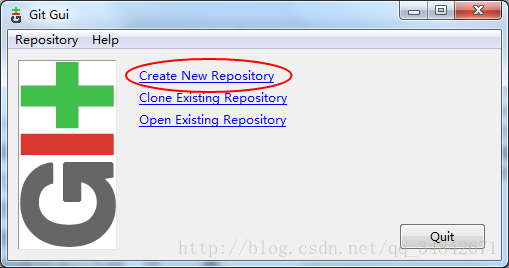
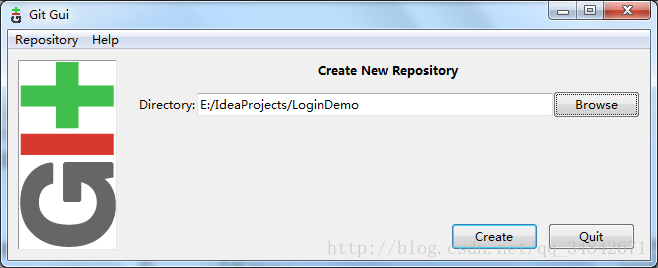
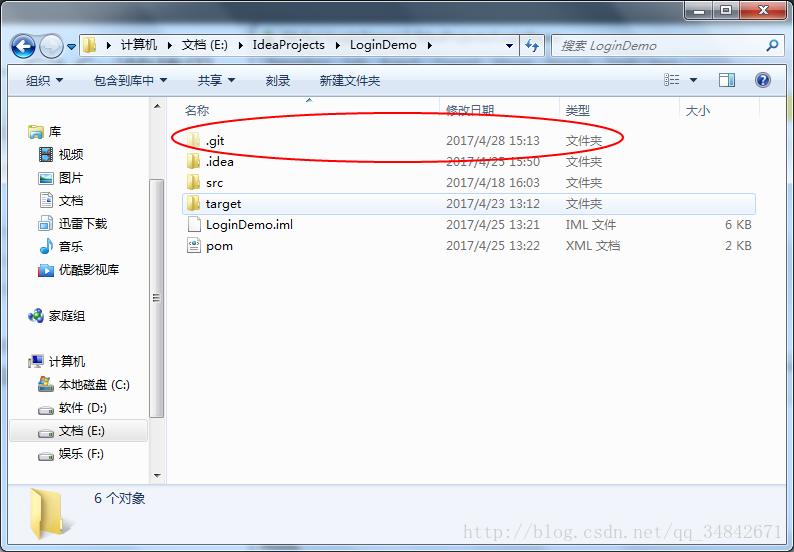
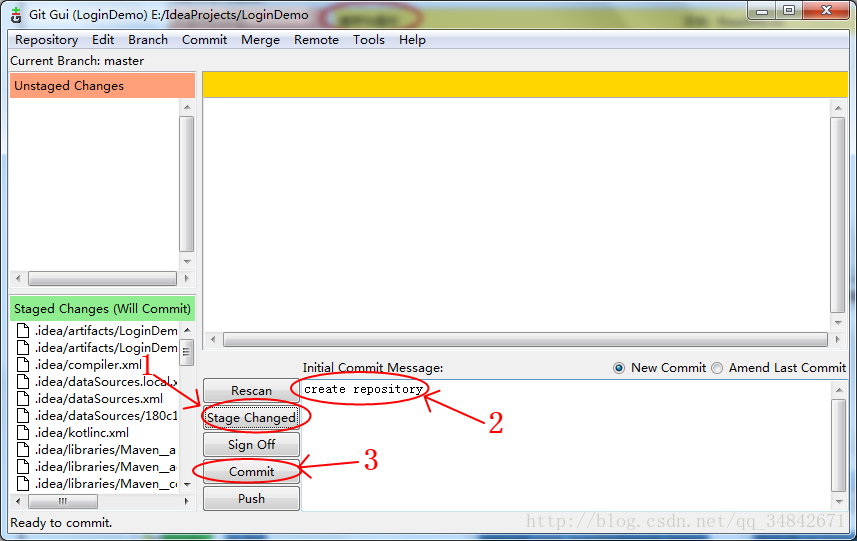
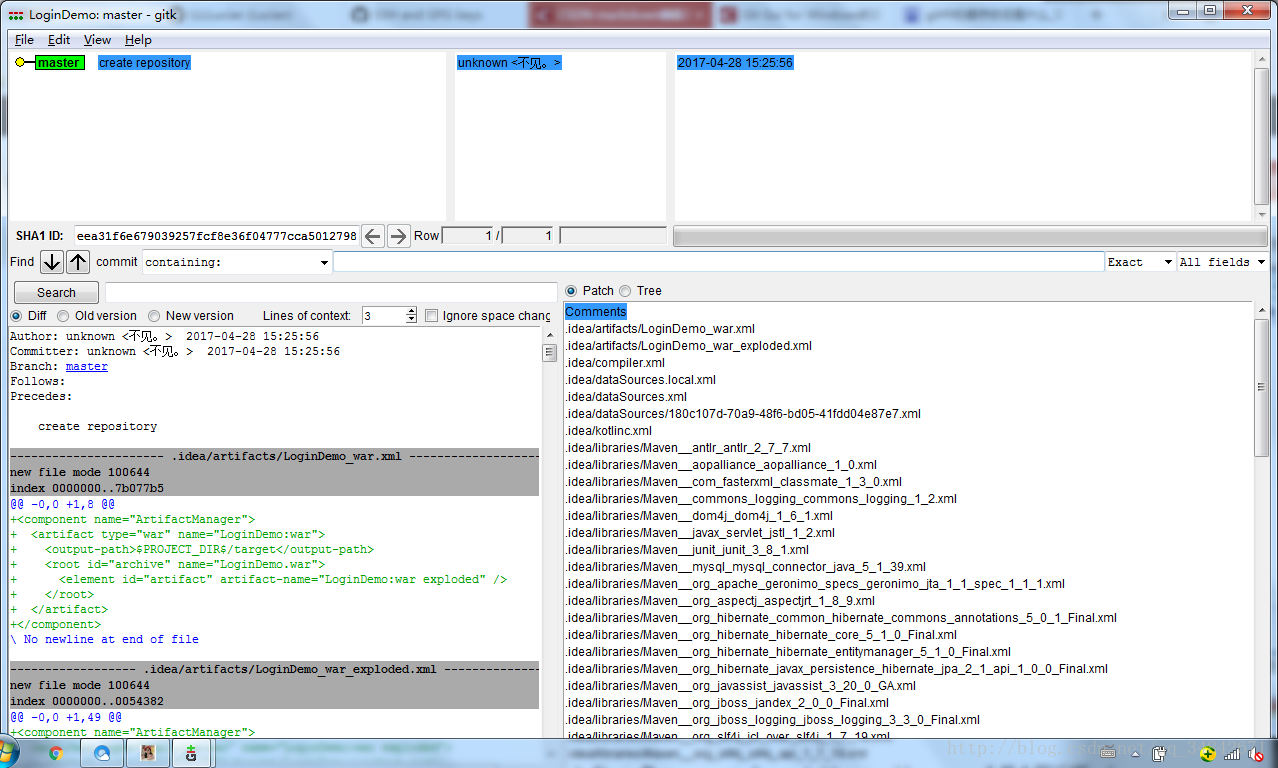
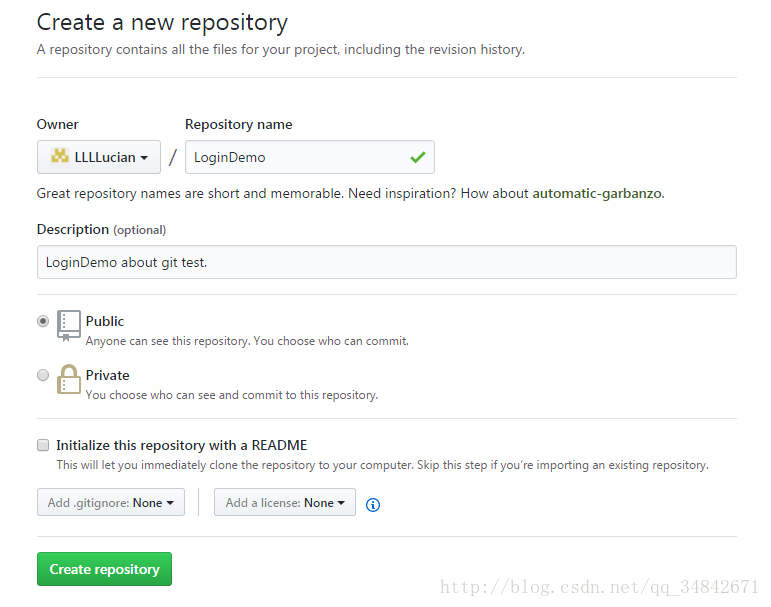
[**Git GUI可视化操作教程**](https://www.cnblogs.com/zhoumiao/p/8018249.html)

