# Git功能及管理

## 摘要：

工欲善其事必先利其器，做任何事情都应该先去思考他的目标，在目标确立的情况下实行他的细则，然后在执行的过程中优化过程，并提出新的目标，缺一不可。

Git目标一：建立git仓库，设立主分支 main 管理分支master 建立其余配合分支 shaoshan xiangqing tianxin等 此外 还有 Develop Hotfix Release Feature分支；

Git目标二： 建立推送机制， 包括 add commit push fetch；

Git目标三：建立合并冲突机制 ，目前为重中之重，包括 配合分支与master 合并冲突 master与main合并冲突；

Git目标四： 工具使用与指令使用，包括 git指令使用 sourcetree使用；

目录

[Git需要完成的功能 1](#_Toc5302)

[摘要： 1](#_Toc27905)

[第一章 Git仓库建立与分支设置 3](#_Toc6092)

[第二章 Git推送机制 3](#_Toc8665)

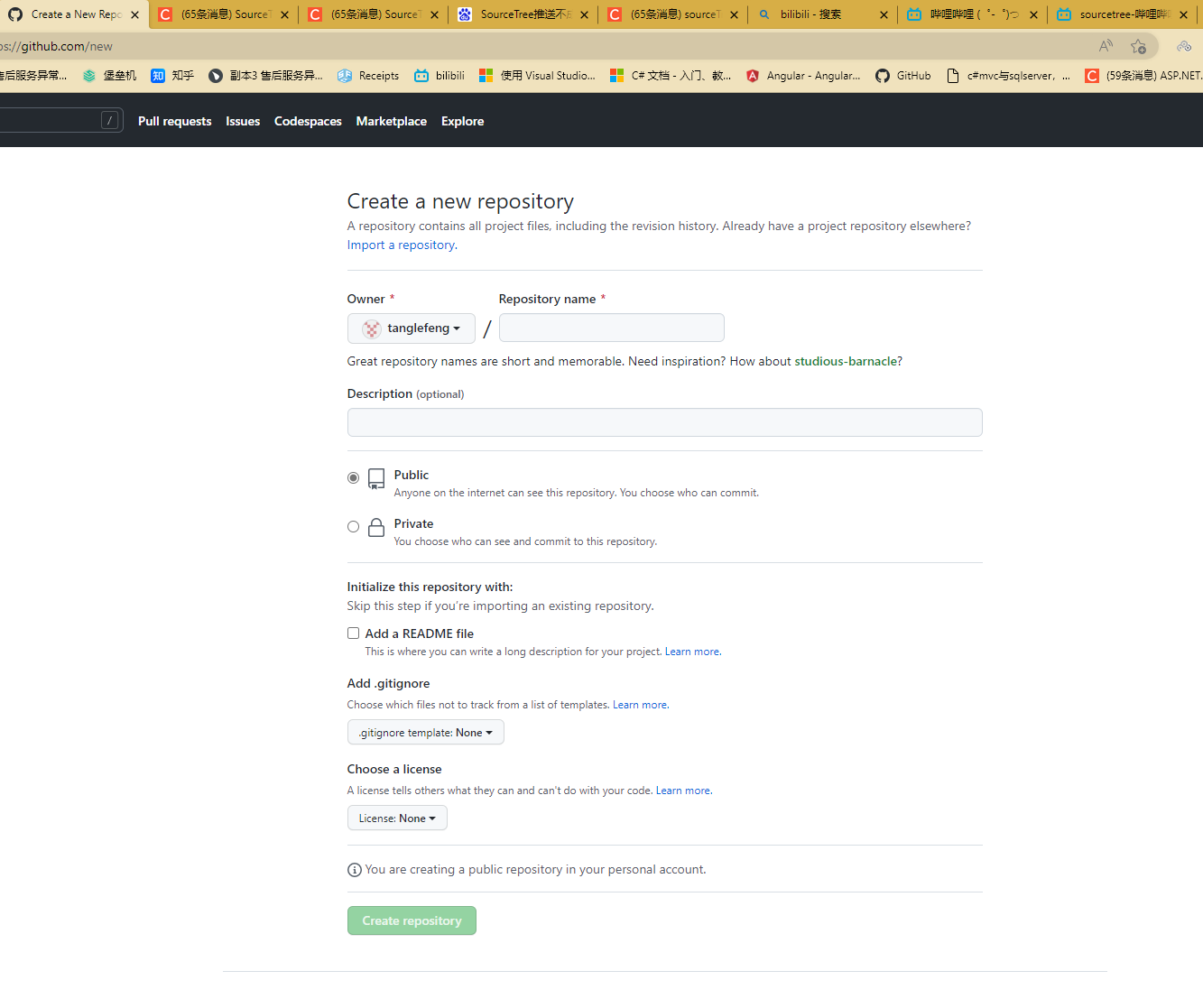
[第三章 Git合并冲突机制 3](#_Toc13329)

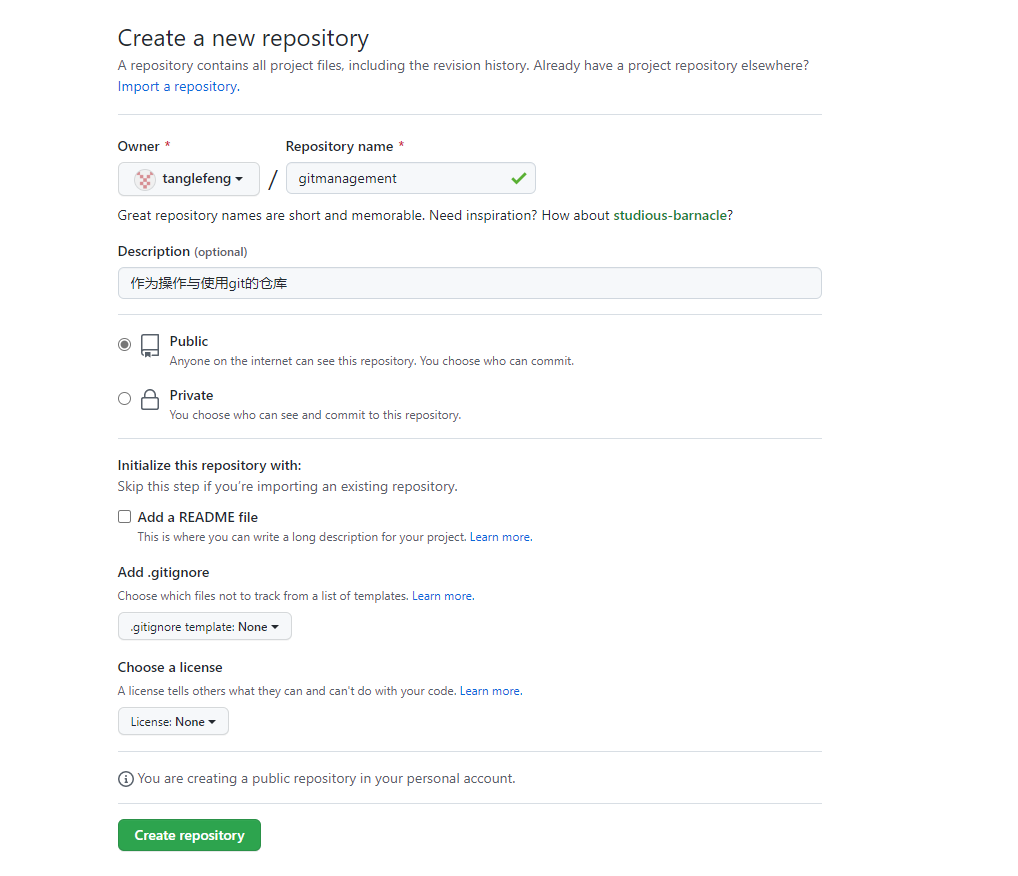
[第四章 工具使用与指令使用 3](#_Toc15675)

