

基础命令及其说明

git 是开源的分布式的版本控制器
用于记录文本文件的内容变化,便于多人协作 repository 库

git 三个工作区域
仓库 用于储存元数据和对象数据库的
工作目录
暂存区

git init	初始化本地版本库 .git
git status	查看工作区和暂存区的状态
git add .	把工作区的所有修改提交到暂存区 . 代表所有的文件
git commit -m "描述"	把暂存区的修改提交到本地版本库
git log	查看历史记录
git diff	查看工作区具体的修改
git diff --cached	查看暂存区具体的修改
git checkout -- 文件路径	撤销工作区的修改
git reflog	查看所有历史记录
git clone	克隆远程仓库的地址
git revert	反转撤销提交
git clean	从工作目录溢出没有track的文件
get stash	把当前改动压入一个栈
git branch	用于列出分支,创建分支和删除分支
git merge	把一个远程分支合并到当前的分支
git tag	在一个提交上建立永久性的书签
git config --global user.name	全局配置用户名
git config --global user.email	全局配置邮箱
git config --list	查看全局配置
git push origin master (分支)	推送代码到远程仓库

git pull origin master (分支)	从远程仓库拉取代码
分支语句 从远程仓库拉取代码	
git checkout -b 分支名	创建并切换分支
git branch 分支名	新建分支
git checkout 分支名	切换分支
git branch	查看本地所有的分支
git branch -r	查看远程分支
git branch -a	查看所有分支(远程/本地)
git branch -d 分支名	删除本地分支
git push origin -d 分支名	删除远程分支
git stash	保存工作区的修改
git stash pop	恢复工作区的修改
git pull	远程拉取代码 和本地的代码合并
git fetch	把远程的代码拉取到本地不合并
git merge origin/ 分支名	合并代码
git pull = git fetch + git merge	

config 配置

把工作区的修改提交到本地版本库

1. git add . (add后加 "." 为全部提交 add后加文件路径则只添加当前文件)
2. git commit -m "描述"

撤销工作区的修改 git checkout -- 文件路径

撤销暂存区的修改 git reset 文件路径

1. git reset 文件路径 从暂存区撤回工作区
2. git checkout -- 文件路径

回退到上一个版本 git reset --hard HEAD^ 回退到上一个版本(^数目为回退版本个数)

回退到指定版本 git reset --hard (commit_id ⑦位id)

github

面向开源及私有软件的托管平台

支持两种协议

1. https 协议
2. ssh 协议

ssh 协议 : 配置公钥和密钥

- ssh-keygen 生成公钥密钥

https 协议: 需要输入用户名和密码

- 需要输入用户名和密码

本地添加远程仓库:

- git remote add origin 仓库地址
- git remote -v 查看关联的远程仓库

需要手动解决冲突,执行 git add ---> git commit -m --->git push origin 分支

报错

- 当 git pull origin master 跳出 fatal: refusing to merge unrelated histories
 - git pull origin master --allow-unrelated-histories