**学习日志**

姓名： 喻子强 学号：1813046 日期： 2020/6/15

**学习内容安排：（以6.15为例）**

上午：

Git简介、安装Git、创建版本库、版本回退、工作区与暂存区、管理修改、撤销修改、删除文件

下午：

远程仓库、分支管理、标签管理、使用GitHub、忽略特殊文件、配置别名、使用SourceTree

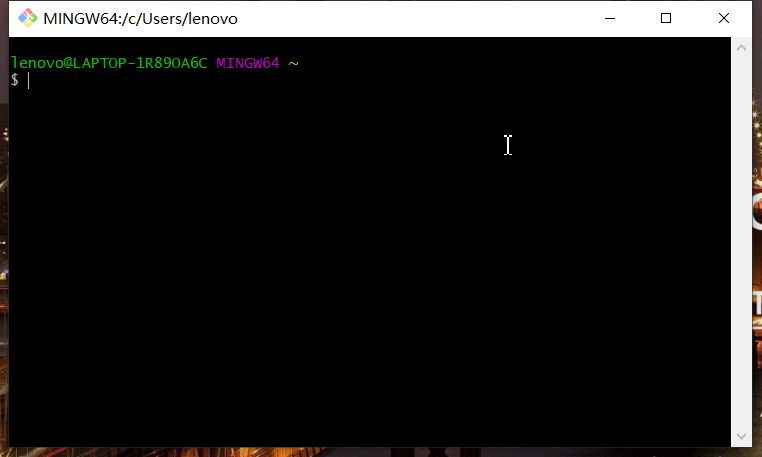
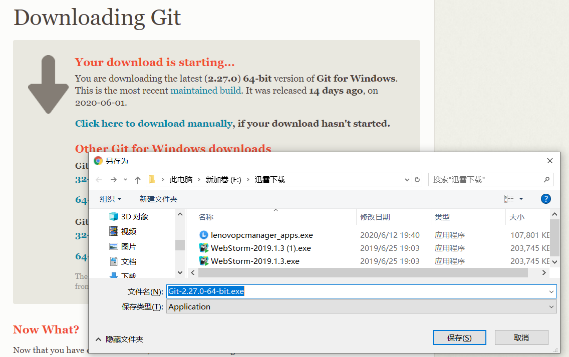
**学习反馈：**

