使用教程<https://www.zhihu.com/question/20070065/answer/117017972>

是已有应用程序的使用

（ account wudeape

Address：[1497595896@qq.com](mailto:1497595896@qq.com)

Password：Cwmn56163596

）

<https://www.zhihu.com/question/20070065>

关于客户端http访问时github

的使用<http://www.cnblogs.com/liuleilei/p/6486665.html#3752102>（初学者强力推荐）

（只是对github使用的简介 但对于Git的配置和具体使用操作详见下文）

<http://doc.orz520.com/a/doc/2013/1030/2021235.html?from=haosou>

<http://www.jb51.net/article/118229.htm>（）

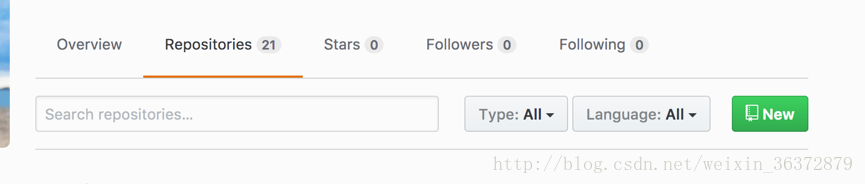
master分支就是整个项目的根节点。

每次代码写完毕可以点击提交到github

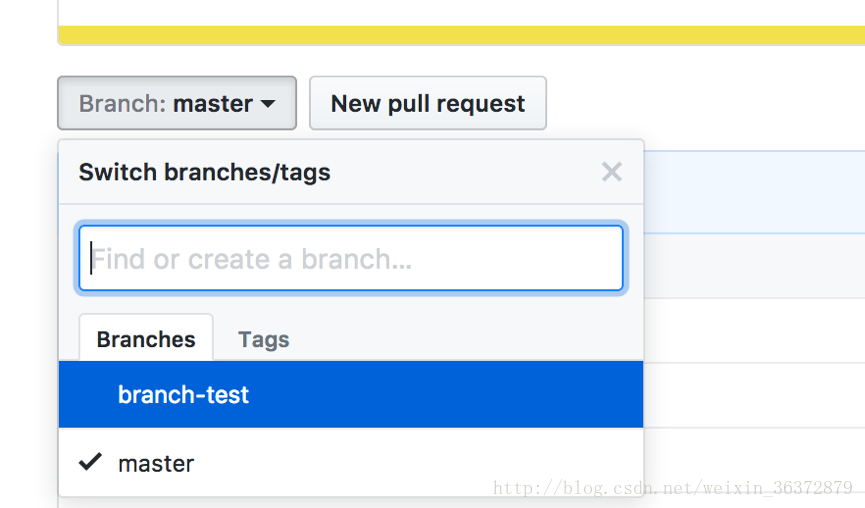
Git bash 的使用<http://blog.sina.com.cn/s/blog_4d9c3fec0102w7g9.html>

上传本地的代码或者是文件

1. 创建仓库

仓库存储着包括修订历史记录在内的所有项目文件。仓库可以是公开的或者是私有的。公开的仓库可以被任何人查看，但是，只有项目所有者授予权限的人才可以提交修改到这个仓库。另一方面，私有仓库提供了额外的控制，可以将项目设置为对谁可见。因此，公开仓库适用于开源软件项目，而私有仓库主要适用于私有或闭源项目。   


1. 添加分支   
   分支不是GIT的专利，早在CVS的时候就有了。分支对于团队开发的作用来说太大了。比如说现在这个项目是1.0版本，那么开发团队可能会同时进行1.0和2.0版本的开发，这样代码就会出现比较大的分歧，这个时候，就需要用到分支了，不同的任务在不同的分支上开发，互相之间不会影响。

一般情况下，如果要开发一个新的功能，一般团队都不会在主分支master上面修改，都会建立一个新的分支，在上面更改代码。这样做的好处就是保证了主线代码的完整性以及可用性，也就是说，主线上面的代码都是稳定的，可以用来直接发布的。建立分支刚开始只是对master分支里面所有文件进行拷贝，如果要修改，就成为了不同的分支。   
   
