

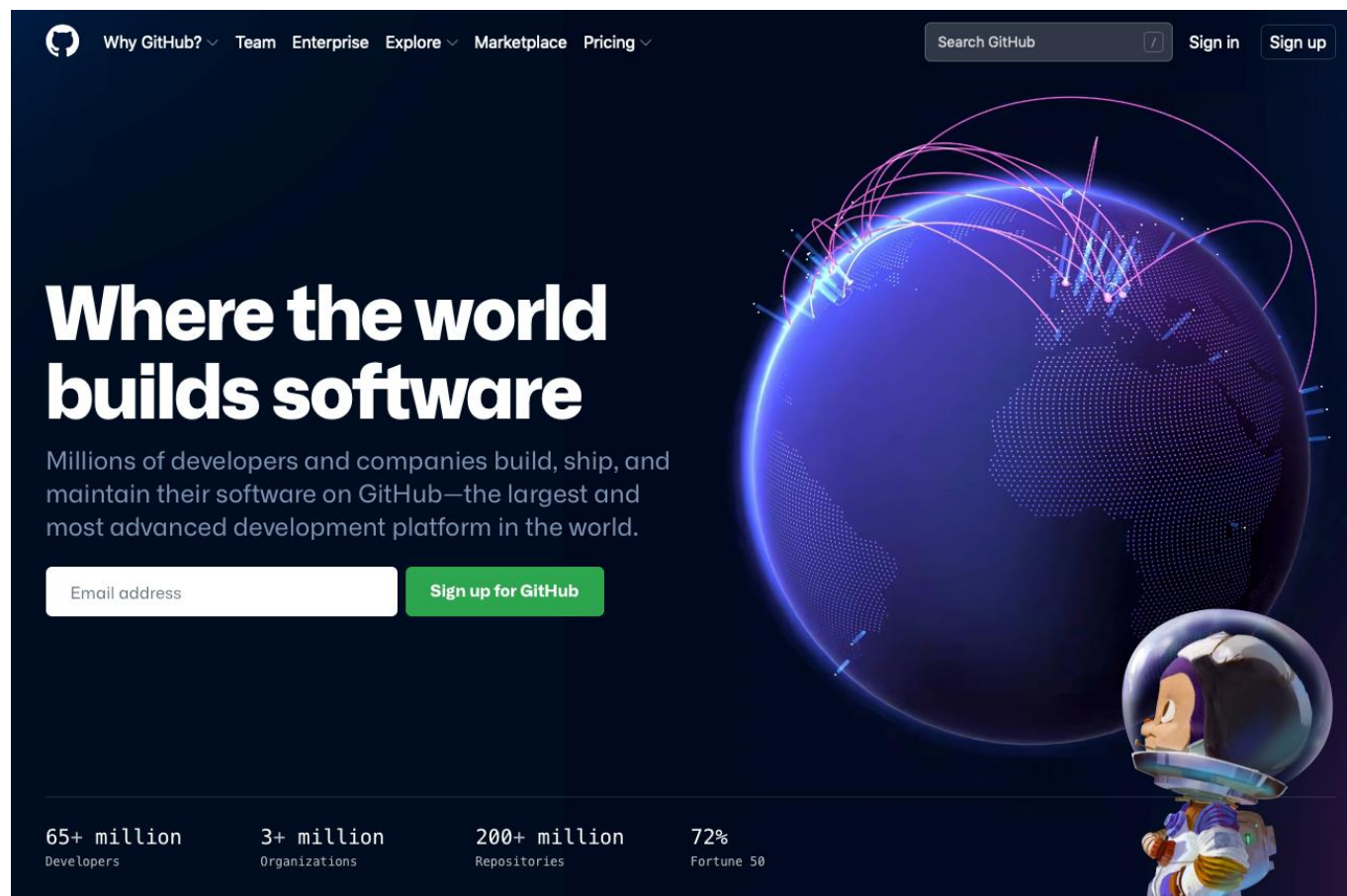
使用GitHub进行项目托管

纲要

- 注册GitHub账号
- 新建GitHub项目
- 查找GitHub项目
- 远程修改GitHub项目
- 对GitHub项目进行反馈
- GitHub的提醒和通知
- GitHub上的组织
- 实践：在课程组织中创建个人项目

