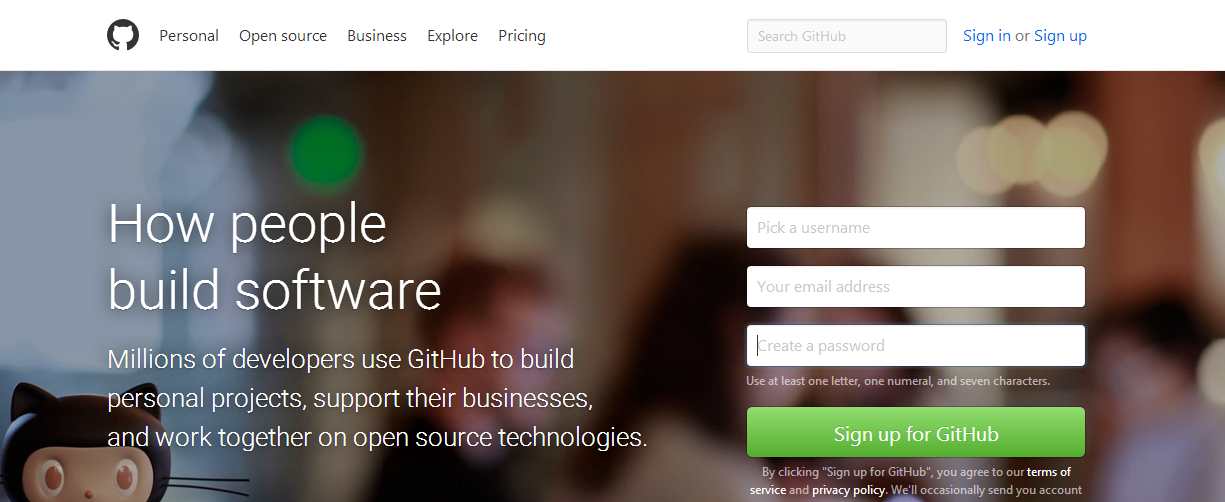
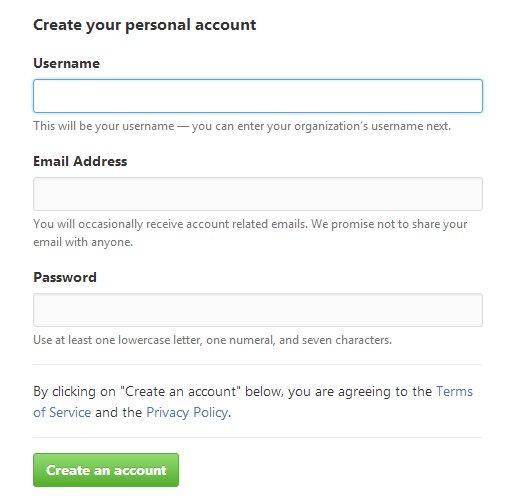
[Github使用教程（一）------ 初识Github](http://www.cnblogs.com/liuleilei/p/6486665.html)

