

GitHub

一、为什么要引入Git?

1) 用简单一点的例子来讲：

- 一个公司有几个甚至十几个人共同合作开发一款项目的时候，公司对项目代码有明确规范，不可能再像学校做的小项目一样，你觉得哪个同学的代码写的不好，或者和你的代码有冲突，‘Duang’直接给别人改掉或删掉，也就是想怎么改就怎么改。这种情况对于十几个人合作的项目来说肯定是不行的。这个时候就需要版本的控制

• 2) 版本控制：

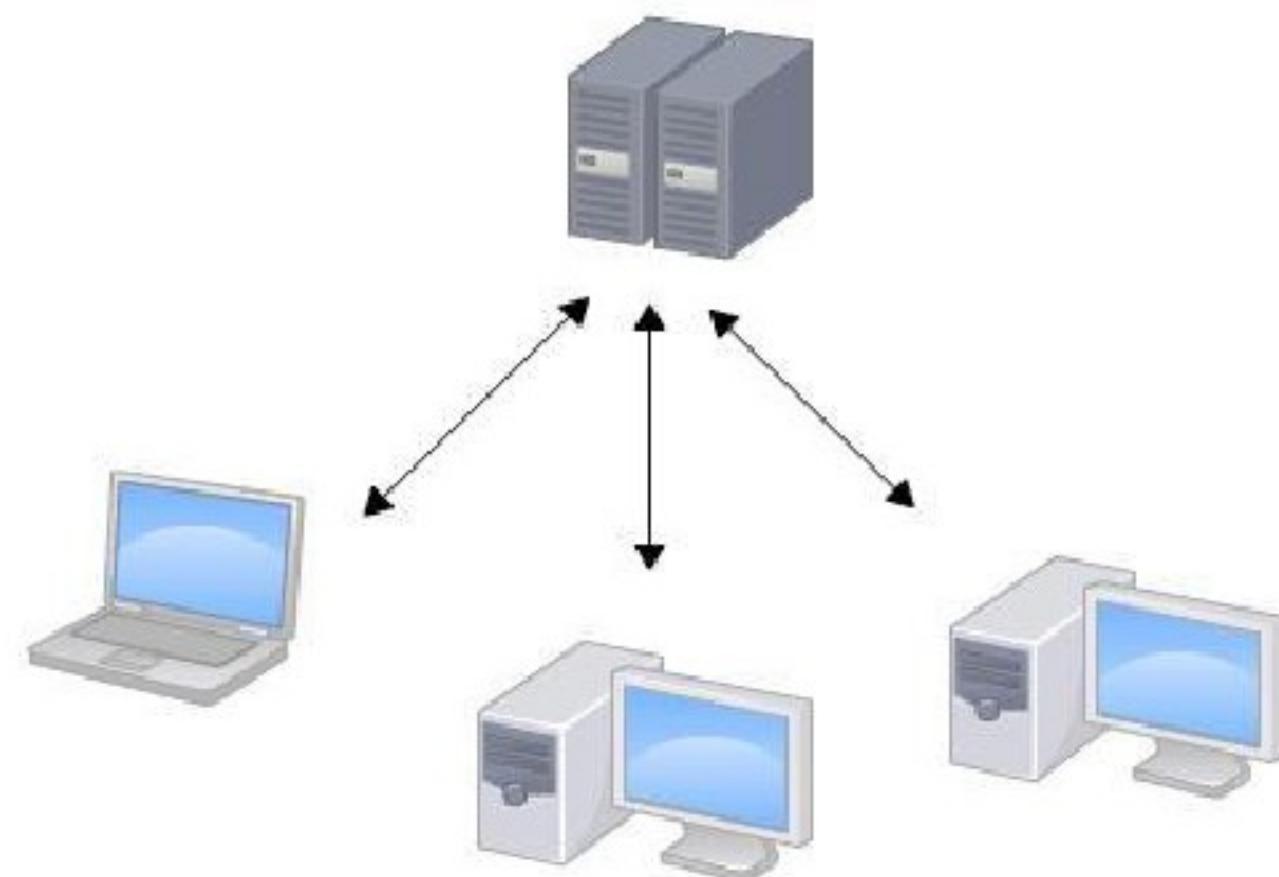
- 通俗来说：你可以在这个版本管理工具上清楚的看到每次不同的人在同一项目上所做的修改说明，并且可以选择下载你需要的版本下来再进行维护。

版本控制：

- 1、版本控制分为集中式版本控制和分布式版本控制
- 1) 集中式的版本控制：

Linus一直痛恨的CVS及SVN都是集中式的版本控制系统，版本库是集中存放在中央服务器的，而干活的时候，用的都是自己的电脑，所以要先从中央服务器取得最新的版本，然后开始干活，干完了，再把自己的活推送给中央服务器。中央服务器就好比是一个图书馆，你要改一本书，必须先从图书馆借出来，然后回到家自己改，改完了，再放回图书馆。

- 集中式版本控制系统最大的毛病就是**必须联网**才能工作，如果在局域网内还好，带宽够大，速度够快，可如果在互联网上，遇到网速慢的话，可能提交一个10M的文件就需要5分钟，这还不得把人给憋死啊。



- 分布式版本控制系统：

其根本**没有“中央服务器”**，每个人的电脑上都是一个完整的版本库，这样，你**工作的时候，就不需要联网了**，因为版本库就在你自己的电脑上。既然**每个人电脑上都有一个完整的版本库**，那多人如何协作呢？比方说你在自己电脑上改了文件A，你的同事也在他的电脑上改了文件A，这时，你们俩之间只需把各自的修改推送给对方，就可以互相看到对方的修改了。

2、分布式版本控制和集中式版本控制区别：

- 1) **分布式版本控制系统**的安全性要高很多，因为每个人电脑里都有完整的版本库，某一个人的电脑坏了不要紧，随便从其他人那里复制一个就可以了。
- 2) **集中式版本控制系统**的中央服务器要是出了问题，所有人都没法干活了。

二、GitHub 和 Git

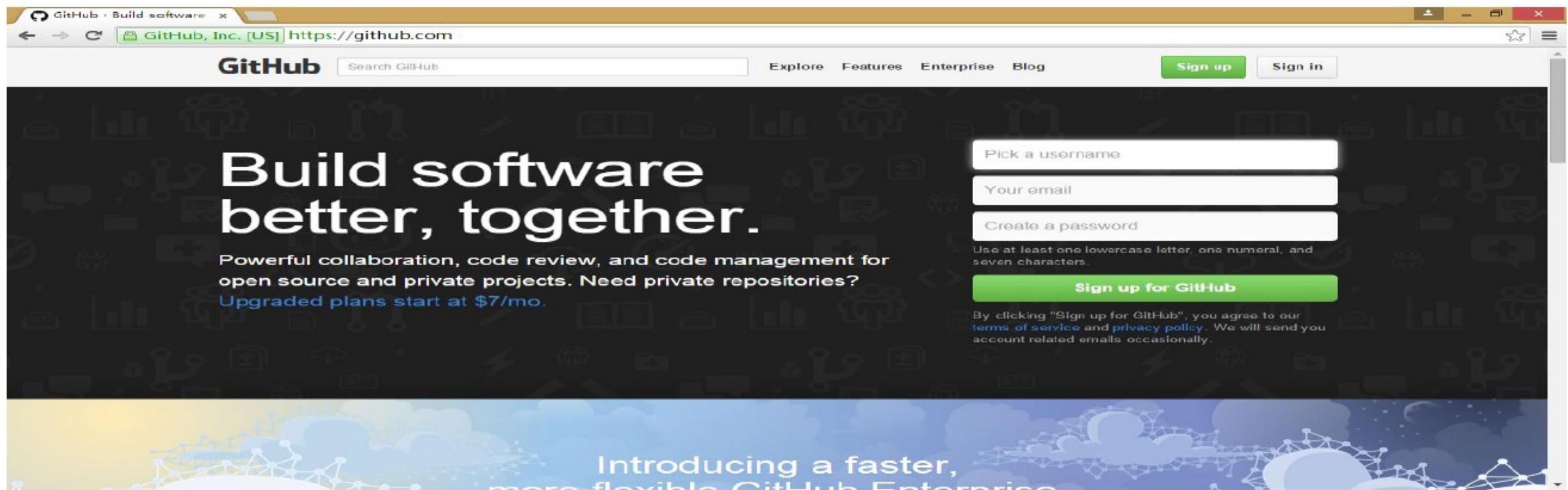
- 1、**Git**: 它是一款**免费、开源的分布式版本控制系统**，他是著名的 Linux 发明者 Linus Torvalds 开发的。它是用来进行版本控制的，就是用来**保存项目的地方**。但是项目要是运行，还是需要你本地的环境，它只不过是用来保存代码罢了。
- 2、**GitHub**: 主要提供基于 **git** 的**版本托管服务**。也就是说现在 GitHub 上托管的所有项目代码都是基于 **Git** 来进行版本控制的，所以 **Git 只是 GitHub 上用来管理项目的一个工具而已**，GitHub 的功能可远不止于此！

三、GitHub如何操作？

- 1、可以通过客户端进行代码提交，更新。
- 2、也可以通过指令来进行操作。

四、GitHub网站及简单的GitHub使用步骤：

- 1) 首先当然是注册一个账户，点击 Sign Up注册



2) 分别填入昵称,邮箱,登录密码,确认密码

The screenshot shows the GitHub 'Join GitHub' sign-up page. At the top, there's a navigation bar with the GitHub logo, 'Explore', 'Features', 'Enterprise', 'Blog', 'Sign up' (in a green button), and 'Sign in'. Below the navigation, the title 'Join GitHub' is displayed, followed by the subtitle 'The best way to design, build, and ship software.' A progress bar at the top indicates three steps: 'Step 1: Set up a personal account' (highlighted with a red box), 'Step 2: Choose your plan' (grayed out), and 'Step 3: Go to your dashboard' (grayed out). The main form area is titled 'Create your personal account'. It contains four input fields: 'Username' (redacted), 'Email Address' (redacted), 'Password' (redacted), and 'Confirm your password' (redacted). To the right of the form is a sidebar titled 'You'll love GitHub' which lists 'Unlimited collaborators', 'Unlimited public repositories', and three benefits: 'Great communication', 'Friction-less development', and 'Open source community', each preceded by a green checkmark.

