**GIT命令**

**$ git init //初始化一个git本地仓库**

**$ git remote -v //查看远程仓库地址**

**$ git remote add <仓库名> <仓库地址> //添加要链接的远程仓库及地址**

**$ git remote remove <仓库名> //删除指定仓库及对应地址**

**$ git clone --branch or -b <分支名or标签名> <下载地址> //从远程仓库克隆到本地**

**$ git branch //查看本地分支**

**$ git branch -a //查看本地及远程分支**

**$ git branch -m <分支名> //将当前分支设定为主分支，并重新命名**

**$ git branch -D <分支名> //删除本地分支**

**$ git branch <分支名> //创建分支**

**$ git branch <分支名> --track //创建分支并链接当前所处的本地分支**

**$ git branch <分支名> --track <要链接的分支名> or git branch --track <分支名> <要链接的分支名>//创建分支并链接本地的某个分支**

**$ git branch --set-upstream-to <远程分支名> <本地分支名> //将本地分支与远程分支链接**

**$ git checkout <分支名> //切换分支,若branch本地不存在，且与远程分支同名，则以远程同名分支为模版创建分支，并自动链接**

**$ git checkout -b <分支名> //创建并切换至分支（-B可强制创建）**

**$ git checkout -b <分支名> --track //创建并切换至分支，且跟踪当前所处的本地分支**

**$ git checkout -b <分支名> --track <要链接的分支名>//创建并切换至分支，且跟踪指定的本地分支**

**$ git checkout --orphan <分支名> //创建一个裸分支**

**$ git status //查看本地仓库状态**

**$ git diff <document> //查看文件修改情况**

**$ git add . //提交所有至暂存区（i、u、d）**

**$ git add <document> //提交指定文件至暂存区**

**$ git commit -m "describe" //提交至本地配置库**

**$ git restore . //撤销工作区的修改（只会撤销U）**

**$ git restore --staged . //取消暂存**

**$ git log //查看历史提交日志（加--pretty=oneline或--oneline可简化）**

**$ git reflog //查看所有分支的所有操作记录（包括已经被删除的 commit 记录和 reset 的操作）**

**$ git reset //撤销当前add，默认--soft**

**$ git reset <commit\_id>//回滚至指定commit，只回退了commit的信息，若想再次提交，直接commit即可，默认--soft**

**$ git reset --hard <commit\_id>//彻底回滚至指定的某个版本，本地工作区内容也会变回至某版本**

**$ git reset --hard HEAD~n //彻底回滚至当前往前的第n个版本**

**$ git tag -a <name> -m "describe" //打标签**

**$ git pull //将远程分支的更新状况同步到本地（已形成链接时可用，加--allow-unrelated-histories忽略历史commit是否有关联）**

**$ git pull <仓库名> <远程分支名> //将指定的远程分支更新状况同步到本地**

**$ git push //将本地当前分支的内容跟新到远程分支上（已形成链接时可用，加-f强推）**

**$ git push <仓库名> <远程分支名> //若要更新的远程分支未与本地分支链接，则将当前分支跟新到远程分支上；若更新的远程分支存在与本地分支的链接，则将对应的本地分支跟新到远程分支上**

**$ git push <仓库名> --tag //将本地所打标签推至远程仓库**

**$ git push --set-upstream <仓库名> <远程分支名> //将当前分支的内容更新到远程分支上，并与远程分支建立链接**

**$ git fetch <仓库名> <远程分支名> //将远程仓库指定分支内容抓取到本地**

**$ git fetch --all //将远程仓库所有内容抓取到本地（后可使用git reset --hard <远程分支名，如：myori/1314**

**>将当前分支内容强制更改为与远程分支一致）**

**$ git cherry-pick <commit\_id> //将指定的commit合并至当前分支**

**$ git stash //提交暂存后可使用该命令进行搁置**

**$ git stash list //查看被搁置的所有版本**

**$ git stash apply //取回搁置内容（工作区存在冲突文件且未add时无法取回，工作区存在冲突文件但已add时可取回，自动合并冲突）**

**$ git stash pop //取回搁置区的第一个**

**$ git stash apply stash@{n}//取回搁置区的第n个（n=0...∞，0为最新一次搁置的信息，依次类推）**

**$ git stash drop <id>//删除某一条搁置记录**

**$ git stash clear //清空搁置记录**

**$ git commit --amend // 修改当前commit注释**

**$ git merge <分支名> //将某分支合并入当前分支，commit合并方式遵循修改时间，且会新增合并commit记录**

**$ git merge --abort //撤销当前合并（出现冲突时可用）**

**$ git rebase <分支名>// 将某分支合并入当前分支，并入分支的历史commit会被放至当前分支根处，不会新增合并commit记录**

**$ git rebase -i <commit\_id>// 回至某次提交**

**简述：**

**\*本地存在未提交的代码时无法pull；**

**\*pull成功一次后，若云端代码未更新，之后的pull操作不会再将云端代码同步到本地；**