

精通Git（第2版）



[精通Git（第2版）_下载链接1](#)

著者:Scott Chacon

出版者:人民邮电出版社

出版时间:2017-9

装帧:平装

isbn:9787115463067

Git
仅用了几年时间就一跃成为了几乎一统商业及开源领域的版本控制系统。本书全面介绍Git 进行版本管理的基础和进阶知识。全书共10章，内容由浅入深，展现了普通程序员和项目经理如何有效利用Git提高工作效率，掌

握分支概念，灵活地将Git用于服务器和分布式工作流，如何将开发项目迁移到Git，以及如何高效利用GitHub。

作者介绍:

作者简介:

Scott Chacon

GitHub联合创始人，曾任GitHub首席信息官，现任在线教育创业公司Chatterbug的首席执行官。在GitHub工作的7年里，他负责维护Git主页git-scm.com，并在许多技术大会上发表过演讲，以此传播Git文化和开源精神。GitHub ID: schacon。

Ben Straub

软件开发者，曾就职于GitHub，参与开发了Libgit2开发包以及GitHub for Windows客户端，另著有《GitHub实践》。GitHub ID: ben。

译者简介:

门佳

GNU/Linux深度用户，喜欢溯本求源，挖掘技术背后的来龙去脉，对程序语言设计理论、编译技术、操作系统设计与实现、Web开发等领域均有涉猎，译著包括《Linux Shell脚本攻略》《TCP Sockets编程》《精通JavaScript（第2版）》《Linux命令行与shell脚本编程大全（第3版）》等。

刘梓懿

ThoughtWorks高级软件工程师，亚马逊AWS认证架构师，曾为多个世界知名客户提供软件交付和咨询服务，在大型组织的云架构应用、持续交付、分层自动化测试体系建设和DevOps转型等方面拥有丰富的实践经验。GitHub ID: richardzone。

目录: 第1章 入门 1

1.1 关于版本控制 1

1.1.1 本地版本控制系统 1

1.1.2 集中式版本控制系统 2

1.1.3 分布式版本控制系统 3

1.2 Git简史 4

1.3 Git基础 4

1.3.1 快照，而非差异 4

1.3.2 几乎所有操作都在本地执行 5

1.3.3 Git的完整性 6

1.3.4 Git通常只增加数据 6

1.3.5 三种状态 7

1.4 命令行 8

1.5 安装Git 8

1.5.1 Linux上的安装方法 8

1.5.2 Mac上的安装方法 8

1.5.3 Windows上的安装方法 9

1.5.4 从源码安装 9

1.6 Git的首次配置	10
1.6.1 用户身份	11
1.6.2 个人编辑器	11
1.6.3 检查个人设置	12
1.7 获取帮助	12
1.8 小结	12
第2章 Git基础	13
2.1 获取Git仓库	13
2.1.1 在现有目录中初始化Git仓库	13
2.1.2 克隆现有仓库	14
2.2 在Git仓库中记录变更	14
2.2.1 查看当前文件状态	15
2.2.2 跟踪新文件	16
2.2.3 暂存已修改的文件	16
2.2.4 显示更简洁的状态信息	18
2.2.5 忽略文件	18
2.2.6 查看已暂存和未暂存的变更	19
2.2.7 提交变更	21
2.2.8 跳过暂存区	22
2.2.9 移除文件	23
2.2.10 移动文件	24
2.3 查看提交历史	25
2.4 撤销操作	30
2.4.1 撤销已暂存的文件	30
2.4.2 撤销对文件的修改	31
2.5 远程仓库的使用	32
2.5.1 显示远程仓库	32
2.5.2 添加远程仓库	33
2.5.3 从远程仓库获取和拉取数据	34
2.5.4 将数据推送到远程仓库	34
2.5.5 检查远程仓库	35
2.5.6 删除和重命名远程仓库	36
2.6 标记	36
2.6.1 列举标签	36
2.6.2 创建标签	37
2.6.3 注释标签	37
2.6.4 轻量标签	38
2.6.5 补加标签	38
2.6.6 共享标签	39
2.6.7 检出标签	39
2.7 Git别名	40
2.8 小结	41
第3章 Git分支机制	42
3.1 分支机制简述	42
3.1.1 创建新分支	44
3.1.2 切换分支	45
3.2 基本的分支与合并操作	48
3.2.1 基本的分支操作	48
3.2.2 基本的合并操作	52
3.2.3 基本的合并冲突处理	53
3.3 分支管理	55
3.4 与分支有关的工作流	56
3.4.1 长期分支	57
3.4.2 主题分支	58

