# Git常用命令

git init 这个命令的意思是把这个目录变成git管理的仓库

git add 文件名 新增文件到暂缓区

git commit -m “注释信息” 提交文件到版本控制

git log 查看提交历史记录日志（日志：包括每次提交的版本号、f493d4b0dc32038c4b4822a9b8a6f386b4b23920、作者、日期）

git log --pretty=oneline 只显示日志：版本号和注释说明

git diff 文件名 查看修改记录（如果没有变就没有任何提示）

git reset --hard HEAD^ 当前版本退回到上一个版本

git reset --hard HEAD^^ 当前版本退回到上上个版本（^一次类推）

如果退回到前100个版本的话就使用：

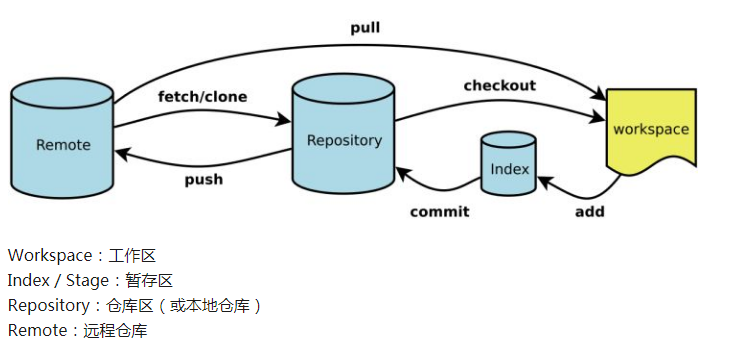
git reset --hard HEAD~100即可。

git reset --hard 版本号 通过指定版本号回退（如果记不住版本号可以使用git reflog来查看）

# 具体使用教程：

## git是什么？

* 1. Git是目前世界上最先进的分布式版本控制系统。
  2. 工作原理/流程：



## SVN与Git的最主要的区别？

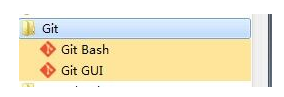
* 1. SVN是集中式版本控制系统，版本库是集中放在中央处理器中的，而干活的时候，用的都是自己的电脑，所以首先要从中央服务器哪里得到最新的版本，然后干活，干完后，需要把自己做完的活推送到中央服务器。集中式版本控制系统是必须联网才能工作，如果在局域网还可以，带宽够大，速度够快，如果在互联网下，如果网速慢的话，就尴尬了。
  2. Git 是分布式版本控制系统，那么它就没有中央服务器的，每个人的电脑就是一个完整的版本库，这样，工作的时候就不需要联网了，因为版本都是在自己的电脑上。既然每个人的电脑都有一个完整的版本库，那么多个人如何协作呢？比如说自己在电脑上改了文件A，其他人也在电脑上改了文件A，这时，你们两之间只需把各自的修改**推送**给对方，就可以互相看到对方的修改了。

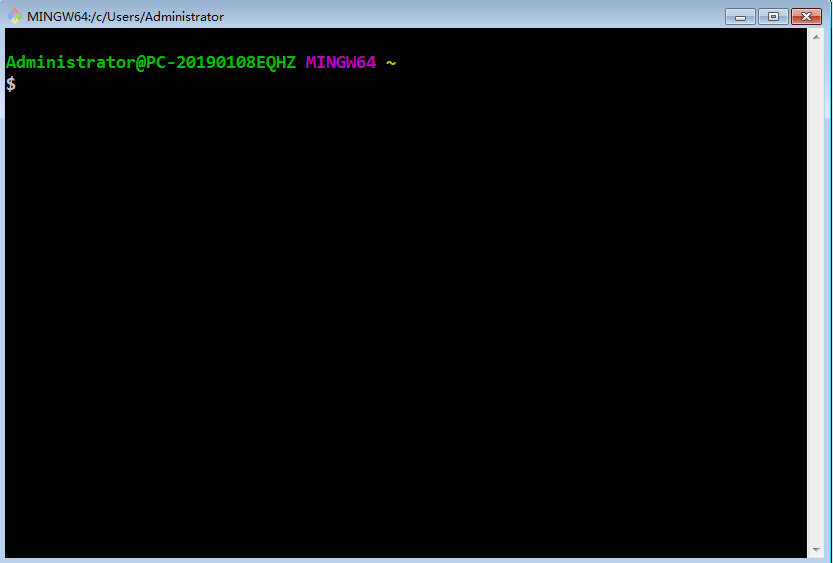
## 在windows上如何安装Git？

* 1. Msysgit 是windows版的git,如下：

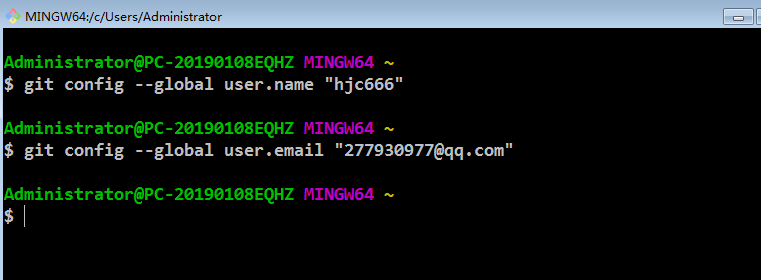


需要从网上下载一下，然后进行默认安装即可。安装完成后，在开始菜单里面找到“Git -->Git Bash”，如下：



会弹出一个类似的命令窗口的东西，就说明Git安装成功。如下  
 

安装完成后，还需要最后一步设置，在命令行输入如下



因为Git是分布式版本控制系统，所以需要填写用户名和邮箱作为一个标识。

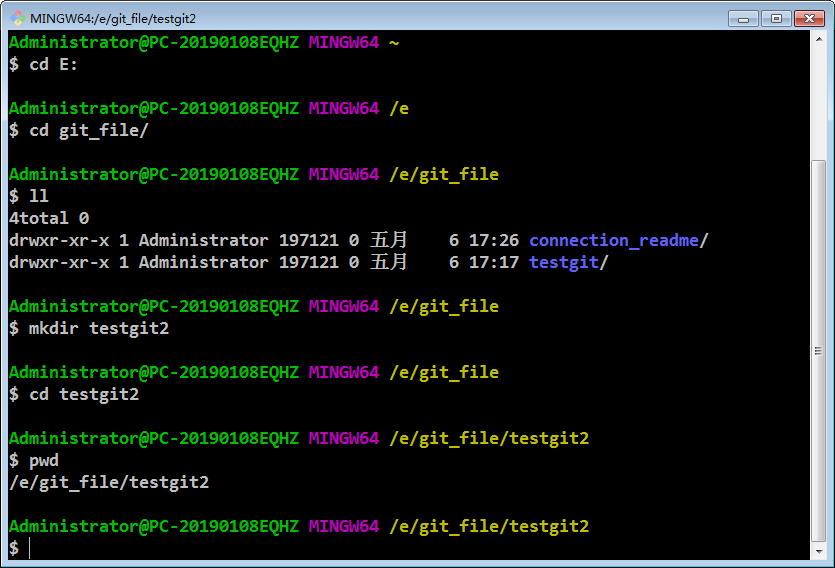
注意：git config --global参数，有了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置，当然你也可以对某个仓库指定不同的用户名和邮箱

## 如何操作

### 创建版本库。

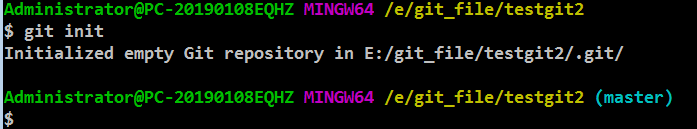
什么是版本库？版本库又名仓库，英文名repository，你可以简单的理解为一个目录，这个目录里面的所有文件都可以被Git管理起来，每个文件的修改，删除，Git都能跟踪，以便任何时刻都可以追踪历史，或者在将来某个时刻还可以将文件“还原”。

所以创建一个版本库也非常简单，如下我是E盘->git\_file下新建一个testgit2版本库

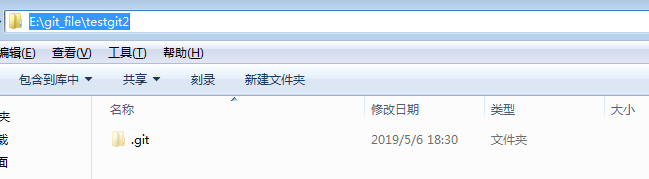


pwd命令是用于显示当前的目录。

1. 通过命令git init 把这个目录变成git可以管理的仓库，如下：



这时候你当前的testgit2目录下会多了一个.git的目录，这个目录是Git来跟踪管理版本的，没事千万不要手动乱改这个目录里面的文件，否则，会把git仓库给破坏了。如下

