**Github及Git**

官网：github.com

借助github托管项目代码

仓库（Repository）：用来存放项目代码，每个项目对应一个仓库

关注（Watch）：当项目更新可接收通知

收藏（Star）：方便下次查看

事务卡片（Issue）：发现代码bug，需要讨论时用

克隆（Fork）：forked from 张三/test仓库（该fork的项目是独立存在的）

发起请求（Pull Request）

**开源项目贡献流程：**

1、新建Issue——提交问题/建议/想法

2、Pull Request——步骤：①Fork项目 ②修改代码 ③发起Pull Request ④等待作者审核

新建文件要注意加文件后缀

github在国外服务器，所以访问较慢或无法访问，需要翻墙（Shadowsocks）

官网：git-scm.com

通过git管理github，托管项目代码

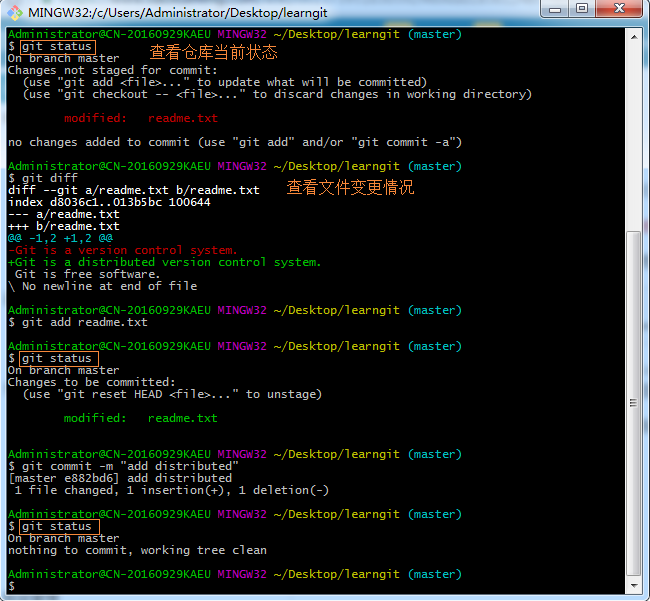
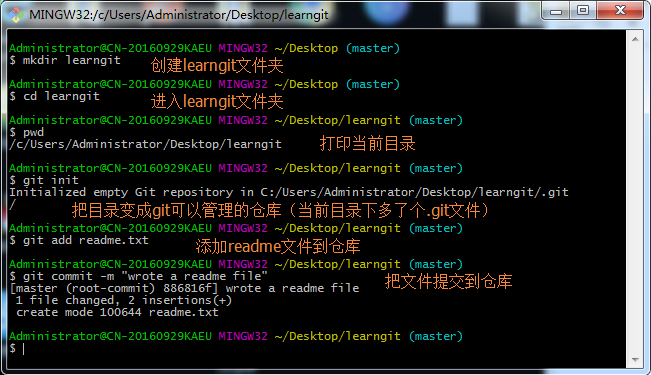
安装时选择使用命令行环境：Use Git from Git Bash only

安装完后设置用户和地址：

$ git config --global user.name "Your Name"

$ git config --global user.email "email@example.com"

Git Bash Here命令行模式 Git GUI Here图形界面



git diff HEAD -- readme.txt命令可查看工作区和版本库里最新版本的区别

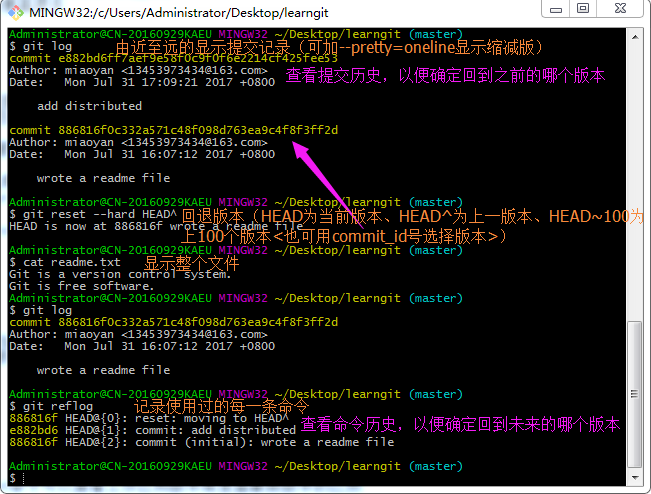
git checkout -- readme.txt可以丢弃工作区的修改（总之，就是让这个文件回到最近一次git commit或git add时的状态）

git reset HEAD readme.txt命令可把暂存区的修改撤销掉，重新放回工作区

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令git checkout -- file。

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令git reset HEAD file，就回到了场景1，第二步按场景1操作。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，可用[版本回退](https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013744142037508cf42e51debf49668810645e02887691000)，不过前提是没有推送到远程库。



