

本文由 [简悦 SimpRead](#) 转码，原文地址 kaiwu.lagou.com

在组件实现的章节，我们分析了组件生成到页面 DOM 会经历创建 vnode、渲染 vnode 到 DOM 的过程。但其实我们编写组件时，并不会直接去手写组件 vnode，其中创建 vnode 的过程，实际上是 Vue.js 内部帮我们完成的。

我们知道在组件的渲染过程中，会通过 renderComponentRoot 方法渲染子树 vnode，然后再把子树 vnode patch 生成 DOM。renderComponentRoot 内部主要通过执行组件实例的 render 函数，创建生成子树 vnode。

而我们最常见的开发组件的方式就是编写 template 模板去描述组件的 DOM 结构，很少直接去编写组件的 render 函数，那么 Vue.js 内部就需要把 template 编译生成 render 函数，这就是 Vue.js 的编译过程。

组件 template 的编译过程，可以离线完成，也可以运行时完成，在前面的章节我们已经介绍过了。Vue.js 3.0 为了运行时的性能优化，在编译阶段也是下了不少功夫，所以我们这一模块的学习目标主要就两点：**了解编译过程以及背后的优化思想。**

由于编译过程平时开发中很难接触到，所以不需要你对每一个细节都了解，你只要对整体有一个理解和掌握即可。另外，后续我们在分析 Vue.js 的一些特性时，也会结合编译过程一起分析，也会经常回顾编译的过程和结果，帮你加深印象。

最后，在学习这章节内容的过程中，希望你可以使用官方的一个[模板导出工具](#)，在线调试模板的实时编译结果，辅助学习。如果你想在线调试编译的过程，可以在 vue-next 的源码 packages/template-explorer/dist/template-explorer.global.js 中的关键流程上打 debugger 断点，然后在根目录下运行 npm run dev-compiler 命令，接着访问 <http://localhost:5000/packages/template-explorer> 调试即可。