# 1/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" \t "_blank)。

**权限校验**

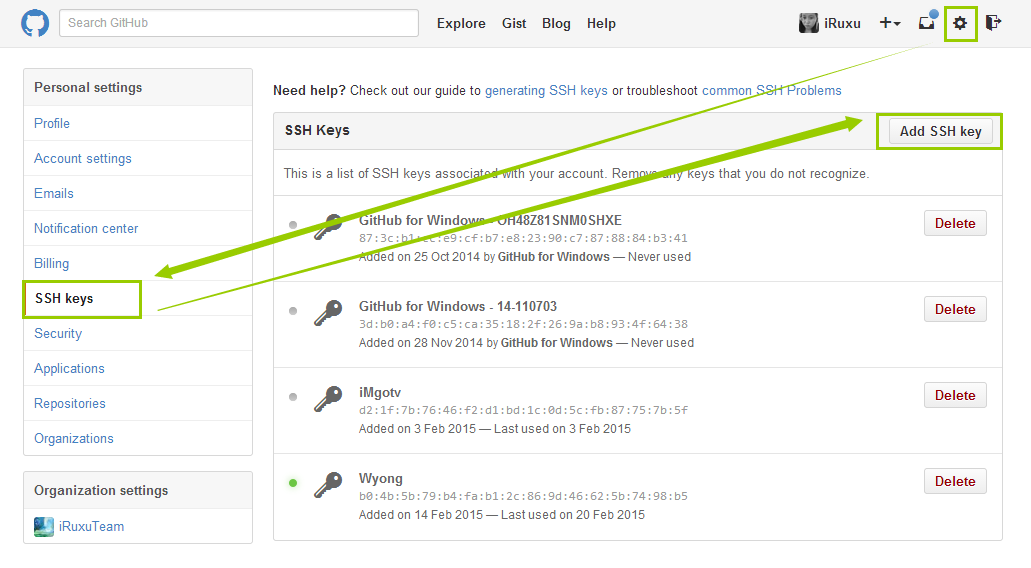
首先,您的数据保存在远端服务器一份，服务器需要对您的身份识别。一段RSA加密字符串。

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

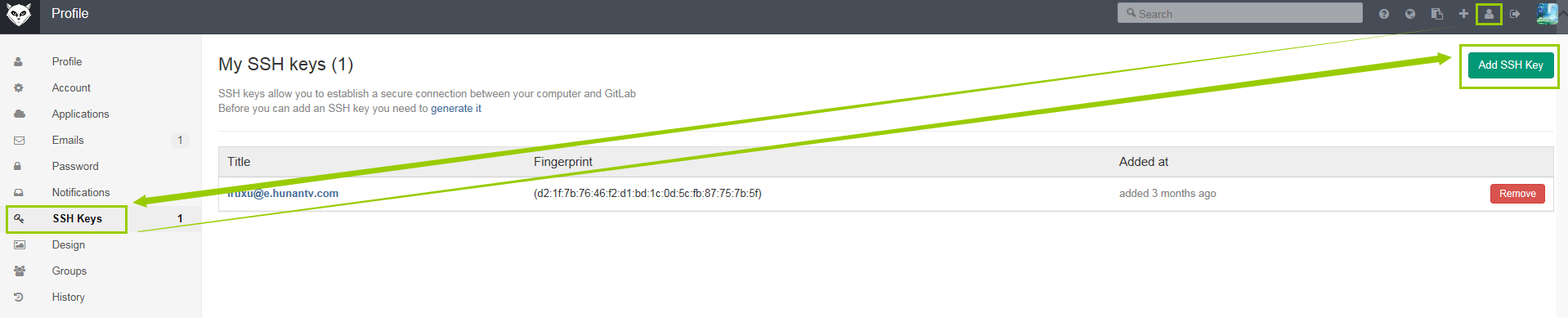
[](http://www.runoob.com/wp-content/uploads/2015/03/271102462211960.png)

【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)

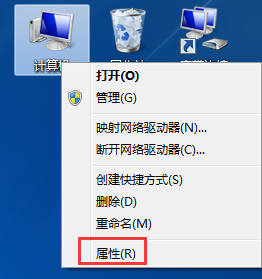
**账号保存**

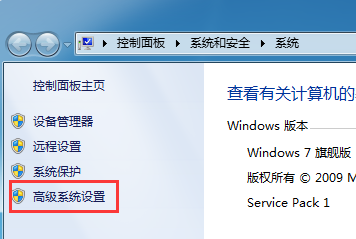
如果不做设置的话，每次提交的时候，都会询问你填写密码。于是我们先来把这个设置好。