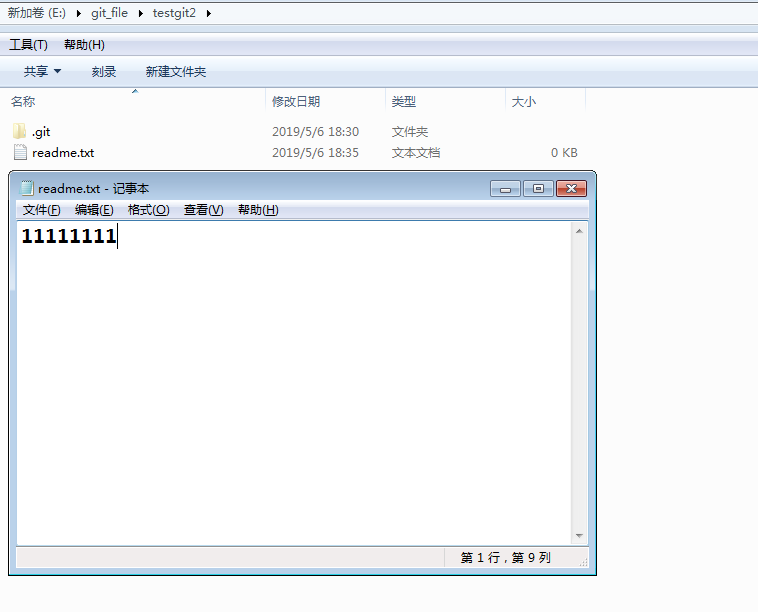


1. 把文件添加到版本库中。

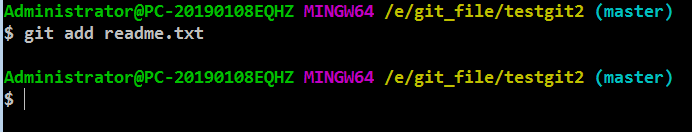
首先要明确下，所有的版本控制系统，只能跟踪**文本文件**的改动，比如txt文件，网页，所有程序的代码等，Git也不例外，版本控制系统可以告诉你每次的改动，但是**图片，视频这些二进制文件**，虽也能由版本控制系统管理，但没法跟踪文件的变化，只能把二进制文件每次改动串起来，也就是知道图片从1KB变成2KB，但是到底改了什么，版本控制也不知道。

### 下面先看下demo演示：

我再版本库testgit2目录下新建一个记事本文件readme.txt 内容如下：11111111

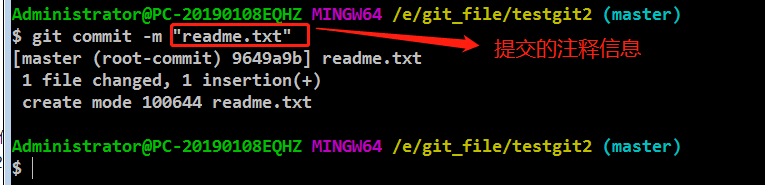


第一步：使用命令git add readme.txt**添加到暂存区**也叫（index索引库）如下：

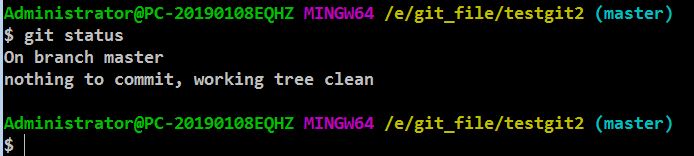


如果和上线一样，没有任何提示，说明已经添加成功了。

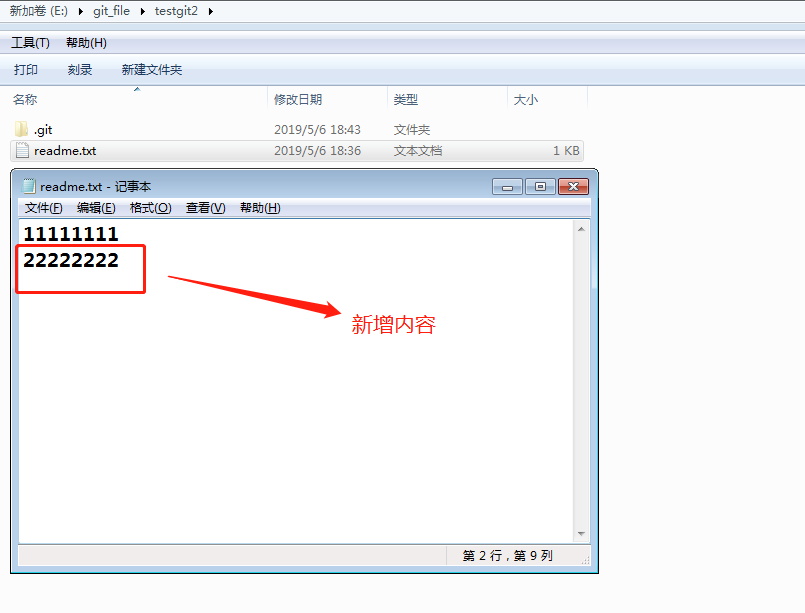
第二步：用命令git commit 告诉Git，**把文件提交到仓库**。

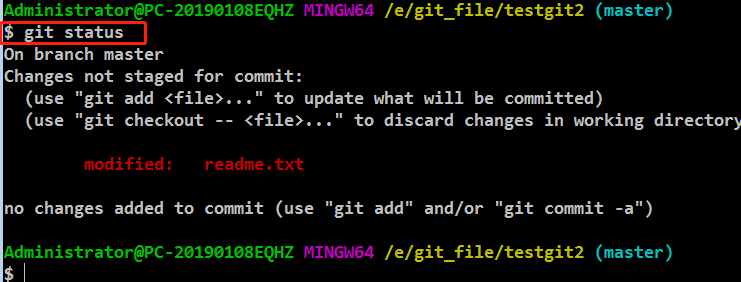


现在我们已经提交了一个readme.txt文件，我们下面可以通过命令git status来**查看是否还有文件未提交**，如下：



说明没有任何文件未提交，现在我们继续来改下readme.txt内容，比如我们在下面添加一行内容为：22222222，添加保存后继续使用git status来查看下结果，如下：

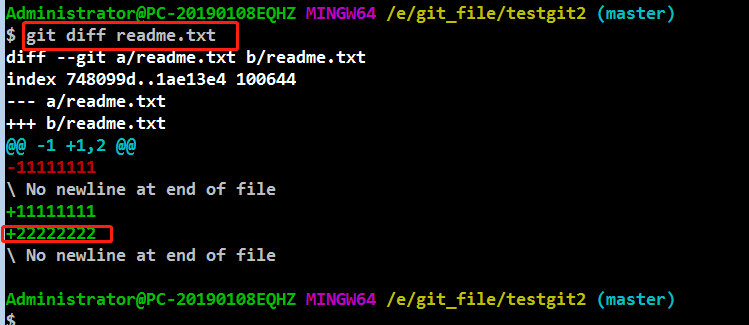




上面的命令告诉我们readme文件已被修改，但是是一个未被提交的修改。

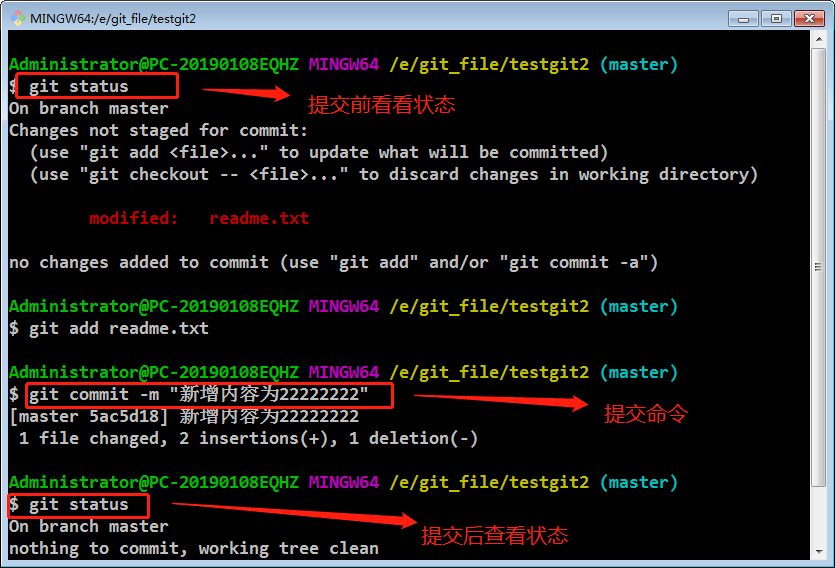
接下来我想看下readme.txt文件到底改了什么内容，如何查看呢？可以使用如下的命令

