上课笔记

安装git

创建版本库

- 创建文件夹-----git bash
  - mkdir 文件名
  - cd 文件名
  - git init ----本地有了一个版本库

第一次使用这个git时候：

- ```
  $ git config --global user.name "Your Name"
  $ git config --global user.email "email@example.com"
  ```

把这个文件添加到版本库

告诉我们这个版本控制系统它能追踪文件的变更

> 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

注意不要用windows自带的记事本文件来操作

git添加到git仓库的几步：

1.git add 文件名

2.git commit -m "说明，描述语"

为什么Git添加文件需要`add`，`commit`一共两步呢？

git add 一次可以添加多个文件以及可以连续添加多次

git commit 一次可以提交多个文件

常见的时间管理

当我们对这个已经被版本仓库管理的文件，做了修改，但是没有提交，这时候，我们可以
  ```

用这个查看文件状态的命令 -------git status

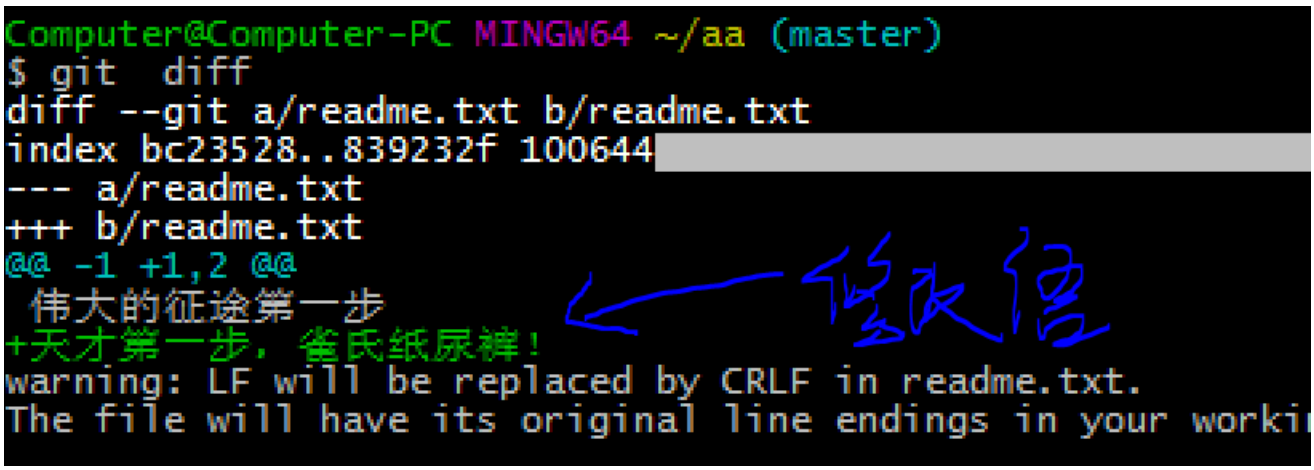
```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   readme.txt   --表示文件已经修改了

no changes added to commit (use "git add" and/or "git commit -a")

Computer@Computer-PC MINGW64 ~/aa (master)
```

使用git diff 查看修改的内容



查看历史的修改记录 git-log

```
$ git log
commit ae0c1e36193f862869a8ba50c6be7f38b3b337cc
Author: wulibobo <liubo942094@foxmail.com>
Date:   Fri Jan 11 09:46:14 2019 +0800

    加了点东西

commit 53642dba88f1ac11268cd32c0a88e1c54f352aba
Author: wulibobo <liubo942094@foxmail.com>
Date:   Fri Jan 11 09:21:05 2019 +0800