- 3.5 远程分支 59
 - 3.5.1 推送 63
 - 3.5.2 跟踪分支 64
 - 3.5.3 拉取 66
 - 3.5.4 删除远程分支 66
- 3.6 变基 66
 - 3.6.1 基本的变基操作 66
 - 3.6.2 更有趣的变基操作 69
 - 3.6.3 变基操作的潜在危害 71
 - 3.6.4 只在需要的时候执行变基操作 74
 - 3.6.5 变基操作与合并操作的对比 75
- 3.7 小结 75
- 第4章 Git服务器 76
 - 4.1 协议 76
 - 4.1.1 本地协议 76
 - 4.1.2 HTTP协议 78
 - 4.1.3 SSH协议 79
 - 4.1.4 Git协议 80
 - 4.2 在服务器上搭建Git 80
 - 4.2.1 将裸仓库放置在服务器上 81
 - 4.2.2 小型团队配置 82
 - 4.3 生成个人的SSH公钥 83
 - 4.4 设置服务器 84
 - 4.5 Git守护进程 85
 - 4.6 智能HTTP 87
 - 4.7 GitWeb 88
 - 4.8 GitLab 90
 - 4.8.1 安装 90
 - 4.8.2 管理 91
 - 4.8.3 基本用法 93
 - 4.8.4 协作 93
 - 4.9 第三方托管选择 94
 - 4.10 小结 94
- 第5章 分布式Git 95
 - 5.1 分布式工作流 95
 - 5.1.1 集中式工作流 95
 - 5.1.2 集成管理者工作流 96
 - 5.1.3 司令官与副官工作流 97
 - 5.1.4 工作流小结 97
 - 5.2 为项目做贡献 98
 - 5.2.1 提交准则 98
 - 5.2.2 私有小型团队 100
 - 5.2.3 私有管理团队 105
 - 5.2.4 派生的公开项目 110
 - 5.2.5 通过电子邮件接受补丁的公开项目 113
 - 5.2.6 小结 115
 - 5.3 维护项目 115
 - 5.3.1 使用主题分支 115
 - 5.3.2 应用来自电子邮件的补丁 116
 - 5.3.3 检出远程分支 118
 - 5.3.4 确定引入内容 119
 - 5.3.5 整合所贡献的工作结果 120
 - 5.3.6 为发布版打标签 125
 - 5.3.7 生成构建编号 126

