Github是一个流行的代码管理网站，同时也是全球最大的同性交友网站（滑稽）。Github网页上你可以自由地托管自己的项目，也可以fork别人的项目过来玩耍，非常之方便，今天笔者就来介绍一下github桌面版程序上，针对常见需求的那些相关使用方法，此处以[Mac](https://www.baidu.com/s?wd=Mac&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)版本的为例，win上的基本类似。

而学习Github桌面版的过程，其实也是在加深了解Git技术的各方各面，所以git小白也可以通过Github桌面版的使用来入门git哟！

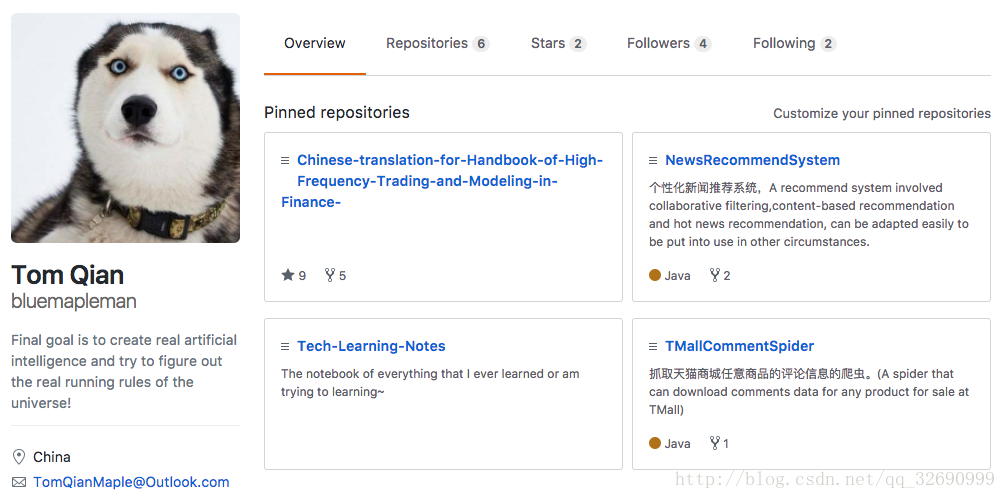
接下来，我会通过github中比较基本的概念包括clone，branch，push，pull等来入手，一一介绍Github Desktop的使用方法。

* [将仓库中的代码下载到本地 (Clone Github Project to Local Repository)](https://blog.csdn.net/beashaper_/article/details/80704124#%E5%B0%86%E4%BB%93%E5%BA%93%E4%B8%AD%E7%9A%84%E4%BB%A3%E7%A0%81%E4%B8%8B%E8%BD%BD%E5%88%B0%E6%9C%AC%E5%9C%B0-clone-github-project-to-local-repository)
* [在本地更新网页端的改变 （pull）](https://blog.csdn.net/beashaper_/article/details/80704124#%E5%9C%A8%E6%9C%AC%E5%9C%B0%E6%9B%B4%E6%96%B0%E7%BD%91%E9%A1%B5%E7%AB%AF%E7%9A%84%E6%94%B9%E5%8F%98-pull)
* [在本地提交新的代码版本到仓库 (push local changes to remote repository)](https://blog.csdn.net/beashaper_/article/details/80704124#%E5%9C%A8%E6%9C%AC%E5%9C%B0%E6%8F%90%E4%BA%A4%E6%96%B0%E7%9A%84%E4%BB%A3%E7%A0%81%E7%89%88%E6%9C%AC%E5%88%B0%E4%BB%93%E5%BA%93-push-local-changes-to-remote-repository)
* [建立新的分支 (creating new branch)](https://blog.csdn.net/beashaper_/article/details/80704124#%E5%BB%BA%E7%AB%8B%E6%96%B0%E7%9A%84%E5%88%86%E6%94%AF-creating-new-branch)
* [请求合并自己的代码 (pull request)](https://blog.csdn.net/beashaper_/article/details/80704124#%E8%AF%B7%E6%B1%82%E5%90%88%E5%B9%B6%E8%87%AA%E5%B7%B1%E7%9A%84%E4%BB%A3%E7%A0%81-pull-request)