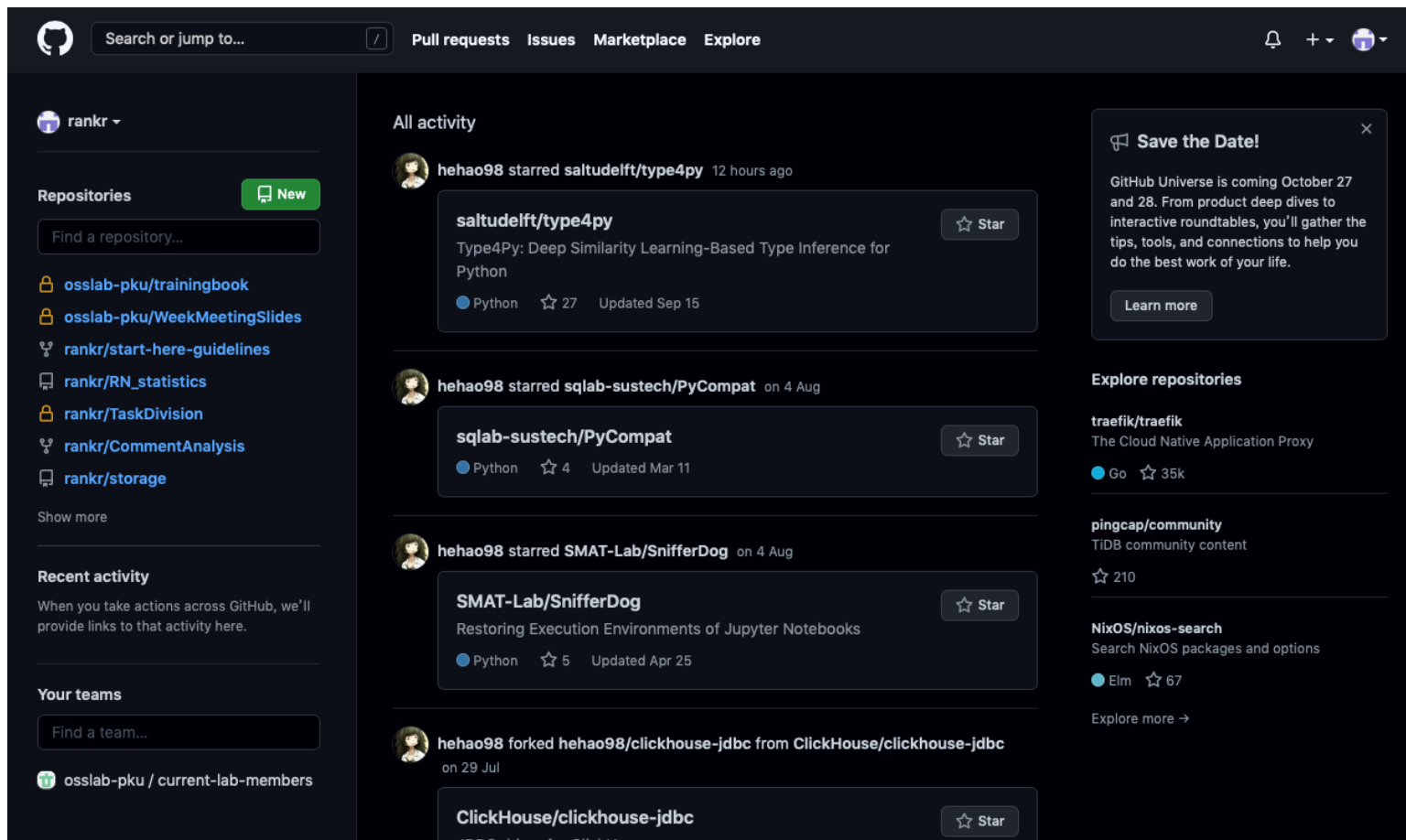
注册GitHub账号

- 官方网站 <https://github.com>
- 点击Sign up即可注册



GitHub个人主页

- 在左侧可以看到自己拥有的项目
- 中间可以看到自己所关心的活动（通过star项目、star其他开发者等方式）
- 在右侧可以看到一些项目推荐



新建GitHub项目


- 点击项目列表上方的New按钮即可
- 然后填写项目名、描述等少量信息即可创建新的项目，非常便捷

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *


Repository name *


 rankr ▾

/

Great repository names are short and memorable. Need inspiration? How about **verbose-succotash**?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

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

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

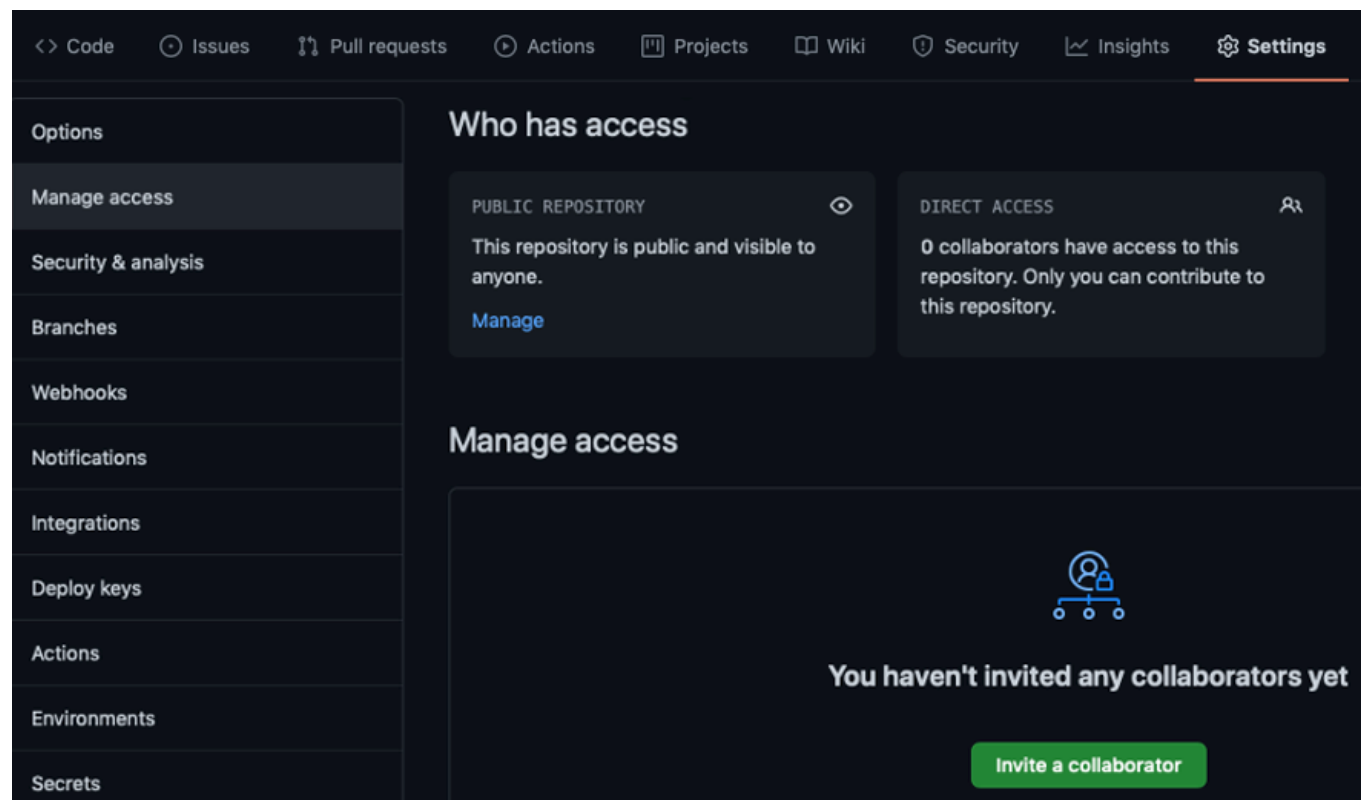
☒ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☒ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