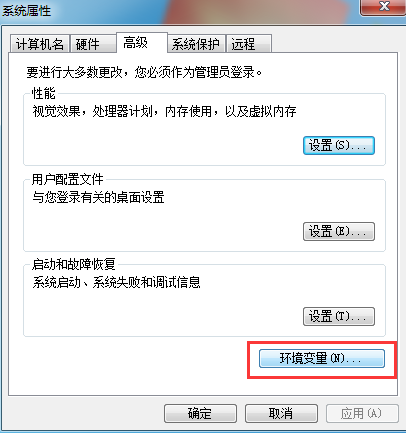
【Step3.1-添加环境变量**】**

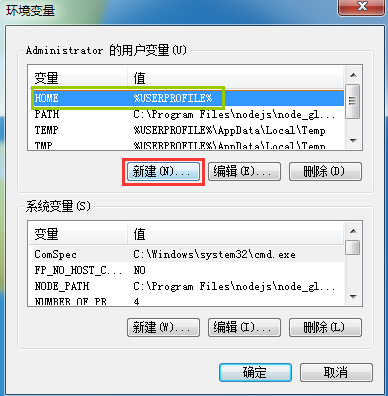
**我的电脑 - 属性 - 高级系统设置 - 环境变量 - 新建变量**

**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)

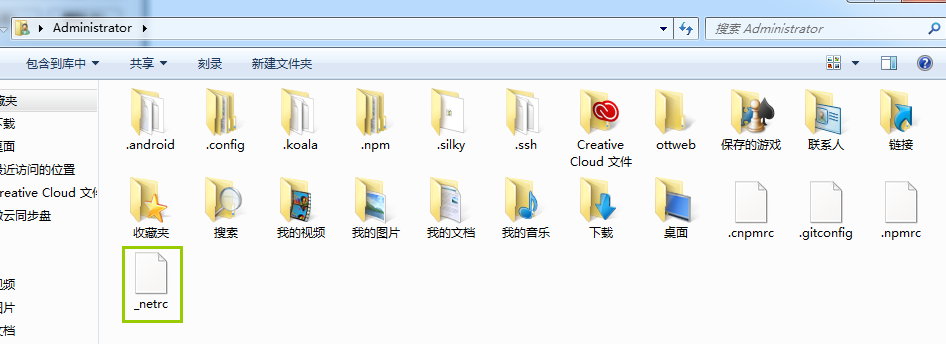
[](http://www.runoob.com/wp-content/uploads/2015/03/271117300493045.png)

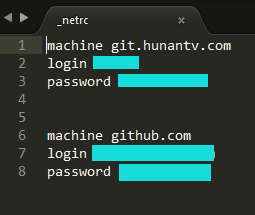
[](http://www.runoob.com/wp-content/uploads/2015/03/271117384245853.png)

**【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)**

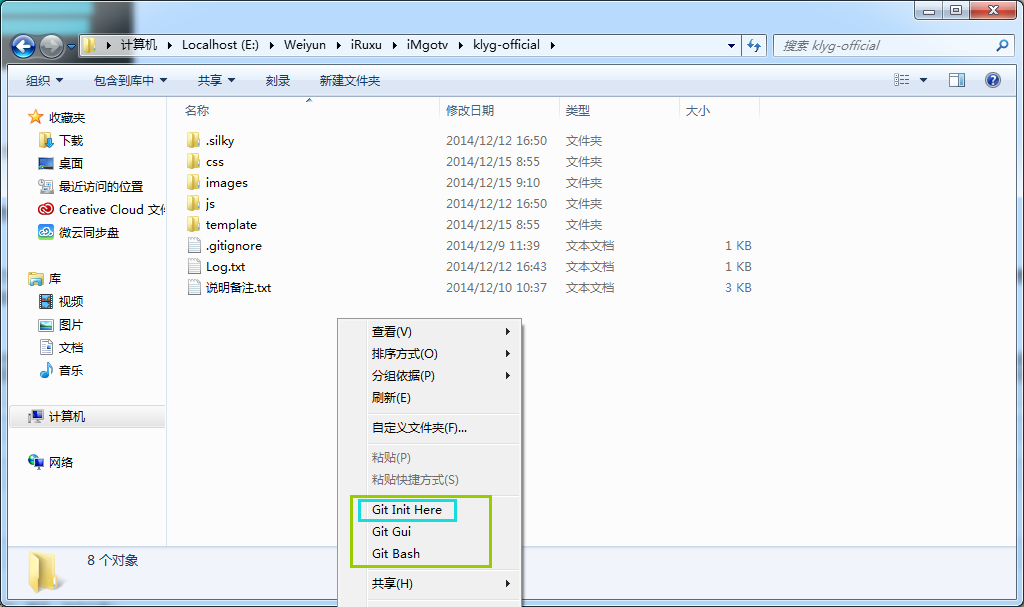
[](http://www.runoob.com/wp-content/uploads/2015/03/271123401437312.png)

