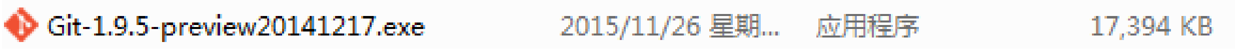
一：Git是什么？

Git是目前世界上最先进的分布式版本控制系统。

二：SVN与Git的最主要的区别?

SVN的存储需要依赖一个服务器，而git所有的东西是放在线上的。节约成本，省时省力。

三：在windows上如何安装Git?

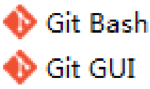


Git-1.9.5-preview20141217.exe	2015/11/26 星期...	应用程序	17,394 KB

Git-1.9.5-preview20141217.exe	2015/11/26 星期...	应用程序	17,394 KB

下一步 ---（next）

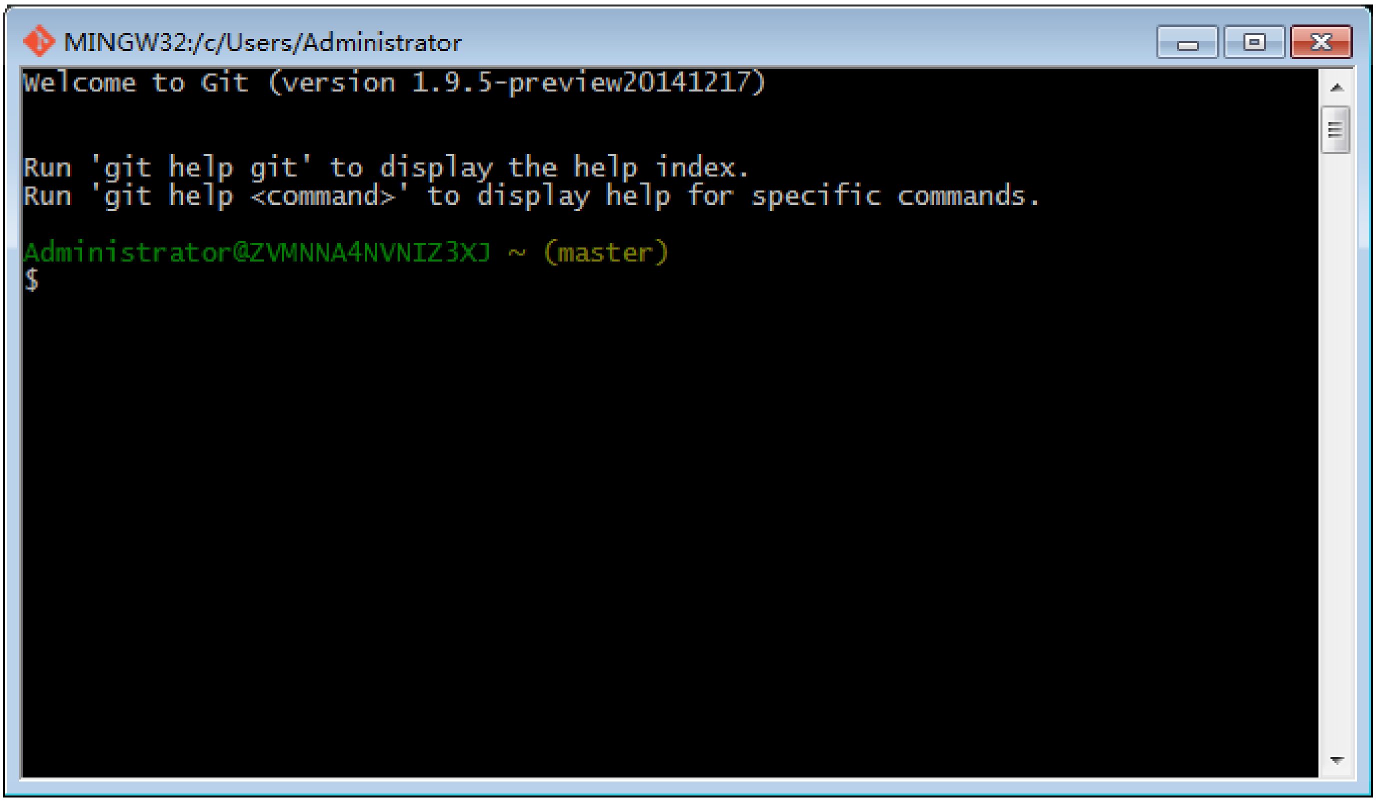
安装成功后会出现这两个东西



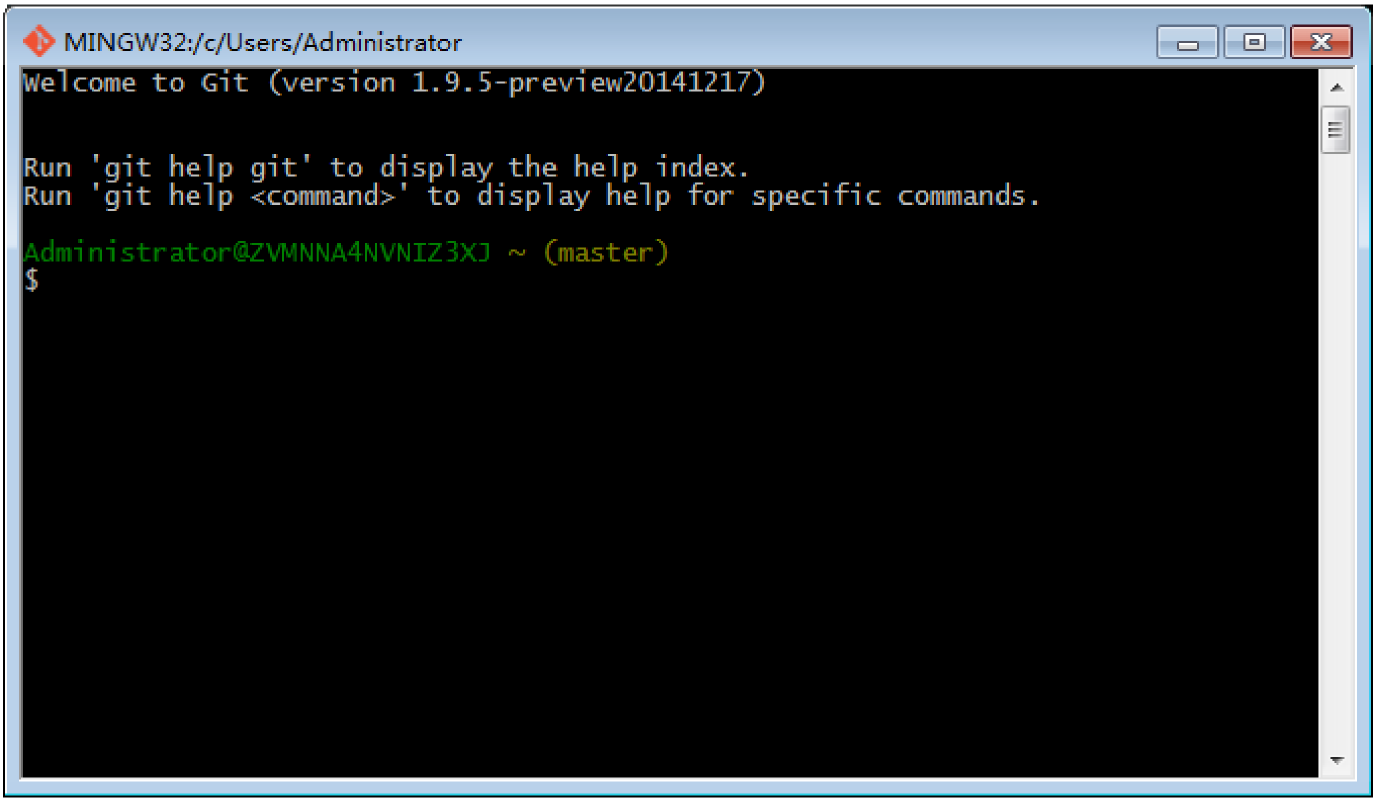
Git Bash
Git GUI

Git Bash
Git GUI

运行Git Bash会出现如下结果:



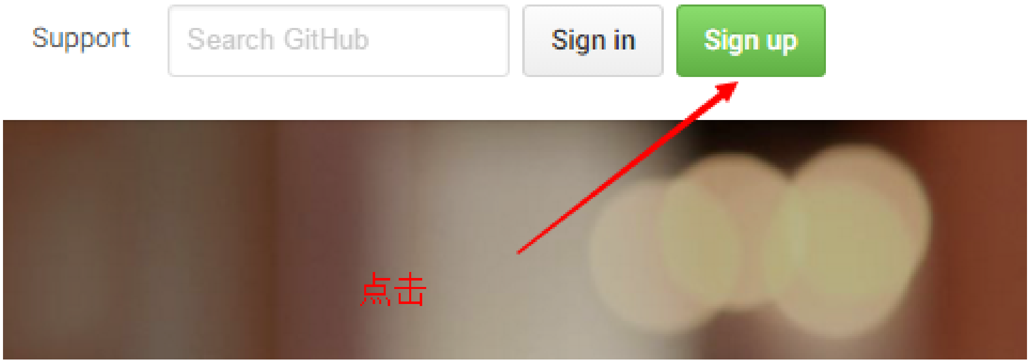
```
Welcome to Git (version 1.9.5-preview20141217)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Administrator@ZVMNNA4NVNIZ3XJ ~ (master)
$
```



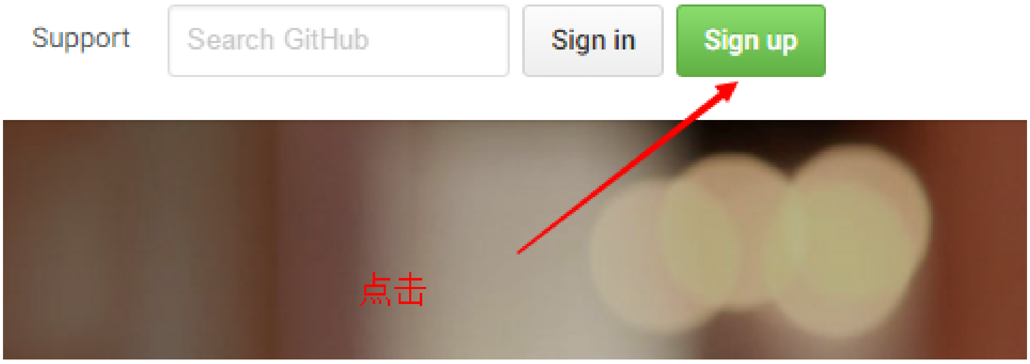
```
Welcome to Git (version 1.9.5-preview20141217)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Administrator@ZVMNNA4NVNIZ3XJ ~ (master)
$
```

四. GitHub注册

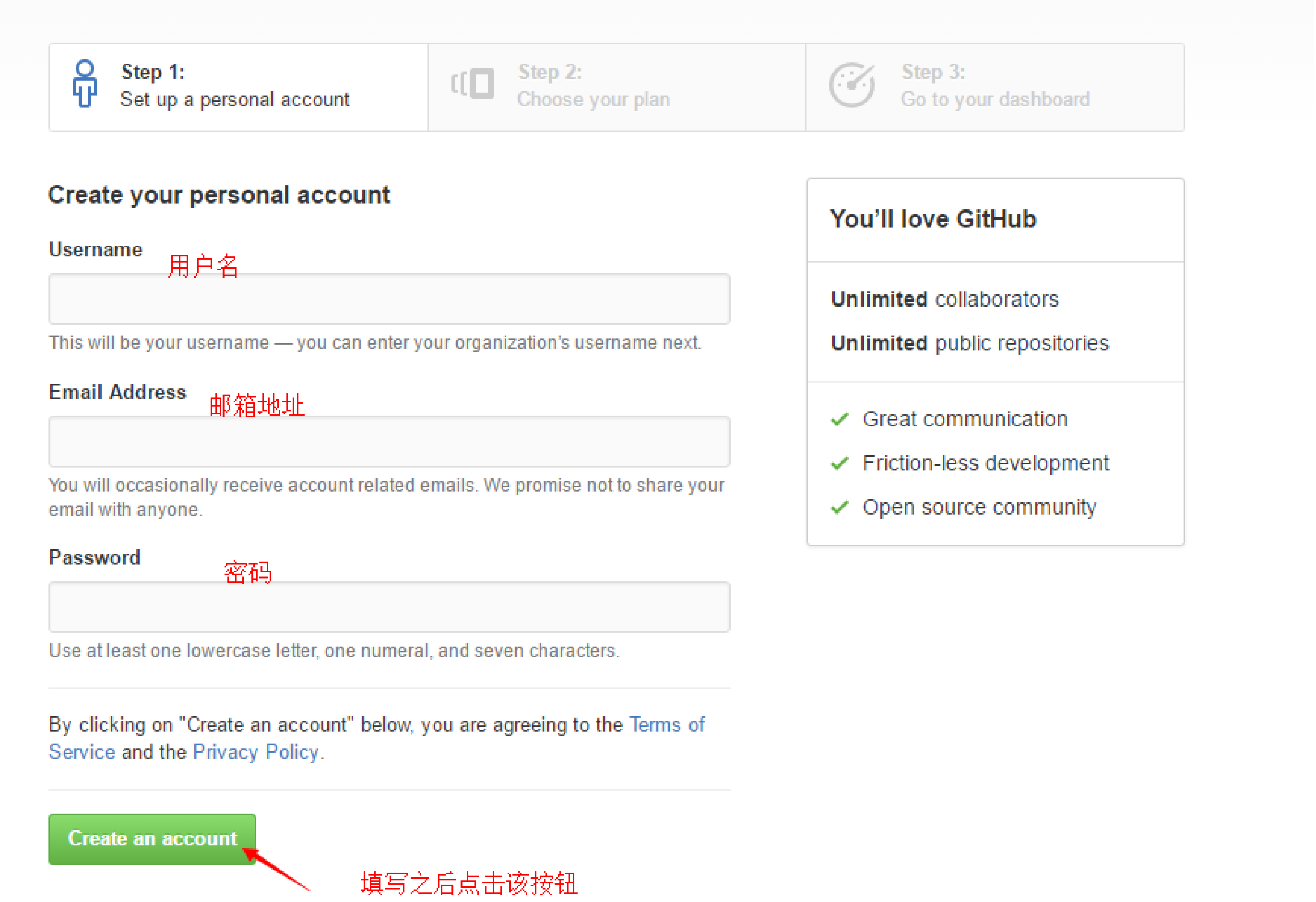
https://github.com/ 打开GitHub官网 。



点击



点击



Create your personal account

Username 用户名

This will be your username — you can enter your organization's username next.

Email Address 邮箱地址

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password 密码

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the Terms of Service and the Privacy Policy.

填写之后点击该按钮

You'll love GitHub

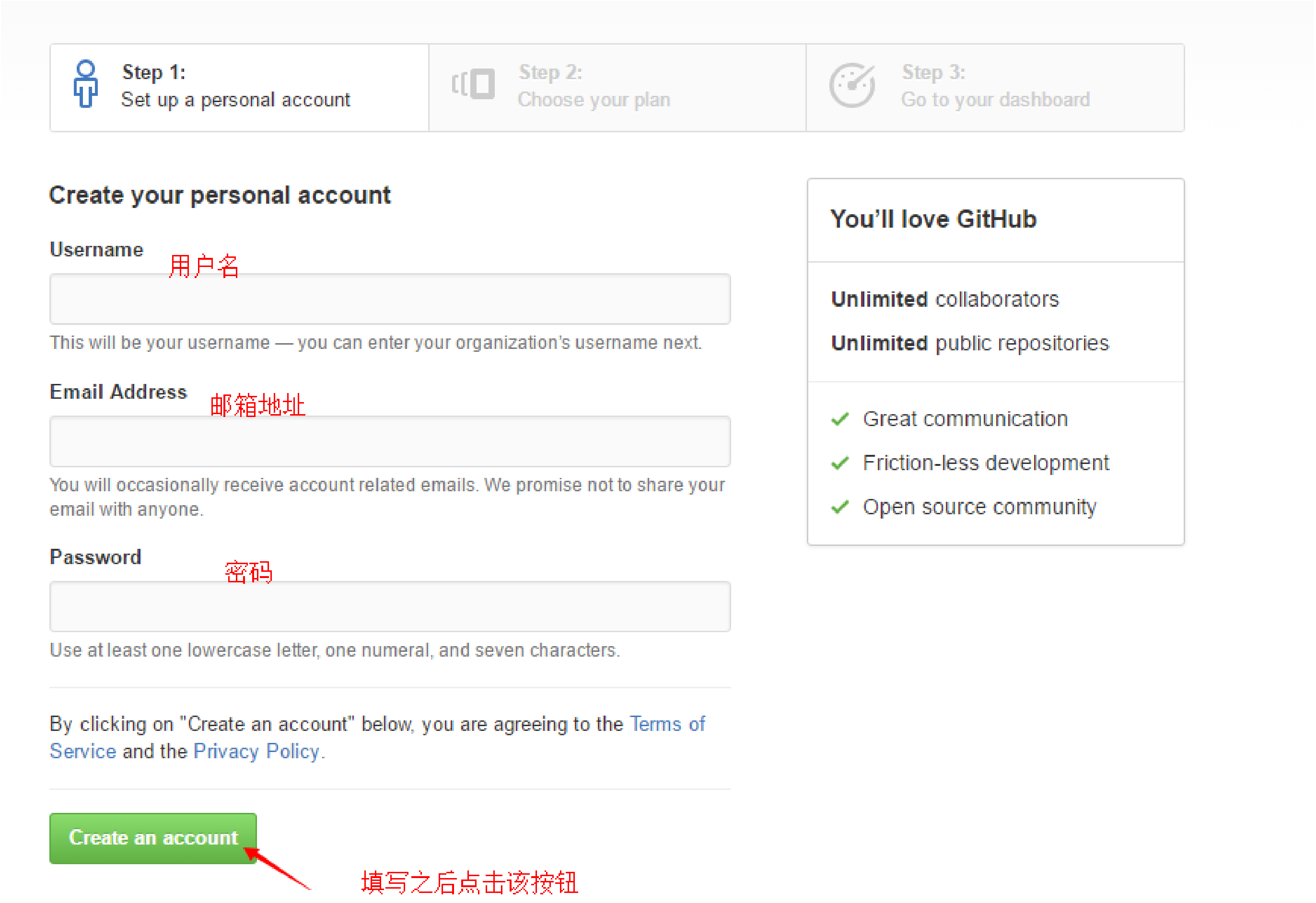
Unlimited collaborators

Unlimited public repositories

✓ Great communication

✓ Friction-less development

✓ Open source community



Create your personal account

Username 用户名

This will be your username — you can enter your organization's username next.

Email Address 邮箱地址

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password 密码

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the Terms of Service and the Privacy Policy.

