# Git常用命令

添加、提交

git add

git commit

查看历史记录

git log

版本回退

git reset

记录每一次命令

git reflog

放弃工作区的修改

git checkout --<file>

git checkout 加上 -b参数表示创建并切换，相当于下面两条命令：

git branch dev 创建分支dev

git checkout dev 切换分支到dev / git switch dev

git branch 查看当前分支

git branch -d dev 删除dev分支

git branch -D dev 强行删除dev分支

表示将在工作空间但是不在暂存区的文件撤销更改

git restore <file>

将暂存区的文件撤出，但不会更改本地文件

git restore --staged <file>

把dev分支合并到当前分支

git merge dev

在本地仓库中创建一个远程仓库的别名origin, 指令结尾是仓库的地址

git remote add origin [git@github.com:XXXXX/XX.git](mailto:git@github.com:XXXXX/XX.git)

git push -u origin master

第一次推送master分支时，由于远程库是空的，加上-u参数，把本地master分支推送到远程origin主机，同时指定origin为默认主机，关联起本地和远程分支

push分支dev到远程仓库

git push origin dev

指定提交到远程仓库的某个分支上,此处为master

git push origin dev:master

查看分支合并情况

git log --graph

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

通常，合并分支时，如果可能，Git会用Fast forward模式，但这种模式下，删除分支后，会丢掉分支信息。

如果要强制禁用Fast forward模式，使用 –no-ff命令，Git就会在merge时生成一个新的commit，这样，从分支历史上就可以看出分支信息。

git merge --no-ff -m "merge with no-ff" dev

保存工作现场

git stash

查看保存的工作现场

git stash list

恢复工作现场

git stash apply stash@{0} # 0代表恢复的序号

删除stash内容

git stash drop

回复并删除

git stash pop

复制特定提交到当前分支

git cherry-pick

查看远程仓库信息

git remote

查看更详细的信息

git remote -v

把本地的dev分支推送到远程库

git push origin dev

拉取最新提交

git pull

git pull失败，原因可能是未建立指定本地分支与远程分支的链接,建立链接：

git branch –set-upstream-to=origin/dev dev

git rebase 把本地未push的分叉提交历史整理成直线

给最新提交打上标签

git tag v1.0

查看所有标签

git tag

给历史提交打上标签

git tag v0.8 f52c633

查看标签信息

git show v0.9

创建带说明的标签，-a指定标签，-m指定说明文字

git tag -a v0.1 -m “version v0.1 released” f52c633

删除标签

git tag -d v0.1

推送标签到远程

git push origin v1.0

一次性推送全部尚未推送到远程的本地标签

git push origin –tags

删除远程标签：

1. 删除本地标签
2. 从远程删除：git push origin :refs/tags/v0.9