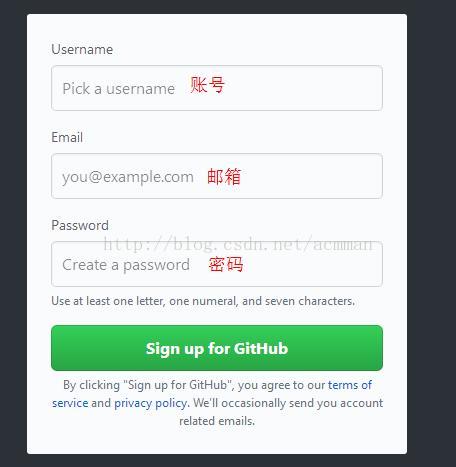
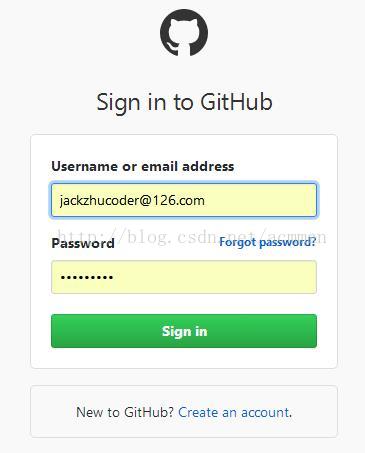
GitHub仓库及本地Git介绍说明

1. 概念及关系说明：
2. 目前我们接触的与Git有关系的有三种，分别是Git、GitHub以及GitLab。
   * 1. Git--就是最原始的分布式版本控制系统，是开源的，是一个单独的软件。
     2. GitHub--与Git不同的是，Git是一个单独的软件，而GitHub是一个网站。GitHub是目前最大的版本控制社区网站，用户可以在上面申请账号，并创建自己版本仓库。不过对于免费的仓库，用户必须公开自己仓库中的所有代码，而付费仓库可以上传私密代码。
     3. GitLab是一款单独的软件，在Git的基础上搭建起来的Web服务。它可以创建私有库，解决了GitHub的免费仓库必须公开源码的问题。一般的企业或者工作室开发者会选择使用它。
3. GitHub社区的使用
4. 注册GitHub账号：

* 首先我们需要在GitHub上注册一个账号，gitHub官网地址为：[https://github.com/](https://github.com/%E3%80%82%E5%A6%82%E6%9E%9C%E4%B9%8B%E5%89%8D%E6%B2%A1%E6%9C%89%E7%99%BB%E5%BD%95%E8%BF%87%EF%BC%8C%E9%82%A3%E4%B9%88%E5%9C%A8%E9%A6%96%E9%A1%B5%E5%B0%B1%E4%BC%9A%E5%87%BA%E7%8E%B0%E6%B3%A8%E5%86%8C%E7%9A%84%E7%95%8C%E9%9D%A2%EF%BC%9A) 。
* 如果之前没有登录过，那么在首页就会出现注册的界面：