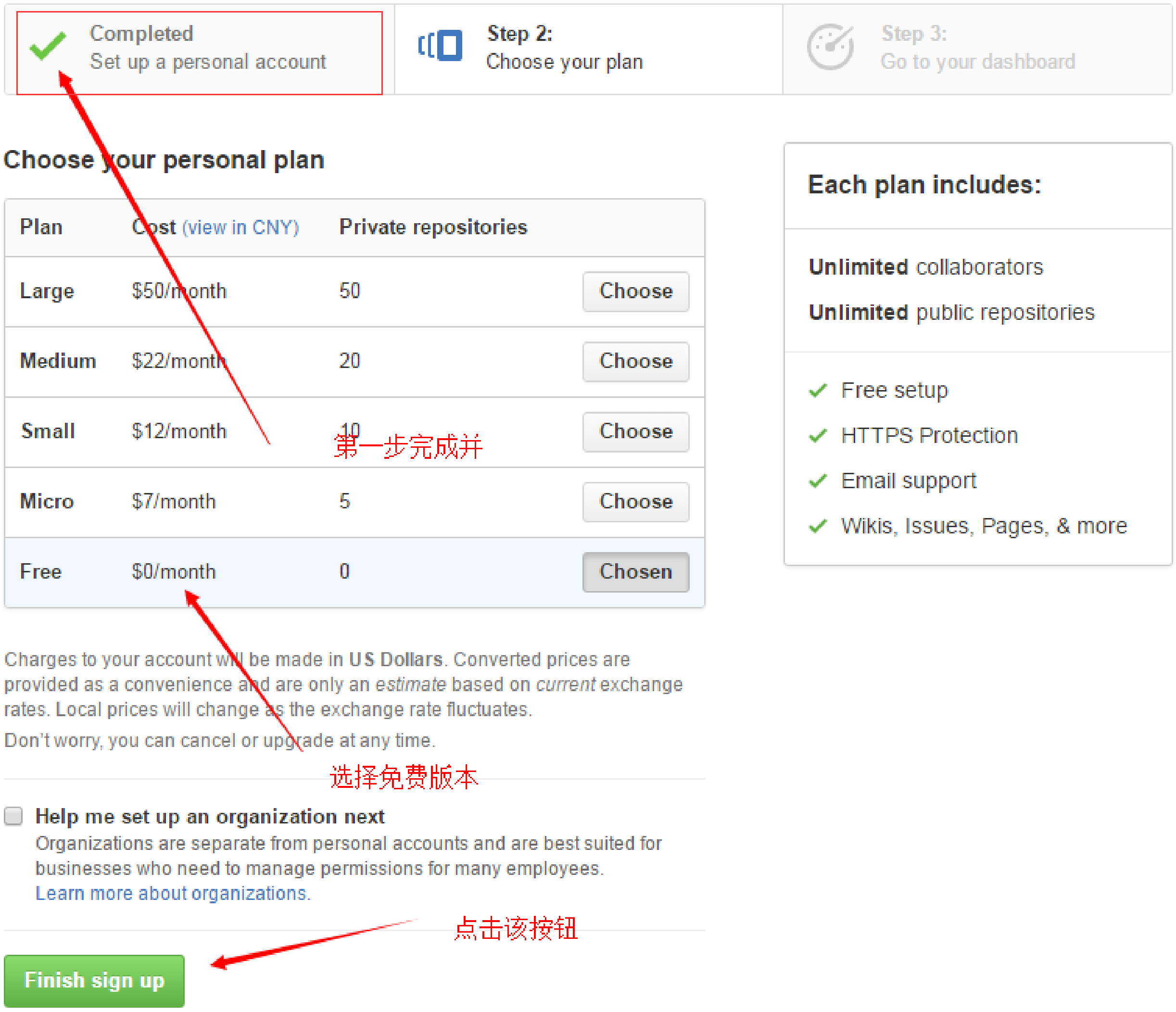
填写之后点击该按钮

You'll love GitHub

Unlimited collaborators

Unlimited public repositories

- ✓ Great communication
- ✓ Friction-less development
- ✓ Open source community



✓ Completed
Set up a personal account

Step 2:
Choose your plan

Step 3:
Go to your dashboard

第一步完成并

Choose your personal plan

Plan	Cost (view in CNY)	Private repositories	
Large	$50/month	50	Choose
Medium	$22/month	20	Choose
Small	$12/month	10	Choose
Micro	$7/month	5	Choose
Free	$0/month	0	Chosen

Charges to your account will be made in **US Dollars**. Converted prices are provided as a convenience and are only an *estimate* based on *current* exchange rates. Local prices will change as the exchange rate fluctuates.

Don't worry, you can cancel or upgrade at any time.

选择免费版本

□ **Help me set up an organization next**
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
Learn more about organizations.

点击该按钮

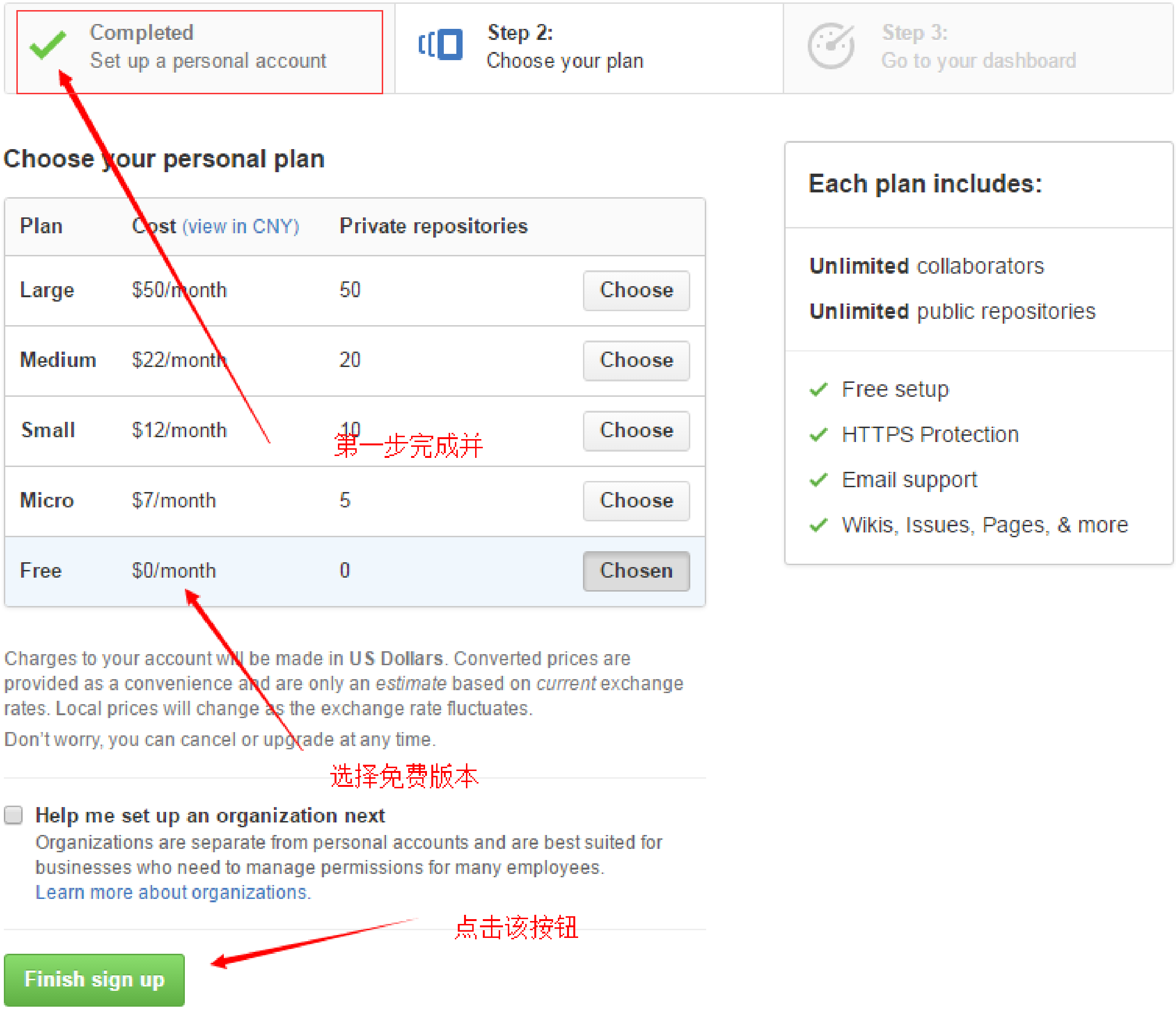
Finish sign up

Each plan includes:

Unlimited collaborators

Unlimited public repositories

✓ Free setup
✓ HTTPS Protection
✓ Email support
✓ Wikis, Issues, Pages, & more



✓ Completed
Set up a personal account

Step 2:
Choose your plan

Step 3:
Go to your dashboard

Choose your personal plan

Plan	Cost (view in CNY)	Private repositories	
Large	$50/month	50	Choose
Medium	$22/month	20	Choose
Small	$12/month	10	Choose
Micro	$7/month	5	Choose
Free	$0/month	0	Chosen

第一步完成并

Charges to your account will be made in **US Dollars**. Converted prices are provided as a convenience and are only an *estimate* based on *current* exchange rates. Local prices will change as the exchange rate fluctuates.

Don't worry, you can cancel or upgrade at any time.

选择免费版本

□ **Help me set up an organization next**
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
Learn more about organizations.

点击该按钮

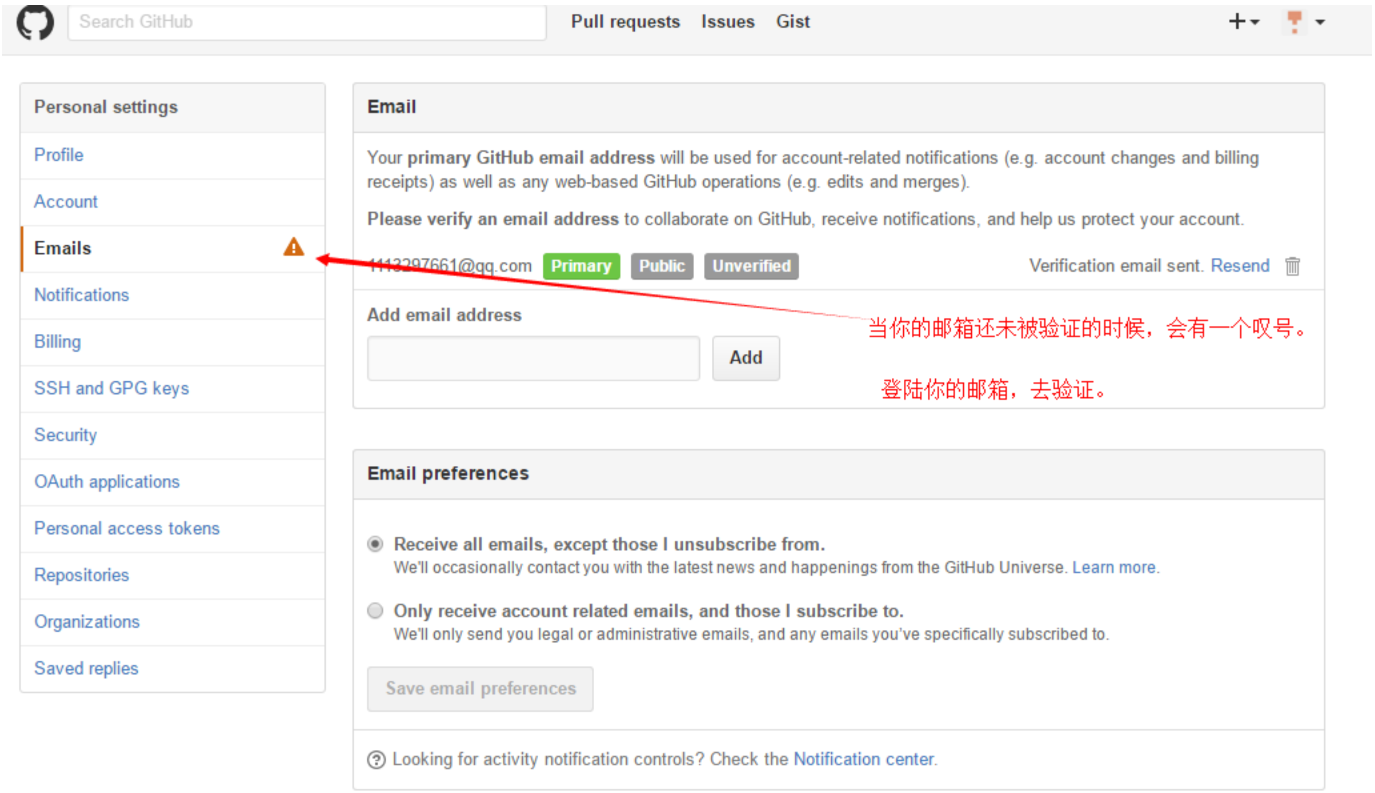
Finish sign up

Each plan includes:

Unlimited collaborators

Unlimited public repositories

✓ Free setup
✓ HTTPS Protection
✓ Email support
✓ Wikis, Issues, Pages, & more



Personal settings
- Profile
- Account
- Emails ⚠
- Notifications
- Billing
- SSH and GPG keys
- Security
- OAuth applications
- Personal access tokens
- Repositories
- Organizations
- Saved replies

Email

Your **primary GitHub email address** will be used for account-related notifications (e.g. account changes and billing receipts) as well as any web-based GitHub operations (e.g. edits and merges).

Please verify an email address to collaborate on GitHub, receive notifications, and help us protect your account.

1113297661@qq.com Primary Public Unverified Verification email sent. Resend

Add email address

Add

当你的邮箱还未被验证的时候，会有一个叹号。

登陆你的邮箱，去验证。

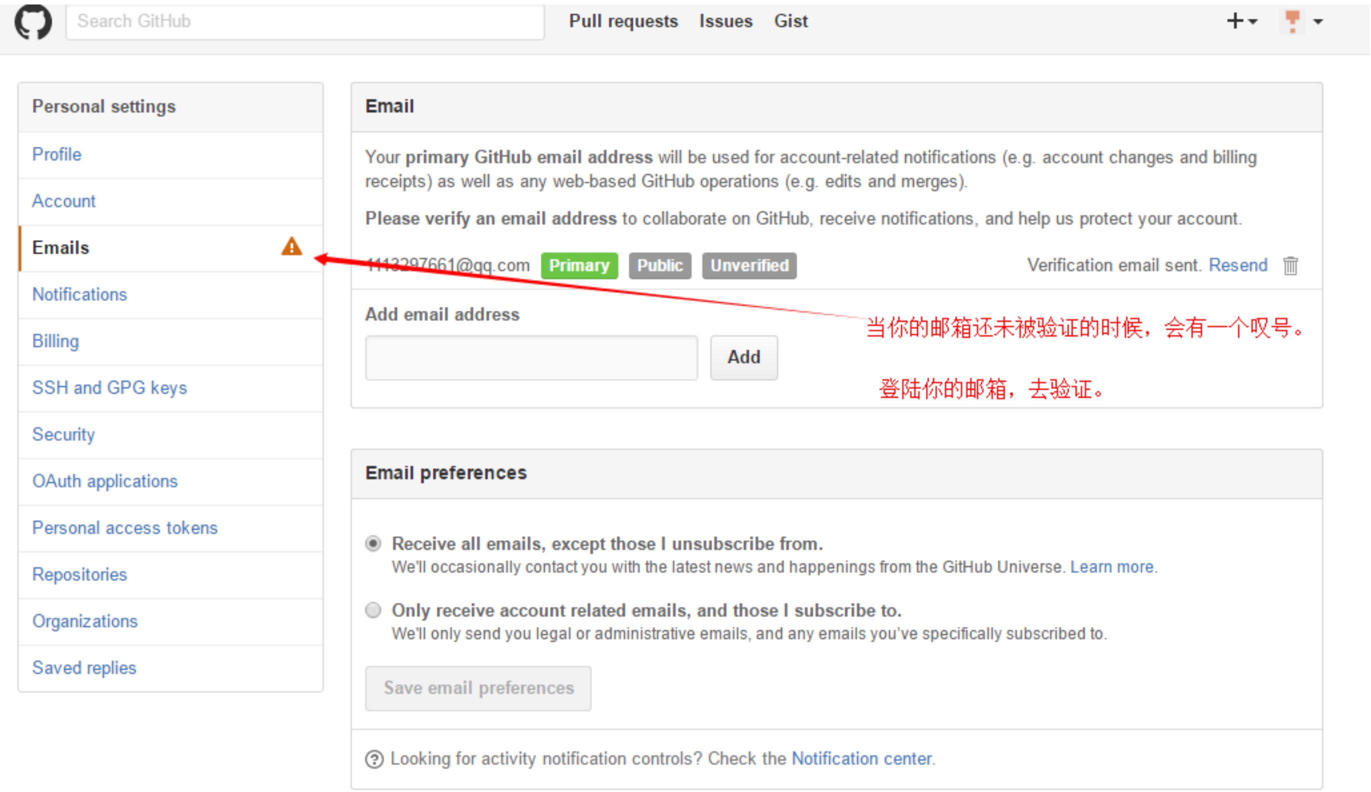
Email preferences

● **Receive all emails, except those I unsubscribe from.**
We'll occasionally contact you with the latest news and happenings from the GitHub Universe. Learn more.

○ **Only receive account related emails, and those I subscribe to.**
We'll only send you legal or administrative emails, and any emails you've specifically subscribed to.

Save email preferences

Looking for activity notification controls? Check the Notification center.



Personal settings

- Profile
- Account
- Emails ⚠
- Notifications
- Billing
- SSH and GPG keys
- Security
- OAuth applications
- Personal access tokens
- Repositories
- Organizations
- Saved replies

Email

Your **primary GitHub email address** will be used for account-related notifications (e.g. account changes and billing receipts) as well as any web-based GitHub operations (e.g. edits and merges).

Please verify an email address to collaborate on GitHub, receive notifications, and help us protect your account.

