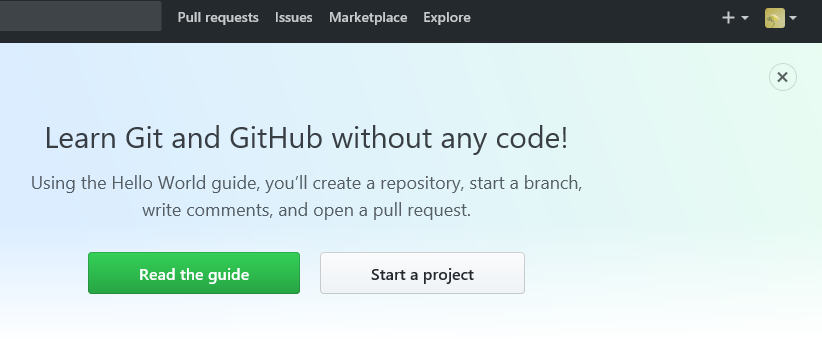
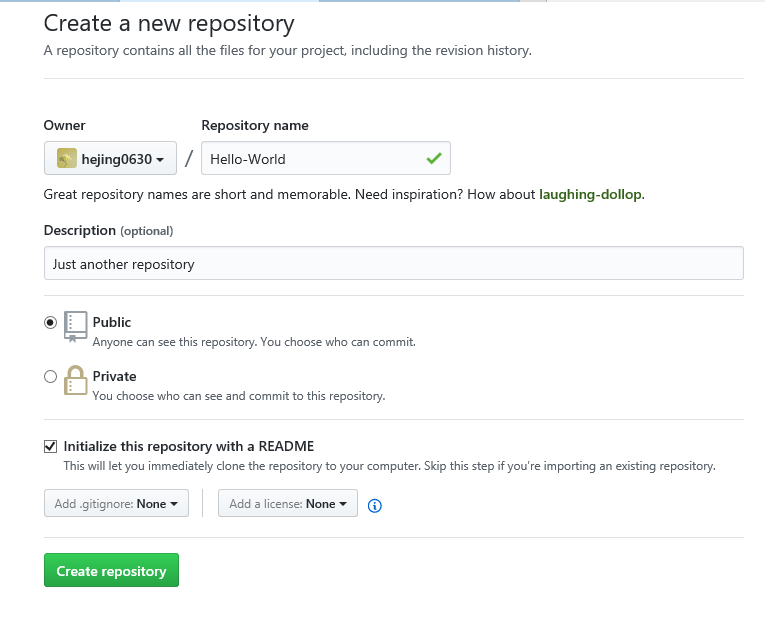
何静