**1.在本地新建版本库**

首先，我们打开Git GUI是这样的一个界面，选择第一项，新建版本库。   
   
然后选择你需要进行版本管理的项目路径，我选择了一个LoginDemo的项目。   
   
当你创建了版本库的时候，你可以在该项目的路径下看见多了一个.git文件夹（设置了隐藏文件不可见的话看不见）   
  
在Git Gui中，如果Unstaged Changes（未缓存的改动）中包含文件，则先点击Stage Changed，将未缓存的改动加入缓存，再点击Commit；否则需要先点击Rescan扫描项目中进行过改动的文件。注：提交描述必须要填写，不然无法提交。   
  
提交成功后，我们可以在Repository->Visualize All Branches History中看到我们提交的历史记录（这是提交到本地而不是提交到远程服务器）。至此，本地版本库就创建成功了。   


**2.将项目提交到远程**

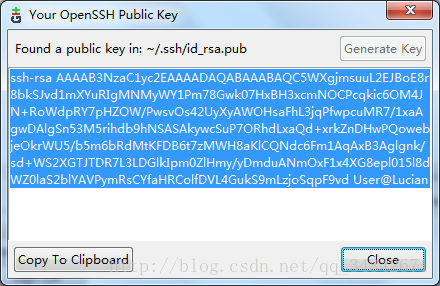
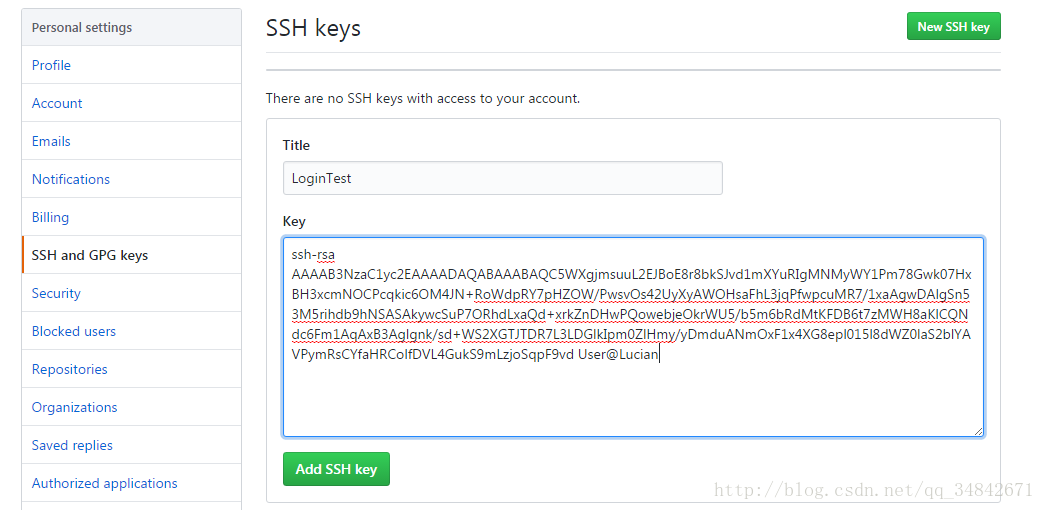
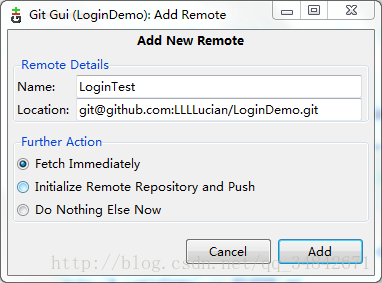
上一步已经使用Git在本地建立起了版本库，然后我们需要将该项目提交到远程服务器以便同事或其他合作者共同参与开发。一般的开源项目使用github作为远程服务器。其实在本地简历版本库后，只需要点击push就可以直接上传。但是我们还没有将本地的Git与github建立联系以及设置安全协议。