1113297661@qq.com Primary Public Unverified Verification email sent. Resend

Add email address

当你的邮箱还未被验证的时候，会有一个叹号。

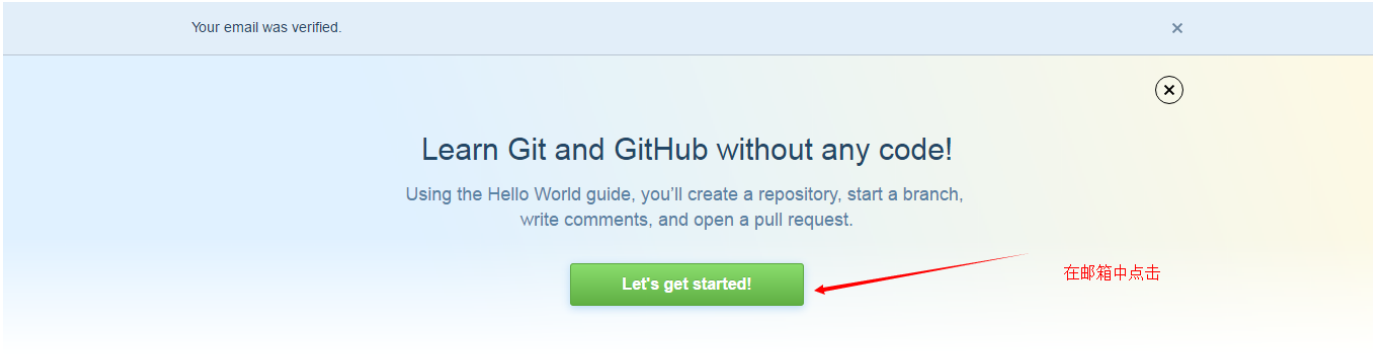
登陆你的邮箱，去验证。

Email preferences

● **Receive all emails, except those I unsubscribe from.**
We'll occasionally contact you with the latest news and happenings from the GitHub Universe. Learn more.

○ **Only receive account related emails, and those I subscribe to.**
We'll only send you legal or administrative emails, and any emails you've specifically subscribed to.

Looking for activity notification controls? Check the Notification center.

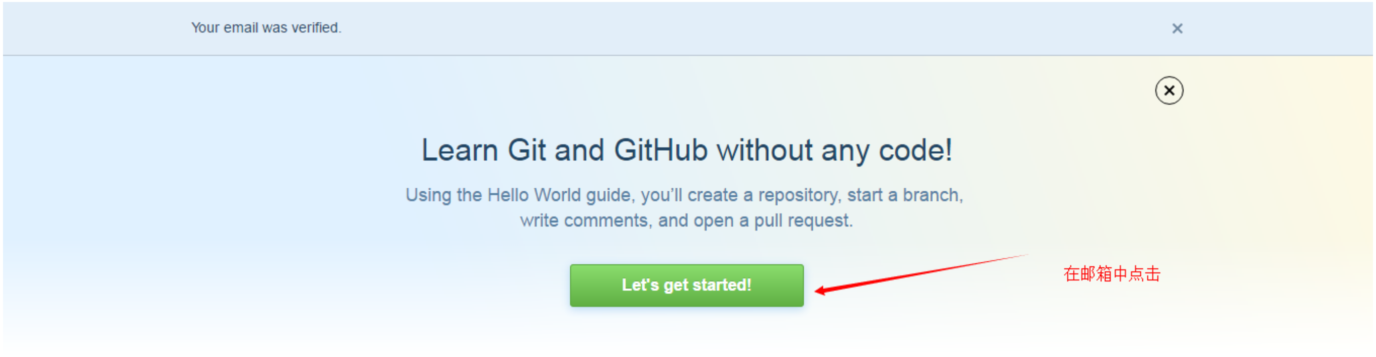


Your email was verified.

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

在邮箱中点击



Your email was verified.

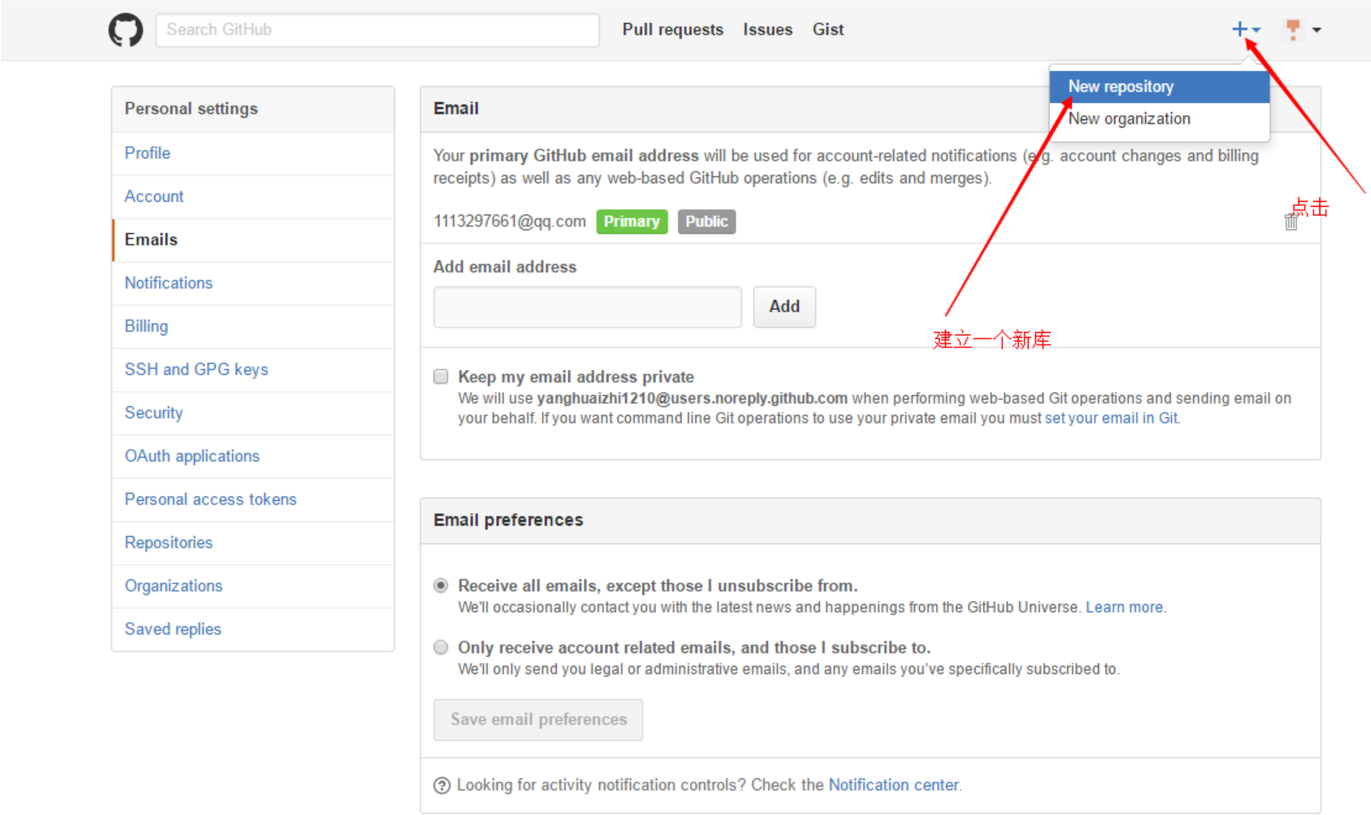
Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Let's get started!

在邮箱中点击

熟悉的Hello world,然而并没有什么用。



Personal settings

- Profile
- Account
- Emails
- Notifications
- Billing
- SSH and GPG keys
- Security
- OAuth applications
- Personal access tokens
- Repositories
- Organizations
- Saved replies

Email

Your **primary GitHub email address** will be used for account-related notifications (e.g. account changes and billing receipts) as well as any web-based GitHub operations (e.g. edits and merges).

1113297661@qq.com Primary Public

Add email address

Add

□ **Keep my email address private**
We will use **yanghuaizhi1210@users.noreply.github.com** when performing web-based Git operations and sending email on your behalf. If you want command line Git operations to use your private email you must set your email in Git.

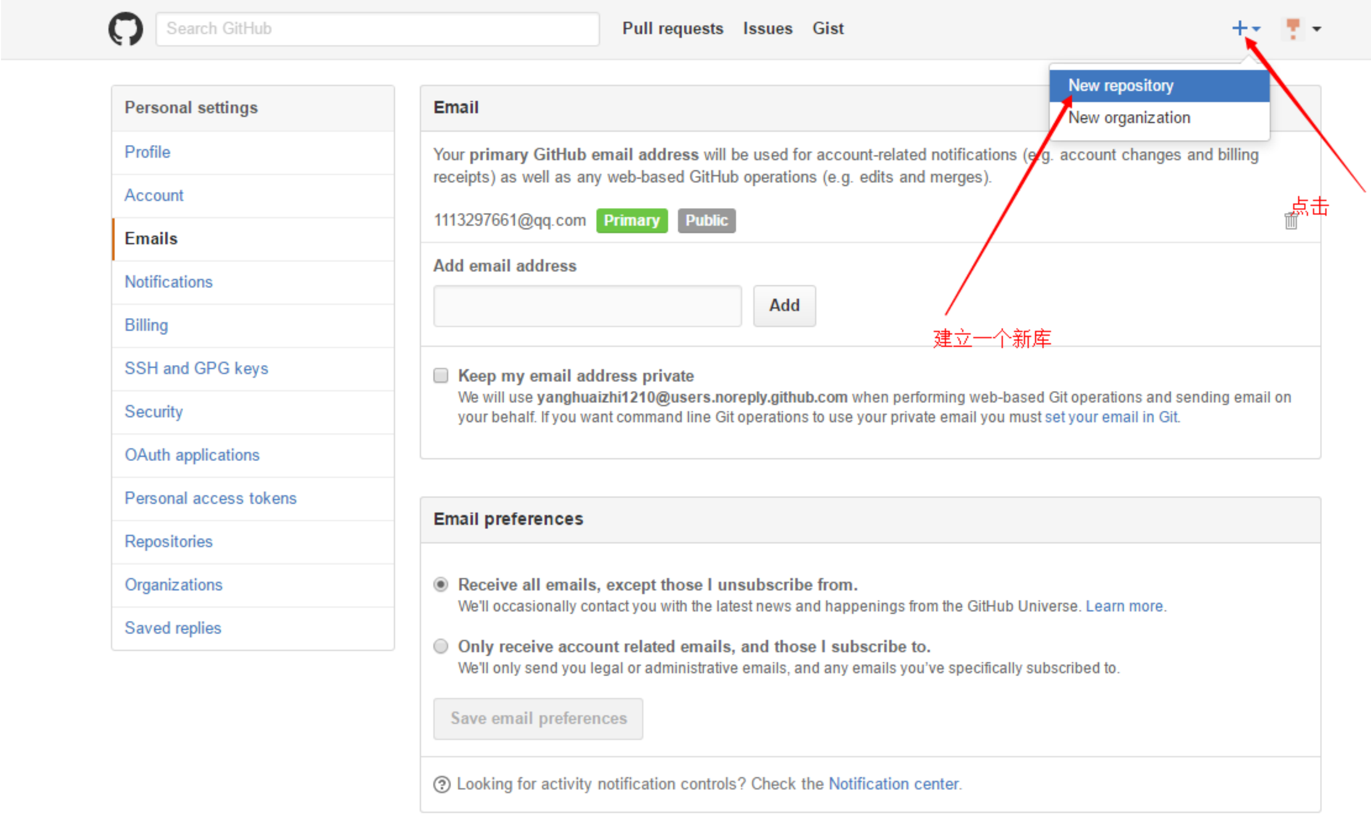
Email preferences

● **Receive all emails, except those I unsubscribe from.**
We'll occasionally contact you with the latest news and happenings from the GitHub Universe. Learn more.

○ **Only receive account related emails, and those I subscribe to.**
We'll only send you legal or administrative emails, and any emails you've specifically subscribed to.

Save email preferences

Looking for activity notification controls? Check the Notification center.


Personal settings

- Profile
- Account
- Emails
- Notifications
- Billing
- SSH and GPG keys
- Security
- OAuth applications
- Personal access tokens
- Repositories
- Organizations
- Saved replies

Email

Your **primary GitHub email address** will be used for account-related notifications (e.g. account changes and billing receipts) as well as any web-based GitHub operations (e.g. edits and merges).

1113297661@qq.com Primary Public

Add email address

Add

□ **Keep my email address private**
We will use **yanghuaizhi1210@users.noreply.github.com** when performing web-based Git operations and sending email on your behalf. If you want command line Git operations to use your private email you must set your email in Git.

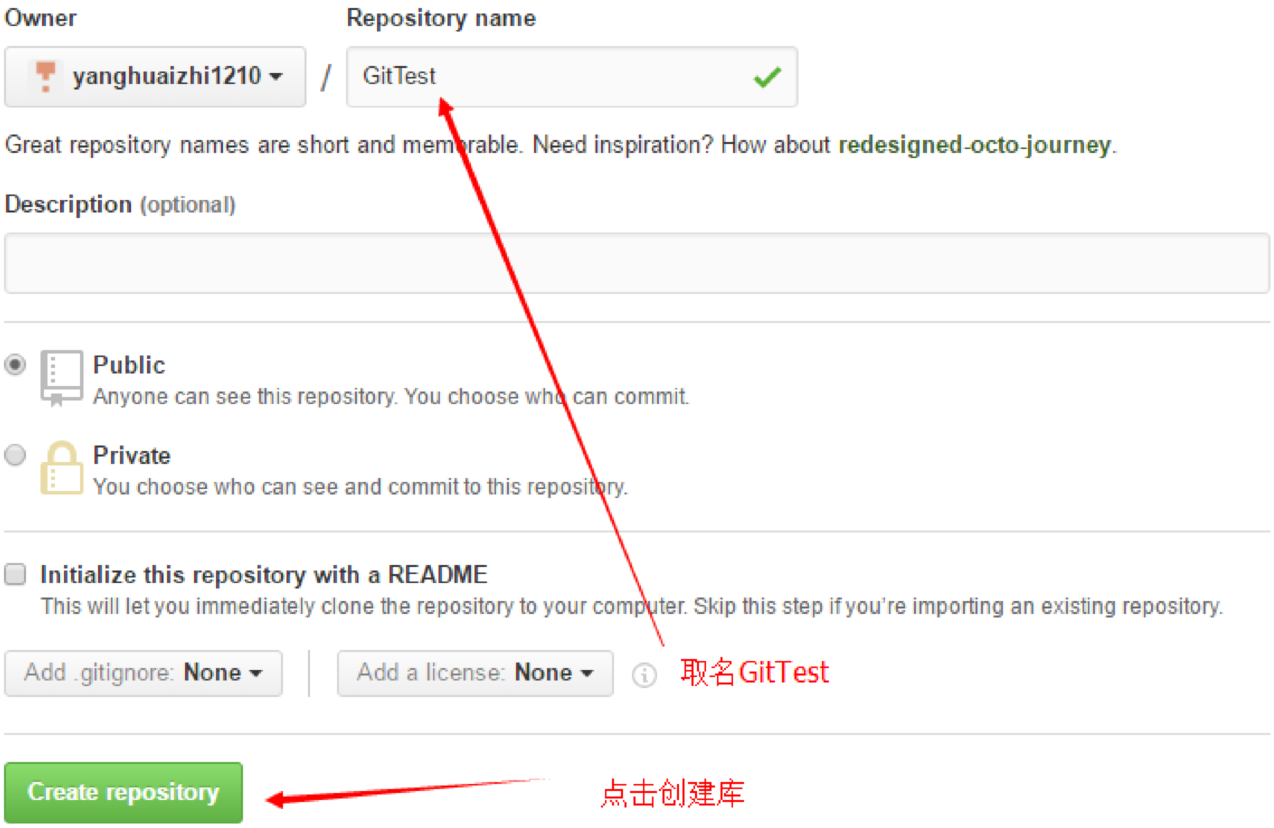
Email preferences

● **Receive all emails, except those I unsubscribe from.**
We'll occasionally contact you with the latest news and happenings from the GitHub Universe. Learn more.

○ **Only receive account related emails, and those I subscribe to.**
We'll only send you legal or administrative emails, and any emails you've specifically subscribed to.

Save email preferences

Looking for activity notification controls? Check the Notification center.



Owner

yanghuaizhi1210 ▾

Repository name

/ GitTest ✓

Great repository names are short and memorable. Need inspiration? How about **redesigned-octo-journey**.

Description (optional)

● **Public**
Anyone can see this repository. You choose who can commit.

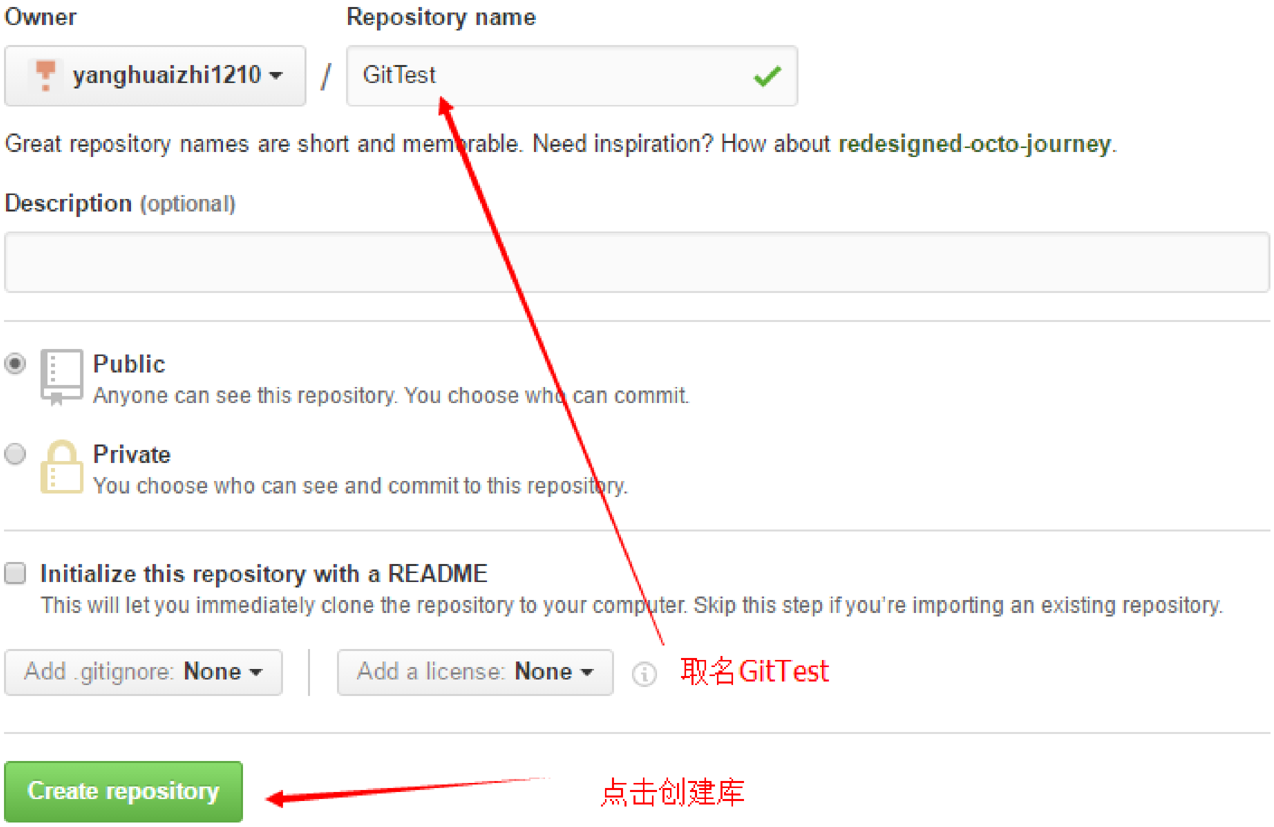
○ **Private**
You choose who can see and commit to this repository.

□ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

取名GitTest

点击创建库



Owner

yanghuaizhi1210 ▾

Repository name

/ GitTest ✓

Great repository names are short and memorable. Need inspiration? How about **redesigned-octo-journey**.

Description (optional)

● **Public**
Anyone can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

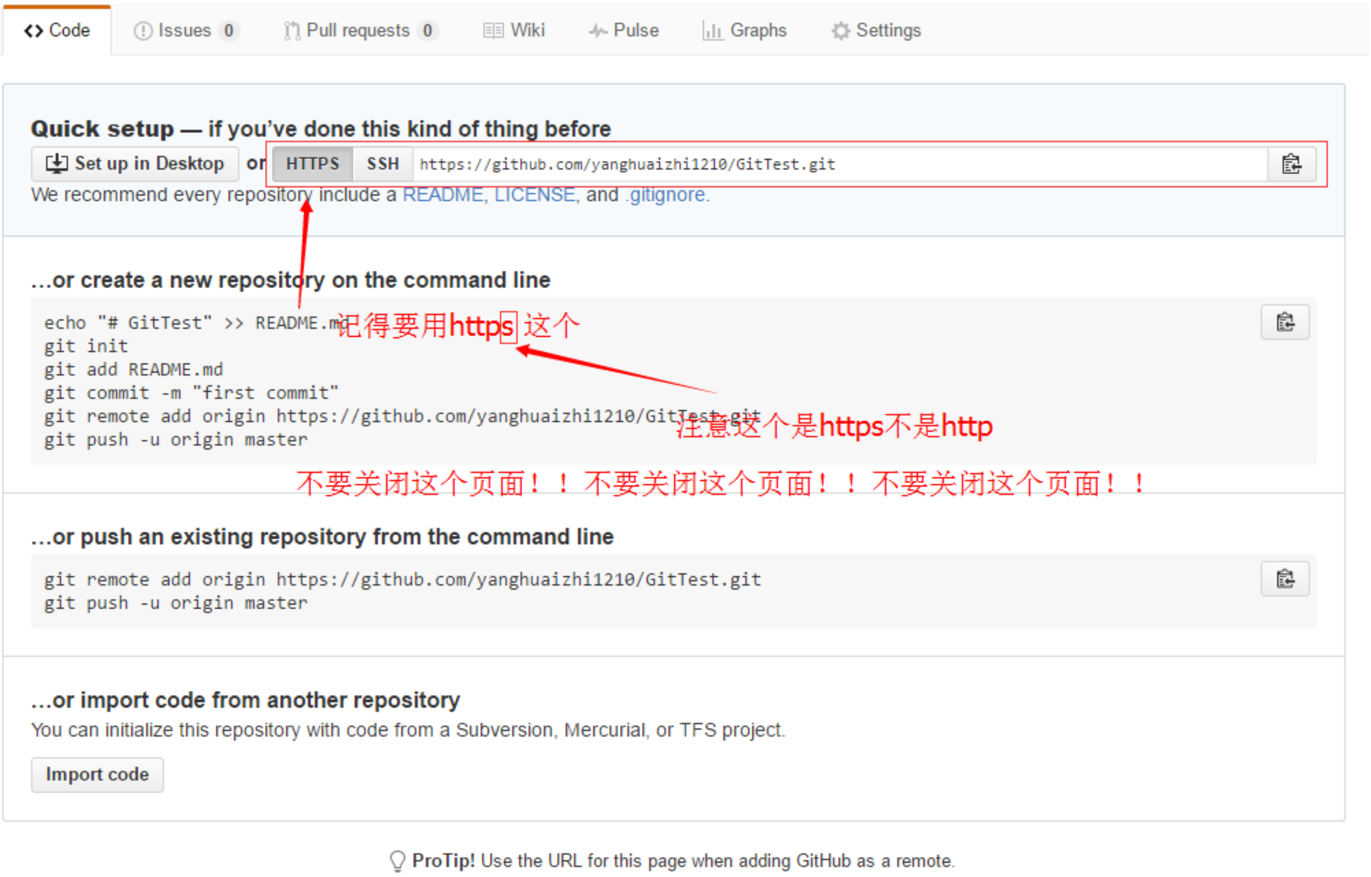
□ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

取名GitTest

Create repository

点击创建库



Quick setup — if you've done this kind of thing before

https://github.com/yanghuaizhi1210/GitTest.git

We recommend every repository include a README, LICENSE, and .gitignore.

…or create a new repository on the command line

```
echo "# GitTest" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/yanghuaizhi1210/GitTest.git
git push -u origin master
```

记得要用https 这个

注意这个是https不是http

不要关闭这个页面！！不要关闭这个页面！！不要关闭这个页面！！

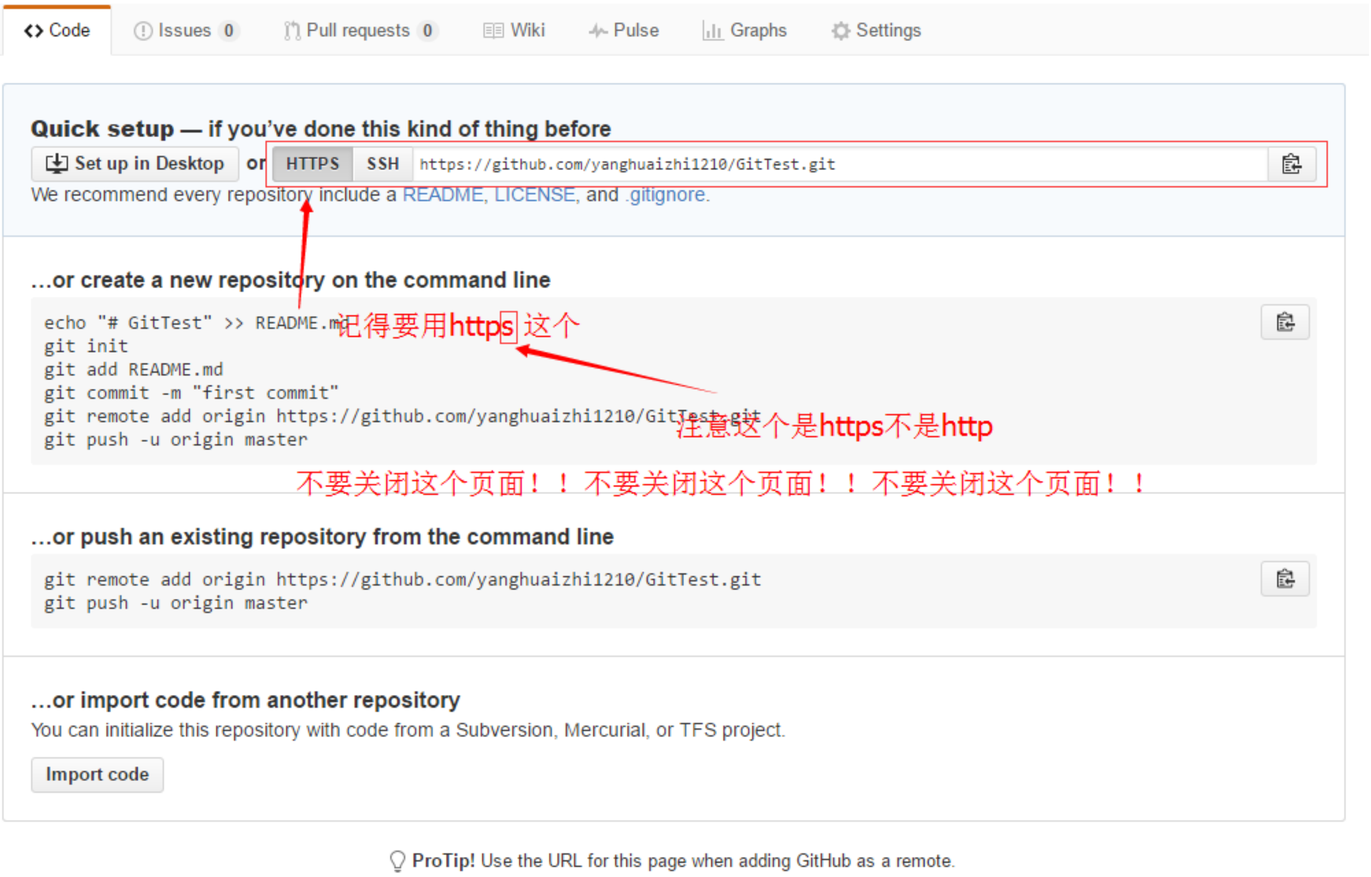
…or push an existing repository from the command line

```
git remote add origin https://github.com/yanghuaizhi1210/GitTest.git
git push -u origin master
```

…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

ProTip! Use the URL for this page when adding GitHub as a remote.



Quick setup — if you've done this kind of thing before

https://github.com/yanghuaizhi1210/GitTest.git

We recommend every repository include a README, LICENSE, and .gitignore.

…or create a new repository on the command line

```
echo "# GitTest" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/yanghuaizhi1210/GitTest.git
git push -u origin master
```

记得要用https 这个

注意这个是https不是http

不要关闭这个页面！！不要关闭这个页面！！不要关闭这个页面！！

…or push an existing repository from the command line

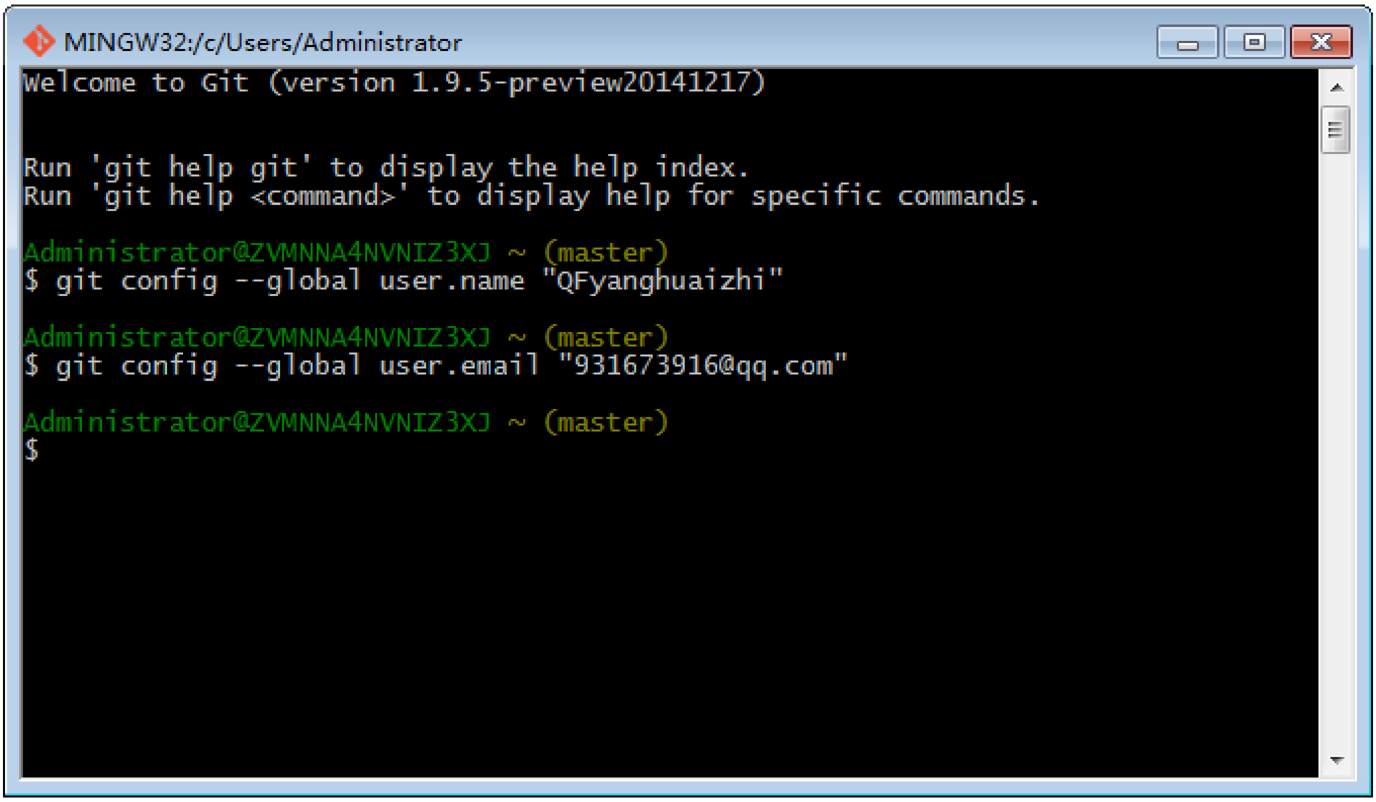
```
git remote add origin https://github.com/yanghuaizhi1210/GitTest.git
git push -u origin master
```

…or import code from another repository

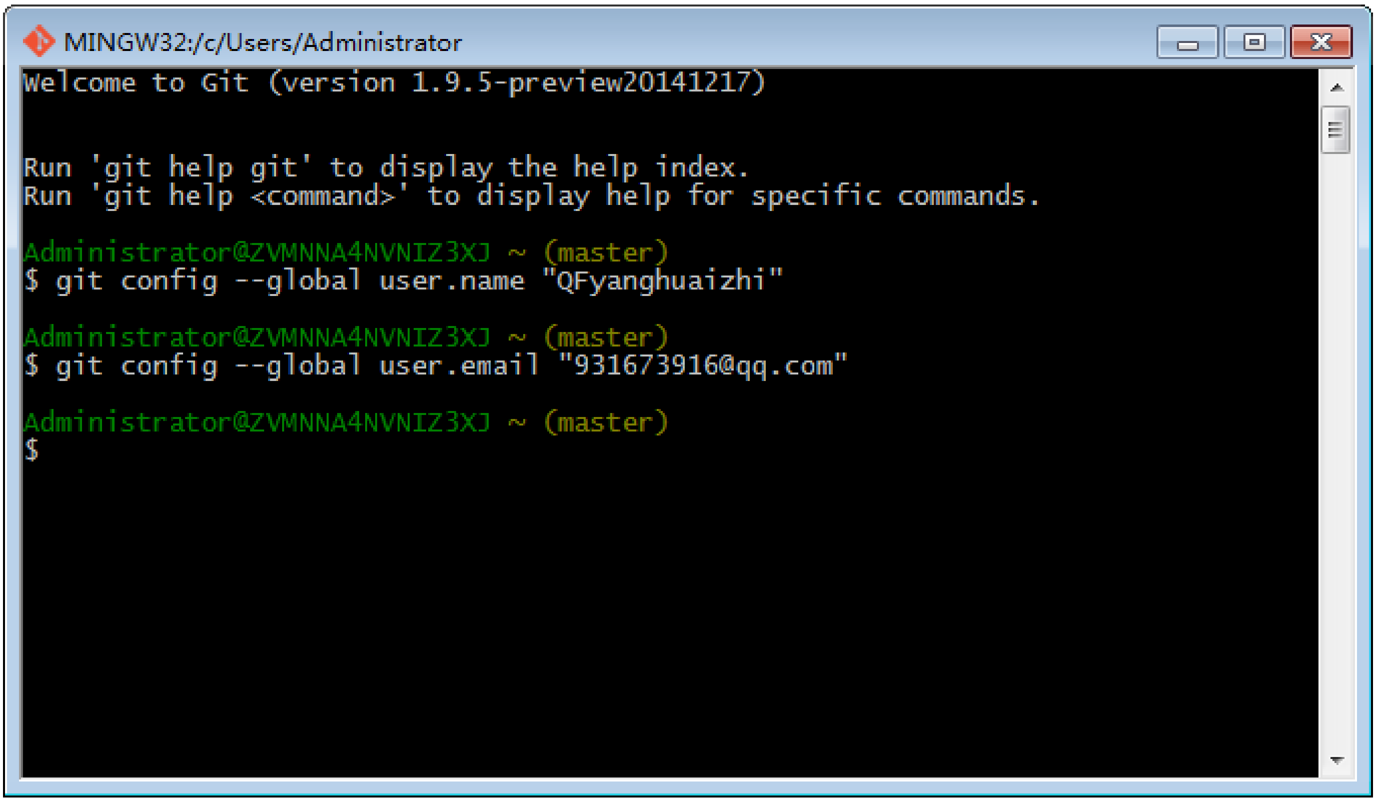
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

ProTip! Use the URL for this page when adding GitHub as a remote.

