Git常用命令

获取仓库

# 可以用

git init

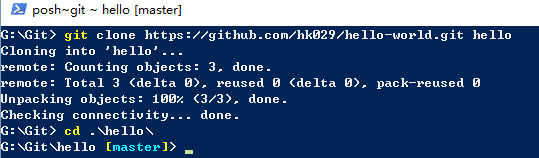
初始化一个仓库，但是一般常用的都是克隆现有的仓库，这里还是用刚才创建的helloworld仓库

git clone https://github.com/hk029/hello-world.git

url后面可以用一个新的路径名，让它保存到特定的目录下，默认就是当前路径下创建一个和仓库名一样的目录。

注意：这个命令是克隆了一个仓库而不是简单的拷贝了文件下来，还保存了有关仓库的信息，基本就是克隆出了一个小的本地仓库。

然后可以cd进去看看



这里有个比较有意思的地方，你会发现目录名后一个’[master]’这个表示目前这是一个master分支。当前目录有个.git目录，它会记录仓库的信息，所以你能看到‘[master]’这个标签。之后你对当前目录的文件做的操作，都会被记录。

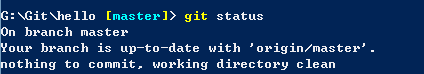
状态与暂存

检查目前仓库的状态是挺重要的一个环节，以免你提交代码的时候提交的不是最新代码。   
一般来说，你目前目录下的文件就两种状态

* 跟踪
* 未跟踪

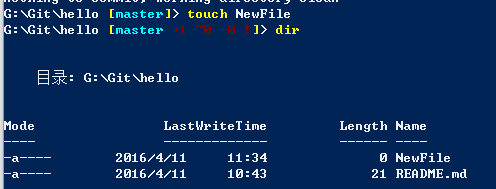
我们可以先输入命令看看当前仓库的状态

git status



可以发现目前目录下很干净，一切都和当初克隆下来一样，所以状态显示也是很干净。

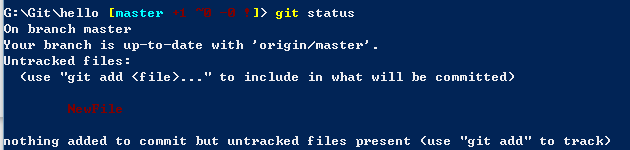
我们新键一个文件，或者从别的地方移动一个文件到当前目录下，看看有什么变化。



我们可以看见，‘[master]’之后多了几个东西。

+1 表示目前有1个新文件   
~0 表示0个修改的文件   
-0 表示0个删除的文件   
! 表示未保存

我们再输入status命令看看



发现目前有个未跟踪的文件了。

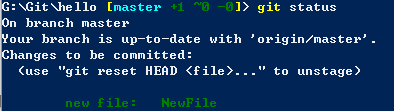
我们把它暂存到暂存区

git add NewFile

这个命令会把这个文件放到暂存区（还是在本地）到时候提交的时候就会把暂存区的东西提交到网上。

这里写图片描述

这个时候，我们发现红色的字变成了绿色，感叹号也没有了，说明目前修改都已经保存了。

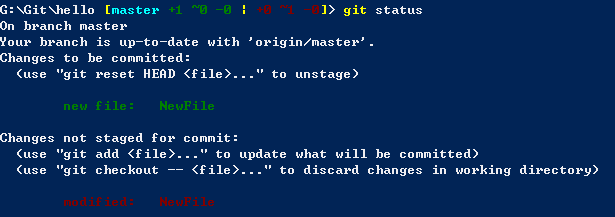
再输入satus命令看看   


这个时候已经是跟踪状态了。

如果我们这时候修改NewFile会发生什么呢？

这里写图片描述

可以发现，又变红了，这时候出先了~1,说明有一个修改文件。使用status命令看看



果然，出现了一个NewFile在未跟踪里面。这不是很奇怪吗，一个文件既是跟踪又是未跟踪？其实这很好理解，你可以理解为暂存区还有一个NewFile拷贝，就是原来那个我们add的“空”的NewFile。修改的这个不在暂存区，如果这时候我们把所有修改提交，那么提交的是“空”的NewFile。