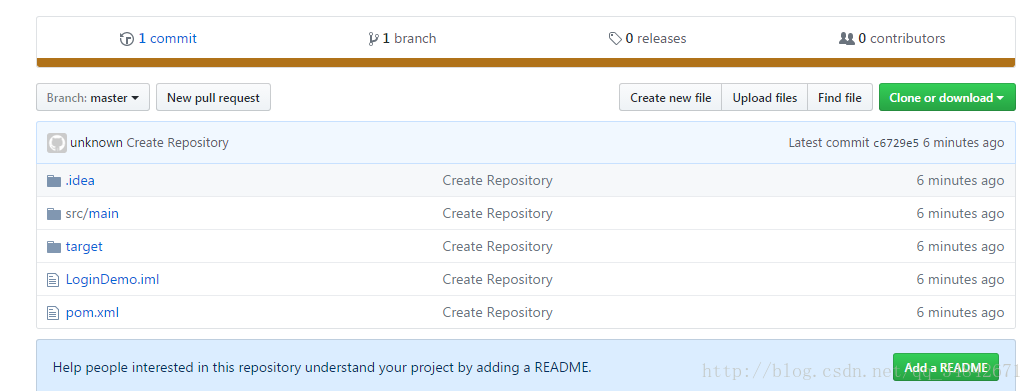
首先，我们需要在有一个自己的github帐户，然后在github上新建一个repository，名字也叫做LoginDemo（可以和本地项目名不一样）。   
   
在Git Gui中，选择Remote->add添加远程服务器，远程服务器信息有两种填写方式，填写https地址或ssh地址，对应github项目的https和ssh地址，推荐使用ssh方式。

**1.https地址**

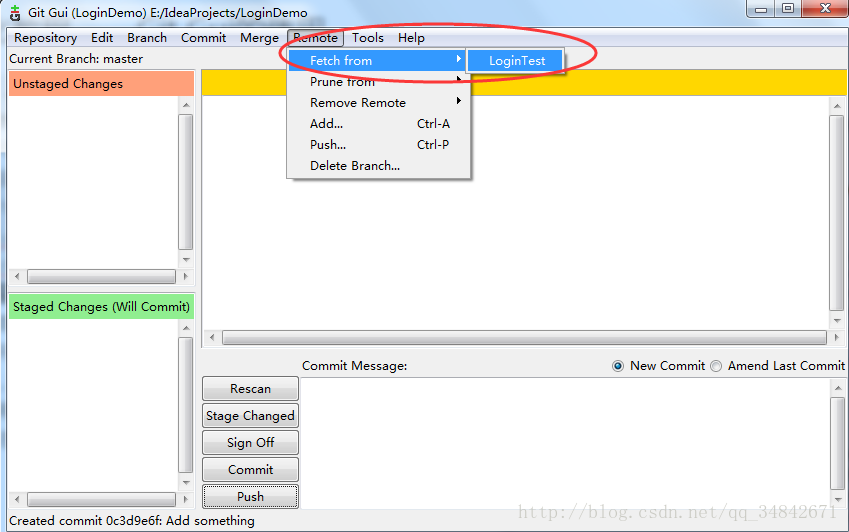
这种方式需要输入你的github帐户和密码，意味着通过这种方式你只能够操作自己的项目。 

**2.ssh地址**

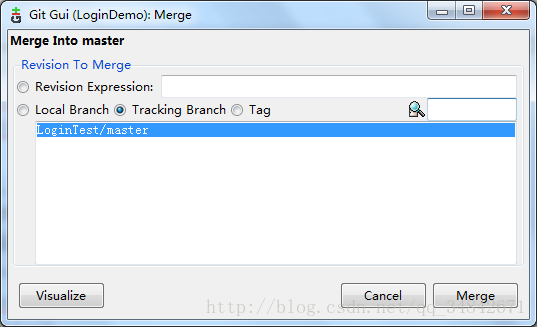
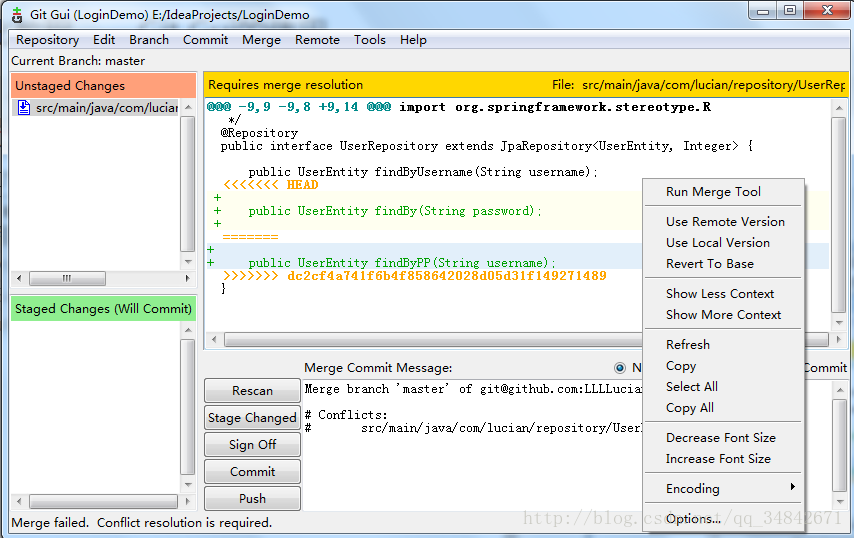
这种方式需要进行授权设置，在Git Gui的菜单栏，点击Help->Show SSH key->Generate SSH KEY创建密钥。   
   
