind `<input type='text' v-model='dataStoreElt'`  
forms `>`

**Cheatography**

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Directives (cont)



Share data between parent/child components

Direct data sharing between 2 linked components is possible in 3 ways: **props** & **slot** for *parent* communicating to *child*, and **\$emit** for *child* communicating to *parent*.

Cheatography

## Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)







```
import Child from './url'  props : ['url'] or props : {url : 'url'}
```

```
components : {Child}
```

```
<Child url= '/contact ' >  <a :href= 'url' >
```

**Cheatography**

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)



Parent / Child /

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

slot (Parent > Child) (cont)

```
import Child from './url'
```

```
components : {Child}
```

```
<Child v-slot:slotName >anything to replace default content </Child> <button> <span>fixed data</span>  
<slot name='slotName'>default free content </slot>{{nl}} </button>
```

Slots are **free areas** that can be named to use more than one. Default content of slots set in *child component* can be replaced as needed.

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)





```
import Child from './url'                                methods : { eventPerso(){ this.$emit('event-perso',{msg : 'hello'}) }}
components : {Child}
data() { return{ msg : "" } }
methods : { setMsg(event) { this.msg = event.msg } }
<Child @event-perso = 'setMsg' />                        <button @click = 'eventPerso'>
```

Cheatography

## Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)





**Install:** `vue add router` // *App.vue will be erased!*

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Router (cont)

```
route = { path: '/', name: 'Name', component: ImportedName OR () => import('../path/Component') }
```

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Router (cont)

`<router-view />` - Display  
point

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Router (cont)

`<router-link to='/path'></router-link>` - Link using routes (no reload page)

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)



`beforeCreate()/created() beforeMount()/mounted() beforeDestroy()/destroyed()`

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Lifecycle (cont)

Function used in `export default {...}` to define the moment some operations must be initiated.

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)





**State** (dataStore) Paths: **\$store.state**  
pluggin: { **mapState** } - use spread operator  
...

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Vuex - General dataStore (cont)

Pluggin must be used in **computed**

\* **objects** can be used in place of **strings** to rename variables

**Vuex store/ state:** `{key: 'value'}`

**Component/ Import** `{ mapState }` from 'vuex'

`computed: { ...mapState(['var1','var2']) }` \*

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Vuex - General dataStore (cont)

**Getters** (computed) Path: `$store.getters`  
pluggin: { **mapGetters** } - use spread operator ...

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Vuex - General dataStore (cont)

Pluggin must be used in **computed** **Vuex store/ getters:** `{ setDate(state) { return `${state.day}:${state.month}:${state.year} }`  
**Component/**  
**direct/ computed :** `{ date() {this.$store.getters('setDate')}} }`  
**pluggin/ import { mapGetters } from 'vuex'**  
`computed : {...mapGetters(['setDate']) }`

Cheatography

## Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Vuex - General dataStore (cont)

**Mutations** (change State) Path: - no access from components -  
Pluggin: - none -

**Cheatography**

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Vuex - General dataStore (cont)

Isolated from components **Vuex store/ state:** `{ count: 0 }`  
CAPITALIZED\_NAME **mutations:** `{ INCREASE_COUNT(state, amount) {state.count += Number(amount)} }`  
args: `state, payload`

Cheatography

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Vuex - General dataStore (cont)

**Actions** (commit changes to State)

Path: `$store.dispatch`

Pluggin: `{ mapActions }` - use spread operator ...

**Cheatography**

Vue.js Cheat Sheet

by Peanuts-83 via [cheatography.com/140542/cs/29858/](https://cheatography.com/140542/cs/29858/)

Vuex - General dataStore (cont)

Pluggin must be used in **methods** **Vuex store/ actions:** `{ updateCount(context,amount) {context.commit('INCREASE_COUNT', amount)} }`  
Apply mutators to State **Component/**  
**args:** `context, payload` **direct/ methods:** `{ update() {this.$store.dispatch('updateCount')} }`  
**pluggin/ import** `{ mapActions } from 'vuex'`  
**methods:** `{ ...mapActions(['updateCount']) }`

All data contained in this dataStore is accessible and changeable to any component.



By **Peanuts-83**  
[cheatography.com/peanuts-83/](https://cheatography.com/peanuts-83/)

Published 19th November, 2021.  
Last updated 20th November, 2021.  
Page 39 of 100.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>