# Git初学者必备

序言：前期学习目标不明确，想一口吃成个胖子。导致研究的太细碎，处处碰壁。

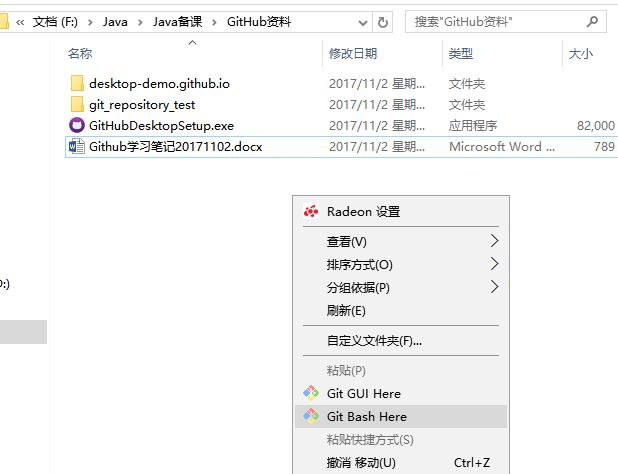
要明确学习Git的目的是“版本控制”，以此为主线。

## 常用命令

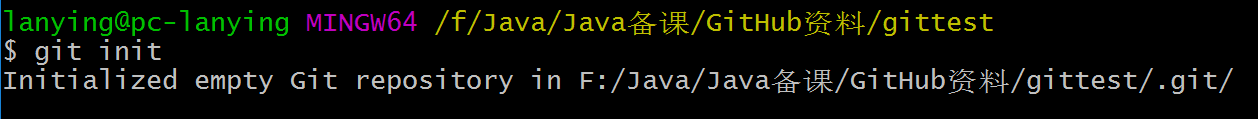
前置条件：安装Git Bash [下载地址](#_Git_Bash)

### 如何创建本地新仓库 git init

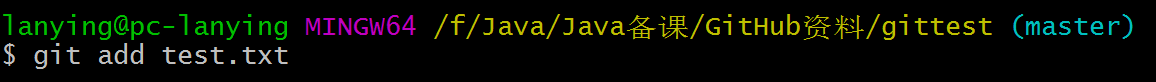
* 1. 在你需要的目录下右键打开 Git Bash Here



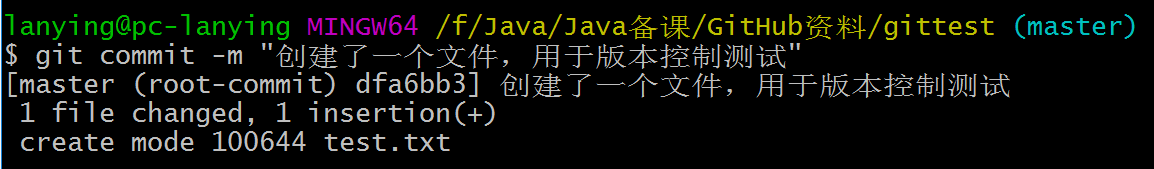
* 1. 然后使用 git init 命令创建本地仓库



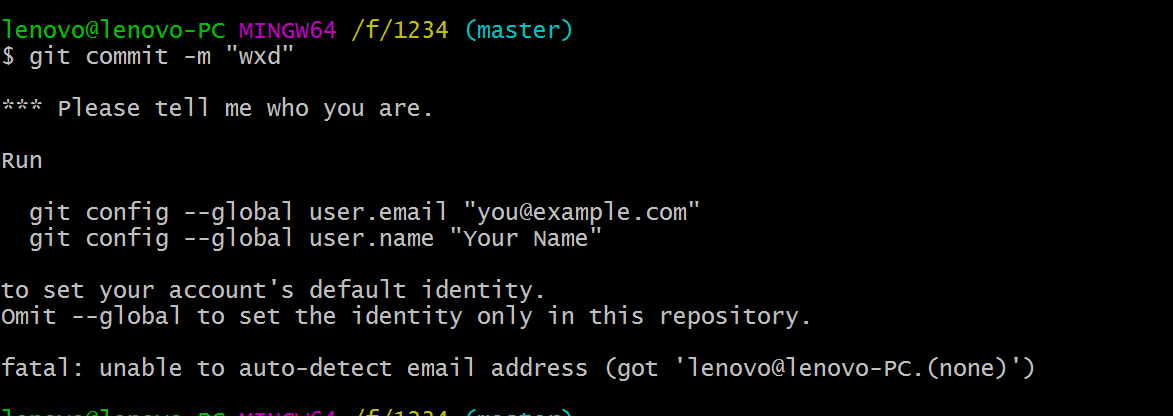
### 想要跟踪哪些文件 git add <filename> / git add \*



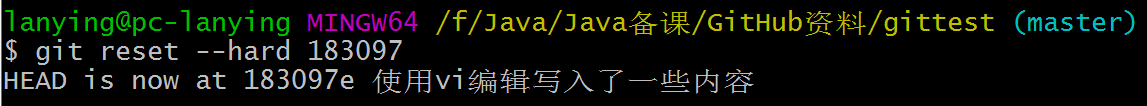
### 如何提交变化 git commit -m "代码提交信息" （提交前需要add）



【备注】第一次使用的用户会有如下提示，需要**配置**自己的邮箱和用户名



### 如何版本回退 git reset --hard 183097



【问题】--soft、[--mixed]、--hard

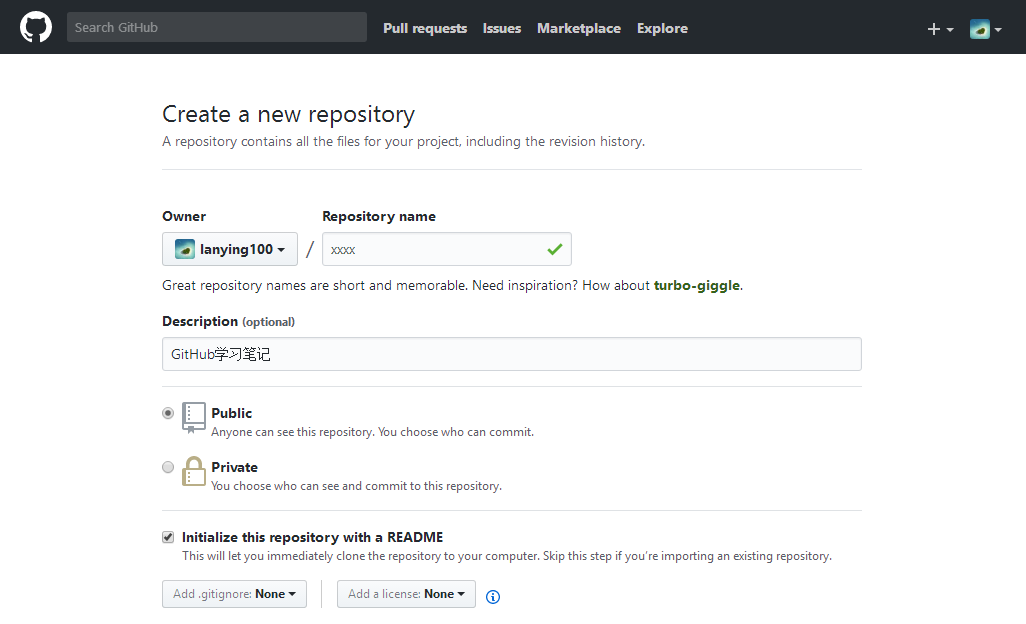
【参考网址】<https://www.cnblogs.com/kidsitcn/p/4513297.html>