然后在github的Personal settings中添加它，title随意，可以用Home，company等作为标识来区别。   
  
添加远程服务器信息。   


接下来，我们便可以直接在Git Gui点击push提交至远程客户端，刷新一下github，便可以看到项目已经在repository中了。   


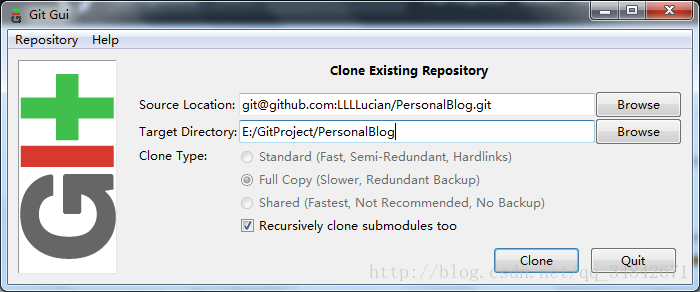
**3.从远程下载更新**

设想多人参与项目开发，每个人都只push到远程，完全不顾其他成员的进度，这样的开发模式无疑是有问题的。我们最好的做法是，每次push到远程的时候，先从远程把目前进度fetch下来，在和自己目前项目进度merge后，再将它push到远程。打开Git Gui，在菜单栏中选择Remote->Fetch from->LoginTest，便可以从远程服务器更新到本地，但是尚未与自己当前项目合并。   


**4.合并解决冲突**

从远程fetch后，选择Git Gui的Merge->Local Merge进行合并，选择Tracking Branch。   
   
如果本地有一个方法名叫findUser，而远程服务器中该方法的名字改变了，例如变成了findPassword，则合并遇到冲突，可以右键空白部分选择保留本地版本或远程版本进行合并（图中绿色的代码表示冲突部分）。   


**5.克隆github上的项目到本地**

在Git Gui的主界面选择克隆已有版本库。   
   
Source Location即github中项目的地址，和第二部分（将项目提交到远程）中一样，可以选择https地址或ssh地址，Target Directory是在本地存放该项目的路径。点击Clone，成功从github将项目克隆到本地。   


**Git可视化极简易教程 — Git GUI使用方法**

*分类*[**编程技术**](http://www.runoob.com/w3cnote_genre/code)

**前言**

之前一直想一篇这样的东西，因为最初接触时，我也认真看了廖雪峰的教程，但是似乎我觉得讲得有点多，而且还是会给我带来很多多余且重复的操作负担，所以我希望能压缩一下它在我工作中的成本，但是搜索了一下并没有找到满意的教程，新的一年自己梳理一下自己的经验。

可能男生们大神比较多，觉得Git是如此简单，便已觉得命令行操作就是SO EASY，甚至或许有看不起可视化这样面对低端用户的心理，好的，那您就当我是水货可以右上角了。

我一直觉得类似GIT这样的东西，他对于我而言只是个不完全必须的工具，我并不想成为使用他的专家，类似的东西，今天有GIT，明天可能有GAT，或者GAY？所以快速地掌握它我需要的重要日常操作，最好是10分钟，那就好了，如果你有类似的想法，好吧，那不要废话了，咱们赶紧开始。

**（全文限windows系统。）**

**何为GIT？**

[安装GIT](http://git-scm.com/download/)，大致了解下[GIT是做某子的](http://baike.baidu.com/link?url=_aNWZrVpQm9L89S4CYR66kVd6MZhQWXY5mO8zJfoTCEvQ7rkZdZOYOOyzA5IGO6kL2hw34M7r2wXixw6GNTTmk9kRAXHuIav23kej67ITc7)。

**权限校验**

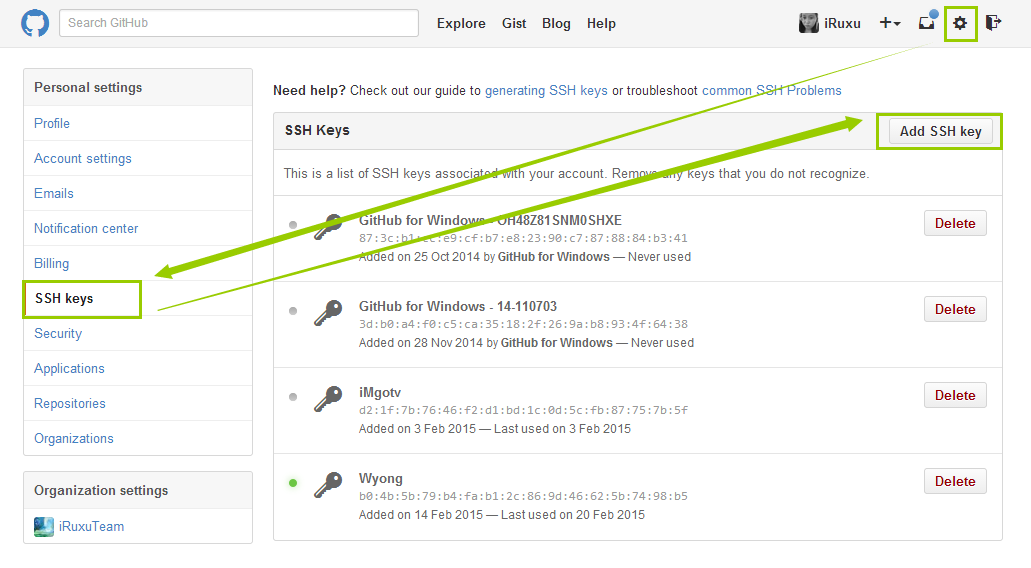
首先,您的数据保存在远端服务器一份，服务器需要对您的身份识别。一段RSA加密字符串。