**一、注册账号**



 点击右上角sign up，进行注册，注册界面如下：

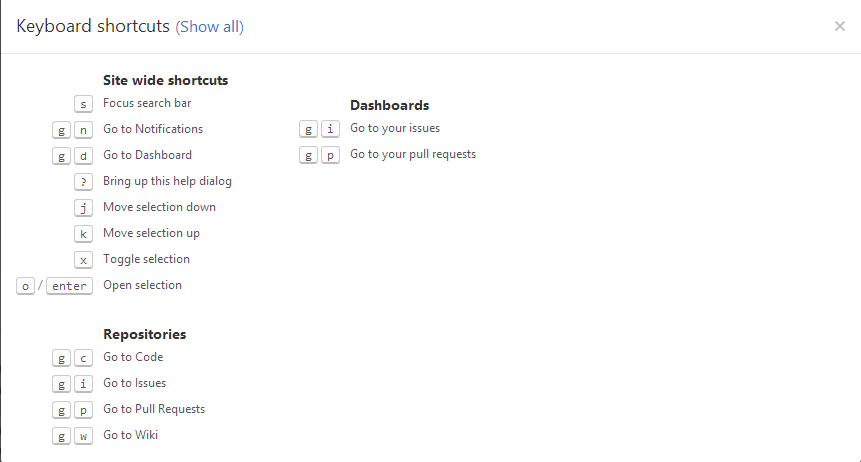


填写用户名，邮箱地址，密码，点击绿色按钮：Create an account  即可注册成功。

点击首页的sign in进行登录，另外顺便再补充一个词组：sign out（登出）。

**二、快捷键**

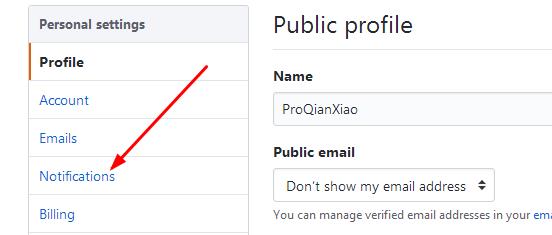
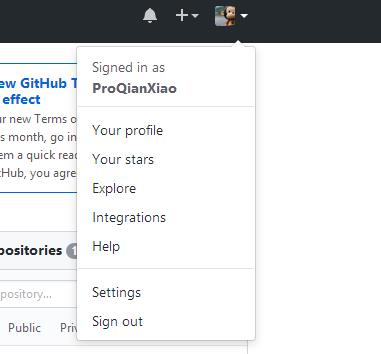
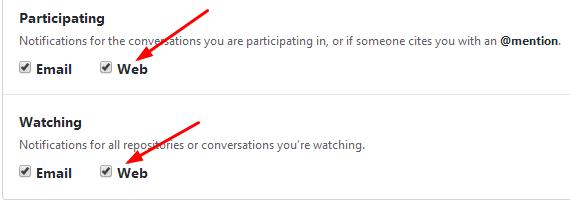
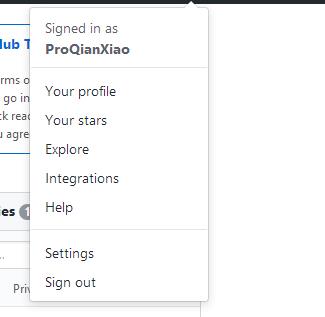
网站也有快捷键，这个功能其实我还是第一次听说，这里和大家分享一下，在Github任一页面按“shift + /”会出现快捷键目录，如下所示：



