Git url：<https://git-scm.com/download/win>

TortoiseGit url:<https://tortoisegit.org/download/>

Git配置

git安装好去GitHub上注册一个账号，注册好后，点击桌面上的Git Bash快捷图标

# 配置用户名

git config --global user.name "username" //（ "username"是自己的账户名，）

# 配置邮箱

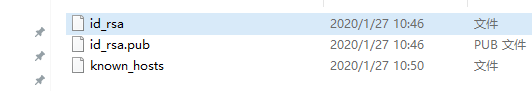
git config --global user.email "username@email.com" [//("username@email.com"注册账号时用的邮箱](mailto:/(\"username@email.com\"注册账号时用的邮箱)

以上命令执行结束后，可用 git config --global --list 命令查看配置是否OK

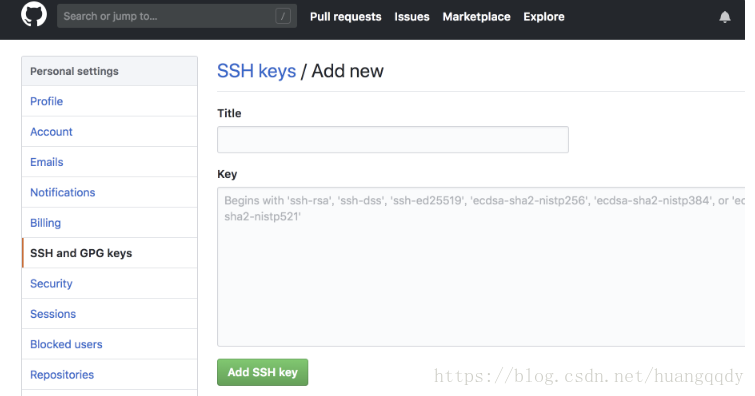
生成ssh  
继续刚才的操作，在命令框中输入以下命令

ssh-keygen -t rsa

然后连敲三次回车键，结束后去系统盘目录下（一般在 C:\Users\你的用户名.ssh）(mac: /Users/用户/.ssh）查看是否有。ssh文件夹生成，此文件夹中以下两个文件



将ssh文件夹中的公钥（ id\_rsa.pub）添加到GitHub管理平台中，在GitHub的个人账户的设置中找到如下界面



title随便起一个，将公钥（ id\_rsa.pub）文件中内容复制粘贴到key中，然后点击Ass SSH key就好啦

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

IMG_256

（拒绝合并不相关的历史）

# 解决

出现这个问题的最主要原因还是在于本地仓库和远程仓库实际上是独立的两个仓库。假如我之前是直接clone的方式在本地建立起远程github仓库的克隆本地仓库就不会有这问题了。

查阅了一下资料，发现可以在pull命令后紧接着使用--allow-unrelated-history选项来解决问题（该选项可以合并两个独立启动仓库的历史）。

git pull origin master --allow-unrelated-histories

以上是将远程仓库的文件拉取到本地仓库了。  
紧接着将本地仓库的提交推送到远程github仓库上，使用的命令是：

$ git push <远程主机名> <本地分支名>:<远程分支名> 也就是 $git push origin master:master 提交成功。