git diff readme.txt 如下：



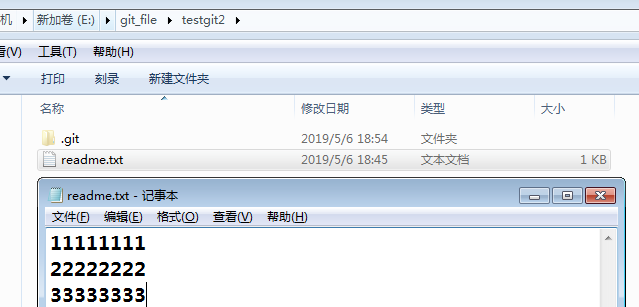
如上可看到，readme.txt文件内容从一行的：11111111，修改成二行新添加了一行：22222222的内容。

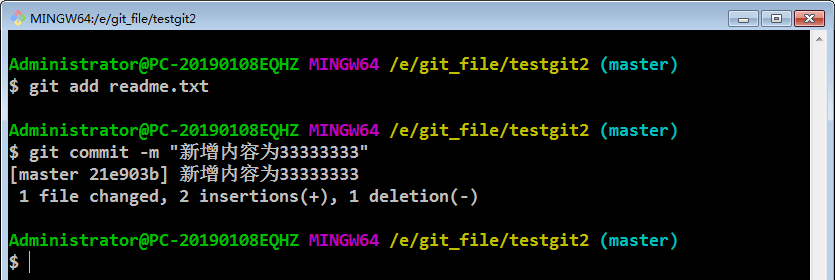
知道了对readme.txt文件做了什么修改后，我们可以放心的提交到仓库了，提交修改和提交文件是一样的两步：（第一步是git add 文件名 、第二步是：git commit -m 注释信息）。



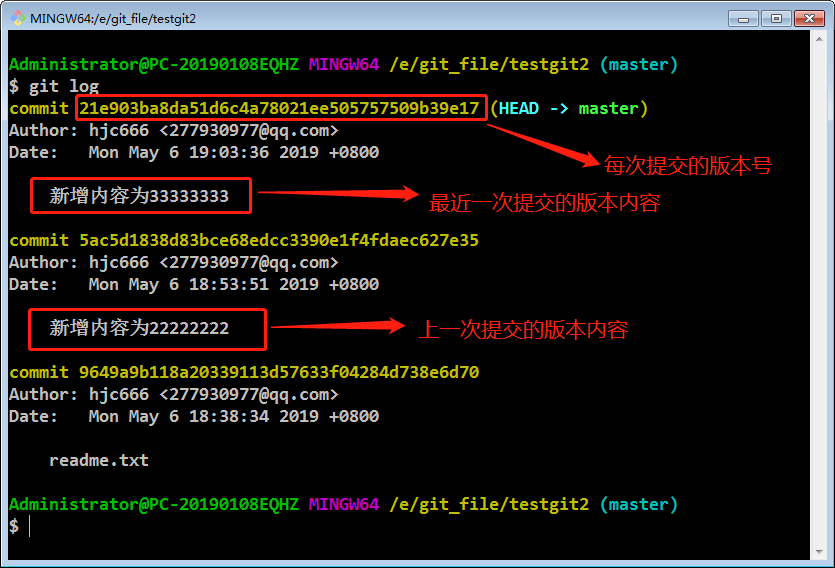
## 版本退回

* 1. 如上，我们已经学会了修改文件，现在我们继续对readme.txt文件进行修改，再增加一行内容为：33333333，继续执行命令如下：





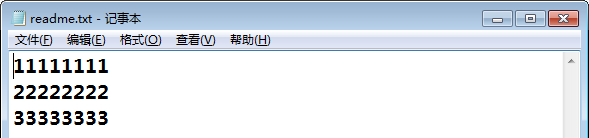
现在我们已经对这个readme.txt文件修改了三次了，那么我们现在想查看下历史记录，如何查呢？我们现在可以使用命令git log 演示如下：



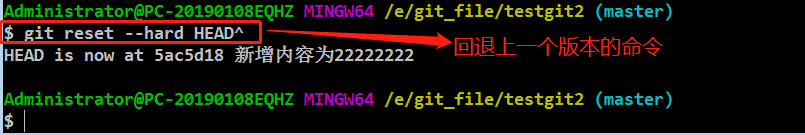
git log 命令显示从最近最远的显示日志，我们可以看到最近三次提交，最近的一次是，增加内容为33333333，上一次是增加内容为22222222，第一次默认是11111111，如果你觉得上面的信息太多的话，我们可以使用命令git log -pretty=oneline 演示如下：



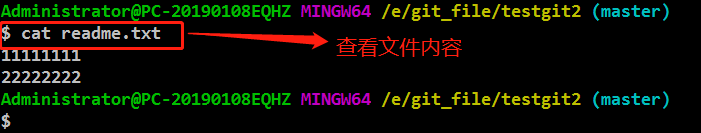
现在我想使用版本回退操作，我想把当前的版本回退到上一个版本，要使用说明命令呢？可以使用如下2种命令，第一种是：git reset --hard HEAD^ 那么如果要回退到上上个版本呢？只需要把HEAD^改成HEAD^^以此类推。那如果要回退到前100个版本的话，使用上面的方法肯定不方便，我们可以使用下面的简便命令操作：git reset --hard HEAD~100即可。未回退之前的readme.txt内容如下：



如果想回到上一个版本的命令如下操作：



再来查看一下readme.txt内容如下：通过命令cat readme.txt查看

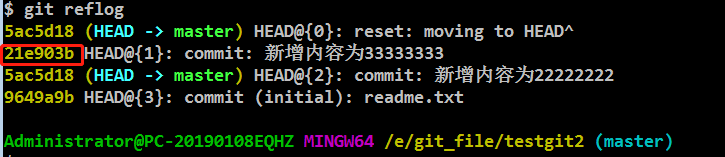


可以看到内容已经回退到上一个版本了。我们可以继续使用git log 来查看一下历史记录消息，如下

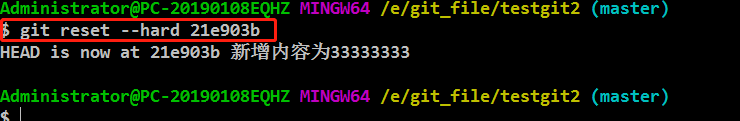


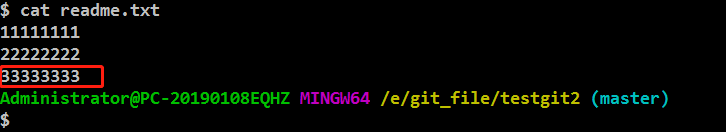
我们看到 增加的内容为3333333的内容我们没有看到了，但是现在我想回退到最新版本，如有：33333333的内容，要如何恢复呢？我们可以通过版本号回退，使用命令如下：

git reset --hard 版本号，但是现在的问题假如我已经关掉过一次命令行或者3333的内容版本号我并不知道没有做记录就回退了呢？要如何知道增加3333内容的版本号呢？可以通过如下命令即可获取到版本号：git reflog 如下



通过上面的显示我们可以知道，增加内容3333的版本号是21e903b。我们现在可以命令git reset --hard 21e903b来恢复了。演示如下





可以看到目前已经是最新版本的了。

## 三：理解工作区与暂存区的区别？

工作区：就是你在电脑上看到的目录，比如目录下testgit里的文件（.git隐藏目录版本库除外）。或者以后需要再新建的目录文件等等都属于工作区范畴。

版本库（Repository）：工作区有一个隐藏目录.git，这个不属于工作区，这是版本库。其中版本库里面存了很多东西，其中最重要的就是stage（暂存区），还有Git为我们自动创建了第一个分支master，以及指向master的一个指针HEAD。