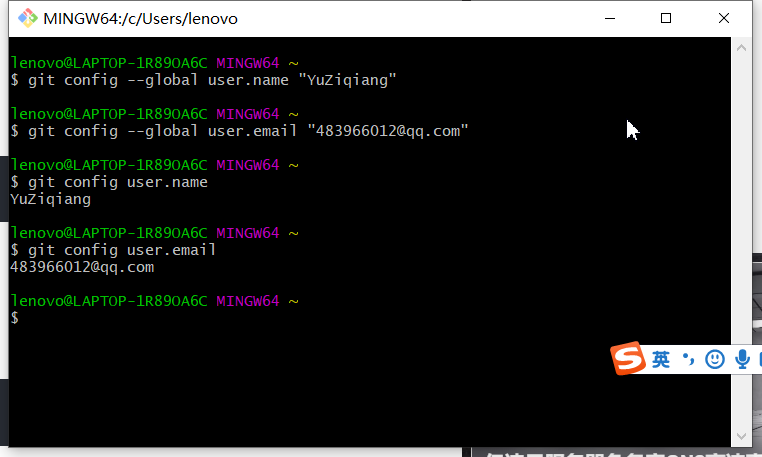
已掌握知识：

一.上午

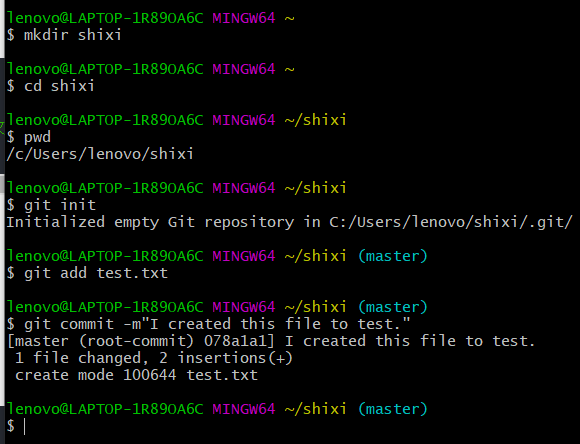
1.Git简介：Git是目前世界上最先进的分布式版本控制系统，在处理各种项目时都十分高效，它没有中央服务器，每个人的电脑就是一个完整的版本库，工作时就不需要联网了，因为版本都是在自己的电脑上。

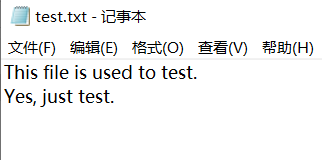
2.Git安装：在官网下载Git并安装

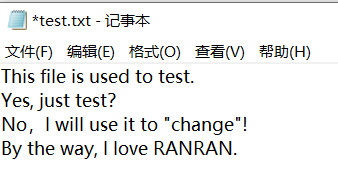


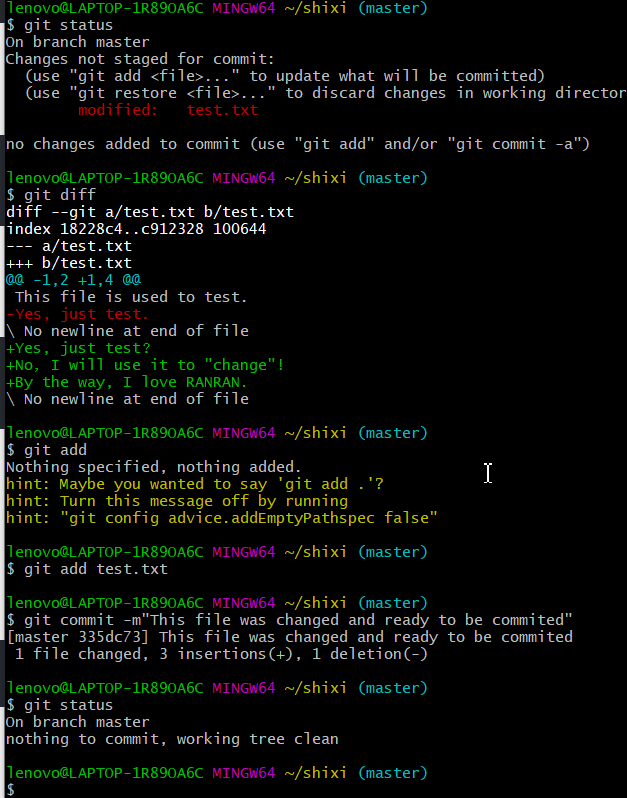
并设置用户名和邮箱并尝试查看。

3.创建版本库：版本库(repository)也叫仓库，可以看做一个目录，这个目录里的所以文件都由Git进行管理，每个文件的修改、删除，Git都能跟踪。

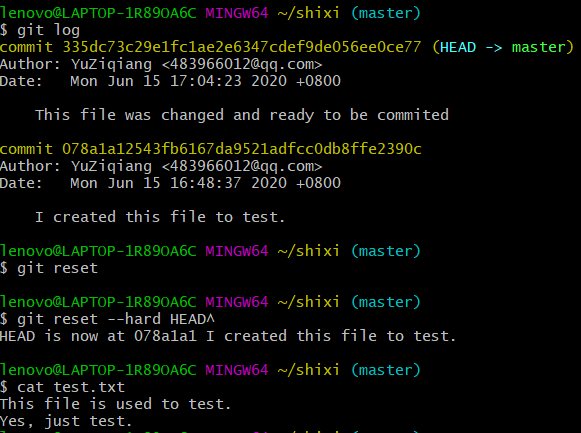
①创建空目录→将其变为Git可以管理的仓库→将自己编写的test.txt文件添加到仓库→