**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。

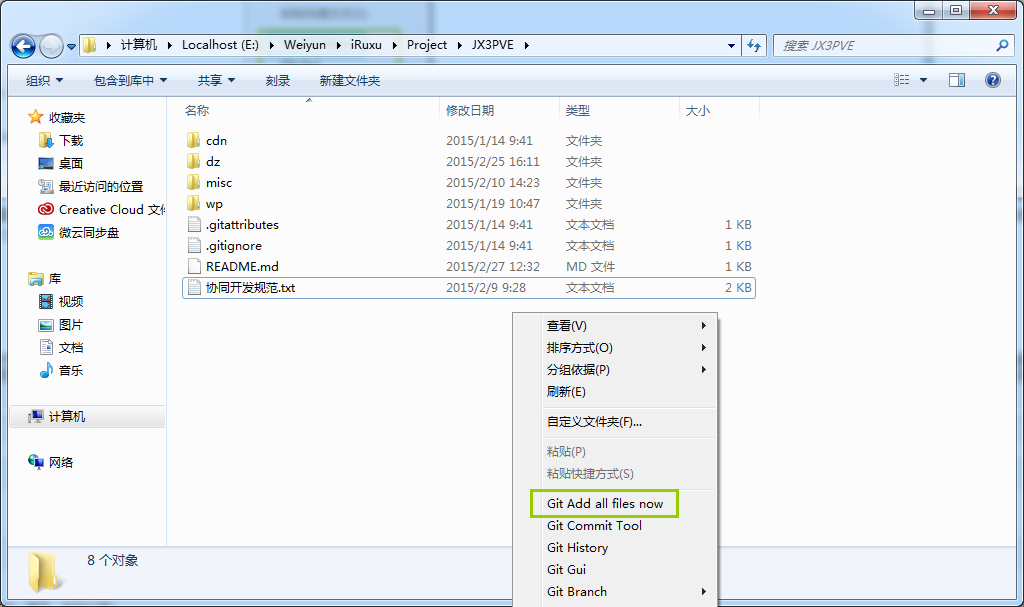
**初始化**（Git init）

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)

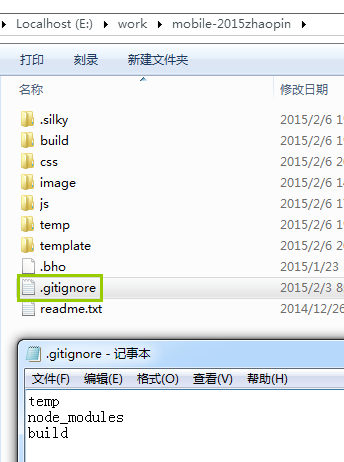
**添加**（Git add）

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)

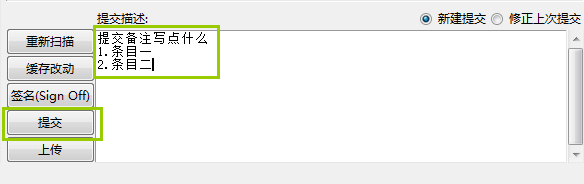
**忽略**（.gitignore）

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)

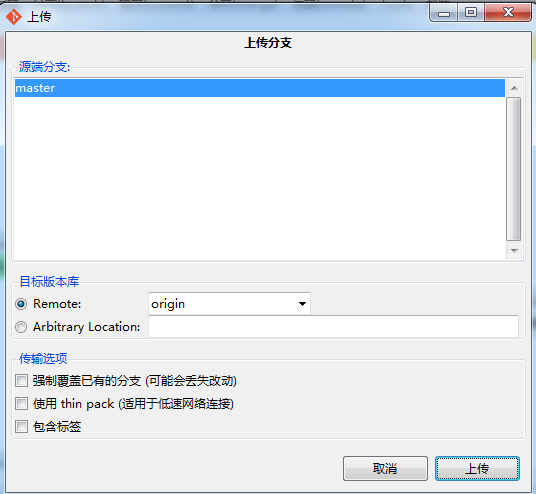
**提交**（Git commit）

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。

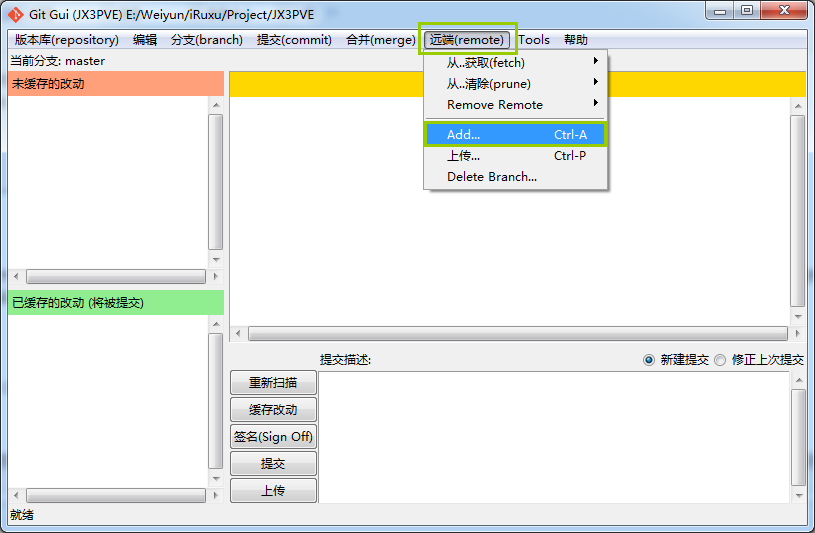
[](http://www.runoob.com/wp-content/uploads/2015/03/271241402522046.png)

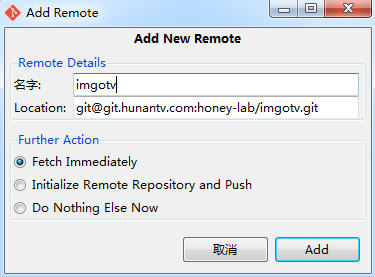
**获取远程代码**（Git remote/fetch）

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)