git reset命令既可以回退版本，也可以把暂存区的修改回退到工作区

git reset --hard HEAD^/commit\_id 回退到上一版本/id对应的版本

git reset HEAD readme.txt命令可把暂存区的修改撤销掉，重新放回工作区

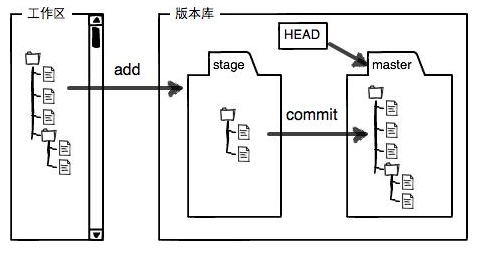
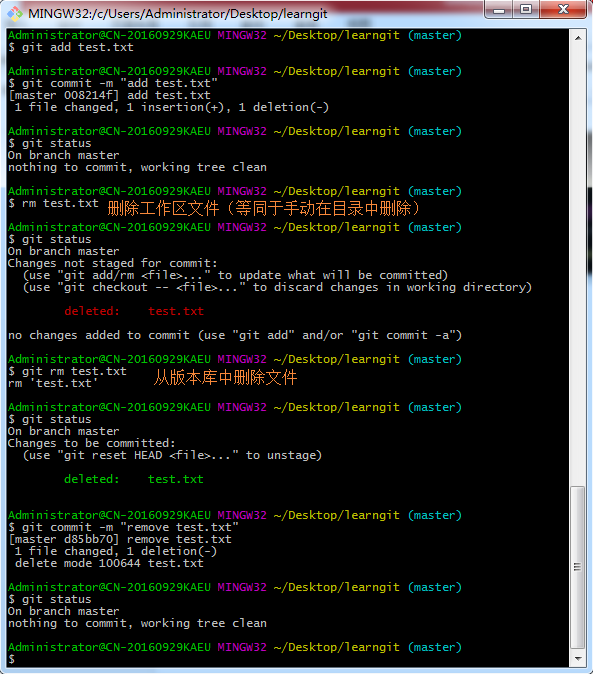


