### 一、在GitHub新建托管项目

#### 1、注册

1. 1

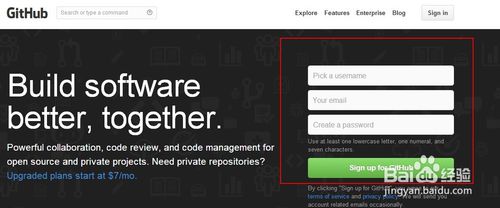
要托管到github，那你就应该要有一个属于你自己的github帐号，所以你应该先到github.com注册

打开浏览器

在地址栏输入地址：github.com

填写用户名、邮箱、密码

点击Sign up即可简单地注册



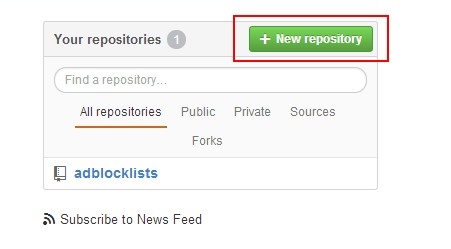
1. 2

完成注册，进入github平台，

#### 2、新建项目

点击new repositories

新建一个新项目（你也可以加入到一个已有的项目）



1. 3

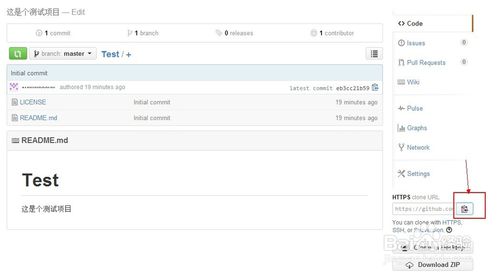
新建项目的操作，其实很简单，输入项目名就可以直接Create了

如图



1. 4

此时的界面，我想大家都应该很熟悉了。点击右边栏的剪切板图标，记录下你的项目地址。



### 二、上传项目至GitHub

#### 1、安装

https://git-for-windows.github.io/

下载并安装最新版本的msysgit

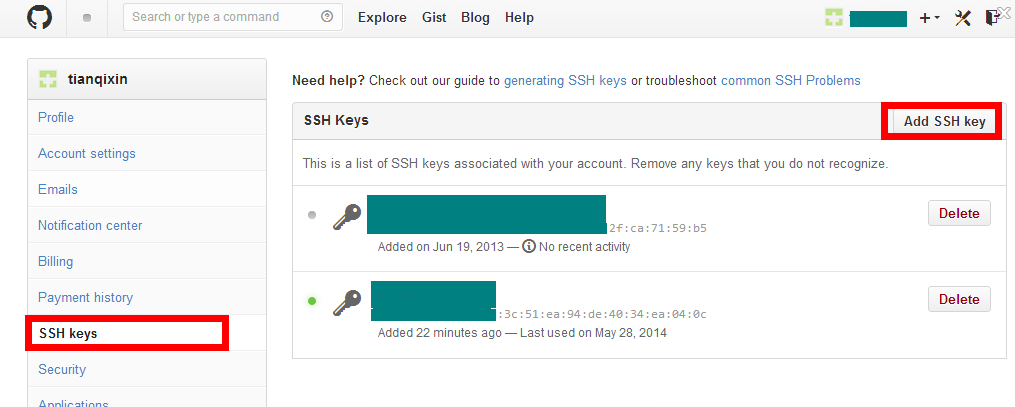
#### 2、设置用户

首先在本地创建ssh key；

$ ssh-keygen -t rsa -C "your\_email@youremail.com"

后面的your\_email@youremail.com改为你在github上注册的邮箱，之后会要求确认路径和输入密码，我们这使用默认的一路回车就行。成功的话会在~/下生成.ssh文件夹，进去，打开id\_rsa.pub，复制里面的key。

回到github上，进入 Account Settings（账户配置），左边选择SSH Keys，Add SSH Key,title随便填，粘贴在你电脑上生成的key。



为了验证是否成功，在git bash下输入：

$ ssh -T git@github.com

如果是第一次的会提示是否continue，输入yes就会看到：You've successfully authenticated, but GitHub does not provide shell access 。这就表示已成功连上github。

接下来我们要做的就是把本地仓库传到github上去，在此之前还需要设置username和email，因为github每次commit都会记录他们。

$ git config --global user.name "your name"

$ git config --global user.email "your\_email@youremail.com"

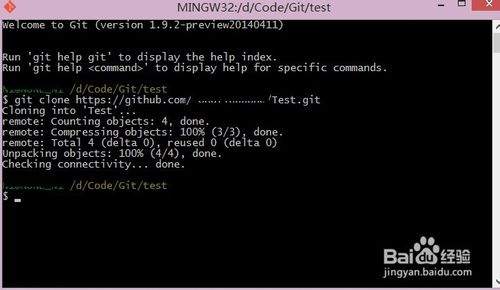
#### 3、下载仓库

启动Git Bash命令行

如下图所示



1. 输入git  clone  接着将先前记录下来的地址复制到后面，回车



1. 将下载下来的项目文件夹的所有文件及文件夹，包括.git文件夹在内，全部拷贝到你的托管项目的根目录（或者将你的托管项目拷贝到该目录）

cd test

进入到该托管项目的根目录。



1. 输入

git add .