[第五章 小结 4](#_Toc12152)

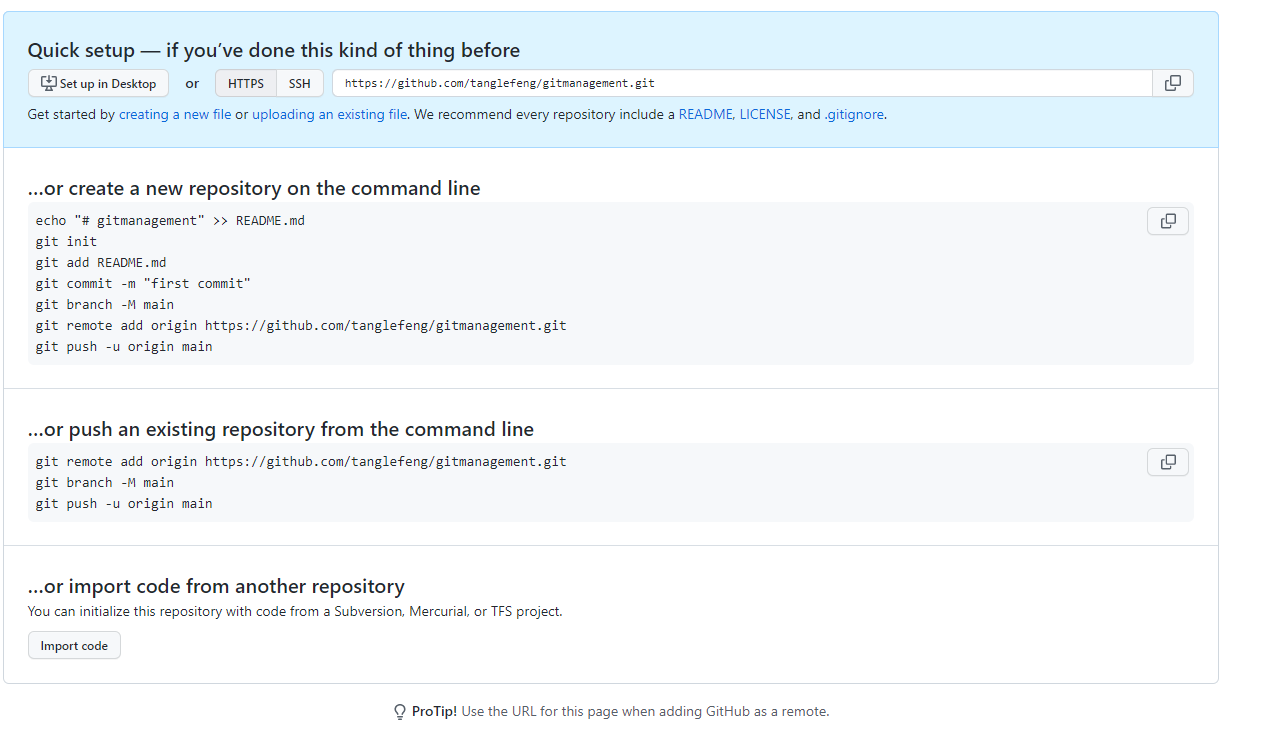
## Git仓库建立与分支设置

### 1 建立一个仓库

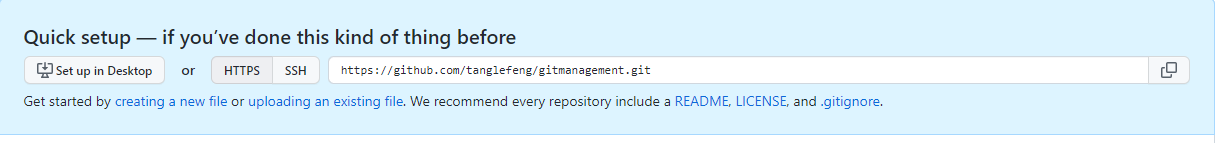




会出现一下提示



简单描述一下



<https://github.com/tanglefeng/gitmanagement.git>

这个是我们git仓库的网址 也就是我们要clone的网址



echo "# gitmanagement" >> README.md

git init

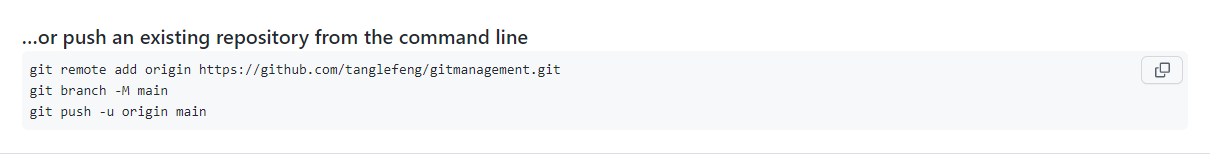
git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/tanglefeng/gitmanagement.git