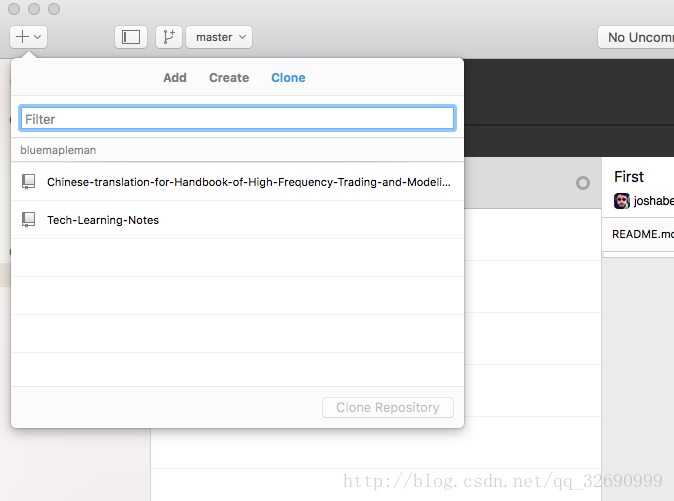
**将仓库中的代码下载到本地 (Clone Github Project to Local Repository)**

首先，当你不想每次都打开github网站去对你的项目文件做编辑时，你可以选择把它们down下来到本地磁盘上。而从此以后，你在本地上对项目文件的任何修改都会被记录下来，并且只要通过github桌面程序就可以把这些修改同步到对应的github网站仓库上，非常方便。

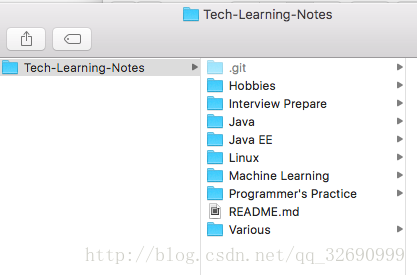
1. 假设我们在github上有一个名为Tech-Learning-Notes的项目。



1. 接下来，打开github桌面版，点击左上角的“+”按钮，选择clone，你可以看到你在github上存放的（尚未down到本地的）项目列表。选中Tech-Learning-Notes项目，选择clone Repository，然后选择本地路径以存放项目文件夹。



* 1. 稍等一会儿，待clone完成后，到之前选择的对应路径下，就可以看到你的项目文件夹了。

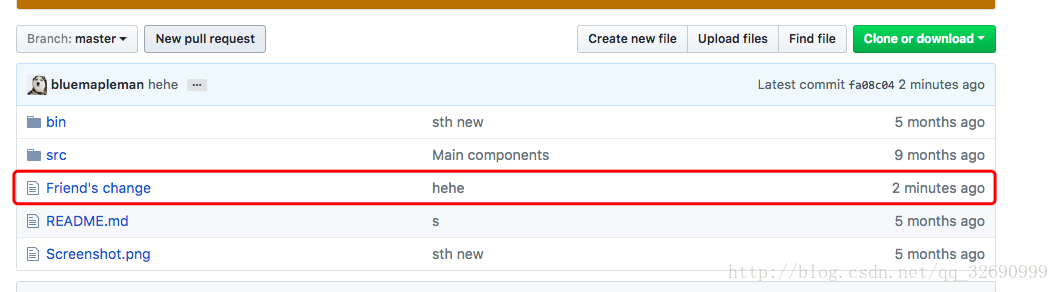


当然，你可以随意挪动文件夹的位置，因为github桌面程序已经锁定了文件夹本身，不会因为你挪动了文件夹的位置与之前保存的位置不一样就失去关联。而“锁定”其实是通过项目文件夹下的”.git”文件夹实现的，所以大家不要轻易动这个文件夹哟~