我们前面说过使用Git提交文件到版本库有两步：

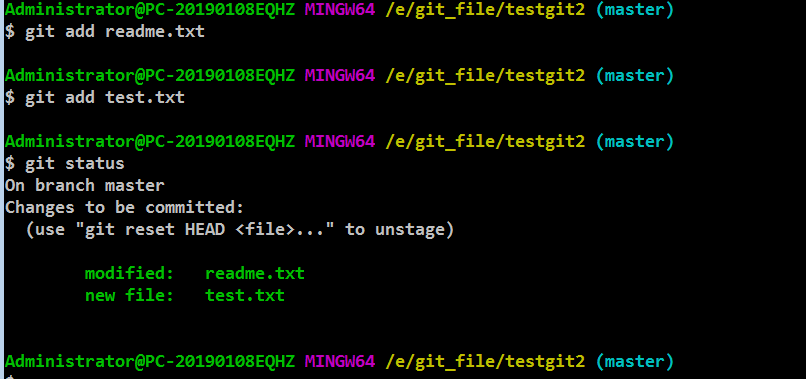
第一步：是使用git add把文件添加进去，实际上就是把文件添加到暂存区（也有叫索引库的）

第二步：使用git commit提交更改，**实际上就是把暂存区的所有内容提交到当前分支上。  
我们继续使用demo来演示一下：**

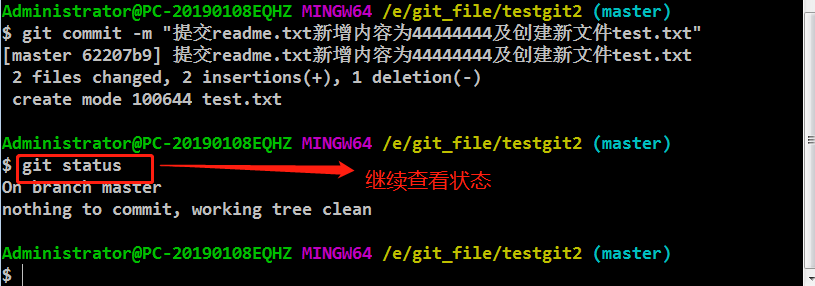
**我们在readme.txt再添加一行内容为44444444，接着再目录下新建一个文件为test.txt内容为test，我们先用命令git status来查看下状态，如下：**



现在我们使用git add命令把2个文件都添加到暂存区中，再使用git satus来查看下状态：



接着我们可以使用git commit一次性提交到分支上，如下



## 四：Git撤销修改和删除文件操作。

一：撤销修改

比如：我们现在在readme.txt文件里增加一行内容为：55555555，我们先通过命令查看如下：

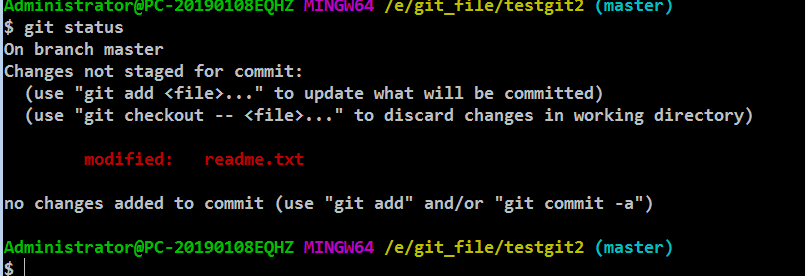


比如在我未提交之前我发现添加的内容：55555555有错，所以我得马上恢复以前的版本，现在我可以有如下几种方法可以做修改：

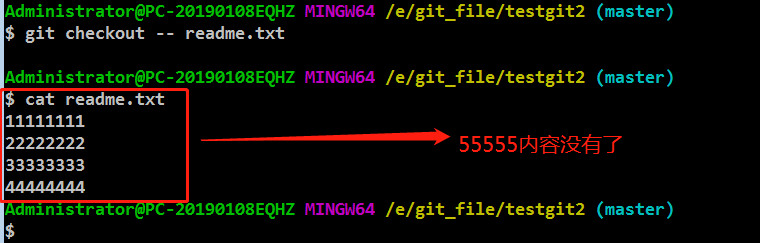
第一：如果我知道要删掉哪些内容的话，直接手动更改去掉哪些需要的文件，然后add添加到暂存区，最后commit掉。

第二：我可以按以前的方法直接恢复到上一个版本。使用 git reset --hard HEAD^或者通过git reflog查看版本号通过git reset --hard 版本号来恢复。

但是现在我不想使用上面的两种方法，我想直接使用撤销命令该如何操作呢？首先在做撤销之前，我们可以用git status查看下当前文件的状态，如下：



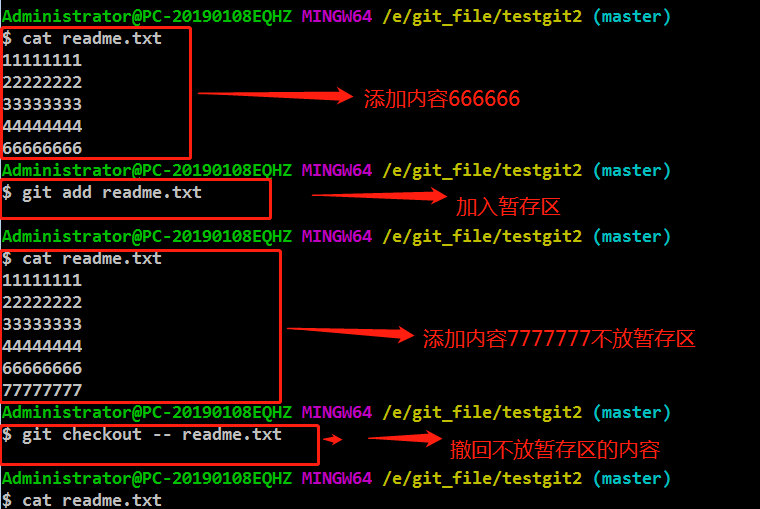
可以发现，Git会告诉你，git chckout -- file（文件名） 可以丢弃工作区的修改



命令git checkout --readme.txt意思就是，把readme.txt文件在工作区做的修改全部撤销，这里有2种情况，如下：

1. readme.txt自动修改后，还没有放到暂存区，使用撤销修改就回到和版本库一模一样的状态。
2. 另外一种是readme.txt已经放入暂存区了，接着又作了修改，撤销修改就回到添加暂存区后的状态。

对于第二种情况，我想我们继续做demo来看下，假如现在我对readme.txt添加一行内容为：666666，我git add增加到暂存区后，接着添加内容7777777，我想通过撤销命令让其回到暂存区后的状态如下所示：

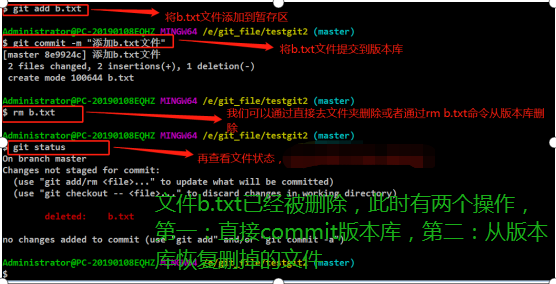




注意：命令git checkout -- readme.txt中的 -- 很重要，如果没有 -- 的话，那么命令变成创建分支了。

## 二：删除文件

假如我现在版本库testgit2目录添加一个文件b.txt然后提交。如下：

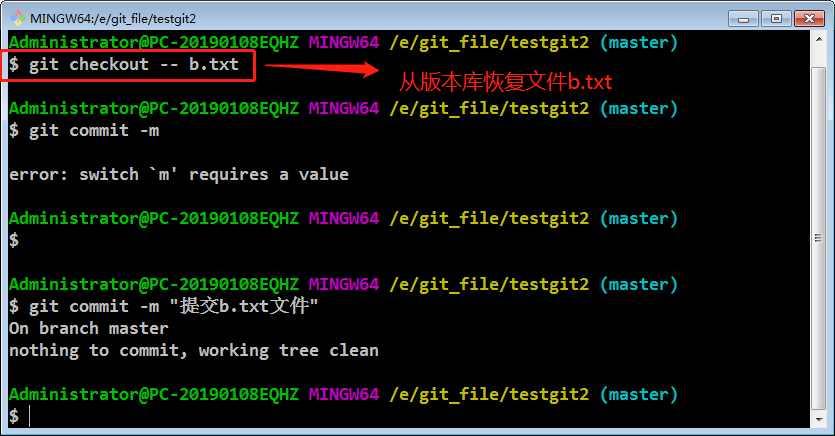


如上：一般情况下，可以直接在文件目录中把文件删了，厚泽使用如上rm命令：rm b.txt，如果我想彻底从版本库中删掉了此文件的话，可以再执行commit命令，提交掉，现在目录是这样的，



