

Vue 作者尤雨溪在一次演讲中说道：框架的设计过程其实是一个不断取舍的过程。

这代表的是什么意思呢？

想要搞明白这个，那么再来明确一下之前说过的概念：

1. 命令式的性能 > 声明式的性能
2. 命令式的可维护性 < 声明式的可维护性
3. 声明式的框架本质上是由命令式的代码去实现的
4. 企业项目开发时，大多使用声明式框架

当我们明确好了这样的一个问題之后，那么我们接下来来思考一个问題：**框架的开发与设计原则是什么呢？**

我们知道对于 `Vue` 而言，当我们使用它的是通过 **声明式** 的方式进行使用，但是对于 `Vue` 内部而言，是通过 **命令式** 来进行的实现。

所以我们可以理解为：Vue 封装了命令式的逻辑，而对外暴露出了声明式的接口

那么既然如此，我们明知 **命令式的性能 > 声明式的性能**。那么 **Vue** 为什么还要选择声明式的方案呢？

其实原因非常的简单，那就是因为：**命令式的可维护性 < 声明式的可维护性**。

为指定的 div 的子元素 div 的子元素 p 标签，展示变量 msg

以这个例子为例。

对于开发者而言，不需要关注实现过程，只需要关注最终的结果即可。

而对于 Vue 而言，他所需要的做的就是：封装命令式逻辑，同时 **尽可能的减少性能的损耗！** 它需要在 **性能与可维护性** 之间，找到一个平衡。从而找到一个 **可维护性更好，性能相对更优** 的一个点。

所以对于 Vue 而言，它的设计原则就是：在保证可维护性的基础上，尽可能的减少性能的损耗。

那么回到我们的标题：为什么说框架的设计过程其实是一个不断取舍的过程？

答案也就呼之欲出了，因为：

我们需要在可维护性和性能之间，找到一个平衡点。在保证可维护性的基础上，尽可能的减少性能的损耗。

所以框架的设计过程其实是一个不断在 **可维护性和性能** 之间进行取舍的过程

