
# 项目升级 Vue3，避坑指南

## 概述

Vue3.0 正式版，在2020年9月18号正式发布，到现在已经快有一年的时间了，Vue3 的周边生态已经基本完善，很多新项目也已经开始使用 Vue3 进行生产环境开发了；目前这个节点上，正处于 Vue2 与 Vue3 共同存在的节点；不管你是刚入门学习，还是新项目技术选型，抑或是已有项目的升级迭代，都会面临一些问题；

更有很多小伙伴觉得技术的更新迭代太快，根本不知道怎么办？

今天就借着 Vue3 升级这个事，就来说说这些问题，希望能够给大家带来一些思考和启发；

在中大型项目，技术架构，非必要，不变更，是一个非常重要的原则；如果变更的必要性非常大，那我们需要怎么做？我们就以 Vue 框架为例，Vue2 升级到了 Vue3 ，如果你是刚接触Vue 框架，正在学习阶段，强烈建议直接学习 Vue3 即可，这也就表明了一种态度，Vue2 肯定会被淘汰，Vue3 才是未来，但是，未来还没来啊，绝大部分Vue项目依然是使用Vue2 这个版本的。

如果你已经是 Vue 框架的使用者了，那就有可能面临两种情况，第一个就是新项目刚开始，选择比较熟悉的 Vue2 还是比较新的 Vue3 ？ 第二个就是你的项目已经使用 Vue 开发运行了很长时间了，要不要替换升级到 Vue3 ？

这些问题，我最后再给你解答，因为无论如何，你只要是前端开发者，现在都应该学习 Vue3 了；

我们针对的还是有一些 Vue 编程能力的小伙伴 ，因此 Vue 的基本用法，我这里就不说太多了，那么，接下来的内容，我会分为几个部分：

- 以前能用，现在不能用的；
- 原来没有，Vue3 新增的；
- 原有功能改善提升

官方升级指南：https://v3.cn.vuejs.org/guide/migration/introduction.html

# 非兼容性更新或移除

**v-for 中的 Ref 数组 - 不兼容**

**attribute 强制行为 - 不兼容**

**$attrs 包含 class & style - 不兼容**

**$children - 移除**

**自定义指令 - 不兼容**

**自定义元素交互 - 不兼容**

*** Data 选项 - 不兼容：data 只能是一个返回对象的函数，根实例上也是一样的；**

**事件 API - 不兼容**

*** 过滤器 - 移除： 建议用方法调用或计算属性替换**

**函数式组件 - 不兼容**

*** 全局 API - 不兼容：**

一个新的全局 API：`createApp`

当前 Vue2 全局 API 及其相应实例 API 的表：

| 2.x 全局 API | 3.x 实例 API ( `app` ) |
| --- | --- |
| Vue.config | app.config |
| Vue.config.productionTip | *removed* (见下方) |
| Vue.config.ignoredElements | app.config.compilerOptions.isCustomElement (见下方) |
| Vue.component | app.component |
| Vue.directive | app.directive |
| Vue.mixin | app.mixin |
| Vue.use | app.use (见下方) |
| Vue.prototype | app.config.globalProperties (见下方) |
| Vue.extend | *removed* (see below) |

**全局 API Treeshaking - 不兼容**

**内联模板 Attribute - 不兼容**

### key attribute - 不兼容：

- **新增**：对于 `v-if` / `v-else` / `v-else-if` 的各分支项 `key` 将不再是必须的，因为现在 Vue 会自动生成唯一的 `key`.
  - **非兼容**：如果你手动提供 `key`，那么每个分支必须使用唯一的 `key`。你不能通过故意使用相同的 `key` 来强制重用分支。
- **非兼容**：`<template v-for>` 的 `key` 应该设置在 `<template>` 标签上 (而不是设置在它的子节点上)。

