

Git 教程

为什么要用Git

因为别人都在用。

Git是世界上最流行的版本控制系统，现在已经成为了新项目采用版本控制的事实标准。Git的功能及其强大，能够胜任各种不同的或简单或复杂的版本控制需求。在Git周围，已经建立起了包括GUI、IDE集成、云托管等生态。学会Git是一个优秀程序员所需的基本素养。

怎样学习Git

时间充足的话，建议先学习Git的命令行使用，因为命令行更接近Git的本质。然而你可能随时需要一个GUI工具来查看版本图。

网络上的教程汗牛充栋，但质量良莠不齐。选择一个好的教程，能够少走弯路。学习Git时，需要特别注意「正确」的操作方式，即「最佳实践」，否则，将来利用Git与他人协作时，你可能会不小心创建一些愚蠢的提交，最终，人民群众会仇恨你，你的朋友和家人也会嘲笑你，唾弃你。¹

基于上面的理由，我们推荐下面的Git教程。

Pro Git Book

网址：<https://git-scm.com/book/zh/v2>

这是最权威的Git官方教程，不仅可以在线浏览，同时可以下载PDF、EPUB、MOBI格式的电子书，方便装到iPad或Kindle中随时阅读。这个教程事无巨细地讲述了Git的方方面面，即使你已经会使用Git了，也可能需要时时回去翻阅这个教程。当然在初学的时候，可以抱着「会用就行」的心态，而不必在许多细节上纠缠。下面是每章简要的导读：

1. **起步**：简要看看就好，能够安装Git就行。
2. **Git 基础**：2.1到2.5节的内容是重中之重，务必认真掌握。
3. **Git 分支**：可以跳过3.4节「分支开发工作流」，因为这个不太实用。
4. **服务器上的 Git**：请直接跳过这章。
5. **分布式 Git**：5.1、5.2节介绍的是多人协作的流程。这些内容看起来会有些复杂，但在实际的小规模的代码合作中，并不是每个细节都需要得到遵守。另外，你的团队可能会要求你遵守一些额外的准则。
6. **GitHub**：不用太在意这一章，里面的一些概念，例如GitHub上的Pull Request，可以了解一下。

和所有的计算机技术一样，光学不做，是不可能真正掌握Git的。学习的时候请务必随时实际操作，遇到不懂的问题，在向学长请教之前，先自己做一点试验。这样有助于加深对Git本质的理解。

好了，当你已经粗略看完了教程的前半部分，并已经有了一些实际经验，对Git的工作原理有了一些基本认识之后，你可以回到第7和8章，学习一些使你的Git生活更加轻松的工具和命令。一般来说，同一章的各个小节，是按照重要程度排序的。同时，附录中的一些内容，可能也会对你有所帮助。

廖雪峰的Git教程

地址：<https://www.liaoxuefeng.com/wiki/896043488029600>

廖雪峰的各种技术教程非常有名，放在这里，作为官方教程的一个替代。笔者本人没有读过这个教程，无法提供导读，如果你选择这个教程，请自行参照上面 *Pro Git* 的导读，酌情学习。

Google 是你的朋友！

任何不明白的地方，或者你突然想到的「Git能做xxx吗」的问题，都可以借助Google求解。当然如果你真的没有办法突破互联网封锁，Bing、Yahoo! 等搜索引擎也是可以的。但是有一点需要特别强调：

永远，永远，永远不要使用百度²

如果Google也帮不了你，那说明你还没有能够高效地使用搜索引擎。你可以在明白「[提问的智慧](#)」的前提下，向学长或其他会Git的人提问。

Git托管平台

一个在初学者中常见的错误是把Git和GitHub给搞混淆。事实上，GitHub是一个「代码托管平台」，用于在云端存储的代码。这样的托管平台很多，就笔者所知，西工大足球机器人基地采用的托管平台至少有Gogs和GitLab。这些平台的基本使用方式是一致的。我们在校内搭建了一个Gitea托管平台 <http://10.69.42.139:3000/>，供大家使用学习。

彭冠文 2019-12-15

1. 这句话出自[Pro Git 中文版](#)

2. 珍爱生命，远离百度。这无论是对你的学习生活，还是对整个社会，都有好处。