Join GitHub
The best way to design, build, and ship software.

Step 1: Set up a personal account Step 2: Choose your plan Step 3: Go to your dashboard

Create your personal account

Username:

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

Email Address:

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

Password:

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

Confirm your password:

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

Create an account

3) 选择Free(默认)完成注册

The screenshot shows the GitHub sign-up interface. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for Explore, Gist, Blog, and Help. A user icon labeled "YIFer" is also present. Below the navigation, the main heading is "Welcome to GitHub" with the subtext "You've taken your first step into a larger world, @YIFer.". There are three progress steps: "Completed Set up a personal account" (green checkmark), "Step 2: Choose your plan" (blue square with a person icon), and "Step 3: Go to your dashboard" (grey circle with a checkmark).

The central part of the page is titled "Choose your personal plan". It displays a table of five plan options:

Plan	Cost (view in CNY)	Private repos	Action
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

To the right of the table, a box lists "Each plan includes:" with the following features:

- Unlimited collaborators
- Unlimited public repositories
- ✓ Free setup
- ✓ SSL Protection
- ✓ Email support
- ✓ Wikis, Issues, Pages, & more

Below the table, a note states: "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."

At the bottom left, there's a checkbox for "Help me set up an organization next" with explanatory text and a link to "Learn more about organizations". A large green button at the bottom right says "Finish sign up".

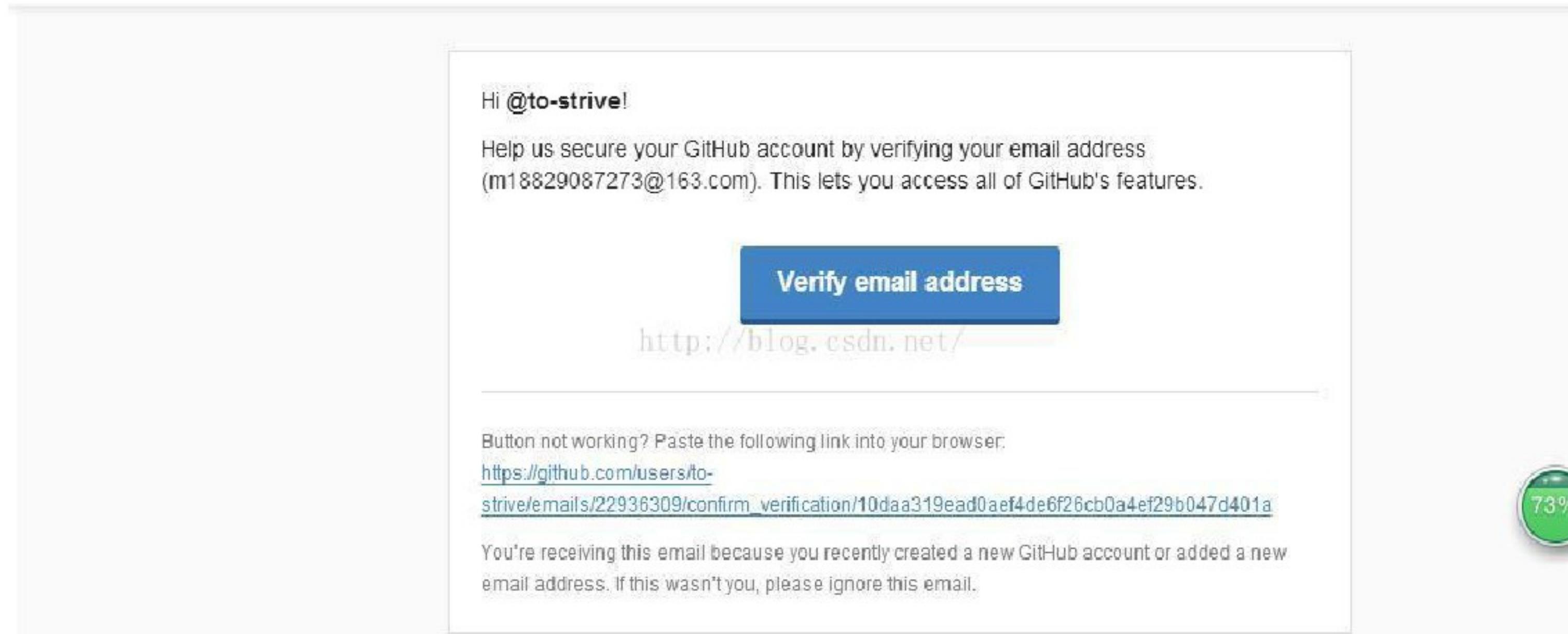


4) 点击 Finish sign up 完成注册，直接跳转到邮箱验证(非常重 要)

The screenshot shows the GitHub Personal settings page with the 'Email' tab selected. The left sidebar lists 'Personal settings' with options: Profile, Account settings, **Emails**, Notification center, Billing, SSH keys, Security, Applications, Repositories, and Organizations. The 'Emails' option is highlighted with an orange border. The main content area is titled 'Email' and contains the following text:
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).
You have not verified any email addresses. We recommend you verify at least one email for password recovery.
A list of emails shows one entry: yif422@163.com, labeled as Primary, Public, and Unverified. To the right of this list are two buttons: 'Send verification email' and a trash can icon. Below this, there is a form field labeled 'Add email address' with an 'Add' button next to it.



5) 点击Send verification email,GitHub会发送验证邮件到你的邮箱,进入你的邮箱,按照默认链接激活注册。注意这一步非常重要,如果不激活邮箱验证,后面的操作将都不会产生效果

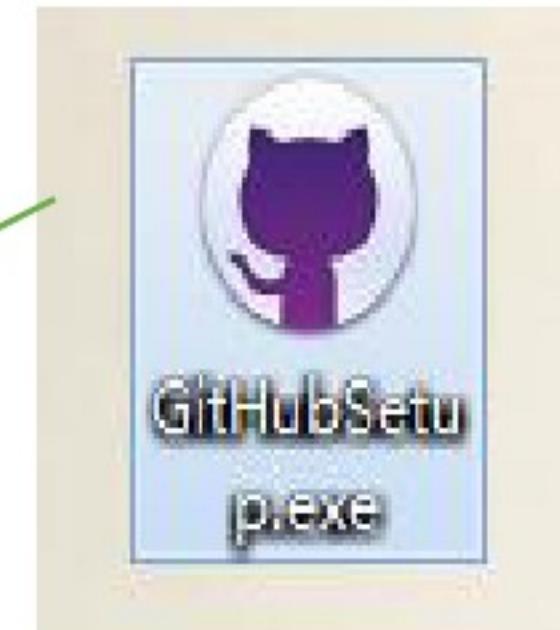


六、GitHub for Windows

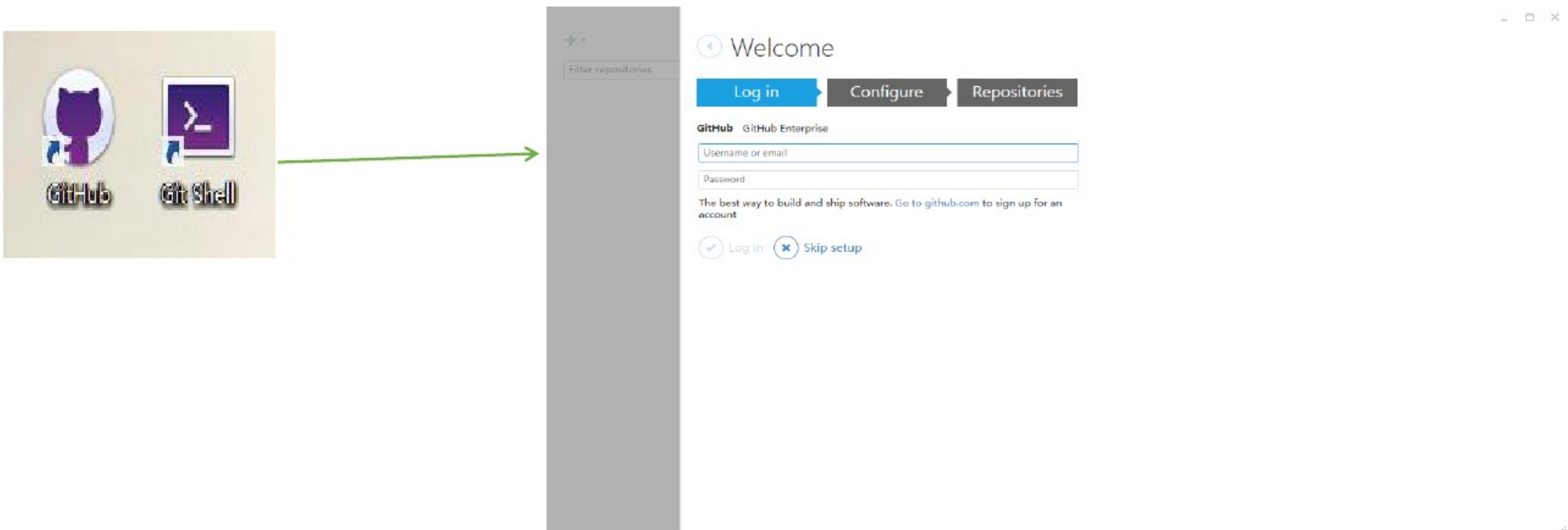
- 1、下载安装
- 1) 下载地址:<https://windows.github.com/>



2) 双击进行安装



3) .安装好之后桌面出现下面的图标,**GitHub**就傻瓜的图形界面,**Git Shell**是命令行的版本,我们这里直接使用图形界面就ok 如果第一次进入软件,会直接提示让你登录,输入你刚刚在**GitHub**注册的账户即可

