

- 本地安装配置 Git:

`git config --global user.name "Your Name"`

`git config --global user.email "email@example.com"`

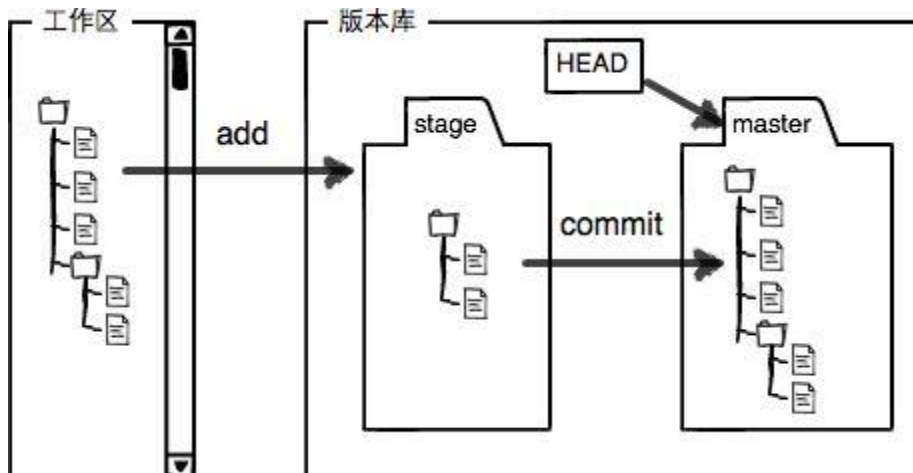
.gitconfig : 在用户根目录下, git 的配置文件(含本机 name 和 email 等), 用于标识本地 git

- 创建/拉取**版本库**(本地仓库):

1. 本地创建:

直接在某一目录下 `mkdir <filename>` → `cd` 进去 → 执行 `git init`(会在当前目录中生成一个 .git 隐藏目录, 这就是后面的版本库, 并默认自动创建一个 master 分支)。即完成了一个空仓库的创建。

2. 从远程拉取



- 修改(增、删、改)版本库内容并提交

`git add/rm -p/<filename>` (可执行多次, 将**工作区**所有修改内容提交到**暂存区**) →

`git commit -m "修改内容说明"` (将暂存区待修改内容一次性提交到本地版本库)。

`git status` 命令可时刻查看本地仓库当前状态(哪些文件被修改但未放入暂存区 or 未提交, 如果没有的话, 也就说明版本库与工作区内容一致, 会显示 clean)。

`git diff <filename>` 命令, 查看该文件的被修改历史。

- 版本回退

`git reset --hard HEAD^ / HEAD~n / commit id`: 加多少^即回退几步, n 为回退的步数。

`git reflog`: 查看历史 git 命令(可用于找 commit id)

`git log( --graph --pretty=oneline --abbrev-commit)`: 注意图是从下往上看, 上面是最

新状态。图中每个 commit id 是独一无二的。

- 撤销修改

`git checkout -- <filename>` : 暂存区/版本库(当暂存区全部提交完成后)覆盖工作区(只覆盖指定文件,下同)

`git reset HEAD <filename>` : 本地版本库覆盖暂存区

- 管理远程仓库

本地创建 SSH key : `ssh-keygen -t rsa -C youremail@example.com`, 用以远程仓库识别你自己 → 用户主目录下的.ssh 文件中包含私钥、公钥等文件, copy 公钥文件(.pub)中的内容, add 到远程仓库的 ssh key 管理。

远程仓库创建: Create a new repo

- 本地仓库关联/克隆远程仓库

1. 关联远程库

在本地声明(添加)远程库地址 : `git remote add origin git@github.com:xxx/xxxx.git`, origin 即为本地为远程仓库起的名字。

查看远程库信息: `git remote (-v)`

指定关联分支: `git branch --set-upstream-to=origin/<远程分支> <本地分支>`

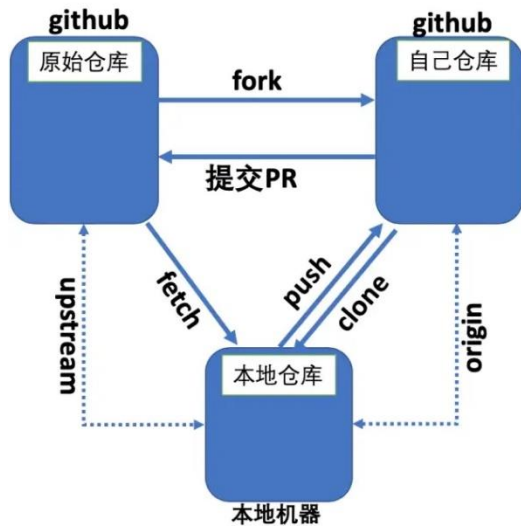
创建关联分支: `git checkout -b <本地分支> origin/<远程分支>`

推送本地修改到远程 master 分支 : `git push -u origin master`, -u 参数同时完成了本地 master 分支与远程 master 分支(没有的话就新建一个)的关联, 后面推送时便不用再加-u。

抓取远程已关联的分支: `git pull`

2. 克隆远程库到本地

`git clone <远程库 ssh/https 路径>`



## ● Git 分支

git branch : 查看所有分支

git branch <分支> : 创建分支

git checkout -b <分支> / git switch -c <分支> : 切换到该分支, -b/-c 参数表示没有则新建。

git merge (--no-ff) <分支> (-m “合并说明”) : 将所指分支 merge 到当前分支中, --no-ff 禁用快进(fast forward)合并方式, 有助于查看历史合并。

git branch -d/-D <分支> : 删除分支, -D 强制删除, 不管该分支有没有完成 merge。

冲突解决: 直接手动修改冲突文件后重新提交, 冲突的分支相应也自动被 merge(可通过 log graph 查看), 即强制分支向前走一步以完成 merge。

git stash: 暂存紧急分支, 前提是暂存区与工作区已同步

git stash list: 查看紧急分支

git stash apply/pop : 恢复(不删除副本)/弹出紧急分支

git cherry-pick <commit id> : 把 bug 提交的修改“复制”到当前分支

git rebase : 把本地未 push 的分叉提交历史整理成直线, 方便查看历史提交 log 图, 但是可能会修改本地分支的提交顺序(最后 merge 结果是一样的)。

Zoom 网络研讨会

快速入门新项目的部署 查看选项

远程仓库-基于分支的协作

- `$ git remote -v` # 查看远程仓库
- `$ git fetch origin master` # 拉取远程分支, 但是不合并
- `$ git pull origin master` # 拉取远程分支并且合并, 等于 `fetch+merge`
- `$ git branch --set-upstream-to=/u LOCAL_BRANCH ORIGIN_BRANCH`
- `$ git push origin dev` # 推送到远程
- `$ git remote add origin REMOTE_URL`
- `$ git remote set-url origin REMOTE_URL`
- `$ git remote remove origin`
- 远程仓库的分支对应在本地的名称, 例如: `origin/master`, 指针不可修改

屏幕设置

聊天 举手 问答

在这里输入你要搜索的内容

21:04 2020/8/29

学习Git的优秀资料

访问官网 <https://git-scm.com>, 使用命令行 `man git`, `git help merge`

通过玩游戏练习 Git 分支操作  
<https://zhuanlan.zhihu.com/p/134346782>

动图展示Git的10大难懂命令  
<https://zhuanlan.zhihu.com/p/132573100>

图解git diff的用法  
<http://marklodato.github.io/visual-git-guide/index-zh-cn.html>

Git内部原理揭秘  
<https://zhuanlan.zhihu.com/p/96631135>