只要没有commit之前，如果我想在版本库中恢复次文件如何操作呢？

可以使用如下命令git checkout -- b.txt，如下所示：



再来看看我们testgit2目录，添加了b.txt文件了如下所示：



## 五：远程仓库

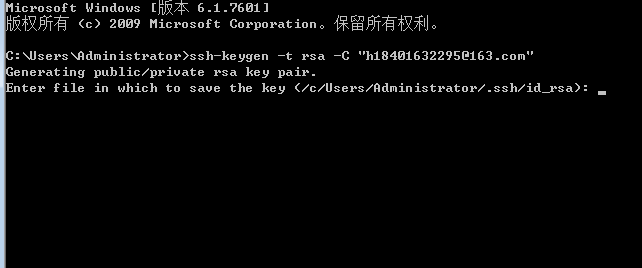
在了解之前，先注册github账号，由于你的本地Git仓库和github仓库之间的传输是通过SSH加密的所以需要一点设置：

第一步：创建SSH Key。在用户主目录下，看看有没有.ssh目录，如果有，再看看这个目录下哟没有id\_rsa和id\_rsa.pub这两个文件，如果有的话，直接跳过此。如下命令，如果有没有的话，打开命令行，输入如下命令：

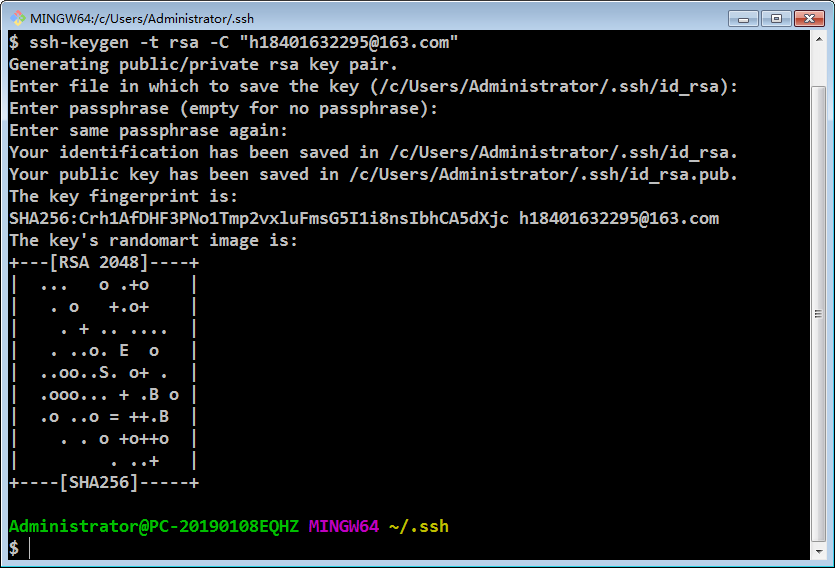
ssh-keygen -t rsa -C [“h18401632295@163.com”](mailto:\“h18401632295@163.com\”) ，如果出现不是内外部命令

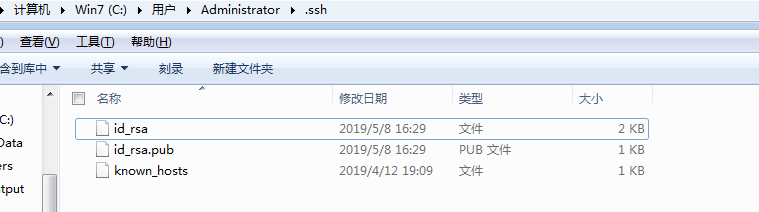


再次执行ssh-keygen -t rsa -C [“h18401632295@163.com”](mailto:\“h18401632295@163.com\”)



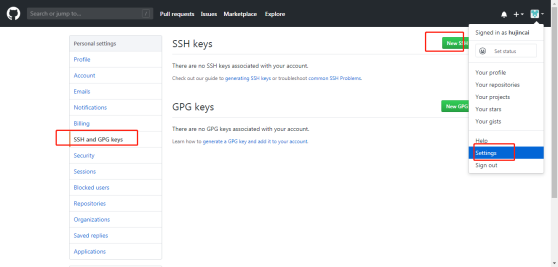
三个回车

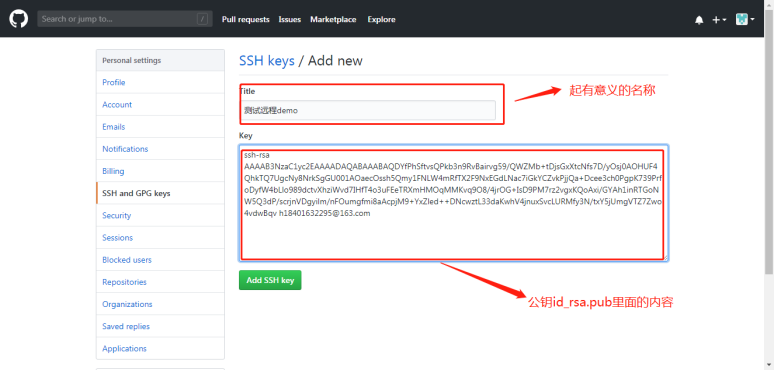




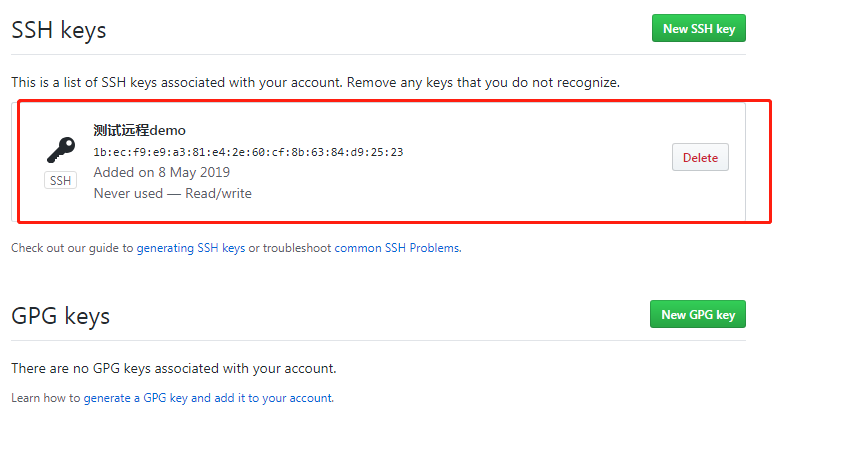
Id\_rsa是私钥，不能泄露出去，id\_rsa.pub是公钥，可以放心的告诉任何人。

第二步：登录github，打开”settings” 中的SSH and GPG keys 页面，然后点击 “NEW SSH”,填上任意title，在key文本框里粘贴id\_rsa.pub文件的内容。





