主题：日报

世间：2018年7月23日星期一

内容：

1. git的基础内容
2. 创建自己的项目，用自己的名字创建分支
3. 上传自己的学习计划

## 一 git简介

1. git是什么？

git 是目前世界上最先进的分布式版本控制系统

（2）集中式版本控制系统和分布式版本控制系统

1. 集中式版本控制系统

版本库是集中放在中央服务器的而干活的时候，用的都是自己的电脑，所以要先从中央服务器取得最新的版本，然后开始干活，干完活了，再把自己的活推送给中央服务器。

缺点：必须联网才能使用。

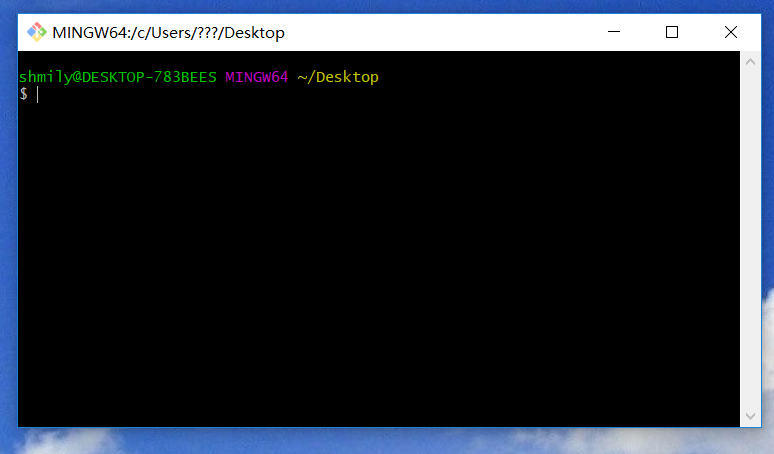
B．分布式版本控制系统

分布式版本控制系统根本没有“中央服务器”，每个人的电脑上都是一个完整的版本库，这样，你工作的时候，就不需要联网了，因为版本库就在你自己的电脑上。

优点：安全性能比较高

（3）git的安装

在Windows上使用Git，可以从Git官网直接[下载安装程序](https://git-scm.com/downloads" \t "_blank)，然后按默认选项安装即可。安装完成后，在开始菜单里找到“Git”->“Git Bash”，蹦出一个类似命令行窗口的东西，就说明Git安装成功！如图：类似窗口：



## 二．创建版本库

（1）什么是版本库

版本库又名仓库，英文名**repository**，你可以简单理解成一个目录，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可以“还原”。

（2）创建

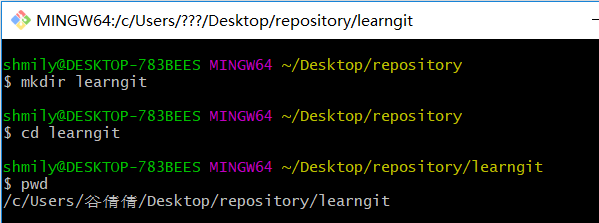
第一步，创建一个文件

**$ mkdir  learngit**

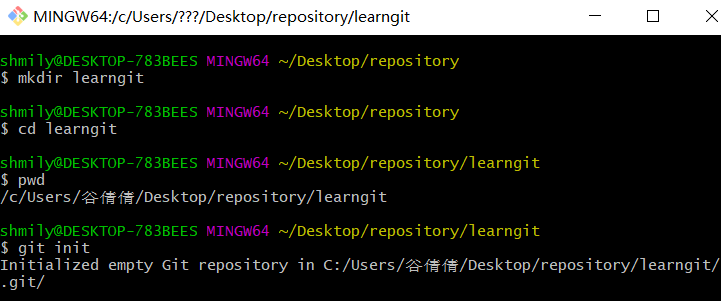
**$ cd learngit**

如果想知道路径可以运行 **$ pwd**来查看

如图：



第二步，同构git init命令把这个目录变成git可以管理的仓库



瞬间Git就把仓库建好了，而且告诉你是一个空的仓库（empty Git repository），

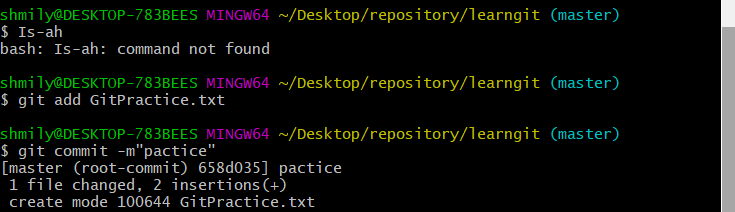
这个时候运行**ls -a**命令可以看到会有一个**.git**的隐藏目录，隐藏的肯定是不建议我们去修改的，这个目录是Git用来跟踪管理版本库的，因此我们不要去改动而造成混乱。