然后就可以愉快地在本地随意编辑你的项目了。

**在本地更新网页端的改变 （pull）**

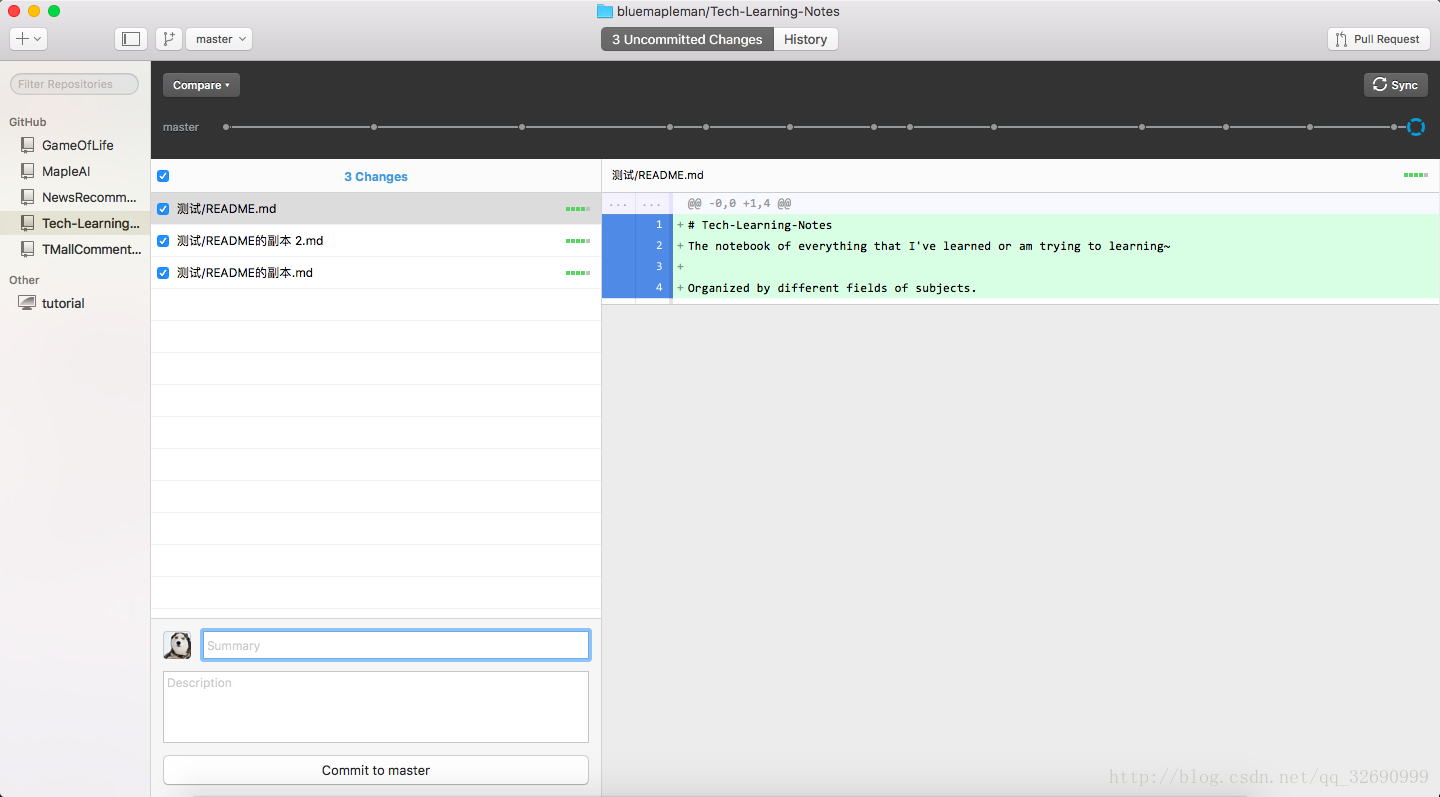
有一次，你的项目合作者在页面那边提交了一个新文件，名为【Friend’s change】，于是你很感兴趣，想把新文件同步到本地，很简单：点击【Repository】-【Pull】，接着，你就可以看到本地仓库里出现了合作者提交的新文件。



**在本地提交新的代码版本到仓库 (push local changes to remote repository)**

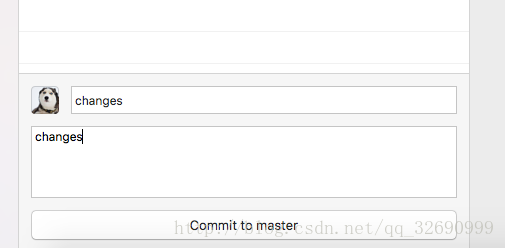
问题来了，我们在本地对我们的项目做了一些新的修改，比如改了一些文件的源代码，新增了和删除了一些文件，我们怎么把这些变化同步到github仓库呢？