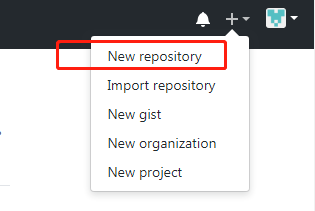
点击Add key，你就应该可以看到已经添加的key。

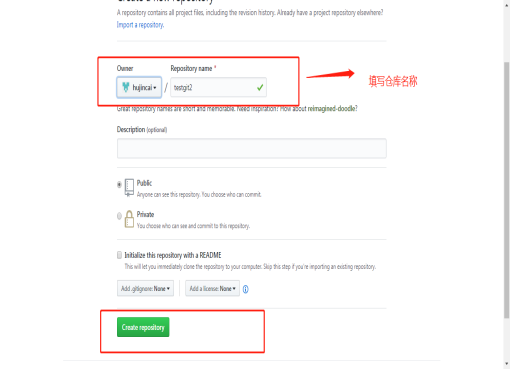


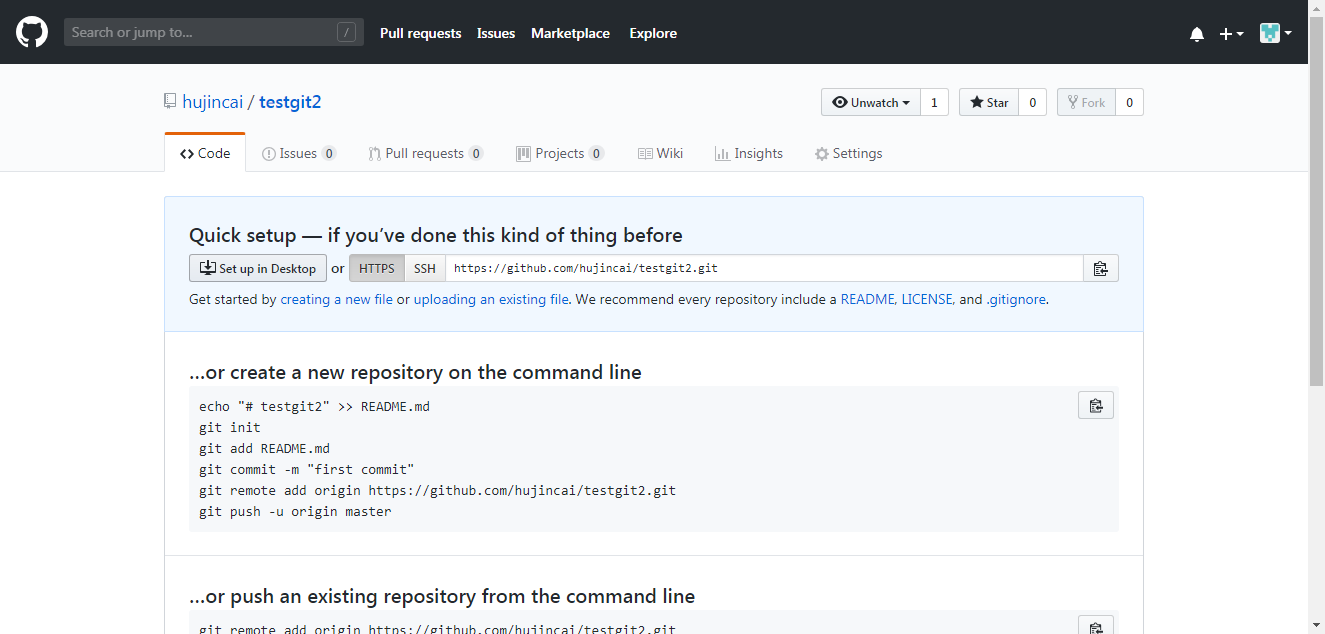
如何添加远程库？

现在的情景是：我们已经在本地创建了一个Git仓库后，又想在github创建一个Git仓库，并且希望这两个仓库进行远程同步，这样github的仓库可以作为备份，又可以其他人通过该仓库来协作。

首先，登录github上，然后在右上角找到”New repository” 创建一个新的仓库。如下：





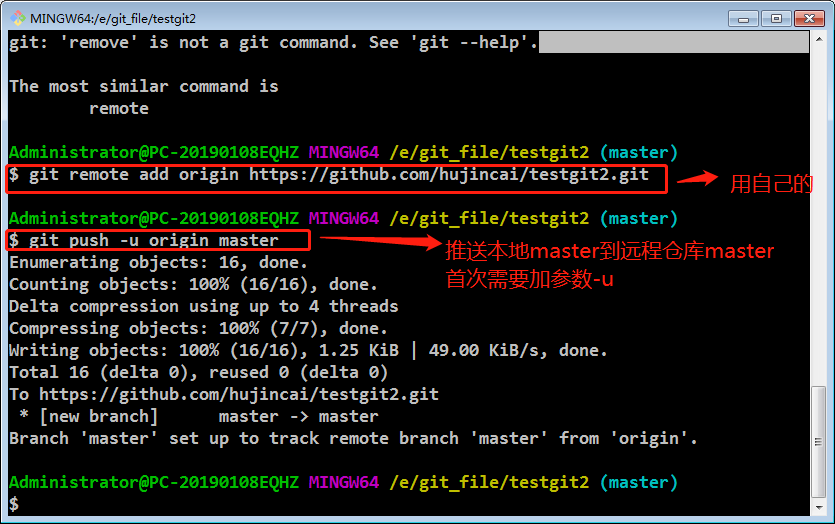


目前，在GitHub上的这个testgit2仓库还是空的，GitHub告诉我们，可以从这个仓库克隆出新的仓库，也可以把一个已有的本地仓库与之关联，然后，把本地仓库的内容推送到GitHub仓库。

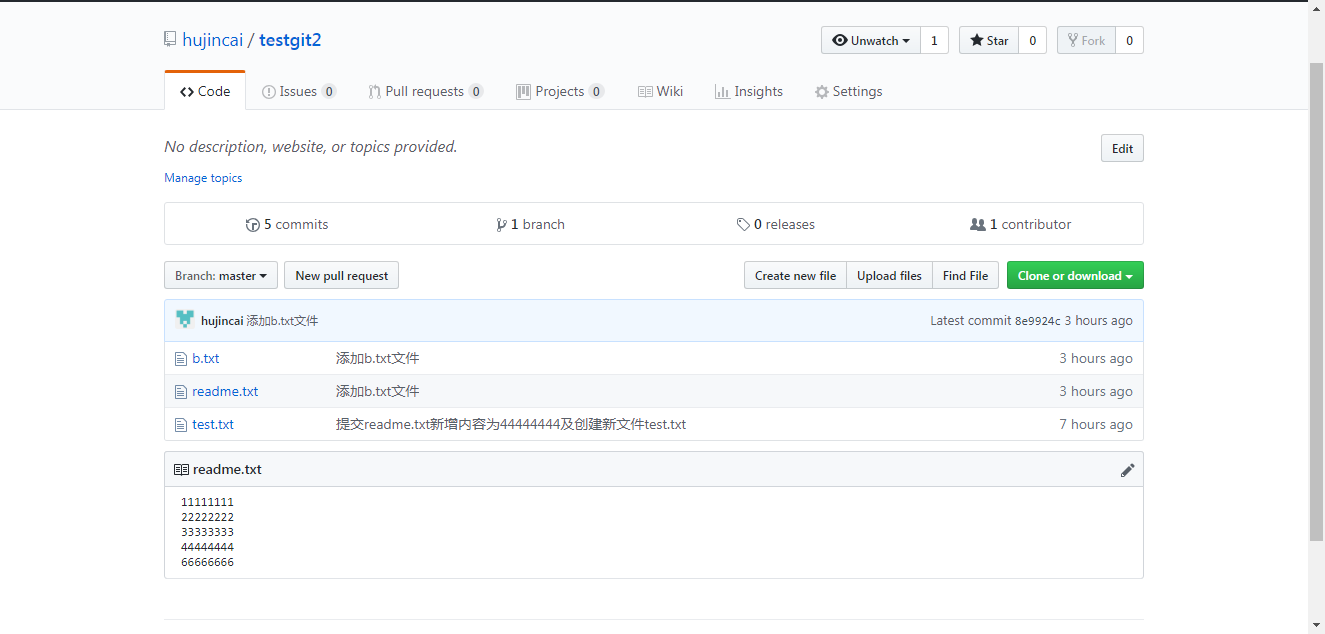
现在我们根据GithHub提示，在本地的testgit仓库下运行命令：



git remote add origin github.com/hujincai/testgit2.git



把本地库的内容推送到远程，使用git push命令，实际上是把当前分支master推送到远程。由于远程库是空的，我们第一次推送master分支时，加上了-u参数，Git不但会把本地的master分支内容推送到远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令了。推送成功后，可以立刻在github页面中看到远程库的内容已经和本地一模一样了，上面的要输入github的用户名和密码如下所示：