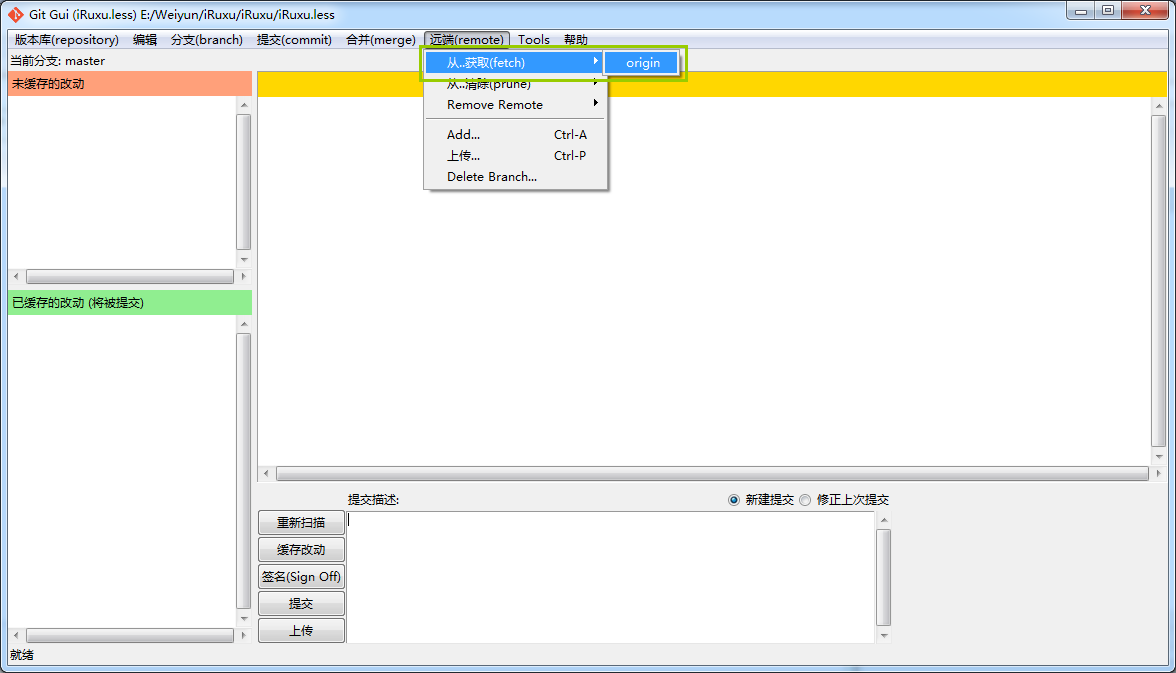
填写SSH地址与项目名。下面有3个选项：

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。

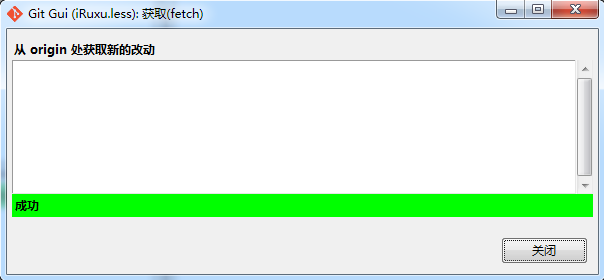
第三个：什么也不做。

**在项目的进行过程中，获取仓库的最新改动Git fetch**

[](http://www.runoob.com/wp-content/uploads/2015/03/271259025495085.png)

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。

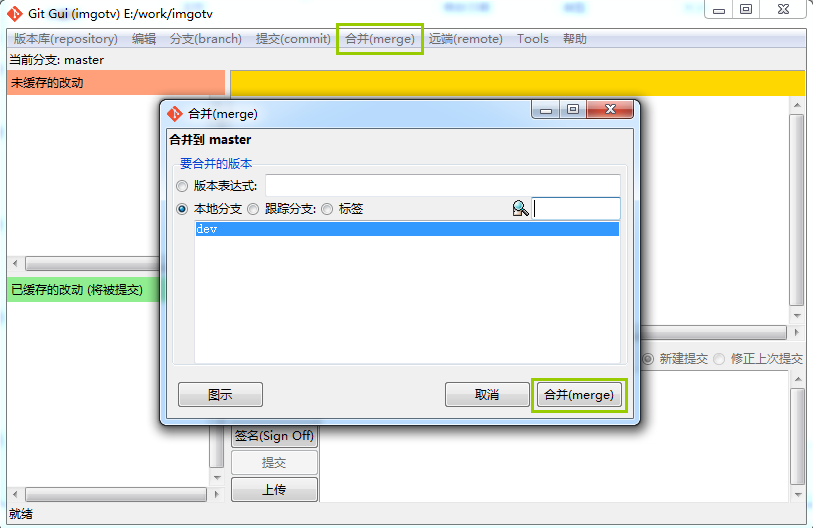
提示成功则改动的已经被存放到临时区了，你一会还需要进行合并操作，如果没有任何改动，则列表中是空的，比如：

