概述：懂得git，第一件重要的事就是要清楚它与Svn、Perforce或者任何你用过的版本控制工具都有着很大差别。通常，忘掉你预想的版本控制方式，改以Git的方式思考，能够帮助你更好的学习Git。

1.$ cp -R EfDemo Efdemo.bak

这样，你就可以在事情变得一团糟的时候很方便的返回到之前的状态，或者通过对比当前与之前的拷贝，看看自己在之后的工作中，都做了哪儿些修改。

git add a.txt 提交单个文件， git add . 提交所有文件，git add \*.txt 提交所以.txt文件

:

git commit -m "增加了xxx个文件"

:

git push origin master 将分支master提交到远程库origin 别名 ,git push [alias] [branch]

:

git pull origin master 从远端仓库提取数据并尝试合并到当前分支

git clone “git url” “本地文件夹”

git fetch 从远端仓库下载分支与数据

git merge [branch1]将分支合并到当前分支

git remote 管理远程仓库 默认列出远程仓库名

git remote –v 查看仓库及实际地址

git remote add [alias] [url] 为你的项目新加一个远程仓库 ,别名，地址

git remote rm [alias] 通过别名删除远端

git log [branch] 显示一个分支中提交的更改记录，默认当前分支

git log –oneline 显示日志简短信息

git log –graph 查看历史中什么时候出现过分支、合并。

git log –decorate 可以查看tag标签

git log branchA ^branchB 通过查看分支中另一个分支看不到的记录

git log –author=[name] [-num] 只查看用户name的记录,-num 前几条

ls查看当前分支下目录、文件

git reset HEAD 取消之前添加到缓存(git add)文件，但不希望包含在下一提交快照中的缓存。

echo ‘test content’ > test.txt 创建一个文件,如果文件已存在 则覆盖。

git rm file 会将条目从缓存区中移除。默认情况 也会将文件从缓存区和硬盘中删除，如果要在工作目录中保留该文件，可以使用 git rm file –cached。

分支操作

git branch 查看所有分支，标\*号为当前分支

git branch testbranch 新建一个分支testbranch

git checkout testbranch 切换到testbranch分支

git checkout –b branch2 创建分支branch2并立即切换到该分支

git branch –d branch2 删除一个分支且分支已合并或没有未提交更新，当前分支不能删除,

如果分支有未提交内容且未合并需要用-D

git merge branche2 将branch2合并到当前所在分支

说明：当在某个分支(testbranche)下新增一个文件(git echo ‘测试内容’ > test.txt)，

执行git add \*.txt ; git commit –m “新增了一个文件”. 切换到分支master(git checkout master);

执行ls 发现test.txt文件消失，切换到分支git checkout testbranch，test.txt文件又重现。

git tag 给历史记录中某个重要的节点打上标签

git tag –a v1.0 ,git 会打开你的编辑器，让你写一句注解，就像你给提交写注解一样。ZZ保存并退出编辑器。

git tag –a v0.5 7eea869 追加tag.

git log –oneline –decorate –graph 查看tag。

ssh-keygen -t rsa 生成GIT公钥

git config –global user.name “charles.he” //设置用户名

git config –global user.email “[charles0525@sina.cn](mailto:charles0525@sina.cn)” //设置邮箱