### 结合GitHub（情况一：没有本地Git仓库）

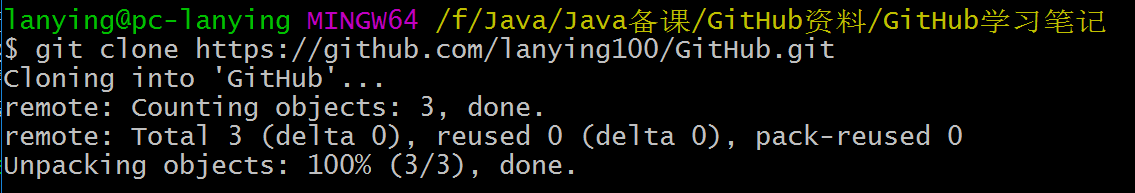
【官方网址】<https://github.com/>

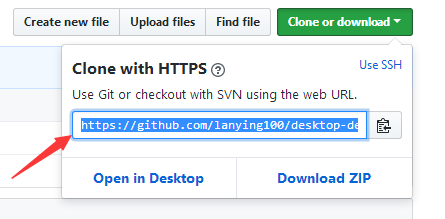
#### 创建本地的项目工程（Eclipse、Intellij IDEA等）

#### 在GitHub上创建远程仓库（不需要创建本地仓库）

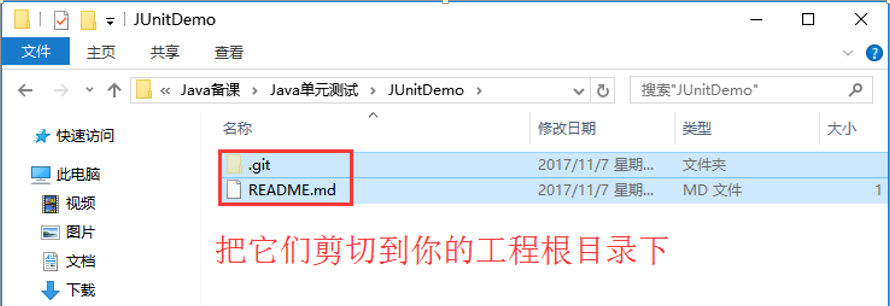


#### 在任意目录git clone远程仓库 git clone +xxxx





#### 将得到的隐藏.git文件及README.md等文件剪切到本地工程根目录下

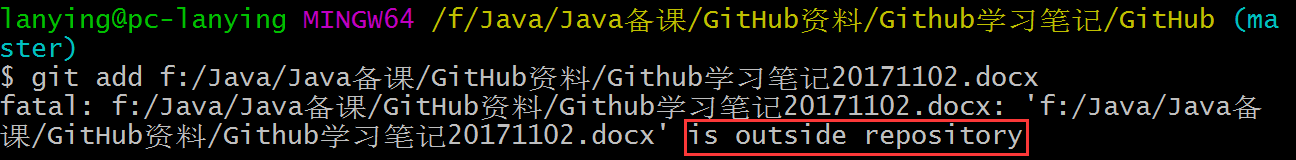


#### 将本地仓库push到GitHub上 git push origin master/自己的branch

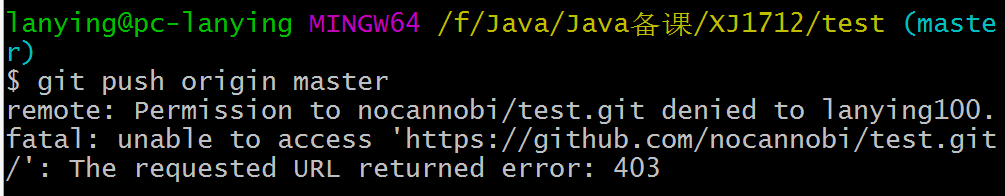
在工程根目录下打开Git Bash ，然后执行push命令



【PS】只能将仓库里的文件push出去。其它目录的文件无法push上去。



没有权限向其它用户的Github仓库提交数据



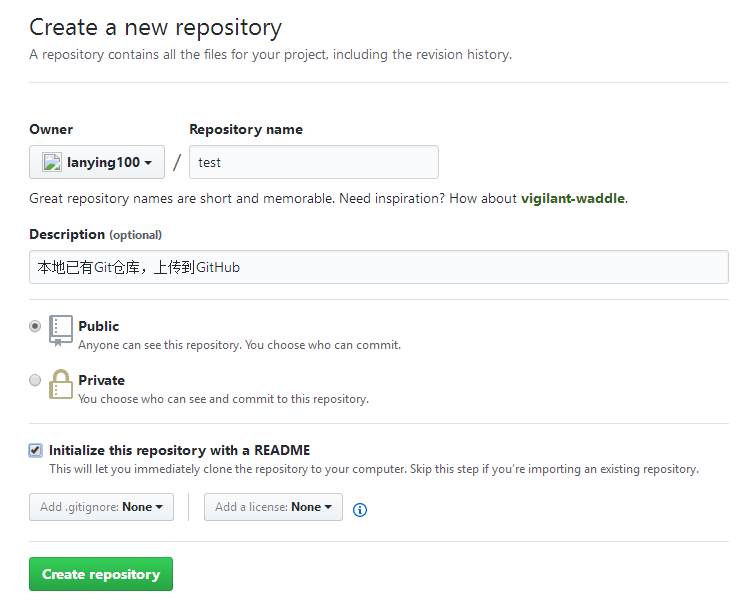
### 结合GitHub（情况二：已有本地Git仓库）

#### git init

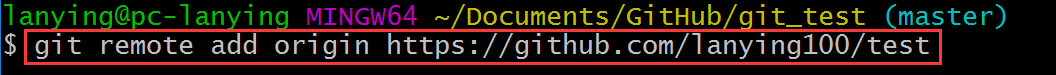
#### git add

#### git commit –m “xxxx”

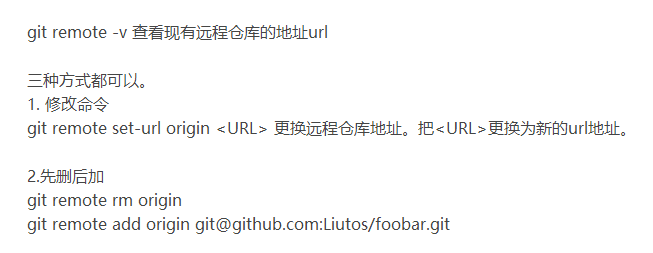
#### GitHub.com创建新仓库



#### 将本地Git仓库关联到GitHub上



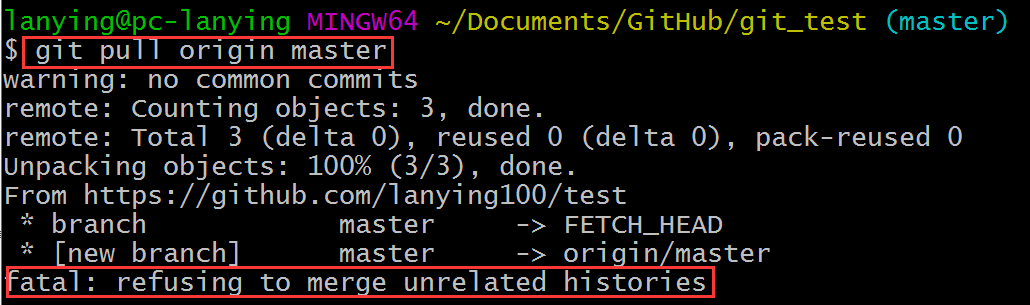
【备注】如果提示已经添加过了，则根据需要先删除旧的。



【参考网址】<http://blog.csdn.net/s0228g0228/article/details/45368155>

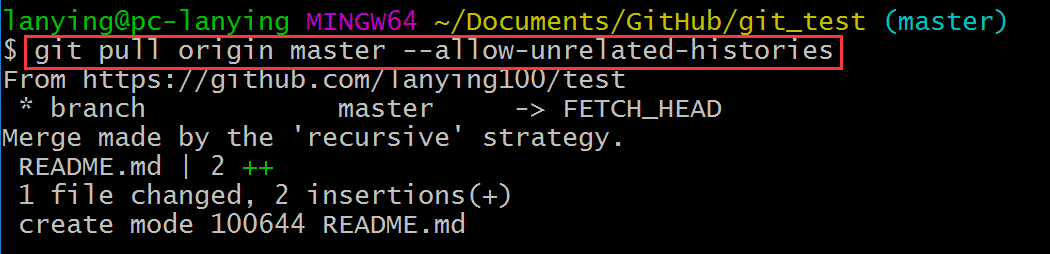
#### 将GitHub新仓库的master分支与本地的master分支合并

git pull origin master

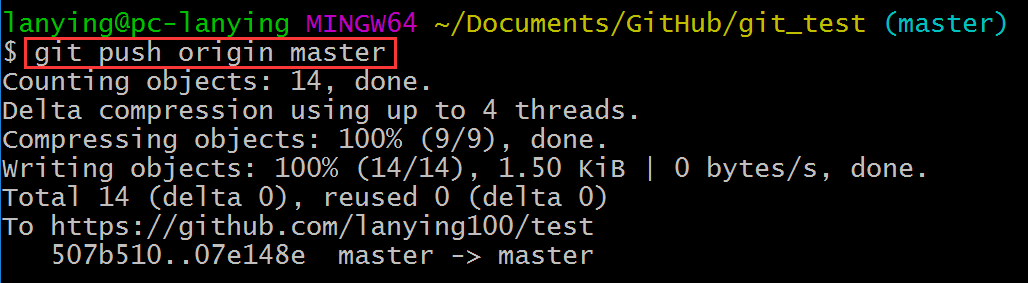


如果出现上述错误：“拒绝合并没有关联的历史”，则使用下面的命令

git pull origin master **--allow-unrelated-histories**



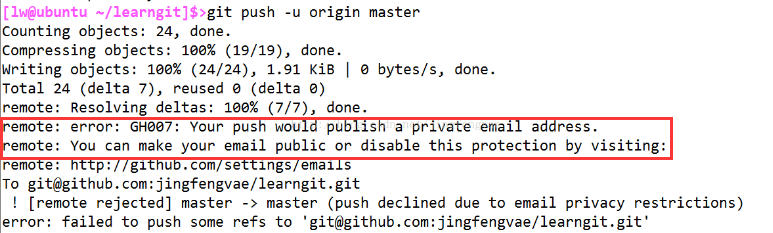
#### 推送本地仓库到GitHub



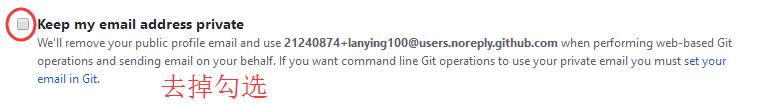
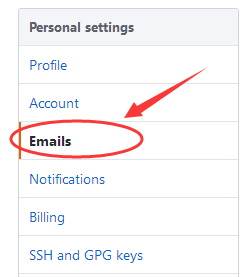
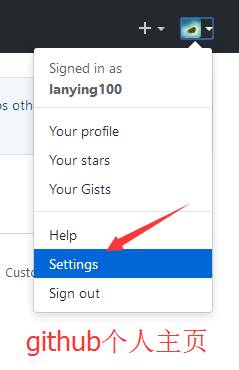
【学习网址】<http://blog.csdn.net/hanhailong726188/article/details/46738929>

【问题1】

remote: error: GH007: Your push would publish a private email address.



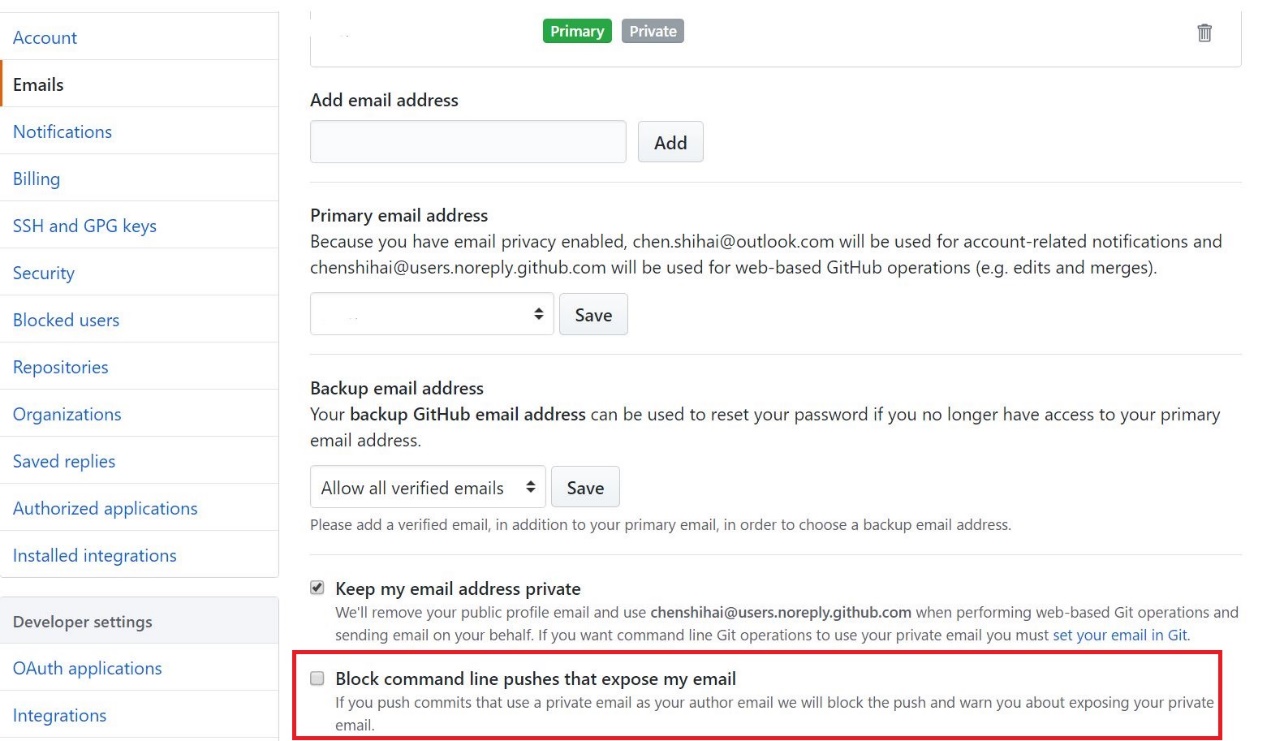
【解决方案】



【参考网址】<http://blog.csdn.net/jingfengvae/article/details/72859130>

【问题2】push declined due to email privacy restrictions

【解决方案】



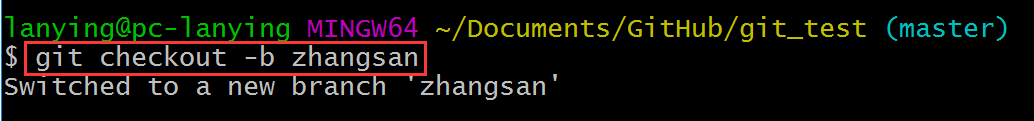
【参考网址】<https://www.jianshu.com/p/400dbf77f6cf>

### 分支命令

#### 创建分支（在当前分支下创建新分支）

git checkout –b <branchName>

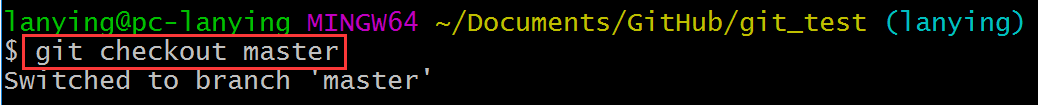
如下图，在master分支下创建zhangsan分支



#### 切换分支

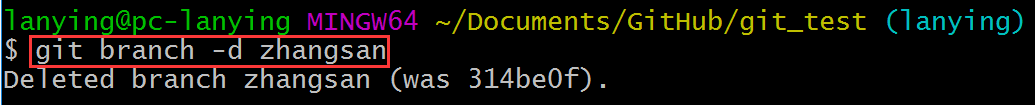
git checkout master

切换到master分支



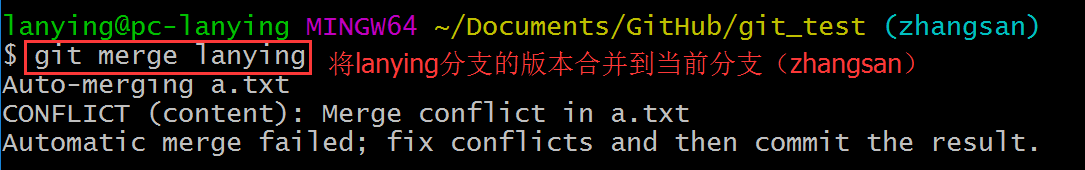
#### 删除分支

git branch –d zhangsan 删除zhangsan分支

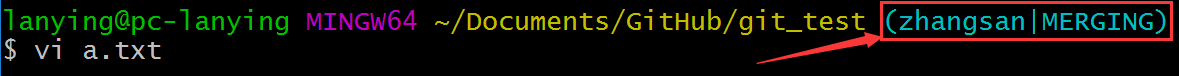


#### 合并分支

git merge lanying 将lanying分支合并到**当前分支**



#### 处理冲突



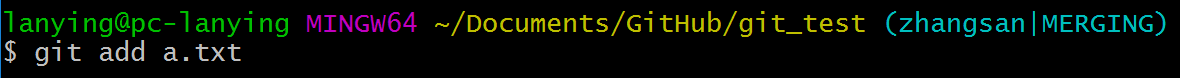
注意此时提示的是当前分支|MERGING

① **打开文件**

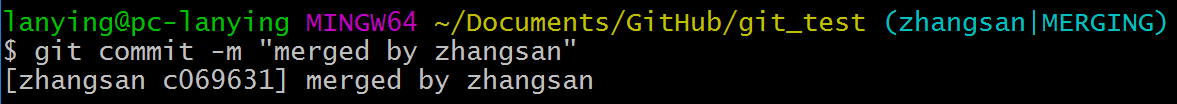
**里面会有详细的冲突信息，不同版本的内容用=======隔开。修改为想保留的内容**