GIT的技术可以说是非常强大，说一切开发都依赖于分支一点都不过分。一般项目的master分支都是稳定的代码，可以直接发布或者被项目之外的人使用。新特性和BUG修改都在不同的分支上进行开发和测试。这样规范了整个软件的开发流程。分支之间的互不影响这种特性可以增加团队合作的效率。

GIT分支的另外一个重要特性就是可以合并不同软件仓库（fork）的分支。大体就是：开发团队有一个主软件仓库，开发者可以fork这个仓库，相当于克隆了一个私有的软件仓库，然后开发者就可以在自己的软件仓库中建立分支并开发测试，测试完毕之后，可以向主软件仓库提交merge request，这样新开发的特性就会被合并到主软件仓库的某个分支中。这种特性极大地推动了开源项目，多年前参加开源项目都是神话，自从GIT出现后，每个人都可能成为开源项目的贡献者，就是基于这种特性。

GitHub可以托管各种git库，并提供一个web界面，但与其它像 SourceForge或Google Code这样的服务不同，GitHub的独特卖点在于从另外一个项目进行分支的简易性。为一个项目贡献代码非常简单：首先点击项目站点的“fork”的按钮，然后将代码检出并将修改加入到刚才分出的代码库中，最后通过内建的“pull request”机制向项目负责人申请代码合并。已经有人将GitHub称为代码玩家的MySpace

1. 修改项目并提交

修改和提高是非常常规的操作了，在 GitHub 上，提交commit被定义为保存的修改的意思。每一次提交都与一个提交信息commit message相关联，该提交信息包含了保存的修改的历史记录，以及为何进行这些更改。这使得其他贡献者可以很轻松地知道你做出的更改以及更改的原因，点击“commit to master”就会将修改的代码提交到主分支上面，然后点击fetch orgin之后，就会将你提交的修改同步到github云端库里面。当然也可以使用undo来撤销对刚刚版本的提交。提交了之后，可以在history里面找到提交版本的具体修改与详细的时间等具体信息。

1. 拉取与合并请求

打一个比方：你做的试卷就像仓库repository,试卷里面肯定会有很多的   
错误，就相当于程序里面的bug，老师把你的试卷拿过来，相当于先fork。（fork的理解就是相当于一个代码拷贝，当你fork了别人的repository之后，，项目就会出现在自己的github仓库下面了。）在你的试卷上面进行批注，批注完毕之后，相当于git mommit。最后再把试卷给你，相当于往你的库里面加东西，就发起一个pull request，你同意之后拿到试卷，重新改正错误，就相当于merge request。

当你想要更改别人的仓库的时候，就需要用到pull request了，一般需要走一个流程。   
· 先fork想要修改的仓库，相当于拷贝一份到自己的git账号下面。   
· 然后clone到本地，做一些debug   
· debug完毕之后，发起pull request给原仓库   
· 原仓库review你的修改，如果接受的话，就通过merge request来合并到自己的仓库下面。   
到此为止，整个pull request merge request就结束了。

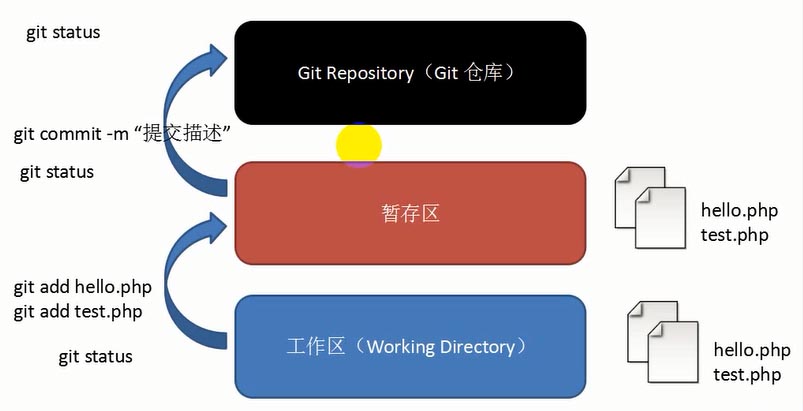
Git的安装及配置

**二、Git基本工作流程**

**1、Git工作区域**

[](http://files.jb51.net/file_images/article/201707/2017070918242822.jpg)