②修改文件：自行修改test.txt文件→用git status命令查看当前状态，发现test.txt被修改过但未提交→git diff命令查看是如何修改→提交test.txt→git status检查



4.版本回退：git log 查看历史记录→使用git reset 命令回到上一版本→cat命令查看当前状态（确实回到了，如果想回到最新版本得之前使用$ git log --pretty=oneline命令查看commit ID，之后利用git reset – hard commit ID来实现）



5.工作区和暂存区：

shixi文件夹就是一个工作区，工作区有个隐藏目录 .git ，这个不算工作区，而是 Git 的版本库，版本库里面的 index(stage) 文件叫暂存区，还有Git为我们自动创建的第一个分支 master ，以及指向 master 的一个指针叫做 HEAD。

如果我们想把文件添加到Git里面时，需要分两步：

第一步是用 git add 把文件添加进去，实际上就是把文件修改添加到暂存区。

第二步是用 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。（我们现在只有唯一一个分支 master，所以现在就是往 master 分支上提交更改）



6.管理修改：

Git 如此的优秀是因为，Git 跟踪并管理的不是文件，而是修改。

比如对 test.txt 文件进行修改，再添加，然后再修改，然后提交

发现：第二次修改并没有被提交，因为第一次修改被放入了暂存区（add），第二次没有。

提交后，我们可以用 git diff HEAD -- readme.txt 命令去查看工作区和版本库里面最新版本的区别：

7.撤销修改：

①git add之前：可以手动修改，手动把文件恢复到上一个版本的状态。然后再用 git checkout -- file 命令丢弃工作区的修改：

②git add后 但没有commit：这时候的修改添加到了暂存区，但没有提交到分支，这时候我们可以使用 git reset HEAD file 命令把把暂存区的修改撤销掉，重新放回工作区，再丢弃工作区的修改（git checkout）