将改动的地方添加到版本管理器

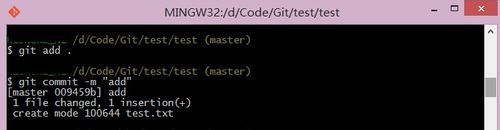


1. 输入

git  commit -m "changes log"

提交到本地的版本控制库里，

引号里面是你对本次提交的说明信息。

[](http://jingyan.baidu.com/album/f7ff0bfc7181492e27bb1360.html?picindex=9)

1. 最后输入

git push -u origin master

将你本地的仓库提交到你的github账号里，

此时会要求你输入你的github的账号和密码。

[](http://jingyan.baidu.com/album/f7ff0bfc7181492e27bb1360.html?picindex=10)

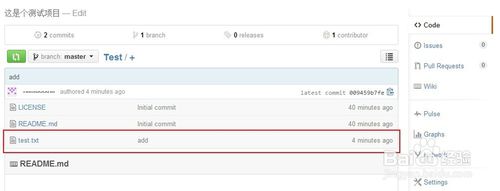
1. 回到你的github.com中的项目页。

里面的文件是不是已经发生改变了？

至此，

github项目已经成功更新。

更多的github的操作，你可以自己慢慢的体会。但有了这第一步认识，我相信你以后的摸索会更加容易得多。

[](http://jingyan.baidu.com/album/f7ff0bfc7181492e27bb1360.html?picindex=11)

END

### 注意事项

* 当你首次使用git的时候，会要求你输入你的用户信息，和你github的一致即可
* 将github中的项目同步更新至你本地目录你需要用到git fetch，这些都是以后进阶的事了
* 在下载或者上传的过程中，建议你不要断开你的网络，要耐心地等待，以免出现些不良的麻烦事
* 具体的git指令，可自行查看帮助（help）

### 分支

1. 下载新代码：git pull。

2. 从master出捡出一个新的分支：git checkout -b feature。feature是分支名称，你可以随意取名，但请用英文。

3. 在feature分支上随意修改，改完后你可以提交你的修改：git commit -m "some message"。

4. 此时要同步代码，请先切回主分支：git checkout master，然后更新git pull。

5. 如果想删除自己建立的分支，使用git branch -D feature，注意执行此命令后分支被强制删除，无法恢复。

### 创建一个版本库

创建一个版本库非常简单，

首先，选择一个合适的地方，创建一个空目录：

$ mkdir learngit

$ cd learngit

$ pwd

/Users/michael/learngit

如果你使用Windows系统，为了避免遇到各种莫名其妙的问题，请确保目录名（包括父目录）不包含中文。

第二步，通过git init命令把这个目录变成Git可以管理的仓库：

$ git init

Initialized empty Git repository **in** /Users/michael/learngit/.git/

瞬间Git就把仓库建好了，而且告诉你是一个空的仓库（empty Git repository），细心的读者可以发现当前目录下多了一个.git的目录，这个目录是Git来跟踪管理版本库的，没事千万不要手动修改这个目录里面的文件，不然改乱了，就把Git仓库给破坏了。

如果你没有看到.git目录，那是因为这个目录默认是隐藏的，用ls -ah命令就可以看见。

### 文件添加到版本库

现在我们编写一个readme.txt文件，内容如下：

Git **is** a version control system.

Git **is** free software.

一定要放到learngit目录下（子目录也行），因为这是一个Git仓库，放到其他地方Git再厉害也找不到这个文件。

和把大象放到冰箱需要3步相比，把一个文件放到Git仓库只需要两步。

第一步，用命令git add告诉Git，把文件添加到仓库：

$ git add readme.txt

执行上面的命令，没有任何显示，这就对了，Unix的哲学是“没有消息就是好消息”，说明添加成功。

第二步，用命令git commit告诉Git，把文件提交到仓库：

$ git **commit** -m "wrote a readme file"

[master (root-**commit**) cb926e7] wrote a readme file

1 file changed, 2 insertions(+)

**create** mode 100644 readme.txt

简单解释一下git commit命令，-m后面输入的是本次提交的说明，可以输入任意内容，当然最好是有意义的，这样你就能从历史记录里方便地找到改动记录。

嫌麻烦不想输入-m "xxx"行不行？确实有办法可以这么干，但是强烈不建议你这么干，因为输入说明对自己对别人阅读都很重要。实在不想输入说明的童鞋请自行Google，我不告诉你这个参数。

git commit命令执行成功后会告诉你，1个文件被改动（我们新添加的readme.txt文件），插入了两行内容（readme.txt有两行内容）。

为什么Git添加文件需要

$ git add file2.txt file3.txt

$ git commit -m "add 3 files."