分布式版本控制系统 - 适合于社会化编程 - Git / Mercurial

集中式版本控制系统 - 需要中央服务器 - SVN

安装......

安装好之后：

1、在本地创建本地Git仓库

新建目录

cd 进入目录

git init - 初始化该目录为仓库

2、配置全局用户名和邮箱

git config --global user,name "your username"

git config --global user.email "your email"

3、将文件、目录纳入本地仓库，注意：要先在新建的目录中放入要进行版本控制的文件、目录

git add 文件 - 将文件添加到暂存区

git add . - 将当前目录下的所有文件、目录都添加到暂存区

git status - 查看哪些文件在暂存区

git commit -m 'string' - 将在暂存区的文件、目录提交到本地仓库（即纳入了版本控制），string 为当前版本的提示信息，便于区分不同的版本

其他操作：

git log - 查看git提交日志

git log --pretty=oneline - 单行查看日志

git reset --hard 版本号 - 回到版本号所在的历史版本

git reflog - 查看所有的提交日志，包括回退后未来的版本信息，这样可以找回未来的版本

git checkout -- 文件/目录 - 将暂存区的该文件/目录撤回，这样就可以重新修改内容，让后在添加、提交

git checkout -- - 撤回所有暂存区的文件、目录，这样就可以重新修改内容，让后在添加、提交

git rm 文件 - 从本地库移除文件

从远程仓库克隆到本地仓库：

git clone 远程仓库地址 - 从远程仓库克隆一个仓库

git push origin master - 将本地仓库同步到远程仓库

git pull - 将远程仓库的文件拉到本地仓库

将本地仓库发送到远程仓库：

先在远程仓库中新建一个仓库

git remote add origin 远程仓库地址 - 添加远程仓库，和本地仓库对应

git push -u origin master - 提交到远程仓库，第一次需要加参数 -u ，以后不用

创建本地分支

git branch - 查看分支列表

git branch 分支名 - 创建新分支

git checkout 分支名 - 切换到该分支

将其他分支合并到master分支

先切换到master分支

git merge 其他分支

git push origin master

