**一、本地操作：**

**1.其它**

git init：初始化本地库

git status：查看工作区、暂存区的状态

git add <file name>：将工作区的“新建/修改”添加到暂存区

git rm --cached <file name>：移除暂存区的修改

git commit <file name>：将暂存区的内容提交到本地库

　　tip：需要再编辑提交日志，比较麻烦，建议用下面带参数的提交方法

git commit -m "提交日志" <file name>：文件从暂存区到本地库

**2.日志**

git log：查看历史提交

　　tip：空格向下翻页，b向上翻页，q退出

git log --pretty=oneline：以漂亮的一行显示，包含全部哈希索引值

git log --oneline：以简洁的一行显示，包含简洁哈希索引值

git reflog：以简洁的一行显示，包含简洁哈希索引值，同时显示移动到某个历史版本所需的步数

**3.版本控制**

git reset --hard 简洁/完整哈希索引值：回到指定哈希值所对应的版本

git reset --hard HEAD：强制工作区、暂存区、本地库为当前HEAD指针所在的版本

git reset --hard HEAD^：后退一个版本

　　tip：一个^表示回退一个版本

git reset --hard HEAD~1：后退一个版本

　　tip：波浪线~后面的数字表示后退几个版本

**4.比较差异**

git diff：比较工作区和暂存区的**所有文件**差异

git diff <file name>：比较工作区和暂存区的**指定文件**的差异

git diff HEAD|HEAD^|HEAD~|哈希索引值 <file name>：比较工作区跟本地库的某个版本的**指定文件**的差异

**5.分支操作**

git branch -v：查看所有分支

git branch -d <分支名>：删除本地分支

git branch <分支名>：新建分支

git checkout <分支名>：切换分支

git merge <被合并分支名>：合并分支

　　tip：如master分支合并 hot\_fix分支，那么当前必须处于master分支上，然后执行 git merge hot\_fix 命令

　　tip2：合并出现冲突

　　　　①删除git自动标记符号，如<<<<<<< HEAD、>>>>>>>等

　　　　②修改到满意后，保存退出

　　　　③git add <file name>

　　　　④git commit -m "日志信息"，此时后面不要带文件名

**二、本地库跟远程库交互：**

git clone <远程库地址>：克隆远程库

　　功能：①完整的克隆远程库为本地库，②为本地库新建origin别名，③初始化本地库

git remote -v：查看远程库地址别名

git remote add <别名> <远程库地址>：新建远程库地址别名

git remote rm <别名>：删除本地中远程库别名

git push <别名> <分支名>：本地库某个分支推送到远程库，分支必须指定

git pull <别名> <分支名>：把远程库的修改拉取到本地

　　tip：该命令包括git fetch，git merge

git fetch <远程库别名> <远程库分支名>：抓取远程库的指定分支到本地，但没有合并

git merge <远程库别名/远程库分支名>：将抓取下来的远程的分支，跟当前所在分支进行合并

git fork：复制远程库