**1.Git本地操作**

桌面右键点击Git \_Bash here



**1． 配置基本信息**

设置用户名

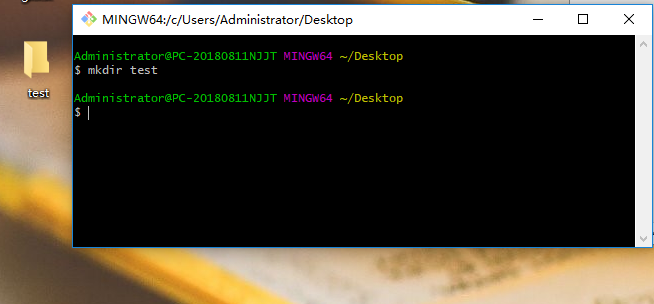
git config --global user.name "你的用户名"

设置注册github的邮箱

git config --global user.email "你的邮箱"

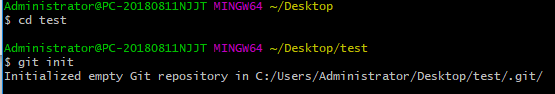
**2． 如何创建在本地创建一个git仓库**

mkdir 仓库名

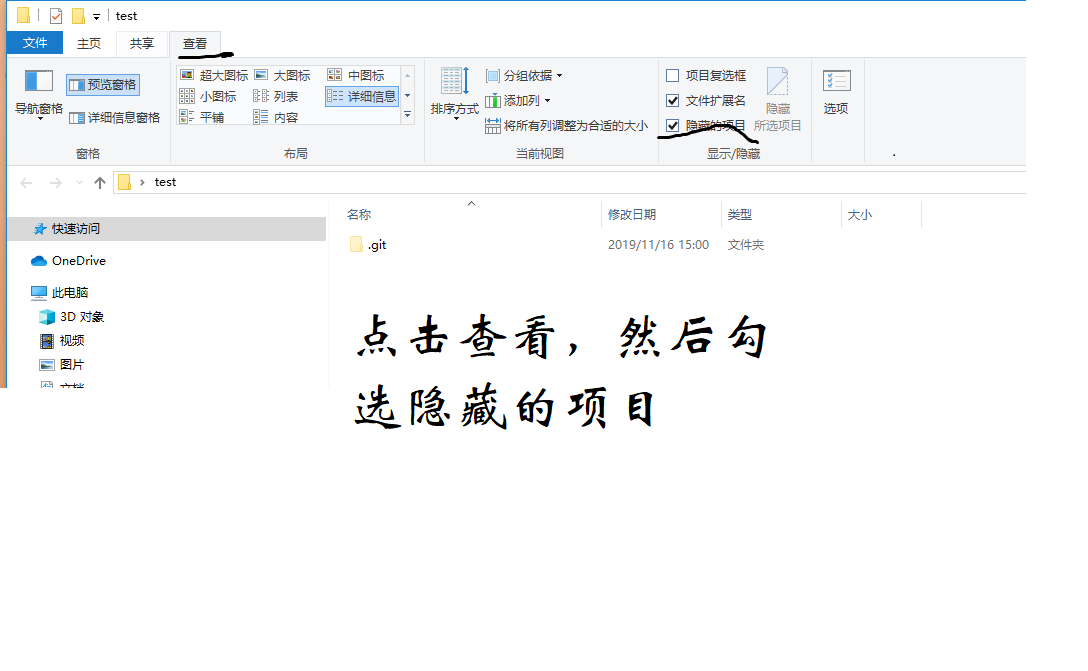


cd 目录名 进入目录

git init 初始化.git目录



.git目录默认隐藏



注意：仓库名和各目录名不要中文，避免一些编码上的错误

创建文件和提交文件到仓库

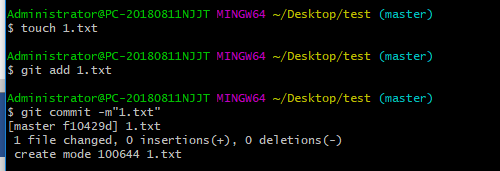
touch 文件名 创建文件

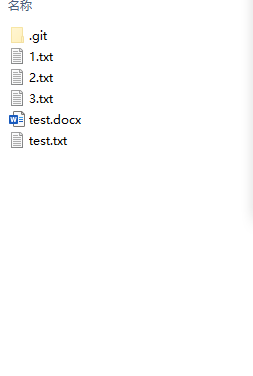
git add 文件名 将文件添加到暂存区

git add 文件名1 文件名2 文件名3 将多个文件添加到暂存区