新建GitHub项目

- 可以为项目邀请合作者
- Settings – Manage access



查找GitHub项目

- 通过GitHub页面的左上角的搜索栏可以对GitHub上所有公开的事物进行搜索，包括项目、组织、开发者、commit、pull request、issue等等
- 通过设置一些参数，可以进行高级搜索
 - 如 **cats stars:>1000** matches repositories with the word "cats" that have more than 1000 stars
 - 搜索语法参考 <https://docs.github.com/en/github/searching-for-information-on-github/getting-started-with-searching-on-github/understanding-the-search-syntax>

查找GitHub项目

- 如，以linux作为关键词，可以看到搜索结果中出现了知名开源项目linux，和很多其他相关结果（相关项目、commit、issue等等）

The screenshot shows the GitHub search interface with the keyword 'linux' entered in the search bar. The search results are displayed in a dark theme. On the left, there is a sidebar with filters for Repositories (263K), Code (1B), Commits (634M+), Issues (5M), Discussions (14K), Packages (2K), Marketplace (59), Topics (1K), Wikis (240K), and Users (29K). Below this, there is a 'Languages' section with a list of languages and their respective counts: Shell (52,635), C (45,906), Python (21,576), C++ (15,250), JavaScript (5,735), HTML (5,498), and Vim script (4,693). The main content area shows the search results for 'linux'. At the top, there is a card for the 'Linux' project, which is an open source kernel. Below this, there is a section titled '263,620 repository results' with a 'Sort: Best match' dropdown. The first three results are listed: 1. 'torvalds/linux' (Linux kernel source tree) with 118k stars and updated 15 hours ago. 2. 'raspberrypi/linux' (Kernel source tree for Raspberry Pi Foundation-provided kernel builds) with 8.3k stars and updated 4 hours ago. 3. 'jaywcjlove/linux-command' (Linux命令大全搜索工具) with 16.7k stars, HTML license, and updated 5 days ago. The third result is marked as a 'Sponsor'. Below the third result, there is a partial view of the 'OxAX/linux-insides' repository.

Filter	Count
Repositories	263K
Code	1B
Commits	634M+
Issues	5M
Discussions	14K
Packages	2K
Marketplace	59
Topics	1K
Wikis	240K
Users	29K

Language	Count
Shell	52,635
C	45,906
Python	21,576
C++	15,250
JavaScript	5,735
HTML	5,498
Vim script	4,693

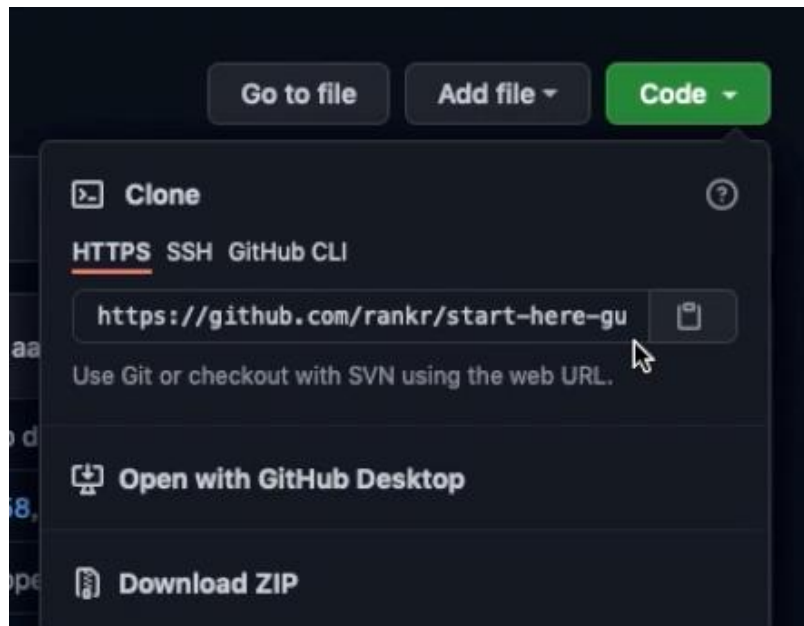
Repository	Description	Stars	License	Updated
torvalds/linux	Linux kernel source tree	118k	C	Updated 15 hours ago
raspberrypi/linux	Kernel source tree for Raspberry Pi Foundation-provided kernel builds. Issues unrelated to the linux kernel should be...	8.3k	C	Updated 4 hours ago
jaywcjlove/linux-command	Linux命令大全搜索工具，内容包含Linux命令手册、详解、学习、搜集。https://git.io/linux	16.7k	HTML	Updated 5 days ago
OxAX/linux-insides	A little bit about a linux kernel			

远程修改GitHub项目

- 开发中，往往需要在本地对项目进行修改，再将修改传到GitHub上合并。流程包括：
 - 从GitHub上获取项目的最新版本到本地（往往需要fork项目，在fork的项目上修改）
 - 在本地对项目进行修改和提交（往往需要新建一个分支）
 - 向远程项目发起一个拉取请求（Pull Request）
 - 与项目维护者进行沟通，对此次修改进行完善
 - 最终项目维护者合并或关闭拉取请求