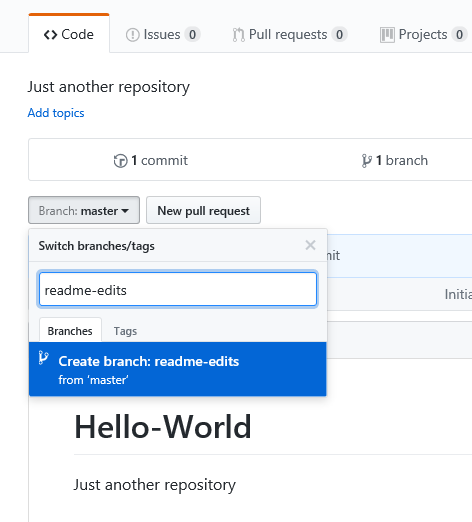
首先注册guthub账号，创建代码仓库，如图：



点击Start a project,开始建立一个新的repository



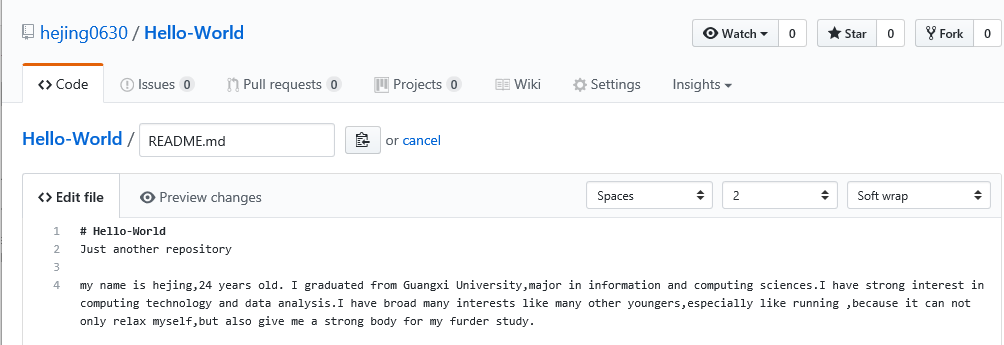
建立分支，如图：

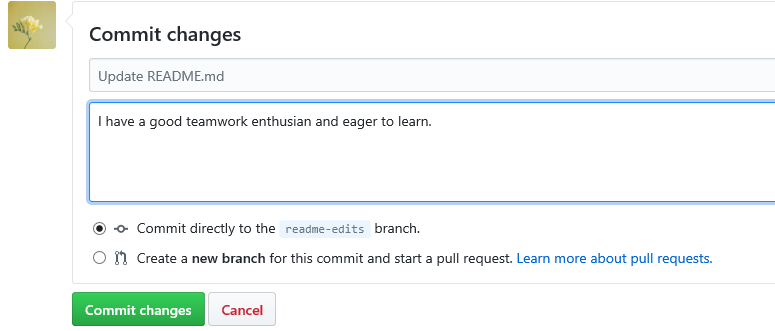


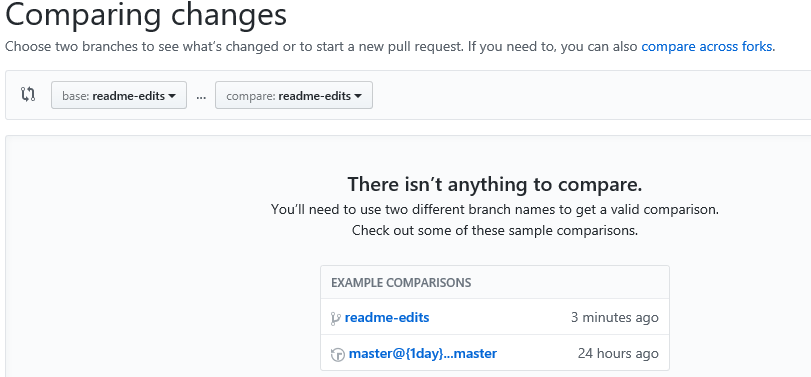
点击readme：

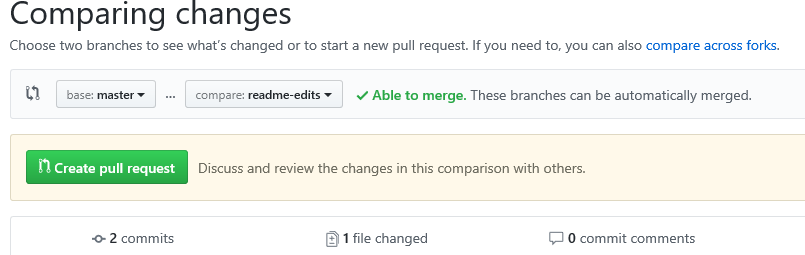