    向这个仓库提交一个readme.txt文件
```

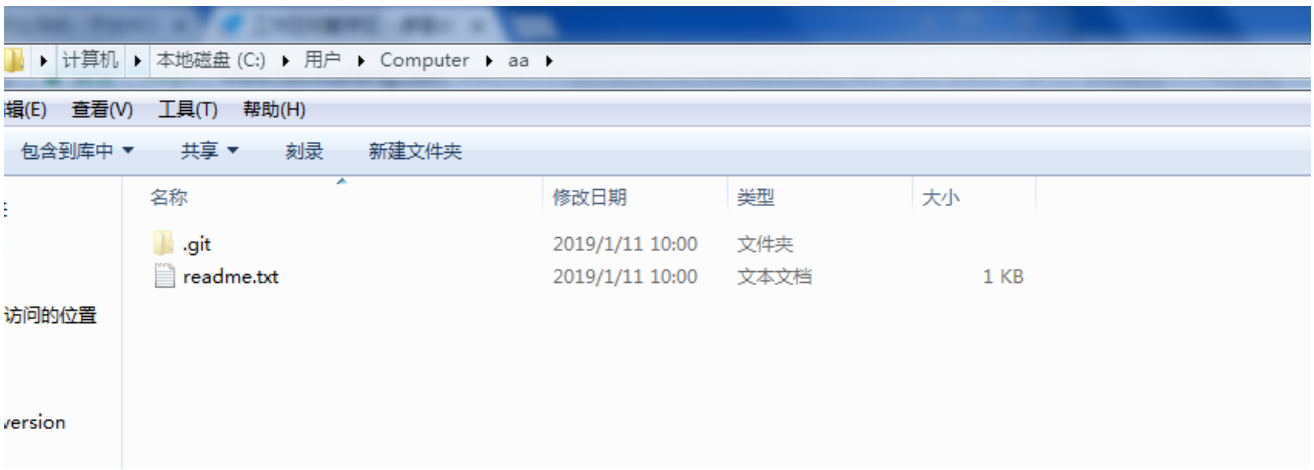
git的版本回退：

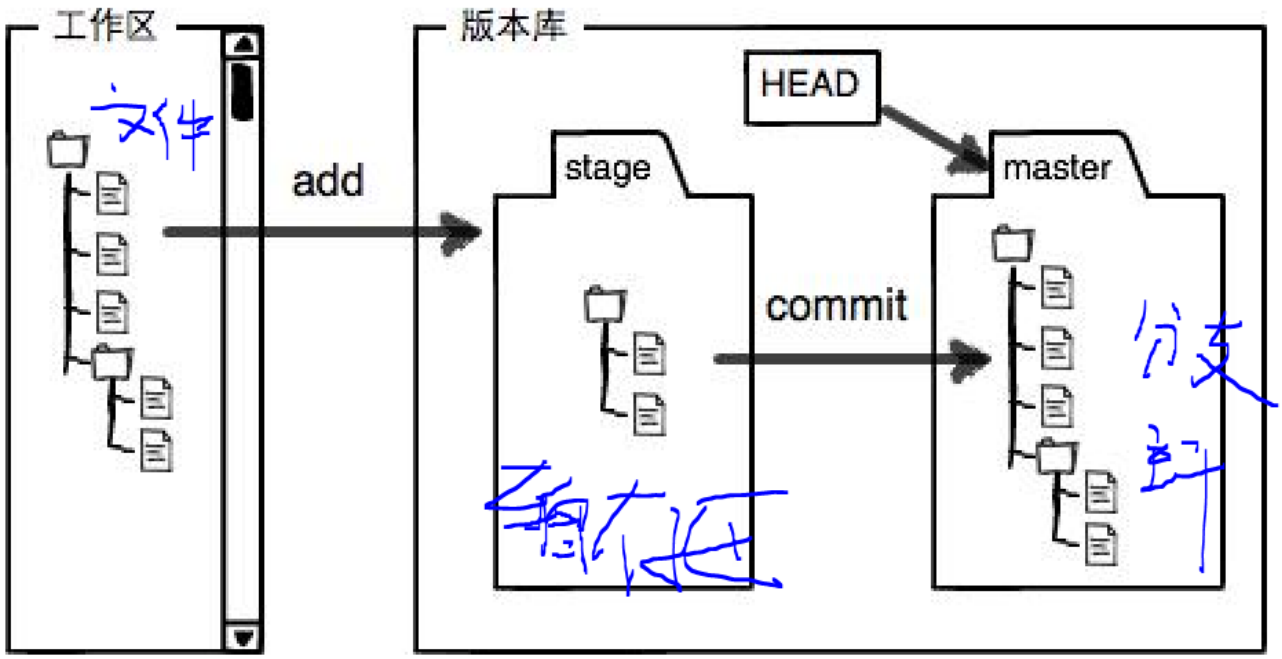
```
git  reset   --hard HEAD^
```

注意：HEAD^ 这个^符号一个表示回退一次两个回到上上次