1.当你对本地的项目文件夹下的文件做了任何改动后，打开github桌面程序，你会发现，你的项目页面变成了这样：

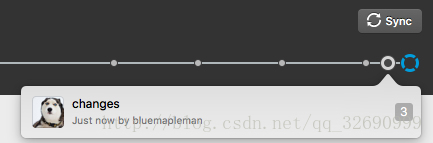


这里出现的所有新条目都是你对项目文件做过的改动，包括修改代码，增加或删除文件，而你可以通过勾选条目前的[复选框](https://www.baidu.com/s?wd=%E5%A4%8D%E9%80%89%E6%A1%86&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)去选定将那些改动提交到仓库。

2.选择好改动后，在下面的注释框里填写改动的相关信息，以方便后续回溯。（不填写改动注释的标题的话是无法提交改动的。）

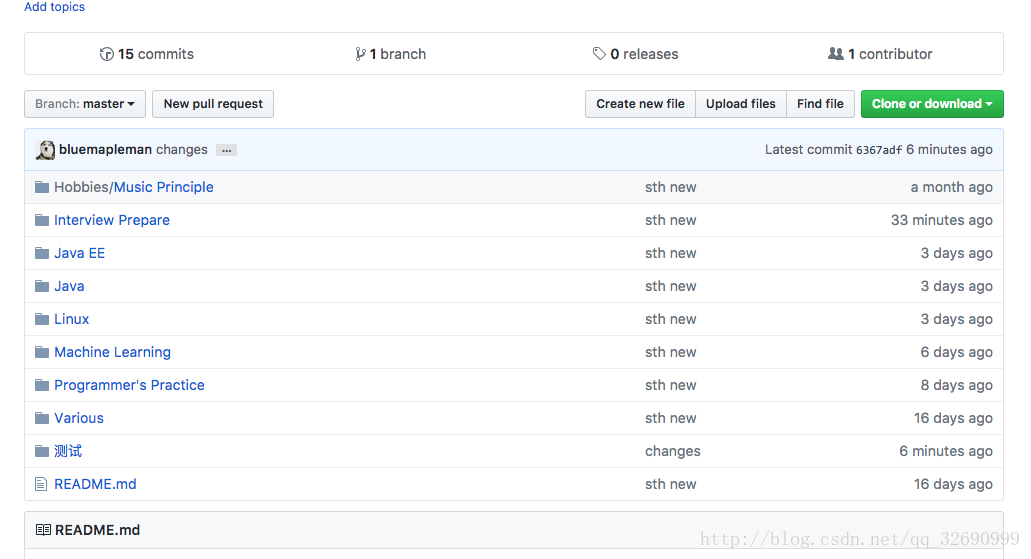


3.commit后，你会发现你的项目历史结点上多了一个新的环，代表你的上一次commit记录。



4.但是commit后只是将你的改动信息保存了下来并提交到本地代码库，如果要让远程仓库/Github页面也变成和本地仓库完全一样的状态，还需要点击一下【Repository】-【Push】，代表将新提交的本地改变**推进到远程仓库**。另外，也可以点击【Repository】-【Sync】，可以完成一样的效果，只是sync同步[顾名思义](https://www.baidu.com/s?wd=%E9%A1%BE%E5%90%8D%E6%80%9D%E4%B9%89&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)，还包含将远程仓库的改变更新到本地的动作，换言之，sync可以看做push和pull操作的联合体。

5.待sync完成后，小环变成了一个点，代表同步完成，然后打开github的网站，进入到项目下，就会发现改动都同步了（新增了一个测试文件夹）。



**建立新的分支 (creating new branch)**

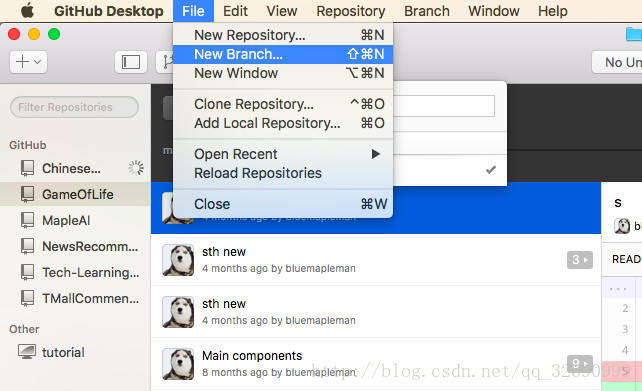
Github项目中还有一个很重要的概念是**分支**（Branch）。

假设，你有一天发现，你的项目可能用普通Java项目的结构来做和用Java EE项目的结构来做都可以，但是你不确定到底用哪种好，于是你和朋友商量一个人做一版。但是，你们做的还是同一个项目，只是架构不同，那么这时，你们就可以开两个分支（Branch），每个分支分别代表一种架构方式。