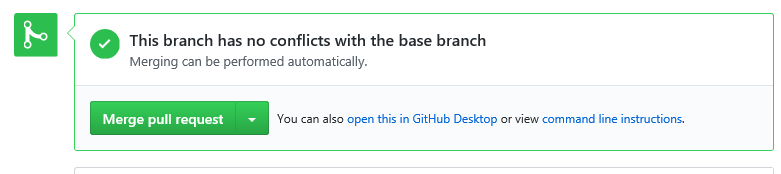
进入编辑，并确认：







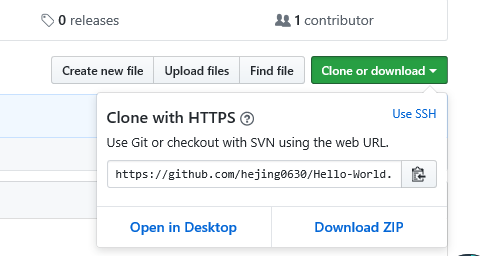




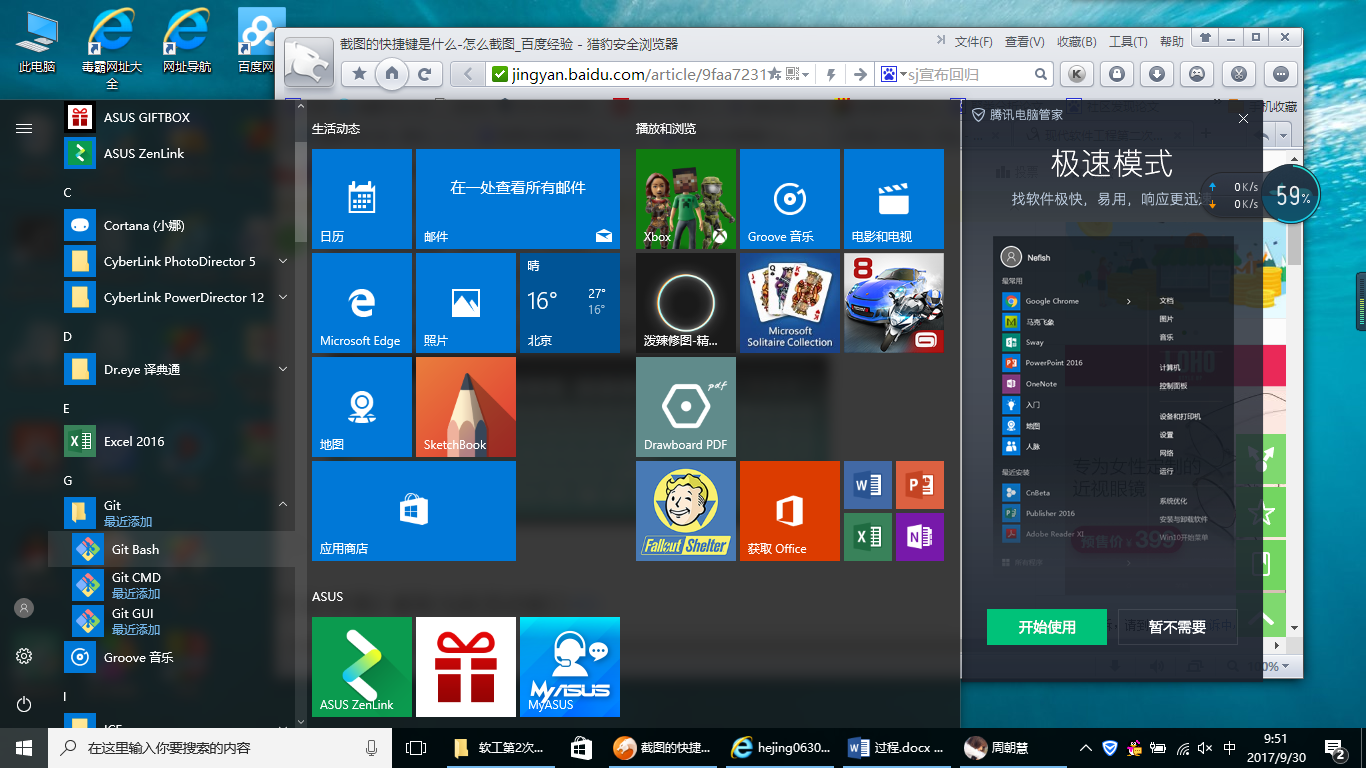
**下面练习使用git的add/commit/push/pull/fetch/clone等基本命令**

git clone

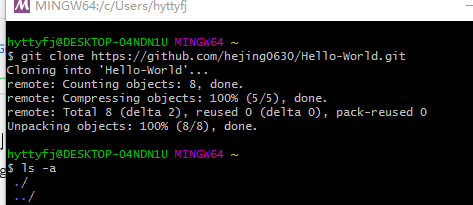
使用git clone 克隆一个git 的仓库到本地，便于修改和查看，我的地址是https://github.com/hejing0630/Hello-World.git