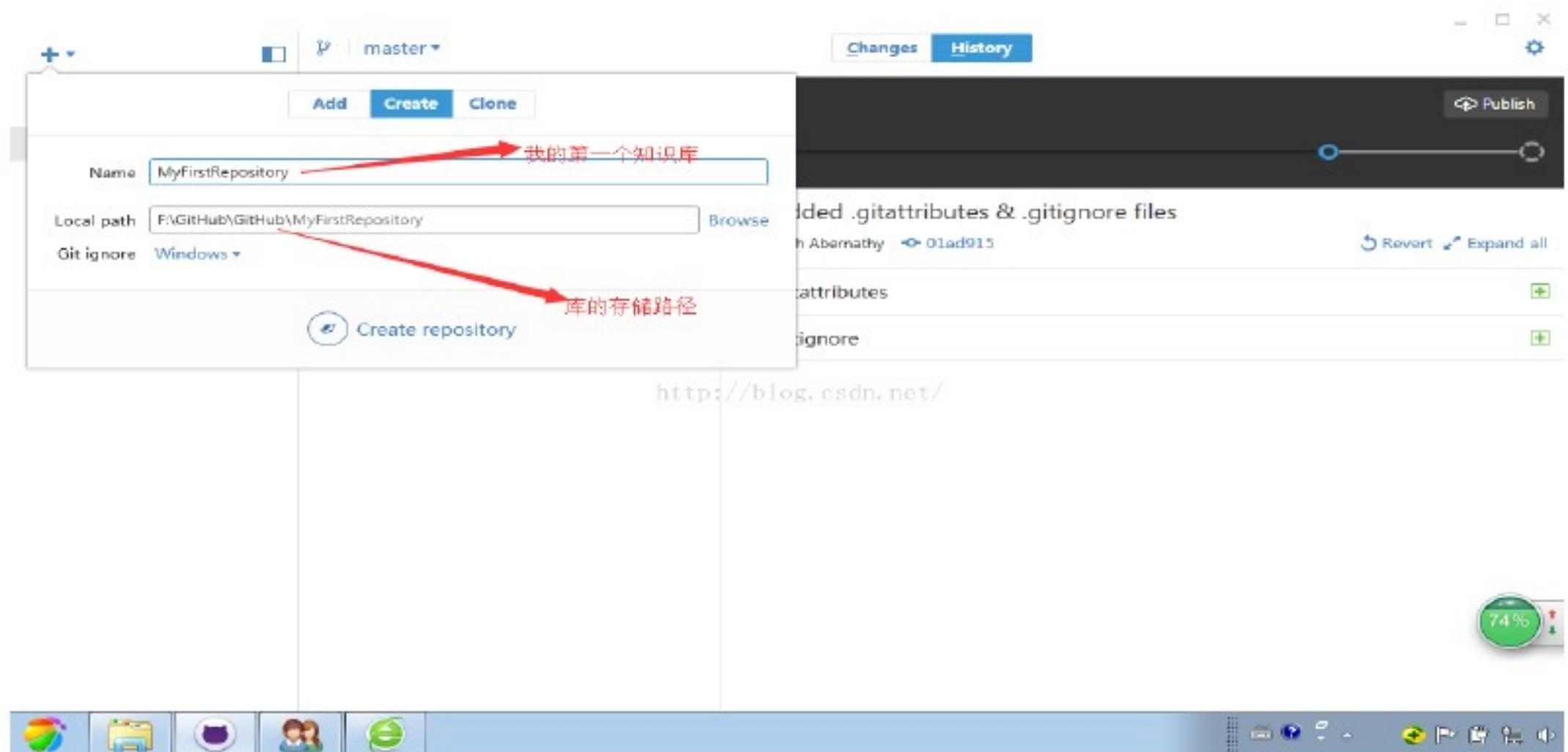


5) 登录之后新建仓库

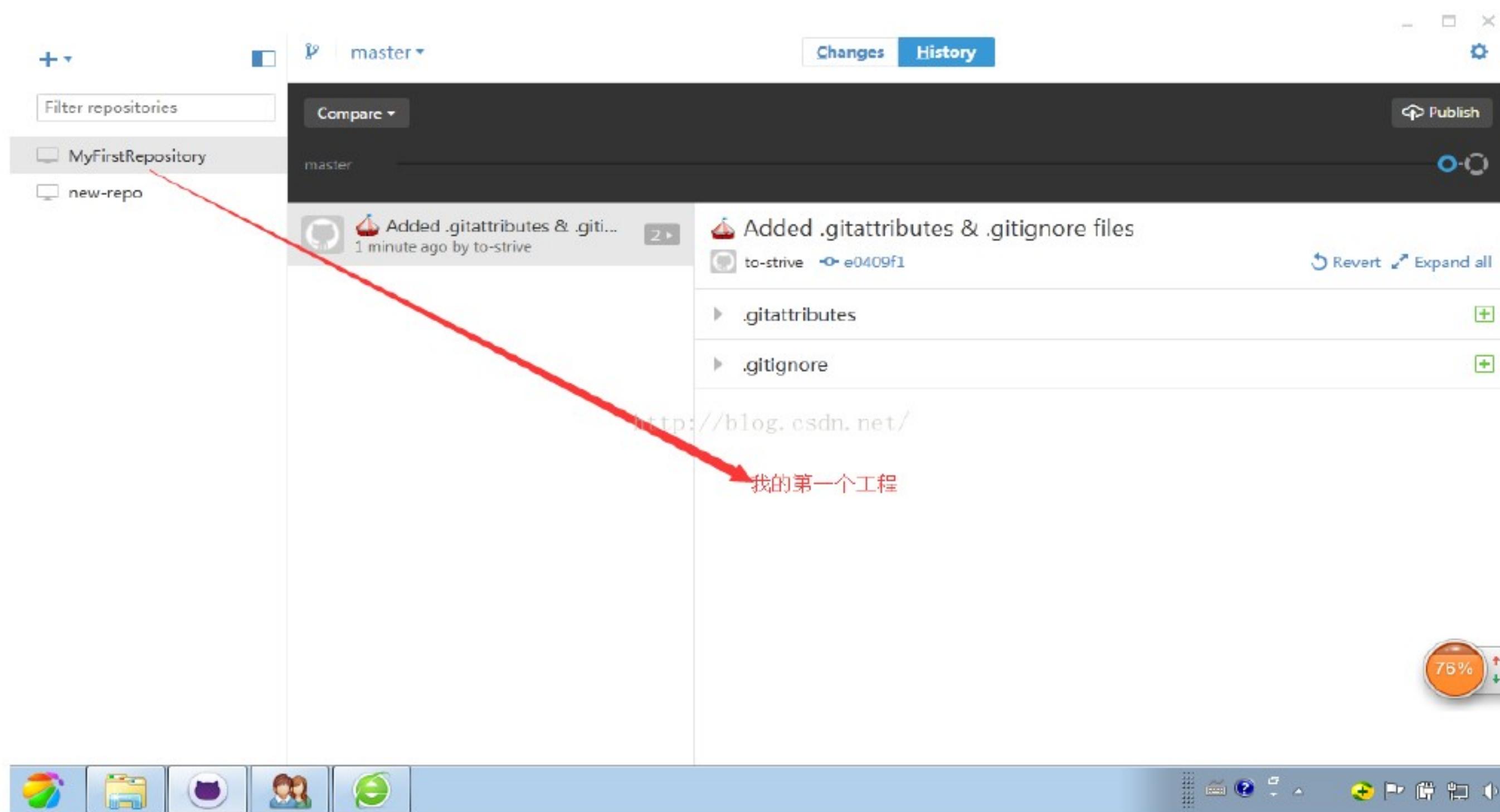
- 点击左上角：



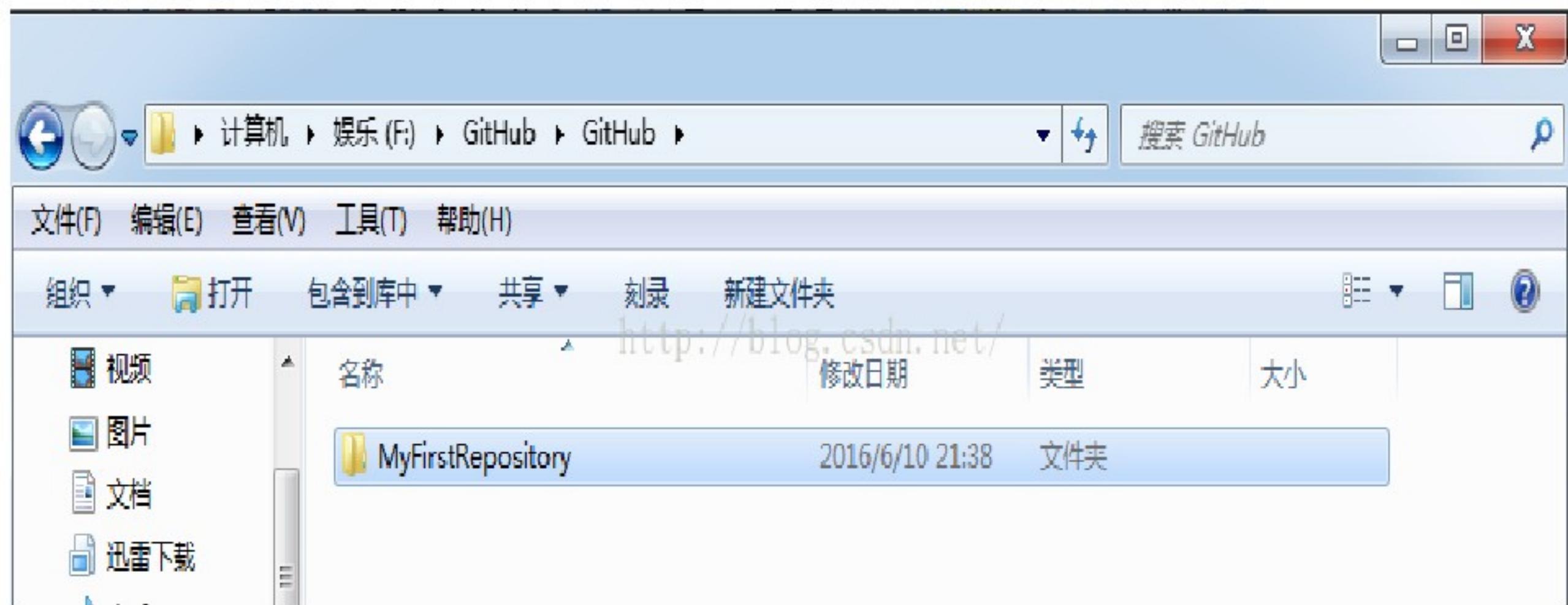
进行新建仓库，如下图：



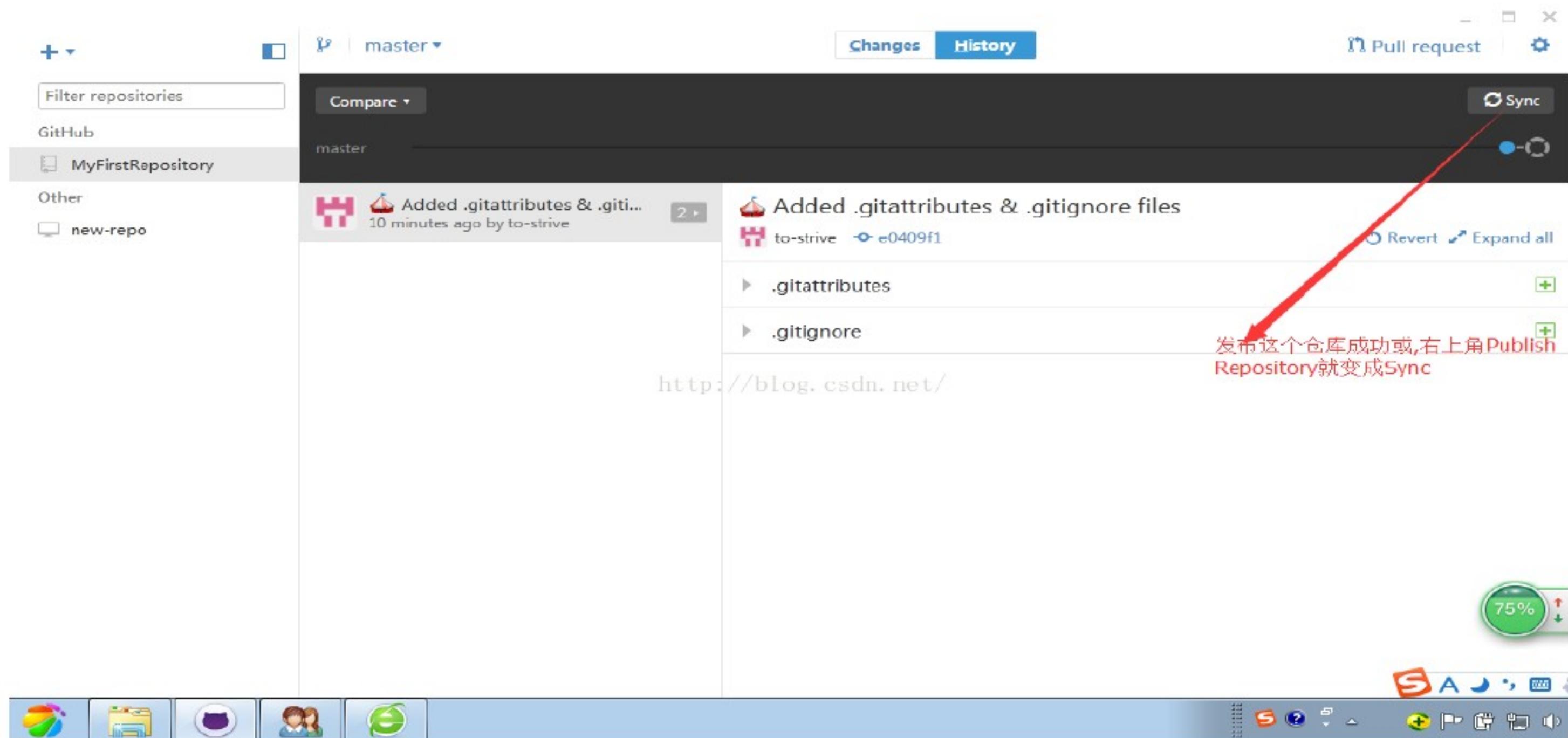