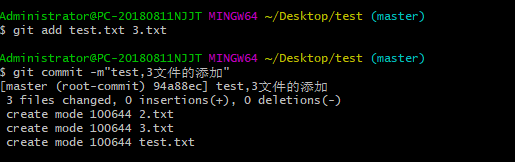
git add . 添加当前目录下的文件到缓存区，注意有空格在.前面

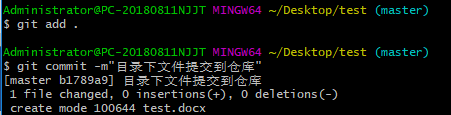
git commit -m"注释" 提交到本地仓库



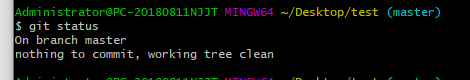


多文件添加



修改仓库文件

git status 查看缓存区情况

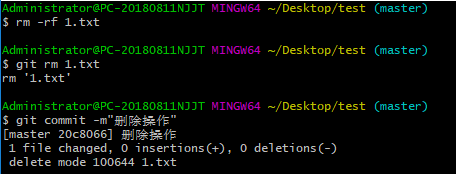


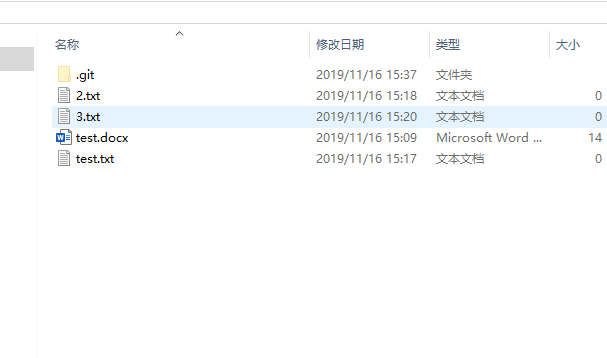
**3. 删除仓库文件**

rm -rf 文件名 通知要删除某文件

git rm 文件名 删除缓存区的该文件

git commit -m"注释" 将仓库文件删除



****

**4. 版本回退**

查看版本，确定回退时刻

git log 查看当前版本的提交历史

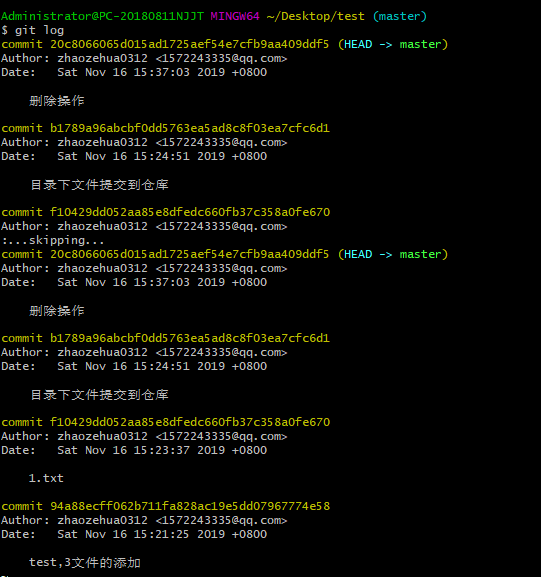
git log --pretty=oneline 以一行的形式查看历史记录,建议用这种方式

