git的使用

git教学游戏

- 工作区：本地文件夹
- 暂存区：暂存文件区
- git仓库：所有的操作记录

命令详解：

- git config --global user.name，用户名 ，这只是本地库使用，和GitHub的用户名和邮箱无关
- git config --global user.email，邮箱,全局配置，任何一个本地库都是使用相同的用户名和邮箱
- git status,查看状态
- git init,初始化本地库
- git add <文件名>，将本地文件添加到暂存区
- git commit -m "提交信息" <文件名>，将暂存区的文件提交到git仓库
- git pull <远程库名> <远程分支>：<本地分支> ，把远程库某一分支中的内容合并到本地分支上，放入工作区中

    若本地库和远程库历史内容不匹配，即本地库最开始不是克隆远程库得来，则加上 --allow-unrelated-histories

- git push <远程库名> <本地分支>：<远程分支> ，把本地分支更新到远程分支，结果是远程分支和本地分支一模一样

    若远程库在本地库拉取之后又提交了，则本地库在push之前应该拉取。git push <远程库名> <本地分支> :<远程分支>

- git clone <URL>，克隆库，放入工作区
- git log，查看当前分支历史版本,只包括commit的记录
- git reflog，查看所有操作记录
- git remote add <远程库命名> <URL> ，添加一个远程库，git remote -v查看对应的地址
- git branch <分支名> <版本号>，创建版本号的分支，git branch -v查看详细信息
- git checkout <分支名>，切换分支，当前工作区也会随之变化
- git merge <分支名>，把分支名合并到当前分支，工作区会更新
- git rebase <分支名>，把当前分支与分支名的共同父级开始的所有当前分支历史版本添加到分支名的当前版本后面
- git reset --hard "版本号"，版本穿梭，可切换到任意版本，工作区会变化，历史记录消失
- git revert <版本号>，撤销版本号的提交，回退到版本号的上一个版本，后新加个提交即是前一个版本
- git cherry-pick <版本号> -n(不自动提交)，把版本号添加到当前版本的后面，可以一次接多个，适用与不同分支之间单笔提交

操作流程：

1.本地库由克隆得来，git clone -> git add -> git commit -> git push

2.本地库初始化创建，git init -> git remote add -> git pull -> git add -> git commit -> git push

注：

1. 冲突时，文件里的特殊符号含义：<<<<<<< HEAD 当前分支的代码 ======= 合并过来的代码 >>>>>>>
2. 冲突发送的原因是两个分支对同一个文件的同一处有不同的修改
3. 冲突提交时，不能带文件名，因为无法区分是哪一个文件，是当前分支还是合并的那个分支不确定
4. push需要GitHub开权限和登录，不是所有人都能push到某一个仓库
5. 所有操作都能在git reflog操作记录中找到