获取项目

- 以初学者项目[zero-to-mastery/start-here-guidelines](https://github.com/rankr/start-here-guidelines) 为例
 - 点击项目主页右上方的fork按钮，这个项目的拷贝就会出现在你的个人项目中
- 可以通过点击项目右上方的绿色Code按钮，直接下载项目到本地
 - 也可以通过git clone <https://github.com/用户名/项目名> 下载项目到本地



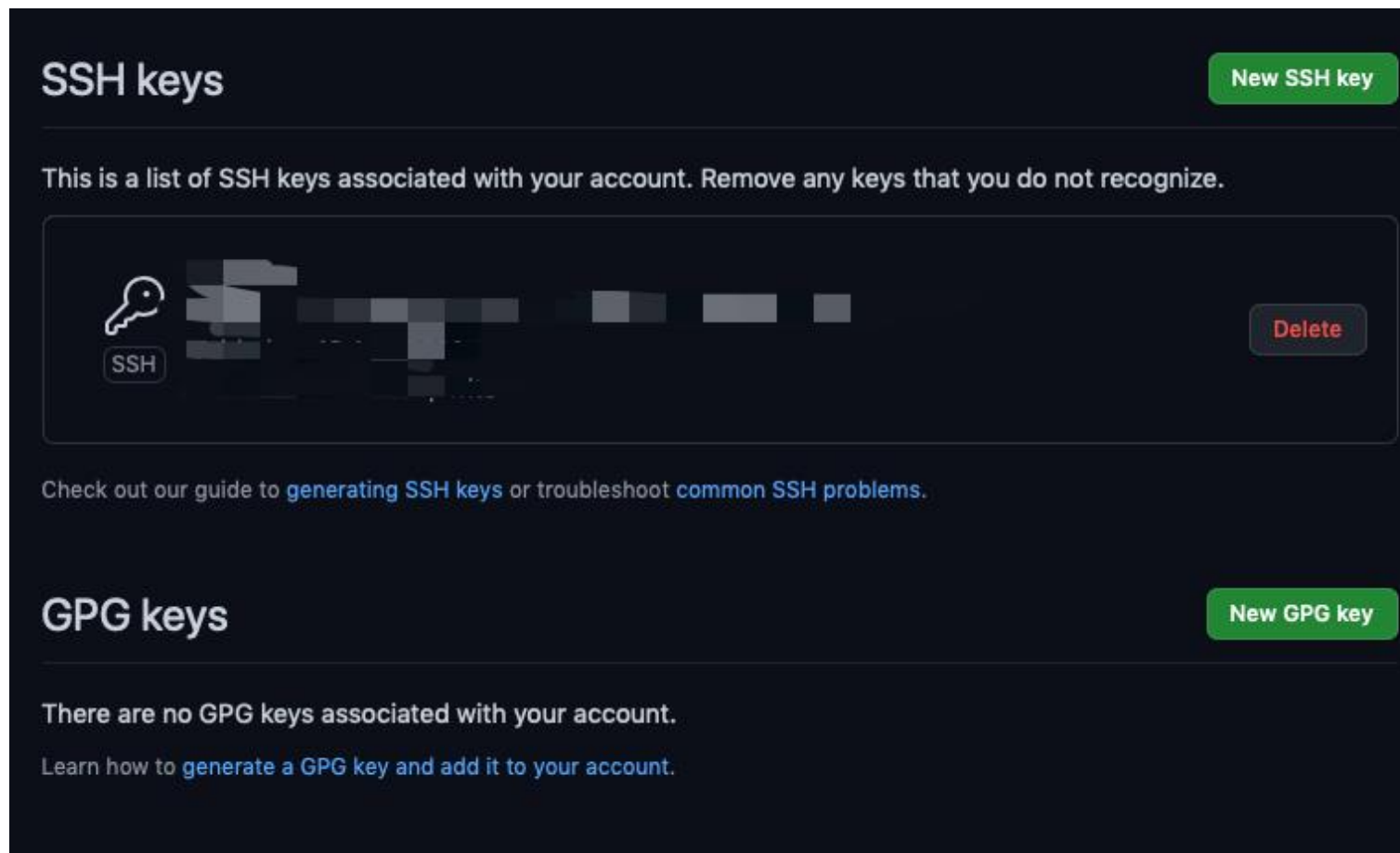
进行GitHub认证

- GitHub通过双重身份验证、SSH、和提交签名验证等功能保持帐户和数据的安全。如果本地没有进行认证，则无法将本地的修改提交到GitHub上。
- 以SSH验证为例：
 - 本地生成ssh公私钥（ssh-keygen -t rsa），执行后，在~/.ssh 目录下会发现有两个文件id_rsa, id_rsa.pub, 前者是ssh私钥，后者是公钥
 - 点击GitHub的个人头像，点击下拉菜单中的Settings，可以看到很多设置选项，点击SSH and GPG keys, 就能看到设置ssh验证的页面。

```
[(base) pengchengdeMacBook-Pro:~$ ls -l  
total 40  
-rwx-----@ 1 apple  staff  1675  8 21 22:43 MacPro.pem  
-rw-r--r--  1 apple  staff   233  8 21 22:51 config  
-rw-----  1 apple  staff  1675  7  8 16:25 id_rsa  
-rw-r--r--  1 apple  staff   398  7  8 16:25 id_rsa.pub  
-rw-r--r--  1 apple  staff  2075  9 14 22:11 known_hosts
```

进行GitHub验证

- 点击New SSH key按钮，将id_rsa.pub内容复制到指定文本框，设置此ssh key 的权限即可完成GitHub SSH验证的设置



进行GitHub认证

- 确认认证成功：
 - 在本地命令行执行ssh -T [git@github.com](https://github.com)
 - 如果出现类似下图的文本，说明认证成功

```
(base) pengchengdeMBP>HelloGit apple$ ssh -T git@github.com
Hi rankr! You've successfully authenticated, but GitHub does not provide shell access.
(base) pengchengdeMBP>HelloGit apple$ _
```