**② 跟踪这个文件 git add a.txt**



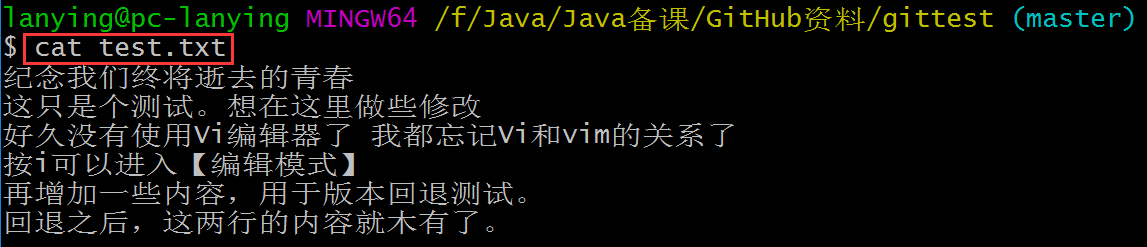
**③ 提交git commit –m “xxx”**



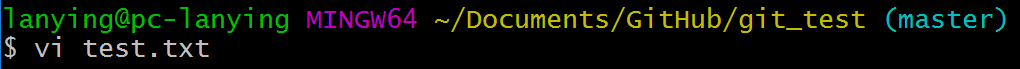
**注意：此处处理的只是将lanying分支合并到zhangsan分支时的冲突。lanying分支的文件和master分支上的文件都没有被修改。**

## 其它命令

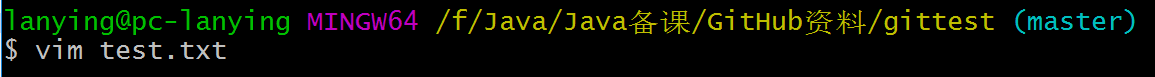
### 在Bash中查看某个文本文件 cat text.txt



### 使用Vi编辑器打开文件



or



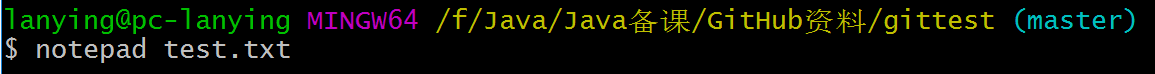


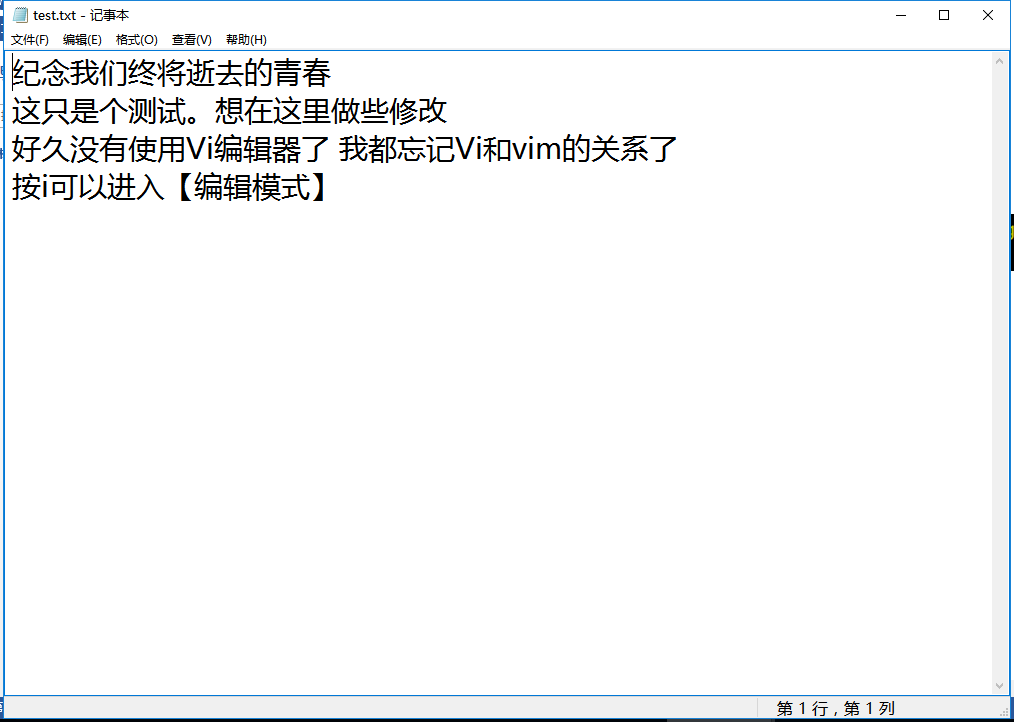
【PS】Vi编辑器的使用：

按**i**或者**a**进入【编辑模式】

按**Esc**进入【命令模式】，此时可以输入**:wq**保存退出

### 使用notepad打开文件（注意另存为UTF-8编码格式）

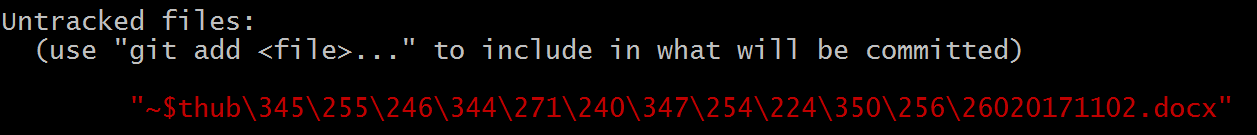




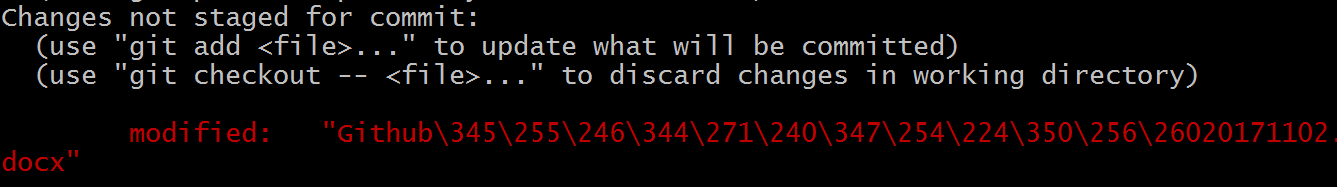
### 查看当前文件状态 git status （很常用）



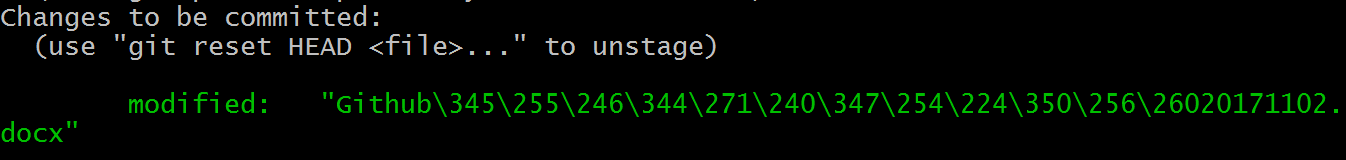
Untracked files 仓库里新建的文件，表示没有被跟踪的文件（通过add跟踪）



