

VUE 3 MIGRATION CHANGES: THE INCOMPATIBLE

Named and Scoped Slot

VUE 2

```
<ItemList>
  <template slot="heading" slot-scope="slotProps">
    <h1>My Heading for {{ slotProps.items.length }} items</h1>
  </template>
</ItemList>
```

VUE 3

```
<ItemList>
  <template v-slot:heading="slotProps">
    <h1>My Heading for {{ slotProps.items.length }} items</h1>
  </template>
</ItemList>
```

Functional Attribute

COMPILER_SFC_FUNCTIONAL

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

```
<script>
export default {
  props: ['text']
}
</script>
```

Need to remove this attribute.



Four learning paths to expand your Vue skills.
Find yours here vuemastery.com/courses

Precedence Between v-if and v-for

COMPILER_V_IF_V_FOR_PRECEDENCE

VUE 2

```
<ul>
  <li v-for="num in nums" v-if="num < 10">{{ num }}</li>
</ul>
```

This used to run first.

VUE 3

```
<ul>
  <li v-for="num in nums" v-if="num < 10">{{ num }}</li>
</ul>
```

Now, this runs first.

SOLUTION

```
<ul>
  <li v-for="num in numsLower10">{{ num }}</li>
</ul>
```

Use computed property.

Mounted Container

GLOBAL_MOUNT_CONTAINER

```
<html lang>
  <head>...</head>
  <body>
    <noscript></noscript>
    <div id="app" data-v-app>
      <div id="app" data-v-7ba>
        ...
      </div>
    </div>
    ...
```

Now the mounted container will still be there, so you might get two container elements in the rendered HTML.

<template> key

COMPILER_V_FOR_TEMPLATE_KEY_PLACEMENT

```
<template v-for="num in nums" :key="num.id">
  <div :key="num.id">{{ num }}</div>
</template>
```

Need to move it to <template>

v-if Keys

COMPILER_V_IF_SAME_KEY

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

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

Transition Classes

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

VUE 2

```
.v-enter — Rename —> .v-enter-from
.v-enter — Rename —> .v-enter-from
```

VUE 3


Looking for more information? We have a blog series that you can dive into now.
Visit vuemastery.com/blog/migration

VUE 3 MIGRATION CHANGES: THE RENAMED

Lifecycle Hooks

OPTIONS_BEFORE_DESTROY / OPTIONS_BEFORE_DESTROY

<div style="background-color: #f8d7da; padding: 5px; margin-bottom: 5px; font-weight: bold; font-size: 0.8em;">VUE 2</div> <pre> export default { name: 'MyComponent', beforeDestroy() { ... }, destroyed() { ... }, ... } </pre>	<i>Rename</i> → <i>Rename</i>	<div style="background-color: #d4edda; padding: 5px; margin-bottom: 5px; font-weight: bold; font-size: 0.8em;">VUE 3</div> <pre> export default { name: 'MyComponent', beforeUnmount() { ... }, unmounted() { ... }, ... } </pre>
---	-------------------------------------	---

Custom Directive Hooks

CUSTOM_DIR

<div style="background-color: #f8d7da; padding: 5px; margin-bottom: 5px; font-weight: bold; font-size: 0.8em;">VUE 2</div> <pre> Vue.directive('my-directive', { created() { ... }, bind() { ... }, inserted() { ... }, update() { <i>Removed</i> }, componentUpdated() { ... }, unbind() { ... } }) </pre>	<i>Rename</i> → <i>Rename</i> <i>Rename</i> <i>Rename</i> <i>Rename</i>	<div style="background-color: #d4edda; padding: 5px; margin-bottom: 5px; font-weight: bold; font-size: 0.8em;">VUE 3</div> <pre> app.directive('my-directive', { created() { ... }, beforeMount() { ... }, mounted() { ... }, beforeUpdate() { <i>NEW</i> }, updated() { ... }, beforeUnmount() { <i>NEW</i> }, unmounted() { ... } }) </pre>
---	--	---

Lifecycle Events

INSTANCE_EVENT_HOOKS

<div style="background-color: #f8d7da; padding: 5px; margin-bottom: 5px; font-weight: bold; font-size: 0.8em;">VUE 2</div> <pre> <template> <MyComponent @hook:mounted="foo"> </template> </pre>	<i>Rename</i> ↓	<div style="background-color: #d4edda; padding: 5px; margin-bottom: 5px; font-weight: bold; font-size: 0.8em;">VUE 3</div> <pre> <template> <MyComponent @vnode-mounted="foo"> </template> </pre>
--	--------------------	---

v-model Prop and Event

COMPONENT_V_MODEL

<div style="background-color: #f8d7da; padding: 5px; margin-bottom: 5px; font-weight: bold; font-size: 0.8em;">VUE 2</div> <pre> Prop: value </pre>	<i>Rename</i> →	<div style="background-color: #d4edda; padding: 5px; margin-bottom: 5px; font-weight: bold; font-size: 0.8em;">VUE 3</div> <pre> modelValue </pre>
<pre> Event: input </pre>	<i>Rename</i> →	<pre> update:modelValue </pre>

VUE 3 MIGRATION CHANGES: THE REMOVED

Keyboard Codes

V_ON_KEYCODE_MODIFIER

VUE 2 Can't use keycode anymore.

```
<input type="text" v-on:keyup.112="validateText" />
```

VUE 3 Use the key name instead.

```
<input type="text" v-on:keyup.enter="validateText" />
```

v-bind sync modifier

COMPILER_V_BIND_SYNC

VUE 2 Can't use **sync** anymore.

```
<MyComponent v-bind:title.sync="myString" />
```

VUE 3 Use **v-model** instead.

```
<MyComponent v-model:title="myString" />
```

Native Event

COMPILER_V_ON_NATIVE

Need to remove the **native** modifier.

```
<SpecialButton v-on:click.native="foo" />
```

Filters

FILTERS

VUE 2 Filters won't work anymore.

```
<template>
  <p>{{ num | roundDown }}</p>
</template>
```

```
...
filters: {
  roundDown(value) {
    return Math.floor(value)
  }
},
```

Reactive Property Setters

GLOBAL_SET / GLOBAL_DELETE / INSTANCE_SET / INSTANCE_SET

Removed

- Vue.set
- Vue.delete
- vm.\$set
- vm.\$delete

Workaround

Child Component References

INSTANCE_CHILDREN

Removed

- vm.\$children

Workaround

Event Emitter

INSTANCE_EVENT_EMITTER

Removed

- vm.\$on
- vm.\$off
- vm.\$once

Workaround

Listeners

INSTANCE_LISTENERS

Removed

- vm.\$listeners

Workaround

Workarounds for these can be found at vuemastery.com/blog/migration