由上面创建好之后,如下图:



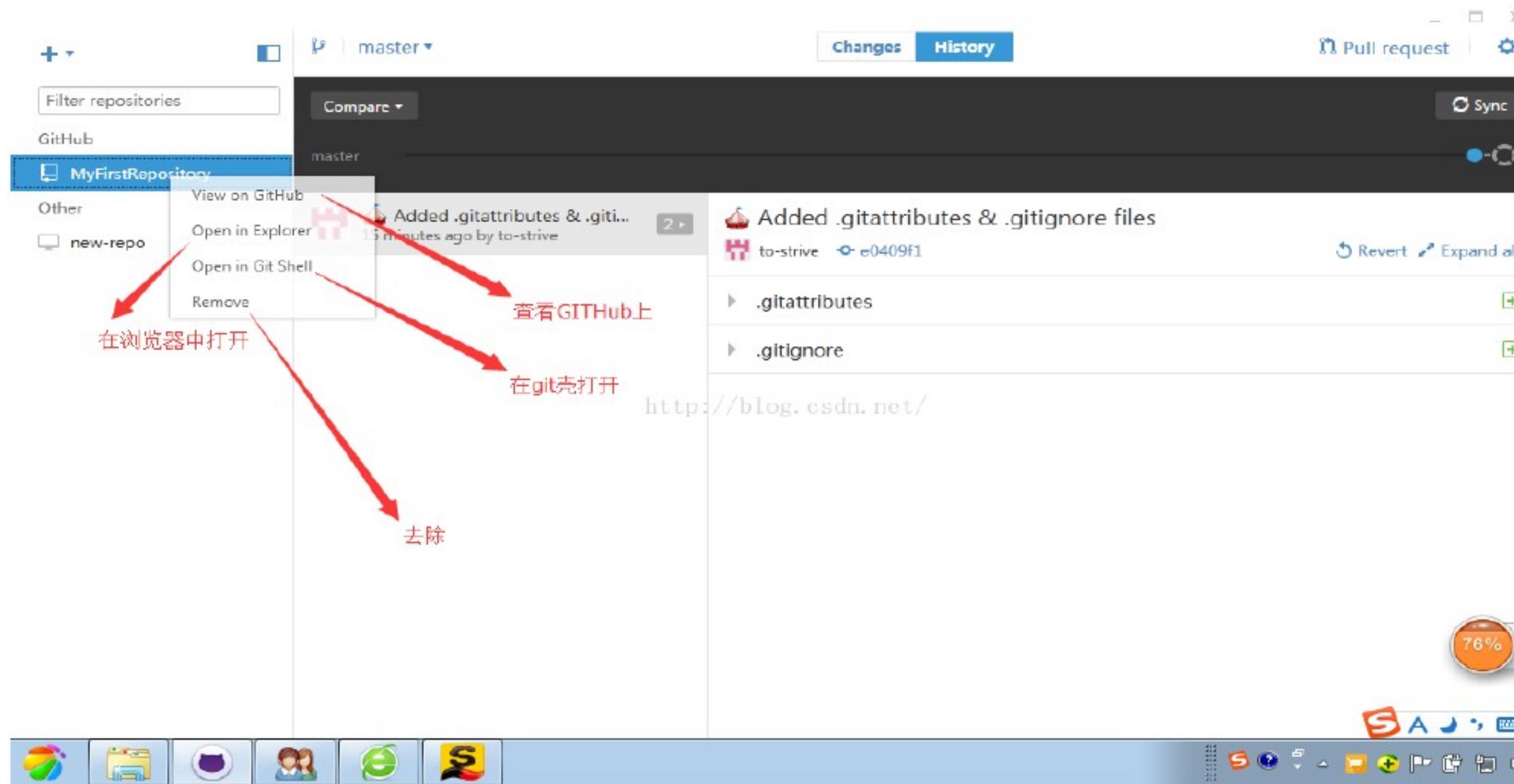
表示这个一个全新的Repository,在自己的电脑上也出现了相应的目录



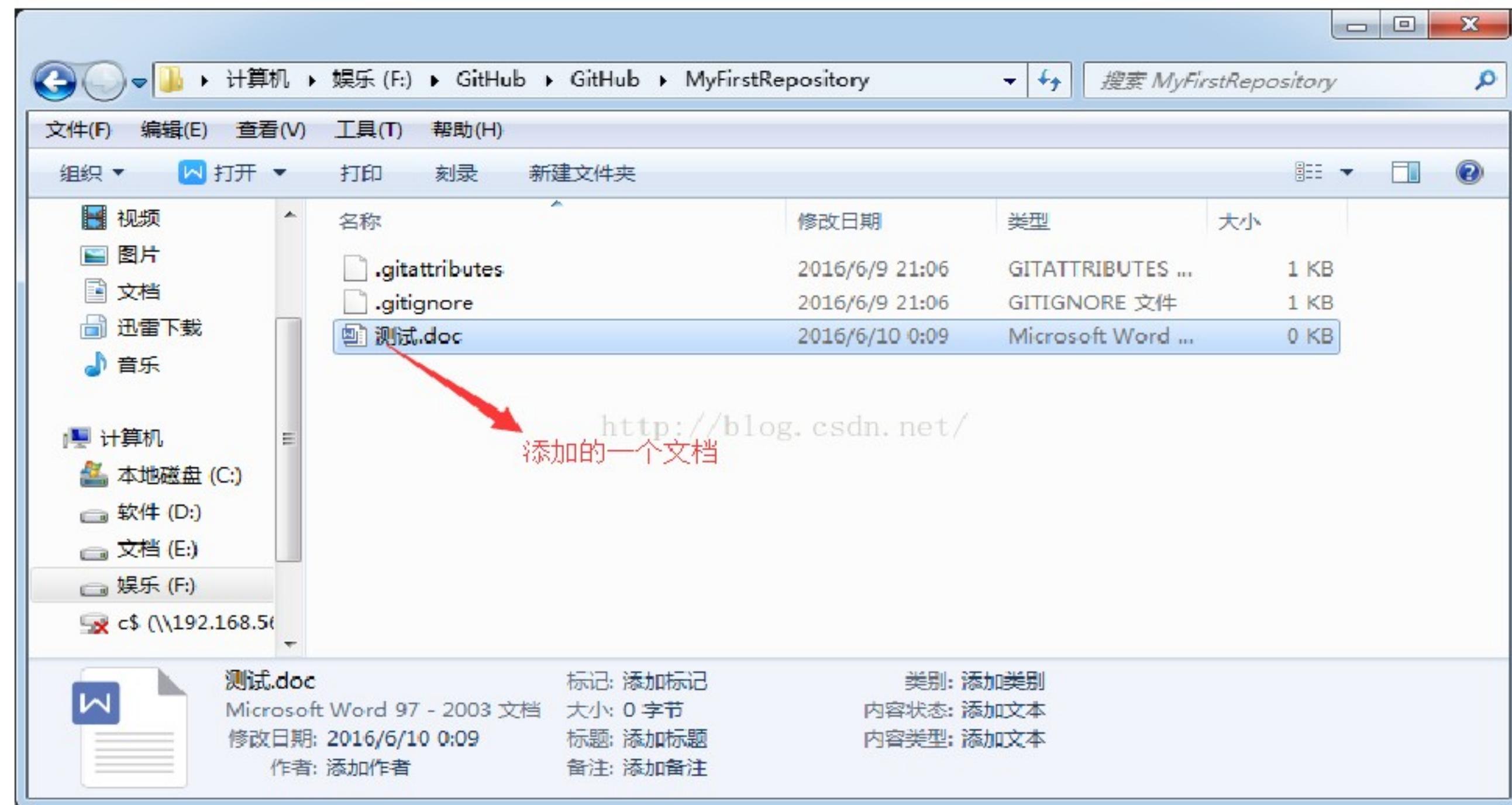
接下来发布这个仓库,点击右上角Publish Repository发布仓库,同时要求填写仓库相应描述