git reflog 查看所有提交历史

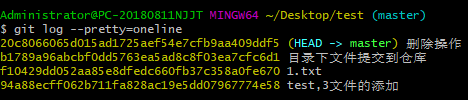
回退指令

git reset --hard commit id

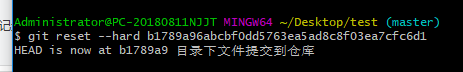
git log



git log --pretty=oneline



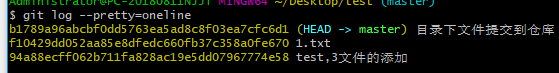
回退到删除操作之前：1.txt恢复





回退到删除操作时间点：无法使用git log,因为此时的

版本是删除之前的版本，不存在删除的历史记录



要用git reflog



**5. 分支的创建，合并，删除**

git branch 查看分支

git branch 分支名 创建分支

git checkout 分支名 切换分支

git merge 被合并分支名 合并分支

git branch -d 删除分支

注意在删除分支时要退出该分支

**2.远程仓库的操作**

**1.更新远程仓库的内容**

1 git clone url 将远程仓库（github对应的项目）复制到本地

2 cd 目录路径

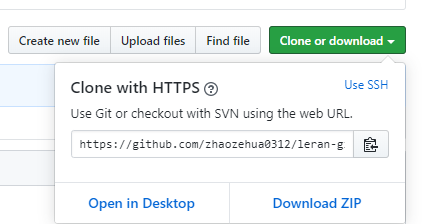
3 git add 文件名

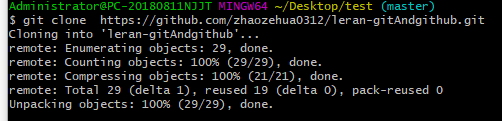
4 git commit -m"注释" //提交缓存区

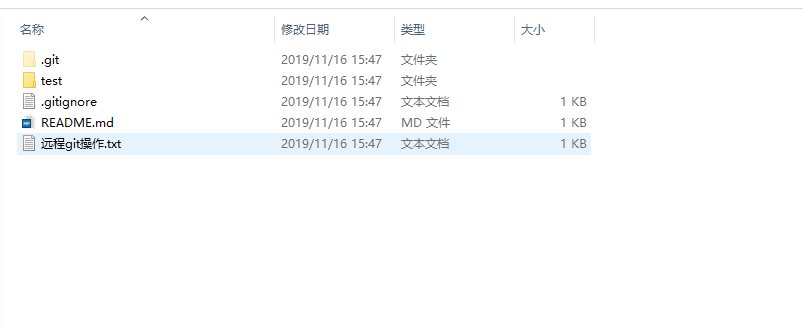
5 git push //注意要联网 和设置权限

//账号密码存在特殊字符，要转化为UrlEncode编码 如@ == %40

url的来源





添加文件到远程仓库

注意首次git push要配置url在.git目录下的

config文件中的url

原

url= <https://github.com/用户名/仓库名.git>

修改为：

url=https://用户名:密码@github.com/用户名/仓库名.git

Github上的远程仓库：



**2. 冲突：**

冲突的产生：本地仓库与线上不一致时未git pull就开始编写代码 并且在编写完代码后试图git push

Git 会报错如图

hint: 
hint: 
hint: 
hint: 
hint: 
Updates were rejected because the remote contains work that you do 
not have locally. This is usually caused by another repository pushin 
to the same ref. You may want to first integrate the remote changes 
git pull 
before pushing again. 
See the 'Note about fast-forwards' in 'git push 
-help' for details. 

解决冲突

先git pull，冲突部分如图



接下来要与冲突的·开发者进行商量，决定冲突代码的留存

**3．忽略某些文件**

在push给远程仓库时需要忽略某些文件时，需要在本地仓库新建.gitignore的文件，注意在windows下无法直接创建该文件，只能通过Git Bash来touch .gitignore 来生gitignore文件

然后在.gitignore中书写规则如下：

常见规则写法

/mtk/ 过滤整个文件夹

\*.zip 过滤所有.zip文件

/mtk/do.java 过滤某个具体文件

!do.java 不过滤某个文件