

# 《开源软件设计与开发》课程总结

姓名：林志刚

学号：51195100036

## 1、开源理解

一、开源精神里最重要的，是拿来主义。

开源需要贡献、需要付出，但在贡献和付出之前，首要的是去了解、去使用已经存在的。

比如前端工程师，会经常使用模板，天天用着，会有不爽，就会想着自己去写一个模板引擎。直接去写一个没什么不对，但更好的方式是，先去开源社区搜索下同类产品。看自己的想法，是否已经有人实现了，看自己的想法中，哪些是靠谱的，哪些是不太靠谱的。

注意不要去做「纯调研」，除非你已经是这方面的专家，有过体系化的研究和深入。「纯调研」是指，找到类似的产品，浅尝辄止地跑跑 demo，就开始出表格，横向比较优劣。更笨但更有效地做法是，直接去用，比如我们选择了 Handlebars，在项目真正使用过程中，才能深触地感受到 Handlebars 的优点及其缺点。有过这么一段深入使用经验后，类似的前端模板引擎，比如 Mustache、Hogan 等，也有大概清楚个七七八八了。

拿来主义不光能省去开发成本，还能让你扩充见识和想法，而不会成为井底之蛙还茫然自得。

二、在开源世界里，参与比主导更重要。

遵循拿来主义，可以让你更懂得他人。在这过程中，会有欣赏、感谢，也会有抱怨、想法。

前端 DOM 操作类库中，很多公司包括支付宝都选择了 jQuery。使用过程中，jQuery 的一些 API 的设计非常赞，一些内部实现也非常漂亮。遇到这些让人欣赏的点时，别忘了将自己的感想写下来。那怕是一篇微博，或是跟朋友闲聊时的只言片语，所有这些，都是参与，都是正能量的传播。社区的赞誉，可以给 jQuery 开发团队最大的鼓励。很多坚持，是因为有肯定。

当然，jQuery 只要存在一天，就一定存在不足。比如 data 的 API 变化，比如动画不支持纯 CSS3 方式，还有 Sizzle 也并非那么完美。所有这些「缺点」，你遇到的 bug，你踩过的坑，当你有好的想法或解决方案时，不要忘记反馈给社区。jQuery 有详细的反馈渠道，一般成熟的开源社区都会有。比如托管在 GitHub 上的项目，一般都可以通过 Issues 给社区提交你的想法，甚至可以通过 Pull Request 直接提交你修改过的代码。当然，jQuery 只要存在一天，就一定存在不足。比如 data 的 API 变化，比如动画不支持纯 CSS3 方式，还有 Sizzle 也并非那么完美。所有这些「缺点」，你遇到的 bug，你踩过的坑，当你有好的想法或解决方案时，不要忘记反馈给社区。jQuery 有详细的反馈渠道，一般成熟的开源社区都会有。比如托管在 GitHub 上的项目，一般都可以通过 Issues 给社区提交你的想法，甚至可以通过 Pull Request 直接提交你修改过的代码。

拿来主义是开源的第一步，参与则是开源中最漫长的一步。开发自己主导的项目不是必须的，除非你真的有很好的想法，同时刚好有旺盛的精力，否则不要轻易开始，那不代表什么，往往什么也不是。比如 Sea.js，如果 RequireJS 当初能接受我的建议，或者 FlyScript 的作者不退隐江湖，那么就不会有 Sea.js。让一些事情自然地发生，让豆瓣从土壤中发芽。

更多地去享受参与，保持简单与自然。

三、开源的是社区，代码仅是很小的一部分。

虽然代码很重要，但在开源项目里，源码只是重要的因素之一。

除了代码，还有文档、测试用例、Issues 管理、版本发布、升级策略、书籍、视频等等。jQuery、Backbone、AngularJS 等成熟项目，成熟的是社区，而不是源码。好的社区，是在 Google 中一搜索，就能找到不少资料。好的社区，是你踩过的坑，经常已经有人也踩过并整理分享出来了。

没有社区的开源项目，其源码会一直停留在「孤芳自赏」阶段。几年前的 KISSY，还有百度的 Tangram，个人觉得都停留在初始阶段，发展很难。有社区的开源项目，其源码会活起来，会自我不断进化升华。我参与过的一个典型项目是 Backbone，倘若你现在再回过头去看 Backbone 最初提交的代码，与现在的相比，真心会觉得有点挫了。最让人不可思议的是，Backbone 最初很挫的代码，在社区的参与下，在不断的讨论和 Pull Request 中，现在已经演化成非常高质量的代码，甚至每一小段代码里，都能说出一个故事来。我现在依旧非常关注 Backbone 的每一次代码变化，特别是自己修改过的部分。从其他人的修改中，能收获很多很多。

## 2、开源贡献

很遗憾，参与得晚，没能给 TIDB 作贡献

## 3、课程反馈

还好，就是课程有点像讲座式，并没觉得有多少干货，但还是想赞一下庄教师

## 4、参考文献

没有