第三步，将文件添加到版本库

首先这里再明确一下，所有的版本控制系统，其实只能跟踪文本文件的改动，比如TXT文件，网页，所有的程序代码等等，Git也不例外。版本控制系统可以告诉你每次的改动，比如在第5行加了一个单词“Linux”，在第8行删了一个单词“Windows”。而图片、视频这些二进制文件，虽然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，也就是只知道图片从100KB改成了120KB，但到底改了啥，版本控制系统不知道，也没法知道。不幸的是，Microsoft的Word格式是二进制格式，因此，版本控制系统是没法跟踪Word文件的改动的，前面我们举的例子只是为了演示，如果要真正使用版本控制系统，就要以纯文本方式编写文件。因为文本是有编码的，比如中文有常用的GBK编码，日文有Shift\_JIS编码，如果没有历史遗留问题，**强烈建议使用标准的UTF-8编码，**所有语言使用同一种编码，既没有冲突，又被所有平台所支持。

添加文件到Git仓库，分两步：

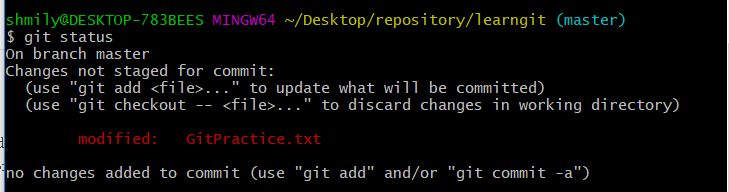
1. 使用命令git add <file>，注意，**可反复多次使用，添加多个文件**；
2. 使用命令git commit -m “”(双引号内填写你本次提交的说明)，完成。



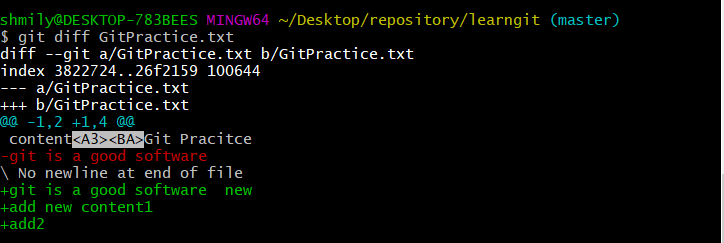
git commit命令执行成功后会告诉你，1 file changed：1个文件被改动（我们新加的readme.txt文件）；2 insertions：插入了两行内容（GitPractice.txt有两行内容）。

修改GitPractice.txt文件的内容

使用git status 查看（提醒我们GitPractice）.txt已经被修改过了 但看不到改了什么）

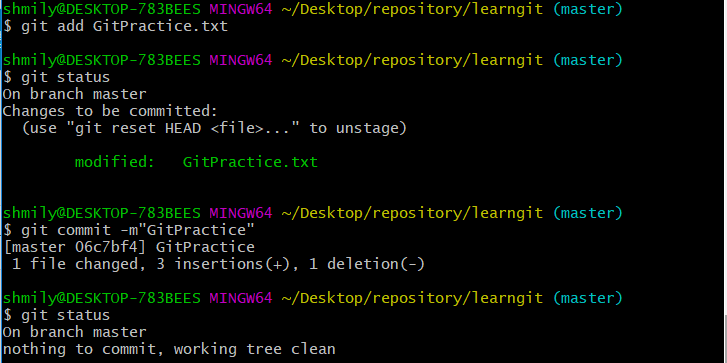


使用 git diff命令查看修改了什么内容



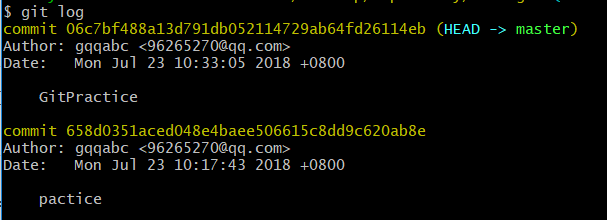
然后提交修改后的文件和新文件的方法是相同的

使用git add和git commit -m””命令最好在执行git commit命令之前执行git status查看文件状态然后进行提交 提交后执行git status 查看状态



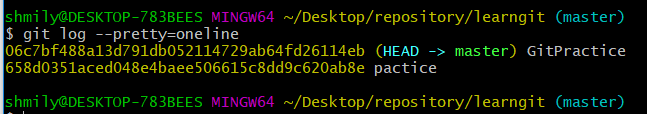
working tree clean（工作目录是干净的）

刚才我们对GitPractice文件进行了多次修改 我们可以使用git log 命令进行查看



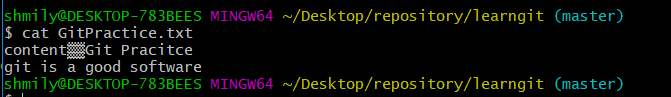
我们可以看到提交日期和提交内容 一共提交了两次 分别为：GitPractice 和 pactice