- 5.3.8 准备发布 126
- 5.3.9 简报 127
- 5.4 小结 127
- 第6章 GitHub 128
- 6.1 账号设置与配置 128
 - 6.1.1 SSH访问 129
 - 6.1.2 头像 130
 - 6.1.3 电子邮件地址 131
 - 6.1.4 双因素身份验证 132
- 6.2 为项目做贡献 132
 - 6.2.1 派生项目 132
 - 6.2.2 GitHub流程 133
 - 6.2.3 拉取请求的高级用法 140
 - 6.2.4 Markdown 144
- 6.3 项目维护 148
 - 6.3.1 创建新仓库 148
 - 6.3.2 添加协作人员 150
 - 6.3.3 管理拉取请求 150
 - 6.3.4 提醒和通知 155
 - 6.3.5 特殊文件 158
 - 6.3.6 项目管理 159
- 6.4 组织管理 160
 - 6.4.1 组织的基本操作 160
 - 6.4.2 团队 160
 - 6.4.3 审计日志 162
- 6.5 GitHub脚本化 162
 - 6.5.1 钩子系统 162
 - 6.5.2 GitHub API 166
- 6.6 小结 170
- 第7章 Git工具 171
- 7.1 选择修订版本 171
 - 7.1.1 单个修订版本 171
 - 7.1.2 提交范围 175
- 7.2 交互式暂存 177
 - 7.2.1 暂存和取消暂存文件 178
 - 7.2.2 暂存补丁 180
- 7.3 储藏与清理 181
 - 7.3.1 储藏工作成果 181
 - 7.3.2 灵活运用储藏 183
 - 7.3.3 从储藏中创建分支 184
 - 7.3.4 清理工作目录 184
- 7.4 签署工作 186
 - 7.4.1 GPG简介 186
 - 7.4.2 签署标签 186
 - 7.4.3 验证标签 187
 - 7.4.4 签署提交 187
 - 7.4.5 所有人都得签署 189
- 7.5 搜索 189
 - 7.5.1 git grep 189
 - 7.5.2 Git日志搜索 190
- 7.6 重写历史 192
 - 7.6.1 修改最近一次提交 192
 - 7.6.2 修改多个提交消息 192
 - 7.6.3 重排提交 194

- 7.6.4 压缩提交 195
- 7.6.5 拆分提交 195
- 7.6.6 超强命令：filter-branch 196
- 7.7 重置揭秘 197
 - 7.7.1 三棵树 198
 - 7.7.2 工作流 199
 - 7.7.3 重置的作用 203
 - 7.7.4 利用路径进行重置 205
 - 7.7.5 压缩 207
 - 7.7.6 检出 209
 - 7.7.7 小结 210
- 7.8 合并的高级用法 211
 - 7.8.1 合并冲突 211
 - 7.8.2 撤销合并 220
 - 7.8.3 其他类型的合并 222
- 7.9 rerere 225
- 7.10 使用Git调试 230
 - 7.10.1 文件标注 230
 - 7.10.2 二分查找 232
- 7.11 子模块 233
 - 7.11.1 开始使用子模块 233
 - 7.11.2 克隆含有子模块的项目 235
 - 7.11.3 开发含有子模块的项目 236
 - 7.11.4 子模块技巧 245
 - 7.11.5 子模块的问题 246
- 7.12 打包 248
- 7.13 替换 251
- 7.14 凭据存储 257
 - 7.14.1 底层实现 258
 - 7.14.2 自定义凭据缓存 259
- 7.15 小结 261
- 第8章 自定义Git 262
 - 8.1 配置Git 262
 - 8.1.1 客户端基本配置 262
 - 8.1.2 Git中的配色 265
 - 8.1.3 外部的合并与diff工具 265
 - 8.1.4 格式化与空白字符 268
 - 8.1.5 服务器配置 270
 - 8.2 Git属性 270
 - 8.2.1 二进制文件 271
 - 8.2.2 关键字扩展 273
 - 8.2.3 导出仓库 276
 - 8.2.4 合并策略 277
 - 8.3 Git钩子 277
 - 8.3.1 安装钩子 277
 - 8.3.2 客户端钩子 278
 - 8.3.3 服务器端钩子 279
 - 8.4 Git强制策略示例 280
 - 8.4.1 服务器端钩子 280
 - 8.4.2 客户端钩子 285
 - 8.5 小结 288
- 第9章 Git与其他系统 289
 - 9.1 作为客户端的Git 289
 - 9.1.1 Git与Subversion 289