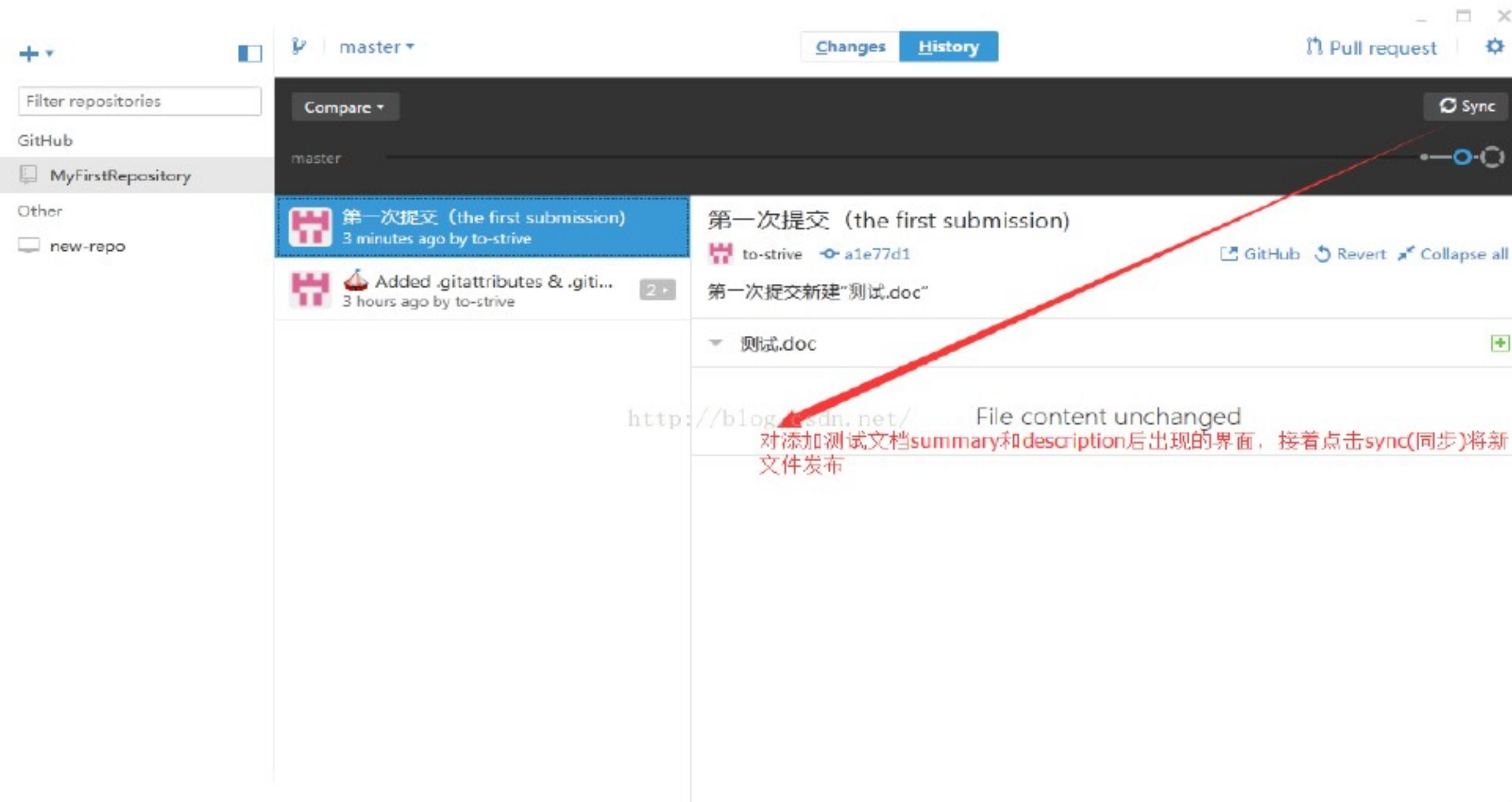
右上角变为Syn表示已经发布成功,右键单击View on GitHub就能直接通过浏览器进入GitHub网站看到我们发布的新仓库了。