**2、向仓库中添加文件流程**

[](http://files.jb51.net/file_images/article/201707/2017070918242823.jpg)

**三、Git初始化及仓库创建和操作**

**1、Git安装之后需要进行一些基本信息设置**

　　a、设置用户名：git config -- global user.name '你再github上注册的用户名';

　　b、设置用户邮箱：git config -- global user.email '注册时候的邮箱';

注意：该配置会在github主页上显示谁提交了该文件

　　c、配置ok之后，我们用如下命令来看看是否配置成功

　　git config --list

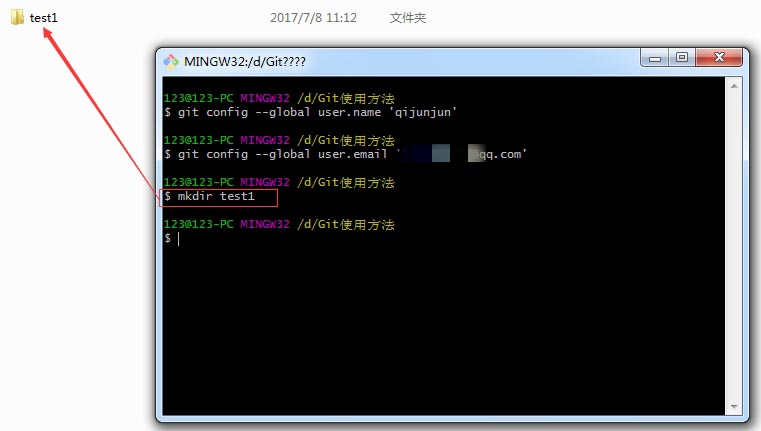
注意：git config --global 参数，有了这个参数表示你这台机器上所有的git仓库都会使用这个配置，当然你也可以对某个仓库指定不同的用户名和邮箱

**2、初始化一个新的git仓库**

**a、创建文件夹**

　　　　方法一：可以鼠标右击-》点击新建文件夹test1

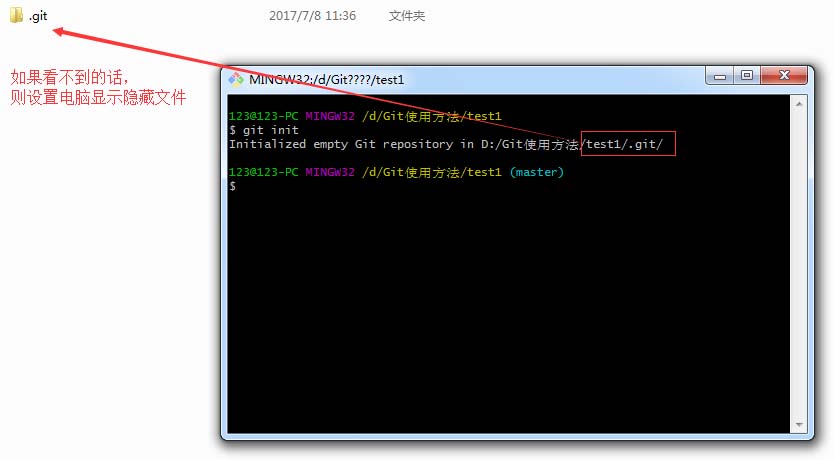
　　　　方法二：使用git新建：$ mkdir test1

[](http://files.jb51.net/file_images/article/201707/2017070918242824.jpg)

