**github上fork别人的代码之后,如何保持和原作者同步的更新**

**1.使用git指令同步**

首先要先确定一下是否建立了主repo的远程源：

git remote –v

如果里面只能看到你自己的两个源(fetch 和 push)，那就需要添加主repo的源：

git remote add upstream URL // URL为原作者的URL

git remote -v

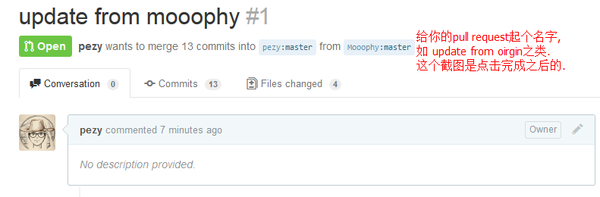
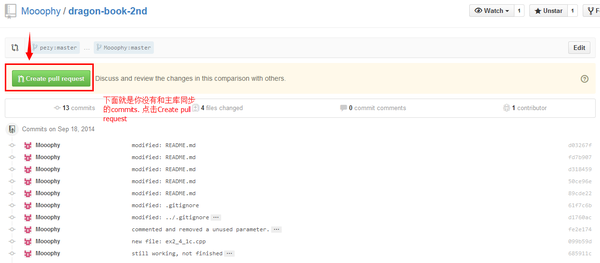
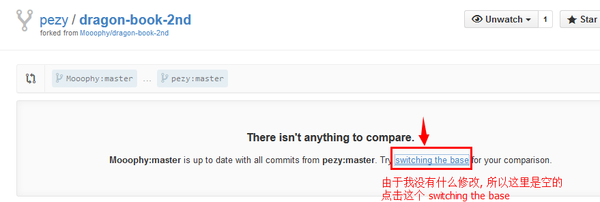
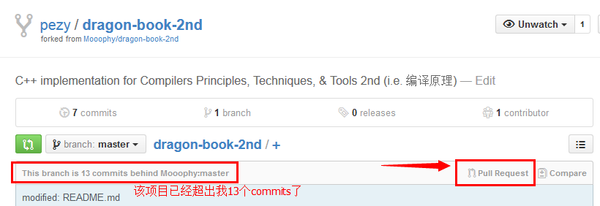
然后你就能看到upstream了。

如果想与主repo合并：

git fetch upstream

git merge upstream/master

注：删除repo源：git remote rm upstream

**2.只用Github不用命令行的方法:**  
  
举例说明: (每一步都有截图)  


**查看当前的远程库**

　　要查看当前配置有哪些远程仓库,可以用 git remote 命令,它会列出每个远程库的简短名字.在克隆完某个项目后,至少可以看到一个名为 origin 的远程库,Git 默认使用这个名字来标识你所克隆的原始仓库:

　　$ git clone git://github.com/schacon/ticgit.git

（1）git remote 不带参数，列出已经存在的远程分支

　　$ git remote

（2）git remote -v | --verbose 列出详细信息，在每一个名字后面列出其远程url

此时， -v 选项(译注:此为 –verbose 的简写,取首字母),显示对应的克隆地址:

　　$ git remote -v

这样一来,我就可以非常轻松地从这些用户的仓库中,拉取他们的提交到本地.

**添加远程仓库**

　　要添加一个新的远程仓库,可以指定一个简单的名字,以便将来引用,运行 git remote add [shortname] [url]:

　　$ git remote

　　$ git remote add pb git://github.com/paulboone/ticgit.git

　　$ git remote -v

　　pb git://github.com/paulboone/ticgit.git现在可以用字串 pb 指代对应的仓库地址了.比如说,要抓取所有 Paul 有的,但本地仓库没有的信息,可以运行 git fetch pb:

　　$ git fetch pb

现在,Paul 的主干分支(master)已经完全可以在本地访问了,对应的名字是 pb/master,你可以将它合并到自己的某个分支,或者切换到这个分支,看看有些什么有趣的更

**通过git remote 建立远程仓库**

1.初始化一个空的git仓库

$ mkdir yafeng

$ cd yafeng/

$ git init

在上面的命令中，真正去初始化的是那句---git init