* 如上图，依次填写账号、邮箱和密码即可创建账户。邮箱验证后就可以点击“Sign In”进行登录操作：

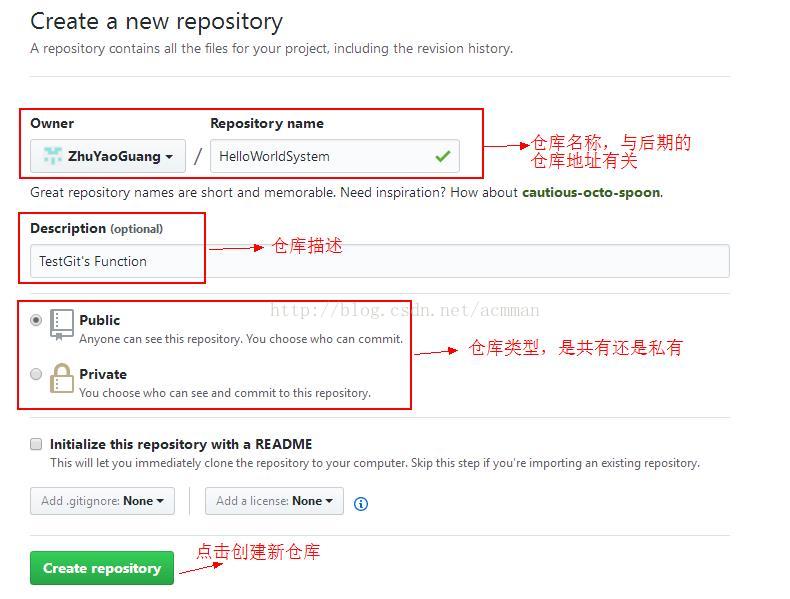


* 登录之后，在右下角，可以创建自己的公开仓库：

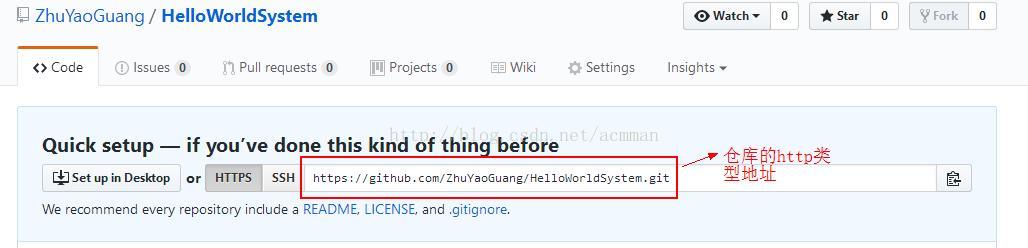


1. 创建一个新的仓库：

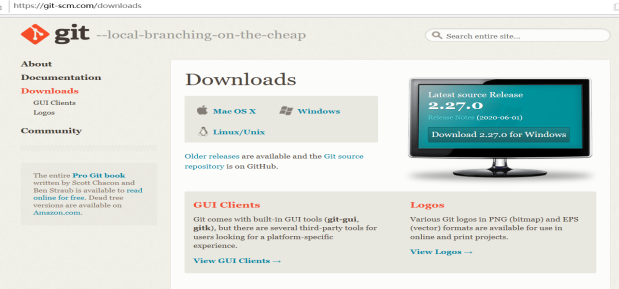
* 点击“New repository” 创建一个新的仓库：

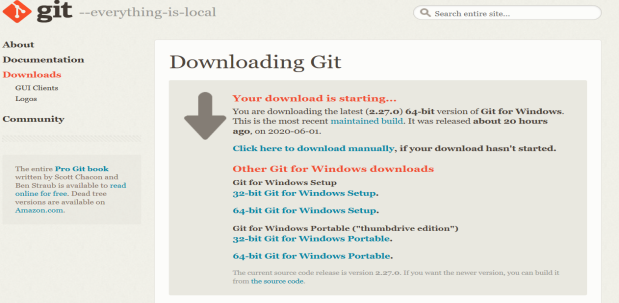


* 这样我们就得到了仓库的http以及ssh的地址：



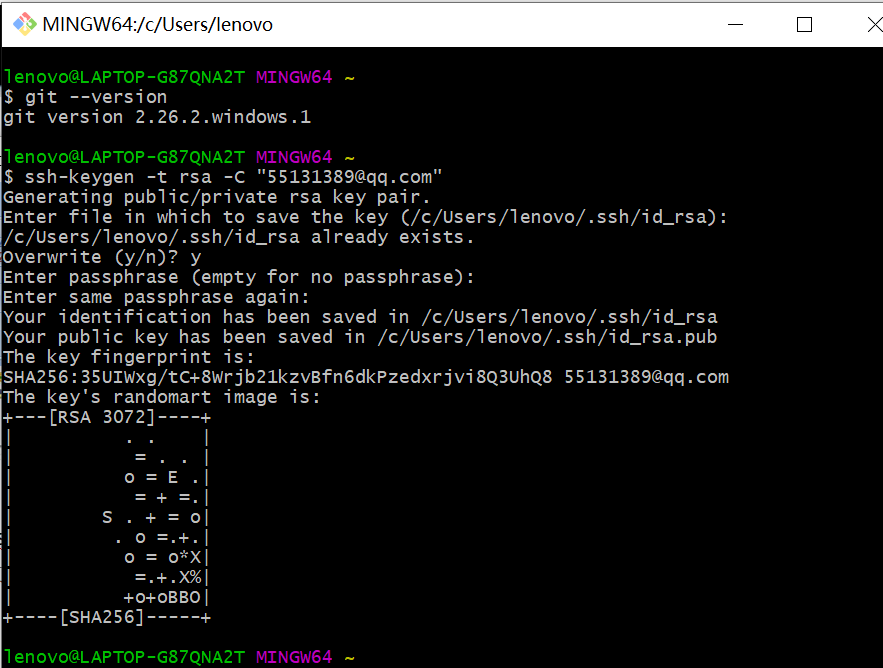
1. 下载Git，
   1. 下载地址为：<https://git-scm.com/downloads>