Changes not staged for commit 被跟踪的文件发生了变化，但是还没有add到暂存区



Changes to be committed 变化已被提交到暂存区，还没有被commit



### 查看提交记录 （很常用）

* 1. git log

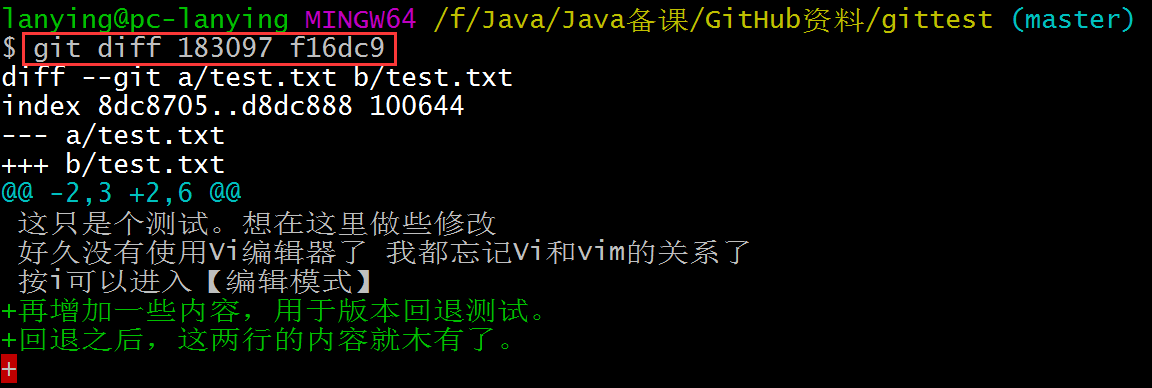


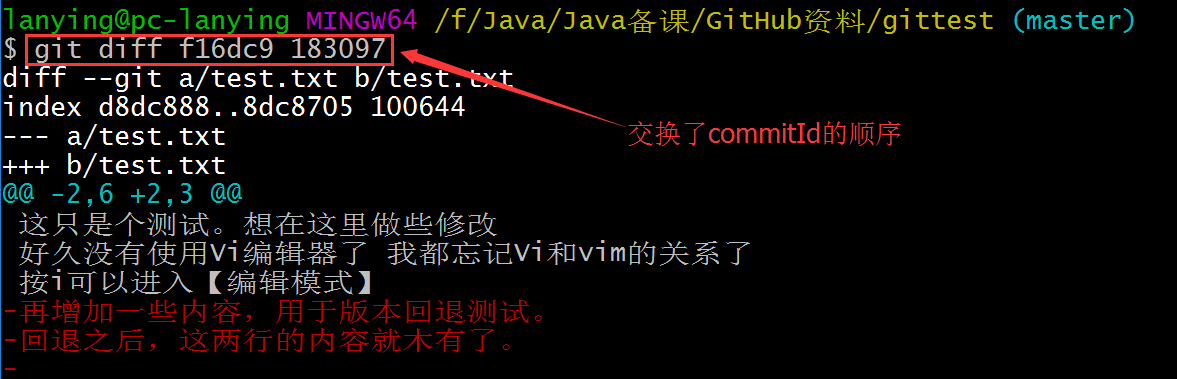
* 1. git log --pretty=oneline 单行显示模式



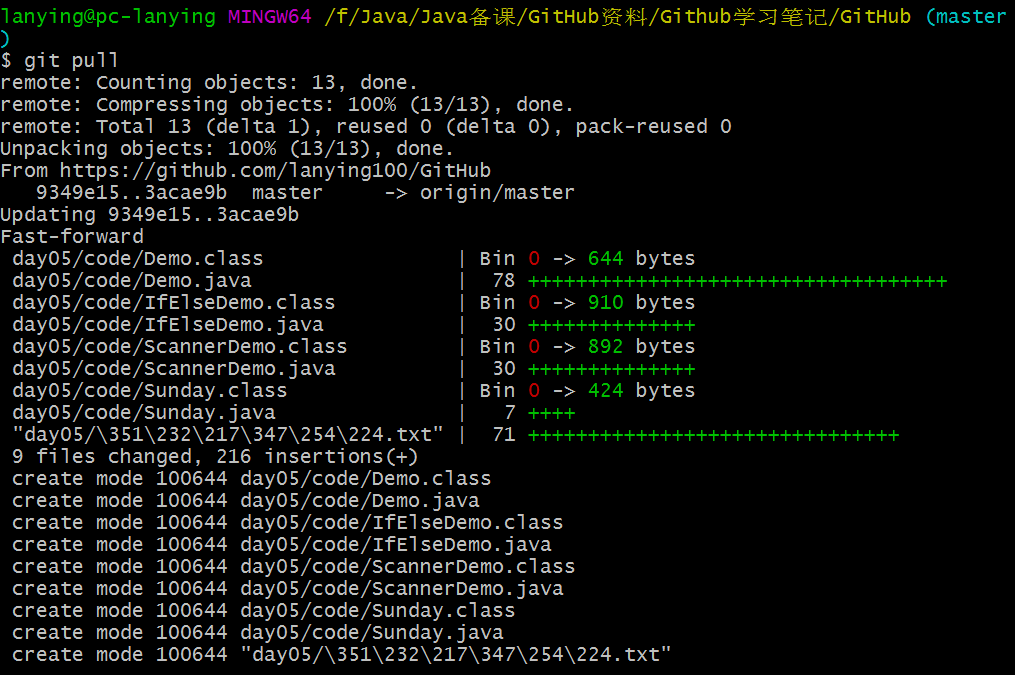
### 比较两个版本之间的差异

git diff <source\_branch> <target\_branch>

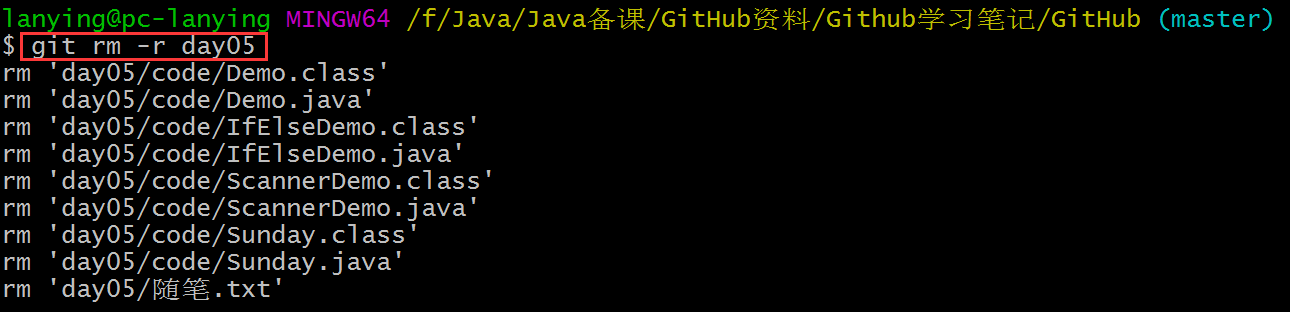




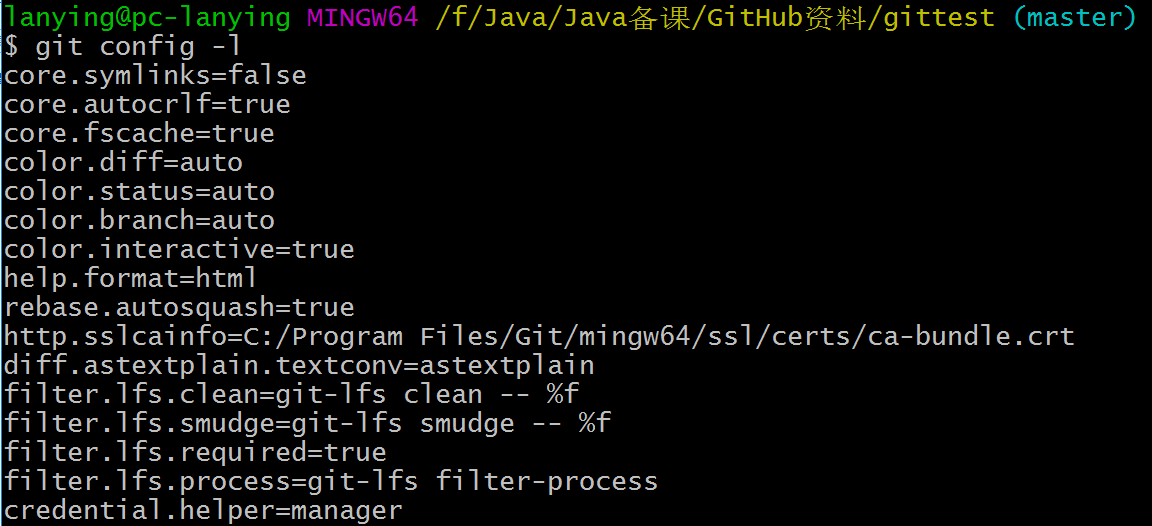
### 更新本地仓库：将GitHub上的仓库Pull到本地 git pull



### 删除目录 git rm --rv dirpath

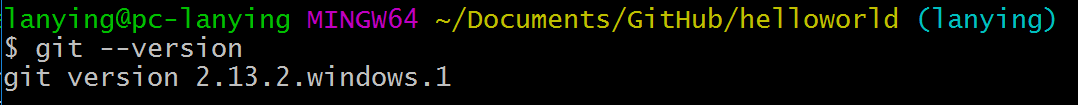


### 查看当前配置信息 git config –l



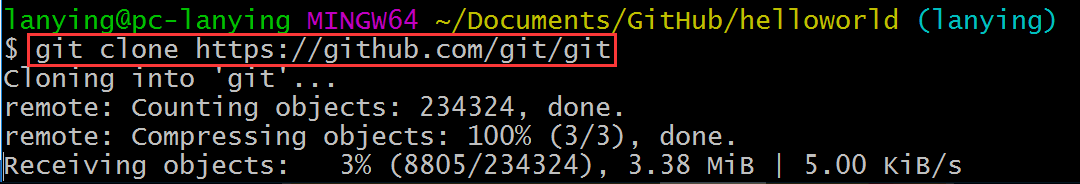
如果忘了，就git config --help

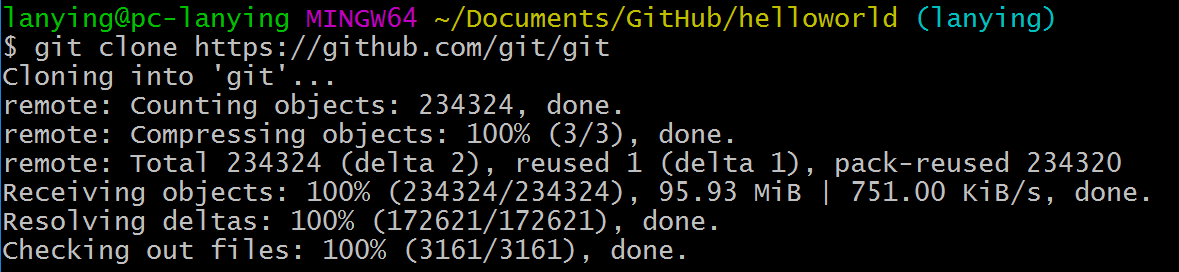
### 查询当前的git版本 git --version



### 联网自动更新到最新版本的git

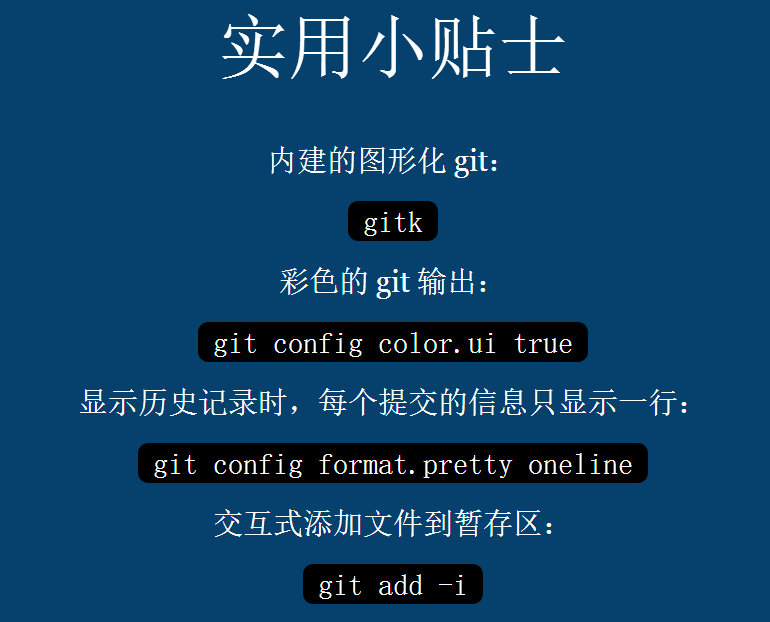
git clone https://github.com/git/git





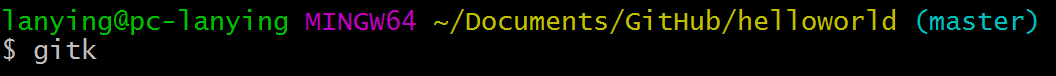
疑问：更新之后，git --version还是之前的版本。

### 辅助命令

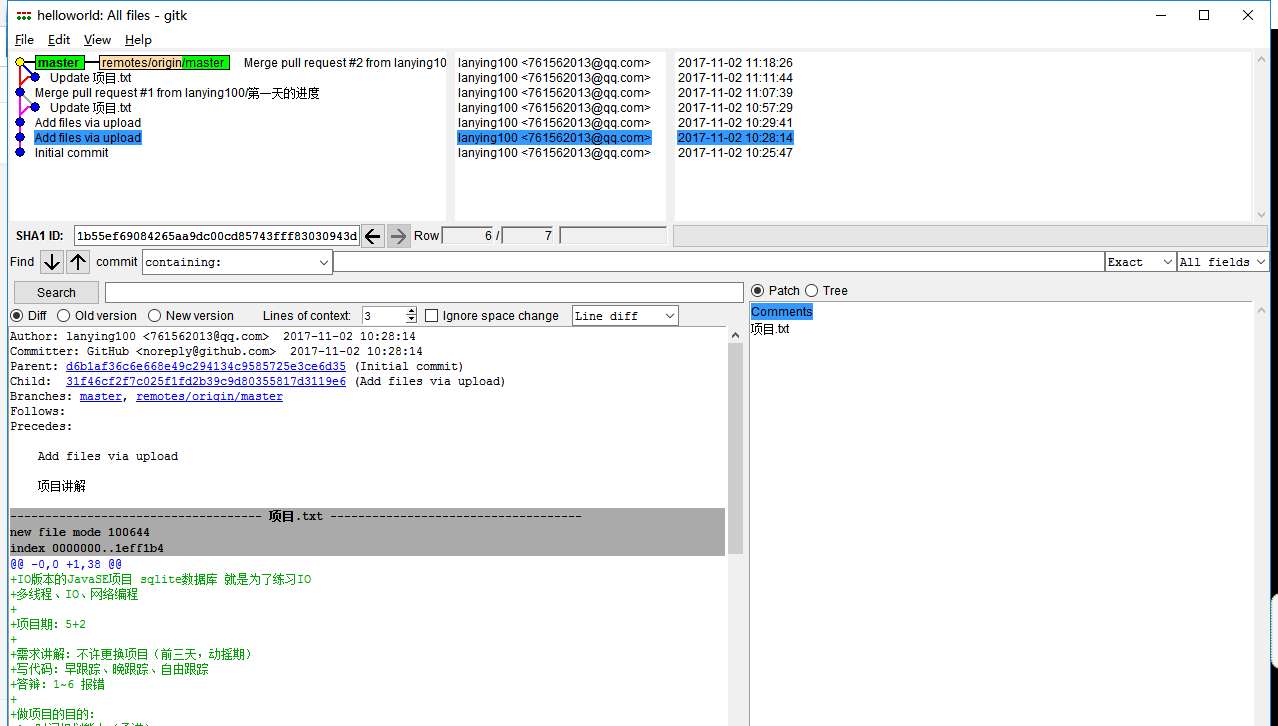


## gitk工具

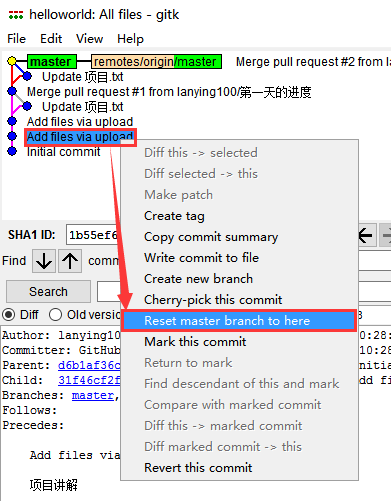
### 打开gitk图形化工具 gitk



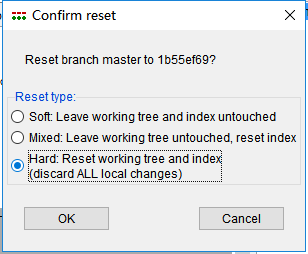
### 查看各个版本之间的变化



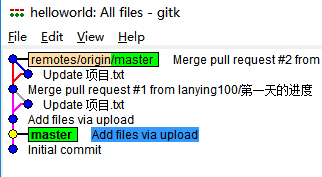
### 选择想要回退的版本，鼠标右键reset



### 选择回退的模式，简单粗暴一些：hard



### 5. 回退之后，master分支退到了目标版本



感想：gitk真的是想去哪儿就去哪儿。对比下来，感觉GitHub Desktop除了新旧文件对比清晰明了一些，没啥吸引力了……（也许是我还不会用GitHub Desktop）

## 安装程序

### GitHub Desktop

【下载地址】<https://desktop.github.com/>

### Git Bash

【下载地址1】<https://git-for-windows.github.io/>

【下载地址2】<https://github.com/git-for-windows/git/releases/tag/v2.15.1.windows.2>

# 学习网址：

## 【官方教程】

<https://services.github.com/on-demand/github-cli/git-configuration>

## 【简明教程：中文】

<http://rogerdudler.github.io/git-guide/index.zh.html>

## 【reset 原理】

<https://git-scm.com/blog>

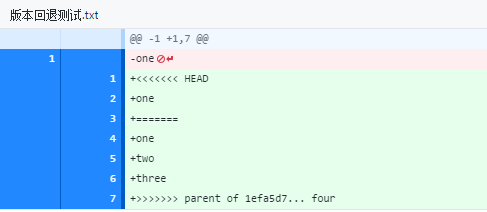
## 【Pro Git : Everything You Need To Know About Git】