将本地修改同步到GitHub

- 在本地修改完项目，并且进行提交之后
 - 执行 `git push <远程仓库名> <本地分支名>:<远程分支名>` 将本地修改同步到GitHub上。此例中本地和远程的分支名相同，可以简写成一个

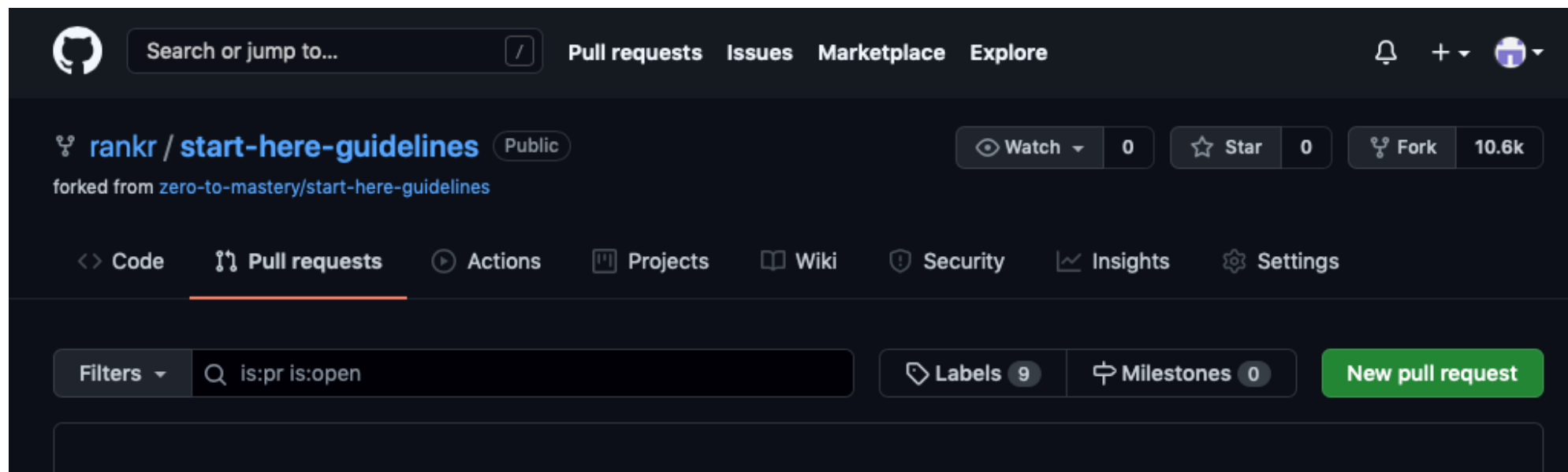
```
(base) pengchengdeMBP:start-here-guidelines apple$ git push origin master:master
Username for 'https://github.com': rankr
Password for 'https://rankr@github.com':
Enumerating objects: 5, 完成.
对象计数中: 100% (5/5), 完成.
用 4 个线程进行压缩
压缩对象中: 100% (3/3), 完成.
上传对象中: 100% (3/3), 293 bytes | 293.00 KiB/s, 完成.
共 3 (差异 2), 复用 0 (差异 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/rankr/start-here-guidelines.git
   de79cbe0..67f1d732  master -> master
(base) pengchengdeMBP:start-here-guidelines apple$ _
```

- 可以通过 `git remote` 查看远程仓库名

```
(base) pengchengdeMBP:start-here-guidelines apple$ git remote
origin
```

创建拉取请求

- 在项目页面，点击pull request，可以看到创建pull request的按钮



创建拉取请求

- 填写好pull request的相关内容之后（取决于项目的贡献要求），该项目（此例中为[zero-to-mastery/start-here-guidelines](#)）的所有相关维护者都会收到关于新pull request的邮件

The screenshot shows a GitHub pull request for the 'blink' repository, forked from 'schacon/blink'. The pull request is titled 'Three seconds is better' and is created by 'tonychacon'. The description states: 'Studies have shown that 3 seconds is a far better LED delay than 1 second.' and includes a link to 'http://studies.example.com/optimal-led-delays.html'. The interface indicates that the pull request is 'Able to merge' and shows '1 commit', '1 file changed', '0 commit comments', and '1 contributor'. The diff view shows changes to 'blink.ino', with line 21 changed from 'delay(1000);' to 'delay(3000);' and line 23 changed from 'delay(1000);' to 'delay(3000);'.

tonychacon / blink
forked from schacon/blink

Unwatch 1 Star 0 Fork 1

schacon:master ... tonychacon:slow-blink Edit

Three seconds is better

Write Preview Parsed as Markdown Edit in fullscreen

Studies have shown that 3 seconds is a far better LED delay than 1 second.

<http://studies.example.com/optimal-led-delays.html>

Attach images by dragging & dropping or [selecting them](#).

✓ Able to merge.
These branches can be automatically merged.

Create pull request

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Oct 01, 2014

tonychacon three seconds is better db44c53

Showing 1 changed file with 2 additions and 2 deletions. Unified Split

4 blink.ino View

```
@@ -18,7 +18,7 @@ void setup() {  
18 18 // the loop routine runs over and over again forever:  
19 19 void loop() {  
20 20     digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
21 - delay(1000); // wait for a second  
21 + delay(3000); // wait for a second  
22 22     digitalWrite(led, LOW); // turn the LED off by making the voltage LOW  
23 - delay(1000); // wait for a second  
23 + delay(3000); // wait for a second  
24 24 }
```


创建拉取请求

- 通知请求的邮件，包括：
 - 合并远程分支的代码
 - 进行合并请求的GitHub链接
 - .diff和.patch的url

[fade] Wait longer to see the dimming effect better (#1)



 **Scott Chacon** <notifications@github.com>
to tonychacon/fade [Unsubscribe](#)

10:05 AM (0 minutes ago) ☆



One needs to wait another 10 ms to properly see the fade.