[](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。

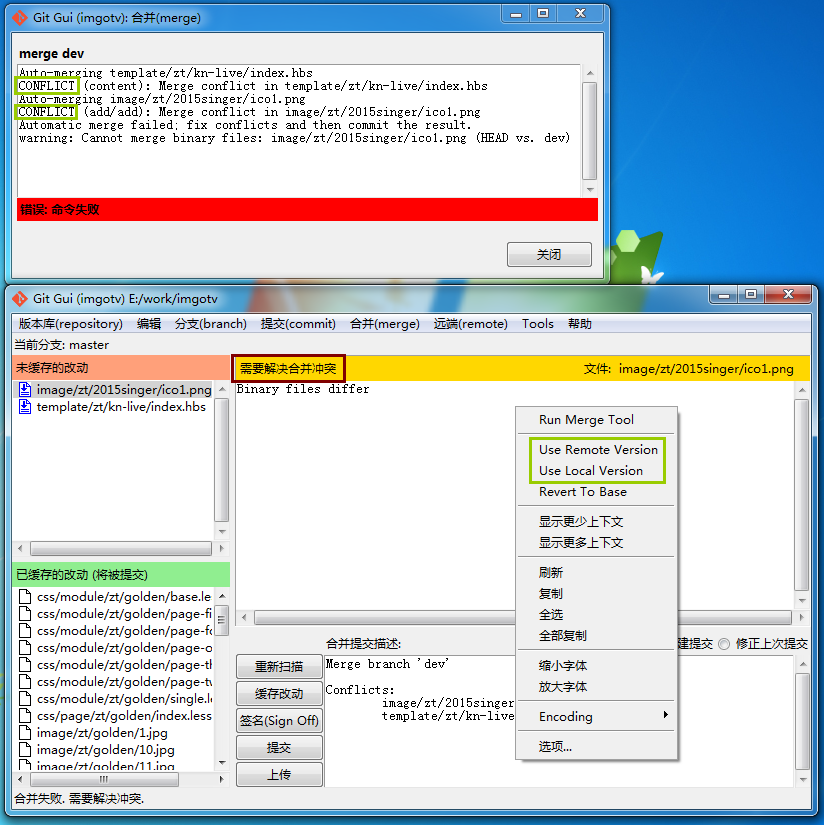
选择合并 - 本地合并，然后选择本地的分支（如果你没有创建分支，则只有1个主支master）

[](http://www.runoob.com/wp-content/uploads/2015/03/271254126278757.png)

**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。

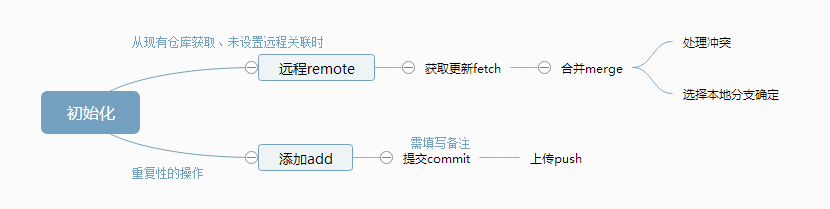
[](http://www.runoob.com/wp-content/uploads/2015/03/271257485644055.png)

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!

# 2/极简 Github 上手教程

随着你写的代码越来越多，就需要去管理这些代码，可能要在不同的电脑上编写同一份代码，可能要和别人合作开发，也可能想把项目分享给别人。更进一步地，你可能还会需要管理代码的不同历史版本。于是你就得和代码管理工具以及代码托管平台打交道。

Github 是现在广泛流行的代码托管平台，很多开源项目都把代码放在这里。之前我们有过[关于 git 使用的入门教程](https://link.zhihu.com/?target=http%3A//res.crossincode.com/wechat/git.html" \t "_blank)（从公众号底部菜单栏可进入）。今天，再来科普一下如何把代码放在 github 上。

1. 去 [http://**github.com**](https://link.zhihu.com/?target=http%3A//github.com)注册一个账号。

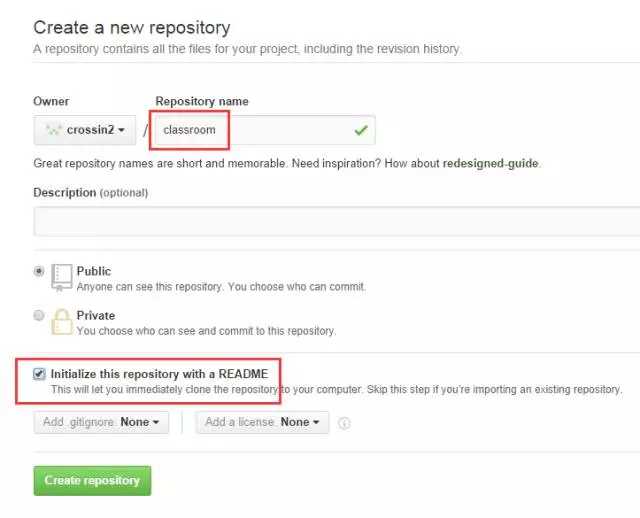


依次输入用户名、邮箱、密码，就可注册成功。需要去邮箱里验证一下，否则不能创建项目。

2. 登录后，点击 **New repository** 创建一个新项目。

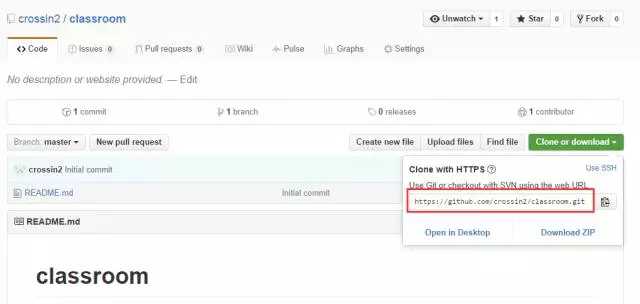


此页面上的 **Let’s get started!**是一个简要的 github 使用教程，不妨也浏览一下。



输入项目名称，然后把创建 README 勾选上，这样可以省去你初始化项目的步骤，方便之后直接在本地获取项目。

点击 **Create repository** 完成创建。

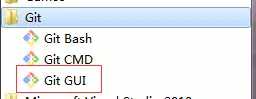


在项目页面上，从**Clone or download**里找到项目的地址，后面会用到它。

3. 安装 git 客户端，获取项目到本地。

Windows 下推荐使用**git for windows**。官网下载有可能会很慢，不行的话就直接从搜索引擎找个安装文件下载。Github 本身也提供了一个客户端，但同样从国内下载也很慢。