<https://git-scm.com/book/zh/v2>

# FAQ

## 问题1：冗余辅助信息

版本回退时，有时会出现一堆辅助描述信息：



真相：这是因为版本冲突，需要手动处理冲突。我们需要在======的上面或者下面选择一个（或者自己调整）。

## 问题2：What does “@@ -8,12 +8,13 @@” mean in Git's diff output?

真相：- 表示old File ; + 表示 new File

-8,12 表示 旧文件中从第8行开始，有12行

+8,13 表示 新文件中从第8行开始，有13行



【参考网址】

<https://stackoverflow.com/questions/10950412/what-does-1-1-mean-in-gits-diff-output/10950496>

## 问题3：Git Bash鼠标左键选中内容后，会自动换行，显示^C

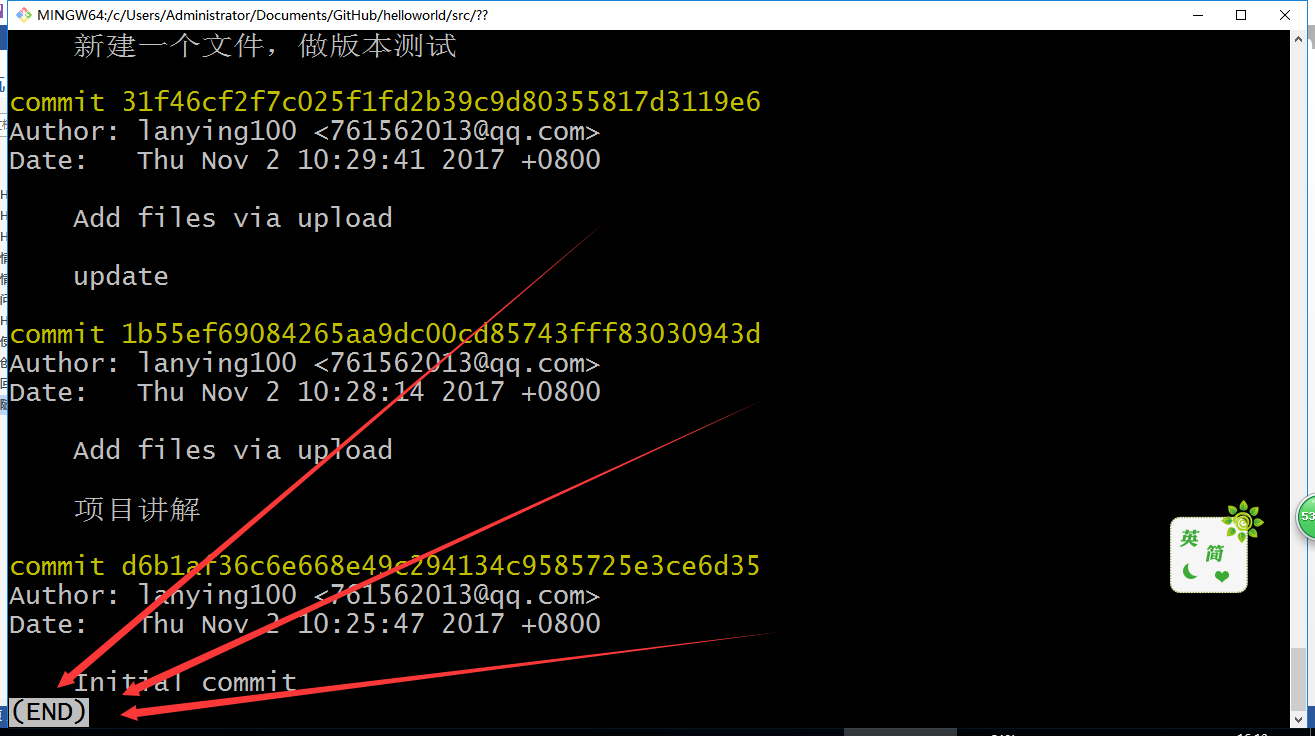
解决方案：是**有道词典**导致的，关闭它即可。

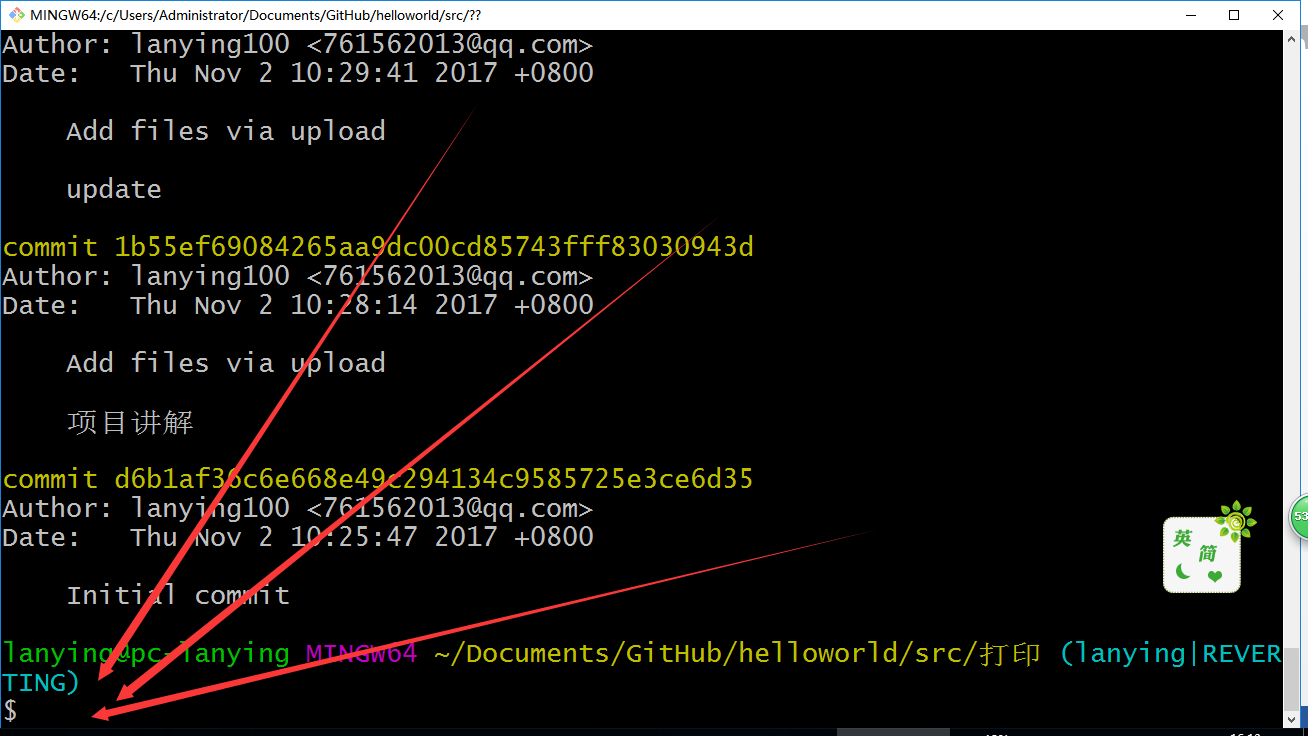
参考网址

<https://stackoverflow.com/questions/38629085/when-copy-text-from-git-bash-window-through-mouse-selection-how-to-suppress-c>

## 问题4：命令 git log 之后，显示内容太多，最后有个(End) 或者其它标签，无法继续输入。

解决方案：按 q 即可





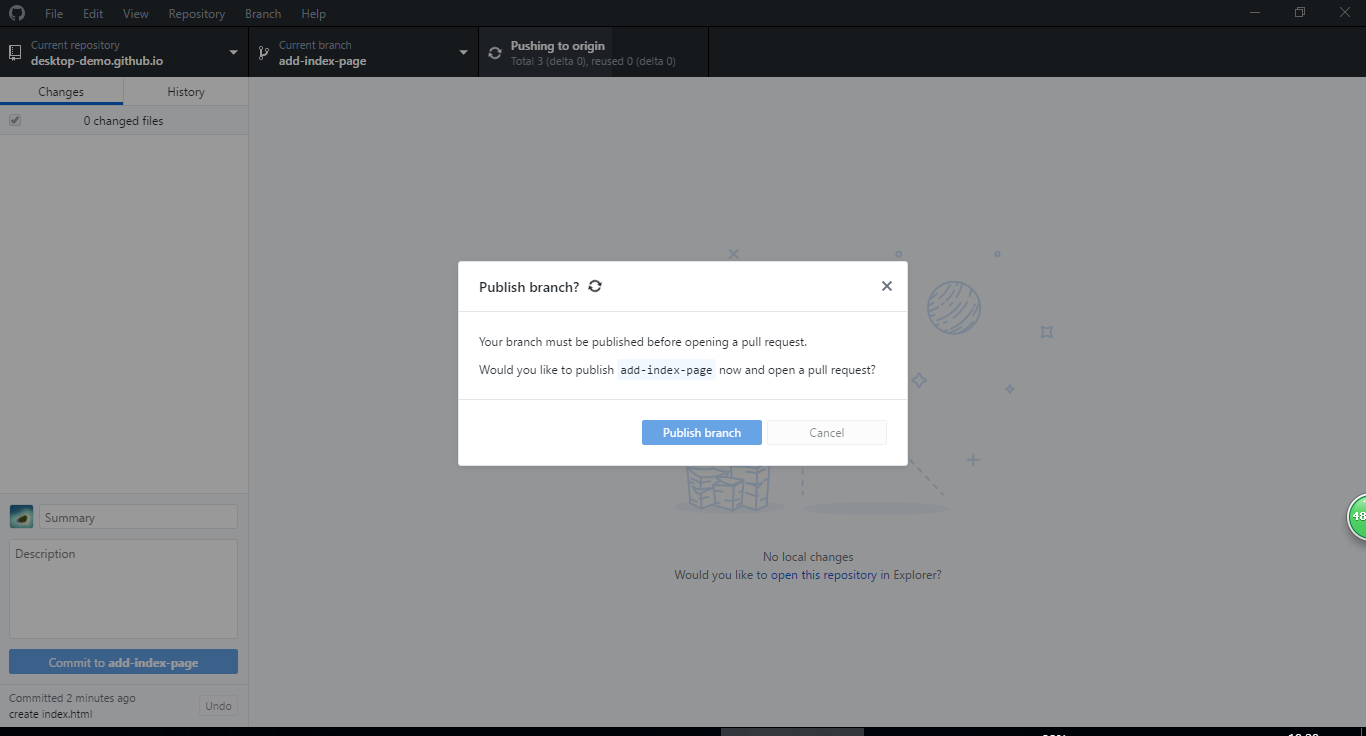
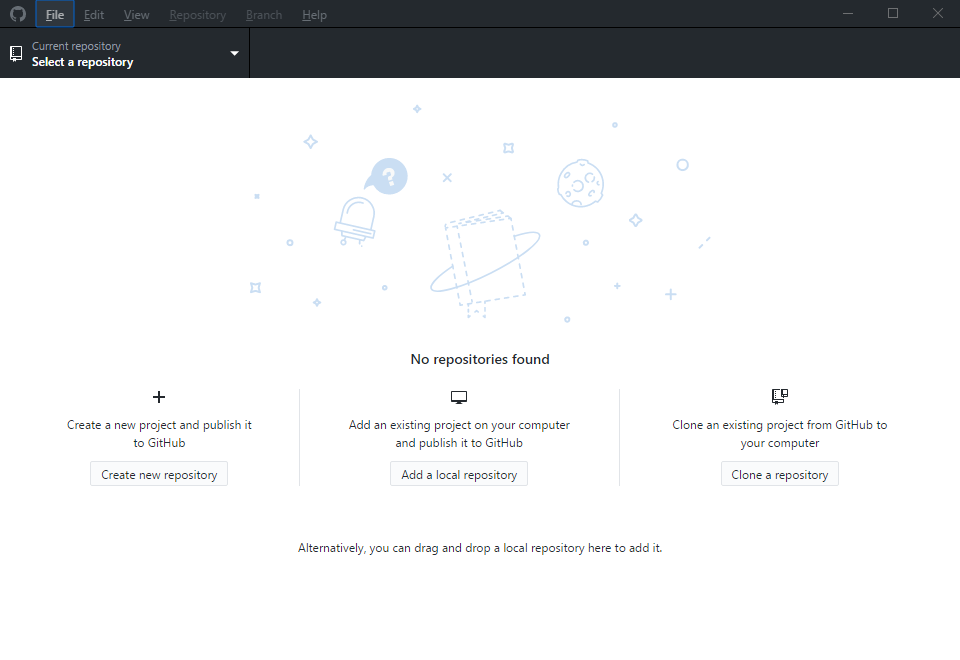
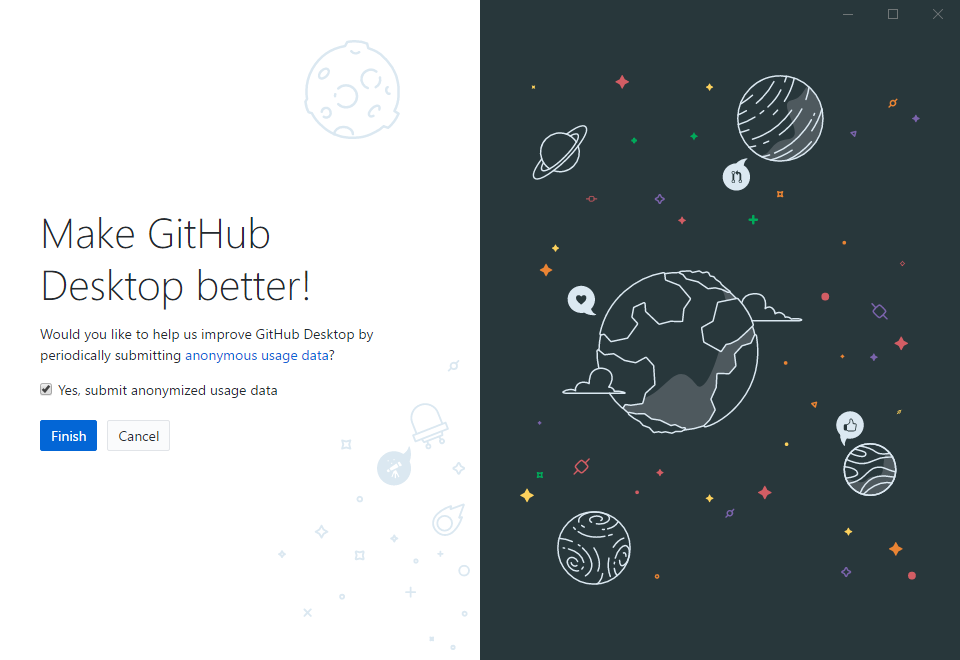
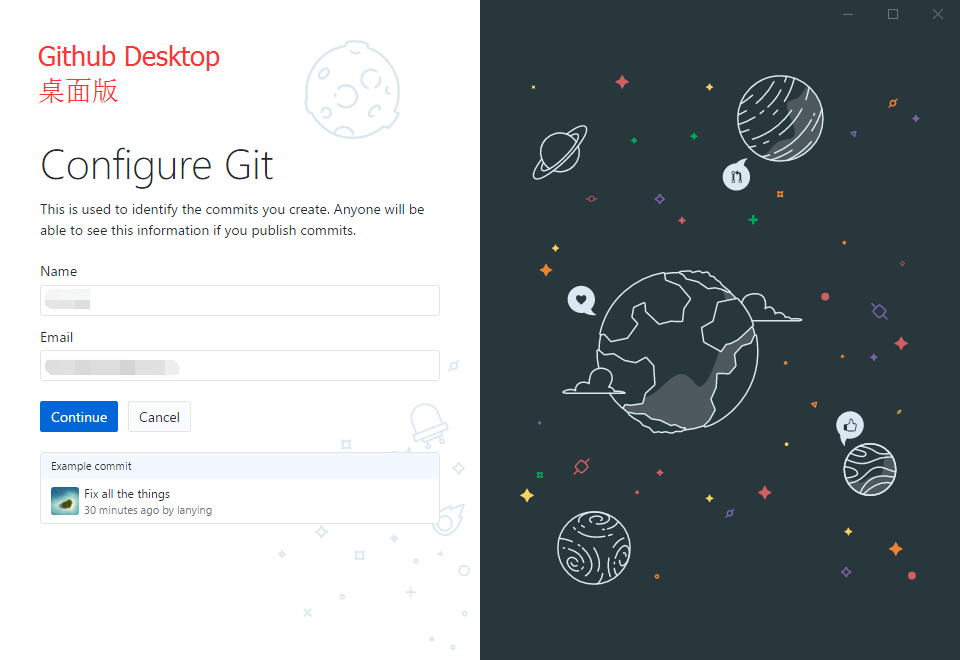
参考网址

