

ssh -key配置

在这一步我已经默认你安装好了Git，打开Git，在Git命令行输入`cd ~/.ssh`检查电脑本机是否有ssh key设置，**注意：.ssh是隐藏文件，一般在电脑C/用户/Administrator下面就能够找到。**

一般情况下，之前没有用过git的同学电脑本机一般不会有ssh key的，下面我讲给大家介绍如何配置ssh key

1、进入~路径下，必须保证当前路径在~路径下

在git命令行敲击 `ssh-keygen -t rsa -C "xxx.@yyy.zzz"` //建议写自己真实有效的邮箱地址。**注意：在敲代码是不要将双引号也敲击进去。**

然后命令行会出现如下代码：

Enter file in which to save the key (/c/Users/xxxx_000/.ssh/id_rsa):
//此时我们什么都不需要操作，直接回车就好

Enter passphrase (empty for no passphrase): //此时要你输入码
(可以为空，直接回车就好，也可以输入你的密码，这个密码在你最后把本地资源推送到github上面的时候会让你填写密码，此时密码隐藏，你输入进去是看不到的)

Enter same passphrase again: //再次确认密码（如果你第一次有输入密码，这次就再输一次，如果没有直接回车就行了）

Your identification has been saved in /c/Users/xxxx_000/.ssh/id_rsa. //生成的密钥

Your public key has been saved in /c/Users/xxxx_000/.ssh/id_rsa.pub. //生成的公钥

The key fingerprint is:

e3:51:33:xx:xx:xx:xx:xxx:61:28:83:e2:81 xxxxxx@yy.com

*本机已完成ssh key设置，其存放路径为：c:/Users/xxxx_000/.ssh/下。其中xxxx_000为你的用户名。

2、添加ssh key 到Github上

首先登陆Github,点击右上角的“▼”→Settings→SSH keys→Add SSH key。
然后在打开c:/Users/xxxx_000/.ssh里面的id_rsa.pub文件，全选复制公钥内容

也可以在git bash中的命令行输入`cat ~/.ssh/id_rsa.pub`，将得到公钥Title自定义，将公钥粘贴到GitHub中Add an SSH key的key输入框，最后“Add Key”

3、配置账户

`$ git config --global user.name "your_username"` #设置用户名

`$ git config --global user.email "your_registered_github_Email"` #设置邮箱地址(建议用注册github的邮箱)

4、测试ssh keys是否设置成功。

`ssh -T git@github.com`

The authenticity of host 'github.com (192.30.252.129)' can't be established.

RSA key fingerprint is 16:27:xx:xx:xx:xx:xx:4d:eb:df:a6:48.

Are you sure you want to continue connecting (yes/no)? yes #确认你是否继续联系，输入yes

Warning: Permanently added 'github.com,192.30.252.129' (RSA) to the list of known hosts.

Enter passphrase for key '/c/Users/xxxx_000/.ssh/id_rsa': #生成ssh key是密码为空则无此项，若设置有密码则有此项且，输入生成ssh key时设置的密码即可。

Hi xxx! You've successfully authenticated, but GitHub does not provide shell access. #出现此句话，说明设置成功。

到这里，git相关的所有配置已经完成，下面我将给大家介绍最常使用的命令

常用命令

1、创建仓库（公开）

点击github右上角的+里面的new repository

在repository name栏中输入仓库的名称

description 栏中可以设置仓库的说明

public（公开）通用 private（非公开）收费

勾选Initialize this repository with a README，如果想向GitHub添加手中已有的Git仓库，建议不要勾选

Add.gitignore 下拉框中选择要使用的框架

Add a license选择添加许可协议文件

点击create repository按钮，完成仓库的创建

2、克隆之前首先要选择克隆文件的位置（cd）

clone已有仓库

git clone git@github.com:XXX/yyyy.git //XXX为github的用户名，yyy为仓库名

3、在对应的文件夹中添加新有项

git status

4、提交

git add mmm.sss //mmm为文件名称，sss为文件拓展名（常用git add .）

git commit -m "hhh" //hhh为git commit 提交信息，是对这个提交的概述

git log//用于查看提交日志

git push //更新GitHub上的仓库

5、用git创建仓库

mkdir nnn //仓库名

cd hhh

git init //初始化仓库

git status //查看仓库状态

touch README.md //创建README.md文件

git add README.md //添加README.md至暂存区

git commit -m "hhh" //如果想要提交信息记录的更详细，请不要加 -m

git log --pretty=short //加--pretty=short 只显示提交信息的第一行

git log ggg //ggg是指指定的文件或目录，用于查看指定的目录、文件的日志

git log -p //查看提交所带来的改动

git log -p ggg //查看指定文件的改动

git diff //可以查看工作树，暂存区，最新提交之间的差别

git diff HEAD //查看工作树与最新提交的差别

6、分支操作

git branch //显示分支一览表，同时确认当前所在的分支

git checkout -b aaa //创建名为aaa的分支，并且切换到aaa分支

(git branch aaa //创建名为aaa的分支

git checkout aaa // 切换到aaa分支

) 能和git branch -b aaa 得到同样的效果

git checkout - //切换到上一分支

7、合并分支

git checkout master //切换到master分支

git merge --no--ff aaa // 加--no--ff 参数可以在历史记录中明确地记录本次分支的合并

git log --graph //以图表形式查看分支

8、更改提交的操作

git reset //回溯历史版本

git reset --hrad //回溯到指定状态，只要提供目标时间点的哈希值

9、推进历史

git reflog //查看仓库的操作日志，找到要推历史的哈希值

git checkout master

git reset --hrad ddd //ddd为要推进历史的哈希值

10、修改提交信息 git commit --amend

压缩历史 git rebase -i 错字漏字等失误称作typo

根据以前的步骤在GitHub上创建仓库，应于本地的仓库名相同 GitHub上面创建的仓库的路径为git@github.com: 用户名/仓库名.git

git remote add eee git@github.com: 用户名/仓库名.git //添加远程仓库，并将git@github.com: 用户名/仓库名.git远程仓库的名称改为eee

git push -u eee master //推送至远程仓库 master分支下 -u 参数可以在推送的同时，将eee仓库的master分支设置为本地仓库的当前分

支的upstream（上游）。添加这个参数，将来运行git pull命令从远程仓库获取内容时，本地仓库的这个分支就可以直接从eee的master分支中获取内容

git checkout -b feature d eee/feature d //获取远程的feature d分支到本地仓库，-b参数后面是本地仓库中新建的仓库的名称

git pull eee feature d //将本地的feature d分支更新为最新状态

在GitHub上面查看两个分支之间的差别，只需要在地址栏中输入http://github.com/用户名/仓库名/分支1...分支2

11、查看master分支在最近七天内的差别

`http://github.com/用户名/仓库名/master@{7.day.ago}...master` (同样, day, week, month, year都是可以哒)

12、查看与指定日期之间的差别

`http://github.com/用户名/仓库名/master@{xxxx-xx-xx}...master`
(xxxx-xx-xx代表年月日)

到这里, git相关的命令已经全部介绍完成, 剩下的就靠大家自己学习了。