这个是在仓库第一次建立的时候，建立分支与提交到远程仓库的指令



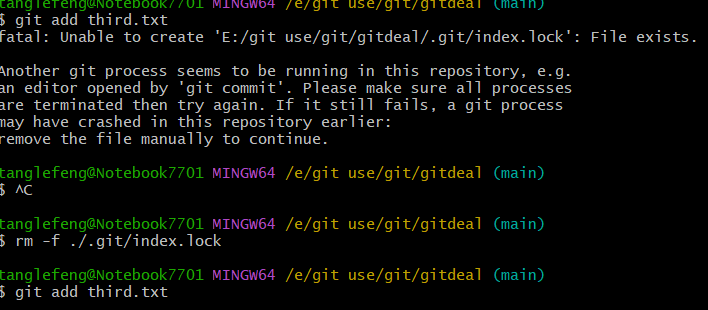
git remote add origin https://github.com/tanglefeng/gitmanagement.git

git branch -M main

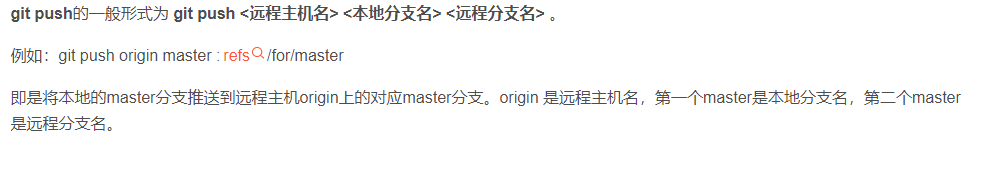
git push -u origin main

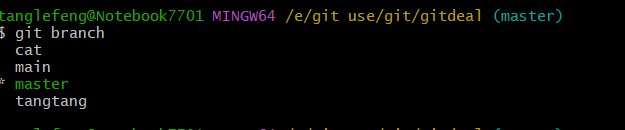
这个是第二次及以上提交代码或者建立分支的指令

2 建立分支(本地仓库 区别于上面git仓库) 思考：一台电脑是否在两个磁盘建立两个文件名来创立两个分支 这个后期可以尝试



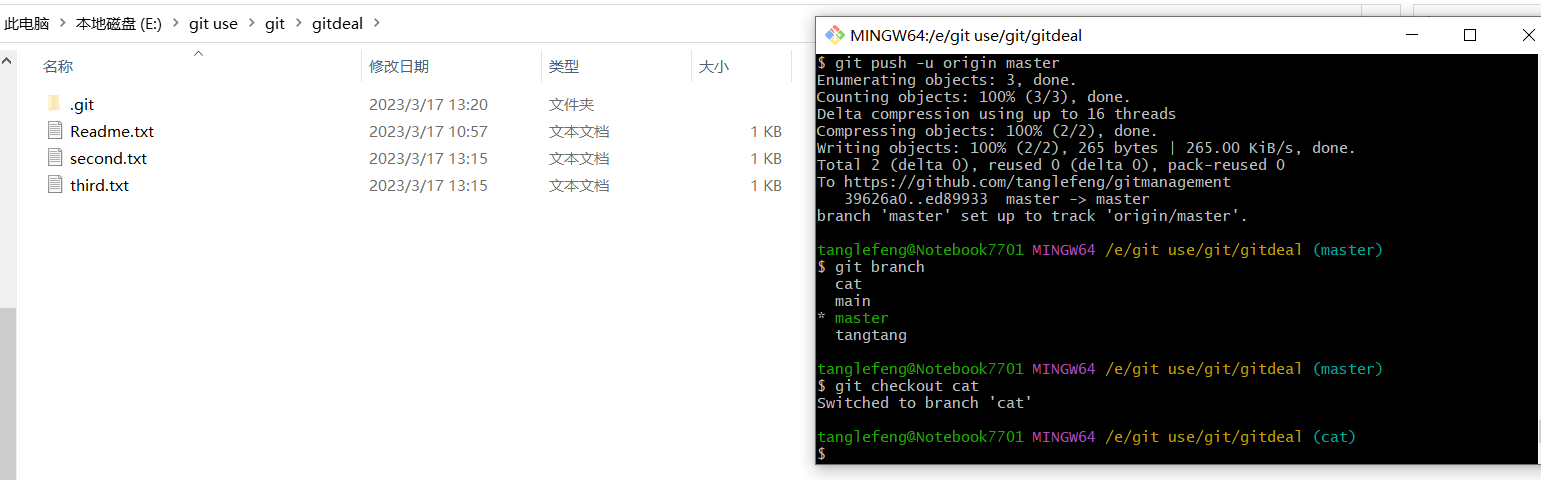
如果碰到这种git冲突的可以选择rm -f ./.git/index.lock的方式