**b、在文件内初始化git（创建git仓库）**

　　　　方法一：直接输入 $ cd test1

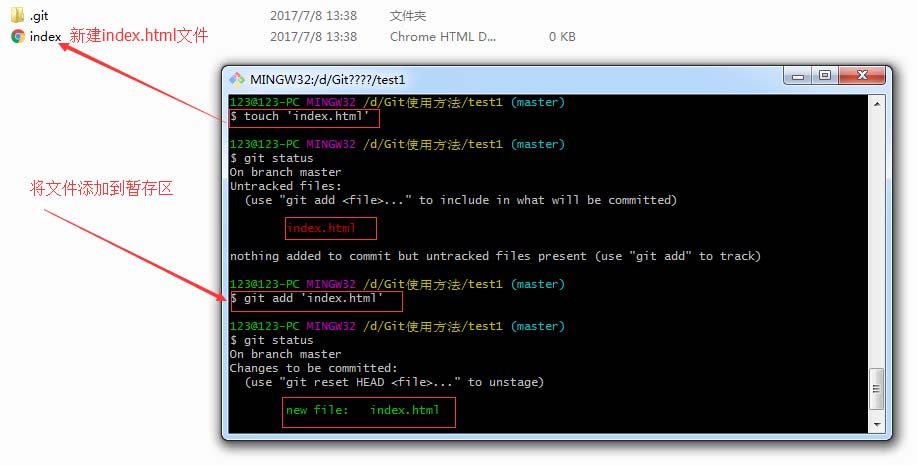
　　　　方法一：点击test1文件下进去之后-》鼠标右击选择Git Bash Here->输入$ git int

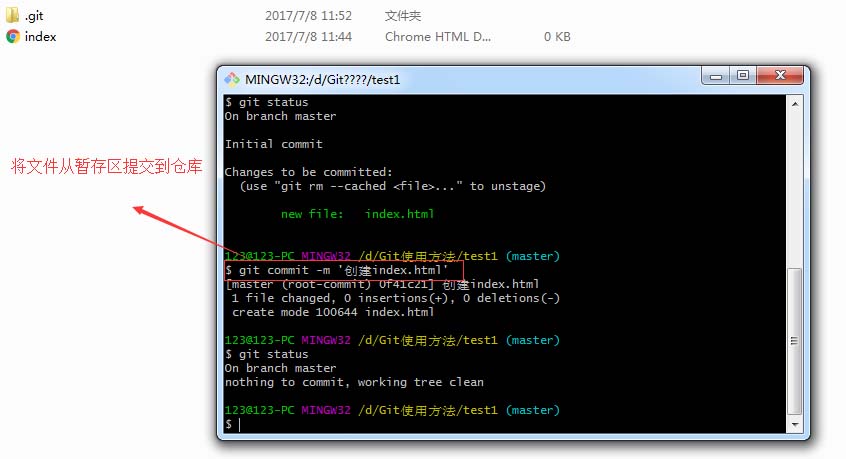
[](http://files.jb51.net/file_images/article/201707/2017070918242825.jpg)

**3、向仓库中添加文件**

**方法一**：**用打开编辑器新建index.html文件**

**方法二：使用git命令。$ touch '文件名'，然后把文件通过$ git add '文件名'添加到暂存区，最后提交操作**

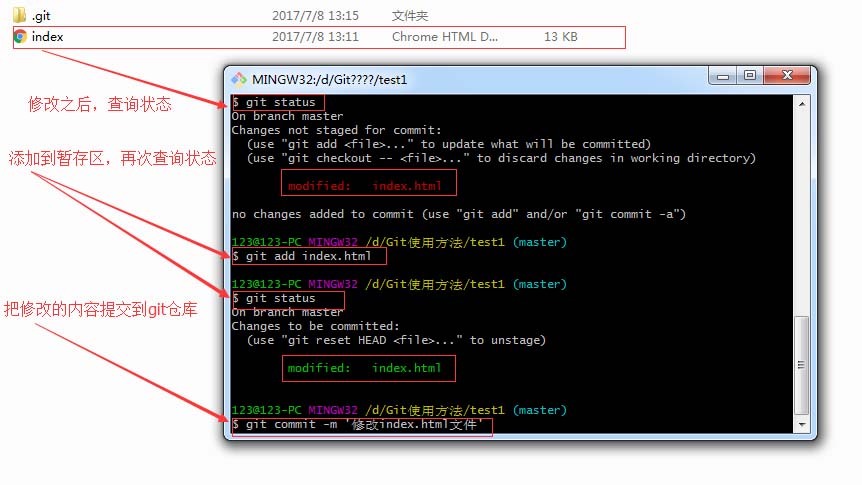
[](http://files.jb51.net/file_images/article/201707/2017070918242826.jpg)