③commit后：可以直接回退到上一个版本

8.删除文件：在工作区下新建文件并添加和提交到Git了，此时删除文件可使用rm命令，删除后工作区和版本库就会产生差异

①确实要从版本库中把其完全删掉：使用git rm命令，并且还git commit

②删错了，因为版本库中还有，可以恢复：git checkout – 文件名（此命令实现了：用版本库里的版本替换工作区的版本）

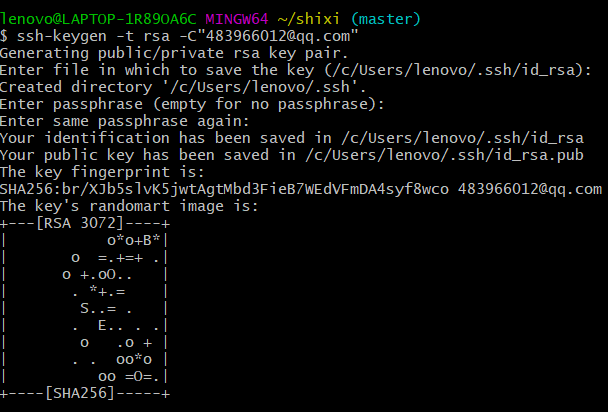
二.下午：

1.远程仓库：在开始这部分之前，我们需要自行注册GitHub账号。而且，因为

