1. 将本地文件夹直接添加到github远程仓库中



Github初始仓库最好什么文件都不要有，不然好像会引起冲突。

1. git全局配置

在安装后首次使用需要进行全局配置。这里用户名与邮箱地址建议与github的一样。



1. 创建仓库

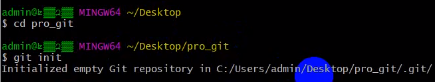
仓库最好是空目录且目录中不要有中文名

1. 创建仓库

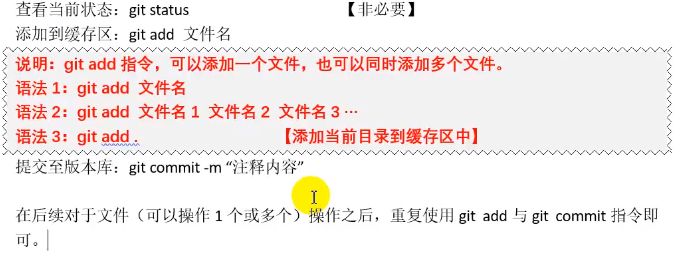
可以右键新建文件夹也可以用指令创建



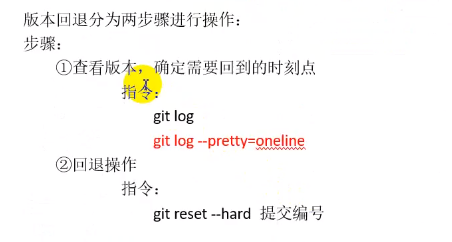
1. 通过cd进入该文件夹，进行git初始化（初始化的目的是为了使git知道要管理这个仓库），初始化之后文件夹下会有个.git的隐藏文件夹，不能删除或随意更改。



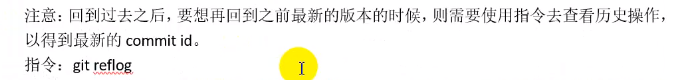
1. git常用指令操作



1. git版本回退

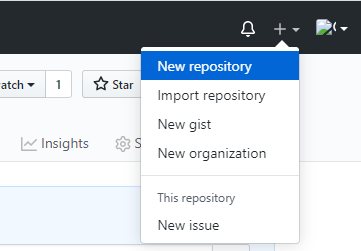


这里查看版本推荐用第二种，因为更简洁直观





1. github远程仓库的创建

repository是仓库的意思。

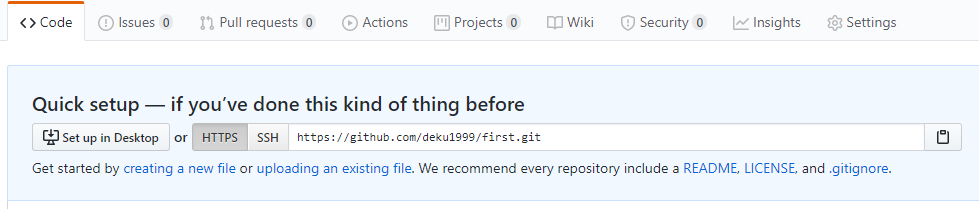
1. github远程仓库的使用

（1）基于http协议

1. 创建仓库，名字与github中仓库名相同。

②使用clone指令克隆线上仓库到本地

git clone 地址



1. 在仓库上做对应的操作（提交暂存区，提交本地仓库，提交线上仓库，拉取线上仓库）

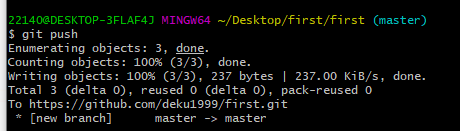
前两个就是git add和git commit -m，前三个步骤可以看作是操作github远程仓库的工作流程。

**提交到线上仓库的指令:git push**

直接往仓库提交的话会报403错误，因为不是每个人都有权限进行操作。需要进入到.git文件夹修改config文件，在url前加上我们的github的用户名密码即可。