<https://www.zhihu.com/question/26423744>

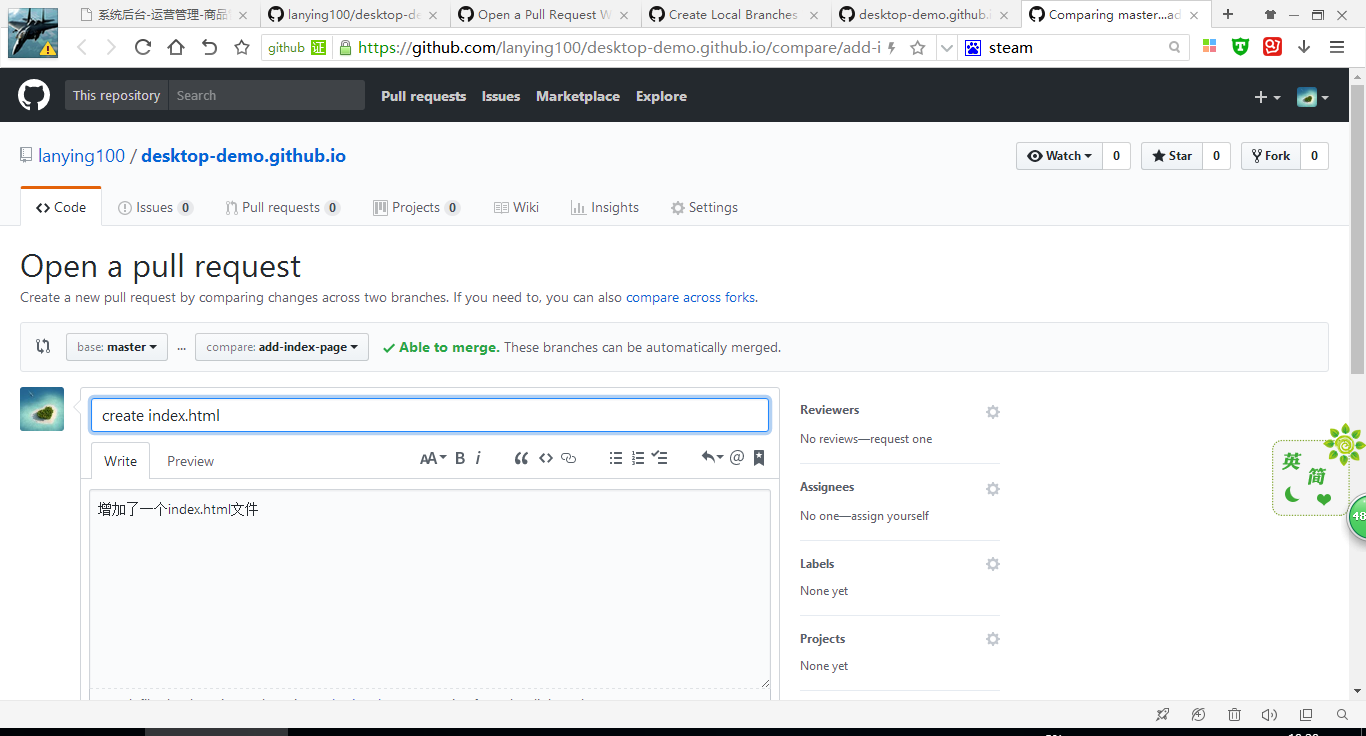
# GitHub Desktop（未完待续）

## GitHub Desktop 从GitHub上clone仓库

【下载地址】<https://desktop.github.com/>



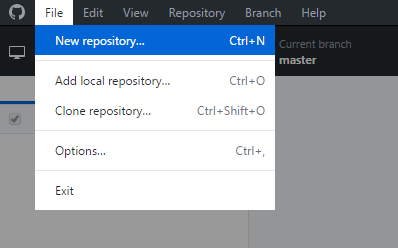
竟然打开了网页，既然要用网页，还要你桌面版何用……



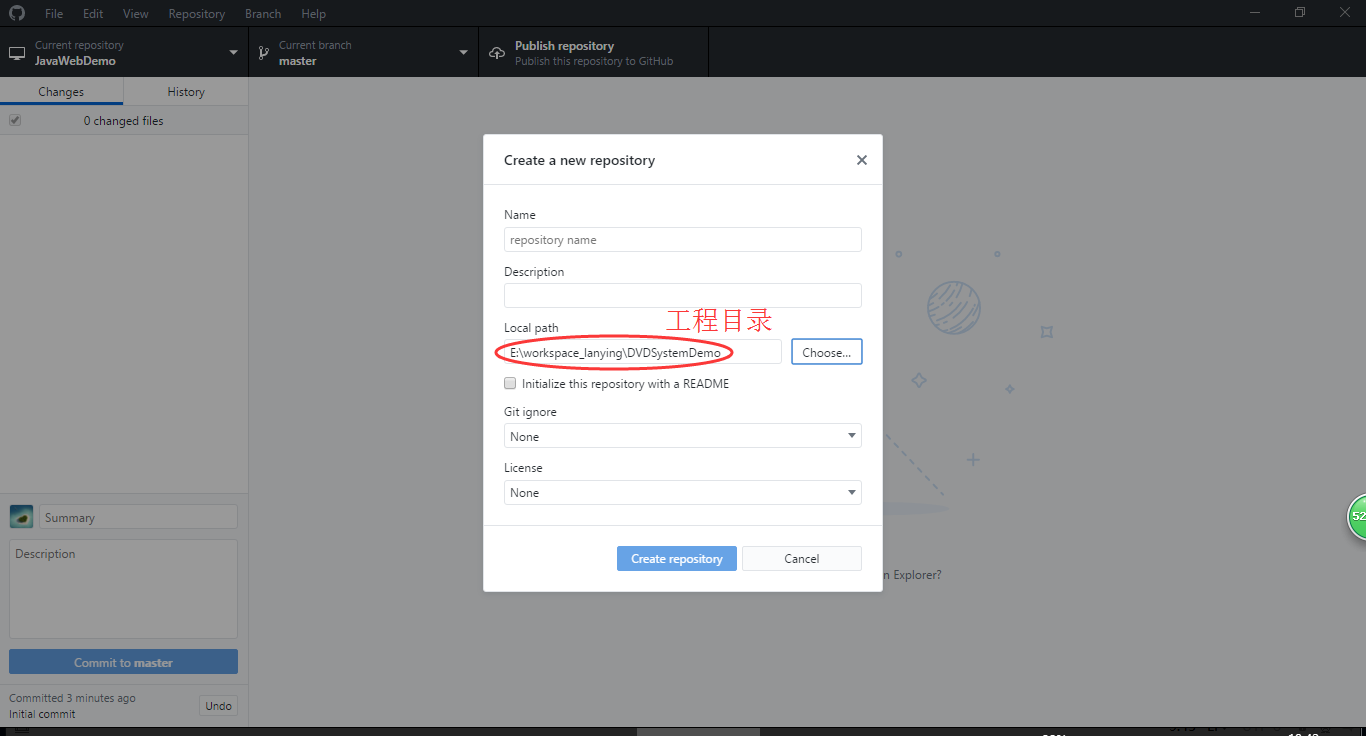
## GitHub Desktop 创建本地仓库

【通过桌面版GitHub添加一个本地项目到GitHub】

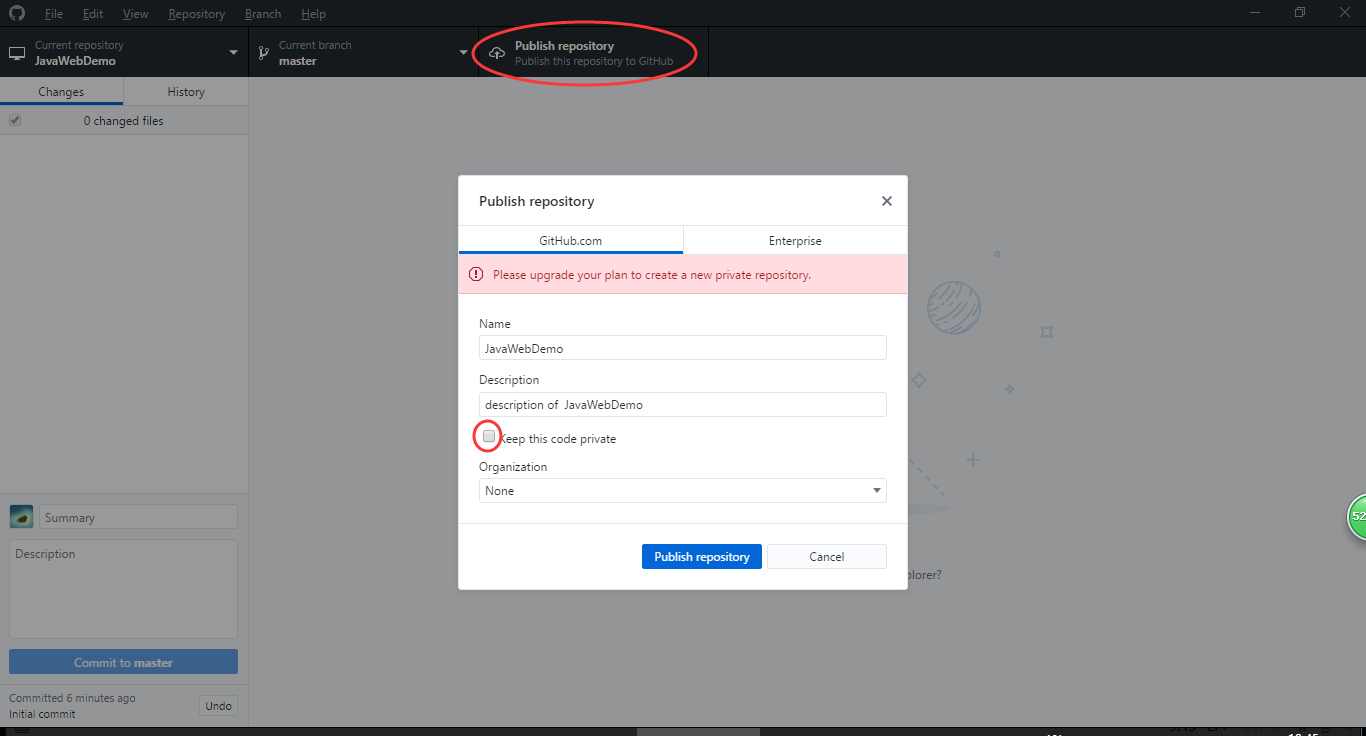
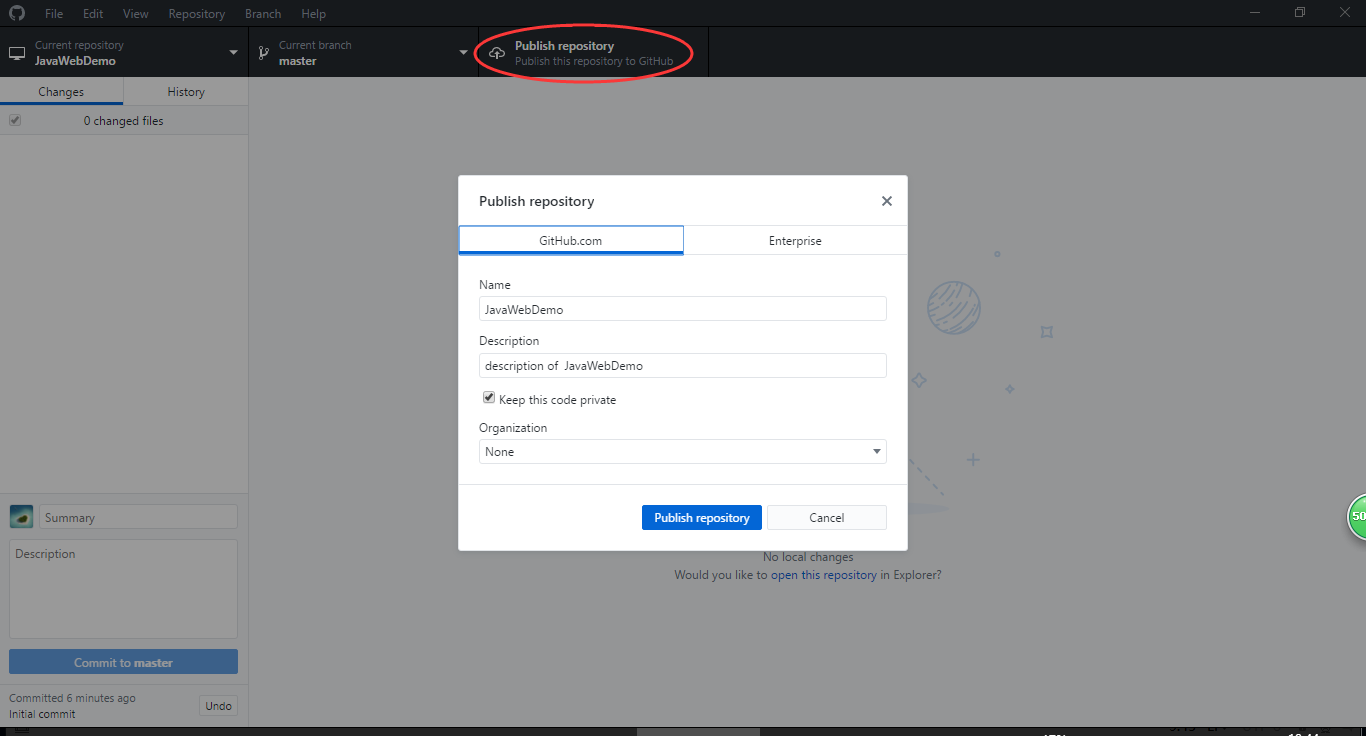
1. 选择File – New repository



