

总结

Git 内容非常多，这本小册我已经尽量克制，可是还是写了二十多节出来。尽管这样，有些很有用的内容我依然没有写出来。因为我写这本小册的目的是解决大部分人「学不会 Git」和「用了很久却总用不好 Git」这两个问题，所以我在这本小册里重点讲的也是 Git 的学习和使用中那些既重要又困难的关键点。

如果你在整个阅读过程中是边读边练的，相信读到这里，你对 Git 已经有一个较为全面和深刻的认识了。接下来你只要在平时使用 Git 的过程中多留心一些，找机会把这本小册中的内容应用在实战，很快就可以成为众人眼中的「Git 高手」了。当然，到时候你也许也会发现，其实大家眼中的「Git 高手」远没有那么神秘，并不一定比别人懂很多，只是更加了解 Git 的工作原理和一些关键概念罢了。

几个「不难但却很有用」的 Git 技能点

除了这本小册里讲到的那些「关键点」，还有些 Git 的相关知识虽然也比较有用，但属于稍微研究一下就可以学会的内容，我就不讲了，只在这里做一个简单的列举，你在平时使用 Git 的时候记得抽空学习一下就好。

tag：不可移动的 branch

tag 是一个和 branch 非常相似的概念，它和 branch 最大的区别是：tag 不能移动。所以在很多团队中，tag 被用来在关键版本处打标记用。

更多关于 tag：<https://git-scm.com/docs/git-tag>

cherry-pick: 把选中的 commits 一个个合并进来

cherry-pick 是一种特殊的合并操作，使用它可以点选一批 commits，按序合并。

更多关于 cherry-pick: <https://git-scm.com/docs/git-cherry-pick>

git config: Git 的设置

git config 可以对 Git 做出基础设置，例如用户名、用户邮箱，以及界面的展示形式。内容虽然多，但都不难，整体看一遍，把 Git 设置成你最舒服的样子，从此就再也不用管它了。属于「一次付出，终身受用」的高性价比内容。

更多关于 config: <https://git-scm.com/docs/git-config>

Git Flow: 复杂又高效的工作流

除了前面讲到的 "Feature Branching"，还有一个也很流行的工作流：Git Flow。Git Flow 的机制非常完善，很适合大型团队的代码管理。不过由于它的概念比较复杂（虽然难度并不高），所以并不适合新手直接学习，而更适合在不断的自我研究中逐渐熟悉，或者在团队合作中慢慢掌握。基于这个原因，我最终也没有在这本小册里讲 Git Flow，但我推荐你自己在有空的时候了解一下它。

更多关于 Git Flow:

<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

以上这几个内容都不难，而且挺有用，所以虽然我没有讲，但都建议你把它们了解一下，会有好处的。

想学习更多的 Git 知识？

如果看完这本小册觉得不够，希望得到更多的 Git 知识补充，可以到它的官网去查看详细的文档：

<https://git-scm.com/>

结语

这本小册《Git 原理详解及实用指南》就到这里，希望能帮到你！