**git config --global user.name “用户名”** 设置用户名和邮箱 注：带--global参数表示为机器上所有git仓库使用此配置

**git config –global user.email “邮箱”**

**git config –global color.ui true** 配置git的显示效果

**git config –l** 查看全局配置信息

**git init** 将当前目录初始化为git仓库

**git add 文件名** 将文件修改添加到暂存区

**git add –A** 将目录下所有文件修改添加到暂存区

**git add –f 文件名** 强制添加某个文件到仓库

**git rm 文件名** 将已删除的文件在暂存区中删除

**git commit –m “版本信息”** 将暂存区提交到仓库

**git status** 查看当前仓库状态（工作区与仓库的区别）

**git diff** 查看工作区与仓库最新版本的具体区别

**git diff HEAD – 文件名** 查看工作区与具体仓库版本具体文件的具体区别

**git log** 查看最近的提交日志

**git log -–pretty=oneline** 简化的日志

**git log –graph** 查看分支合并图

**git reflog** 查看最近的git命令，以及每次命令对应的版本号

可以利用它查看想要恢复的版本号

**git reset –-hard HEAD^** 回退到上个版本

HEAD表示这个版本，HEAD^表示上个版本，HEAD~n表示前n个版本

**git reset -–hard 版本号** 回退某个版本或恢复某个版本

版本号可以不写全，git将自动查找匹配版本号

**git reset HEAD 文件名** 将某个文件修改从暂存区撤销

**git checkout – 文件名** 将某个文件恢复到最近一次add或commit的状态

删除同样适用

**git clone 地址** 将远程库克隆到本地

**git remote** 查看远程库信息

**git remote –v** 查看远程库详细信息

**git remote add 远程库名 地址**  关联本地库和远程仓库

**git pull** 从远程库抓取最新提交

**git push [-u] 远程库名 分支名** 将本地仓库推送到远程仓库 注：-u参数是推送时将本地库分支与远程库分支关联，通常第一次推送时使用

**git push 远程库名 标签名** 将指定标签推送到远程

**git push 远程库名 –tags** 向远程推送全部未推送的标签

**git push 远程库名 :refs/tags/标签名** 删除一个远程标签

**git checkout 分支名** 切换分支

**git checkout –b 分支名** 创建并切换到分支 注：-b表示切换到分支

**git checkout -b 分支名 远程库名/分支名** 创建与远程库对应的分支

**git branch** 查看分支

**git branch 分支名** 创建分支

**git branch –d 分支名** 删除指定分支

**git branch –D 分支名** 强行删除未合并分支

**git branch –set-upstream 分支名 远程库名/分支名** 建立本地分支与远程分支的关联

**git merge 分支名** 合并指定分支到当前分支

**git merge –-no-ff –m “描述” 分支名** 禁用快速合并（fast forward）来合并分支 注：这将创建一个commit来合并分支，而不是直接改变分支指向

**git stash** 储存工作现场

**git stash list** 查看工作现场

**git stash apply** 恢复工作现场

**git stash drop** 删除工作现场

**git stash pop** 恢复同时删除工作现场

**git tag** 查看所有标签

**git tag 标签名** 在当前分支最新commit创建一个标签

**git tag 标签名 commit号** 在指定commit创建一个标签 注：commit号不用写全

**git tag –a 标签名 –m “说明”** 创建带有说明的标签

**git tag –s 标签名 –m “说明” 私钥** 用PGP私钥签名一个标签

**git tag –d 标签名** 删除标签

**git show 标签名** 查看指定标签的详细信息

**小技巧**

**生成SSH key** ssh-keygen -t [rsa|dsa] 注：windows ssh key存放目录：C://Users/用户名/.ssh/

**.gitignore无效** .gitignore只能忽略那些原来没有被track的文件，如果某些文件已经被纳入了版本管理中，则修改.gitignore是无效的。解决方法：git rm -r --cached