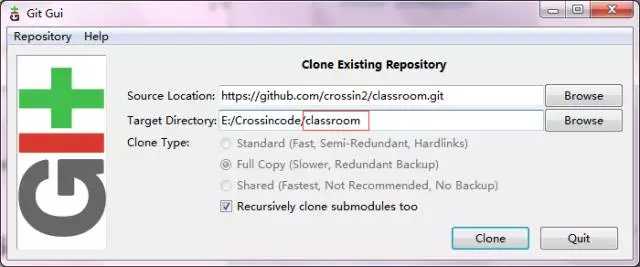
安装好 git for windows 后，可以通过 git 命令行获取项目并进行后续的更新。具体命令可参考之前的 [git 入门教程](https://link.zhihu.com/?target=http%3A//res.crossincode.com/wechat/git.html" \t "_blank)。这里我说一下图形化界面的使用。



从开始菜单中选择**Git GUI**，打开界面，选择 **Clone Existing Repository**，



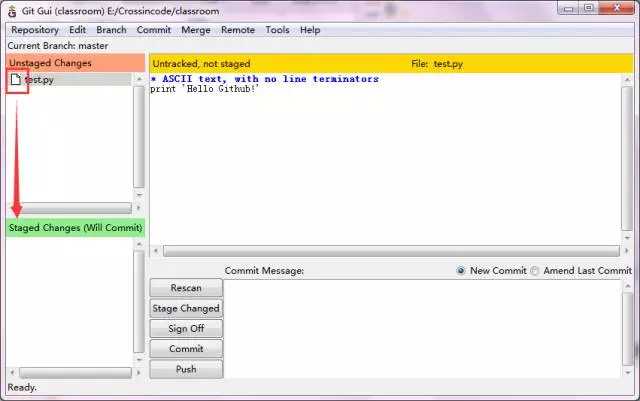
**Source Location** 输入刚才 Github 上创建的项目地址，**TargetDirectory** 设置一个本地目录。这里注意，本地目录必须是个还不存在的目录，否则无法创建成功。



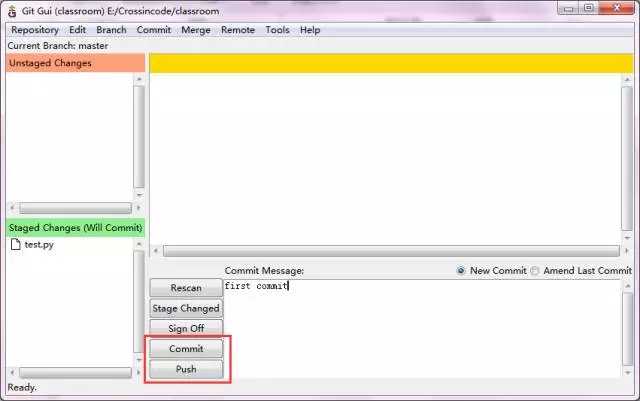
创建成功后，就会将项目代码复制到本地。

4. 修改，上传代码。

在项目目录新增一个文件，test.py。

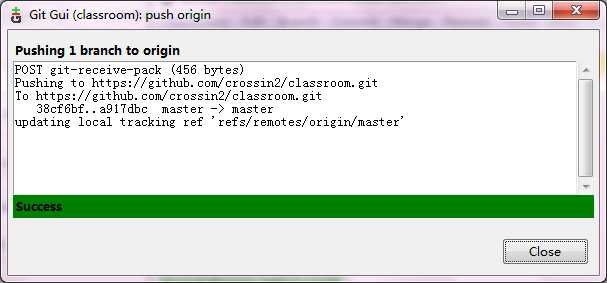


在文件夹里点击右键，选择**Git GUI Here**，打开 Git GUI 并显示当前项目。看到 **Unstaged Changes**里有新增的文件。点击文件前面的图标，将其移动至**Staged Changes**里，就可以准备提交了。

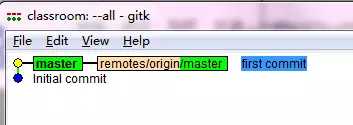


在 **Commit Message**里输入提交注释，点击 **Commit**，完成提交。

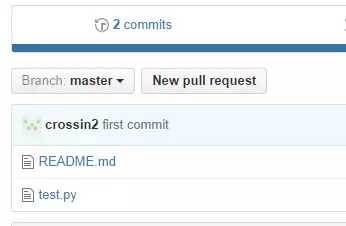
然后再点击 **Push**，就可以将提交更新至 Github 上。此时会提示你输入 Github 的用户名和密码。



完成之后，可以通过点击 Git GUI 菜单栏上的 **Repository**-> **Visualize All Branch History**查看提交历史。



此时到 Github 的项目页面上，就可以看到，test.py 已经被添加到项目中。



**附：**

关于 git 中文件的状态转换，以及 git 命令行的使用，请参考 [git 入门教程](https://link.zhihu.com/?target=http%3A//res.crossincode.com/wechat/git.html" \t "_blank)。

更详细的说明，可查阅 [pro git 教程](https://link.zhihu.com/?target=http%3A//res.crossincode.com/jump.html%3Fto%3Dhttps%3A//git-scm.com/book/zh/v2)。