### 按键修饰符 - 不兼容：

- **非兼容**：不再支持使用数字 (即键码) 作为 `v-on` 修饰符
- **非兼容**：不再支持 `config.keyCodes`

### 移除 $listeners - 不兼容

### 挂载 API 变化 - 不兼容

### propsData - 移除

### 在 prop 的默认函数中访问 this - 不兼容：

生成 prop 默认值的工厂函数不再能访问 `this`。

### 渲染函数 API - 不兼容

### 插槽统一 - 不兼容

过渡的 class 名更改 - 不兼容

Transition 作为 Root - 不兼容

使用一个 `<transition>` 作为一个组件的根结点，当组件从外部被切换时将不再触发过渡效果。

### Transition Group 根元素 - 不兼容

`<transition-group>` 不再默认渲染根元素，但仍然可以用 `tag` prop 创建根元素。

### v-on.native 修饰符 - 移除

### * v-model - 不兼容

### v-if 与 v-for 的优先级对比 - 不兼容

2.x 版本中在一个元素上同时使用 `v-if` 和 `v-for` 时，`v-for` 会优先作用。Vue3 中 `v-if` 会拥有比 `v-for` 更高的优先级。

### v-bind 合并行为 - 不兼容

### VNode 生命周期事件 -不兼容

**Watch on Arrays - 不兼容**

## 新增功能

**异步组件**

**emits 选项**

*** 片段**

支持多根节点的组件，也就是片段！

**Suspense - 实验性**

**响应式系统的更新 - 无感知**

框架由 Object.defineProperty 的数据劫持，改为 Proxy 重写实现，向下兼容，解决数组下标的响应式问题；

（详见《Vue : Vue响应式原理》）

### 组合式 API （ Composition API ）

基本用法，reactive 响应式数据  setup 中 return 后，直接在模板中用；

setup 与 Options API 共同使用,不冲突

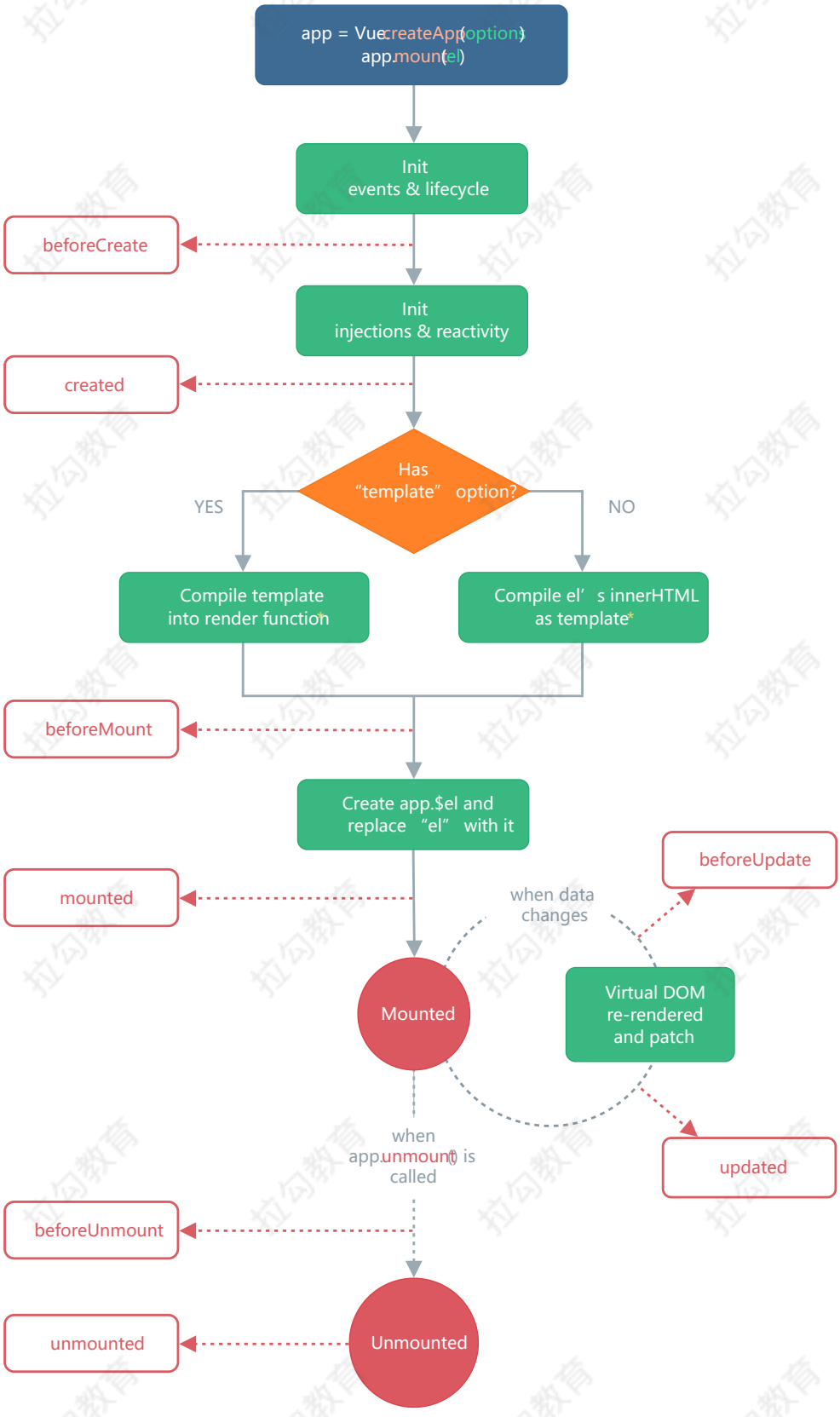
官方警告：

https://v3.cn.vuejs.org/guide/composition-api-introduction.html#%E7%BB%84%E5%90%88%E5%BC%8F-api-%E5%9F%BA%E7%A1%80

由于在执行 setup 时，组件实例尚未被创建，因此在 setup 选项中没有 this。

这意味着，除了 props 之外，你将无法访问组件中声明的任何属性——本地状态、计算属性或方法。

其实也存在另一个问题，看生命周期：