[](http://files.jb51.net/file_images/article/201707/2017070918242827.jpg)

**4、修改仓库文件**

**方法一**：**用编辑器打开index.html进行修改**

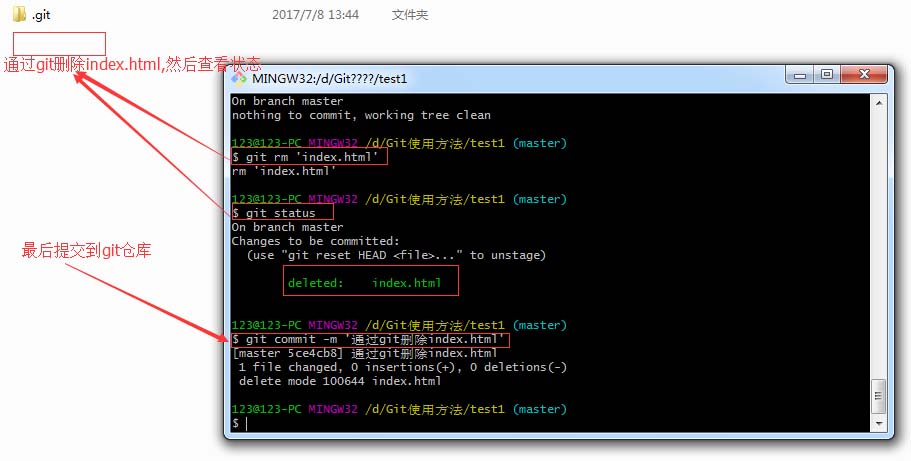
**方法二：使用git命令。$ vi '文件名'，然后在中间写内容，最后提交操作**

[](http://files.jb51.net/file_images/article/201707/2017070918242828.jpg)

**5、删除仓库文件**

**方法一：在编辑器中直接把要删除的文件删除掉**

**方法二：使用git删除：$ git rm '文件名'，然后提交操作**

[](http://files.jb51.net/file_images/article/201707/2017070918242829.jpg)

**四、Git管理远程仓库**

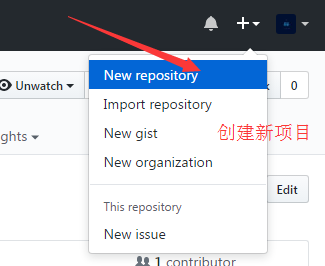
# git上传本地项目至github

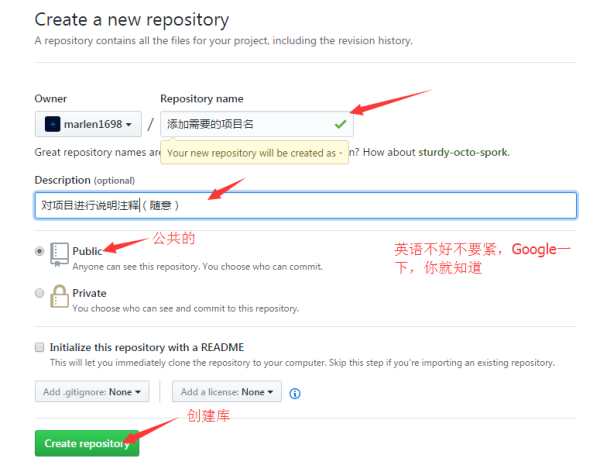
[marlen](https://www.zhihu.com/people/marlen-19)[marlen](https://www.zhihu.com/people/marlen-19)