另外附上一篇极简的 [git 简明指南 - no deep shit!](https://link.zhihu.com/?target=http%3A//res.crossincode.com/jump.html%3Fto%3Dhttp%3A//rogerdudler.github.io/git-guide/index.zh.html" \t "_blank)，里面介绍了最基本的一些概念和命令行命令。

其他文章及回答：

[学习编程的过程中可能会走哪些弯路，有哪些经验可以参考？ - Crossin 的回答](https://www.zhihu.com/question/22863981/answer/22923499)

[你是如何自学 Python 的？ - Crossin 的回答](https://www.zhihu.com/question/20702054/answer/19022301)

[编程初学者如何使用搜索引擎 - Crossin的文章 - 知乎专栏](https://zhuanlan.zhihu.com/p/20683456?refer=crossin)

[如何直观地理解程序的运行过程？- Crossin的文章 - 知乎专栏](https://zhuanlan.zhihu.com/p/20721034?refer=crossin)

[如何在 Python 中使用断点调试 - Crossin的文章 - 知乎专栏](https://zhuanlan.zhihu.com/p/21304838)

[如何在一台电脑上同时使用 Python 2 和 Python 3 - Crossin的编程教室 - 知乎专栏](https://zhuanlan.zhihu.com/p/20832723)

[Python 抓取网页乱码原因分析 - Crossin的编程教室 - 知乎专栏](https://zhuanlan.zhihu.com/p/21057822)

**Crossin的编程教室**

微信ID：crossincode

论坛：[Crossin的编程教室](https://link.zhihu.com/?target=http%3A//bbs.crossincode.com/)

QQ群：521603553

# 3/win下GIT GUI的使用教程

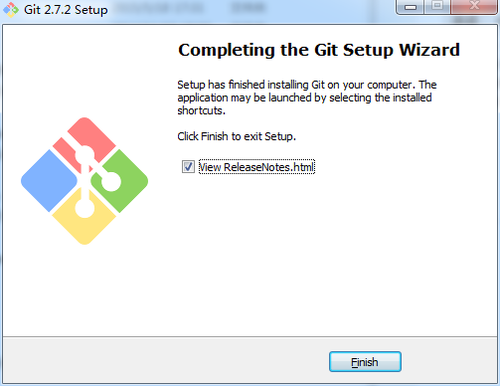
2016-07-18 14:36 本站整理 浏览(34)

现在很多都有git来托管项目或者来查找资料，但是看起来操作不是很方便，现在由于win下可以直接使用git gui，让使用git变得方便，当然这只是针对日常简单的使用，如果想详细的使用，可以去参考廖学峰写的教程

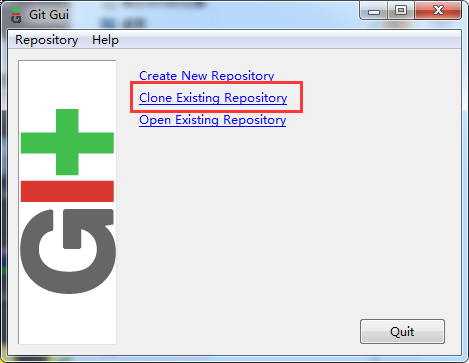
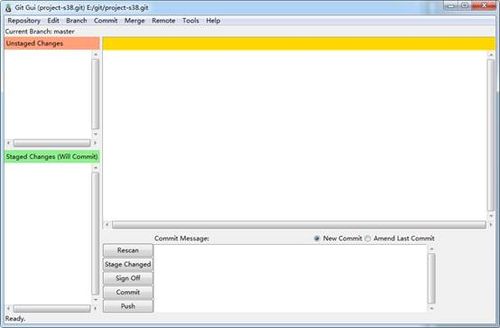
### git下载地址

<http://pan.baidu.com/s/1b8mC3s>

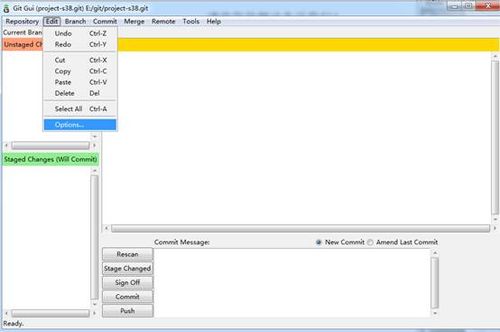
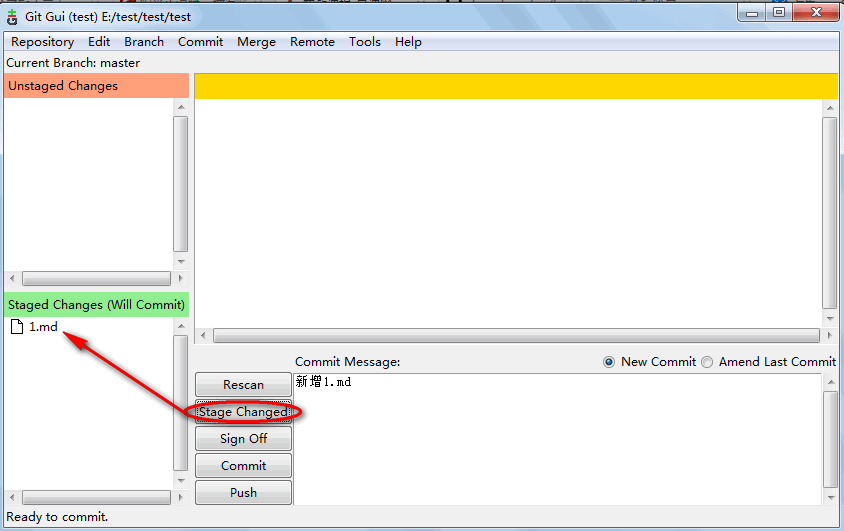
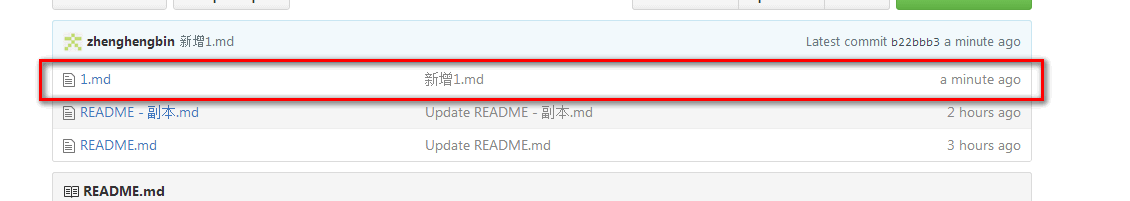
### git安装步骤