安装结束后需要进行一些设置



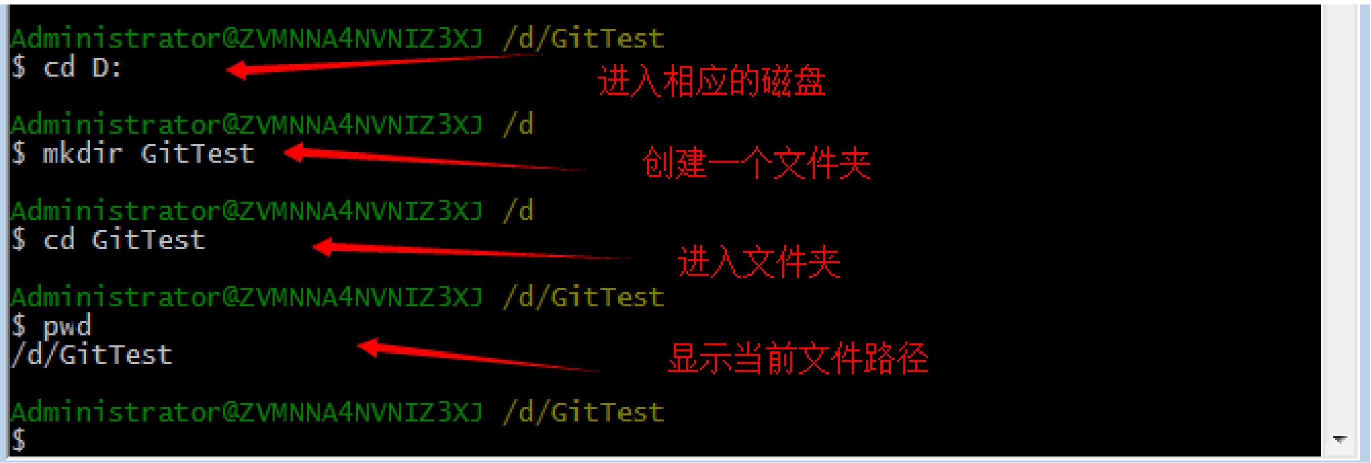
```
Welcome to Git (version 1.9.5-preview20141217)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Administrator@ZVMNNA4NVNIZ3XJ ~ (master)
$ git config --global user.name "QFyanghuaizhi"

Administrator@ZVMNNA4NVNIZ3XJ ~ (master)
$ git config --global user.email "931673916@qq.com"

Administrator@ZVMNNA4NVNIZ3XJ ~ (master)
$
```



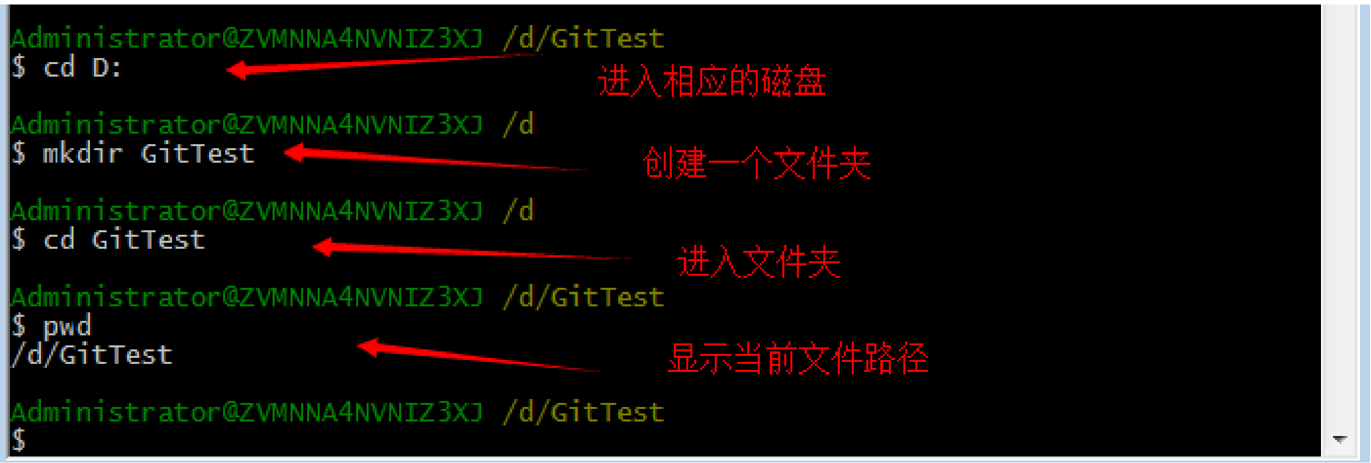
```
Welcome to Git (version 1.9.5-preview20141217)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Administrator@ZVMNNA4NVNIZ3XJ ~ (master)
$ git config --global user.name "QFyanghuaizhi"

Administrator@ZVMNNA4NVNIZ3XJ ~ (master)
$ git config --global user.email "931673916@qq.com"

Administrator@ZVMNNA4NVNIZ3XJ ~ (master)
$
```

命令 1 .　　git config --global user.name ″你的git名称″

命令2.　　git config --global user.email ″你的git验证邮箱″



```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest
$ cd D:

Administrator@ZVMNNA4NVNIZ3XJ /d
$ mkdir GitTest

Administrator@ZVMNNA4NVNIZ3XJ /d
$ cd GitTest

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest
$ pwd
/d/GitTest

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest
$
```



```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest
$ cd D:

Administrator@ZVMNNA4NVNIZ3XJ /d
$ mkdir GitTest

Administrator@ZVMNNA4NVNIZ3XJ /d
$ cd GitTest

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest
$ pwd
/d/GitTest

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest
$
```

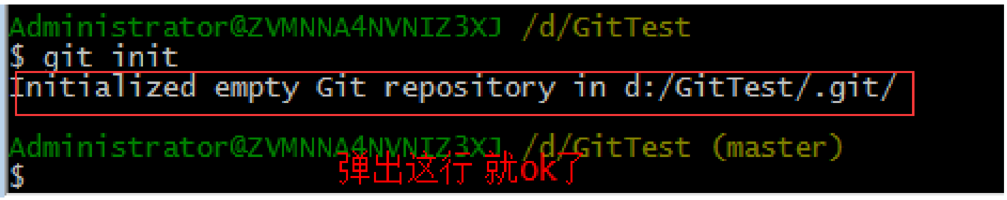
命令 1 . cd D: 进入 相应的磁盘

命令 2 . mkdir '你自己定义的文件夹名'

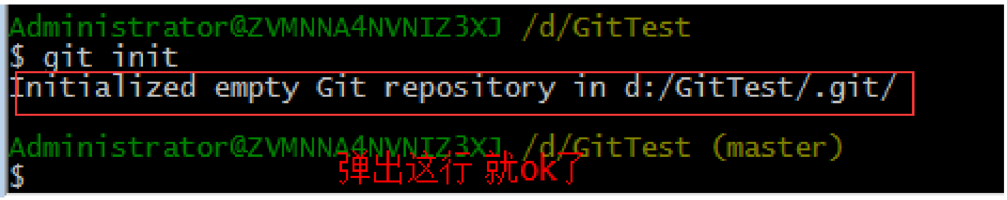
命令 3 . cd '你自己定义的文件夹名'

命令 4. pwd 显示（你自己定义的文件夹名）的路径

然后！最重要的步骤来了！！git init



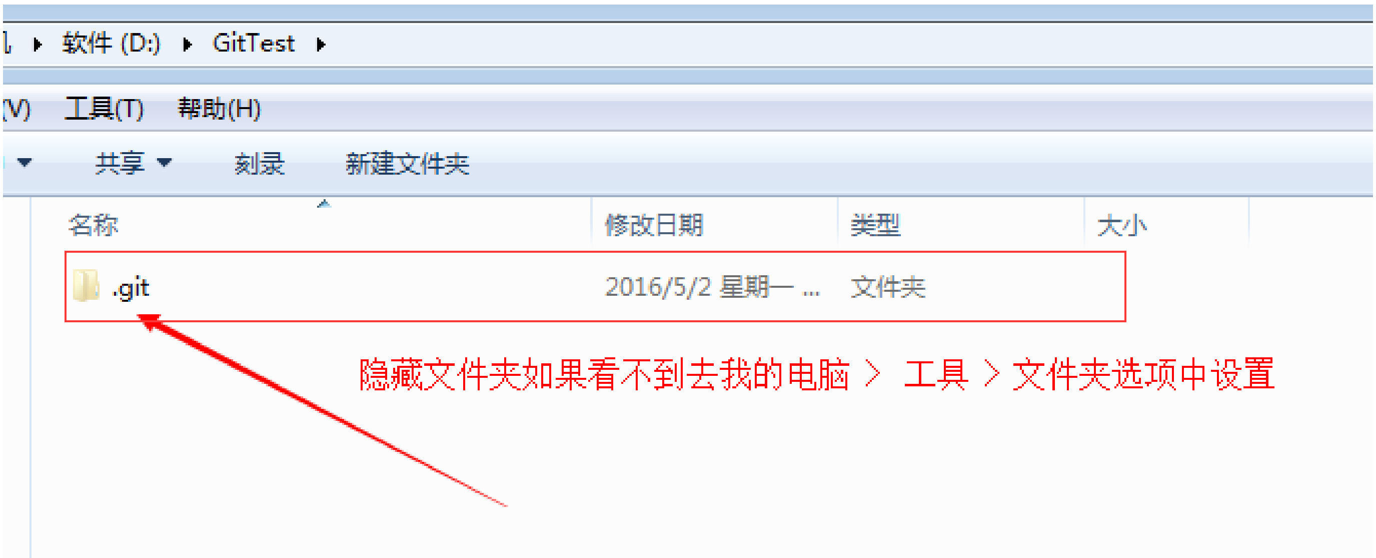
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest
$ git init
Initialized empty Git repository in d:/GitTest/.git/

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
```



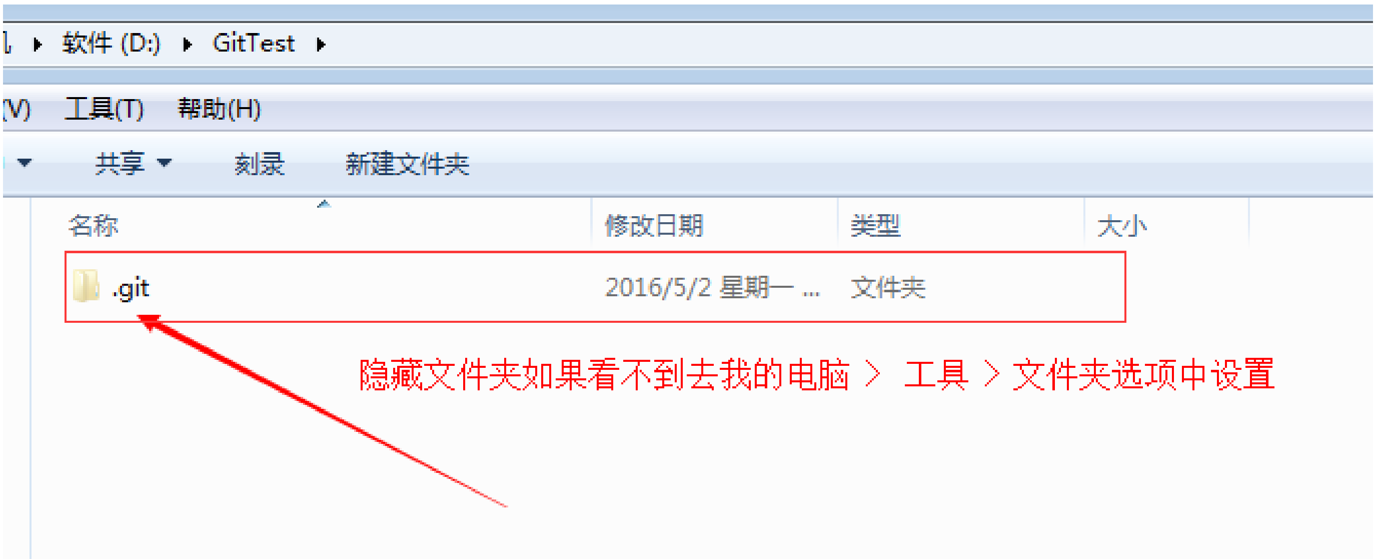
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest
$ git init
Initialized empty Git repository in d:/GitTest/.git/

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
```

命令 1 . git init

显示成功后去相应的文件夹中查看是不是多了一个git文件



隐藏文件夹如果看不到去我的电脑 > 工具 > 文件夹选项中设置

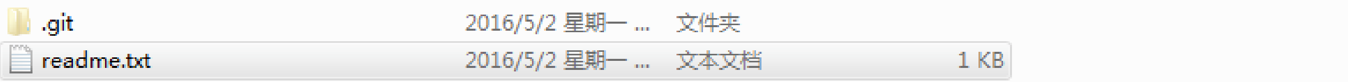


隐藏文件夹如果看不到去我的电脑 > 工具 > 文件夹选项中设置

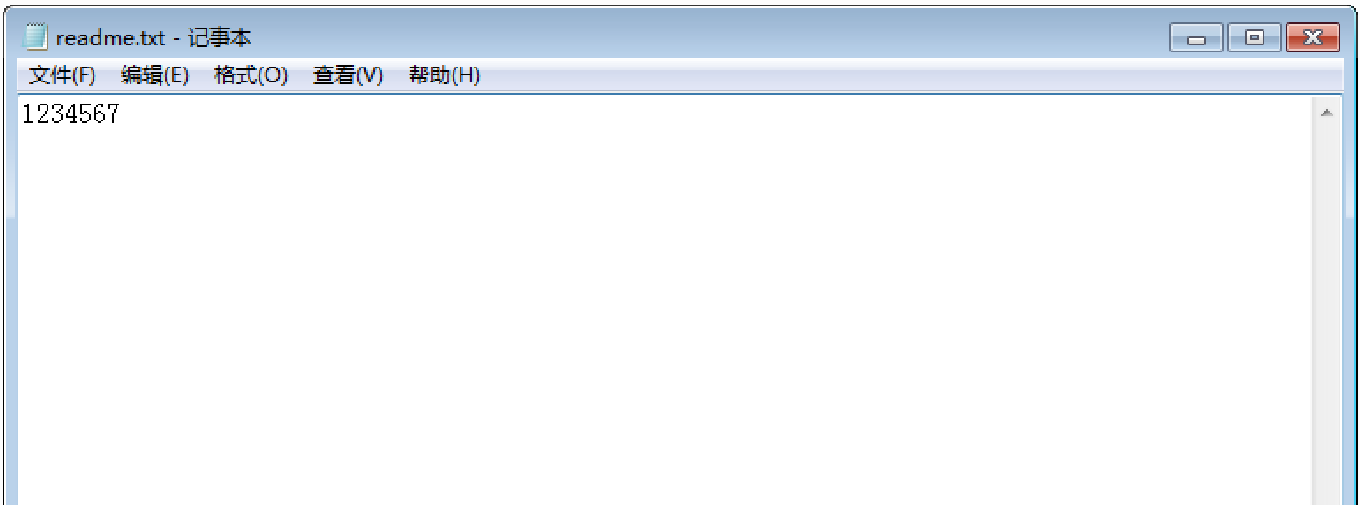
这个文件夹不要乱动！！这个文件夹不要乱动！！这个文件夹不要乱动！！

这个文件夹就是你的版本库了，把项目放在文件夹下。

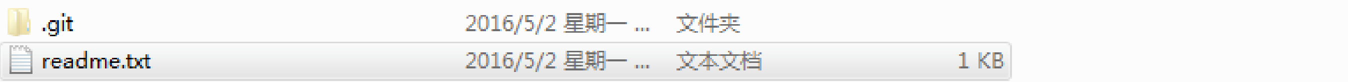
现在进行一个测试。在当前目录先建立一个readme. txt文件，并且随便写的什么。



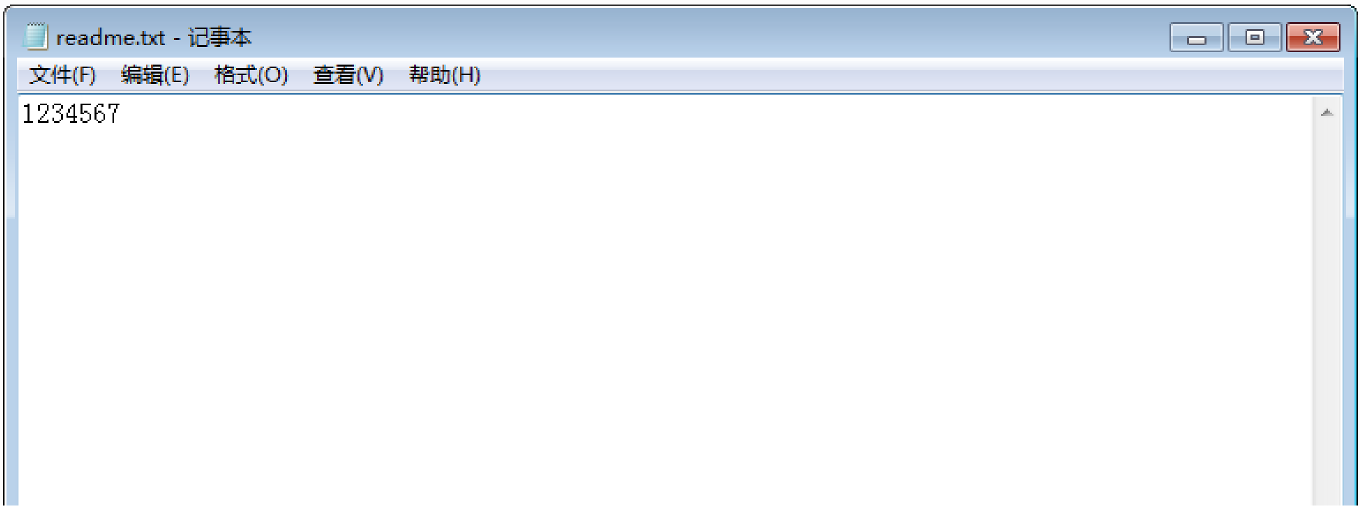
```
.git          2016/5/2 星期一 ...  文件夹
readme.txt    2016/5/2 星期一 ...  文本文档    1 KB
```



```
1234567
```



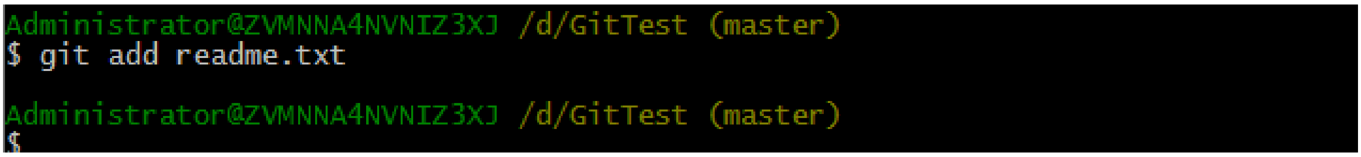
```
.git          2016/5/2 星期一 ...  文件夹
readme.txt    2016/5/2 星期一 ...  文本文档    1 KB
```



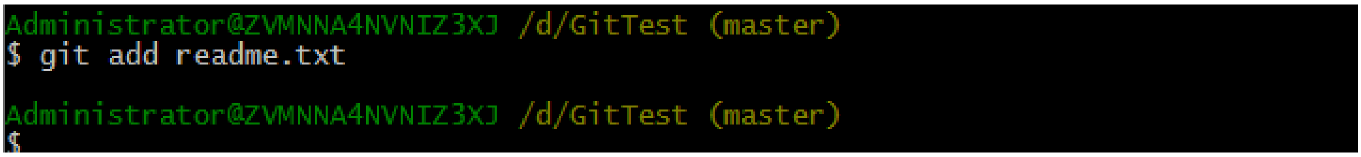
```
1234567
```

五.如何将文件提交到git服务器。

1.打开 Git Bash



```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git add readme.txt

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
```

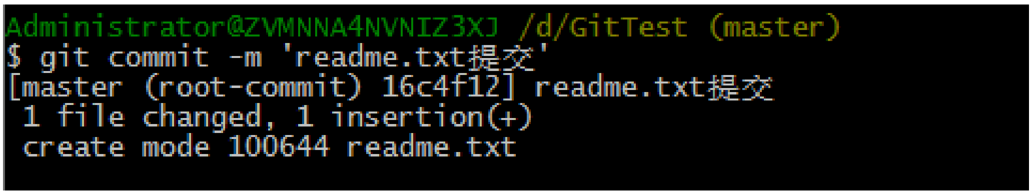


```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git add readme.txt

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
```

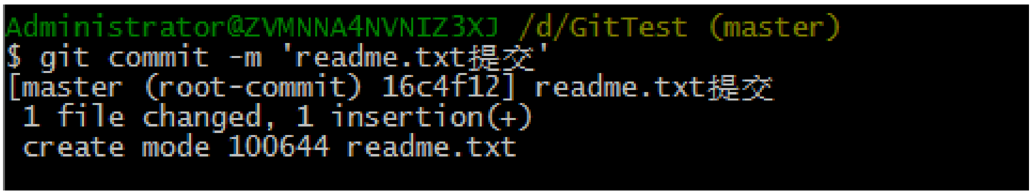
命令 1. git add ‘文件名’