9.1.2 Git与Mercurial	298
9.1.3 Git与Perforce	305
9.1.4 Git与TFS	317
9.2 迁移到Git	325
9.2.1 Subversion	325
9.2.2 Mercurial	327
9.2.3 Perforce	329
9.2.4 TFS	330
9.2.5 自定义导入工具	331
9.3 小结	337
第10章 Git内幕	338
10.1 底层命令和高层命令	338
10.2 Git对象	339
10.2.1 树对象	341
10.2.2 提交对象	343
10.2.3 对象存储	345
10.3 Git引用	346
10.3.1 HEAD	348
10.3.2 标签对象	348
10.3.3 远程引用	349
10.4 包文件	350
10.5 引用规格	352
10.5.1 推送引用规格	354
10.5.2 删除引用	354
10.6 传输协议	354
10.6.1 哑协议	355
10.6.2 智能协议	356
10.6.3 协议小结	359
10.7 维护与数据恢复	359
10.7.1 维护	359
10.7.2 数据恢复	360
10.7.3 移除对象	362
10.8 环境变量	365
10.8.1 全局行为	365
10.8.2 仓库位置	365
10.8.3 路径规格	366
10.8.4 提交	366
10.8.5 网络	366
10.8.6 差异与合并	367
10.8.7 调试	367
10.8.8 杂项	369
10.9 小结	369
附录A 其他环境中的Git	370
附录B 在应用程序中嵌入Git	381
附录C Git命令	390
• • • • •	(收起)

[精通Git（第2版）_下载链接1_](#)

标签

Git

计算机

版本控制

软件工程

计算机科学

编程

[技术.工具框架]

DevOps

评论

<https://git-scm.com/book/zh/v2>

這種東西，直接看網站就好了。。

挺不错的一本书 但是书名翻译的不好 充其量算是Git高级 和 精通差距还是挺大的吧
四星给的有些勉强 不情不愿 准确来说应该是3.5星

看的电子版 <https://git-scm.com/book/zh/v2>

上路

挺好，学到了stash这个命令，rebase占了好大的篇幅，后面的就没什么印象了

git + sphinx-doc + atom， 写作好帮手

Distributed Version Control System.

很好读，图示非常清楚，特别协作部分很有用

<https://git-scm.com/book/en/v2>
系统，全面，读了前6章，后续遇到问题可以再去翻文档。@沈阳市图书馆

git

很喜欢这种实践+原理的技术书籍，理解原理之后就容易理解每个操作背后是怎样的，为什么可行，这种感觉是很好的。

有过一定的实践经验之后再看，很受用

看了4成，看不下去了，感觉过誉了，书的组织结构有点乱乱的

Git不只是一个源代码管理工具，Git更是一种工作方式。

git入门必备，清晰介绍git版本控制基础；后面章节深入git内部原理，理解难度很大。

平时基本都用 github mac desktop，看完可以卸载了，系统了解 git 的一本好书。

感谢翻译人员的专业。

前几章讲原理，还能搞懂，后面讲的许多用法大多没用只能囫圇吞枣了！以后有机会再在工作中理解高级用法吧

很棒的一本书，每章的内容都不是太长，一两个小时就可以理解和消化。也解答了我长期关于git的一些困惑。算是草草的浏览了一遍，重点看了自己关注的几个地方。

[精通Git（第2版）_下载链接1_](#)

书评

先上个在线阅读的链接 <http://git-scm.com/book/zh/> 先说说对书的内容的评价： -
长短适中：每个section大概10-20min能够读完，可以利用饭后休息时间见缝插针地读 -
内容全面：涵盖了Git的常用命令、分支特性、远程仓库、自己搭建Git服务器、配置、内部原理等各个方面。零基础...

可以看 <http://progit.chunzi.me> 上的中文版，比官方的略新一点。
当然也欢迎加入翻译，<http://groups.google.com/group/progit-zh> 另外
<http://planet.chunzi.me/git> 上聚合了些关于 git 的博客文章。

先是看了《版本控制之道——使用Git》，在其豆瓣评论里发现了关于《Pro

Git》的链接，于是就去下了中文版来读。

此书介绍的内容要比《使用Git》深入许多，尤其是对多人合作和分布式项目管理的讨论，是《使用Git》中所缺少的。在学习了Git的一系列命令之后，如何才能有效的使用..