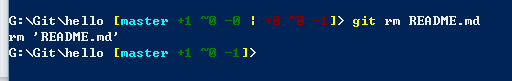
所以当我们提交前，一定要看是不是还有红字？是不是还有什么修改没有更新到暂存区！

最后，还说下，其实git status有个简化输出的形式。

git status -s

这里写图片描述

这里我为了让所有状态都出现，删除了README.md，可以发现这个简化版输出其实更直观。



我们把所有的修改更新到暂存区吧：

git add NewFile

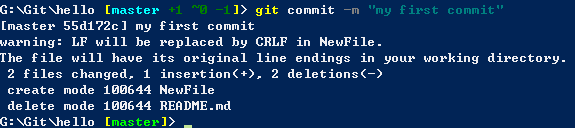
git rm README.md

现在所有的更新都更新到暂存区了，可以提交了！

提交到本地仓库

这里的commit只是保存到了本地。如果你只需要一个本地仓库，那么现在也就够了。

git commit -m "my first commit"



注意：一定要带-m加上说明

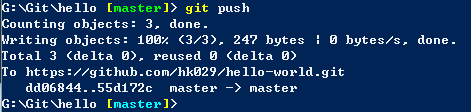
推送到远程仓库

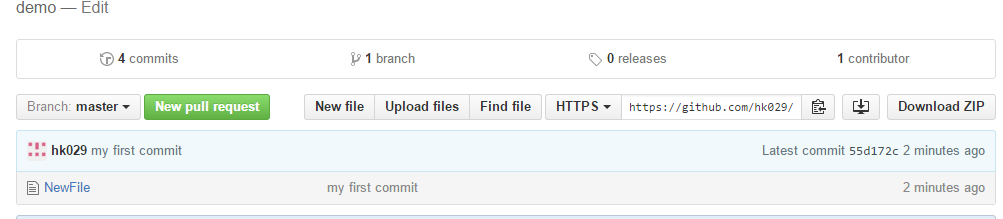
因为你是直接从远程仓库拷贝的，所以你输入

git remote

会发现已经有个orignal了   
这个时候你直接git push就行了

git push

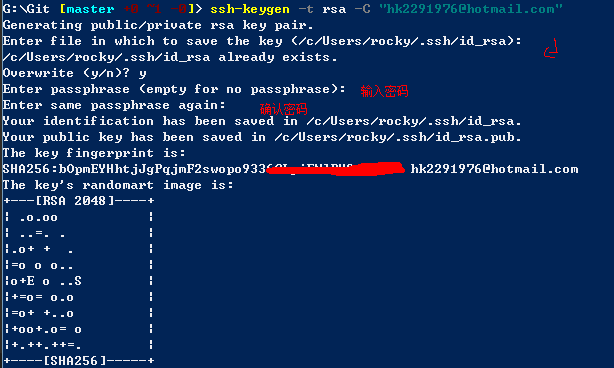




用SSH连接并推送到远程仓库

但是，建议还是使用ssh的模式，安全点。下面说下ssh的模式怎么用，首先你要创建一个私钥，就是在自己电脑里的钥匙。

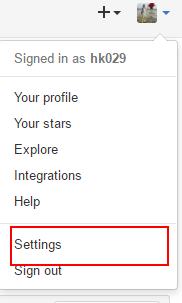
ssh-keygen -t rsa -C "your email addr"