把咱们的文件存放在一个神秘的地方 >>>> 暂存区 （暂存区在本机）

2.把文件提交到仓库中去。



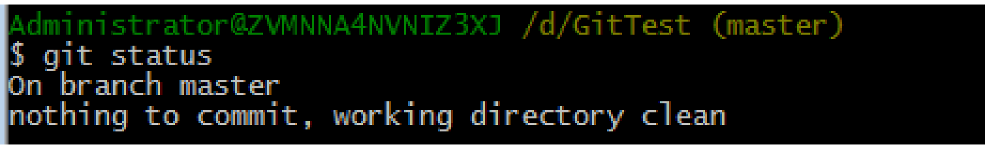
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git commit -m 'readme.txt提交'
[master (root-commit) 16c4f12] readme.txt提交
 1 file changed, 1 insertion(+)
 create mode 100644 readme.txt
```



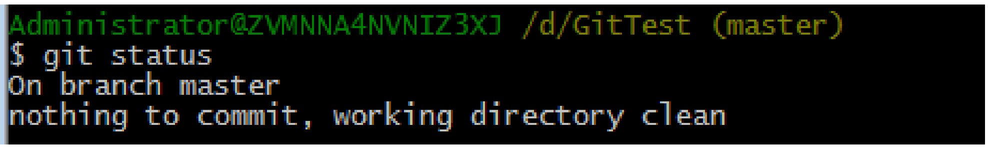
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git commit -m 'readme.txt提交'
[master (root-commit) 16c4f12] readme.txt提交
 1 file changed, 1 insertion(+)
 create mode 100644 readme.txt
```

命令 1 . git commit -m ’我提交时候的备注（相当于注释）’

3.查看commit状态



```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git status
On branch master
nothing to commit, working directory clean
```

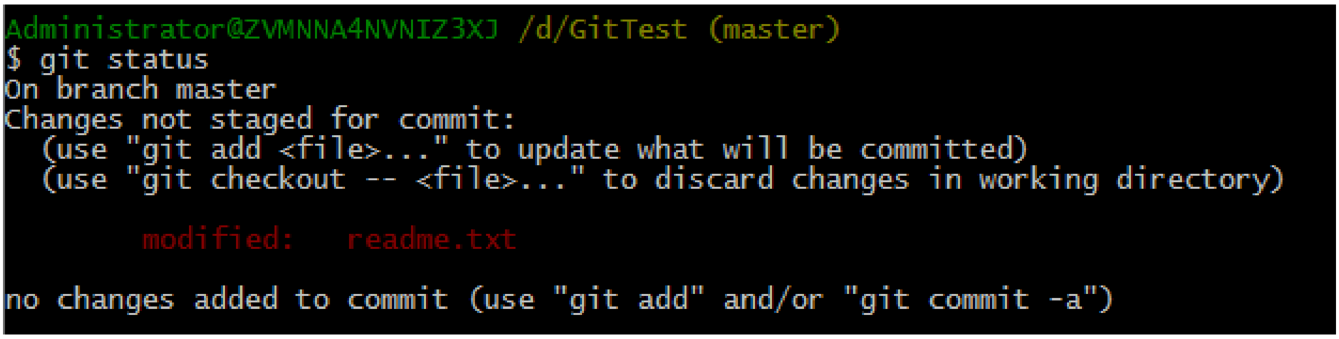


```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git status
On branch master
nothing to commit, working directory clean
```

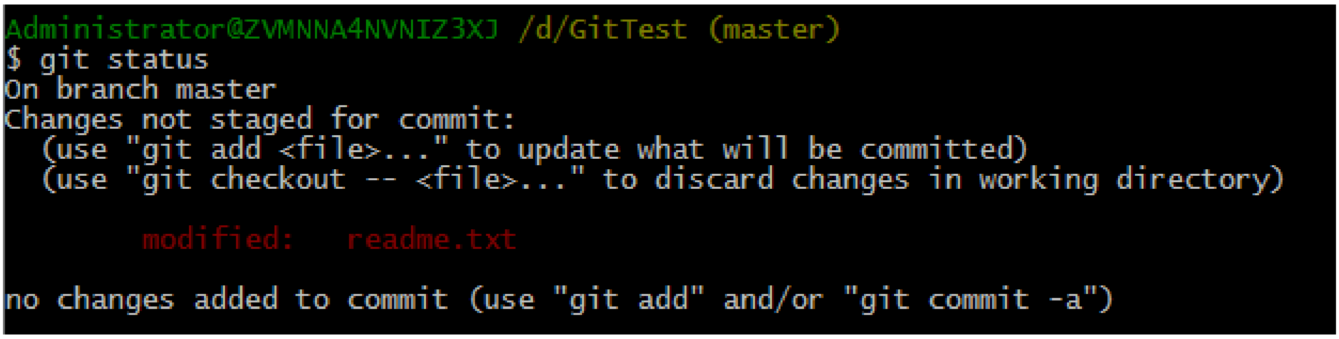
命令1 git status

出现 如下提示就说明咱们的commit是成功的。

修改下readme.txt里面的内容。并重新使用git status命令 。



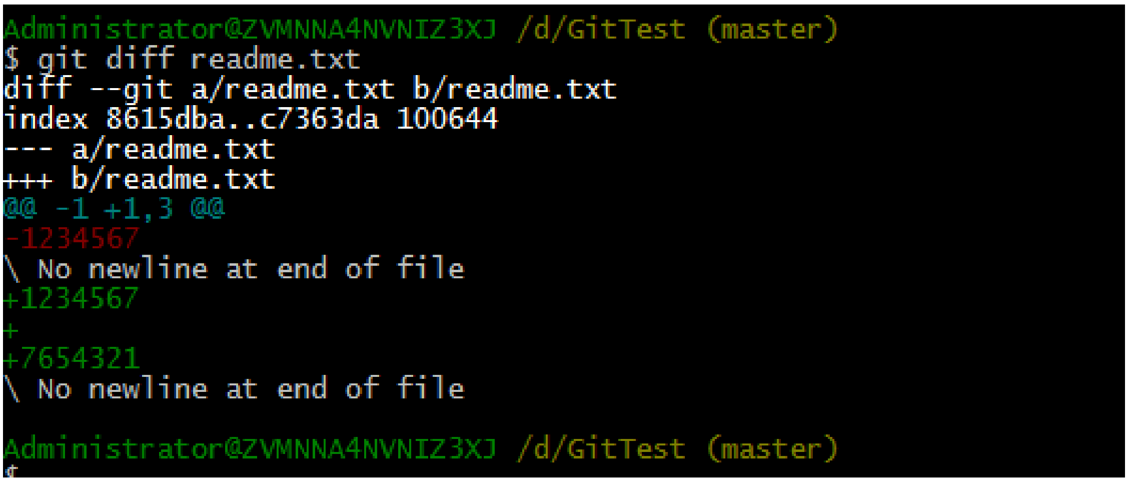
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   readme.txt

no changes added to commit (use "git add" and/or "git commit -a")
```



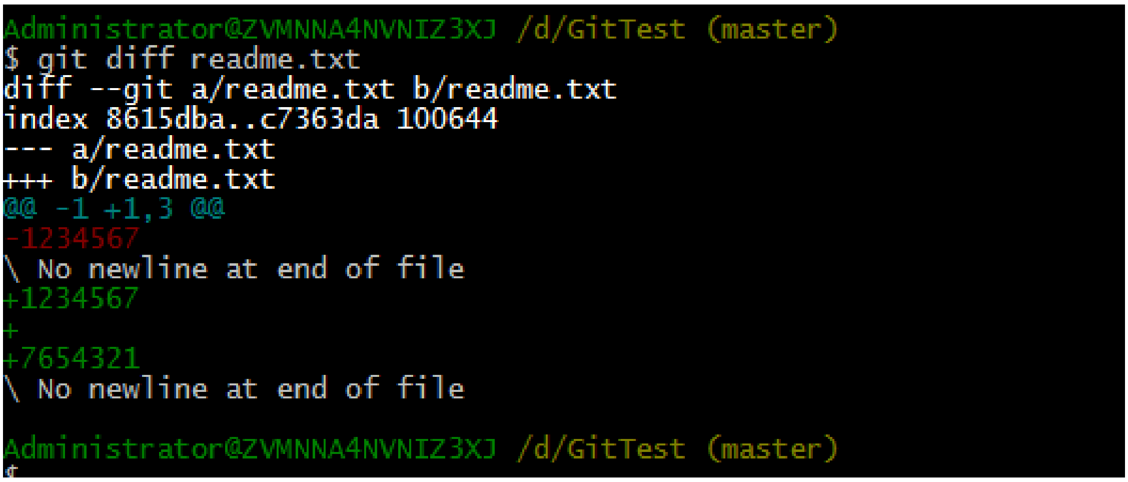
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   readme.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

会发现出现了一些变化。git告诉我们，git进行了一些修改，但是这些修改并没有被提交。

4. 查看修改内容。



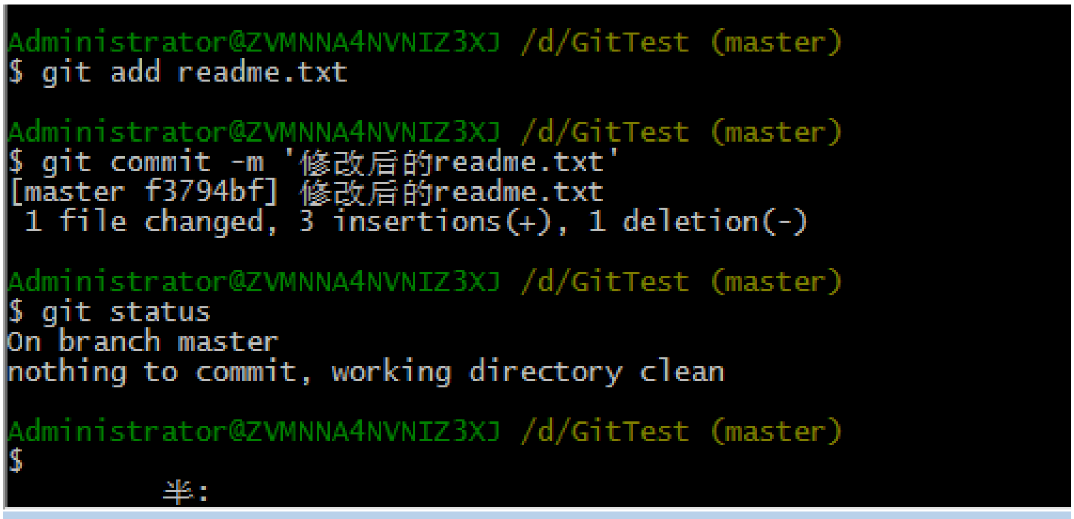
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git diff readme.txt
diff --git a/readme.txt b/readme.txt
index 8615dba..c7363da 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1 +1,3 @@
-1234567
\ No newline at end of file
+1234567
+
+7654321
\ No newline at end of file

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
```



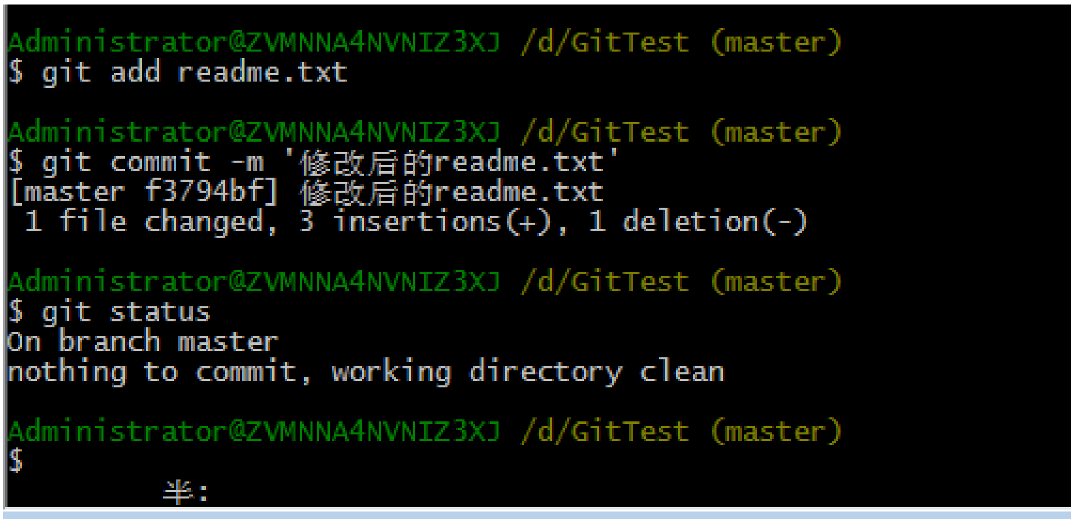
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git diff readme.txt
diff --git a/readme.txt b/readme.txt
index 8615dba..c7363da 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1 +1,3 @@
-1234567
\ No newline at end of file
+1234567
+
+7654321
\ No newline at end of file

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
```

命令1 git diff 你的文件名　 会出现更改信息，发现多了两个换行， 多了7654321内容；

将更改后的东西提交上去，和之前一样。先add 然后再commit 。

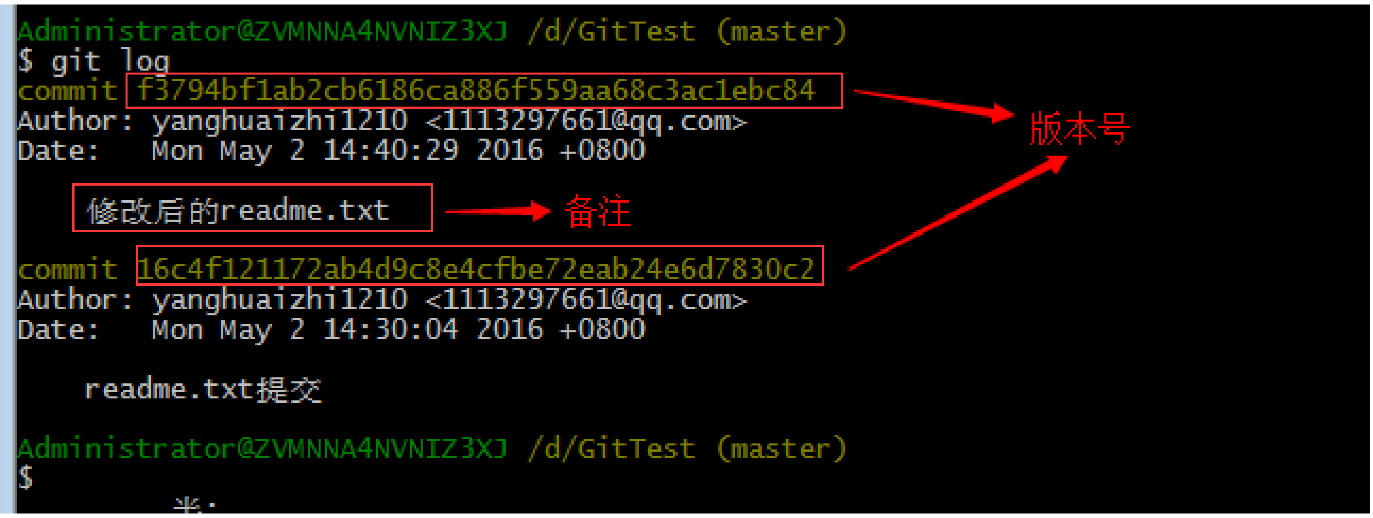


```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git add readme.txt

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git commit -m '修改后的readme.txt'
[master f3794bf] 修改后的readme.txt
 1 file changed, 3 insertions(+), 1 deletion(-)

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git status
On branch master
nothing to commit, working directory clean

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
         半：
```

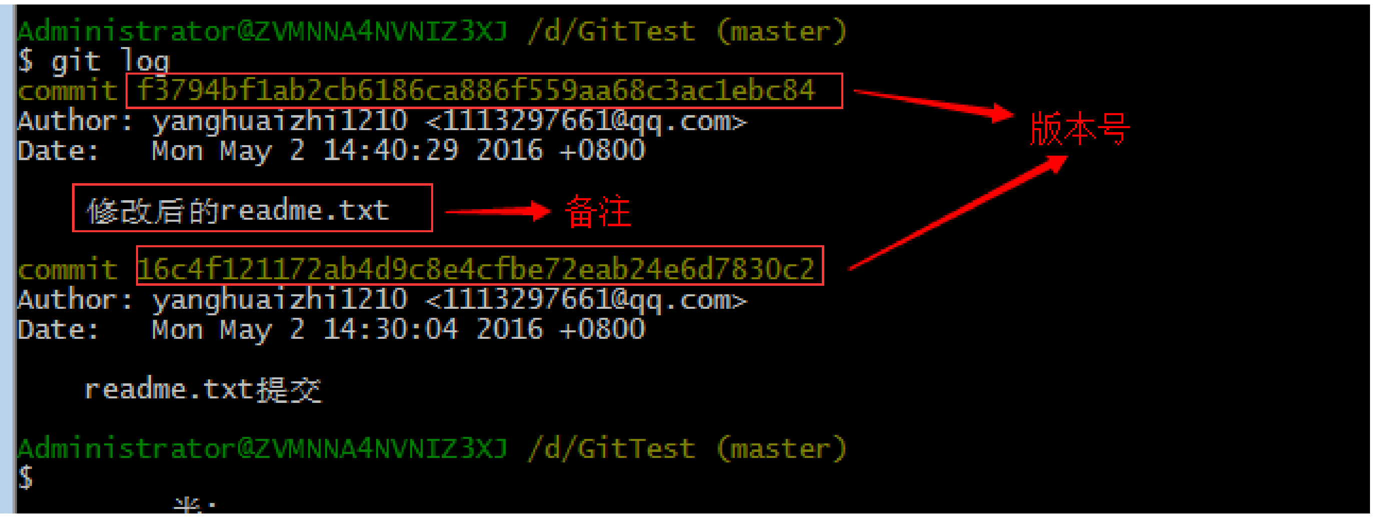


```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git add readme.txt

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git commit -m '修改后的readme.txt'
[master f3794bf] 修改后的readme.txt
 1 file changed, 3 insertions(+), 1 deletion(-)

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git status
On branch master
nothing to commit, working directory clean

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
         半：
```

5. 版本退回



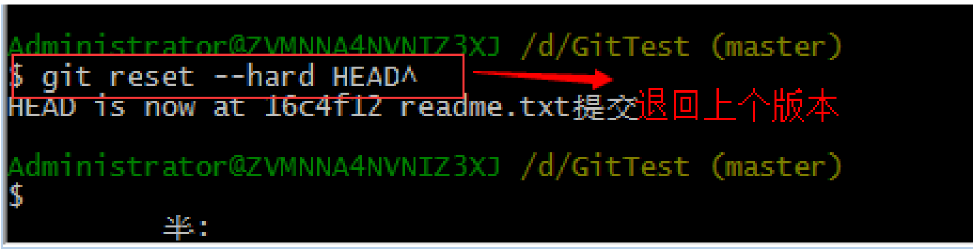
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git log
commit f3794bf1ab2cb6186ca886f559aa68c3ac1ebc84
Author: yanghuaizhi1210 <1113297661@qq.com>
Date:   Mon May 2 14:40:29 2016 +0800

    修改后的readme.txt

commit 16c4f121172ab4d9c8e4cfbe72eab24e6d7830c2
Author: yanghuaizhi1210 <1113297661@qq.com>
Date:   Mon May 2 14:30:04 2016 +0800

    readme.txt提交

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
          半:
```