启动GUI，菜单-帮助，【Step1-**创建密钥】**Generate SSH KEY

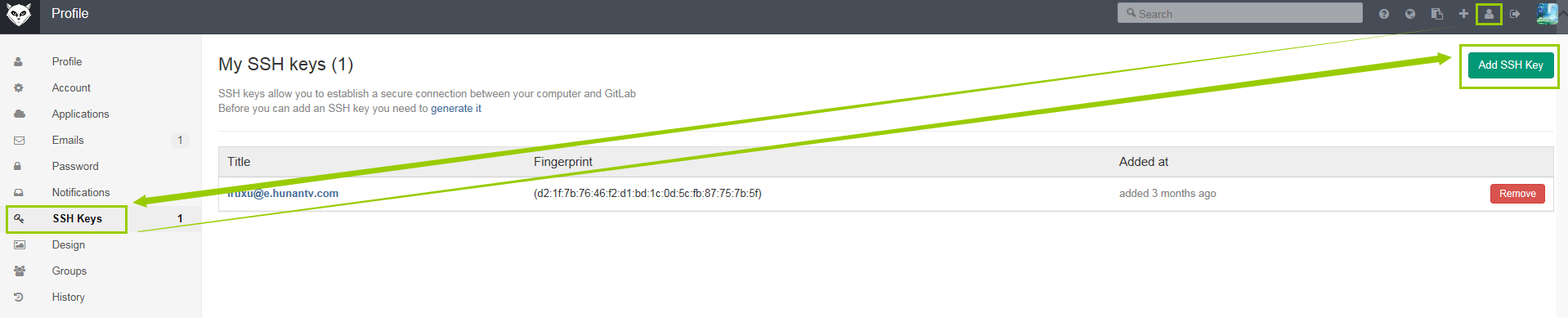
[](http://www.runoob.com/wp-content/uploads/2015/03/271102462211960.png)

【Step2-**添加密钥】**去你的代码托管服务器，你的账号设置中，添加它。

比如在Github中的地址，title随意，比如你可以用Home,company等作为标识来区别。

[](http://www.runoob.com/wp-content/uploads/2015/03/271104566587275.png)

Gitlab中的演示

[](http://www.runoob.com/wp-content/uploads/2015/03/271108115808959.png)

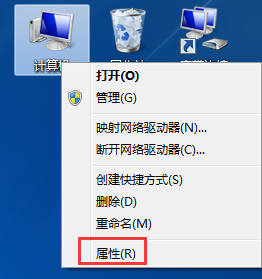
**账号保存**

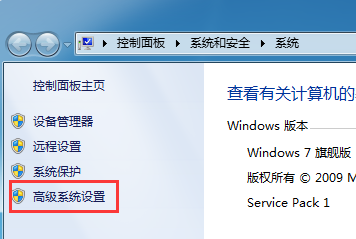
如果不做设置的话，每次提交的时候，都会询问你填写密码。于是我们先来把这个设置好。

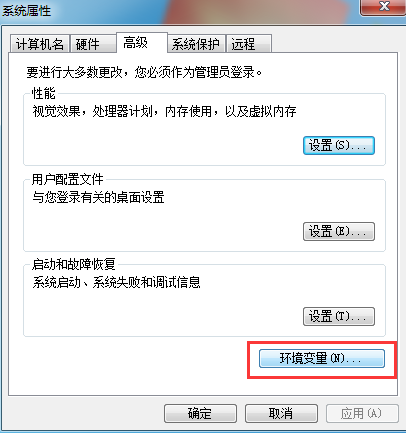
【Step3.1-添加环境变量**】**

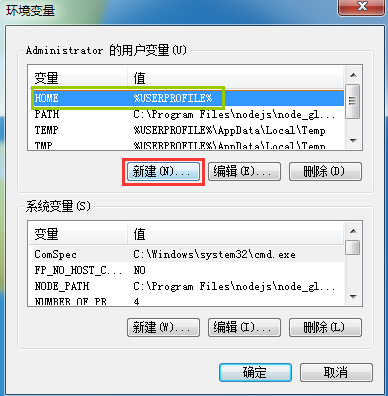
**我的电脑 - 属性 - 高级系统设置 - 环境变量 - 新建变量**

**变量名HOME，变量值%USERPROFILE%**

**[](http://www.runoob.com/wp-content/uploads/2015/03/271117100641718.png)**

[](http://www.runoob.com/wp-content/uploads/2015/03/271117191748667.png)