  
第一个是问你改不改目录，回车就好。然后输入密码，确认（这个密码是生成这个密钥的密码，也可以为空，这样你下次push就不用输入密码了)。

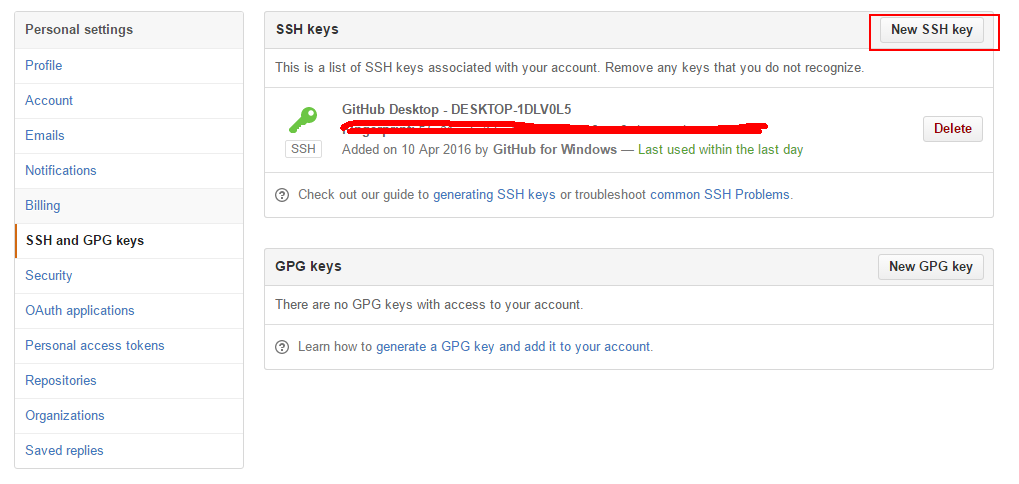
然后你就可以去那个目录下找id\_rsa.pub文件，打开（随便用什么打开），把里面的东西复制。

这里写图片描述

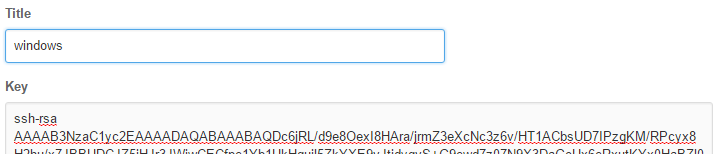
然后去网站上把自己私钥输入进去，头像——settings



根据图片点New SSH key



然后输入一个随便什么title自己知道就好，和你刚才复制的东西在key里



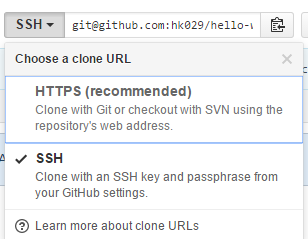
这时候，就算在github上注册了你的私钥，然后在输入

ssh -T git@github.com

看看是不是能ssh连上github

这里写图片描述

好了一切准备就绪。   
我们先把远程仓库加上：



选择ssh，然后复制后面的地址。

（因为之前有一个original的，我们这里方便测试先删了它）

git remote rm origin

输入

git remote

发现没东西了，可以添加新的远程仓库了。

git remote add origin git@github.com:hk029/hello-world.git

这里写图片描述

然后输入

git push -u origin master

master是你的分支，origin是你的远程仓库

第一次git push -u origin master后，就可以用直接用git push指令了

好了这些基本操作就可以满足一些日常使用了，后续再加一些关于分支还有撤销 删除的操作。

[**Git fetch和git pull的区别**](http://blog.csdn.net/hudashi/article/details/7664457)

Git中从远程的分支获取最新的版本到本地有这样2个命令：  
1. git fetch：相当于是从远程获取最新版本到本地，不会自动merge

[**Git**](http://lib.csdn.net/base/git) fetch origin master  
git log -p master..origin/master  
git merge origin/master

    以上命令的含义：  
   首先从远程的origin的master主分支下载最新的版本到origin/master分支上  
   然后比较本地的master分支和origin/master分支的差别  
   最后进行合并  
   上述过程其实可以用以下更清晰的方式来进行：

git fetch origin master:tmp  
git diff tmp   
git merge tmp

    从远程获取最新的版本到本地的test分支上  
   之后再进行比较合并  
2. git pull：相当于是从远程获取最新版本并merge到本地

git pull origin master

上述命令其实相当于git fetch 和 git merge  
在实际使用中，git fetch更安全一些  
因为在merge前，我们可以查看更新情况，然后再决定是否合并