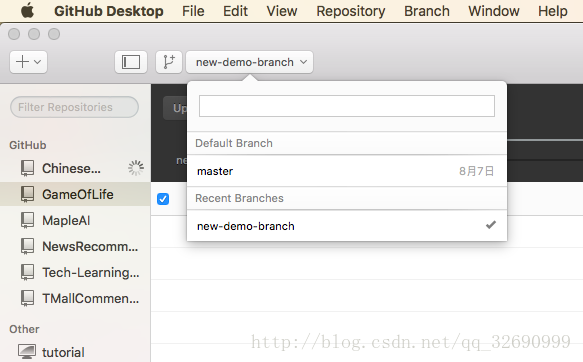
又或者，你和朋友一起翻译一本中文书，你负责翻译成英文，你朋友负责翻译成法文，那么你们也可以分别在两个分支上操作，而这两个分支都属于这本书的翻译项目。

简而言之，所有分支同属一个项目，但每个分支可以互相独立，互不干扰，而有需要时，任意两个分支也可以合并。

在Github Desktop中，我们首先选定要创建新分支的项目，然后点击【File】-【New Branch】（如下图），输入新分支的名字，并且要选择分支克隆的范本，即新分支以哪个已有当前分支为范本进行拷贝。

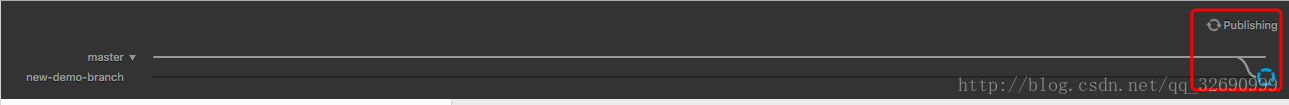


接着，在【Repository】-【show branches】中你就可以看到现在已有的两个分支，一个默认的”Master”，一个新建的你自己命名的”new-demo-branch”，而分支后面的小勾代表你现在操作的分支。



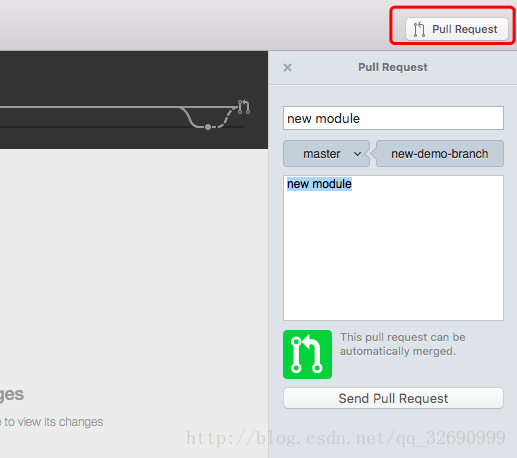
接下来，我们在当前分支下进行的操作，无论是增删改文件，都是针对当前分支的文件，不会对其它分支造成任何影响。

不过，你新建的这个分支目前在github页面是看不到的，因为它还没有公开（publish），所以你会看到，新分支的进度条右侧，会有一个”publish”按钮，当你点击它之后，新的分支才会在页面显示出来。