版本号
备注

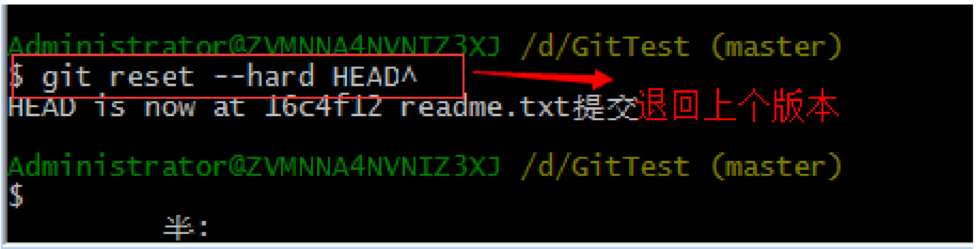
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git log
commit f3794bf1ab2cb6186ca886f559aa68c3ac1ebc84
Author: yanghuaizhi1210 <1113297661@qq.com>
Date:   Mon May 2 14:40:29 2016 +0800

    修改后的readme.txt

commit 16c4f121172ab4d9c8e4cfbe72eab24e6d7830c2
Author: yanghuaizhi1210 <1113297661@qq.com>
Date:   Mon May 2 14:30:04 2016 +0800

    readme.txt提交

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
          半:
```

命令1. git log   （ 修改日志）

退回



```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git reset --hard HEAD^
HEAD is now at 16c4f12 readme.txt提交

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
          半:
```



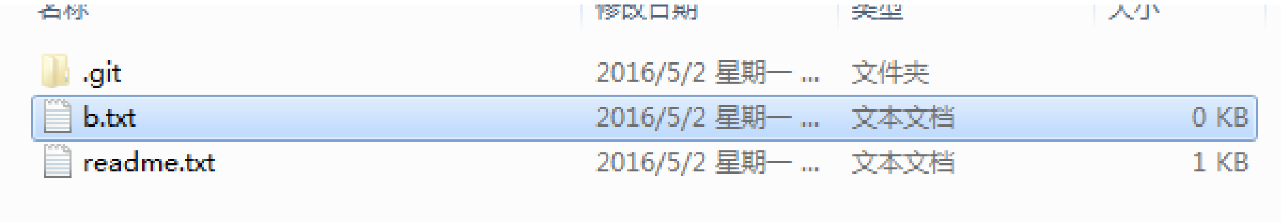
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git reset --hard HEAD^
HEAD is now at 16c4f12 readme.txt提交
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
        半:
```

命令 1 . git reset --hard HEAD^ 退回到上个版本 如果需要退回好多版本就在后面加上 ^ 例： git reset --hard HEAD^^退回两个版本。

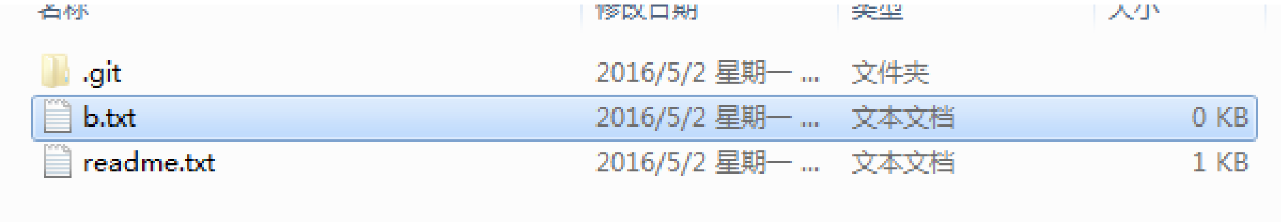
然后去查看readme.txt是否成功退回。

6.删除

新建一个 b.txt

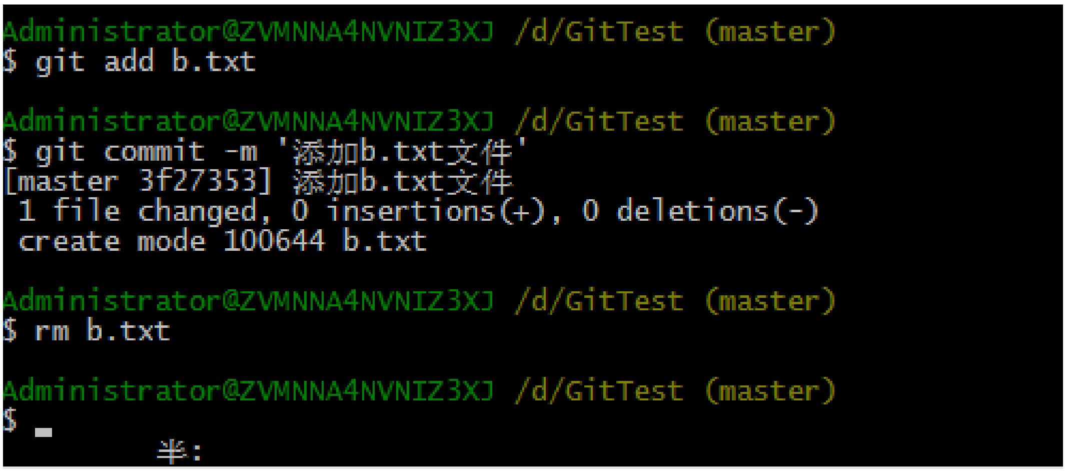


名称	修改日期	类型	大小
.git	2016/5/2 星期一 ...	文件夹	
b.txt	2016/5/2 星期一 ...	文本文档	0 KB
readme.txt	2016/5/2 星期一 ...	文本文档	1 KB

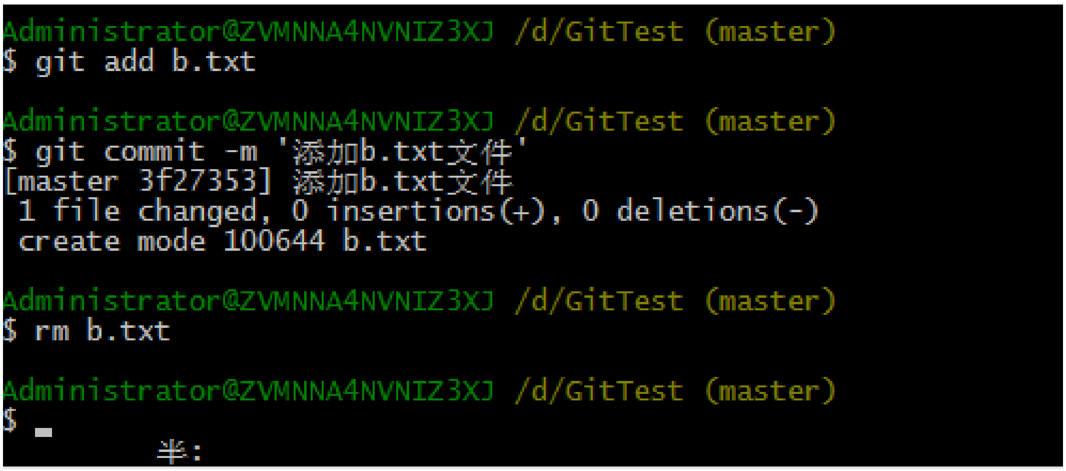


名称	修改日期	类型	大小
.git	2016/5/2 星期一 ...	文件夹	
b.txt	2016/5/2 星期一 ...	文本文档	0 KB
readme.txt	2016/5/2 星期一 ...	文本文档	1 KB

然后commit到库中。 并删除。



```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git add b.txt

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git commit -m '添加b.txt文件'
[master 3f27353] 添加b.txt文件
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 b.txt

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ rm b.txt

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ ▂
         半：
```



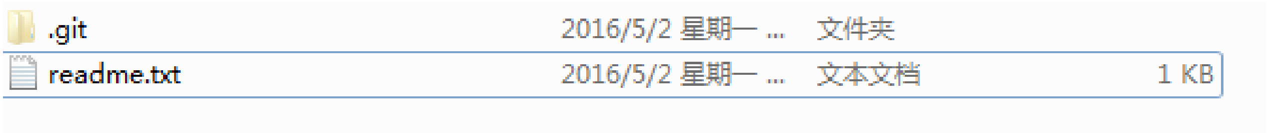
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git add b.txt

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git commit -m '添加b.txt文件'
[master 3f27353] 添加b.txt文件
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 b.txt

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ rm b.txt

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ ▂
         半：
```

命令1. git add

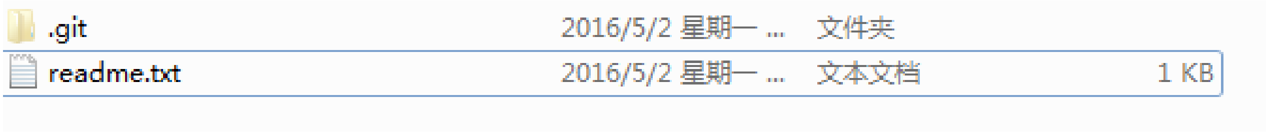
命令2. git commit -m ‘备注’

命令3. rm b.txt　这个是删除b文件的命令

回过头来看文件夹中的b.txt

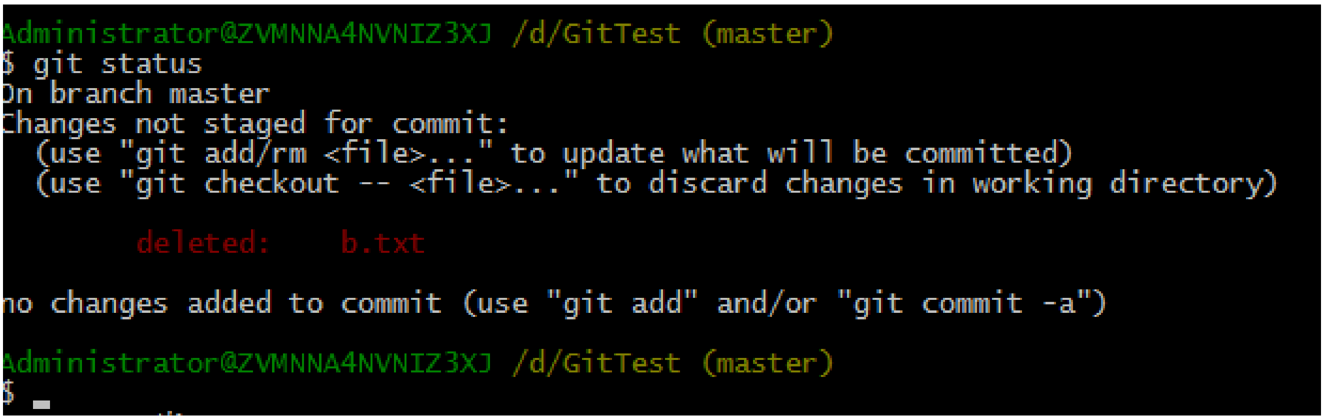


.git	2016/5/2 星期一 ...	文件夹	
readme.txt	2016/5/2 星期一 ...	文本文档	1 KB

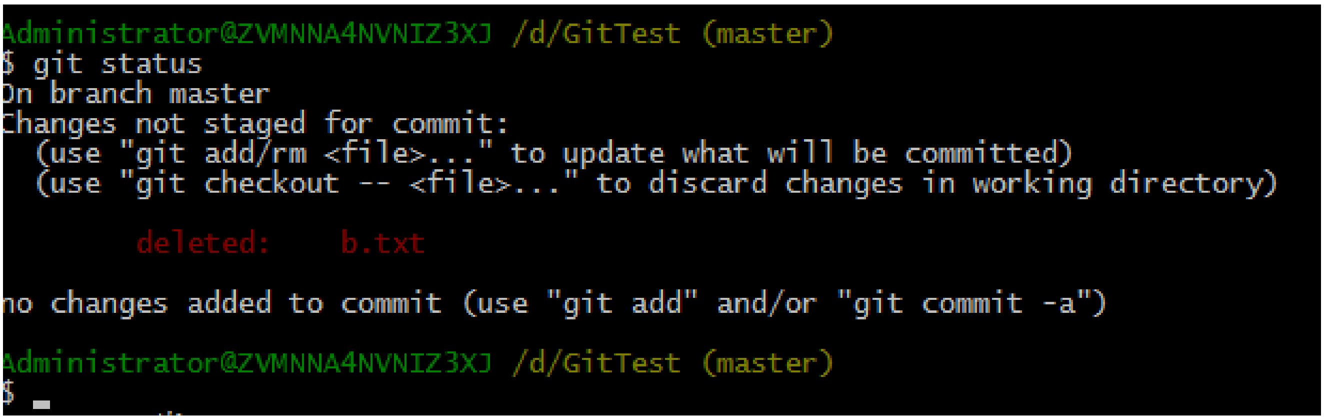


.git	2016/5/2 星期一 ...	文件夹	
readme.txt	2016/5/2 星期一 ...	文本文档	1 KB

b.txt不见了，查看下状态。

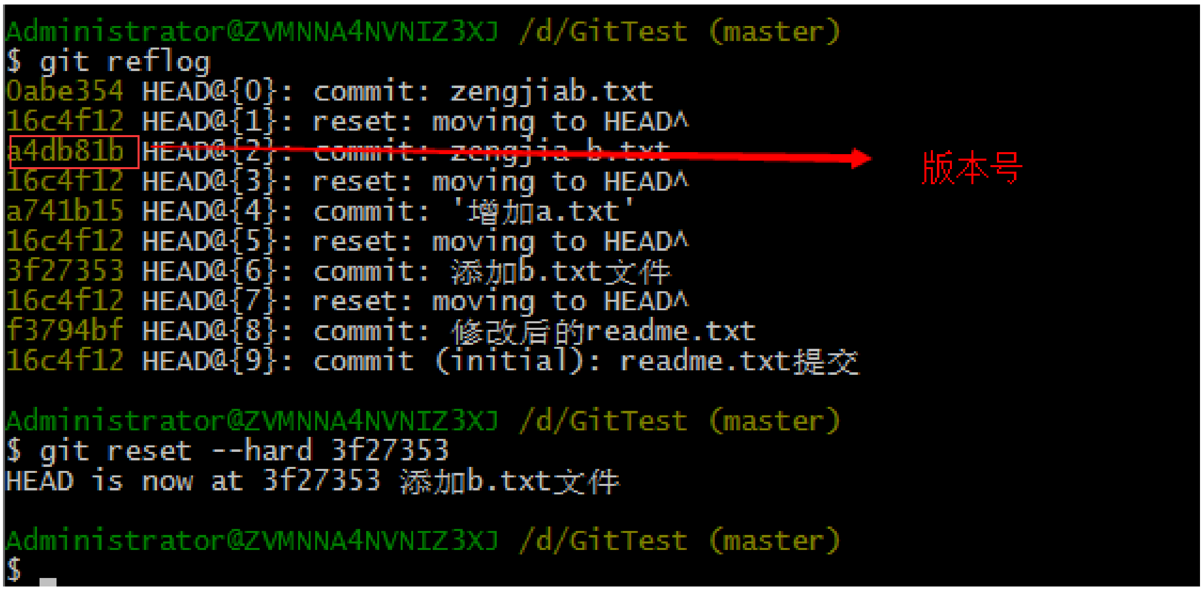


```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    b.txt

no changes added to commit (use "git add" and/or "git commit -a")

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
```



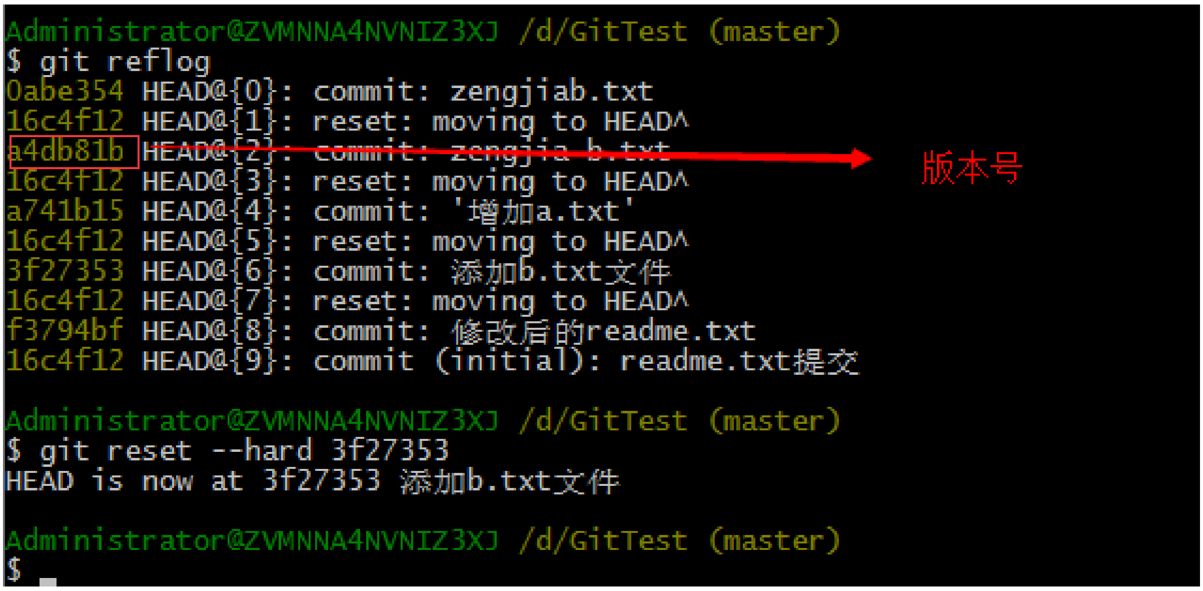
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    b.txt

no changes added to commit (use "git add" and/or "git commit -a")

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
```