You can merge this Pull Request by running

```
git pull https://github.com/schacon/fade patch-1
```

Or view, comment on, or merge it at:

<https://github.com/tonychacon/fade/pull/1>

Commit Summary

- wait longer to see the dimming effect better

File Changes

- M [fade.ino](#) (2)

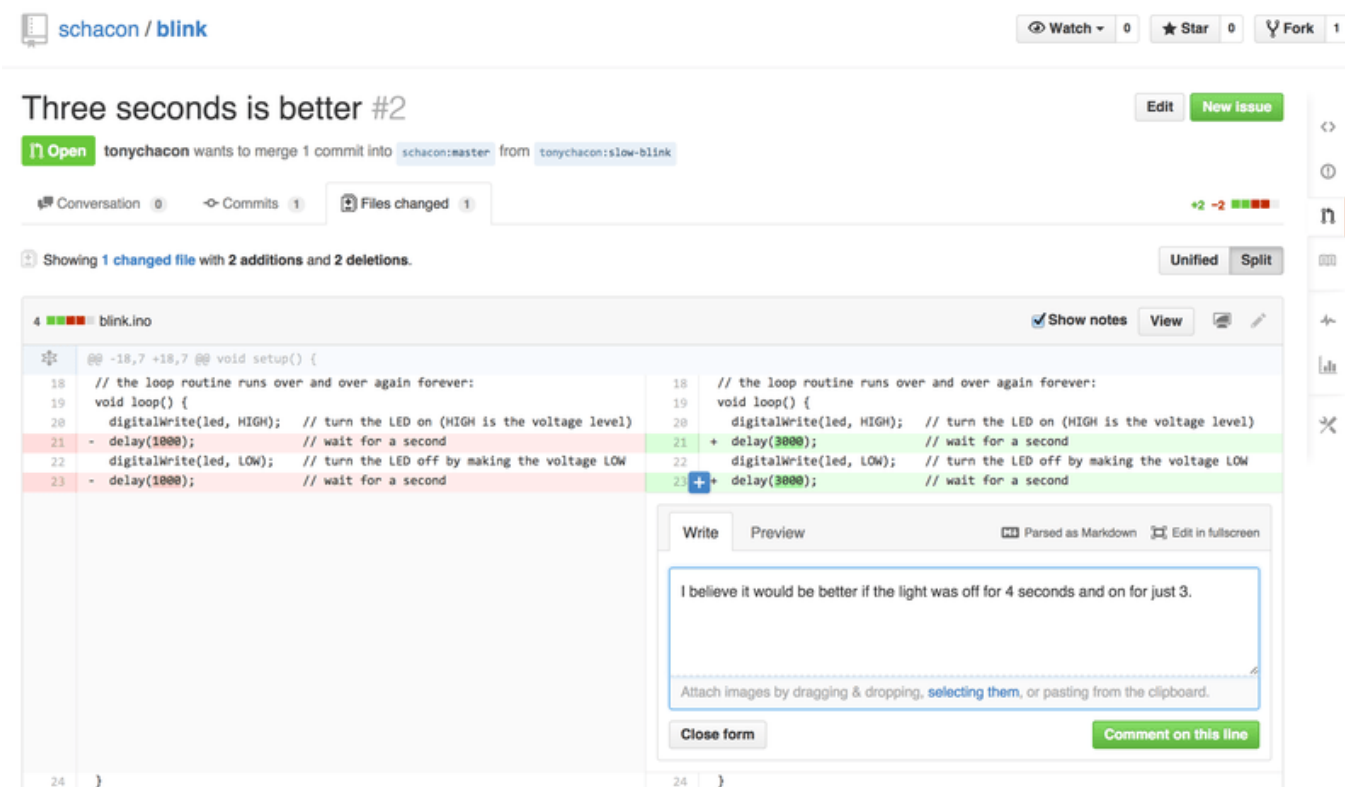
Patch Links:

- <https://github.com/tonychacon/fade/pull/1.patch>
- <https://github.com/tonychacon/fade/pull/1.diff>

—
Reply to this email directly or [view it on GitHub](#).

沟通和完善

- 项目的拥有者可以看到你的改动并合并它，拒绝它或是发表评论
- 维护者做出评论之后，提交拉取请求的人，以及所有正在关注（Watching）这个版本库的用户都会收到通知