PS：虽然并没有什么用，个人感觉还是鼠标方便些。

**三、工具栏**

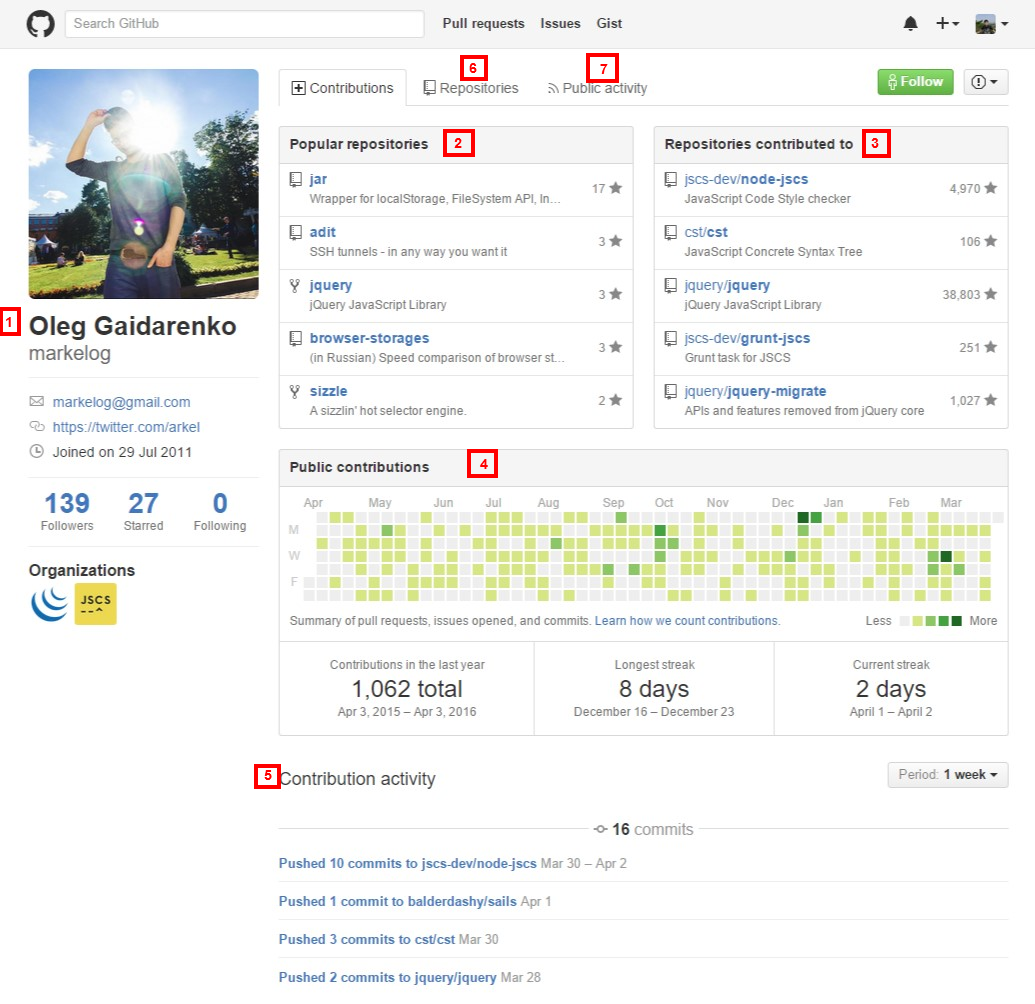


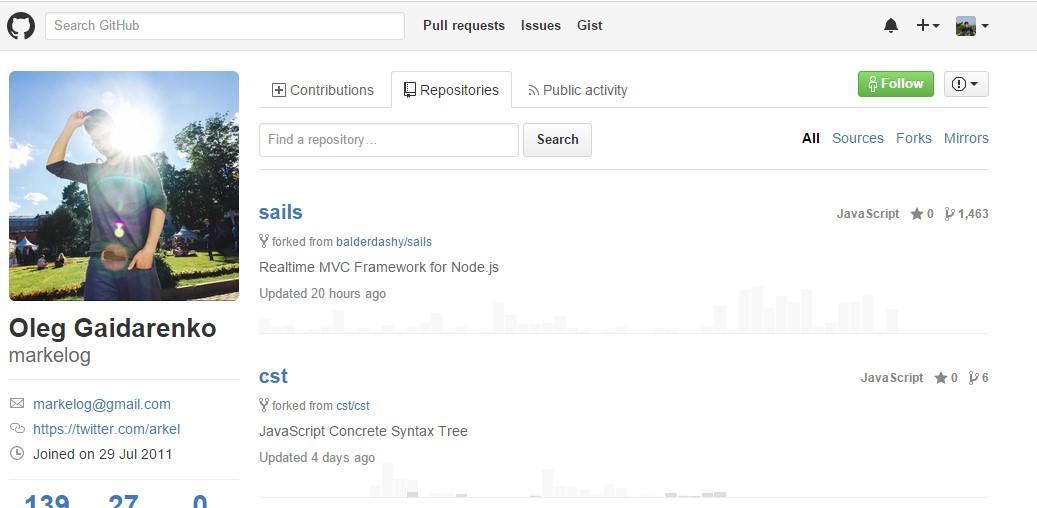
1. LOGO ：点击GitHub的LOGO就会进入控制面板（Github首页）。
2. Search（搜索）： 在这里输入想要找的用户或代码片段，就可以搜索到与之相关的信息。
3. Pull Request（请求）： 显示用户已经进行过的Pull Request。通过这里，开发者可以很方便地追踪Pull Request的后续情况。
4. Issue（问题）： 在这里可以查看用户拥有权限的仓库或分配给自己的Issue。当用户同时进行多个项目时，可以在这里一并查看Issue。
5. Gist（代码片段）： Gist功能主要用于管理及发布一些没必要保存在仓库中的代码，比如小代码片段等。系统会自动管理更新历史，并且提供了Fork功能。在Gist上添加代码示例可以嵌入博客中，如果选择了语言，还会自动添加语法高亮。
6. Notifications（通知） ：这一图标用于提示用户是否有更新的通知。新注册的用户默认是没有该图标的，需要自己进行设置，点击用户头像，在下拉菜单中选择Settings，                       右侧设置栏选择Notifications，                                                                                                                                                                                                                                                                              勾选Web选项，然后点击LOGO回到首页，可以看到通知图标。
7. Create a new... ：创建新的Git仓库或Organization，向Organization（组织）添加成员、小组、仓库，为仓库添加Issue或collaborator（合作者）等操作的菜单都聚集在这里。显示的内容会根据当前页面的不同而改变。
8. 头像：点击头像出现下拉菜单，如图所示：                                                                                                                                                                                                                                                                                                                   Your profile：你的个人主页

