版本控制管理

git的基本命令：

cd ： 进入；

cd .. ： 返回上一目录；

reset ： 清屏；

pwd ： 显示当前目录；

touch ： 新建一个文件；

mkdir ： 新建一个目录；

rm ： 删除一个文件；

rm –r ：删除一个文件夹；

mv ：移动文件git mv Src Dest

ls(ll) ：列出当前目录中的所有文件

help config 获取帮助信息

clone [url] ：辅助服务器端仓库

log ：查看提交

git stash #暂存进度

git stach pop #恢复进度

## 1 Git使用流程

安装Git

用户配置： $ git config --global user.name "John Doe"

$ git config --global user.email [johndoe@example.com](mailto:johndoe@example.com)

检查配置信息：

$ git config –list

上面的只需要写一次，下面描述简单的git仓库创建

git init – 在当前目录下创建初始化仓库

touch .gitignore – 创建忽略文件，其中#表示解释，\*.[oa]表示忽视.o和.a文件

vim FileName – 新建并编辑文件

git checkout –FileName– 撤销文件修改

git add FileName– 为仓库添加文件到暂存区

git add –A – 将所有未忽视的文件添加到暂存区

git reset HEAD FileName --撤销文件暂存

git commit –m ”interpreter“ -- 将暂存区的文件提交

git commit –a –m ”interpreter“ --将工作区中的未忽略文件全部进行提交

git commit –amend – 对上一次提交的内容的注释进行更改

git rm FileName– 从暂存区移除（磁盘中保留）

git rm --cached FileName --从 Git 仓库中删除（磁盘中保留）

远程基础操作：

A

git clone <URL> – 复制远程仓库到本地

cd newGitName

git remote – 查看该仓库的信息

git remote –v – 查看URL信息

git remote add <shortNmae> <URL> -- 添加一个远程仓库

git fetch < shortNmae> -- 拉取远程仓库中的内容，但不会覆盖本地内容

git push [remote-name] [branch-name] --推送到远程仓库

git remote rename <srcName> <destName> -- 重命名远程仓库名字

git remote rm – 删除远程仓库

标签：（轻量标签和附注标签）

轻量标签很像一个不会改变的分支，它只是一个特定提交的引用；附注标签是存储在 Git 数据库中的一个完整对象，是可以被校验的；其中包含打标签者的名字、电子邮件地址、日期时间，还有一个标签信息，并且可以使用 GNU Privacy Guard （GPG）签名与验证。 通常建议创建附注标签，这样你可以拥有以上所有信息；但是如果你只是想用一个临时的标签，或者因为某些原因不想要保存那些信息，轻量标签也是可用的。

git tag -a v1.4 -m 'my version 1.4' – 创建标签