4 个月前

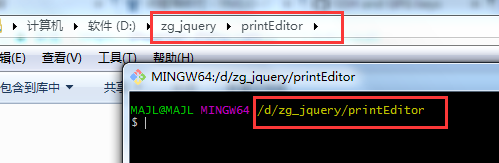
最近学习git上传地项目至github，屡试不爽，今天进行总结，以便今后使用，过程如下：

1，打开github创建New repository



2，创建项目名，项目说明，项目的公开性（public），创建项目库

3，在本地项目的文件，右键 选择git bash Here，即直接指向项目文件夹



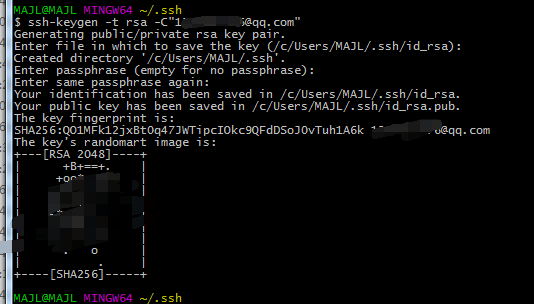
4， 如果已存在，c盘用户去删除.ssh文件，或许指令删除；

终端里输入指令：mkdir .ssh 创建一个 .ssh 文件夹

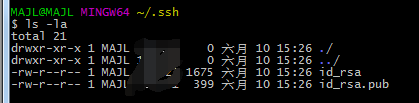
cd .ssh 进入刚创建的 .ssh文件夹目录里

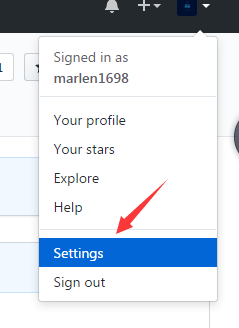
输入指令：ssh-Keygen -t rsa -C “123456[@qq](http://www.zhihu.com/people/65b410e4eb98f327b2fd256197abf915).com ”注册时的邮箱如23456[@qq](http://www.zhihu.com/people/65b410e4eb98f327b2fd256197abf915).com

输入完成之后一直按回车键 中间会提示你要输入密码，不用管一直按回车直到出现这样。

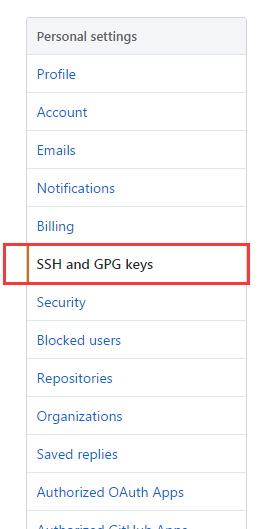


c盘 用户查看即可

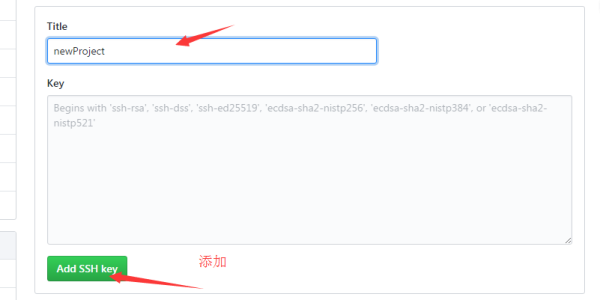
5，输入ls -la验证配置是否成功，以下信息即success

6，打开设置settings

7，选择 SSH and GPG keys



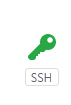
8，创建新的SSH密钥https://pic1.zhimg.com/v2-1eb6cbf5e752d316ef424d3ef42b65a8_b.png



