

# VUE 3 MIGRATION CHANGES: THE INCOMPATIBLE

#### Named and Scoped Slot

### </script>

</template>

export default {

props: ['text']

<script>

Four learning paths to expand your Vue skills.

Find yours here vuemastery.com/courses

Need to remove

this attribute.

#### Precedence Between *v-if* and *v-for*

COMPILER\_V\_IF\_V\_FOR\_PRECEDENCE

#### v-if Keys

COMPILER\_V\_IF\_SAME\_KEY

```
    <!i v-for="num in nums">
        <span v-if="num < 10" :key="myKey">{{ num }}</span>
        <span v-else class="high" :key="myKey">{{ num }}</span>

Now you can't use the same key for branches in the same conditional, either remove them all or use different keys.
```

**Functional Attribute** 

COMPILER SFC FUNCTIONAL

<template functional> <h1>{{ text }}</h1>

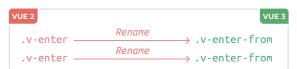
Mounted Container GLOBAL\_MOUNT\_CONTAINER

#### <template> key

COMPILER\_V\_FOR\_TEMPLATE\_KEY\_PLACEMENT

#### Transition Classes

(This deprecation has no warnings in the migration build.)





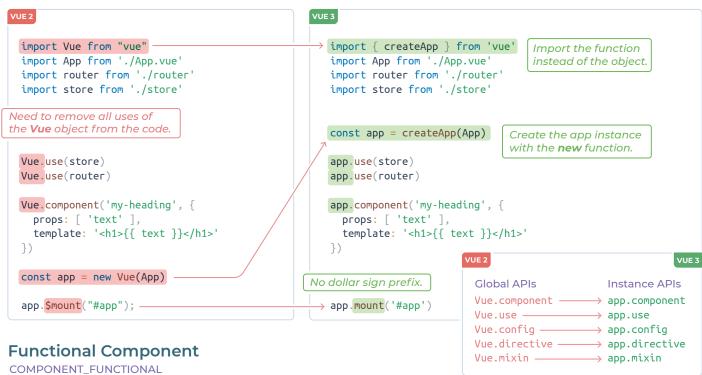
Looking for more information? We have a blog series that you can dive into now.



### **VUE 3 MIGRATION CHANGES:** THE REPLACED

#### App Initialization (main.js)

GLOBAL\_MOUNT / GLOBAL\_EXTEND / GLOBAL\_PROTOTYPE / GLOBAL\_OBSERVABLE / GLOBAL\_PRIVATE\_UTIL



```
VUE 2
 export default {
   functional: true,
   props: ['text'],
   render(h, { props }) {
     return h('h1', {}, props.text)
```

```
VUE 3
  import { h } from 'vue'
  const Heading = (props) => {
     return h('h1', {}, props.text)
                                 Need to use a function
  Heading.props = ['text']
                                 to create a functional
                                 component.
  export default Heading
```

#### **Async Component**

COMPONENT\_ASYNC

```
import { defineAsyncComponent } from 'vue'
const MyComponent = defineAsyncComponent({
 loader: () => import('./MyComponent.vue'),
})
        Import and use the new function.
```



Our blog series will show you a workaround for this. Find it here vuemastery.com/blog/migration

#### v-for References

V FOR REF

```
<template>
 <111>
  Now this way of getting the references
             won't work anymore.
</template>
mounted () {
 console.log(this.$refs.myNodes)
       Workaround
```



# VUE 3 MIGRATION CHANGES: THE RENAMED

#### Lifecycle Hooks

OPTIONS\_BEFORE\_DESTROY / OPTIONS\_BEFORE\_DESTROY

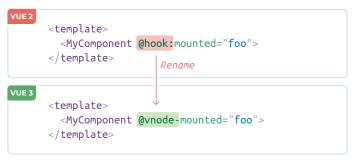
#### **Custom Directive Hooks**

CUSTOM\_DIR

```
VUE 2
                                                            VUE 3
  Vue.directive('my-directive', {
                                                                app.directive('my-directive', {
    created() {
                                                                  created() {
                                                Rename
    bind() { -
                                                                  beforeMount() {
                                                Rename
    inserted() { --
                                                                  mounted() {
    update() { Removed
                                                                  beforeUpdate() { NEW
                                                Rename
    componentUpdated() { —
                                                                  updated() {
                                                                  beforeUnmount() { NEW
                                                Rename
    unbind() { ... } —
                                                                 unmounted() { ... }
  })
                                                                })
```

#### **Lifecycle Events**

INSTANCE\_EVENT\_HOOKS



### v-model Prop and Event

COMPONENT\_V\_MODEL



# VUE 3 MIGRATION CHANGES: THE REMOVED

#### **Keyboard Codes Reactive Property Setters** GLOBAL\_SET / GLOBAL\_DELETE / INSTANCE\_SET / INSTANCE\_SET V\_ON\_KEYCODE\_MODIFIER Can't use keycode anymore. VUE 2 Vue.set -- Removed Vue.delete — <input type="text" v-on:keyup.112="validateText" /> vm.\$set -Removed vm.\$delete — Removed Use the key name instead. VUE 3 Workaround <input type="text" v-on:keyup.enter="validateText" /> **Child Component References** INSTANCE CHILDREN v-bind sync modifier COMPILER\_V\_BIND\_SYNC vm.\$children — Removed Can't use **sync** anymore. VUE 2 Workground <MyComponent v-bind:title.sync="myString" /> **Event Emitter** INSTANCE\_EVENT\_EMITTER Use **v-model** instead. VUE 3 vm.\$on — — Removed <MyComponent v-model:title="myString" /> vm.\$off — Removed vm.Sonce -Removed Workaround **Native Event** COMPILER\_V\_ON\_NATIVE Listeners INSTANCE\_LISTENERS Need to remove the **native** modifier. <SpecialButton v-on:click.native="foo" /> vm.\$listeners -- Removed Workaround Workarounds for these can be found at vuemastery.com/blog/migration

#### **Filters**

**FILTERS** 

```
<template>
                                                          <template>
 {{ num | roundDown }}
                                                            {{ numRoundedDown }}
</template>
                                                          </template>
              Filters won't work anymore.
                                                                        Use computed properties instead.
filters: { —
                                                          computed: {
 roundDown(value) {
                                                            numRoundedDown() {
   return Math.floor(value)
                                                              return Math.floor(this.num)
 }
},
                                                          },
```