工作区和暂存区

工作区：电脑本地仓库所在的文件夹

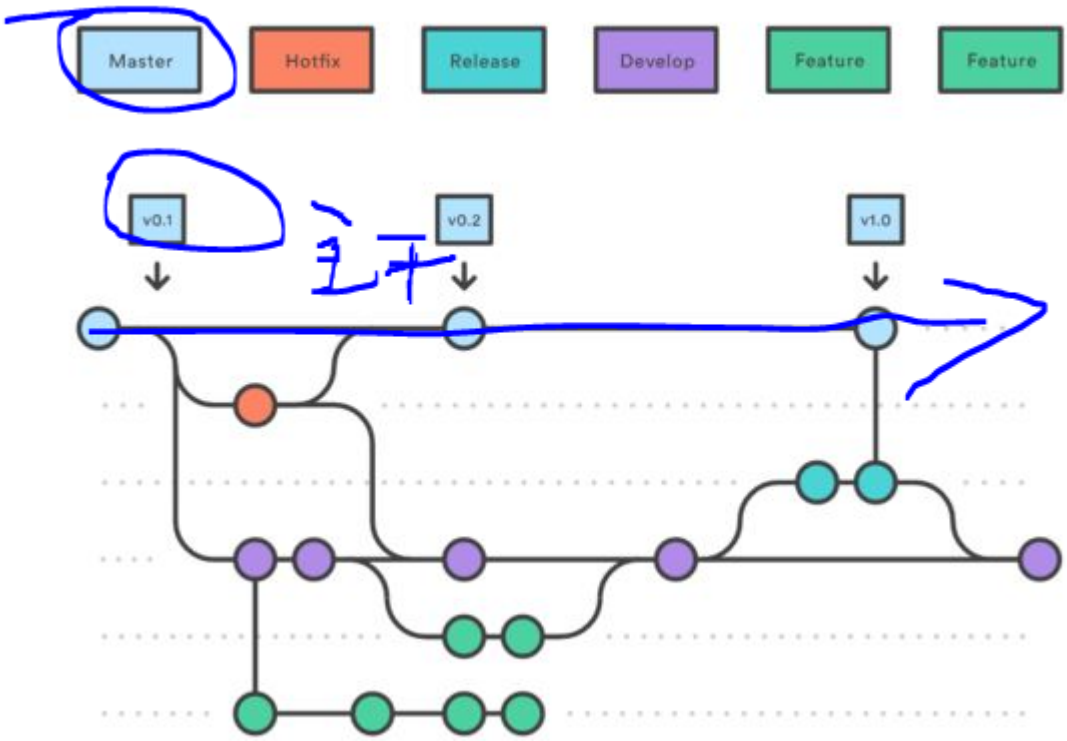




第一步是用git add把文件添加进去，实际上就是把文件修改添加到暂存区；

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。

git的结构图