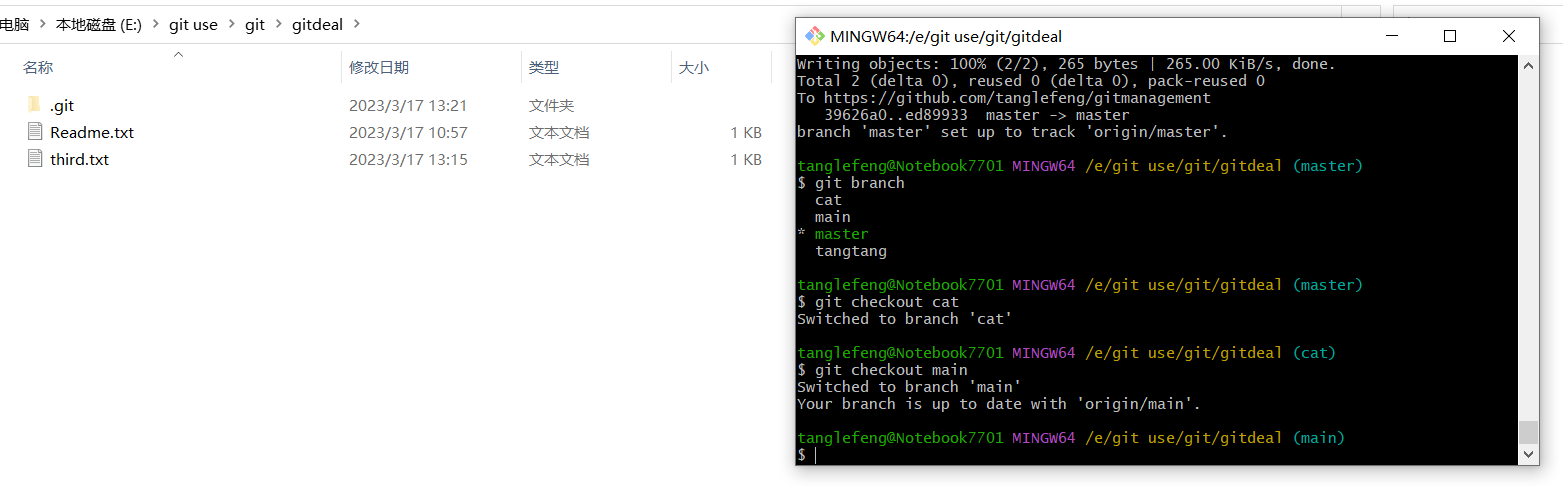


这个代表 一共是四个分支 分别为cat main master tangtang

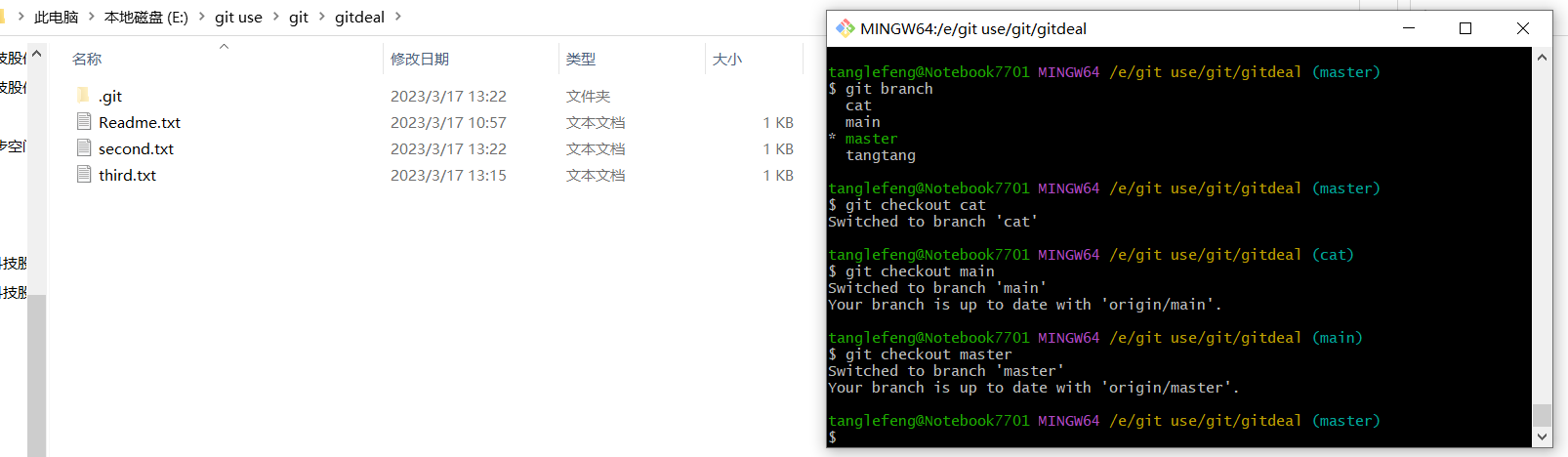
git checkout cat 即有三个文件夹 分别为Readme.txt second.txt third.txt



git checkout main 只有两个文件夹 即 Readme.txt third.txt

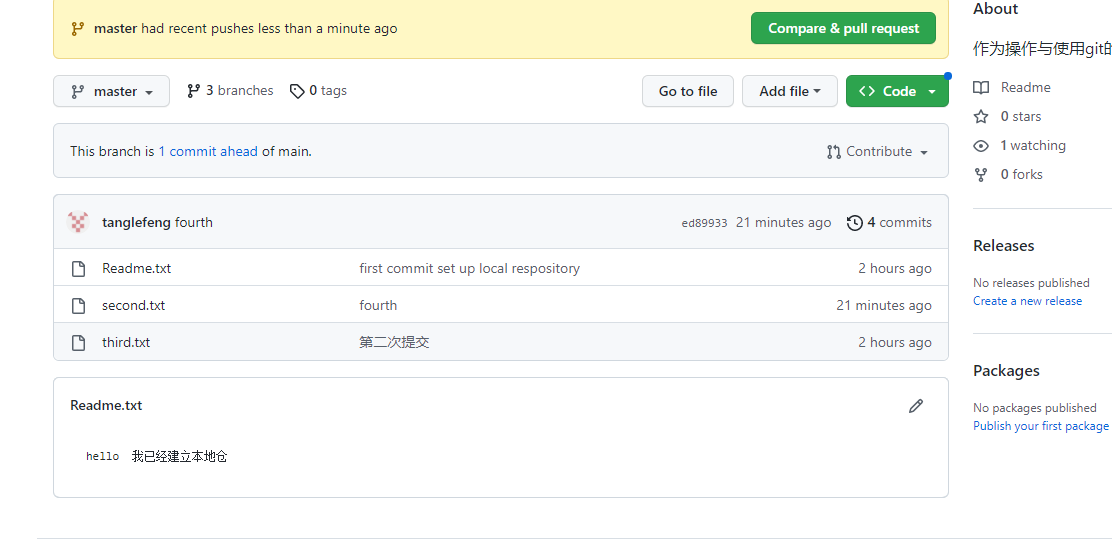


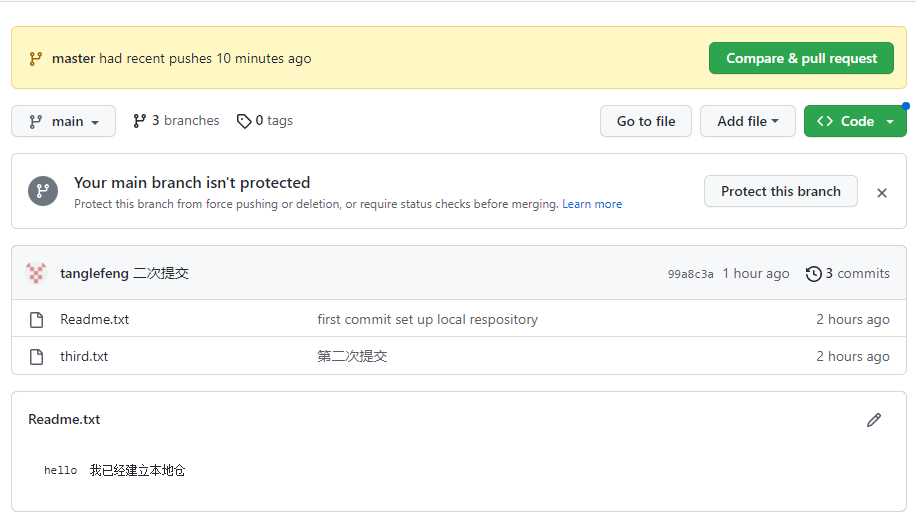
git checkout master 即有三个文件夹 分别为Readme.txt second.txt third.txt

git checkout tangtang 只有一个文件夹 就是Readme.txt



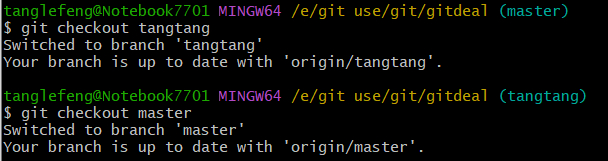
选择那个文件夹 可以对比github的分支进行推送 基本上符合cat/tangtang-master 或者cat/master/tangtang-main其中cat与mater相同 后期我们主要测试点在与master/tangtang向master推送过程中的机制及其冲突解决。