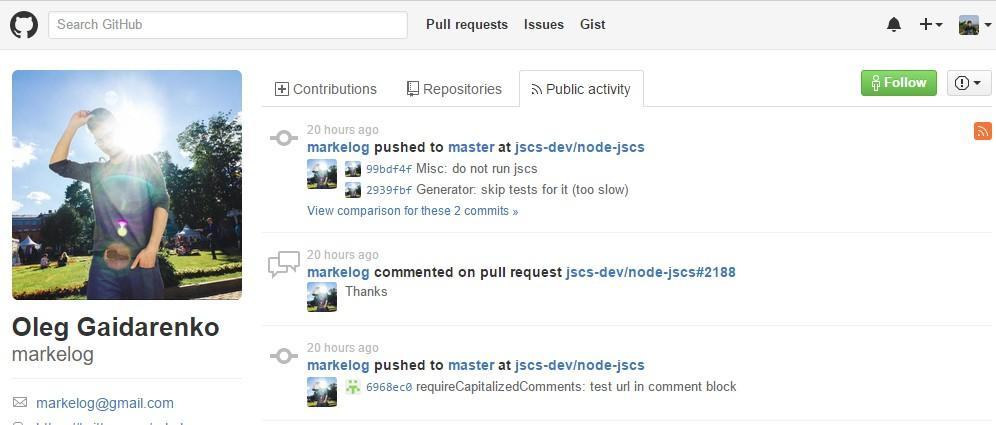
           Your stars：你的收藏

           Explore：搜索

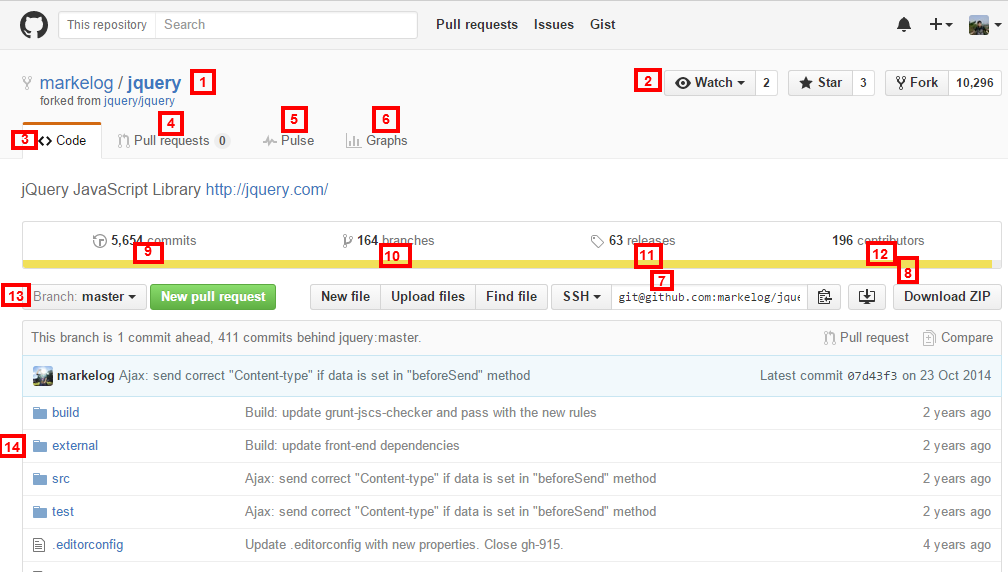
**四、个人信息页**



1. **用户信息**  
   显示注册的基本信息，包括姓名、所属公司、邮箱地址、已加入的Organization（组织）等。如果对该用户感兴趣，可以点击页面右上角的Follow（跟随）按钮（已经Follow的用户会显示Unfollow）。这样一来，这个人在GitHub上的活动都会显示在你的News Feed（新闻提要）中。
2. **Popular Repositories（受欢迎的库）**  
   显示公开仓库中受欢迎的、拥有大量Stars（收藏、标星）的部分热门仓库
3. **Repositories contributed to（做过贡献的库）**  
   按时间先后顺序显示该用户作过贡献的部分仓库。该用户可能是仓库的软件开发者，也可能只是通过发送Pull Request等方式对该仓库做过某些贡献。
4. **Public contributions（公共贡献）**  
   一格表示一天，记录当日用户对拥有读取权限的仓库的大致贡献度。贡献度的衡量标准包括发送Pull Request的次数、写Issue的次数、进行提交的次数等。颜色越深代表贡献度越高，绿色天数越多说明在GitHub上活跃度越高。
5. **Contribution activity（贡献过的活动）**  
   按时间顺序显示具体贡活动的链接
6. **Repositories**  
   显示该用户公共开的仓库。Fork来的仓库也显示在这里。  
   仓库名称、简要说明、使用的语言、最终更新日期都会出现在列表中。星形图案旁边的数字表示这个仓库添加Star的人数，再旁边是杯Fork数。                                   **7. Public Activity（公开的活动）**