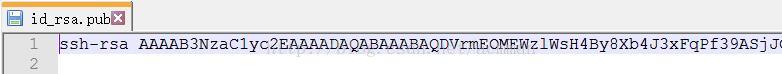


* 1. 同步GitHub
* 绑定gitHub:

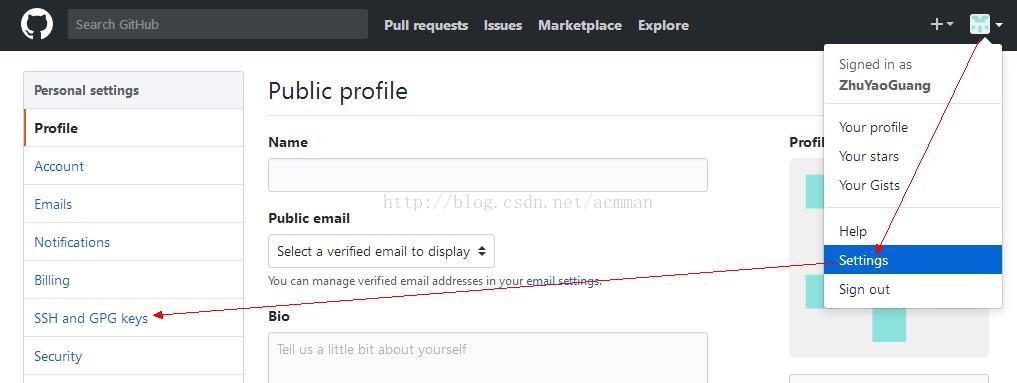
1. 在本地创建ssh key:$ssh-keygen –t rsa –C “[your\_email@yourmail.com](mailto:your_email@yourmail.com)”，后面的是你注册GitHub时候的邮箱地址。

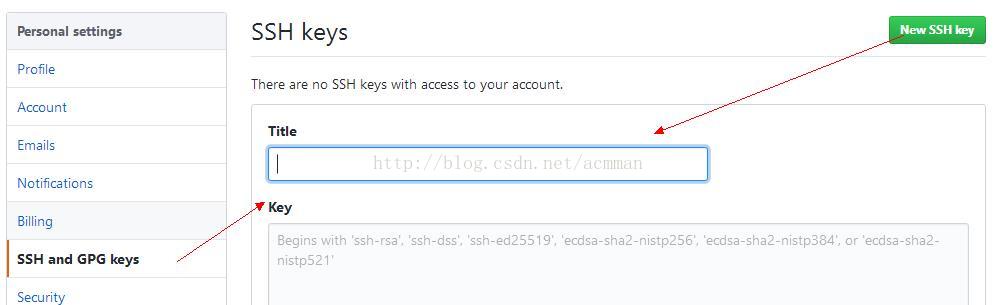


1. 然后成功后，会在User文件夹对应的用户下创建.ssh文件夹，其中有一个id\_rsa.pub文件，复制其中的key:

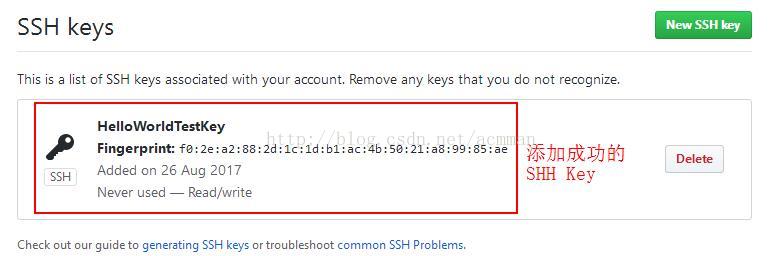


1. 之后返回github，进入Account Settings(账户配置)，左边选择SSH and GPG Keys选项：



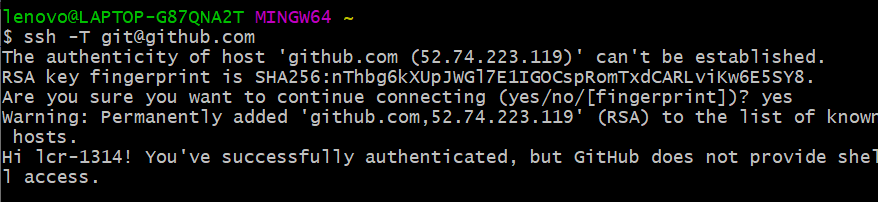


其中的title随便填，下面的粘贴在你电脑上生成的key。点击添加之后，出现如下图，则添加成功。



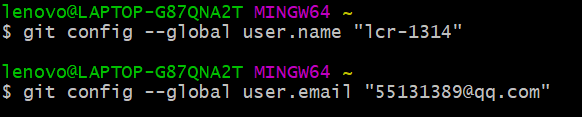
1. 验证是否绑定本地成功，在git-base中验证，输入指令：

$ ssh -T [git@github.com](mailto:git@github.com)



如果第一次执行该指令，则会提示是否continue继续，如果我们输入yes就会看到成功信息。

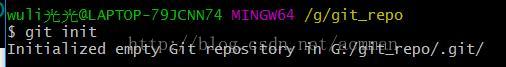
1. 由于GitHub每次执行commit操作时，都会记录username和email，所以要设置它们：



1. 上传文件到GitHub仓库
   1. 在电脑某盘下创建名为“git\_repo”的文件夹(文件夹名随便)。作为本地仓库，然后在其中放置应用的源代码。
   2. 然后在git\_repo文件夹点击右健，选择Git Base Here



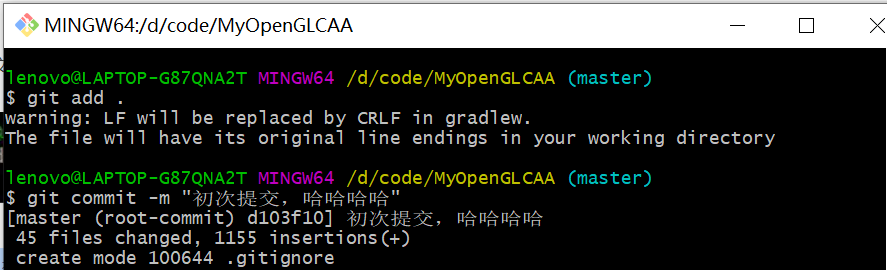
用命令初始化仓库：



* 1. 些时在git\_repo下会找到一个隐藏的.git文件夹，该文件夹是Git用来跟踪管理版本库的:



* 1. 然后添加所有文件到仓库，并提交文件：



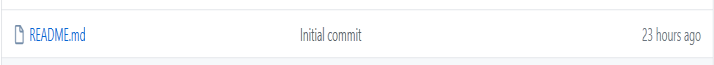
此时，我们的本地仓库就建立好了。

* 1. 本地仓库要关联GitHub的仓库，直接将本地仓库关联远程GitHub仓库地址即可：



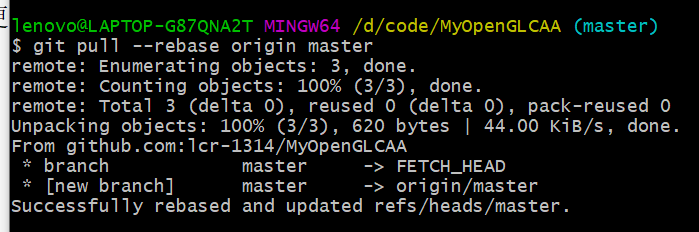
其中，origin 后面跟的是我们的仓库地址。

* 1. 要上传本地代码至GitHub，但是前提是远程仓库不能是空的，所以我们在远程仓库中创建一个README.md的文件：

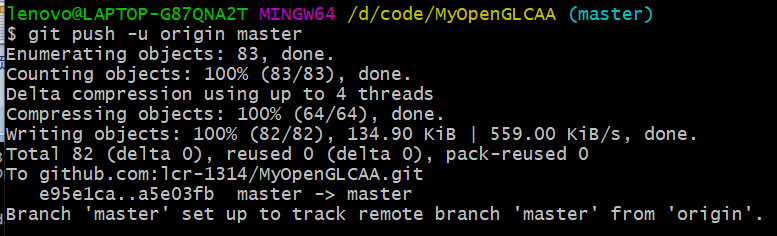


* 1. 本地仓库也创建一个一模一样的README.md文件即可，然后使用

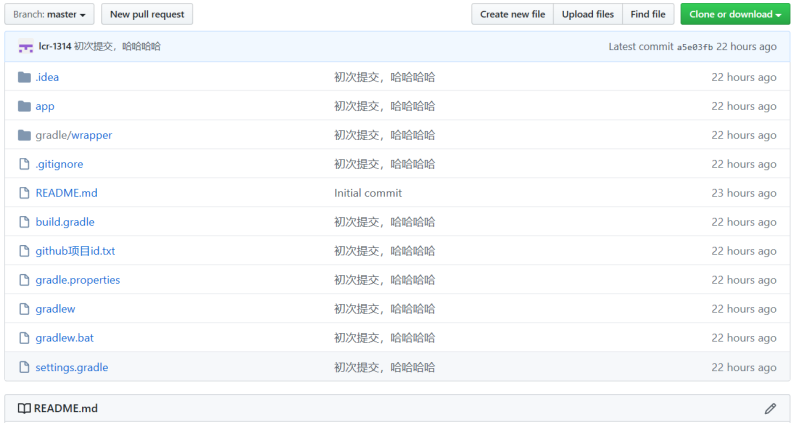
git pull origin master远程更新一下。



* 1. 最后在原来的git base中提交本地仓库工程源代码：



* 1. 最后在GitHub上就可以看到我们的代码已经提交成功：



1. 总结：
   1. 申请注册GitHuB账号；
   2. 创建新仓库；
   3. 安装Git;
   4. 绑定Git到GitHub;
   5. 上传文件到仓库。
2. 使用到的命令有：
   1. $ git –version
   2. $ ssh-keygen -t rsa -C [55131389@qq.com](mailto:55131389@qq.com)
   3. $ ssh -T [git@github.com](mailto:git@github.com) 验证是否绑定本地成功
   4. $ git config –global user.name “lcr-1314”
   5. $ git config --global user.email “[55131389@qq.com](mailto:55131389@qq.com)”
   6. $ git init 初始化本地仓库
   7. $ git add . 所有文件添加到仓库
   8. $ git commit -m "初次提交，哈哈哈哈" 提交文件
   9. $ git remote add origin [git@github.com:lcr-1314/MyOpenGLCAA.git](mailto:git@github.com:lcr-1314/MyOpenGLCAA.git) 将本地仓库关联远程GitHub仓库地址
   10. $ git pull --rebase origin master 远程更新
   11. $ git push -u origin master 提交本地仓库中的工程源代码
3. 参考资料：
   1. <https://blog.csdn.net/hc_ttxs/article/details/79375788>
   2. <https://blog.csdn.net/uotail/article/details/80211897>
   3. <https://blog.csdn.net/qq_28301007/article/details/77917957>
   4. <https://git-scm.com/downloads>