Git相关命令

1. 初始化本地仓库

Git init

1. 克隆远程仓库

Git clone 远程仓库地址

1. 查看当前分支状态

Git status

1. 查看修改内容

Git diff <filename>

1. 添加内容到暂存区

添加具体文件：Git add <filename>

添加当前目录所有的文件： git add .[.表示当前目录所有的内容]

添加任意文件： git add \*

1. 添加修改后的文件到版本库中

Git commit -m “注释内容”[若直接git commit 会弹出一个框要求输入注释内容，最好加上注释]

1. 添加并提交到版本库

Git commit -a -m “注释”

1. 本地仓库代码添加到远程仓库

Git push origin <branchname>

二、还原文件

1、修改了工作区或删除了某个文件，想还原时：

Git checkout <filename>

2、修改的文件添加加到暂存区时（即执行了add操作），想放弃修改时;要两步：

第一步：git reset HEAD <fileName>

第二步：git checkout -- <fileName>

3、修改的文件已经提交了修改（即执行了commit操作），想要撤销本次提交，前提是还没有推送到远程库：

git reset --hard HEAD^

或 git reset –-hard HEAD~1

4、还原到某个版本

Git reset --hard <版本号>版本号通过命令：git log或者git reflog查找

1. 解决冲突
2. 情形：
3. 本地仓库的旧版本和远程仓库新版本都改了同一个文件，在同步远程仓库的文件时会引发冲突。
4. 两个分支都改了某个文件，合并分支是会引发冲突
5. 解决：
6. 先观察命令行提示，比如出现（master|MERGING）
7. 定位冲突文件
8. 修改冲突文件
9. <<<<<<<HEAD 和=======之间是本地仓库的文件的内容，===========和>>>>>>>>>版本号之间是远程仓库之间文件的内容
10. 合并、修改、删除冲突内容
11. 分别git add、git commit git push解决冲突后的文件
12. 分支管理
13. 创建基于master的本地分支：git branch <branchname>
14. 切换分支：git branch <branchname>
15. 删除本地分支：git branch -d <branchname>
16. 强制删除本地分支：git branch -D <branchname>
17. 查看分支：git branch （-a参数查看到远程分支。远程分支显示是红色的）
18. 创建加切换分支 git cheeckout -b <branchname>
19. 合并某分支到当前分支：

先切换到主分支：git checkout master

把分支<branchname>合并到主分支：

git merge <branchname> 或 git merge --no-ff <branchname>

(默认情况下，Git执行快进式合并,会直接将Master分支指向Develop分支

加上--no-ff参数后，会执行正常合并，在Master分支上生成一个新节点,

merge的时候如果遇到冲突，就手动解决，然后重新add，commit即可)

8、查看合并的分支图表：

git log --graph --pretty=oneline --abbrev-commit

9、新建一个基于远程的dev分支：git checkout -b dev origin/dev

比较两个分支的不同：git diff branch branch2 > E:/diff/change.diff

1. 分支管理策略
2. 主分支：
3. Master
4. Devlop
5. 临时性分支
6. 功能分支（feature）
7. 与发布分支（release）
8. 修补bug分支（fixbug）