7) 新加一些“测试.doc”文件，马上可以看到在工具上多出了一项UnCommitted changes

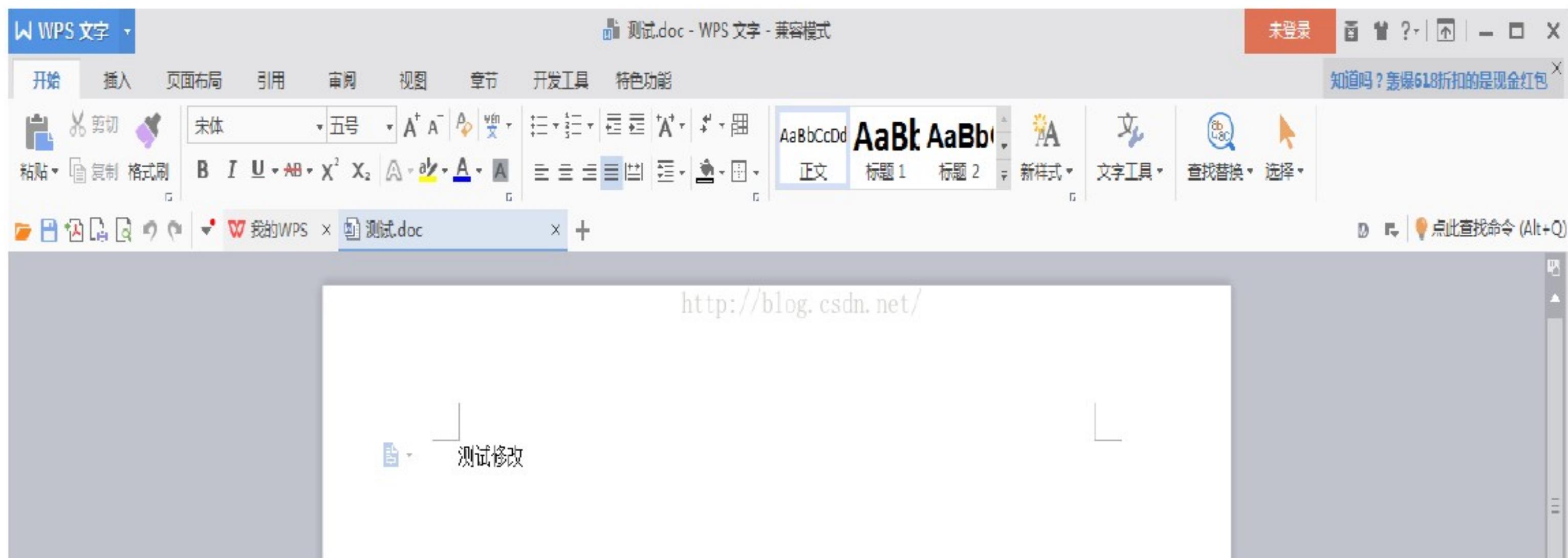


下一步打开GitHub程序，会发现界面有所改变，之后按照提示填写内容，然后点击提交即可，如下图所示：
提交之后如下图所示，然后点击右上角进行同步

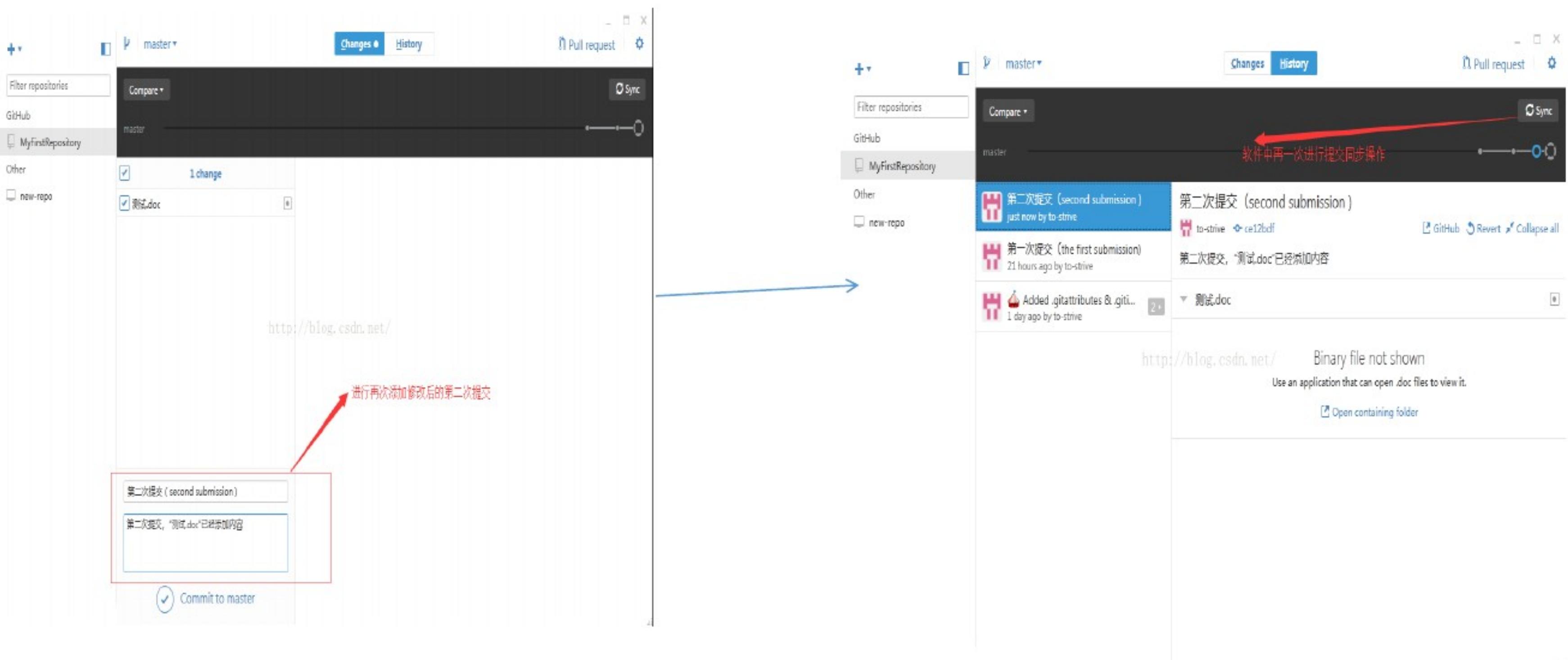


八、修改项目

- 1) 计算机中修改先前新建的空白文档“测试.doc”，如下图所示：



修改保存之后，在软件中再一次进行提交同步操作，如下图所示：



同步之后再浏览器中查看，提交次数已经变成了3次，之后可以分别下载第二次与第三次提交的文档，可以比较看出我们修改的内容。

The screenshot shows a GitHub repository page for '努力 / myfirstrepository'. The repository summary indicates 3 commits, 1 branch, 0 releases, and 1 contributor. A red arrow points to the '3承诺' (3 commits) button. Below the summary, a list of commits is shown:

- 新的承诺 ce12bdf 6分钟前
- .gitattributes .gitattributes > .gitignore文件 added 前一天
- .gitignore .gitattributes > .gitignore文件 added 前一天
- 测试.doc 第二次提交（二提交） 6分钟前

A message at the bottom encourages adding a bio: "帮助人们在这个库有兴趣加入自述了解你的项目。" A green button says "添加一个自述". On the right, there's a network status indicator showing 71% completion, 0.06K/s upload, and 0K/s download. The browser toolbar at the bottom includes: 今日直播, 跨屏浏览, 加速器, 下载, and zoom controls.

对自己的资料进行修改

The image shows two screenshots illustrating how to edit user profile information. A blue arrow points from the left screenshot to the right one.

Left Screenshot (User Profile Page):

- Header: Search GitHub, 技术 问题 仓库
- Profile picture: A large red 'H' icon.
- 简介 (Introduction): "很高兴! 你的名字, 是您独一无二的, 和别人独一无二的用户名! 新的地址: /胡须".
- 仓库 (Repositories):
 - myfirstrepository (1 star)
 - 我的工作 (<http://blog.csdn.net/>) (1 star)
- 努力 (Contributions):
 - 加入2016年
 - 在去年的贡献
- 组织机构 (Organizations):
 - 0 0 0
 - 邀请 主题 贡献
- 底部: 举报水印, 未经许可, 禁止使用, 学习进步的领航者, 少下划线, 0.10%, 今日查杀 0, 防黑功能 0, 反垃圾 0, 下载 0, 0.00%, Q100%

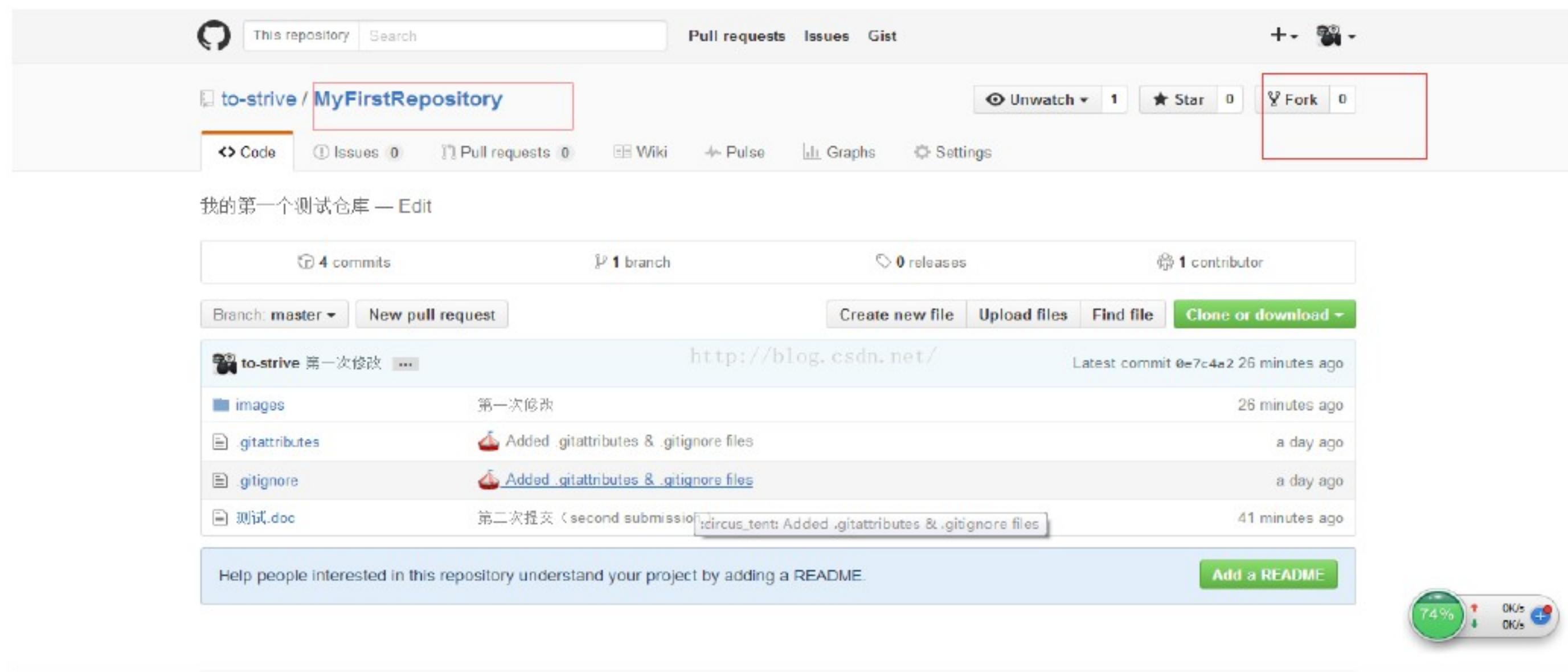
A red arrow points to the introduction text with the annotation: "在这里可以对自己的资料进行修改" (Modify your profile here).

Right Screenshot (Edit Profile Page):

- Header: 个人设置 (Personal Settings)
- 左侧菜单:
 - 简介 (Introduction)
 - 帐户 (Account)
 - 电子邮件 (Email)
 - 通知 (Notifications)
 - 菜单 (Menu)
 - SSH和GPG密钥 (SSH and GPG Keys)
 - 安全部 (Security)
 - OAuth的应用 (OAuth Applications)
 - 个人访问令牌 (Personal Access Tokens)
 - 库 (Repository)
 - 组织 (Organization)
 - 保存的回显 (Saved Echoes)
 - 组织机构的设置 (Organization Settings)
- 右侧公共资料 (Public Profile) section:
 - 资料图片 (Profile Picture): Shows a placeholder image of a person's face. An arrow points to it with the annotation: "在这里进行资料的修改" (Modify your profile here).
 - 上传图片 (Upload Picture): A button to upload a new profile picture.
 - 邮箱 (Email): m1882903723@163.com
 - 公共邮箱 (Public Email): You can add or verify your email address for personal communication.
 - 个人简历 (Personal Bio): <hr> data-src="既然青春留不住, so继续coding" data-dst="既然青春留不住, so继续coding">>既然青春留不住, so继续coding</hr>
 - 你可以邀请其他用户和组织联系他们 (You can invite other users and organizations to contact them):
 - URL: [李先生](#)

七、网站上操作Fork (叉)

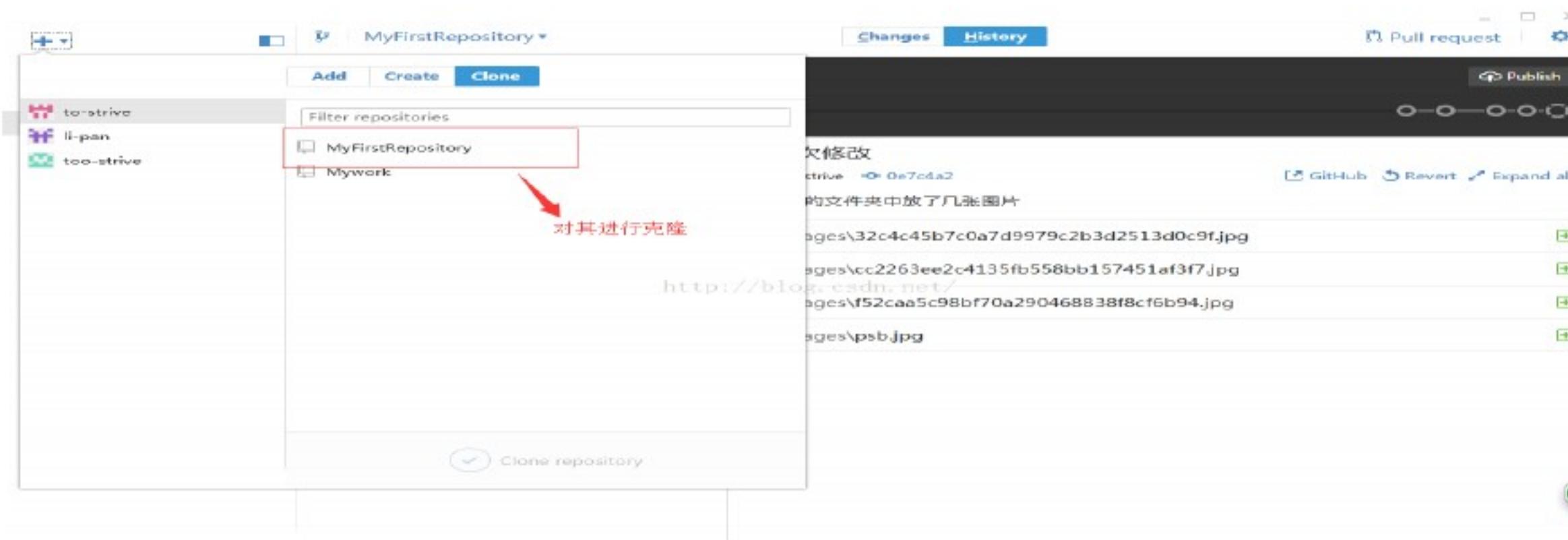
- 1) 现在我们用另外一个用户登录GitHub账户,找到之前to-striver用户上传的叫做myFirstRepository的仓库