本地Git仓库和GitHub仓库之间的传输是通过SSH加密的，所以需要设置：

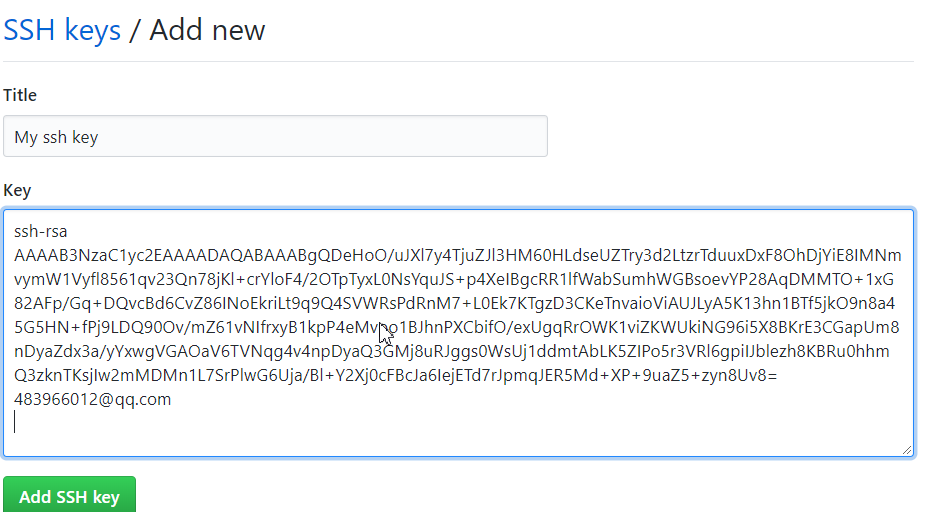
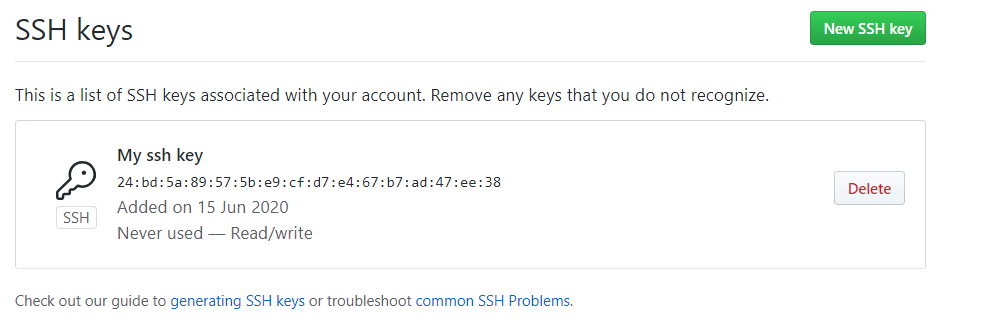
①检查是否有SSH：

可见是没有的

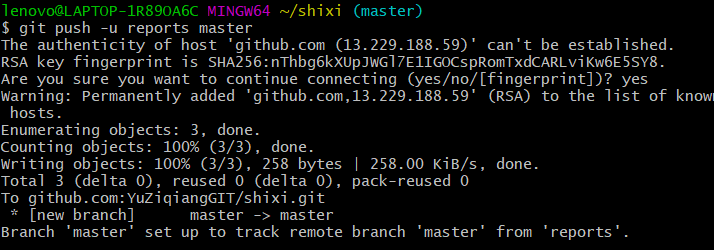
②创建SSH：

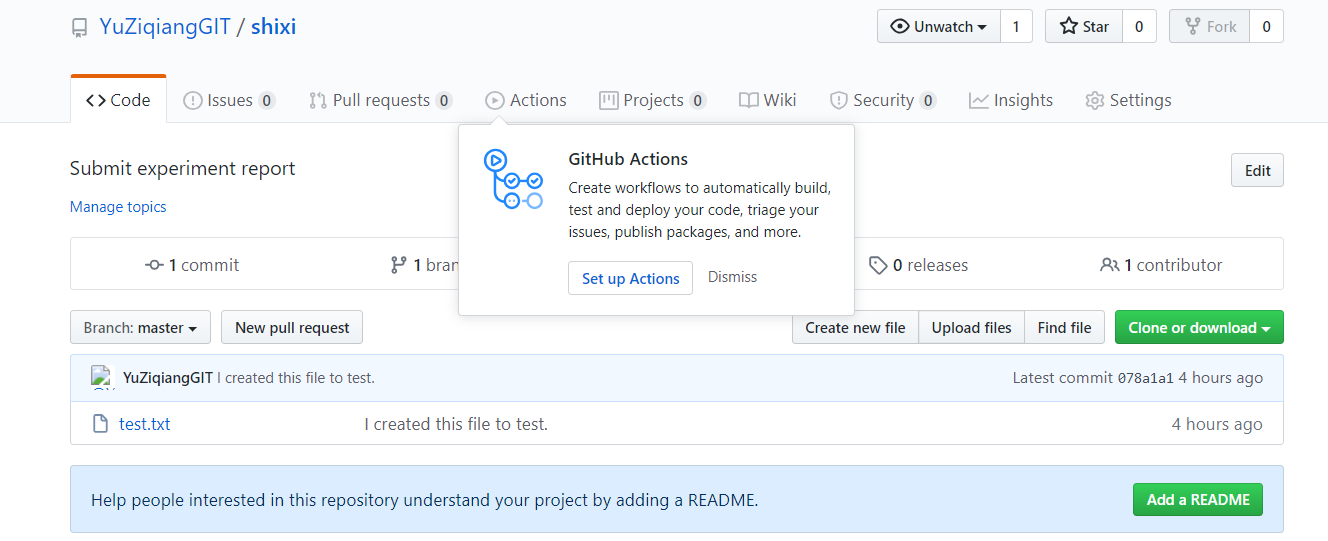
使用$ ssh-keygen -t rsa -C “您的邮箱地址” 这一命令，并且输入密码时输入为空，这样之后提交时则不需要输入密码。