如果嫌输出信息太多，看得眼花缭乱的，可以试试加上--pretty=oneline参数：

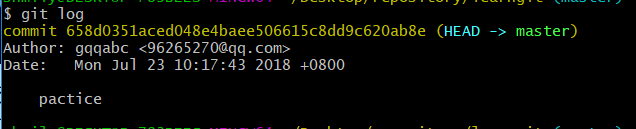


当前版本为 pactice 那我们可以使用 **git reset**命令（由当前版本pactice 退回到GitPractice版本）

通过cat查看



**git log** 查看版本库状态 发现最新版本的GitPracitce 已经没有了

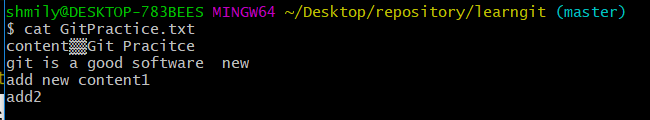


同样也可以再回到最新的状态

方法1：向上翻命令行找到GitPractice的**commit**对应的id使用git reset id的输入只要前几位就可以了（不可以只有一两位 系统会找打很多个文件 无法确定是哪一个）

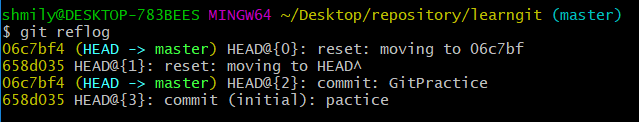


然后我们来查看一下此时的GitPractice。txt的内容



那如果找不到之前的commit命令的ID 使用方法2

方法2：命令git reflog会记录每一次命令



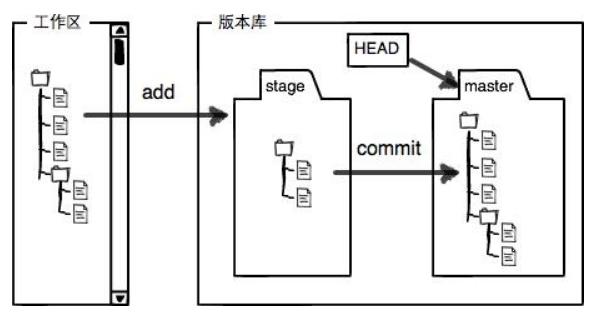
这样就可以找到对应的id了，然后再使用方法一就可回到最新状态了

## 三 工作区和暂存区

工作区：就是在电脑里能看到的目录，比如某个文件夹就是一个工作区

版本库（Repository）：工作区有一个**隐藏目录.git**，这个不算工作区，而是Git的版本库。

Git的版本库里存了很多东西，其中最重要的就是称为**stage（或者叫index）的暂存区，**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HEAD。



Git管理的是修改，不是文件

git checkout – file 把文件在工作区的修改全部撤销掉

git reset HEAD file 把暂存区的修改撤销掉（unstage），重新放回工作区

文件从版本库中删除的步骤（现在文件（flie）从版本库中删除了）：

1. rm flie
2. git status
3. git rm
4. git commit -m

## 四．远程仓库

#### A.关联远程仓库

本地Git仓库和GitHub仓库之间的传输是通过SSH加密的，

第1步：创建SSH Key

$ ssh-keygen -t rsa -C "**邮件地址**"

.ssh目录，里面有id\_rsa（**私钥**）和id\_rsa.pub（**公钥**）

第2步：登陆GitHub，打开“Account settings”，“SSH Keys”页面：

点“Add SSH Key”，填上任意Title，在Key文本框里粘贴id\_rsa.pub文件的内容

将本地的git仓库和github的仓库进行远程同步

$ git remote add origin [git@github.com:michaelliao/learngit.git](mailto:git@github.com:michaelliao/learngit.git)（改成自己的）（添加后，**远程库的名字就是origin**）

关联一个远程仓库**git remote add origin** git@server-name:path/repo-name.git；

关联后，使用命令**git push -u origin master**第一次推送master分支的所有内容；

#### B.远程仓库克隆

第一步，登陆Github，创建一个新的仓库（勾选Initialize this repository with a README）

第二步，用命令git clone克隆一个本地仓库

## 五．分支管理

#### A创建分支

git checkout -b dev(分支名)（-b表示创建并切换）

等同于

git branch dev

git checkout dev

查看当前分支 git branch(\*为当前所在分支)

#### B合并分支

git merge dev分支名 （命令用于合并指定分支到当前分支）

合并后删除分支git branch -d dev（删除dev分支）

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>