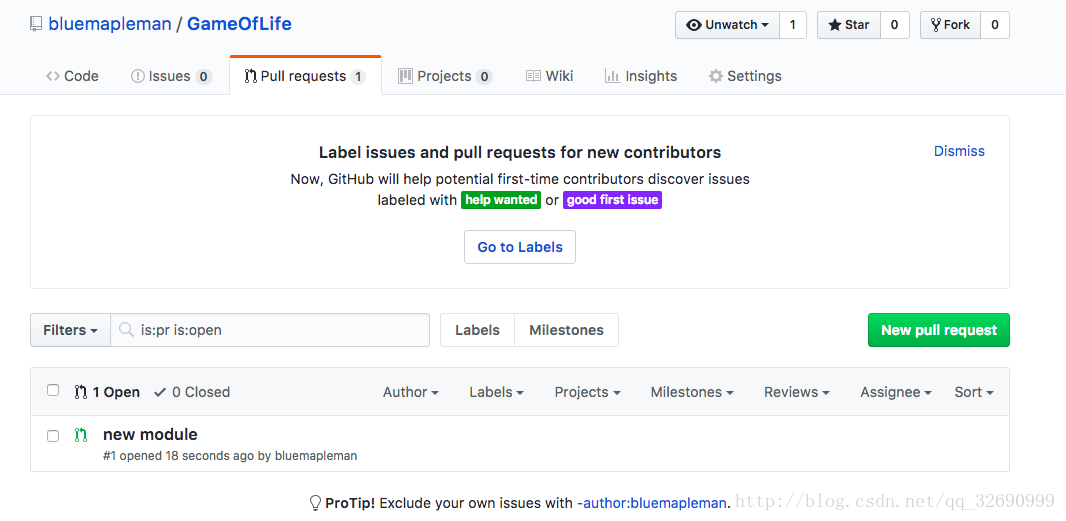


**请求合并自己的代码 (pull request)**

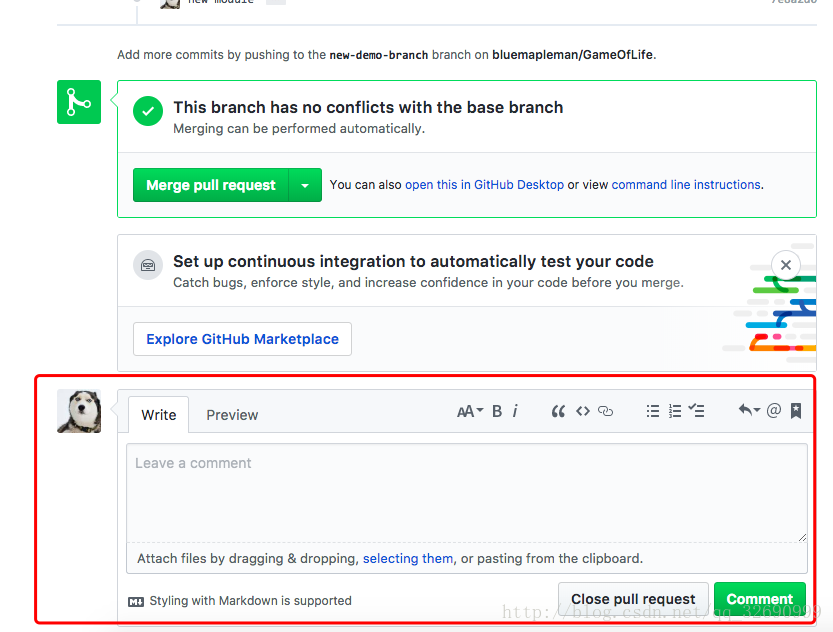
当你写好一版很厉害的代码，觉得是对当前项目的一个非常棒的改进后，你希望可以直接把你的代码并入到项目中去，那么你就可以发起一个pull request。



写好pull request的理由，然后send出去，接着就可以在github页面看到项目下的一个pull request。



在网页端这边，项目负责人就可以对pull request进行讨论和审核，如果大家审核通过，就可以选择【Merge Pull Request】，表示允许合并pull request提交的改动。



为什么要pull，而不是直接自己commit的一个重要原因在于：一般情况下，很多人是fork别人的项目，然后自己进行改进，但是你对别人的项目是没有改动权力的，否则就乱套了，你只是能改变你从别人那边fork过来的项目。但是你觉得自己的改进非常棒，想贡献自己的代码并成为原项目的contributor之一，于是你就需要发动一个pull request，请求原项目的负责人将你的代码合并的原项目中。

另外，pull request中的内容可以很方便地被项目负责的所有人讨论，并共同决策要不要把这些新的改变并入到项目中，这样就可以防止某个贡献者可能因为考虑不全面，提交了不合理的改动，进而对项目造成损害。

转载自：   
<https://blog.csdn.net/qq_32690999/article/details/77504033#%E5%9C%A8%E6%9C%AC%E5%9C%B0%E6%8F%90%E4%BA%A4%E6%96%B0%E7%9A%84%E4%BB%A3%E7%A0%81%E7%89%88%E6%9C%AC%E5%88%B0%E4%BB%93%E5%BA%93-push-local-changes-to-remote-repository>