```
Computer@Computer-PC MINGW64 ~/aa (master)
$ git add readme.txt  tt.txt
warning: LF will be replaced by CRLF in tt.txt.
The file will have its original line endings in your working directory.

Computer@Computer-PC MINGW64 ~/aa (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   readme.txt
        new file:   tt.txt

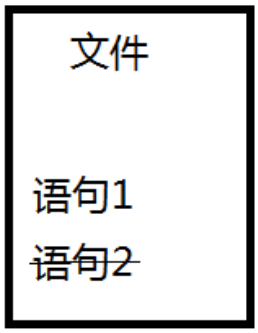
Computer@Computer-PC MINGW64 ~/aa (master)
$
```

文件内容撤销 git checkout -- fileName

当文件添加到暂存区（也就是被add了，但没有commit）,如果想删除掉，分两步：

1.手动删除掉

2.git checkout -- 文件名



手动删除语句2，然后直接git checkout -- 文件名
这样的操作实际上是对手动删除语句2做了一个撤回
手动删除语句2，git add 一下

从暂存区撤退到工作区

git reset HEAD 文件名

当我们把一个文件直接 删除掉了，也能撤销回来

```
rm  文件名

git  checkout  -- 被删除的文件名
```

上面这种删除法，是一个伪删除，工作区删除了，但是版本库里面没有删除掉，依然还能撤销回来，类似windows 删除文件，但回收站还有

```
git  rm 文件名
git   commit  -m "说明"
```

上面这种是永久删除了！ 从回收站里面清空了

远程操作：

1.在自己搭建一个git 服务器

2.远程连接别人提供的git 服务器

> - 连接github

3.自行注册github账户

4.配置SSH Key

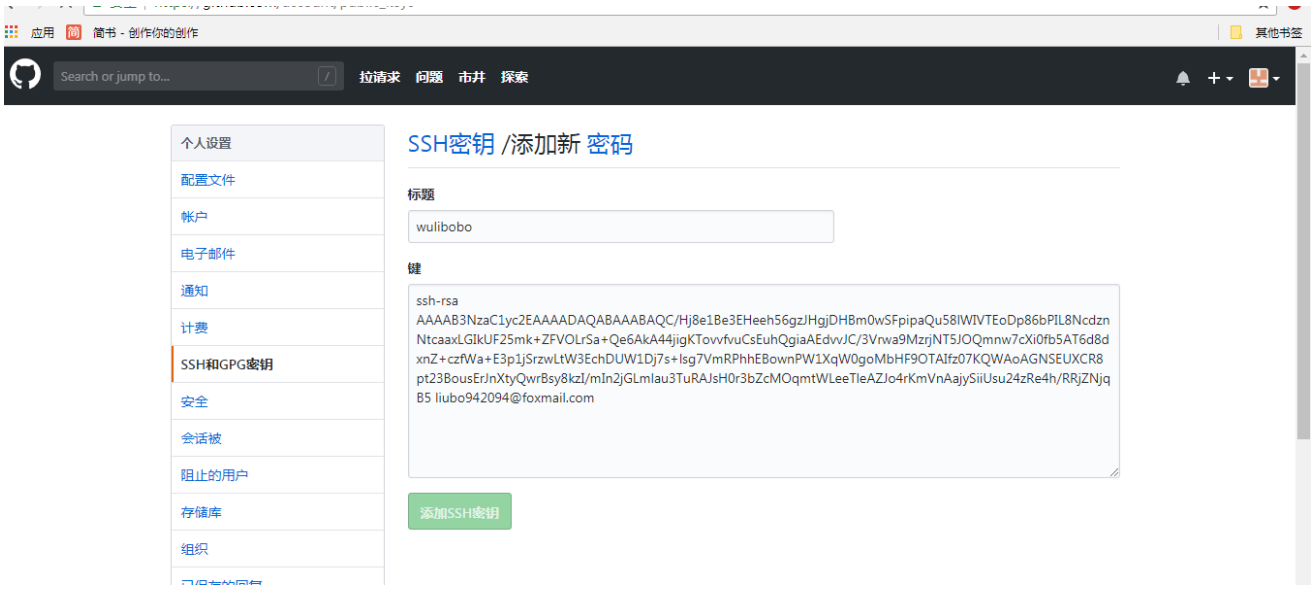
```
由于你的本地Git仓库和GitHub仓库之间的传输是通过SSH加密的