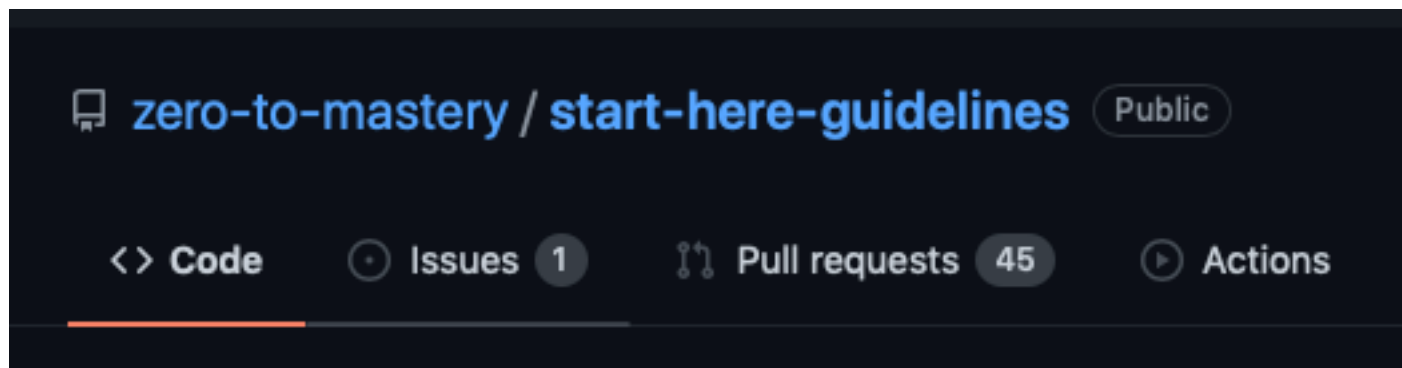


沟通和完善

- 项目维护者审查pull request之后，可能有问题或者建议想与代码提交者沟通，在该pull request（有的项目中是邮件列表）下会有相关评论出现，GitHub对此也会进行提醒
- 反复沟通修改后，pull request会被合并到原始项目中（或者被废弃）。到此，一次对GitHub上项目的贡献就完成了！

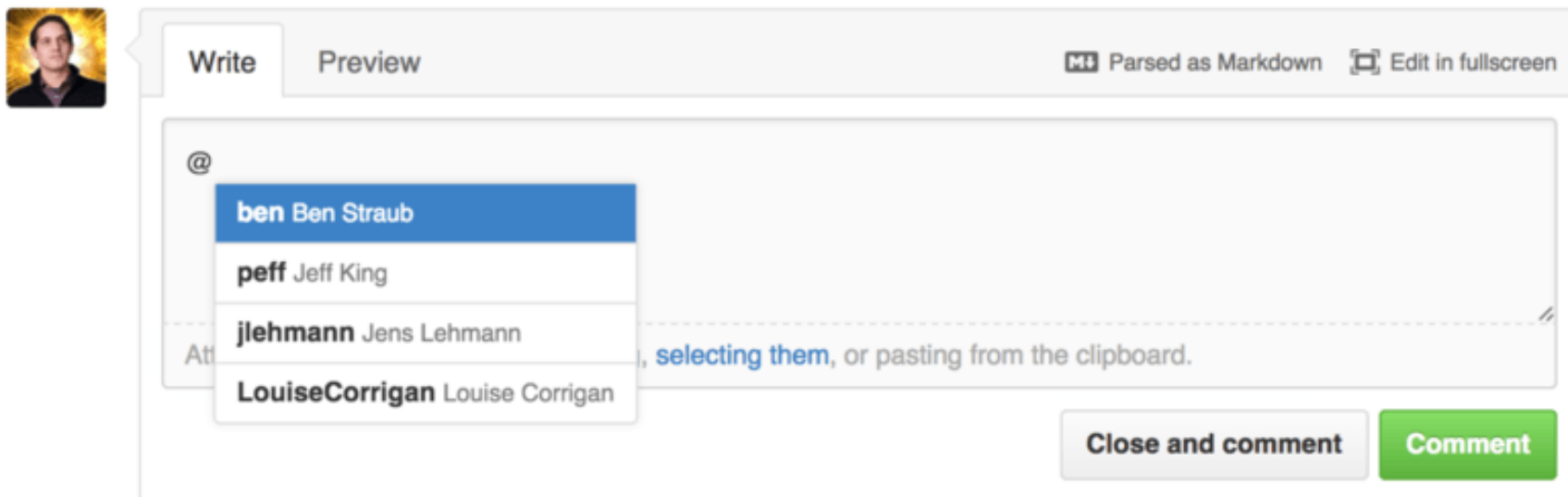
对GitHub项目进行反馈

- 大部分GitHub项目都开启了缺陷追踪系统（Issue Tracking System），发现这些项目的bug，或者发现有哪些可以改进的地方时，可以通过开启一个Issue的方式告知对应项目的开发者和维护者，促进项目的进步
- 点击项目名下方的Issues按钮，即可看到创建Issue的页面
- 填写Issue名称，完善对应描述（如果是bug最好包括出现bug的详细信息，如系统、如何复现、有哪些错误提示等）之后即可创建好一个Issue。如果是活跃的项目，很快就会有开发者进行回复



GitHub的提醒和通知

- GitHub内置了一个很好的通知系统，当你需要与别人或别的团队交流时用起来很方便
 - 在任何评论中输入@，系统会自动补全项目中合作者或贡献者的名字和用户名
 - 当你发布了一个带用户提醒的评论，那个用户会收到通知
 - 如果有人收到了合并请求或问题的提醒，他们会“订阅”它，后面有新的活动发生他们都会持续收到提醒。
 - 合并请求或者问题的发起方也会自动订阅

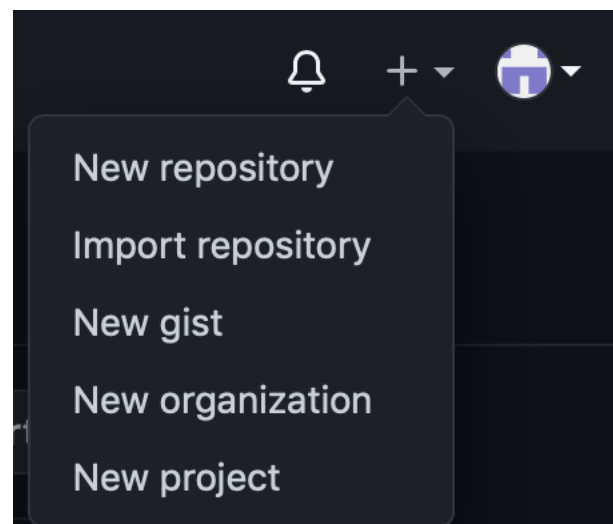


GitHub上的组织 (Organization)