**显示该用户的公开活动信息。活动就是指这个用户做了什么，比如向仓库进行提交或者Pull Request等，其大量的公开信息都会被记录在这里。从这里可以了解到这个用户平常都在GitHub上做些什么，比如查看一下崇拜已久的程序员的公开活动，就可以知道他现在在关注些什么，或者正在热心于开发些什么。                                         **

**五、仓库**

****

* + 1. 用户名（组织名）/仓库名
    2. Watch、Star、Fork  
       眼睛图标标着Watch字样，点击这个按钮就可以Watch该仓库，今后该仓库的更新信息会显示在用户的公开活动中。Star旁边的数组表示给这个仓库添加Star的人数，这个数越高，代表该仓库越受关注。  
       Watch与Star不同的地方在于，Watch之后该仓库的相关信息会在您的个人Notifications中显示，让用户可以追踪仓库的内容，而Star更像是书签，让用户将来可以在Star标记的列表中找到该仓库。另外，Star数还是GitHub上判断仓库热门程度的标志之一。
    3. Code，显示该仓库的文件列表，以及该仓库的简单说明和URL。
    4. Pull Request  
       在Pull Request中可以列表查看并管理Pull Request。代码等更改和讨论都可以在这里进行。旁边的数字表示尚未Close的Pull Request的数量。
    5. Pulse  
       显示该仓库最近的活动信息。该仓库中软件是无人问津还是在热火朝天的开发之中，从这里可以一目了然。
    6. Graphs  
       以图表的形式显示该仓库的各项指标，让用户轻松了解该仓库的活动倾向
    7. SSH Clone URL  
       clone仓库是所需要的URL。点击右侧的剪切板图标可以将URL复制到剪切板中，点击STTPS、SSH图标可以切换到相应协议的URL。
    8. Clone in Desktop  
       启动GitHub专用的客户端应用程序进行clone。GitHub专用的客户端应用程序有Windows和Mac版。以及直接下载仓库的ZIP包
    9. commits，查看当前分支的提交历史，左边的数字表示提交数
    10. branches，查看仓库的分支列表，左边的数字表示当前拥有的分支数
    11. releases  
        显示仓库的标签（Tag）列表。同时可以将标签加入时文件以归档形式（ZIP、tar.gz）下载到本地。软件在版本升级时一般都会打标签，如果需要特定版本的文件，可以从这里寻找。
    12. contributore（贡献者）  
        显示对该仓库进行提交的程序员名单。左边的数字表示程序员的人数。
    13. files（文件列表）  
        可以产看当前分支的文件，顶端为最新提交的相关信息。在文件或目录的列表中，从左至右分别为文件名称、该文件最新的提交日志、更新日期。点击目录或文件可以查看相应内容。

# [Github使用教程（二）------ Github客户端使用方法](http://www.cnblogs.com/liuleilei/p/6533177.html)

。

# Github for windows（客户端）是什么

      GitHub 使用 git 分布式版本控制系统，而 git 最初是 Linus Torvalds 为帮助Linux开发而创造的，它针对的是 Linux 平台，因此 git 和 Windows 从来不是最好的朋友，因为它一点也不像 Windows。GitHub 发布了GitHub for Windows，为 Windows 平台开发者提供了一个易于使用的 Git 图形客户端。

      GitHub for Windows 是一个 Metro 风格应用程序，集成了自包含版本的 Git，bash 命令行 shell，PowerShell 的 posh-git 扩展。GitHub 为 Windows 用户提供了一个基本的图形前端去处理大部分常用版本控制任务，可以创建版本库，向本地版本库递交补丁，在本地和远程版本库之间同步。

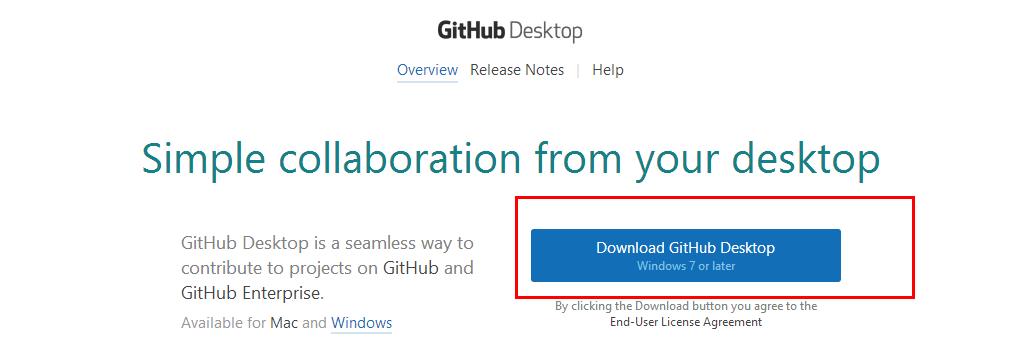
**简而言之，以前我们使用Git命令行进行版本控制任务，如今GitHub for Windows为我们提供一个图形客户端，方便我们的操作。**

