结束语

更新时间：2019-07-31 09:49:42

> 宝剑锋从磨砺出，梅花香自苦寒来。
>
> ——佚名

TypeScript 的基础知识，以及使用 TypeScript 开发 Vue 应用的实战内容，我们到这里就已经学习完了。相信你对 TypeScript 已经有了一个系统、全面的了解了。最后，Lison 最后再跟你絮叨几句。

7.1.1 使用TypeScript心得

如果你过去使用 React 开发 web 应用，在不使用 TypeScript 的情况下，编辑器对代码的提示并不是很好，但是如果使用 TypeScript，因为在编写代码阶段，数据的结构都是很明确的，所以编辑器会进行很友好的代码提示。这样我们在使用一些自己封装的方法，或者第三方对 TypeScript 支持良好的一些插件的时候，能减少很多去翻文档翻 api 介绍的时间。使用 TypeScript 使得代码的可读性大大，比如对于一些数值类型的常量，我们可以使用枚举值定义，这样就可以只用枚举成员代替数值字面量，提高可读性。对于函数的定义，在定义函数的时候定义完整的函数类型，包括参数类型和返回值类型等，在调用的时候，通过编辑器的代码提示，就可以看到需要几个参数，每个参数有什么要求；对函数的返回值进行操作的时候，能够知道操作的是什么类型值，可以做哪些操作。

如果你使用 Vue 开发 web 应用，也是可以使用 TypeScript 的，只不过，vue 是通过补充声明文件的方式，弥补了类型声明，但是并不是很完善，所以我们只能期待 Vue3.0 的到来了。但是如果使用TypeScript 开发 Vue 应用，要获得更好的类型支持，要在 script 里，通过类形式定义组件，这样编译器可以尽可能地推断一些类型。虽然 Vue2.x 对 TypeScript 的支持并不很好，但是如果你使用VSCode 进行开发，即使是不使用 TypeScript，也能得到一些代码提示。

如果你对 TypeScript 掌握的差不多了，想练手的话，可以自己先创建一个小项目来练练手。如果是实际项目开发，小型的个人完成的项目并不推荐使用 TypeScript，因为项目不复杂，内容不是很多的情况下，使用 TypeScript 会增加你的工作量，但带来的收益又不是很大。但如果是大型的多人协作的项目，或者是需要多处复用的开源或内部使用的插件库，极力推荐使用 TypeScript 进行开发，因为他可以帮助你提高代码质量，方便使用者快速上手，减少翻阅文档的次数，再配合单元测试，可以说这套代码就非常可靠了。

7.1.2 TypeScript前景

过去，在 ES6 标准颁布之前，社区有众多语法糖工具，比如当初的 CoffeeScript，可以说还是受到很多人追捧的。但是 ES6 标准公布之后，随着 Babel 等工具对 ES6 标准的支持，再加上 ES6 前卫的语法标准，大家不再需要 CoffeeScript，使用 Babel 即可将 ES6 代码转义为兼容新旧浏览器的代码。

虽然 TypeScript 在短期内不会被取代，但是随着一些新语言新标准的提出，可能 TypeScript 编译为JavaScript 这种方式弥补弱类型的 JavaScript 的思路不会淘汰，但 JavaScript 会不会被取代，这都是难说的。但是学习 TypeScript，在现阶段来看，是很有意义的。

学习完本专栏后，你应该掌握了 TypeScript 的几乎所有知识点，包括语法知识、项目配置以及在使用 TypeScript 进行 Vue 项目的开发。前面没有讲解的 JSX，我们通过实战部分进行了讲解。我们使用的是 Vue2 版本，但是 Vue2 并不是使用 TypeScript 开发的，后来只是通过补充声明文件等方式补充了对部分api的TypeScript的支持。在实际开发中，并不是很友好，所以这也是 TypeScript 还差一部分用户的原因。

但是我们知道，Vue 在我们写本专栏的时候，正在使用 TypeScript 对源码进行彻底重写，也就是即将发布的 Vue3.0 版本。相信随着 Vue3.0 的发布，我们就可以使用 TypeScript 愉快地开发 Vue 应用了。

专栏的内容很多，看一遍肯定消化不了，遇到不会的，再来翻一翻，相信在实践中，你会对学过的TypeScript 知识有越来越清晰的了解，快去实践中尝试吧。

使用TypeScript开发Vue组件和使用Vue组件