③在Github网站上填写id-rsa.pub的内容

④添加远程库：现在我们已经在本地创建了一个Git仓库了，又想在GitHub上创建一个Git仓库，然后让这两个仓库进行远程同步，这样，GitHub上的仓库既可以作为备份，又可以让其他人通过该仓库来协作。

通过一系列操作成功将本地的shixi中的test.txt文件push到我的github账号上的“reports”库



2.分支管理：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严格来说不是指向提交，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。

一开始的时候，master分支是一条线，Git用master指向最新的提交，再用HEAD指向master，就能确定当前分支，以及当前分支的提交点。

每次提交，master分支都会向前移动一步，这样，随着你不断提交，master分支的线也越来越长。

当我们创建新的分支，例如dev时，Git新建了一个指针叫dev，指向master相同的提交，再把HEAD指向dev，就表示当前分支在dev上。

Git创建一个分支很快，因为除了增加一个dev指针，改改HEAD的指向，工作区的文件都没有任何变化。

不过，从现在开始，对工作区的修改和提交就是针对dev分支了，比如新提交一次后，dev指针往前移动一步，而master指针不变。

假如我们在dev上的工作完成了，就可以把dev合并到master上。Git怎么合并呢？最简单的方法，就是直接把master指向dev的当前提交，就完成了合并。

所以Git合并分支也很快！就改改指针，工作区内容也不变！

合并完分支后，甚至可以删除dev分支。删除dev分支就是把dev指针给删掉，删掉后，我们就剩下了一条master分支。

3.标签管理：

切换到需要打标签的分支上

（1）新建一个标签（默认为HEAD，也可以指定一个commit id）

（2）标签不是按时间顺序列出，而是按字母排序的。

git tag查看所有标签，用git show 标签名查看指定的某个标签信息

（3）删除标签

创建的标签都只存储在本地，不会自动推送到远程，所以打错的标签可以在本地安全删除：$ git tag -d <标签名>

（4）推送标签到远程

推送某个标签到远程 git push origin <tagname>

推送全部尚未推送到远程的本地标签 git push origin –tags

4.忽略特殊文件：

①. 在git操作的根目录(即最高一级可以使用git命令的仓库) 建立一个名为.gitignore的隐藏文件

　　 ②.将希望忽略的文件/文件夹加入到.gitignore文件中，使用vi/vim都可以，但是开头记得不能有空格

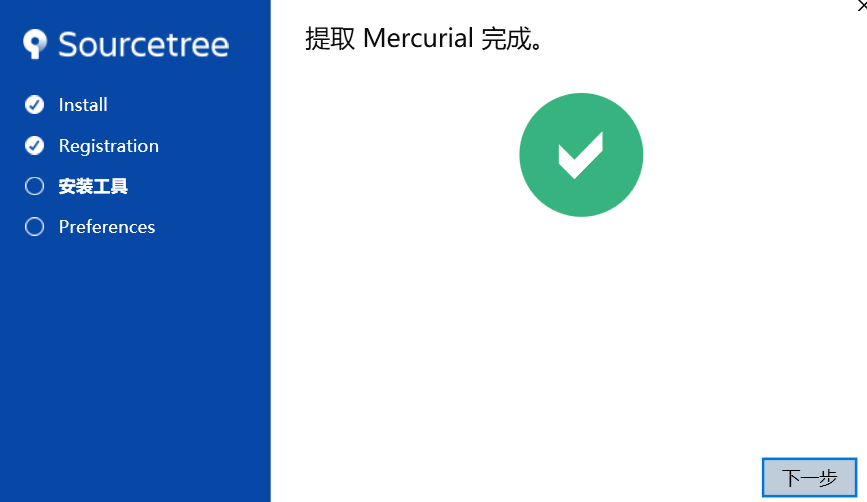
5.配置别名：

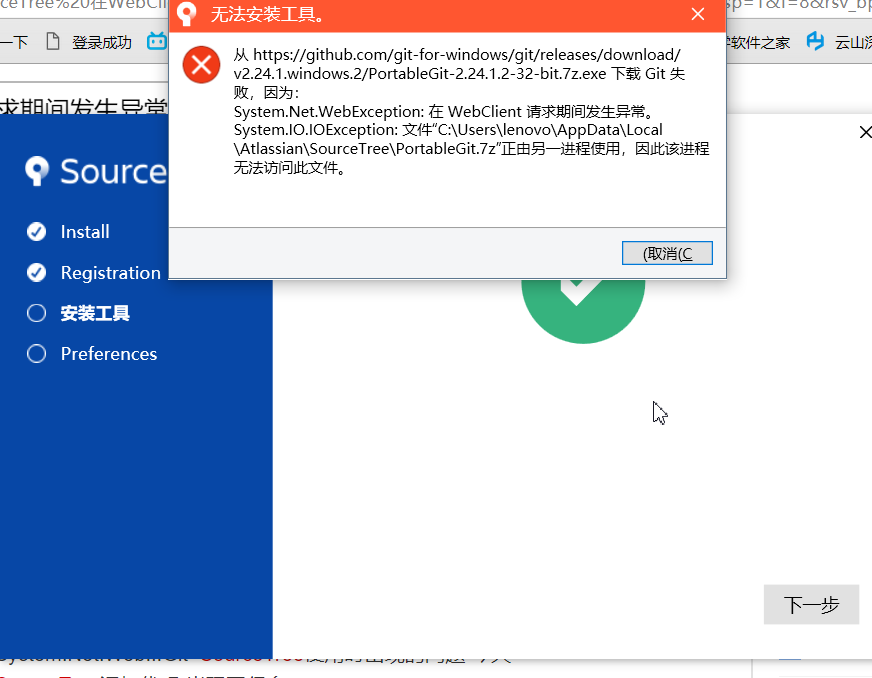
利用Git自带的别名功能：具体方式：git config --global alias.别名 命令

因为设置级别为global，即用户级别，windows下git会将别名写入C盘用户的.gitconfig中，如果不加global条件，则默认写入仓库的config中，只针对当前仓库有效。当然也可以手动在配置文件中按照格式要求写入别名。

6.SourceTree的安装与使用：

SourceTree是一个用于Windows和Mac的免费Git客户端。  
Sourcetree简化了如何与Git存储库进行交互，这样就可以集中精力编写代码。通过Sourcetree的简单Git GUI可视化和管理存储库。

①先去官网安装：

在此遇到问题无法安装

目前网上有关SourceTree的资料较少，难以找到解决方案，暂时存疑

未掌握知识：SourceTree的安装与使用

学习心得：

1.Git 如此的优秀是因为：Git 跟踪并管理的不是文件，而是修改。

2.分支系统十分优越先进。