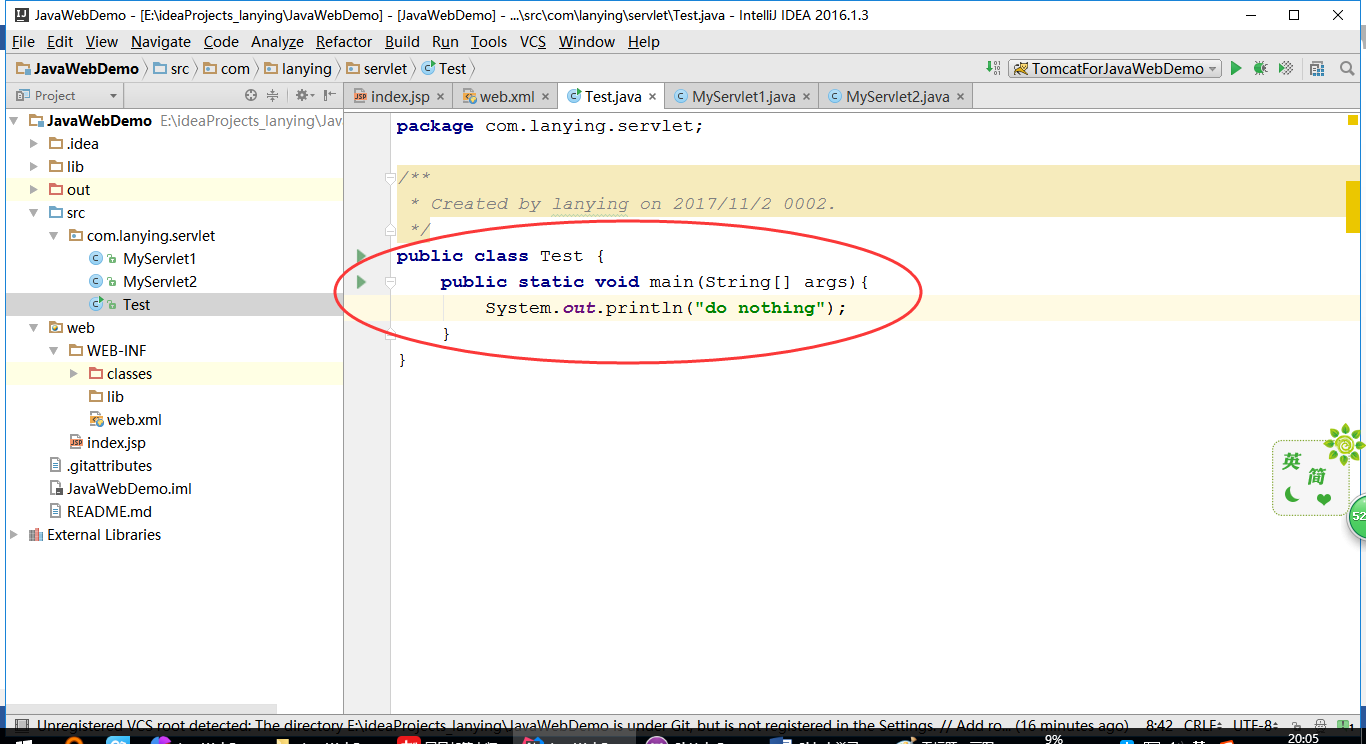
1. 选择工程目录

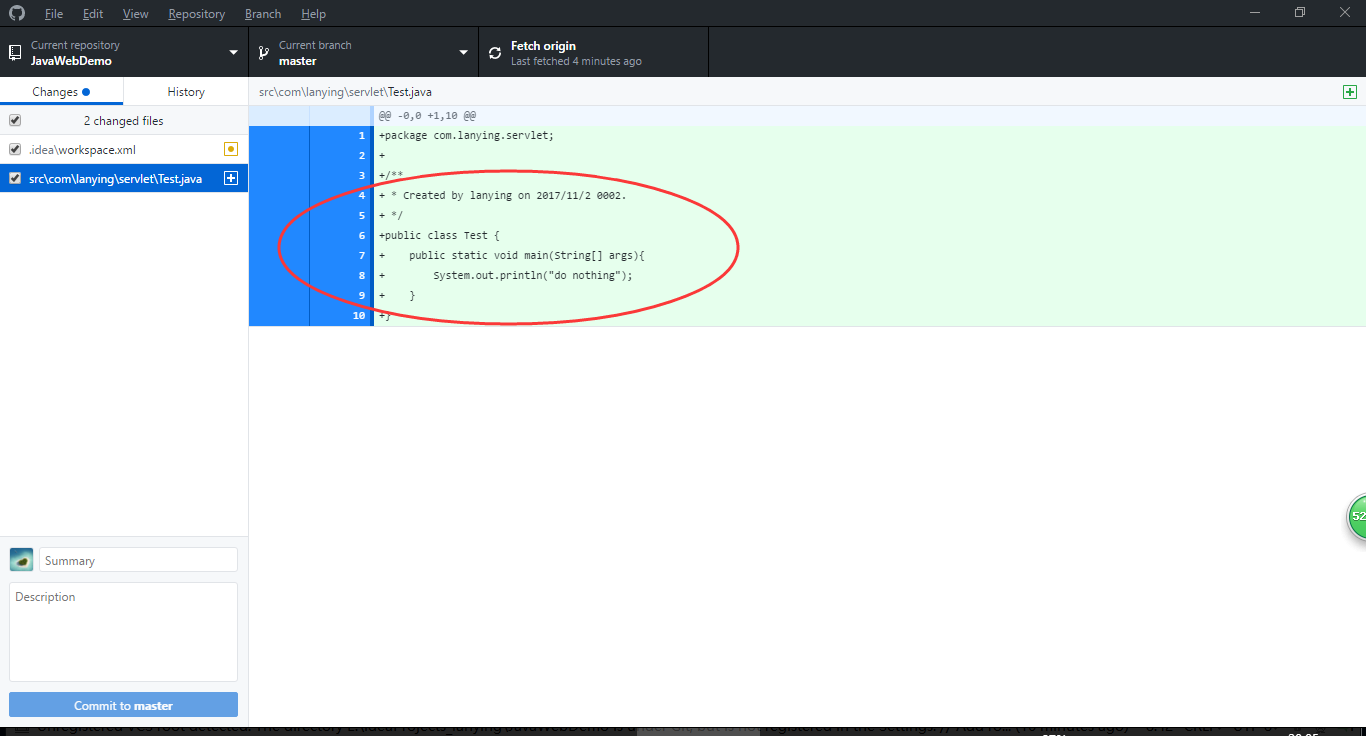


1. 提交到GitHub.com



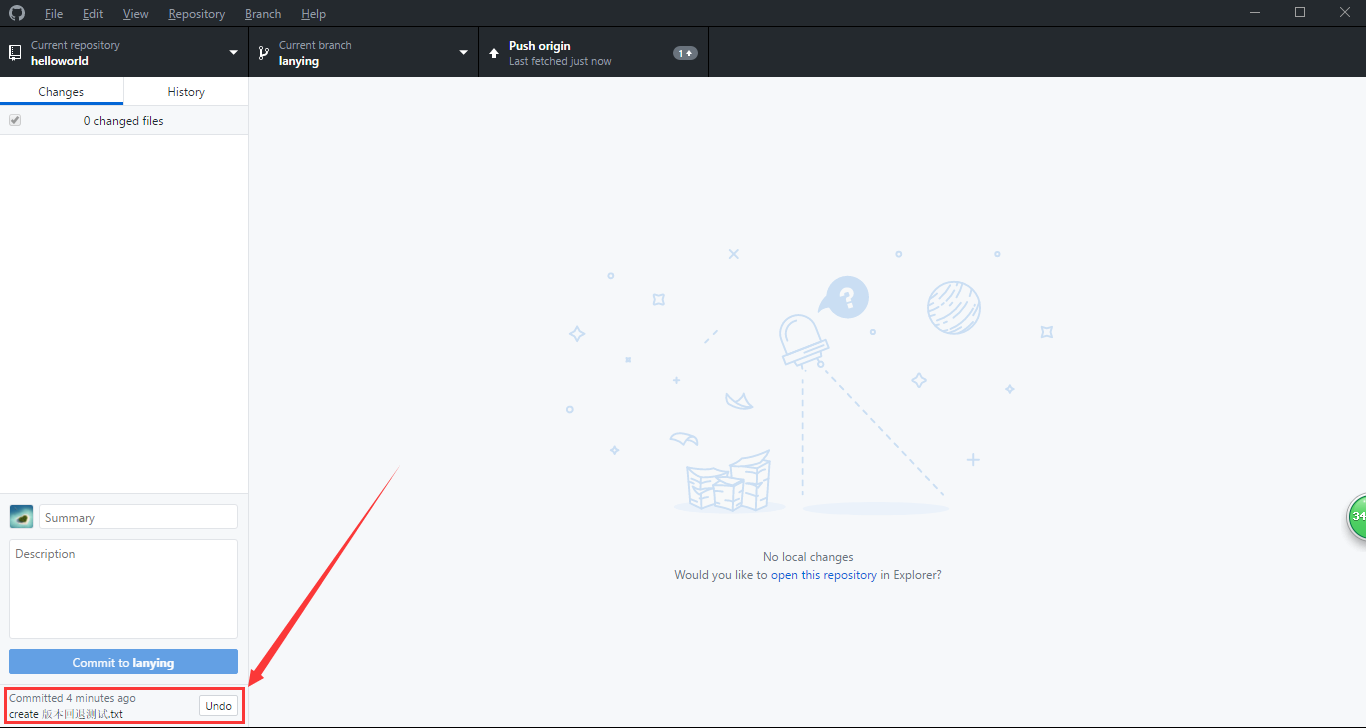
之后再写项目时，就会看到变化





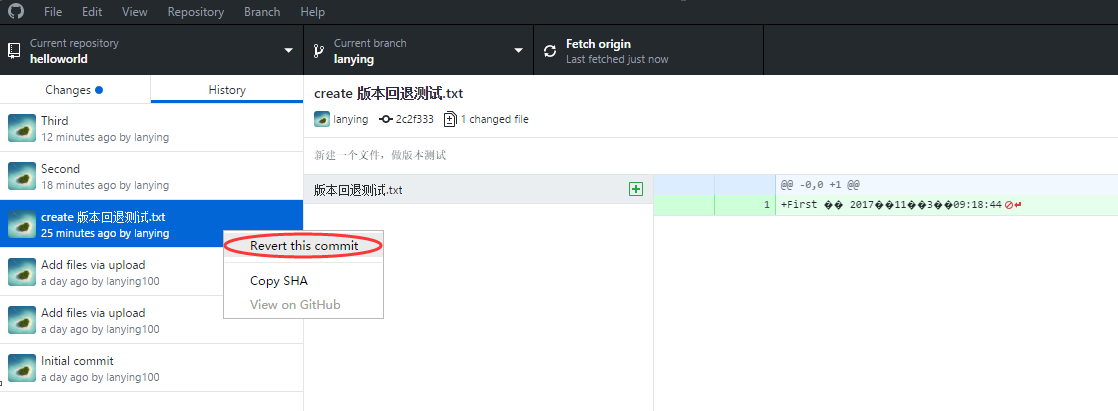
## GitHub Desktop 版本回退

### 情况一 还没有Push过

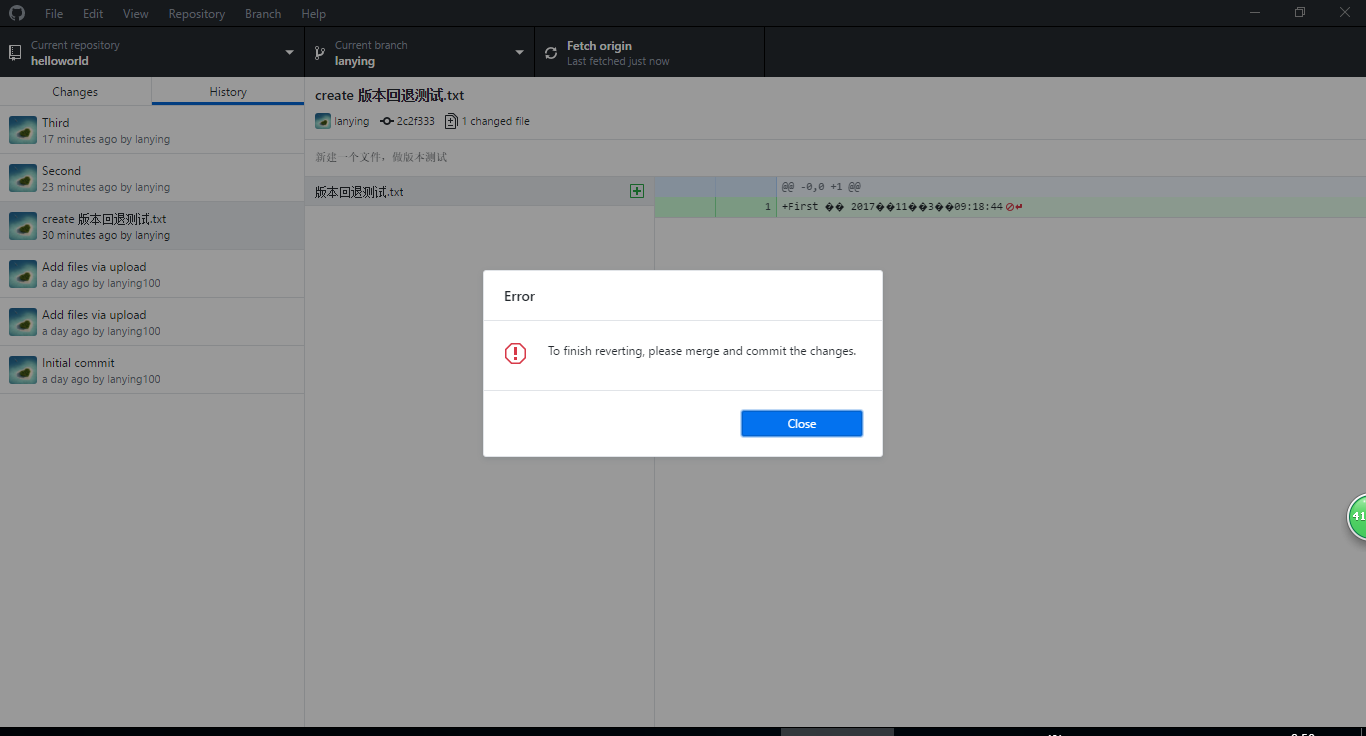