# Github for windows下载及安装

## 安装方法一：在线安装

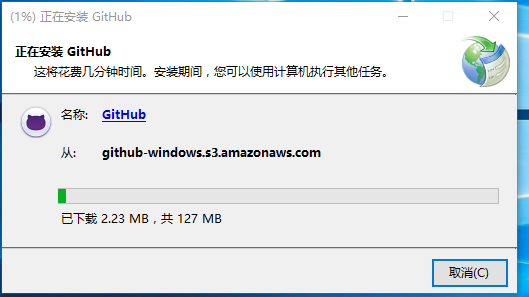
 Github官网，打开我们的仓库，可以看到安装GitHub for Windows的按钮：

 点击“Set up in Desktop”，进入如下界面：

              点击蓝色按钮进行下载，我们可以看到一个名为“GithubSetup.exe”的安装文件



双击打开，进入在线下载界面，等待下载完成，一路默认安装即可。



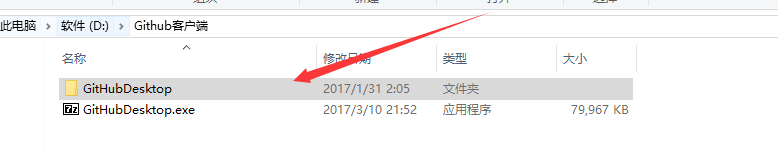
## 安装方法二：离线安装包

方法一安装的时候，在下载的过程常常失败，或者下载速度极慢，所以推荐方法二，使用离线安装包进行安装。[点此下载](https://link.zhihu.com/?target=https%3A//github-windows.s3.amazonaws.com/standalone/GitHubDesktop.exe)

下载后得到如下文件：

https://images2015.cnblogs.com/blog/1111888/201703/1111888-20170310220358857-976751311.png

双击，会自动打开cmd命令窗口（不用管），之后会发现在该文件夹下面出现一个GithubDesktop文件夹：



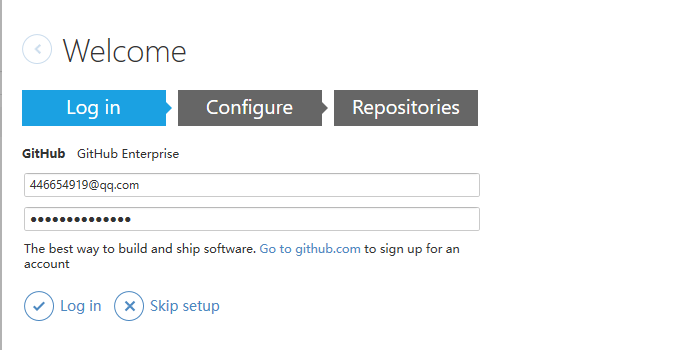
打开该文件夹，点击该文件夹下的Setup.exe即可安装成功。此时，桌面出现两个图标：



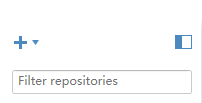
GitHub是图形界面模式，Git Shell是命令行模式，在Windows系统下我们使用GitHub进行代码管理。