从现在起，只要本地作了提交，就可以通过如下命令：

git push origin master

把本地master分支的最新修改推送到github上了，现在你就拥有了真正的分布式版本库了。

## 错误：



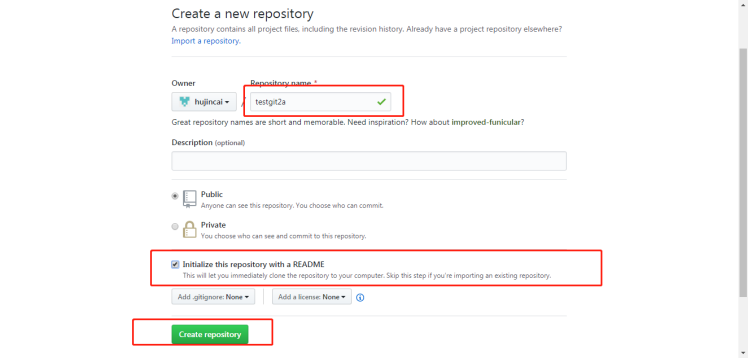
<https://blog.csdn.net/python_neophyte/article/details/83381936> 解决办法

1.如何从远程库克隆？

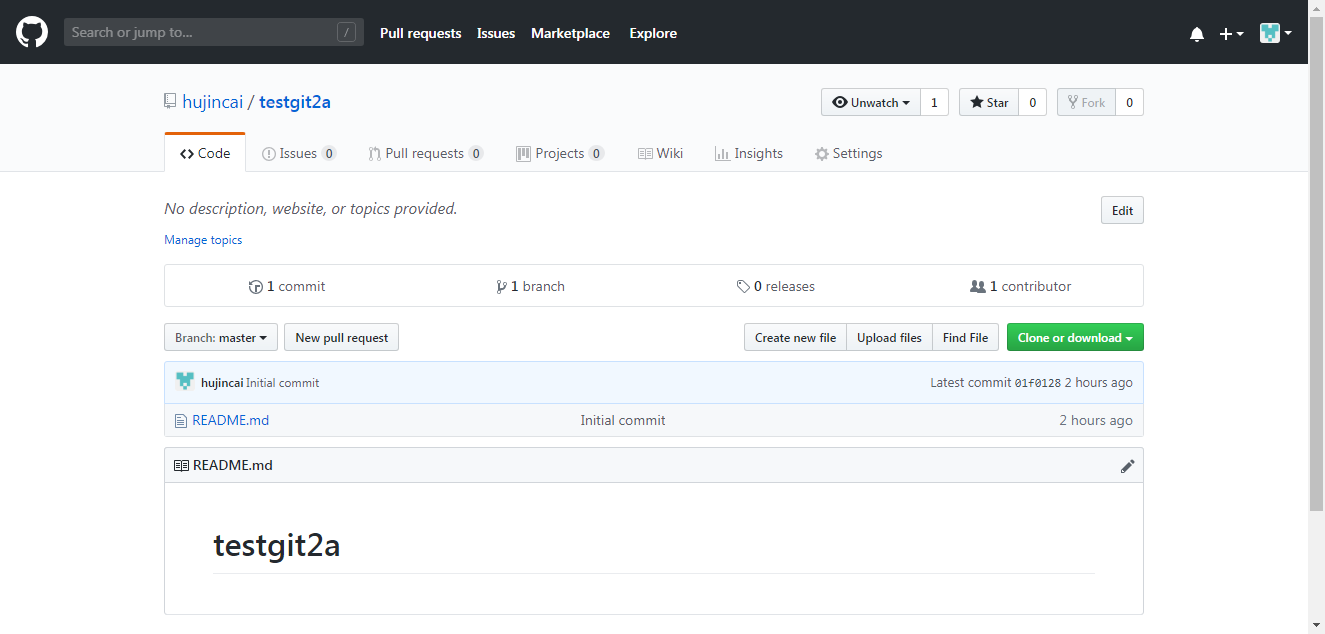
上面我们了解了先有本地库，后有远程库时候，如何关联远程库。

现在我们想，假如远程库有新的内容了，我想克隆到本地来 如何克隆呢？

首先，登录github，创建一个新的仓库，名字叫testgit2a.如下：

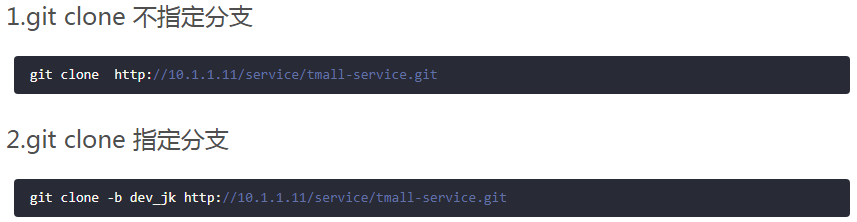


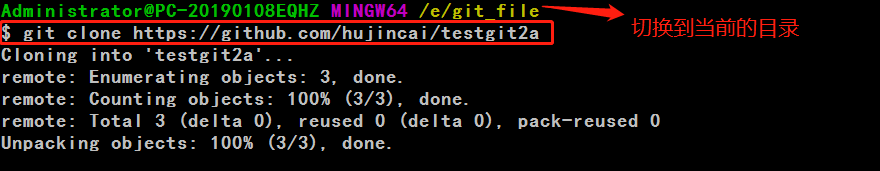
如果我们看到



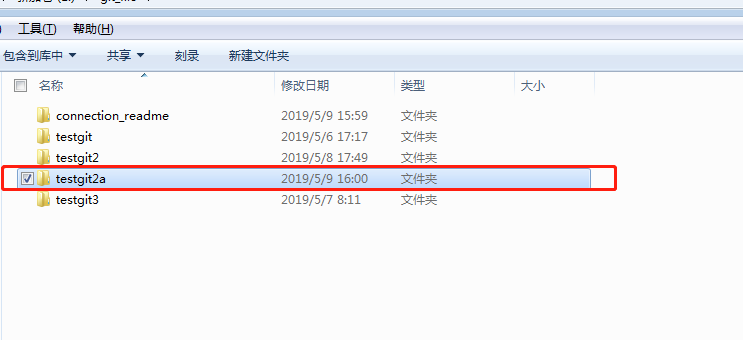
现在，远程库已经准备好了，下一步是使用命令git clone克隆一个本地库了。如下所示：

<https://blog.csdn.net/weixin_39800144/article/details/78205617> git clone指定分支 拉代码





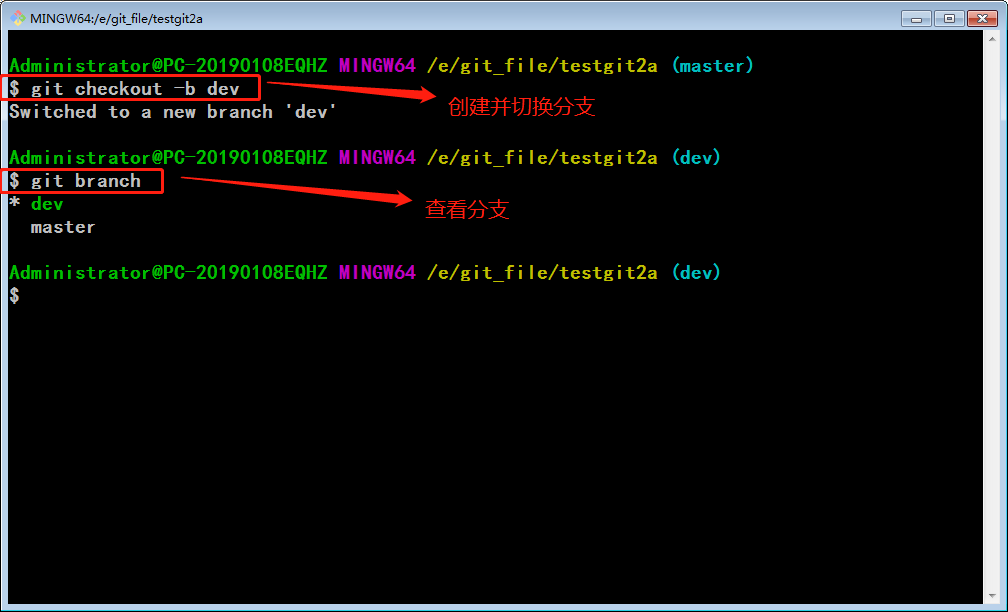
接着在我本地目录下 生成testgit2a目录了，如下所示：



## 六：创建与合并分支。

在 版本回填退里，你已经知道，每次提交，Git都把它们串成一条时间线，这条时间线就是一个分支。截止到目前，只有一条时间线，在Git里，这个分支叫主分支，即master分支。HEAD严格来说不是指定提交，而是指向master，master才是指向提交的，所以，HEAD指向的就是当前分支。

