

Git极简介绍及易错点拾遗

肖斯宇

2023年5月修改

先看这个



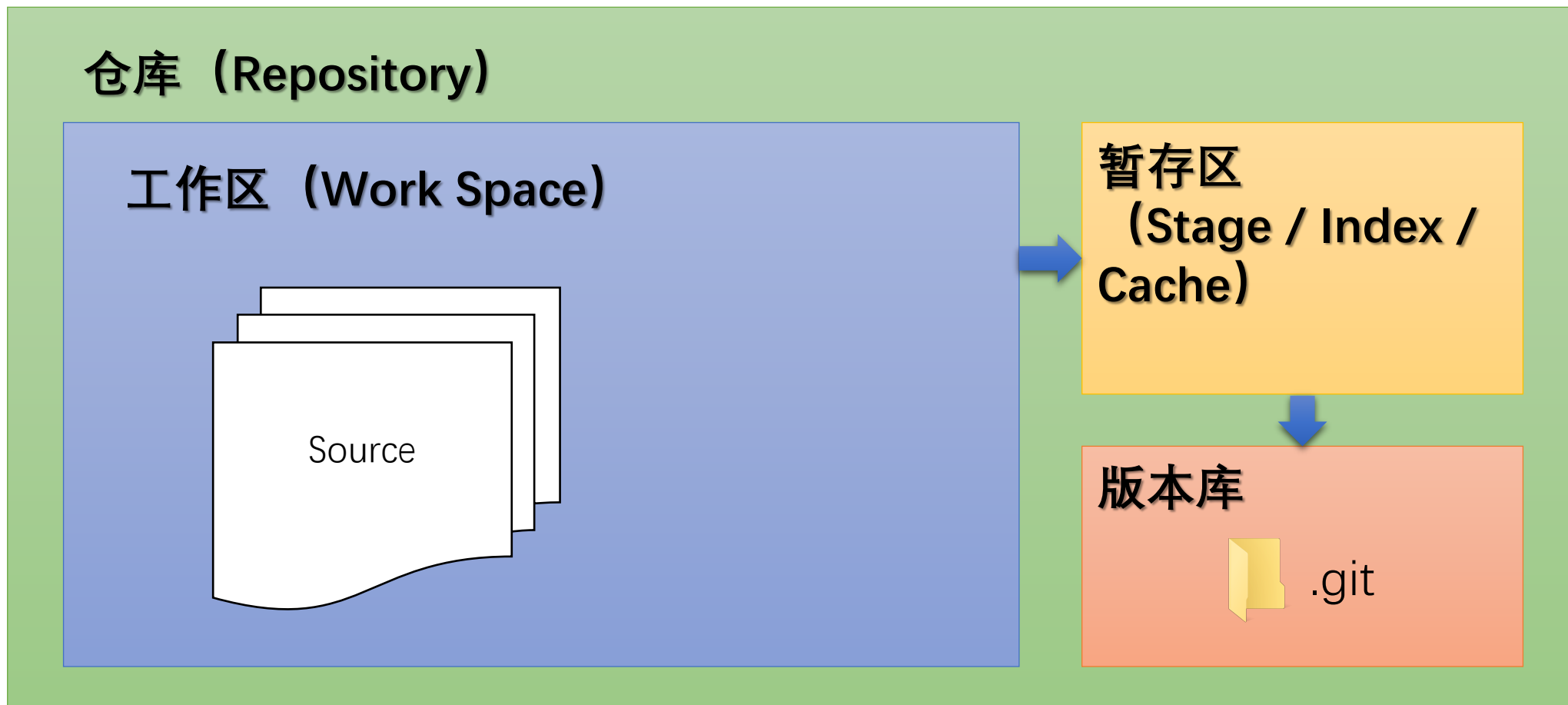
Markdown、Linux、Git快速教程.md



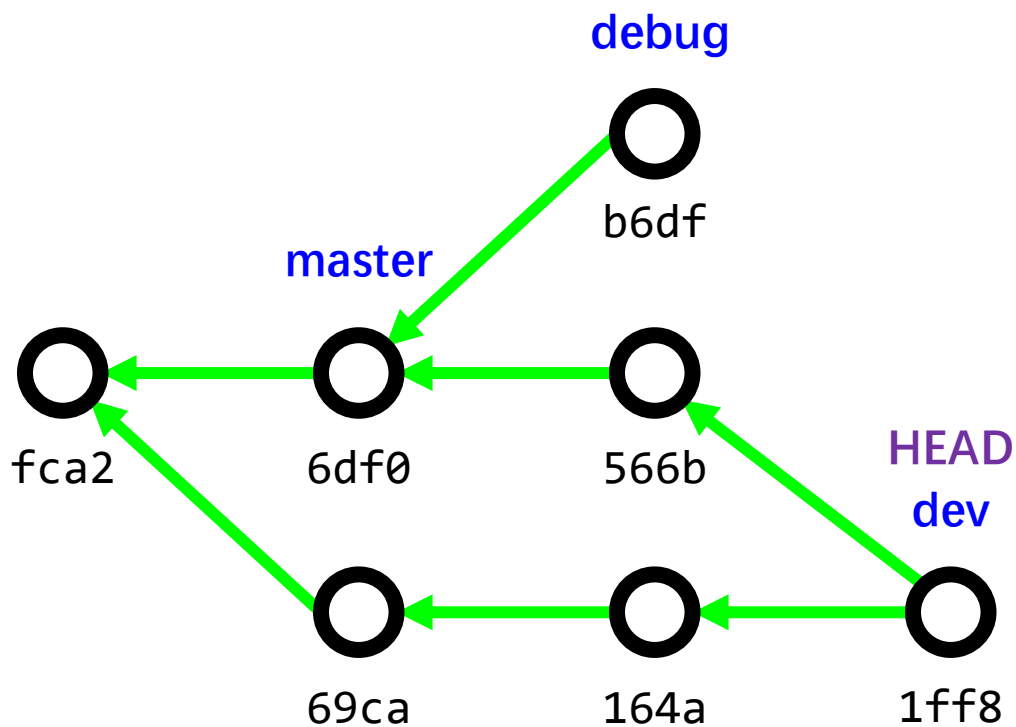
Git是什么

- Git是一个分布式**版本控制**软件
- 核心是版本控制，而非简单的同步文件
- Git记录文件改变的历史

Git仓库的构成



Git仓库的储存拓扑



- 数据结构是树
- 每个提交就是一个节点，使用唯一的十六进制数表示（哈希值）
- 每个节点的父节点是上一次改动（即基于哪一次提交的改动）
- 分支的本质是指向节点的指针
- HEAD指针等于当前工作目录所在分支的最后一次提交

远程仓库

- 远程仓库 (Remote Repository) 是一个逻辑概念，其“远程”相对于仓库所在位置而言
其实远程仓库可能就在电脑上的另一个目录下
- 远程仓库也是一个仓库（但也有一点区别），数据拓扑和分支与本地无异
- Github的SSH服务校园网经常无法访问，建议用Gitee替换（注意信息安全）
教研室有服务器可以使用Git
- 切勿把教研室的代码传到网上（学习相关法律法规知识和近年国家安全教育材料，不要做出给自己带来刑事和民事风险的行为）

补充说明1

- 每次提交（Commit）记录了当前整个工作目录的状态，之后这个提交就不能再改变了，**重要的技术状态一定要保存一个提交以便日后使用！**
- 一般做出一个功能或者做到一个阶段就提交一下，宁愿多提交两次也**不要修改超多地方再提交**！其一，如果提交的代码有问题不方便调试定位，给自己徒增内耗，不值得。其二，有可能写到一半把代码写挂了，白干几天，给自己徒增内耗，不值得。（不要对自己过于自信，这些问题都是发生过几次了，血的教训）

补充说明2