# Github for Windows使用方法：

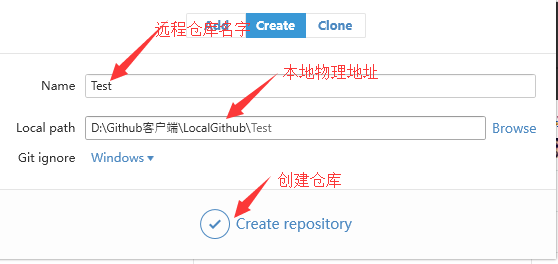
1、打开Github，输入账号密码进行登录。



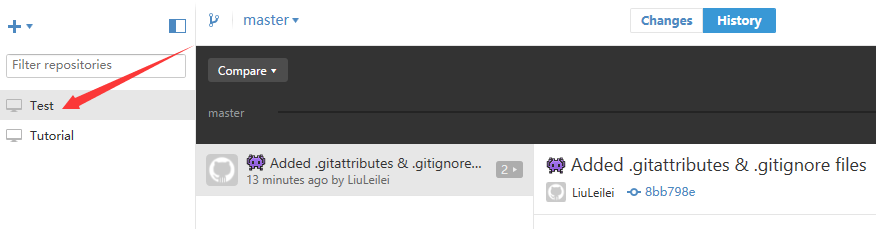
2、登录之后，点击左上角，新建项目。



填写相应信息，创建仓库：



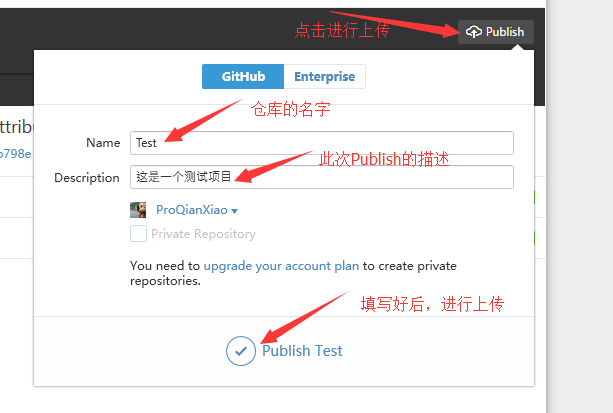
新建之后，如下图所示：



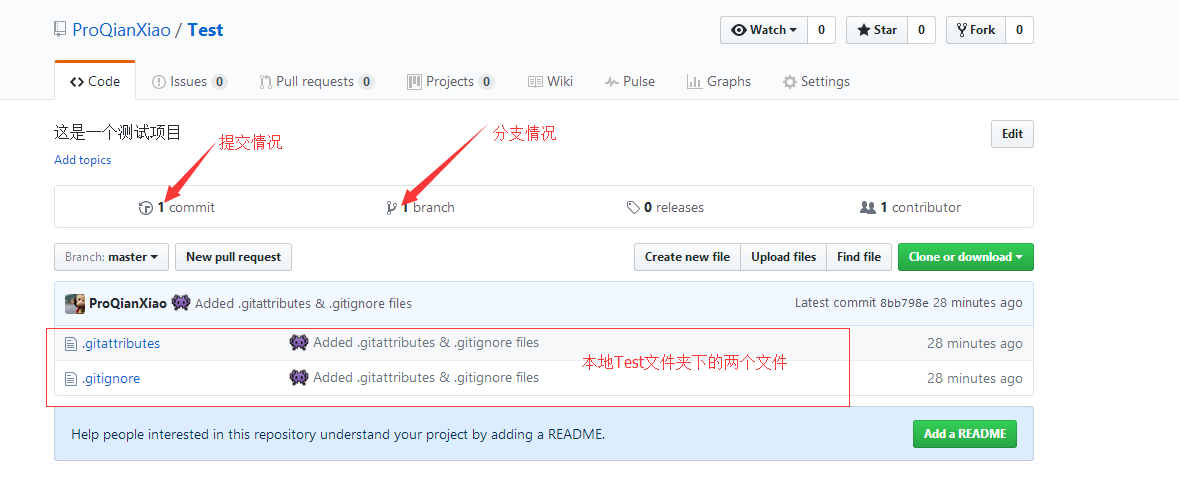
在电脑上找到我们本地路径，目录结构如图所示：



此时，只是在本地建立好了相应的本地仓库，Github网站上面还未上传。点击客户端右上角Publish进行上传。



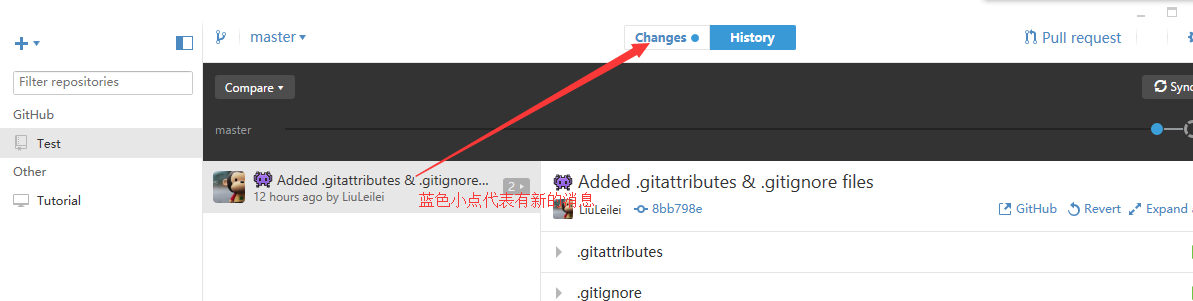
此时新建的仓库已经成功提交到Github，现在点击客户端右上角https://images2015.cnblogs.com/blog/1111888/201703/1111888-20170310230039873-156832387.png，选择View on Github，可以看到我们刚刚创建的Test仓库。



