

[Git 北京](#)

为谁而写？

对 Git 版本控制一无所知的，初入编程之门的新人。

内容简介

Git 版本控制图解教程，介绍版本控制的重要概念。本地版本控制协同 Github 远端操作。高级部分，介绍开辟新分支合并分支，解决 merge 冲突，分析常见错误。

第一章 课程路线图

[如何学习 Git Beijing](#)

第二章 搬进 Github

[浏览器中使用 Github](#)

本章也就先不碰 Git 的比較难的知识，而只是演示日常中最实用的版本控制工作流。具体来说，就是看 Github.com 网站分别和 Github 本地客户端，或者是本地命令行来协同。

通常老手都是用命令行的。

[浏览器中使用 Github](#)

Github 网页操作

- 注册
- 创建仓库
- 编辑项目
- 提交版本
- commit line

[本地安装 Git](#)

建议使用命令行操作， `git clone`

[创建本地仓库](#)

`git init` 初始化一个 git 仓库

[删除一个仓库](#)

[版本控制四步走](#)

`git status` 查看当前项目修改的状态 `git diff` 查看当前的修改，绿色表示添加，红色表示删除
`git add -A` 添加当前修改、添加、删除的文件到仓库中

[回滚历史](#)

[链接 github](#)

第三章：深入命令行

[添加 ssh](#)

[配置系统](#)

[~/.gitconfig 中的东东](#)

- `ci = commit -a -v`
- `throw = reset --hard HEAD`
- `throwh = reset --hard HEAD~`

第四章：搬进 Coding.net

[为何迁移到 Coding](#)

- 服务器在国内，速度快
- `private` 项目免费
- github 项目更多，社区更健康

[Coding Pages 实现网站托管](#)

[官方参考文档](#)

[Atom/git/coding 的三角关系](#)

第五章：分支专题

[简单分支操作](#)

[合并分支](#)

[Github 工作流](#)

Github 团队协作流程 [Github Flow](#), [中文版](#)

[贡献开源项目的流程](#)

[快速 PR](#)

第六章：更多 Github 技巧

[Issue 的使用](#)

[Github Pages 搭建免费网站](#)

[更多 Github 的秘密](#)

第七章：Goodbye

[更多学习建议](#)