打开最新安装的git软件，打开git bash，进入其界面



Clone命令为：



就能看到，guthub 上的项目已经同步到了本地



git status

使用git status 命令查看项目的当前状态。创建一个新目录hello-world-hj，并在其下添加一个C++版本的hello world程序：



使用git status 命令，看到我们所做的修改已经被检测出来。



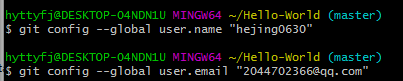
git add

使用git add 命令将该文件添加到缓存，如下图：

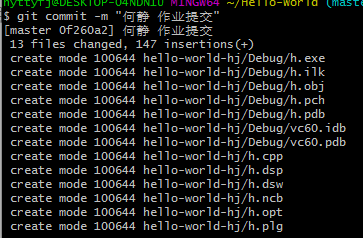


git commit

使用git commit 能将写入缓存区的内容添加到仓库中，但是git在提交时会记录我们的用户名和邮箱，所以首先得设置用户名和邮箱地址：

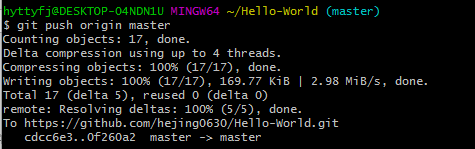


使用-m可以在命令行中提交注释

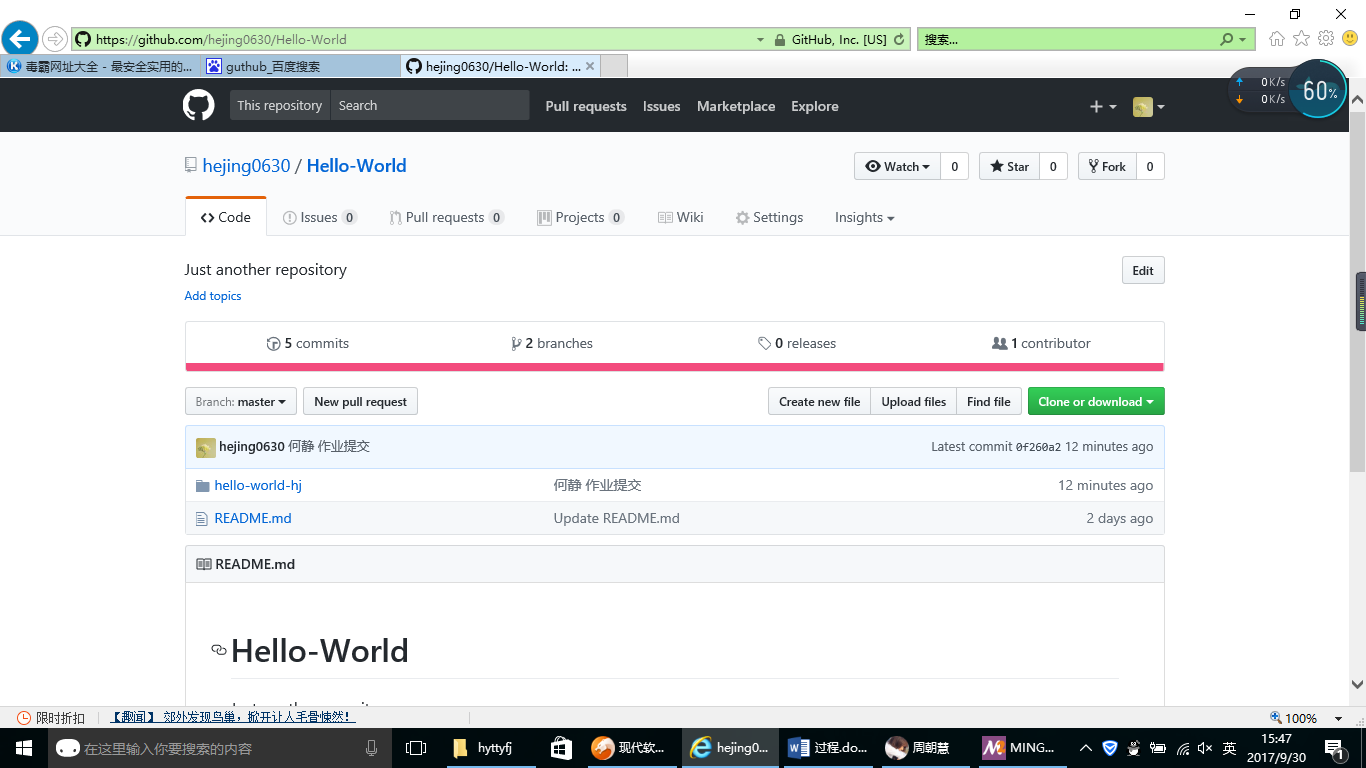


git push

将我们所操作的数据推送（push）到Git仓库中永久地存储：