SSH是一种协议
http协议
TCP/ip协议

SMTP协议
```

- 第1步：创建SSH Key
  - 在用户主目录下，看看有没有.ssh目录，如果有，再看看这个目录下有没有`id_rsa`和`id_rsa.pub`这两个文件，如果已经有了，可直接跳到下一步。如果没有，打开Shell（Windows下打开Git Bash），创建SSH Key：
  - ssh-keygen -t rsa -C "写你自己的邮箱"
  - 然后一路回车，然后再去找 .ssh文件（注意：这个.ssh跟aa是同级目录）

- 第2步：登录你的github



添加远程仓库

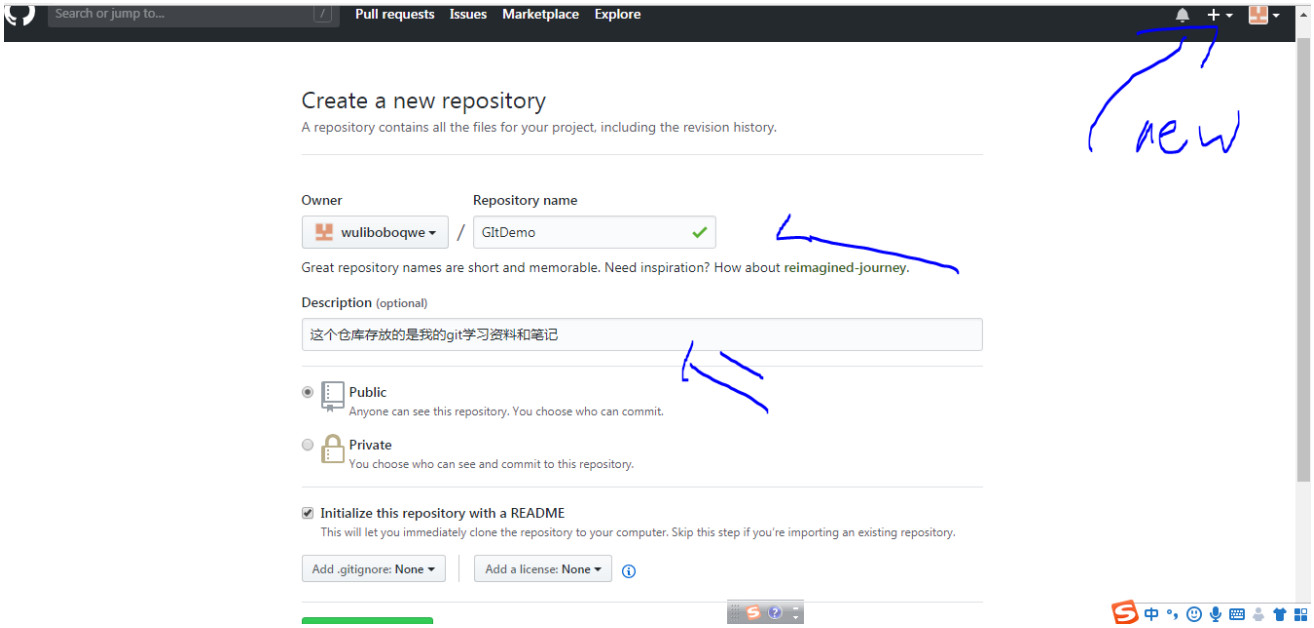
为什么要添加一个远程仓库？

现在的情景是，你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，真是一举多得。

1.创建一个GitHub的仓库

实在不会，参考链接[如何在github创建一个仓库]



2.拿本地的仓库和github上面的仓库关联

```
git remote add origin git@github.com:wuliboboqwe/GItDemo.git

origin：远程
wuliboboqwe：写自己的用户名
GItDemo：远程的仓库名
```

3.把所有的内容推送到github上

```
git  push -u origin  master  --第一次推送

git push origin master    --以后推送上来

这样就能够保证git的本地仓库和远程仓库连接一致！
```

4.从这个github抓取下载下来

```
git  clone +  URL
```

插入一段别的

```
关于这个md文件
    全称叫markdown语法

    关于这个md的写作软件
    typora  ----开源的
    马克飞象  ----印象笔记下面的
    ....
```