- 每次提交时的**注释**（Comment）**写上有意义的内容**，一般为做了哪些修改、修复了什么问题、程序处于什么状态等。不要写诸如纯日期、“进行了修改”等无意义内容，以后找版本的时候给自己徒增内耗，不值得。
- 大多数时候都要**编写.gitignore**文件，把不需要存进仓库的临时文件过滤掉。防止仓库体积爆炸以及速度变慢。为了防止加入自己不知道的文件，**不要使用git add --all**。使用VS Code的Git工具可以在图形界面上直观的查看状态和进行暂存、提交等简单操作，推荐。
（推荐文本编辑器就选VS Code，几乎能实现任何常见需求）

补充说明3

- Git出错时**要看错误原因**，不要野蛮操作；冲突（Conflict）了去解决，不要使用`git push -f`，有破坏仓库的风险（并且曾经在美国引起枪击事件）
- 仓库不是唯一的，一般来说一个项目创建一个仓库，**不要把不同项目的代码用同一个仓库管理！**
- 团队项目维护一个共同的远程仓库，**不要一人一个仓库！**
- 初学不要使用`git add --all`和`git push -f`！

参考

1. Git教程 - 廖雪峰的官方网站（推荐学习）
<https://www.liaoxuefeng.com/wiki/896043488029600>
 2. Git官网（工具书）
<https://git-scm.com/>
- 大部分问题网上都有资料，有必要还可以去官网查手册，不清楚的地方可以在群里提出来大家一起讨论

末尾的一些废话

- 发挥主观能动性，不懂的东西先主动去查资料（尤其是官方手册），带着学习成果问学习效果最好。研究生肯定要动手研究，克服“等靠要”的思想。
- 大胆学习，谨慎操作，当不知道自己的行为有无危害时，先查资料再问一问总归是好的。（不止一个同学野蛮操作把自己的系统炸了的）
- 拉拉杂杂唠叨了这么多，都是血的教训总结出来的，总之就是不要当大聪明！

谢谢大家