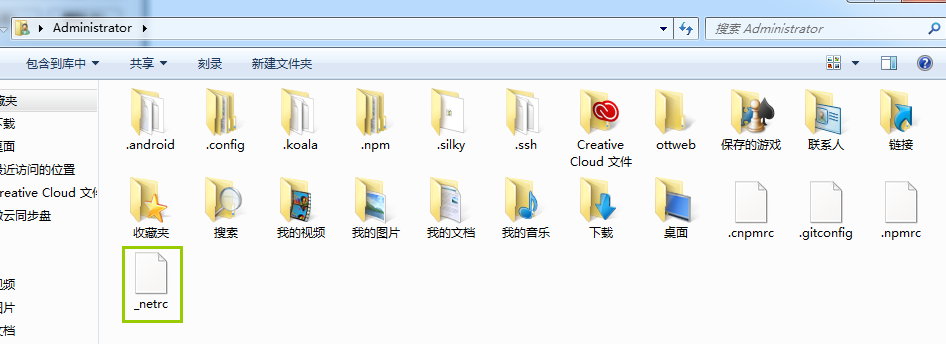
[](http://www.runoob.com/wp-content/uploads/2015/03/271117300493045.png)

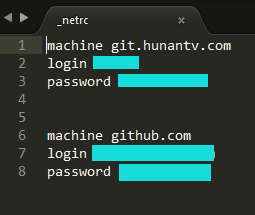
[](http://www.runoob.com/wp-content/uploads/2015/03/271117384245853.png)

**【Step3.2-创建账号文件】**

**开始 - 运行 中打开%Home%，即windows的管理员账号文件夹。**

**新建一个名为"\_netrc"的文件，填写你要保存的服务器地址及账号密码，保存。**

**[](http://www.runoob.com/wp-content/uploads/2015/03/271123307214691.png)**

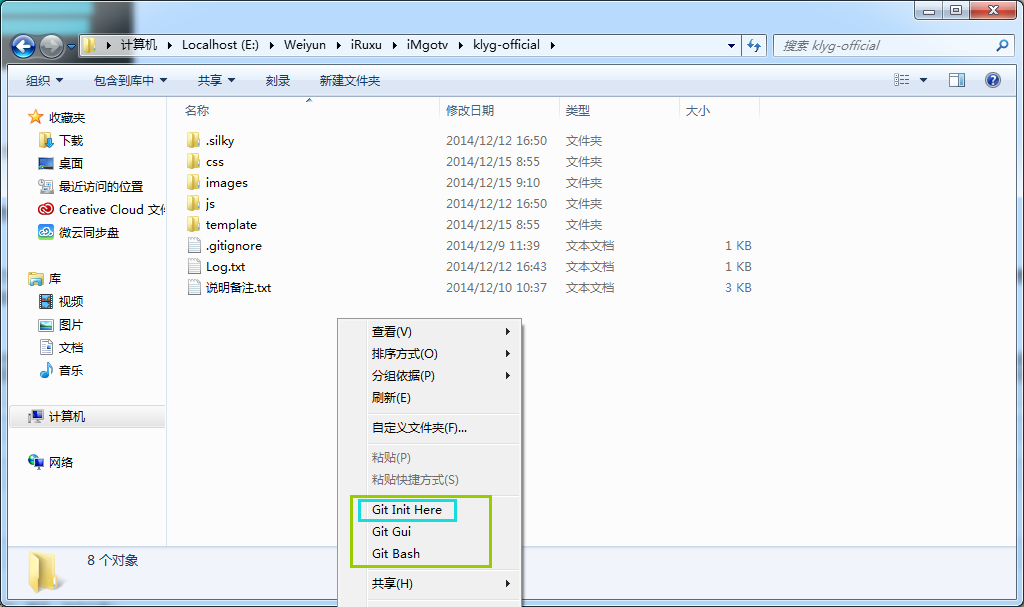
[](http://www.runoob.com/wp-content/uploads/2015/03/271123401437312.png)

**操作流程**

如果你用过SVN的话就会大致了解操作流程，如果没有也没关系。

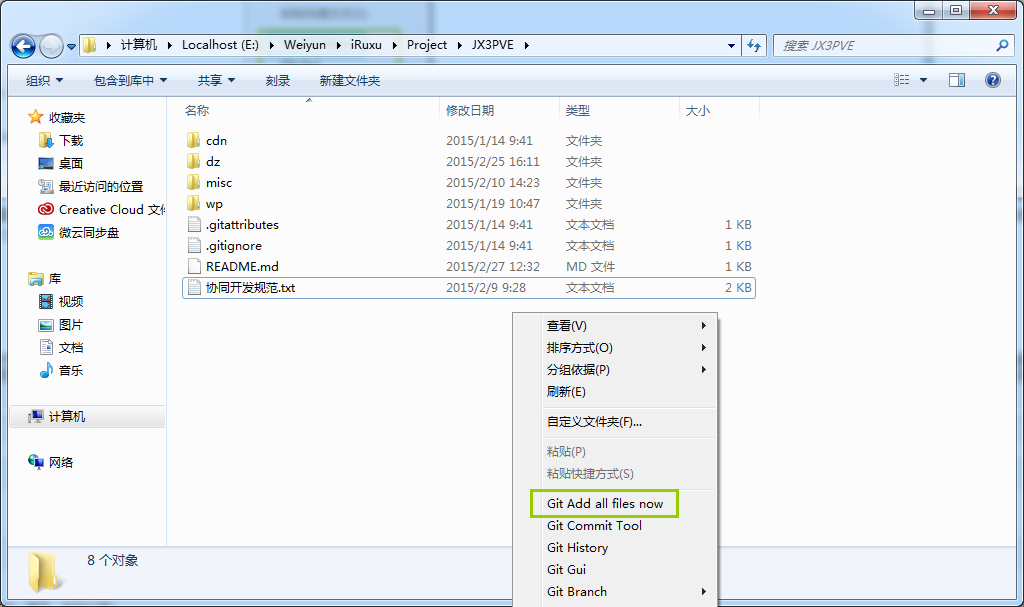
**初始化**（Git init）

顾名思义，就是新建一个项目，跟你用PS新建一张画布一样。在你新建好的文件夹中右键创建即可，若点击Git bash则以此目录作为当前目录进入命令行状态。

[](http://www.runoob.com/wp-content/uploads/2015/03/271234025648572.png)