1. master向github上的master进行提交/冲突解决
2. tangtang向github上的master提交/冲突解决
3. tangtang向github上的tangtang提交/冲突解决
4. Github上的tangtang合并到github上的master上

其中 master与tangtang同属于origin的分支



按照目前的情况 先解决tangtang在github和本地仓库不一致的区别

## Git推送机制

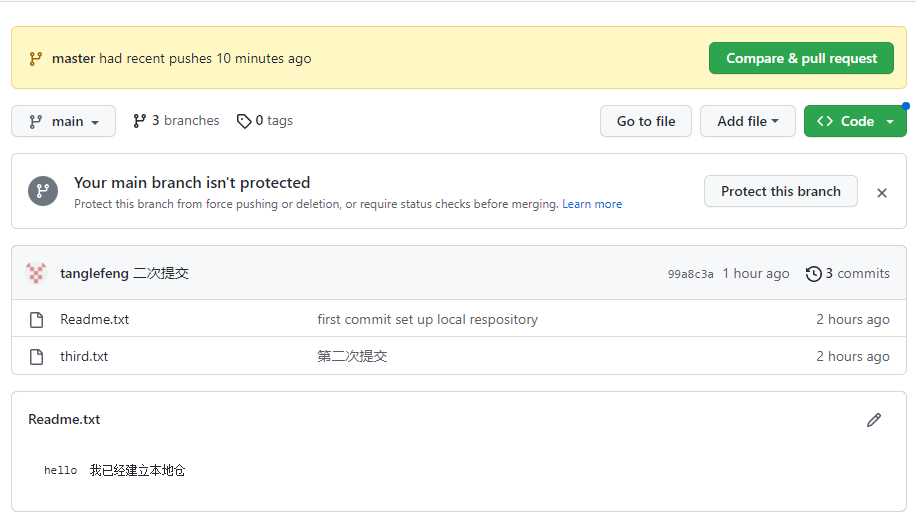
git add

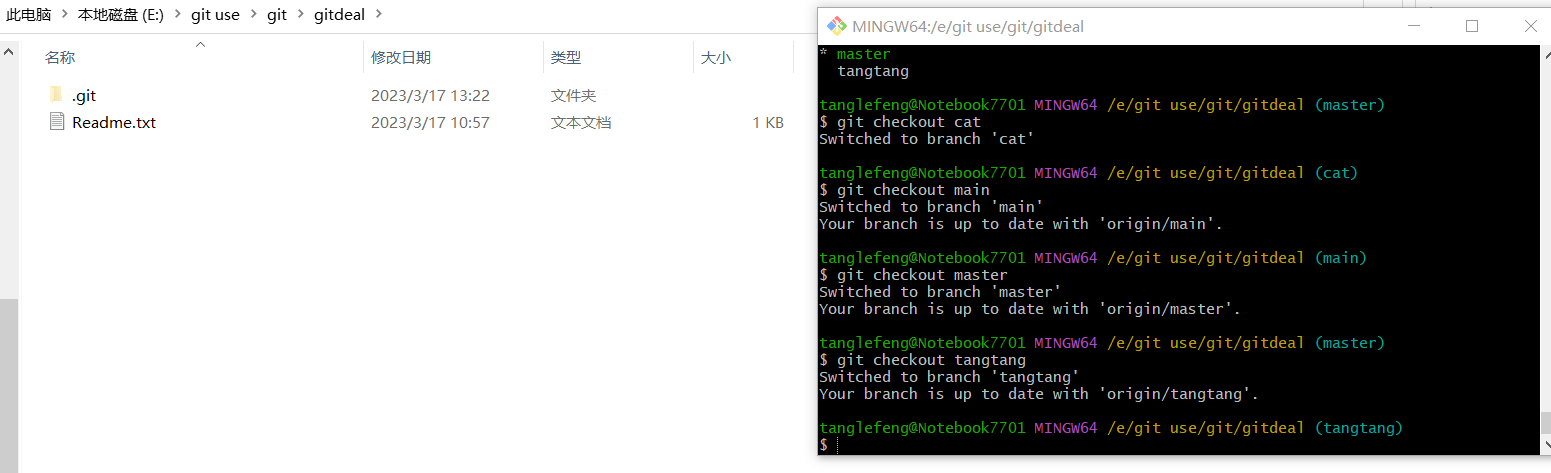
git commit -m “”

git push -u origin master 或者git push -u origin main

## Git合并冲突机制

问题1： tangtang在github和本地仓库不一致的区别





1.1)分析差异性：github上比本地仓库多一个文件夹 而且是全新的 所以冲突点在于third.txt的不一致（思考 如果readme也不一致怎么办）

1.2)解决思路：

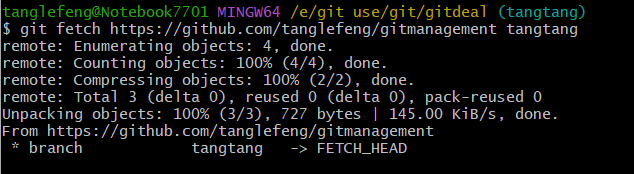
如果是保留本地仓 则 直接采用tortoistgit同步方法就可以（不推荐）

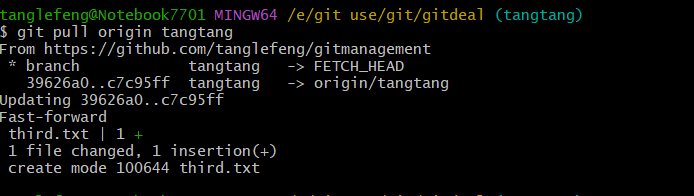
如果是跟github上一致 则采用的是fetch加pull

Fetch语法：fetch+git网址+branchname

或者git fetch [远程主机名] [分支名]