- 除了个人帐户之外，GitHub 还提供被称为组织的帐户
 - 组织账户和个人账户一样都有一个用于存放所拥有项目的命名空间
 - 组织帐户代表了一组共同拥有多个项目的人
 - GitHub提供一些工具用于对成员进行分组管理
 - 通常，这种账户被用于开源群组（例如：“perl”），或者公司（例如：“google”）

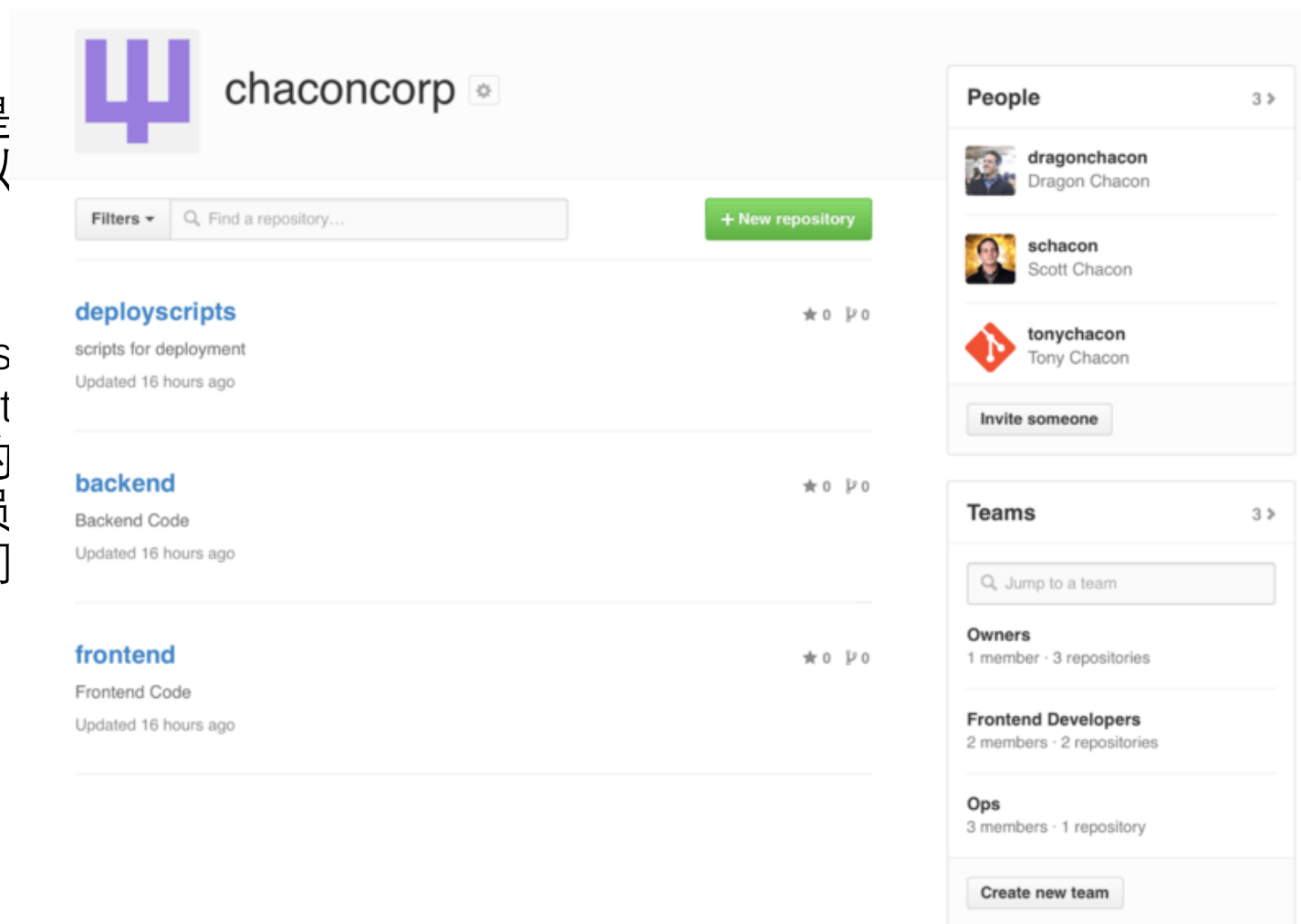
创建组织

- 点击GitHub页面右上角的“+”图标，选择New Organization即可
- 必须提供组织的名称和主要联系邮箱
- 作为一个组织的拥有者，当你在新建或者派生版本库的时候，你可以选择把它派生或者新建到你的组织的命名空间内
- 组织也可以上传头像



团队 (Team)

- 组织使用团队来管理成员，团队就是组织中的一组个人账户和版本库，以及团队成员对这些版本库的访问权限
- 例子：假设你的公司有三个版本库：frontend、backend 和 deployscripts 你会希望你的 HTML/CSS/Javascript 开发者有 frontend 或者 backend 的访问权限，操作人员有 backend 和 deployscripts 的访问权限



团队

- GitHub提供对团队的管理
 - 以添加新成员和版本库到团队中
 - 管理团队的访问权限和其它设置
 - 团队被@的时候，所有成员都会成为这个话题的订阅者
 - 一个用户可以加入任意数量的团队

实践：在课程组织中创建个人项目

- 将GitHub账号以邮件/微信形式发给助教，助教发送GitHub邀请到[OSS-Dev-Course-PKU](#)组织，Lab 1、Lab 4和Lab5中涉及操作GitHub的部分均会在此进行；此外，后续的课程项目也会以此处告知的GitHub账号为准进行评分。
- 在[OSS-Dev-Course-PKU](#)中创建一个GitHub仓库，名为2022Fall-{学号}-Lab1。
- 创建一个md文件，描述自己可能有兴趣贡献的开源项目或任务的特征。