1.选择下载的安装包  
http://szimg.mukewang.com/57046fcb0001035e02690027.jpg  
2.安装步骤省略，直接下一步即可。   
出现下面的图片安装成功   


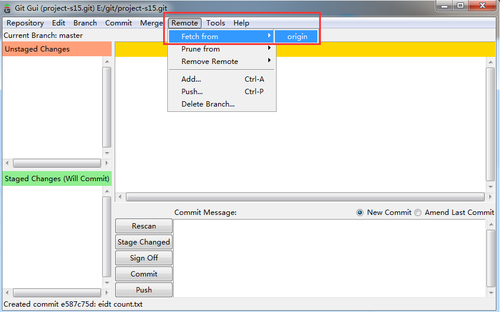
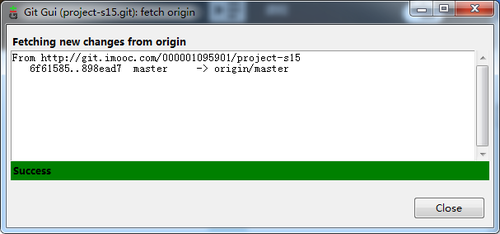
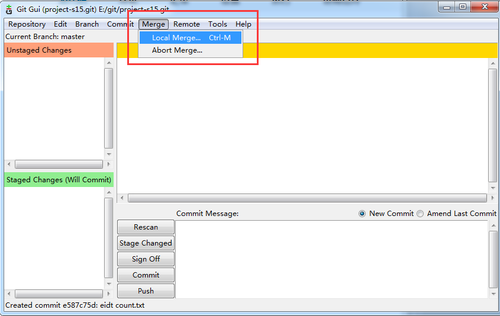
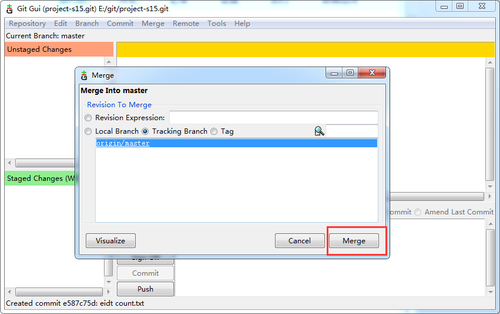
### 使用git GUI克隆已存在的仓库

1.在本地新建一个文件夹test，选中后点击右键，选择Git GUI Here   
出现下图,选择第二个，克隆已经存在的仓库   
  
2.填写Source Location（源地址）和Target Directory   
  
3.完成克隆   
点击上图的clone后，会提示下输入你github的用户名和密码，出现下面的界面，克隆完成。你可以看看你本地克隆的文件。   


### 使用git GUI进行Push操作

1.修改属性配置-改为utf-8   
  
  
2.获取改动文件之后，选择Stage Changed.我在本地添加了一个1.md文件   
  
3.选择Stage Changed，在新弹窗中选择是   
  
4.然后点击 commit，再点击push，输入用户名和密码，就提交到远程仓库了。下图是远程仓库,就看到刚刚提交的内容   


### Remote和Merge操作

1.服务端重置项目后，把本地文件直接push，或修改本地文件，待扫描到有文件改动时在提交 同样也会出错，这是由于git本地与服务器端文件不一致产生冲突导致的。   
  
2.遇到这种情况需要先从远程仓库拉取一下文件，解决下合并冲突的问题再push  
1.获取仓库的最新改动Git fetch。选择Remote->Fetch from-origin，同样需要输入项目的用户名和密码   
  
成功后显示   
  
2.解决一下文件冲突问题   
菜单->Merge->Local Merge,在Merge弹窗中选择Merge按钮，成功后会有Success弹窗出现。   
  
  
3.如果合并之后检测到文件有冲突，是会提示Command Failed失败的，并且Git Gui是会获取到冲突文件，对于冲突文件的解决，可以右键选择使用远程版本还是本地版本，并且本地已经被打开的话，是会提示是否overwritten.按照此方法挨个解决下冲突文件即可。