检查guthub上的项目是否成功推送成功。反馈如下：



从图中我们可以看到，我所做的修改工作已经全部同步到guthub中，并且附带最后一次修改的注释。

git pull 和 git fetch

*git fetch*：相当于是从远程获取最新版本到本地，但不会自动merge到本地。

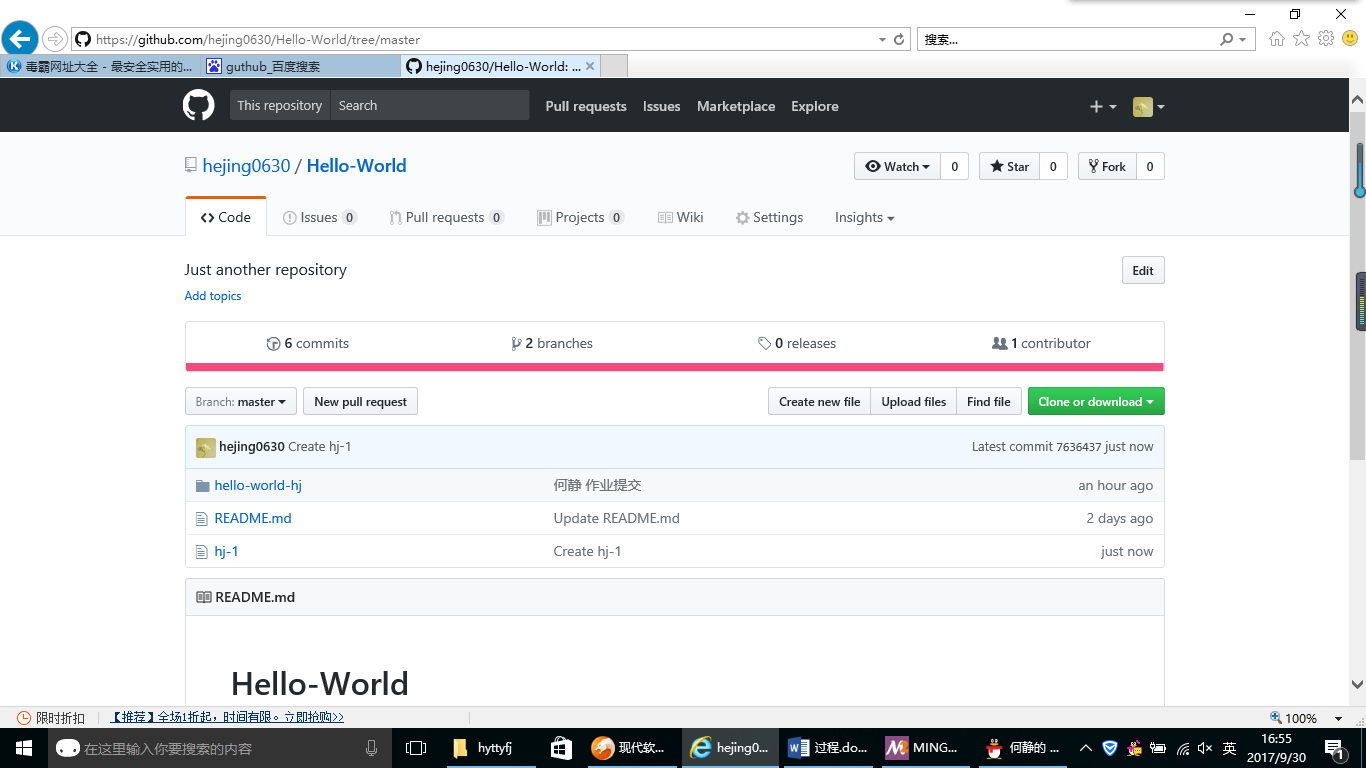
*git pull*：相当于是从远程获取最新版本并merge到本地。

从结果上来看，git pull=git fetch + git merge。为了更好的理解，画出下面的图：

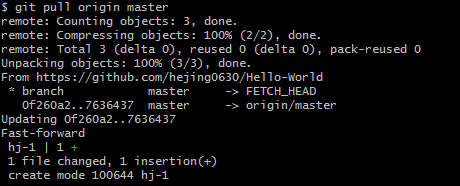


以上的图只是对git pull和git fetch做出了简单的区别描述，限于篇幅，这里就不详细叙述了。在实际的使用中，git fetch更安全一些，因为在merge之前，我们可以查看更新情况，然后再决定是否合并结束。

现在，我来展示git pull的作用。首先在guthub上给项目增加一个文件hj-1



接着在本地使用git pull命令：



即在guthub网页上增加的文件已经被同步到本地仓库。