图1



$ rm test.txt删除工作区文件后的两种情况：

情形一：继续删除

$ git rm test.txt 从版本库中删除文件

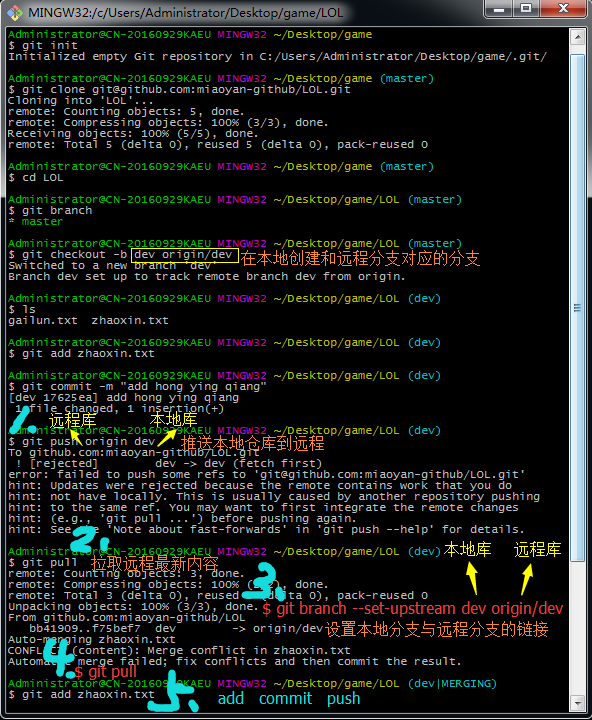
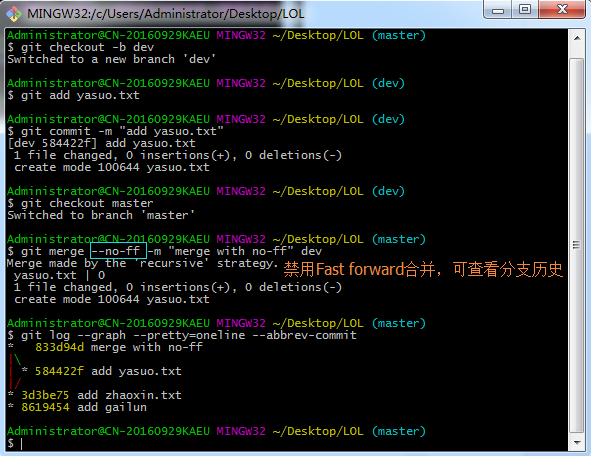
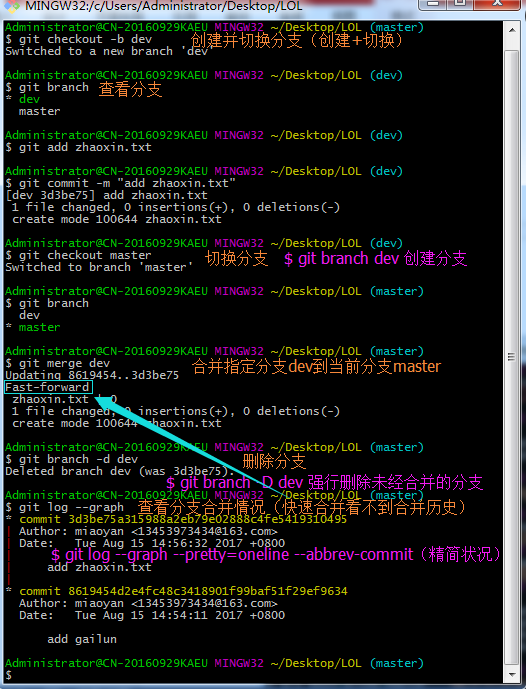
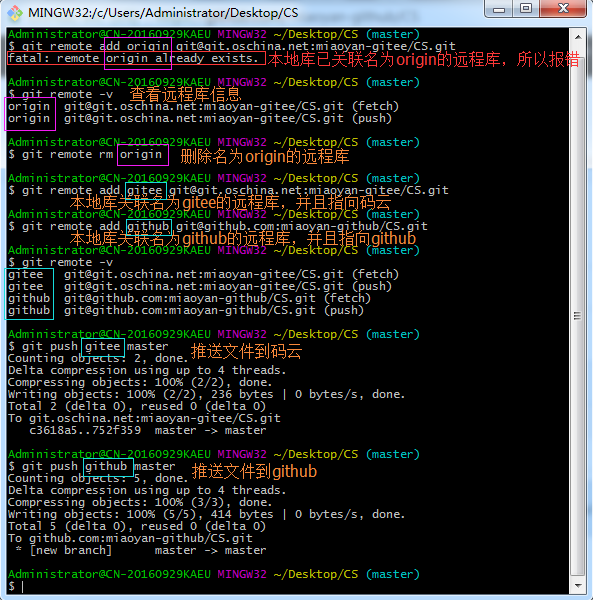
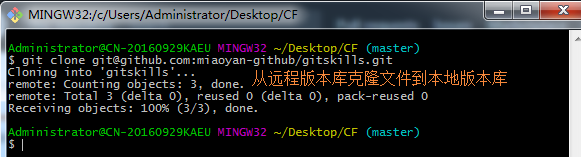
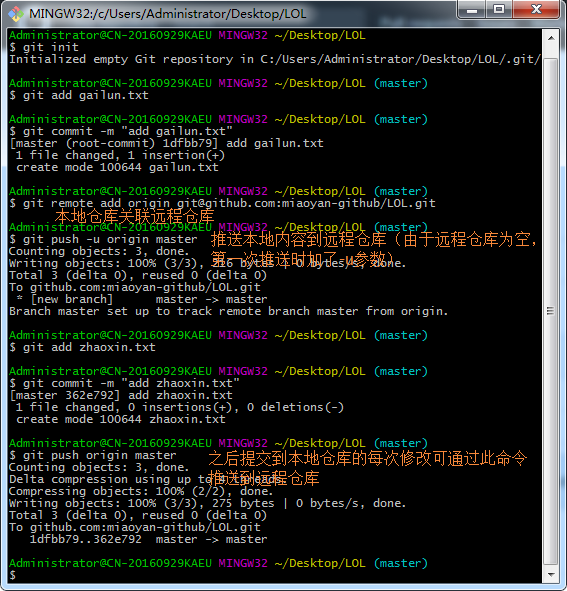
$ git commit –m “remove test.txt”

情形二：误删恢复

$ git checkout -- test.txt ( - - 不能省，否则为切换分支)

git checkout其实是用版本库里的版本替换工作区的版本，无论工作区是修改还是删除，都可以“一键还原”

（参看图一）



1. $ git push origin dev推送自己的修改（失败） 与他人推送内容冲突，需抓取最新提交
2. $ git pull抓取最新提交（失败） 未指定本地分支与远程分支的链接，需建立链接
3. $ git branch --set-upstream dev origin/dev设置本地分支与远程分支的链接
4. $ git pull再次抓取提交（成功） 冲突部分呈现在本地文档，需手动解决
5. add commit push 完成添加、提交、推送

**注：当步骤二成功时则无需第三四步**