Pull语法：pull origin（默认远程主机名）+branchname





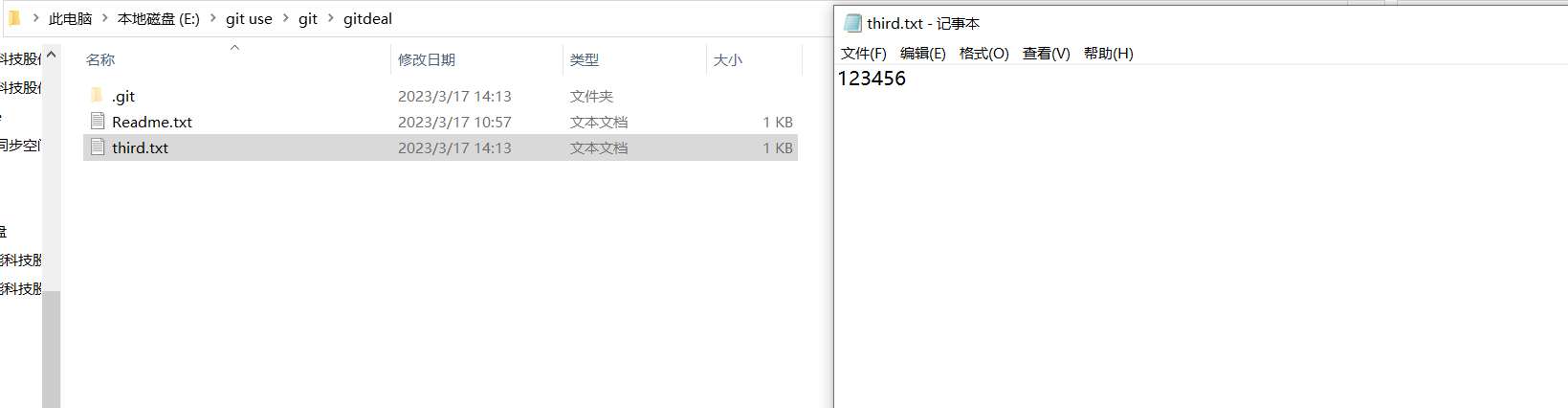
问题2：github上相同的文件third.txt不同于本地的此文件

2.1)修改思路：

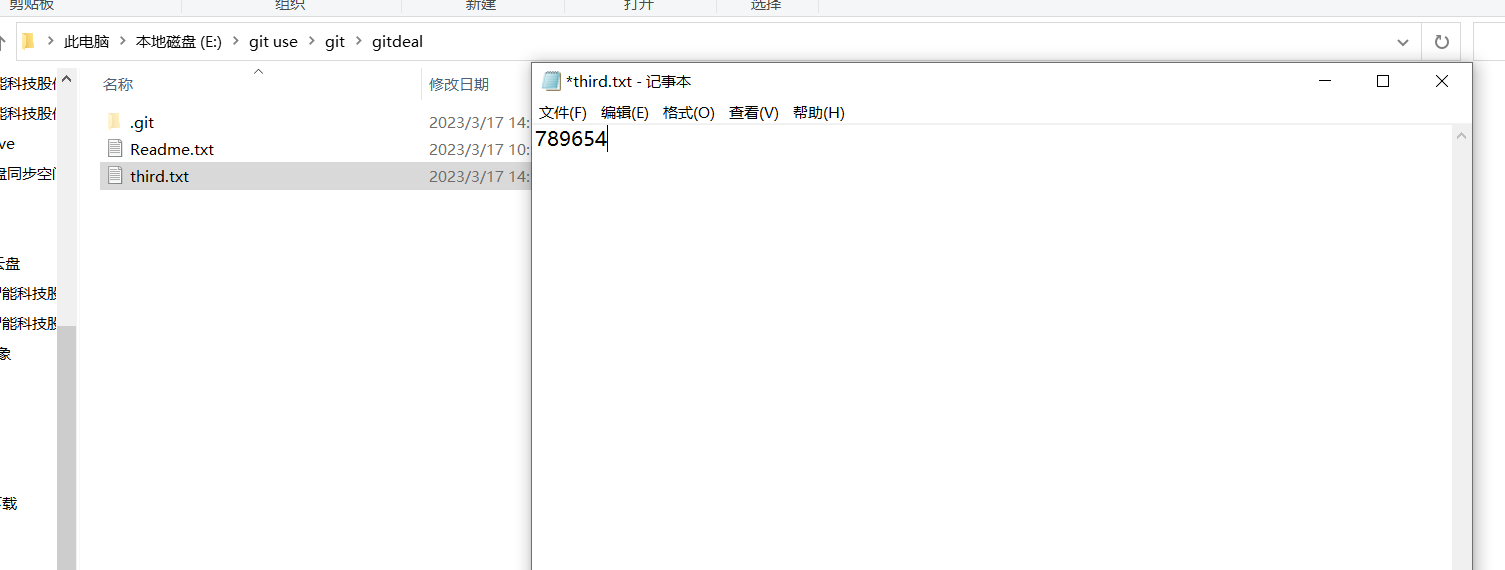
本地进行修改

2.2)跟上述一样的做法





问题3本地仓库上相同的文件third.txt不同于github上的此文件



Github上此文件仍然为123456

分析：两者不同 无外乎github进行修改数据 或者本地仓进行修改

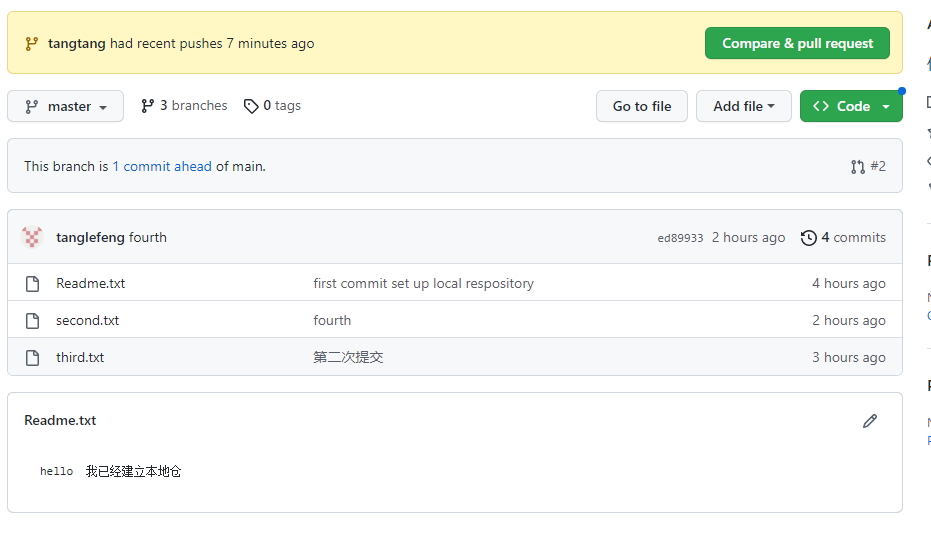
3.1)修改github仓库 不推荐