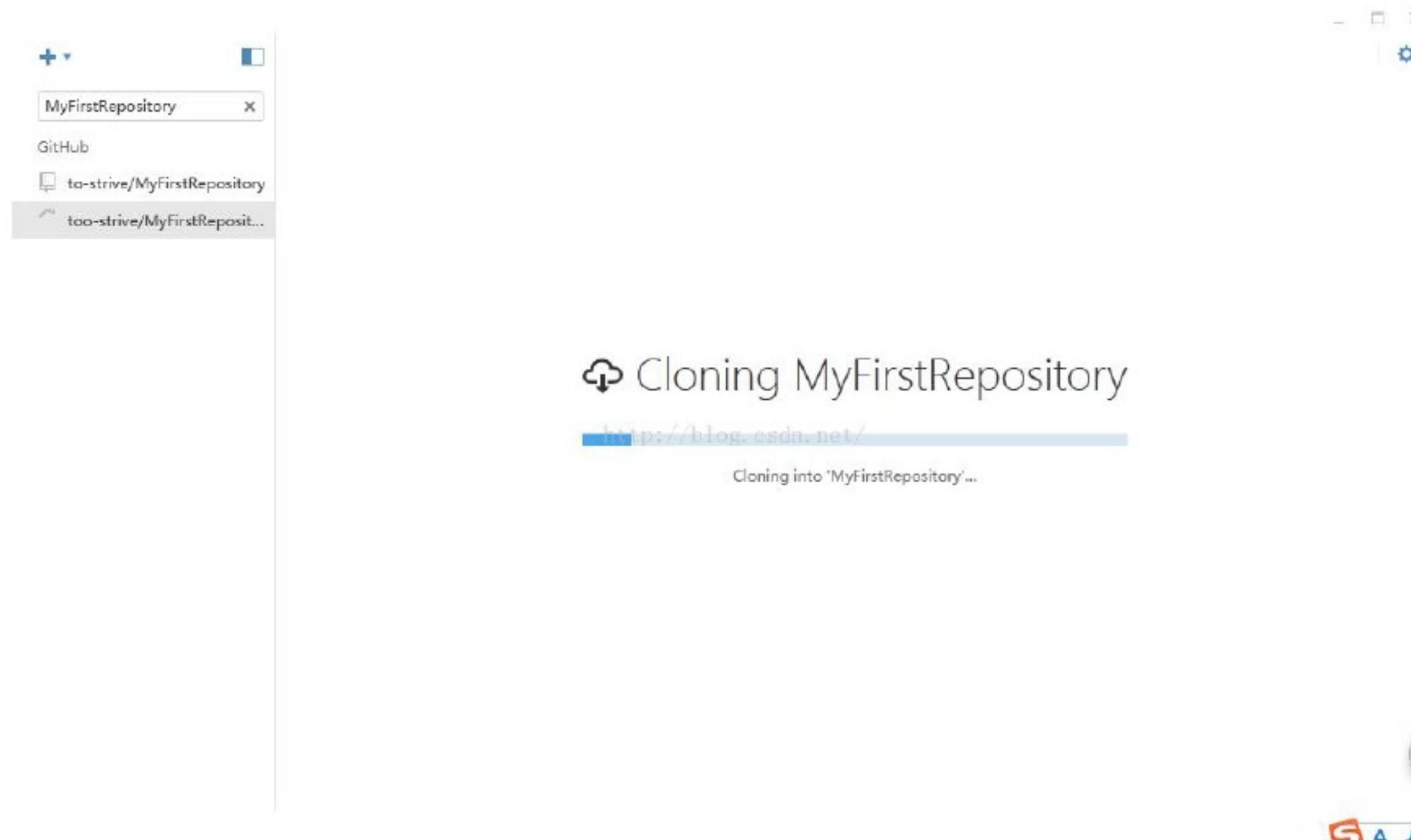
2) 现在你找到了这个仓库,可以将这个仓库直接加入到你的账户当中,点击右上角的Fork,这个仓库就加入了进来



3) 下图可以明显看到现在这个新用户已经Fork了YIFer创建的这份项目：



5)同样将Frok的文件保存到本地目录。



6)当然这里就可以和之前一样,这个新用户同样可以提交修改文件,步骤和之前一样。但是注意一点,这个用户提交修改的文件,只是相当于在他自己 **Fork** 的目录下。简单来说,意思是**你把文件拷贝了一份,然后你在拷贝的这份文件上面进行修改什么的,对源文件是没有影响的**