这本书的作者是github的员工。书写得非常棒，对分支等的讲解使用了很多配图，简单明了，把原理讲得非常易懂。

这书还有一个非常棒的部分就是针对不同的使用场景，设计了几个合适的版本管理策略，比如私有小项目该用什么方式控制，公共项目该用什么方式控制等，这个非常实用。

...

Git is the version control system developed by Linus Torvalds for Linux kernel development. It took the open source world by storm since its inception in 2005, and is used by small development shops and giants like Google, Red Hat, and IBM, and of course ma...

这篇不错。Git版本控制软件结合GitHub从入门到精通常用命令学习手册

Git客户端图文详解如何安装配置GitHub操作流程攻略

<http://www.ihref.com/read-16369.html>

Pro Git中文在线版 <http://iissnan.com/progit/>

非常适合读。比《版本控制之道》好好多。

google还能搜到其它在线阅读网站，我这个个人感觉最好。为什么豆瓣叫豆瓣？

全书对Git的应用场景，以及典型的模式讲得比较透彻。

无论对于初学者还是想进一步了解Git工作原理的同学都比较适合。

初略的看了一遍，不但了解了Git的基本操作，同时也对版本以及分支

的概念有了比较深刻的认识。而且从中能够更深入的理解VCS系统的构造原理。非常赞。对于书...

<https://www.kernel.org/pub/software/scm/git/docs/git-svn.html>

<http://www.blogjava.net/lishunli/archive/2012/01/15/368562.html>

<http://www.open-open.com/lib/view/open1328070404827.html>

<http://www.open-open.com/lib/view/open1328070404827.html>

对于入门和理解Git的思想和内部机制是不错的书~~~~
但感觉还不够，很多实践需要自己通过stackoverflow和git help 来查~~~~
另外，感觉还是没讲清楚merge和rebase的本质区别~~

书名为Pro
，确实是讲解了Git的高级特性，谓之Pro也值。这本书，看了好几天，今天才大概的看完了，没有很详细的看。不过已经开始使用git了，而后把她作为资料查阅了。
在看书的过程中，整理了很多的笔记 <http://blog.supercloudx.com/?cat=43>

从vss到cvs到svn走过来的我，一直把git当“更方便的svn”用，直到看到身边的同事用的出神入化，快捷又方便，才终于找了这本书看。这本书的特点是开始就把概念讲的很清楚，图示做的很传神，即使不看最后一章（讲git内部实现）也能理解得差不多。这样以来对于以前很少自己用的分支...

无疑是学习Git最好的书籍之一。它有几个特点： 1.
Free，包括免费和自由，书的编写本身也是用git来管理的。 2. 深入浅出 3.
涵盖从分布式版本控制的概念、git的使用、git服务的构建、git的customize到最后git的原理，基本印证其Pro的一面。 4. 实践性比较强

书不错，看两个小时就能入手，配图很赞，让你在用git的时候大脑中始终有个版本树，清晰明了，有一种一切都在控制之中的快感。
前三章看完就可以开始正式使用了，没有任何问题。
后面对原理的讲解，尤其是配图很赞，像我这种想象力不高的人也能轻易地明白。

我认为这是目前关于Git最好的一本书，为什么没有出版社把这本书的中文版出版？好吧，豆瓣你到底要我写多少字才肯让我发表？我日！还不够，好吧，，我日日日日日日日日日日！啊！我要疯了！！！原谅我这个语言能力贫乏的人吧！继续日！！！！！

First, this is a FREE book! The second, it is a good book. I read this book on the web, in

2 weeks. It helps me clear many mysteries about git, plus lots of tips & tricks. I don't want to add more praise, but end with its last statement: "feel more comfo..."

[精通Git（第2版）_下载链接1](#)