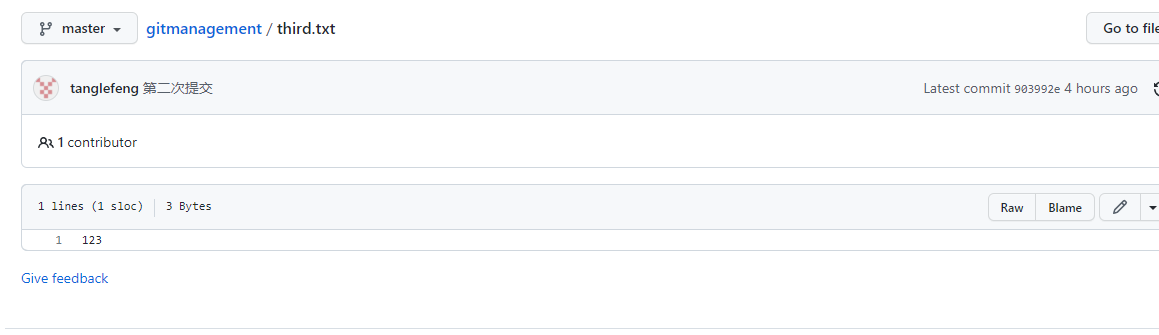
3.2)在本地仓进行修改

依然是上面的操作

4 Github上的tangtang合并到github上的master上

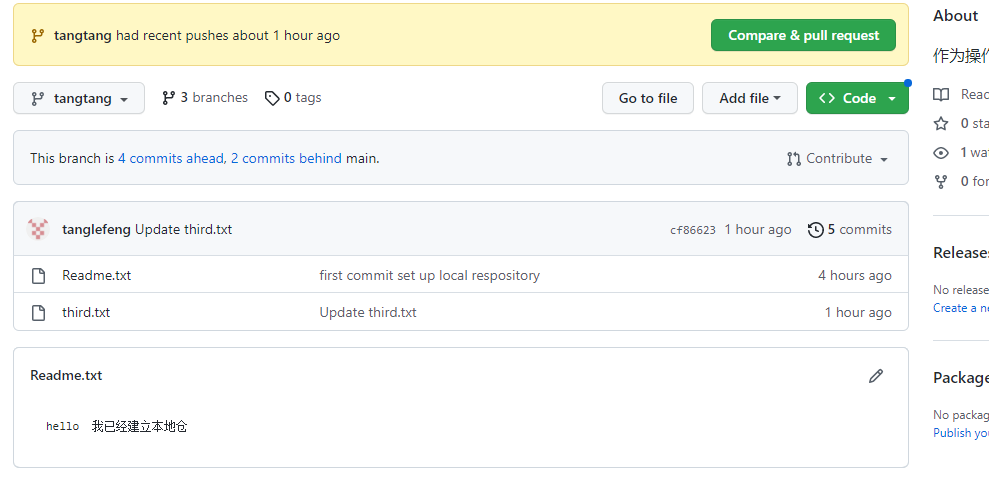
4.1)状况：目前github上tangtang跟master区别

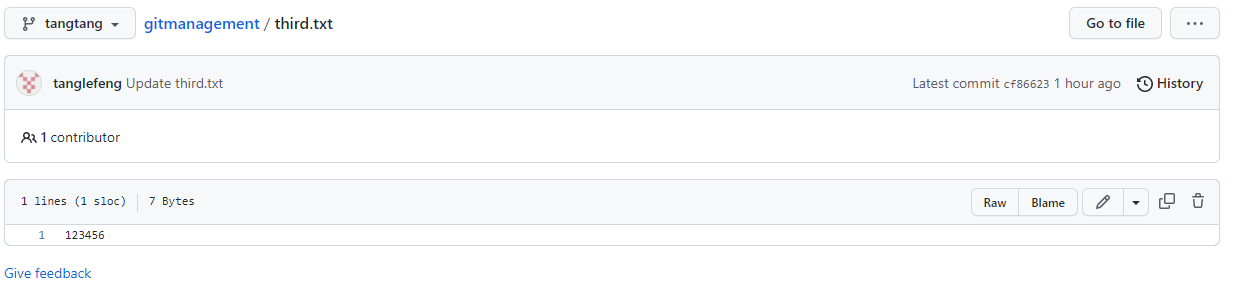




Git上的master有三个文件 Readme second third 其中third是123

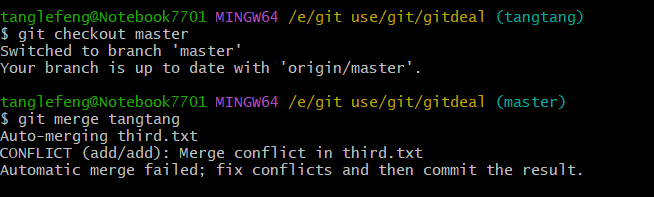
Git上的tangtang有两个文件 Readme third 其中third是123456



