git基础语法

git是一个开源的分布式版本控制系统，用于敏捷高效的处理任何或大或小的项目。
git官网

目录

壹.git工作区、暂存区、版本库

1. 工作区：电脑里能看到的目录
2. 暂存区：add提交的暂存已修改文件的区域
3. 版本库：commit的文件区域

贰.git工作流程

1. 克隆git资源作为工作目录

   `git clone https://github.com/DaiHongquan/LoveStudy.git LoveStudy`
2. 在克隆的资源上添加或修改文件
3. 如果其他人修改了，你可以更新资源

   `git fetch` 更新不合并

   `git pull` 更新合并

   `git add .` 添加到暂存区（不包含删除）

   `git add -u` 添加到暂存区（不包含新增）

   `git add -a` 添加到暂存区
4. 在提交前查看修改

   `git status -s` 查看工作区及暂存区的修改

   `git diff HEAD` 查看工作区及暂存区的修改明细

   `git diff` 查看工作区的修改明细

   `git diff --cached` 查看暂存区的修改明细

   `git diff --stat` 查看摘要而非整个diff
5. 提交修改

   `git commit -m "修改描述"` 提交并添加描述
6. 如若有误，撤回修改并再次提交

   `git log --pretty=oneline` 日志单行显示，查看commit id

   `git reflog --pretty=oneline` 所有日志单行显示，查看commit id

   `git reset --hard HEAD^` 恢复到上一次提交，几个^号表示几次,多个可以使用HEAD~n

   `git reset --hard 576af42` 恢复到commit id为576af42的修改

   `git push -u origin dev` 将当前分支推送到远程dev分支并将当前分支默认关联远程dev分支，后续可以直接使用git push

叁.git基本操作

1. 创建版本库

   `git init` 初始化仓库（需要先切换到新建的目录）
2. 取消已缓存的内容

   `git reset HEAD -- <file>` 取消<file>已缓存的内容
3. 删除文件

   `git rm <file>` 从版本库中删除文件<file>

   `git rm -f <file>` 从版本库中删除文件<file>，如果已经add到暂存区，则需要添加-f参数

   `git rm --cached <file>` 从跟踪清单删除文件<file>，工作区仍然保留文件<file>

   `git rm -r 文件夹` 递归删除文件夹
4. 撤销修改

   `git checkout -- <file>` 让文件回到最近一次commit或add时的状态
5. 版本回退

   `git reset --hard HEAD^` 恢复到上一次提交，几个^号表示几次,多个可以使用HEAD~n，可以使用commit id
6. 移动/重命名

   `git mv 旧文件名 新文件名` 将文件移动并重命名为新文件名

肆.远程仓库

1. 添加远程仓库

   `git remote add origin https://github.com/DaiHongquan/LoveStudy.git` 添加远程仓库
2. 将本地库内容推送到远程仓库

   `git push -u origin master` 将本地仓库当前分支的内容推送到远程仓库的master分支，并将其关联，后续可以使用简化命令
3. 更新数据

```
git checkout dev 本地仓库切换到dev分支
git fetch origin dev 从远程仓库dev分支更新数据到本地仓库的dev分支，但是不合并
git log --pretty=oneline -p dev..origin/dev 显示每次提交的内容差异
git merge --no-ff origin/dev 将origin/dev分支合并到本地仓库的dev分支
```

`git pull --no-ff origin dev` 更新并合并

4. 查看远程仓库信息

`git remote` 查看远程仓库

`git remote -v` 查看远程仓库,显示可以pull和push的地址

5. 推送到远程分支

   `git pull origin dev` 将远程origin dev分支的更新拉取到本地库当前分支

   `git push origin dev` 将本地库当前分支推送到远程origin dev分支

伍.分支管理

1. 分支基础命令

   `git branch -a` 查看所有分支

   `git branch 分支名` 创建分支

   `git branch -d 分支名` 删除分支

   `git push origin -d 分支名` 删除远程分支

`git checkout 分支名` 切换分支

`git checkout -b 分支名` 创建并切换分支

2. 分支合并

   `git merge --no-ff -m "合并描述" 分支名` 将分支名合并到当前分支并产生一个新的提交

3. 解决冲突

   `git commit -m ""` 手动解决冲突后提交

4. 查看分支历史

   `git log --graph --pretty=oneline --abbrev-commit` 手动解决冲突后提交

陆.标签管理

1. 创建标签

   `git tag` 查看标签

   `git show v1.0` 查看标签v1.0信息

   `git tag v1.0` 创建标签，标签默认打在最新提交的commit id上

   `git tag v1.0 d6d07e4` 直接在commit id上打标签

`git tag -a v1.0 -m "说明文字" d6d07e4` 直接在commit id上打标签，-a 指定标签名，-m指定说明文字

2. 标签操作

   `git push origin v1.0` 推送某个标签到远程

   `git push origin --tags` 一次性推送所有标签到远程

   `git tag -d v1.0` 删除标签

   `git push origin :refs/tags/v1.0 To https://github.com/DaiHongquan/LoveStudy.git` 删除远程标签

柒.git配置

`git config --global user.name 295454688@qq.com` 设置当前用户的用户名为295454688@qq.com，--global换成--system或者不加分别对应当前系统可当前项目

`git config --list` 查看当前配置

~/.gitconfig查看或者/etc/gitconfig

回到目录