### 情况二 已经Push过了

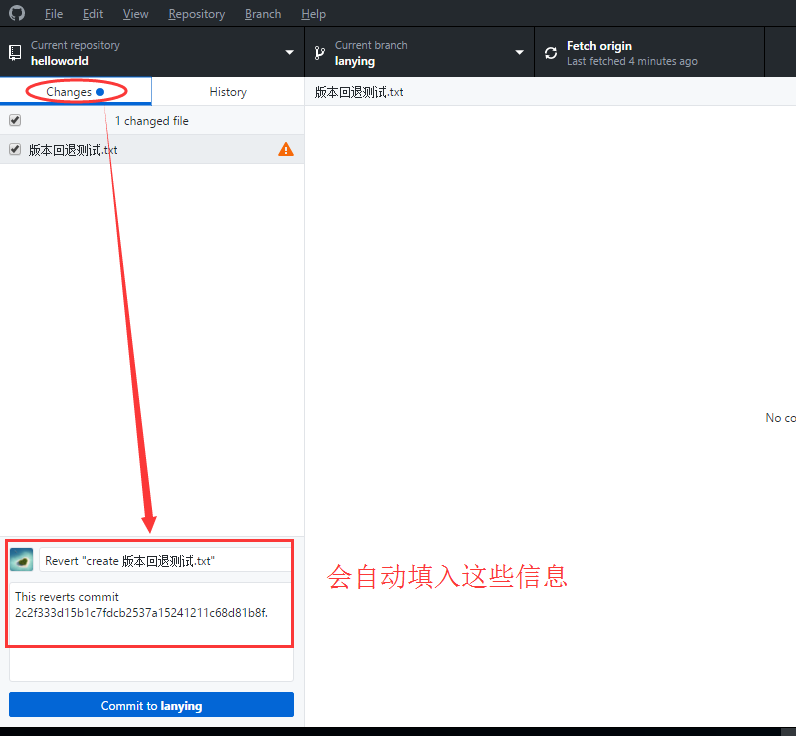
* 1. 选择要回退的版本，右键点击 Revert this commit



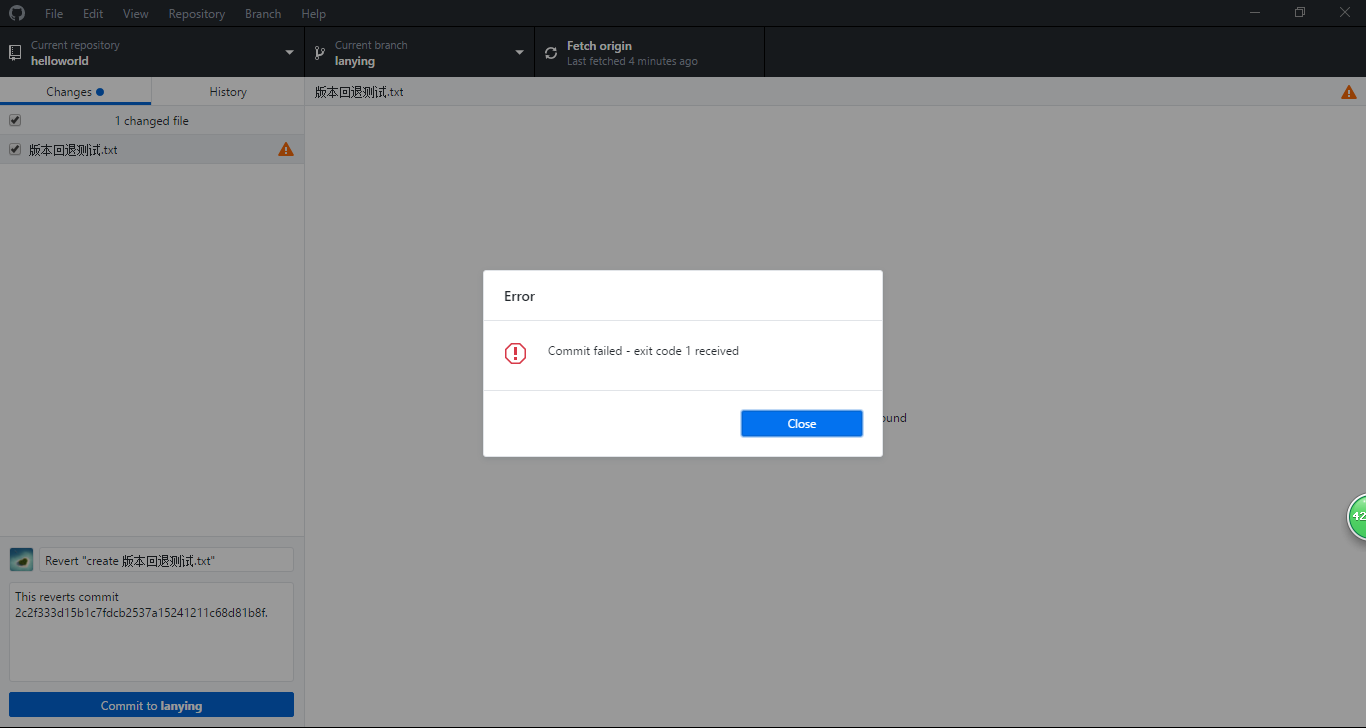
* 1. 此时会提示一个Error信息，点击close即可



* 1. 然后会发现左上方的Changes标签有新消息，同时会在左下方的Commit窗口自动填入一些回退必要信息



* 1. 然后……失败了



注意：回退到x版本，意味着回退到第x版本commit之前的状态。

比如回退到fourth版本（这个版本增加了four这个单词），那么回退之后，文件里的内容是没有four这个单词的