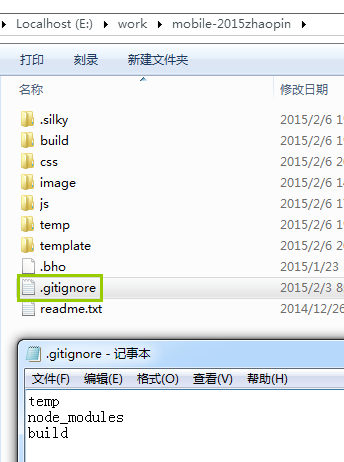
**添加**（Git add）

添加并不是提交代码到远程Git库，Git也并不会你修改了代码它自动帮你保存你修改的每一个过程。你修改了很多文件，但未必所有的修改，最终打算提交上去，那么哪些是你打算提交的，你可以添加进来待会提交，叫做缓存改动。很简单，比如本地电脑上我有整个项目完整的东东，甚至包含了账号密码的一些文件，但是我只是ADD除账号密码之外的文件，并不缓存账号密码文件的改动。不被ADD它就不会参与后续的操作。通常我都会直接全部缓存，它会自动寻找所有有改动的文件，而不需要提交的文件放在忽略的文件夹中。（关于忽略下面我们就会说到）

[](http://www.runoob.com/wp-content/uploads/2015/03/271238259396476.png)

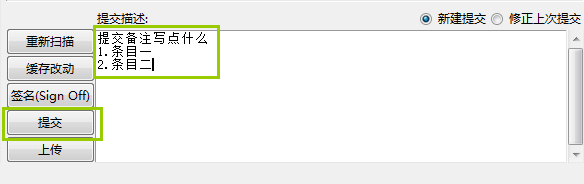
**忽略**（.gitignore）

但实际上大部分我们的文件都是一起提交的，并不会逐一去甄选，又或者类似PSD这样的大源文件以及并不作为产品最终展示的过渡文件，我们可以统一放在临时文件夹中，并忽略此文件夹。

[](http://www.runoob.com/wp-content/uploads/2015/03/271134259394618.png)

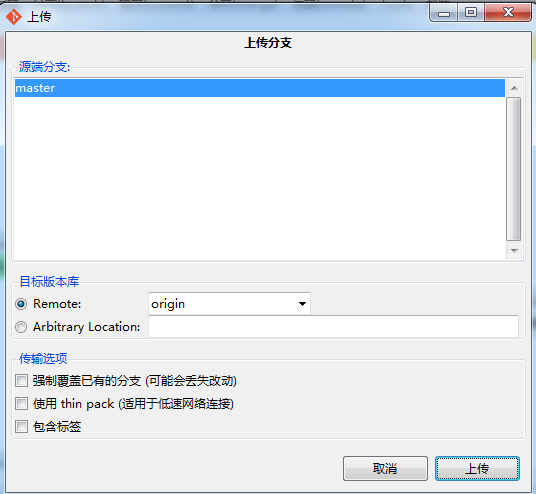
**提交**（Git commit）

提交则代表此前被添加ADD的文件已确认被提交到Git库了。需要注意的是，如果你改变代码的缩进（尽管没有修改内容），默认状态下会被识别为整个代码全部变更。**提交的时候是要求必须要写备注的**。

[](http://www.runoob.com/wp-content/uploads/2015/03/271240398302205.png)

**上传**（Git push）

顾名思义，上传则是上至远端服务器了，小伙伴们可以看到咱们的渣渣代码了（好羞涩。

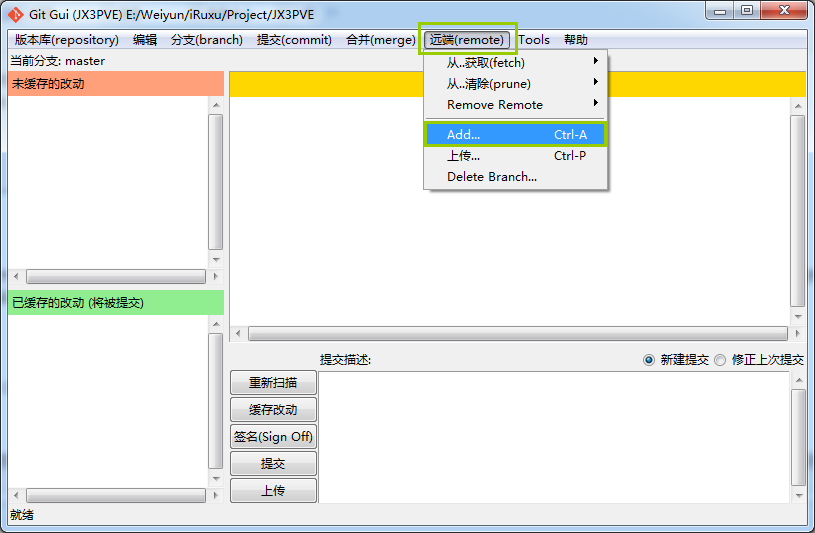
[](http://www.runoob.com/wp-content/uploads/2015/03/271241402522046.png)

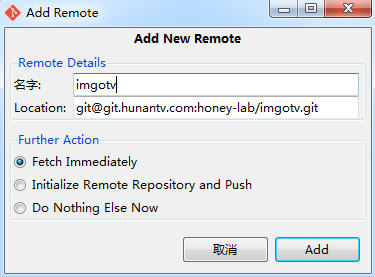
**获取远程代码**（Git remote/fetch）

比如你在公司做好的东东，今夜难眠十分亢奋，回家准备继续搬砖，那咱们就在家里的电脑上，同上进行好各种安装配置账号，先把公司做好的东东嫩下来（不过公司是内网不可以，但是假如是Github上是可以的）。又或者和小伙伴一起开发个啥，你也要先fetch他的下来。至于怎么操作，下面上图。现在你只要知道，大大们下齿全露刷牙表情对你口口念念的fetch是个啥子～