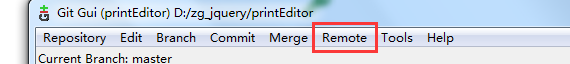
9，打开.ssh文件中**/.ssh/id\_rsa.pub copy并粘贴至key中 保存，钥匙为灰色，**

》》输入ssh -T git@github.com指令 回车

https://pic1.zhimg.com/v2-88ed51a442cb0214461639a15a5f01e4_b.png》》键入yes，提示succfully

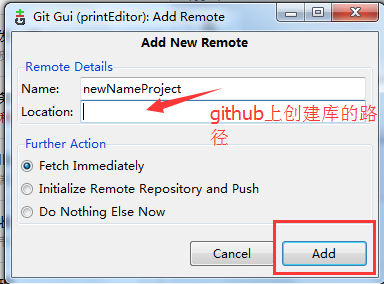
https://pic1.zhimg.com/v2-cf87d8a352f38fcb421ed181bc9ded68_b.png》》刷新网页密钥摇身一变

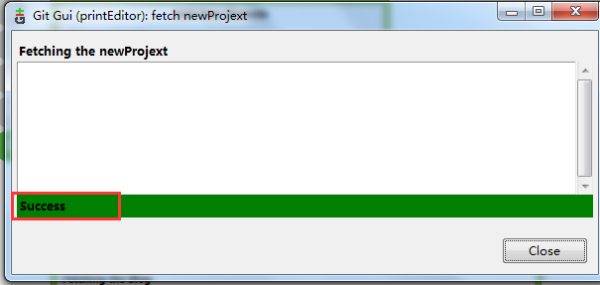
**10，在本地文件右键打开 git GUI here**



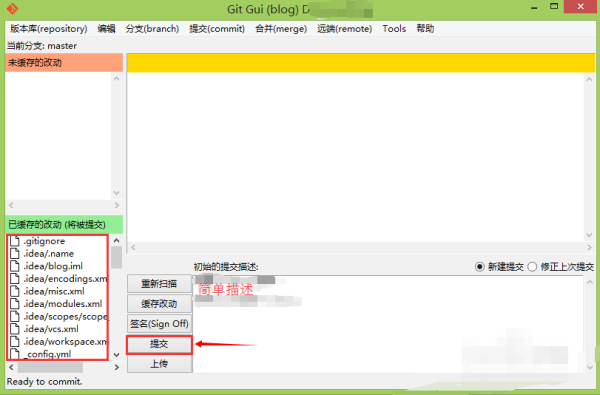
在Add Remote窗口中填写名字、Location。名字：newNameProject

Location：粘贴刚刚复制的Blog仓库路径；最后单击“Add”



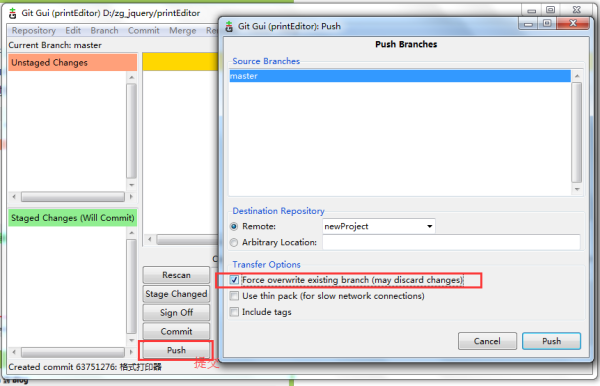
  
等success之后；

将本地文件内容即“已缓存的改动”，提交git管理的master分支上。



将本地文件上传到github上去

单击Git Gui界面“上传”--“上传”窗口随意勾选一个或多个传输选项--点击“上传”--上传Blog过程中需要输入github的登录账号和密码。

[http://jingyan.baidu.com/album/27fa732683ebf546f8271f2e.html?picindex=7](http://link.zhihu.com/?target=http%3A//jingyan.baidu.com/album/27fa732683ebf546f8271f2e.html%3Fpicindex%3D7)

提交看到success即可

Git的命令简集

1,