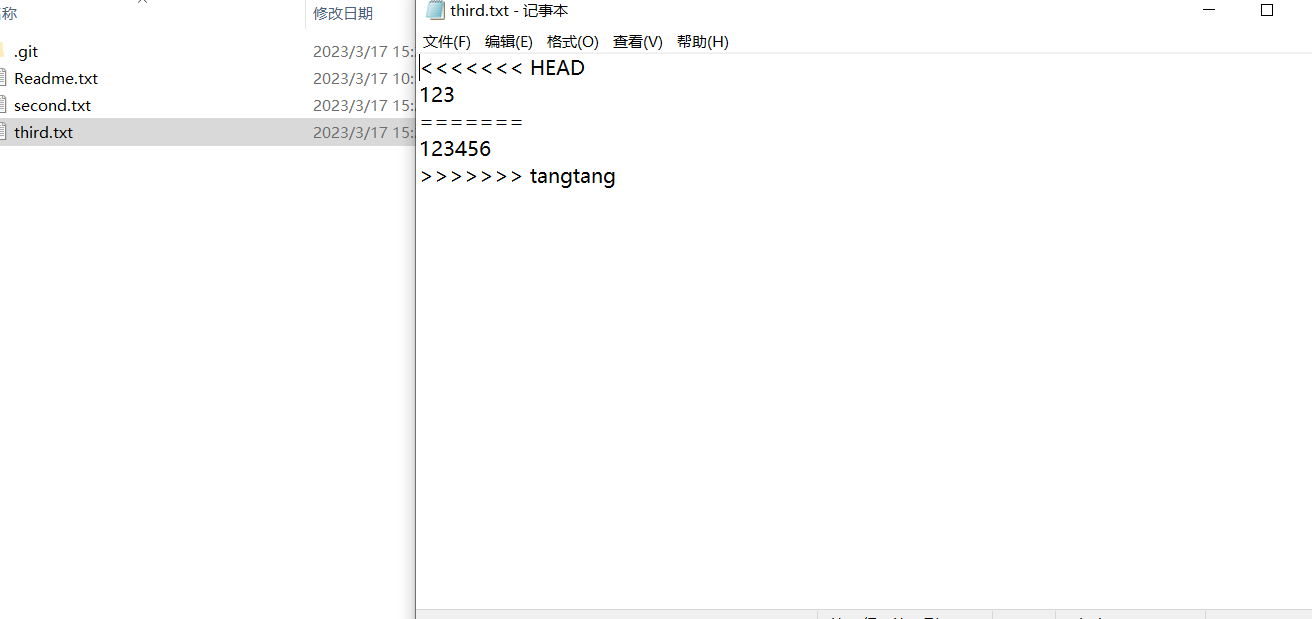


4.2)预测的问题出现：

因为third不一样 合并不了

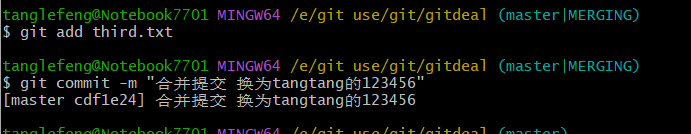


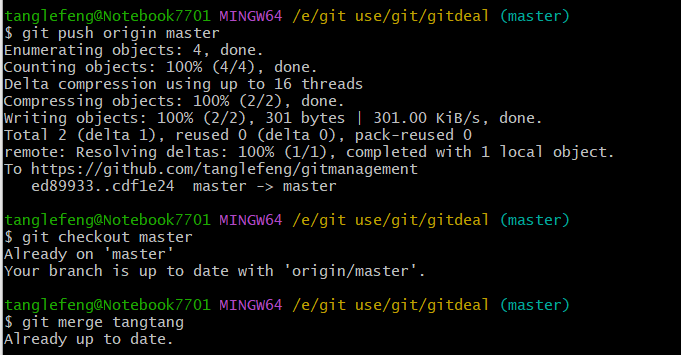
然后会看到不同



代表的含义就是tangtang的123456 要合并到master的123

这样理论上 要么改变tangtang 要么改变master 我们这次推荐使用master 即将master更改为tangtang的123456 然后提交



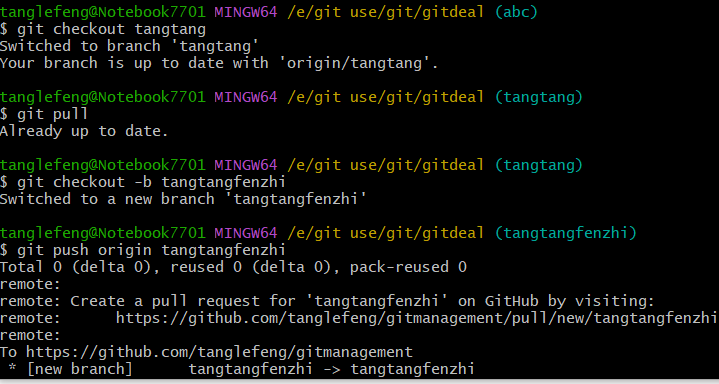


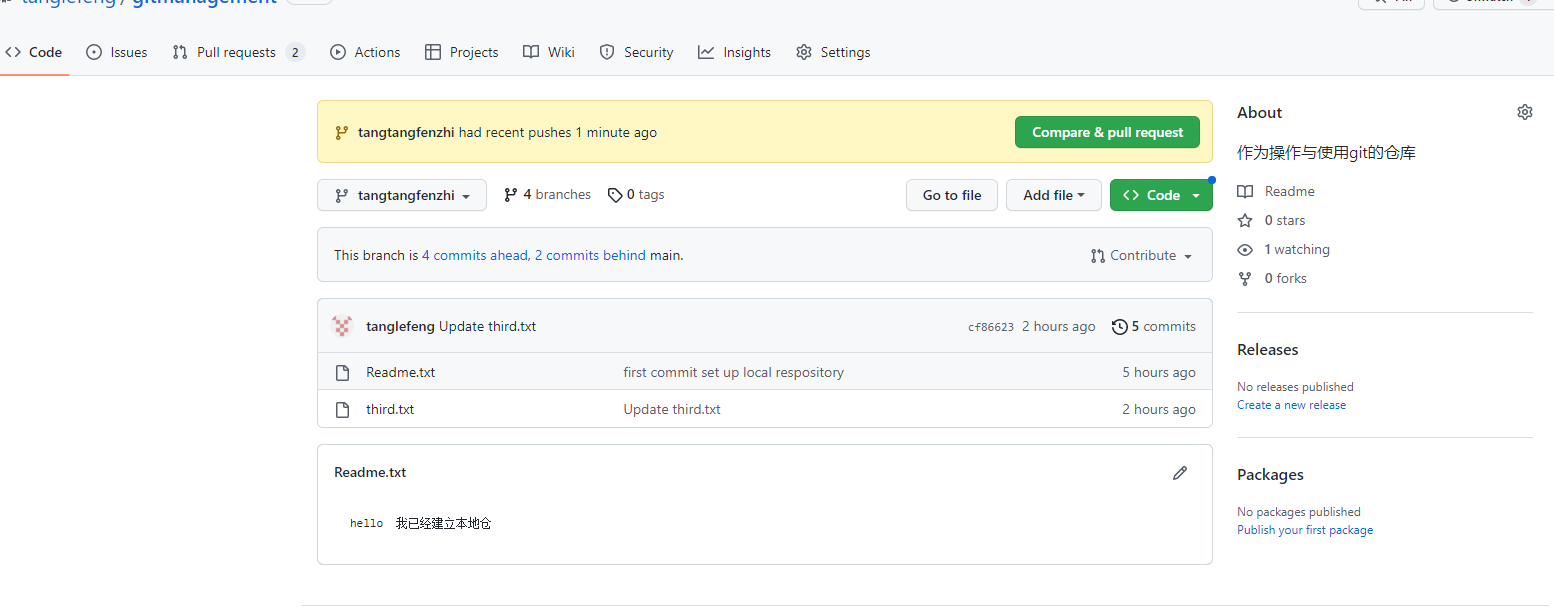


## 工具使用与指令使用

## **第五章 建立分支的分支**

**我们往往会建立一个主分支 如tangtang 然后在分支tangtang的基础上建立其他的分支进行更新**





这样就看到了

## 小结