[](http://www.runoob.com/wp-content/uploads/2015/03/271145255335341.png) 来，跟我念，fetch~~（我怕你们脑补不出来……）

**先来设置与远程地址的关联，Git remote：**

[](http://www.runoob.com/wp-content/uploads/2015/03/271242561118002.png)

[](http://www.runoob.com/wp-content/uploads/2015/03/271244220336453.png)

填写SSH地址与项目名。下面有3个选项：

第一个：立刻获取最新改动（所以如果是本地克隆远程一个项目，也可以这样操作）。

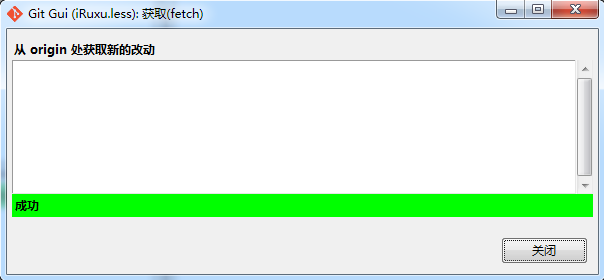
第二个：本地新建的项目，初始化远程仓库并发布过去。

第三个：什么也不做。

**在项目的进行过程中，获取仓库的最新改动Git fetch**

选择从远程仓库哪个分支中获取更新，如果没有则只有主支。

提示成功则改动的已经被存放到临时区了，你一会还需要进行合并操作，如果没有任何改动，则列表中是空的，比如：

[](http://www.runoob.com/wp-content/uploads/2015/03/271300329558531.png)

**合并**（Git merge）

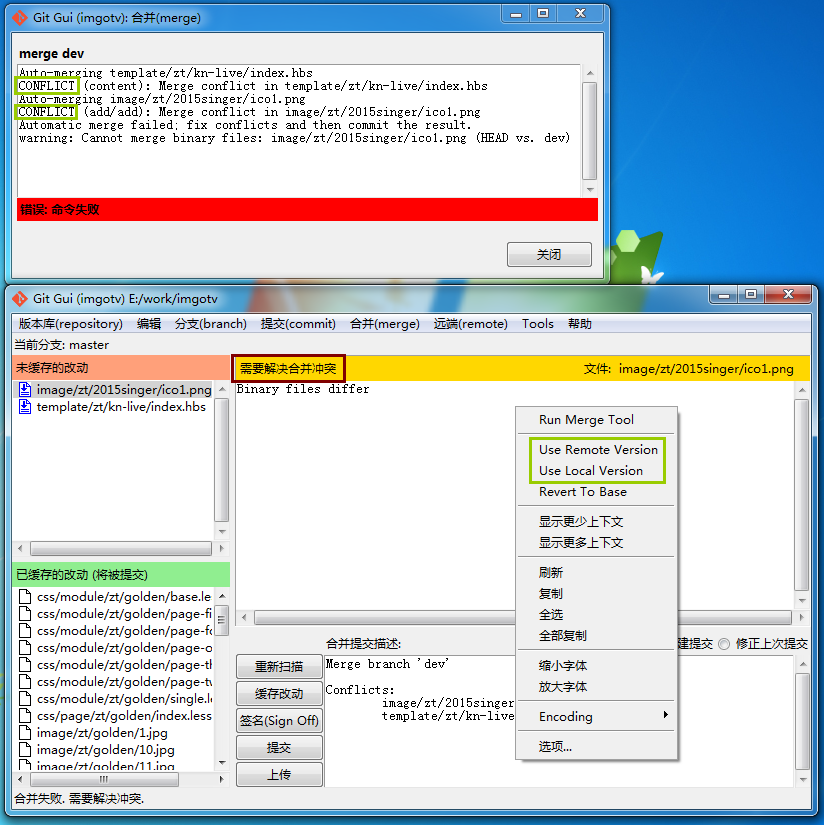
请注意啦，不管你本地有没有代码，fetch之后呢，是都要merge的，也就是说，fetch下来后，大大的代码还在一个小黑屋里，我们需要把它装到自己兜里。

选择合并 - 本地合并，然后选择本地的分支（如果你没有创建分支，则只有1个主支master）

**冲突处理**（Conflict）

合并的过程中可能会出现一些红色的文件与一堆叹号，这时候慌慌张张的点啥它都不管用，不用担心，不是程序坏了，只是有冲突的文件，例如A童鞋写了width:1180px，你写了width:auto。那到底用你们谁的呢。

在GUI界面正文区，**正文区右键**可以选择，Use local version（使用本地版本）或Use remote version（使用远程版本），到底用你的还是小伙伴的？或者你也可以自己再整合。

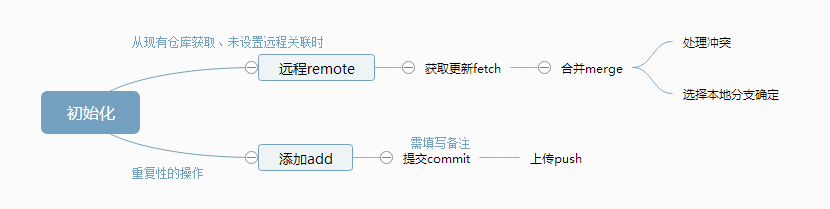
[](http://www.runoob.com/wp-content/uploads/2015/03/271257485644055.png)

其他还有分支和一些高级功能，如果需要了解可以自己再摸索摸索，以上的操作已经可以满足简单的开发需求了。

**总结**

1.先进行安装，密钥添加，账号等一次性操作。

2.操作流程：

[](http://www.runoob.com/wp-content/uploads/2015/03/271314500648180.png)

是不是觉得so easy了呢，赶紧have a try!