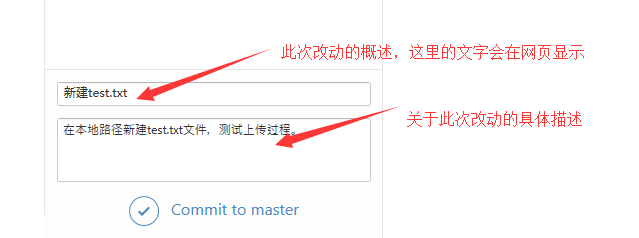
 3、打开我们的本地路径（如果忘记也不要紧，Github客户端鼠标右键我们的仓库，选择选择Open in Explorer，这样我们就可以转到刚刚的路径下），在里面新建test.txt。



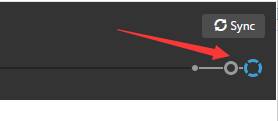
此时，打开客户端可以看到"Test"仓库对应的change选项有一个新消息：



4、在客户端的下方填写相关描述，点击"Commit to master"进行提交。

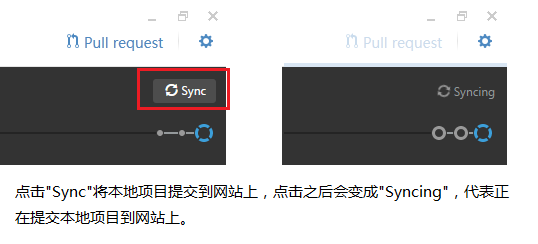


5、点击"Commit to master"只是让Github客户端记录这次进行的改动，相当于一次游戏存档，在将来的某一天如果想重玩这个存档，可以点击上面的圆圈“读取存档”，让项目恢复到这次记录的状态。

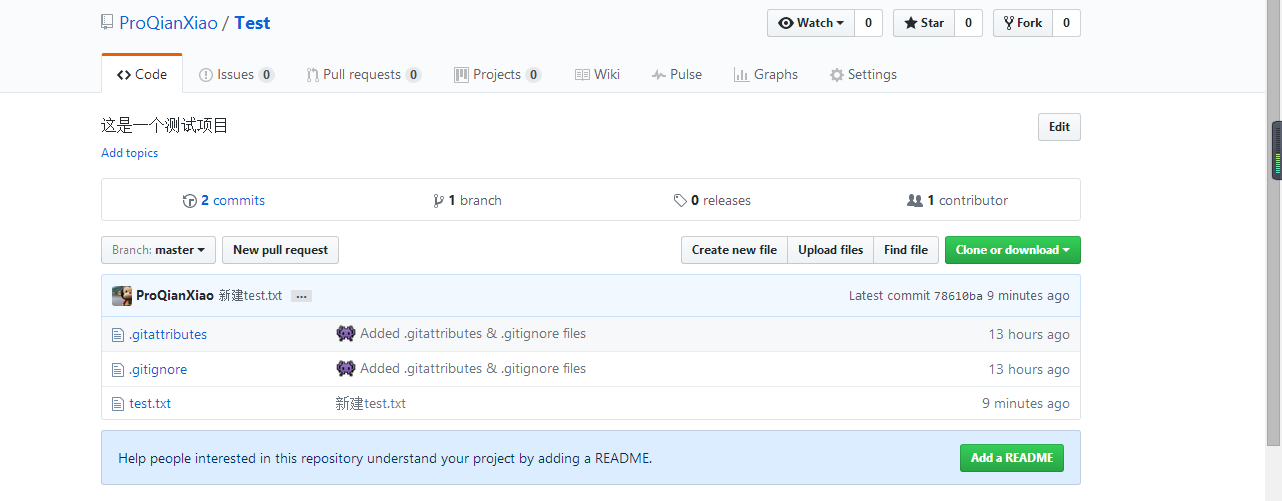


将鼠标移动到这些圆圈之上，会显示之前填写的改动描述。每个圆圈代表一次“存档“，点击某个圆圈可以让项目回到圆圈所代表的那一时刻，相当于读取存档。

6、如果想将本地的改动提交到网站上，可以点击右上角的"Sync"。



7、现在到自己的Github主页上查看Test仓库，可以看到我们新建的"test.txt"已经出现在个人主页上了。



 之后的每次改动，只要重复第4、5、6个步骤即可。