This repository

Search

Pull requests Issues Gist

to-strive / MyFirstRepository

Unwatch 1 Star 0 Fork 1

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

我的第一个测试仓库 — Edit

4 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

to-strive 第一次修改 ... http://blog.csdn.net/ Latest commit 0e7c4a2 an hour ago

images 第一次修改 an hour ago

.gitattributes Added .gitattributes & .gitignore files a day ago

.gitignore Added .gitattributes & .gitignore files a day ago

测试.doc 第二次提交 (second submission) an hour ago

Add a README

72%

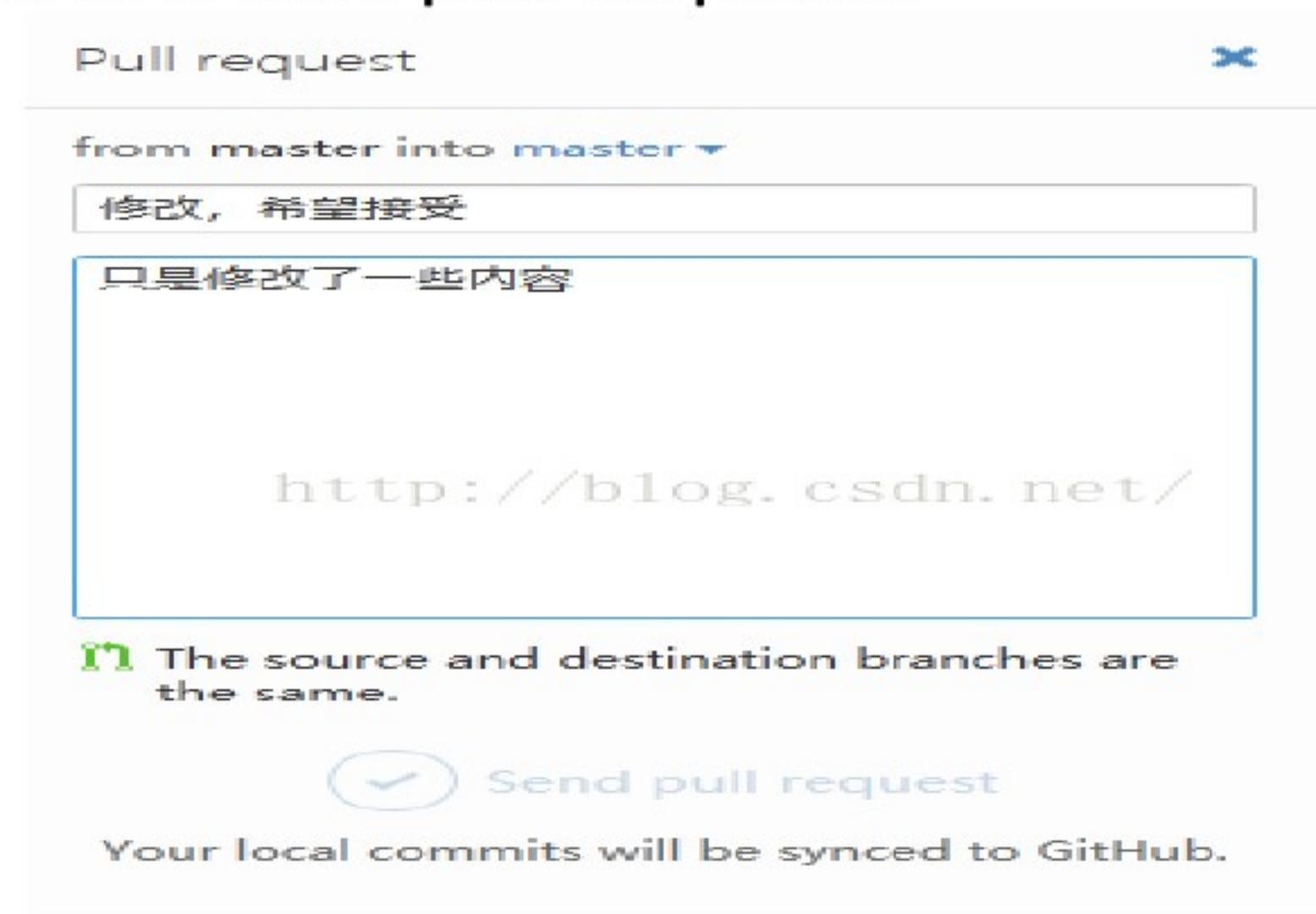
今日直播 跨屏浏览 加速器 下载

https://github.com/to-strive/MyFirstRepository/branches

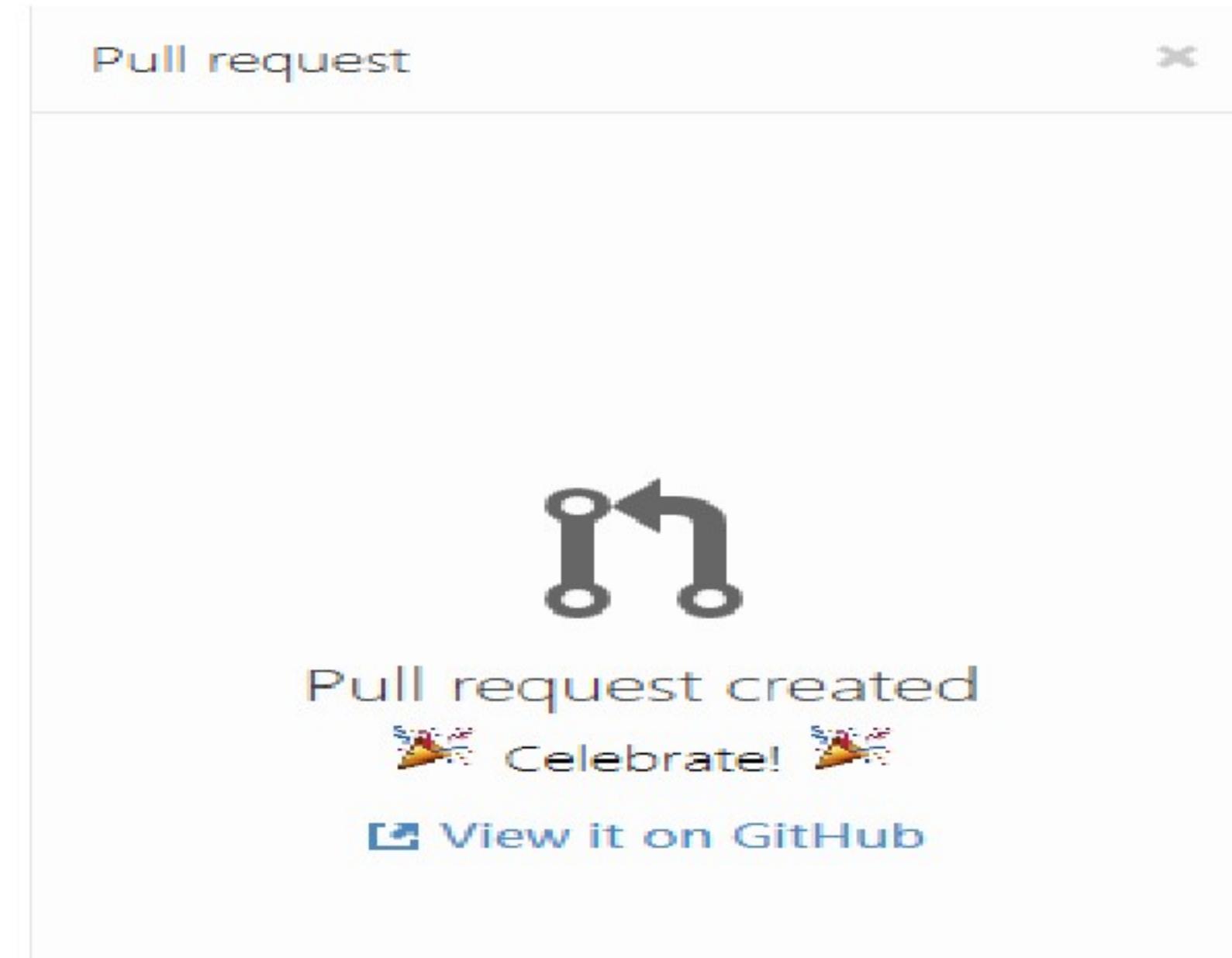
100%

九、Pull Requests(拉请求)

- 1) 如果新用户希望自己的修改的东西对YIFer的代码有影响那么就必须提交申请,让to-strive用户接受,这个就是pull requests.在工具的右上角,点击create pull requests



2) 提交成功,可以直接点击链接,查看这次pull requests



3) 如果是to-strive用户, 接收到别人的pull requests之后, 就会在网站上看到, 注意红色框部分

The screenshot shows a GitHub repository page for '努力 / myfirstrepository'. The repository name is at the top left. At the top right, there are buttons for 'Unwatch' (with 1), '明星' (with 0), 'Y器具' (with 1), and a dropdown menu. Below the header, there are tabs for '代码' (selected), '问题 0', '拉要求 0', '维基', '脉冲', '山图', and '设置'. A summary bar shows '4承诺', '2分支', '0释放', and '1贡献者'. Below this, a section titled '你最近推的树枝:' lists a commit from 'myfirstrepository' (1分钟前). A red arrow points from the text '拉的请求' to the green button labeled '为比较拉的要求'. The main content area shows a list of commits:

- 努力 第一次修改 ... (新的承诺 0e7c4a2 一个小时 ago)
- 图像 (第一次修改 一个小时 ago)
- gitattributes (gitattributes > gitignore文件 added 前一天)
- gitignore (gitattributes > .gitignore文件 added 前一天)
- 测试.doc (第二次提交 (二提交) 2小时前)

At the bottom right, there is a network speed monitor showing 72% completion with speeds of 0.04K/s up and 0.06K/s down. The bottom navigation bar includes links for '今日直播', '跨屏浏览', '加速器', '下载', and a search bar set to 100%.

4) 点击进入,可以看到具体别人提交的pull requests

打开拉请求

通过比较在两个分支变化创造新拉的要求。如果你需要，你也可以比较又.

The screenshot shows the GitHub interface for creating a pull request. At the top, there's a header with user information and a dropdown for '对比' (Compare) with the repository 'myfirstrepository'. A green checkmark indicates that the branches can be merged automatically. Below this, the title of the pull request is '第二次修改' (Second Modification). The body of the pull request contains the following text:

```
<trans data-src="添加了一张图片" data-dst="添加了一张图片">添加了一张图片</trans>
http://blog.csdn.net/
```

On the right side, there are sections for '标签' (Labels), '里程碑' (Milestones), and '受让人' (Assignee). The '标签' section says '没有' (None). The '里程碑' section says '没有里程碑' (No milestones). The '受让人' section has a placeholder '一个好的分配自己的' (A good self-assigned). At the bottom left, it says 'Markdown支持' (Markdown support). On the bottom right, there's a green button labeled '创建接收请求' (Create receive request) and a progress bar indicating 73% completion with upload speeds of 0.6K/s and download speeds of 1.7K/s.

5) 点击查看这次提交,如果可行,选择合并 Merge pull request

打开拉请求

通过比较在两个分支变化创造新拉的要求。如果你需要,你也可以比较又.

The screenshot shows a GitHub pull request interface. At the top, there's a navigation bar with a user icon, the name '基地 硕士', a dropdown menu '对比 myfirstrepository', and a green checkmark indicating '能够合并' (can be merged). Below the bar, a commit titled '第二次修改' is shown. The commit message contains a translatable string: '<trans data-src="添加了一张图片" data-dst="添加了一张图片">添加了一张图片</trans>' and a link 'http://blog.csdn.net/'. There are tabs for '写' (Write) and '预览' (Preview). On the right side, there are sections for '标签' (Labels), '里程碑' (Milestones), and '受让人' (Assignee), each with a gear icon. A progress bar at the bottom indicates '73%' completion. On the far right, there are download and upload speed indicators: '0.6K/s' and '1.7K/s' respectively.

十、创建分支

- 1) 当一个项目有很多个模块，那么我们可以根据模块打出分支，然后让每个开发在对应的**模块的分支上面进行开发**。最后再由一个人进行**分支合并**即可。
- 2) 成功发送一个合并请求之后，图标会发生改变：
- 之后点击会跳转至网页确定真正合并：
- 当我们执行了删除掉已经不用的分支之后，远程库上将不再有这个分支，此时在客户端此分支上执行sync操作之后，会发现sync按钮又变回了publish按钮。此时这个分支是一个本地库分支；

十一、多人协作

- 1) 如果你想发起一个项目，比如翻译一份不错的英文文档，觉得一个人的精力不够，所以你需要更多的人参与进来，这时候 GitHub 是你的最佳选择，感兴趣的人可以参与进来，利用业余时间对这个项目做贡献，然后可以互相审核、合并，简直不要太棒！
- 2) 搭建博客、个人网站或者公司官网这个就不用多说了，现在越来越多的博客都是基于 GitHub Pages 来搭建的了，你可以随心所欲的定制自己的样式，可以给你博客买个逼格高的域名，再也不用忍受各大博客网站的约束与各式各样的广告了！写作如果你喜欢写作，而且基于 Markdown，并准备出版书籍，那么推荐你用 Gitbook。

总结Git一些术语

- 1) **Repository** (仓库) : 一个仓库包括了所有的版本信息、所有的分支和标记信息；在Git中仓库的每份拷贝都是完整的。仓库让你可以从中取得你的工作副本。
- 2) **Branches** (分支) : 一个分支意味着一个独立的、拥有自己历史信息的代码线 (code line)。你可以从已有的代码中生成一个新的分支，这个分支与剩余的分支完全独立。默认的分支往往叫**master**。用户可以选择一个分支，选择一个分支叫做**checkout**.
- 3) **Tags** (标记) : 一个标记指的是某个分支某个特定时间点的状态。通过标记，可以很方便的切换到标记时的状态，例如2009年1月25号在**testing**分支上的代码状态

总结Git一些术语

- 4) Commit (提交) : 提交代码后，仓库会创建一个新的版本。这个版本可以在后续被重新获得。每次提交都包括作者和提交者，作者和提交者可以是不同的人。
- 5) URL: URI用来标识一个仓库的位置
- 6) Revision (修订) : 用来表示代码的一个版本状态。

简易理解GitHub

- GitHub 通俗来说就是 把你已经写好的程序，但还没有完善的更好。这个时候你就可以把你的代码放在GitHub上来得以更好地维护，方便你随时随地对你的代码进行修改。如果过了一段时间，你发现你想看你在修改之前的代码，这个你时候，你只需要在GitHub上查看，这里面它会记录你在何时何地，，在几点进行的修改，你修改的内容是什么。
- 最后，如果你已经觉得你的代码已经非常完美了，这个时候你就可以向软件市场提交你的软件。当然，你的写的代码也可以在次进行修改，修改提交后的另外一款APP