* Template compilation is performed ahead-of-time if using a build step, e.g., with single-file components.

问，setup 的执行时机在哪里?

很多文章都说 setup 是在 `beforeCreate` 和 `created` 之间， 这个结论是错误的。

setup 执行时机是在 beforeCreate 之前执行，也就是 init 初始化时；

setup 的 声明周期钩子 https://v3.cn.vuejs.org/guide/composition-api-lifecycle-hooks.html

需要引入，

下表包含如何在 setup () 内部调用生命周期钩子：

| 选项式 API | Hook inside `setup` |
| --- | --- |
| `beforeCreate` | Not needed* |
| `created` | Not needed* |
| `beforeMount` | `onBeforeMount` |
| `mounted` | `onMounted` |
| `beforeUpdate` | `onBeforeUpdate` |
| `updated` | `onUpdated` |
| `beforeUnmount` | `onBeforeUnmount` |
| `unmounted` | `onUnmounted` |
| `errorCaptured` | `onErrorCaptured` |
| `renderTracked` | `onRenderTracked` |
| `renderTriggered` | `onRenderTriggered` |

# 周边生态

## 使用全新构建工具 vite

vite2.0 使用了浏览器原生 ESM 模块化标准的前端构建工具

Vite 官方中文

（详见《前端工程化：Vite2.0 入门与原理》）

## UI 组件库

Ant Design-v2：https://2x.antdv.com/components/overview/

Vant-v3：https://vant-contrib.gitee.io/vant/v3/#/zh-CN/home

Element-Plus：https://element-plus.org/#/zh-CN

Naive UI: 一个 Vue 3 组件库：https://www.naiveui.com/zh-CN/os-theme

# 项目升级评估标准

项目量级 、团队能力、不兼容代码评估、第三方扩展评估、综合成本

内容概述：

升级指南：Vue2 如何平滑过渡到 Vue3
项目诊断：Vue 项目需要升级的 5 条评估标准
新旧对比：升级 Compostion Api 和全新组件库
构建工具：再见 Vue-cli 拥抱下一代工具 Vite


> 作者: 西岭老湿
>
> 欢迎在 知乎、微信公众号、B站 搜索关注 **西岭老湿**