下图为成功效果

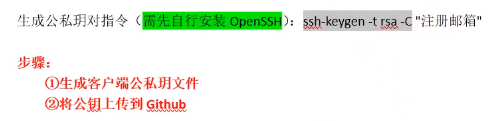


**拉取线上仓库文件：git pull**

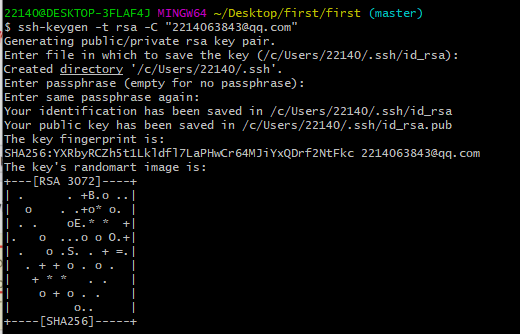
注：每天上班第一件事一般就是git pull，而下班最后一件事就是git push。

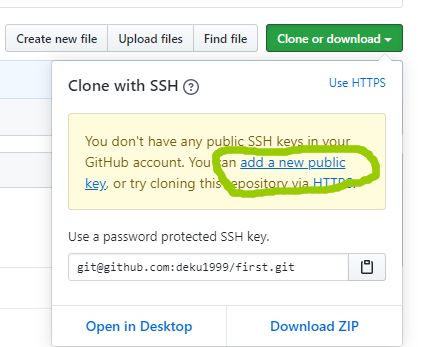
（2）基于ssh协议（推荐）

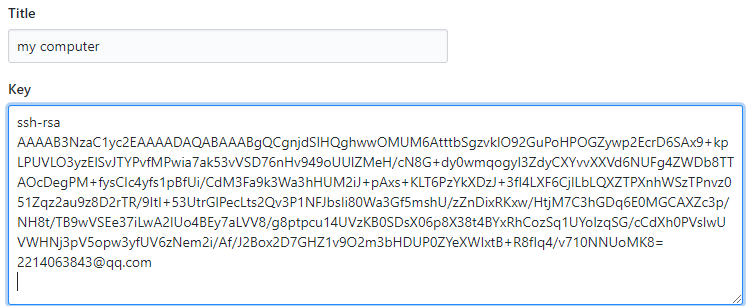
①该方式与前面https方式相比，只是影响github对于用户的身份鉴权方式，对于git的具体操作（如提交本地，添加注释，提交远程等操作）没有任何影响



执行指令之后直接回车即可，注意要通过public key找到公私钥文件位置并复制其中内容到github上去添加一个公钥。标题可以随意取。







1. 执行后续git操作，与先前一样

即先克隆仓库之后进行添加缓存区并push，不过与http的区别是它不用去配置config文件，更方便。

1. 分支管理

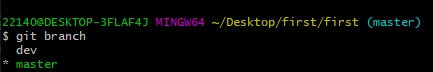
我们一般默认的分支都是在master分支

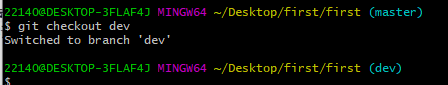




git checkout -b 分支名，这个操作如果分支不存在将会先创建分支再切换。

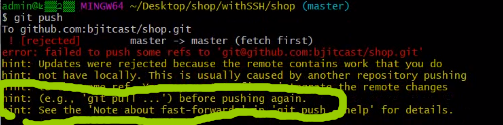
1. 查看当前分支有两种方式，一是看后缀，二是git branch查看所有分支，当前分支前会有梅花标记。





1. 分支之间的操作是互不影响的，即在a分支下对文件进行的修改，回到b分支时该文件还是b分支时候的状态。
2. 合并分支是合并当前分支加git merge后的要合并的分支名。
3. 删除某一分支时不能位于那一分支下。
4. 合并完所有分支后记得提交到远程仓库。
5. 冲突的产生与解决

这里的冲突是指当我们上班时忘了git pull，而同事恰巧又在这段时间进行了修改，当我们进行git push时便会报错。这时需要我们再次进行git pull然后与同事商量文件到底该怎么改。修改结束再次git add，commit，push即可。当然，我们只要记住每天上班先去git pull是可以避免这种情况发生的。



1. 忽略文件

有时一些文件夹基本不会改变，或者改变了我们并不想提交到仓库里，我们可以采用忽略文件。$ touch .gitignore

在文件中以#开头的都是注释