可以commit，可以放弃修改　git reset --hard 版本号来回复文件。



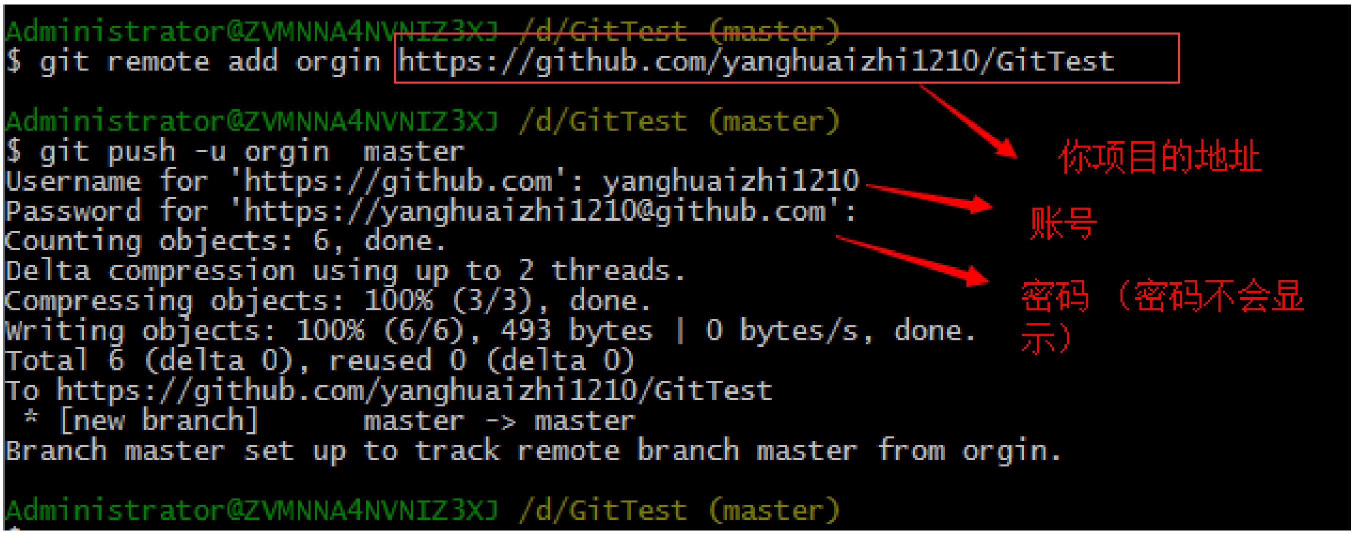
```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git reflog
0abe354 HEAD@{0}: commit: zengjiab.txt
16c4f12 HEAD@{1}: reset: moving to HEAD^
a4db81b HEAD@{2}: commit: zengjia b.txt
16c4f12 HEAD@{3}: reset: moving to HEAD^
a741b15 HEAD@{4}: commit: '增加a.txt'
16c4f12 HEAD@{5}: reset: moving to HEAD^
3f27353 HEAD@{6}: commit: 添加b.txt文件
16c4f12 HEAD@{7}: reset: moving to HEAD^
f3794bf HEAD@{8}: commit: 修改后的readme.txt
16c4f12 HEAD@{9}: commit (initial): readme.txt提交

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git reset --hard 3f27353
HEAD is now at 3f27353 添加b.txt文件

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
```

版本号

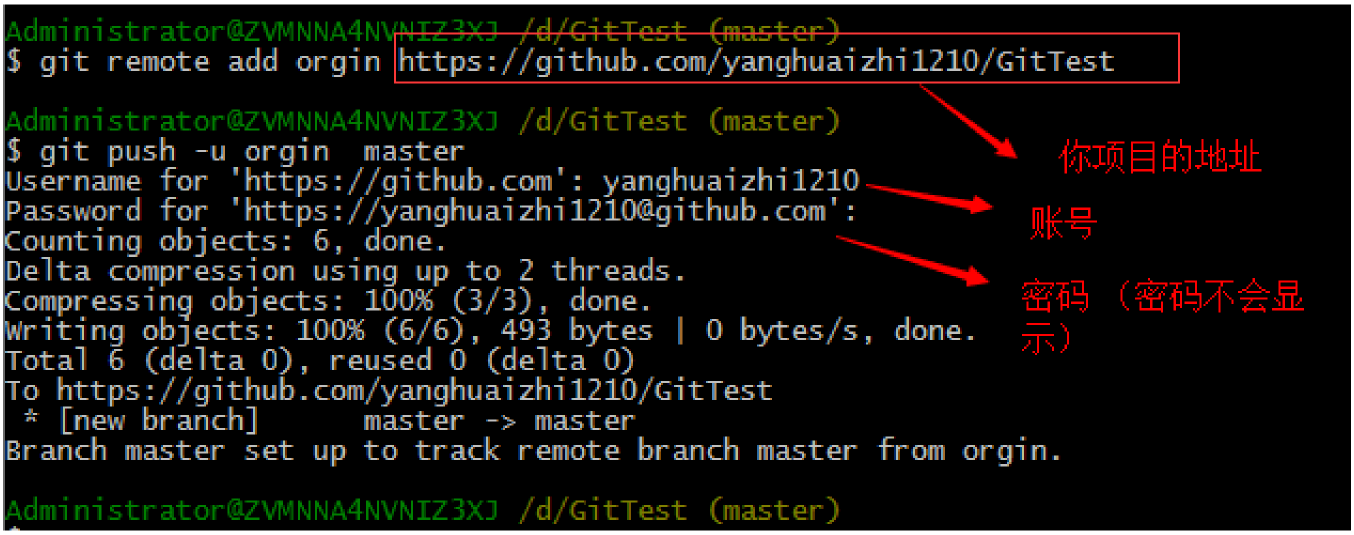


```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git reflog
0abe354 HEAD@{0}: commit: zengjiab.txt
16c4f12 HEAD@{1}: reset: moving to HEAD^
a4db81b HEAD@{2}: commit: zengjia b.txt
16c4f12 HEAD@{3}: reset: moving to HEAD^
a741b15 HEAD@{4}: commit: '增加a.txt'
16c4f12 HEAD@{5}: reset: moving to HEAD^
3f27353 HEAD@{6}: commit: 添加b.txt文件
16c4f12 HEAD@{7}: reset: moving to HEAD^
f3794bf HEAD@{8}: commit: 修改后的readme.txt
16c4f12 HEAD@{9}: commit (initial): readme.txt提交

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git reset --hard 3f27353
HEAD is now at 3f27353 添加b.txt文件

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$
```

然后连接本地仓库，并将自己的库push到服务器中去。



```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git remote add orgin https://github.com/yanghuaizhi1210/GitTest

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git push -u orgin  master
Username for 'https://github.com': yanghuaizhi1210
Password for 'https://yanghuaizhi1210@github.com':
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 493 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/yanghuaizhi1210/GitTest
 * [new branch]      master -> master
Branch master set up to track remote branch master from orgin.

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
```

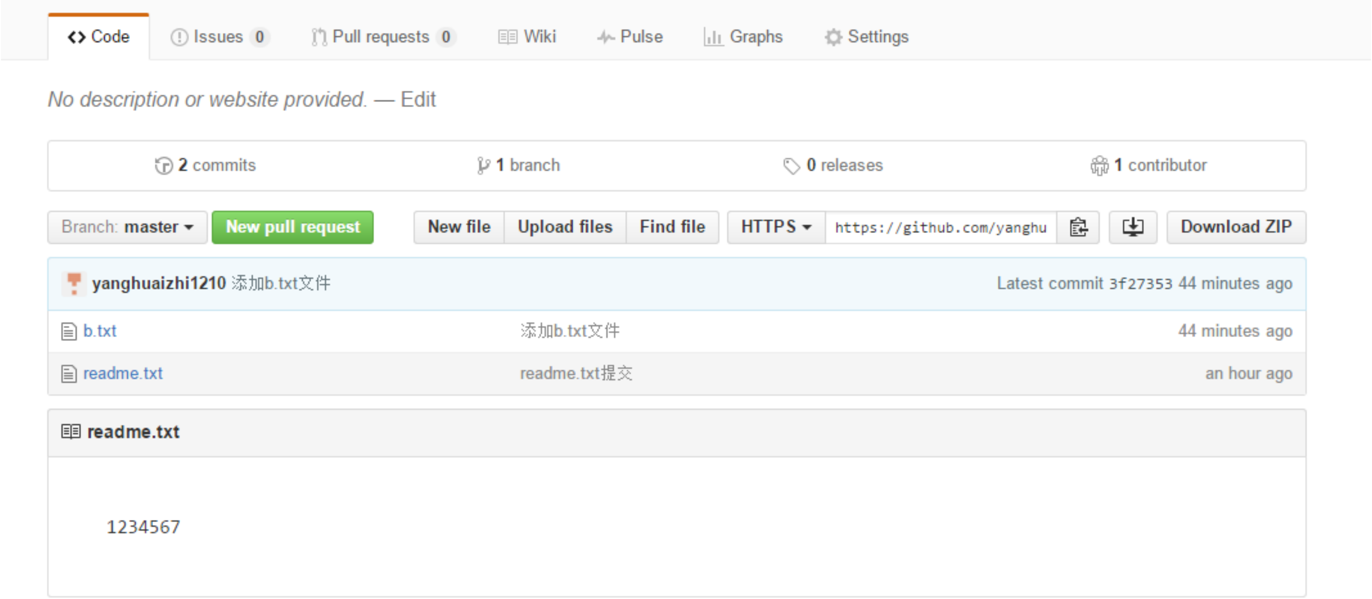


```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git remote add orgin https://github.com/yanghuaizhi1210/GitTest

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git push -u orgin  master
Username for 'https://github.com': yanghuaizhi1210
Password for 'https://yanghuaizhi1210@github.com':
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 493 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/yanghuaizhi1210/GitTest
 * [new branch]      master -> master
Branch master set up to track remote branch master from orgin.

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
```

命令1. git remote add origin https: // github.com/你的账号/你的项目名称

命令2. git push -u origin master 将本地的库推送到master分支 （就是推送到服务器上）

在服务器上查看是否推送成功。



No description or website provided. — Edit

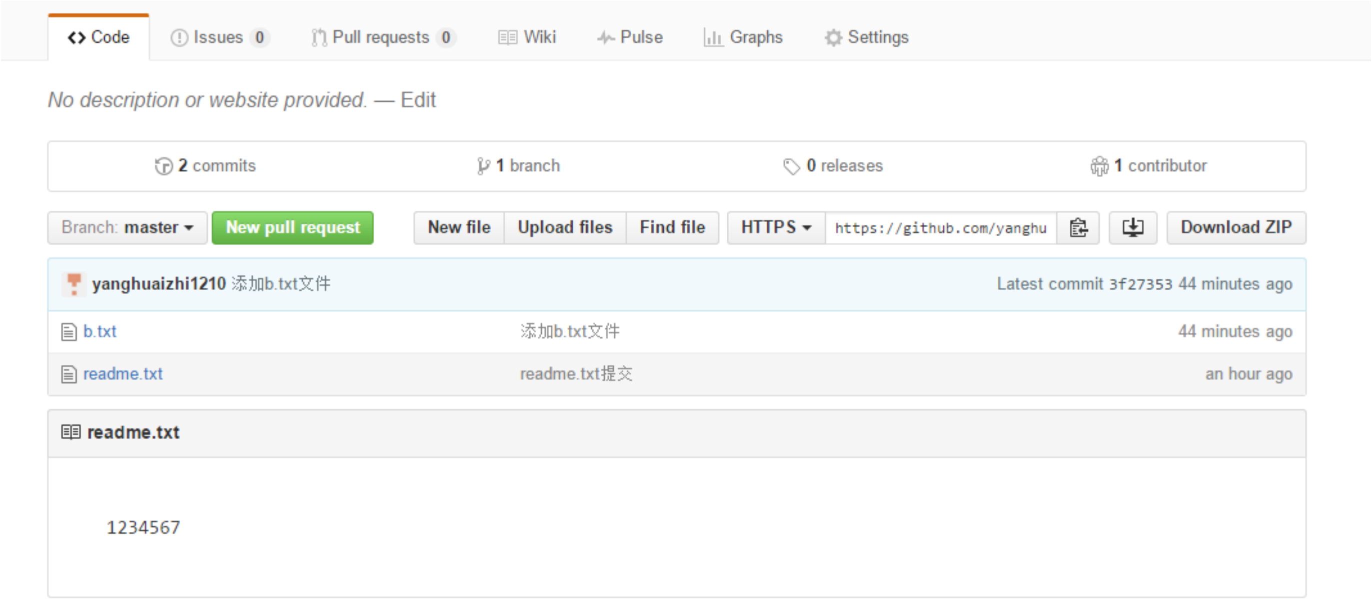
2 commits | 1 branch | 0 releases | 1 contributor

yanghuaizhi1210 添加b.txt文件 Latest commit 3f27353 44 minutes ago

b.txt	添加b.txt文件	44 minutes ago
readme.txt	readme.txt提交	an hour ago

readme.txt

1234567



<> Code
Issues 0
Pull requests 0
Wiki
Pulse
Graphs
Settings

No description or website provided. — Edit

2 commits | 1 branch | 0 releases | 1 contributor

Branch: master
New pull request
New file
Upload files
Find file
HTTPS
https://github.com/yanghu
Download ZIP

yanghuaizhi1210 添加b.txt文件 Latest commit 3f27353 44 minutes ago

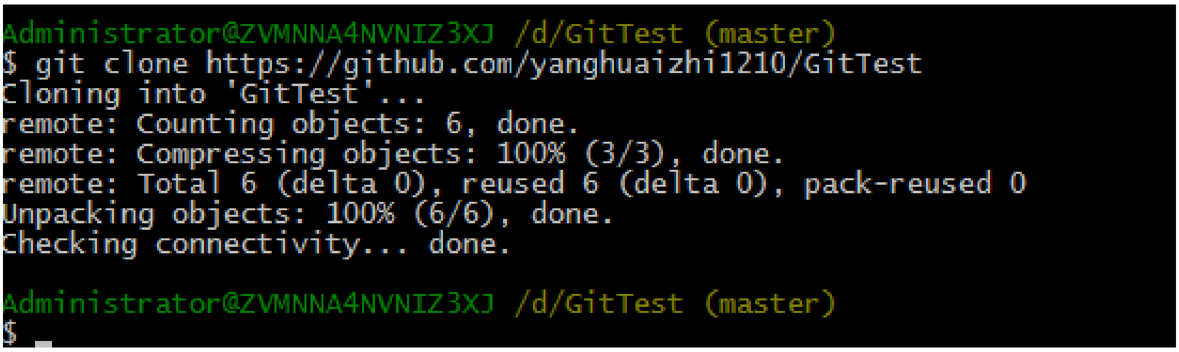
b.txt 添加b.txt文件 44 minutes ago
readme.txt readme.txt提交 an hour ago

readme.txt

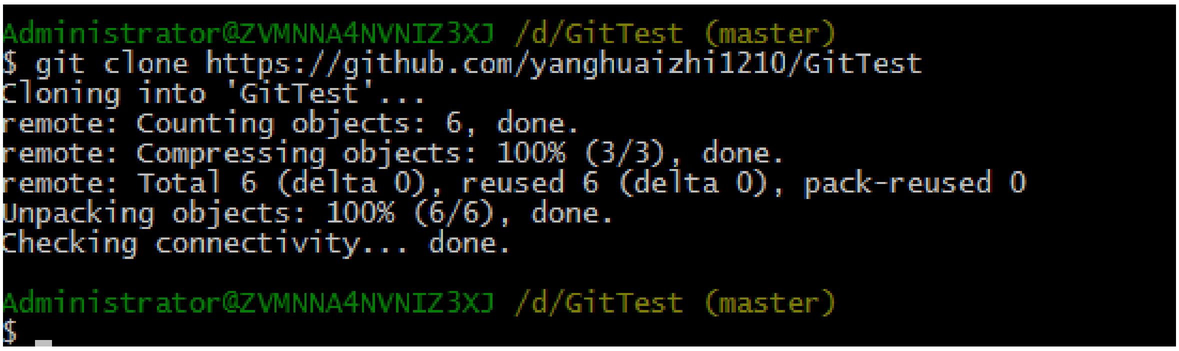
1234567

从现在开始，只需要git push origin master就可以吧本机的推送过去了。

从服务器拉去数据， gitclone



```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git clone https://github.com/yanghuaizhi1210/GitTest
Cloning into 'GitTest'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
Checking connectivity... done.

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ _
```



```
Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ git clone https://github.com/yanghuaizhi1210/GitTest
Cloning into 'GitTest'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
Checking connectivity... done.

Administrator@ZVMNNA4NVNIZ3XJ /d/GitTest (master)
$ _
```

命令1. git clone https://github.com/yanghuaizhi1210/GitTest

将服务器端的数据克隆过来。

如果输入$ git remote add origin git@github.com:djqiang（github帐号名）/gitdemo（项目名）.git

提示出错信息：fatal: remote origin already exists.

解决办法如下：

1、先输入$ git remote rm origin

2、再输入$ git remote add origin git@github.com:djqiang/gitdemo.git 就不会报错了！

3、如果输入$ git remote rm origin 还是报错的话，error: Could not remove config section 'remote.origin'. 我们需要修改gitconfig文件的内容

4、找到你的github的安装路径，我的是C:\Users\ASUS\AppData\Local\GitHub\PortableGit_ca477551eeb4aea0e4ae9fcd3358bd96720bb5c8\etc

5、找到一个名为gitconfig的文件，打开它把里面的[remote "origin"]那一行删掉就好了！

模块化

什么是模块化？

是一种项目的构架模式， 这种构架模式让JS代码重用性变得非常高，让项目构架的一些复杂问题全部得以解决。 例如，多个script标签不会再出现了，我们只要用一个script标签进行引入就可以了。

模块化那有两个规范，AMD规范和CMD规范，当然还有针对于ndoejs后台的commonJS规范;

这些规范目标一致都是为项目实现模块化，期中有遵循AMD规范的 require.js 和遵循CMD规范的sea.js 。 先说先这两种规范的不同;

1. 首先对于依赖：

AMD提前执行(不管有没有用， 先加载过来) //特点 : 前期消耗网络资源大， 但是后期运行效率高.

CMD需求执行(有用我再去加载) //特点: 整个自选消耗曲线比较平缓。

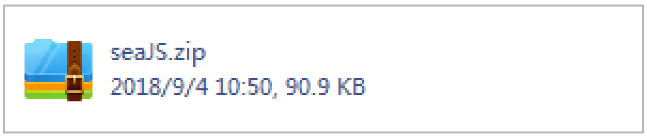
2. CMD 推崇依赖就近，AMD 推崇依赖前置。看代码：

```
// CMD
define(function(require, exports, module) {
var a = require('./a')
a.doSomething()
// 此处略去 100 行
var b = require('./b') // 依赖可以就近书写
b.doSomething()
// ...
})

// AMD 默认推荐的是
define(['./a', './b'], function(a, b) { // 依赖必须一开始就写好
a.doSomething()
// 此处略去 100 行
b.doSomething()
...
})
```

3. AMD 的 API 默认是一个当多个用，CMD 的 API 严格区分，推崇职责单一。比如 AMD 里，require 分全局 require 和局部 require，都叫 require。CMD 里，没有全局 require，而是根据模块系统的完备性，提供 seajs.use 来实现模块系统的加载启动。CMD 里，每个 API 都简单纯粹。

require.zip

2018/9/4 10:50, 21.1 KB



seaJS.zip

2018/9/4 10:50, 90.9 KB