首先，我们来创建dev分支，然后切换到dev分支上。如下操作：



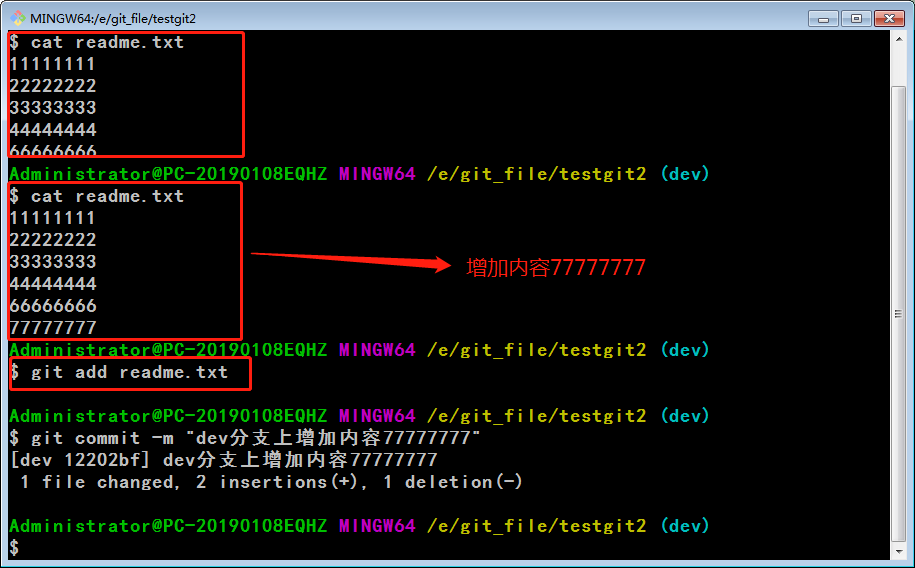
git checkout 命令加上 –b参数表示创建并切换，相当于如下2条命令：

git branch dev

git checkout dev

git branch查看分支，会列出所有的分支，当前分支前面会添加一个星号。然后我们在dev分支上继续做demo，比如我们现在在readme.txt再增加一行 77777777

首先我们先来查看下readme.txt内容，接着添加内容77777777，如下：



现在dev分支工作已完成，现在我们切换到主分支master上，继续查看readme.txt内容如下：



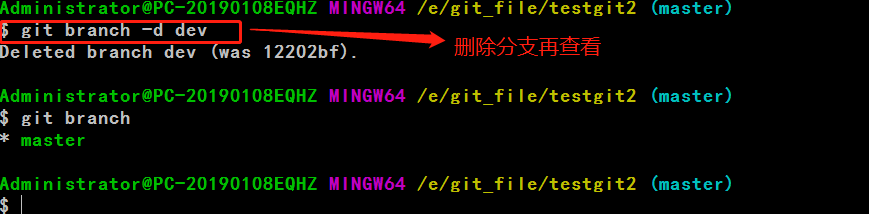
现在我们可以把dev分支上的内容合并到分支master上了，可以在master分支上，使用如下命令 git merge dev 如下所示：



git merge命令用于合并**指定分支**到当**前分支上，**合并后，再查看readme.txt内容，可以看到，和dev分支最新提交的是完全一样的。

注意到上面的Fast-forward信息，Git告诉我们，这次合并是“快进模式”，也就是直接把master指向dev的当前提交，所以合并速度非常快。

合并完成后，我们可以接着删除dev分支了，操作如下：



总结创建与合并分支命令如下：

查看分支：git branch

创建分支：git branch name

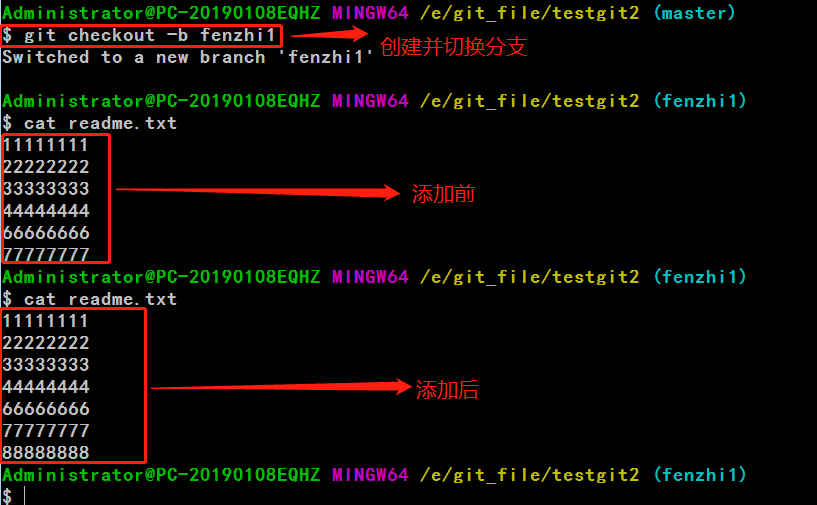
切换分支：git checkout name

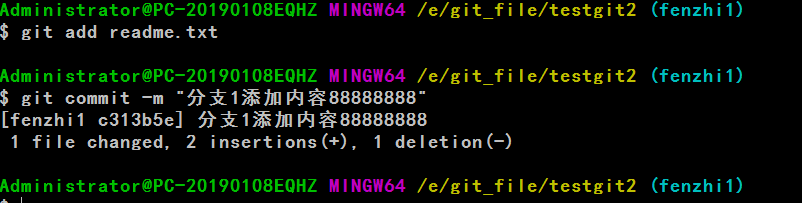
创建+切换分支：git checkout –b name

合并某分支到当前分支：git merge name

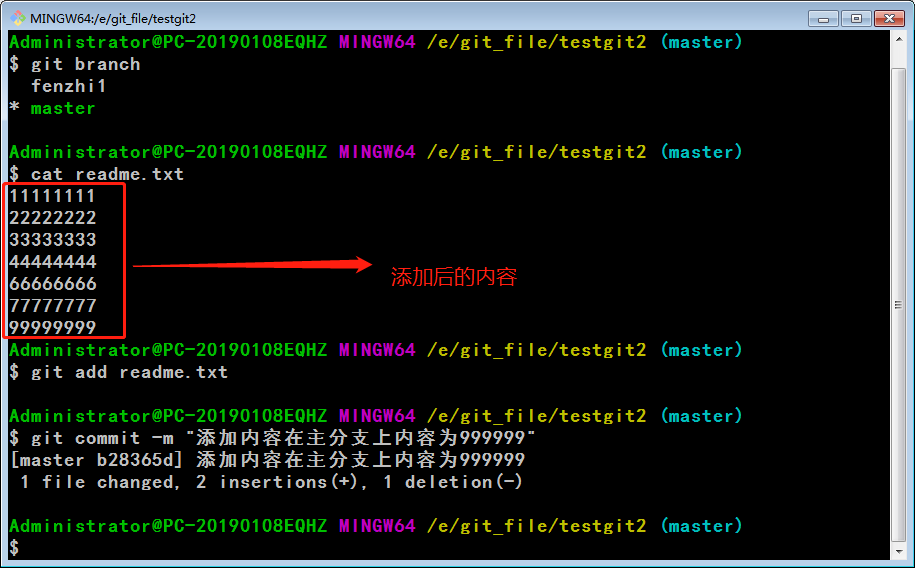
删除分支：git branch –d name

如何解决冲突？  
下面我们还是一步一步来，先新建一个新分支，比如名字叫fenzhi1，在readme.txt添加一行内容88888888，然后提交，如下所示：

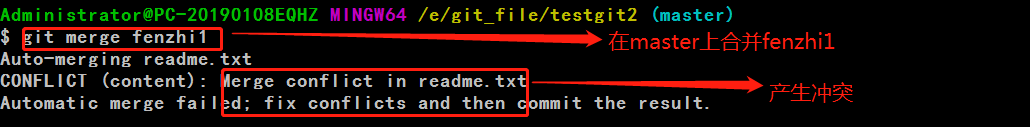




同样，我们现在切换到master分支上来，也在最后一行添加内容，内容为99999999，如下所示：



现在我们需要在master分支上来合并fenzhi1，如下操作：







Git用<<<<<<<，=======，>>>>>>>标记出不同分支的内容，其中<<<HEAD是指主分支修改的内容，>>>>>fenzhi1 是指fenzhi1上修改的内容，我们可以修改下如下后保存：