

本项目为使用Markdown维护的个人主页模板。

# 使用指南

## 1.文件说明

[./source/img/logo.jpg](#)为头像或者logo，使用时可将其替换为自己的照片，需改名为logo.jpg

[./config.json](#)中为配置信息，目前仅有Name值，修改Name的值为自己的名字，这会影响到网页的title以及导航栏左上角的标牌。

[./contact.md](#)中为使用markdown书写的个人联系方式，根据需要更改

[./index.md](#)中为Home界面的主要内容

[./pages/course/index.md](#)为Course界面的主要内容

[./pages/resume/index.md](#)为Resume界面的主要内容

[./pages/publications/index.md](#)为Publications界面的主要内容

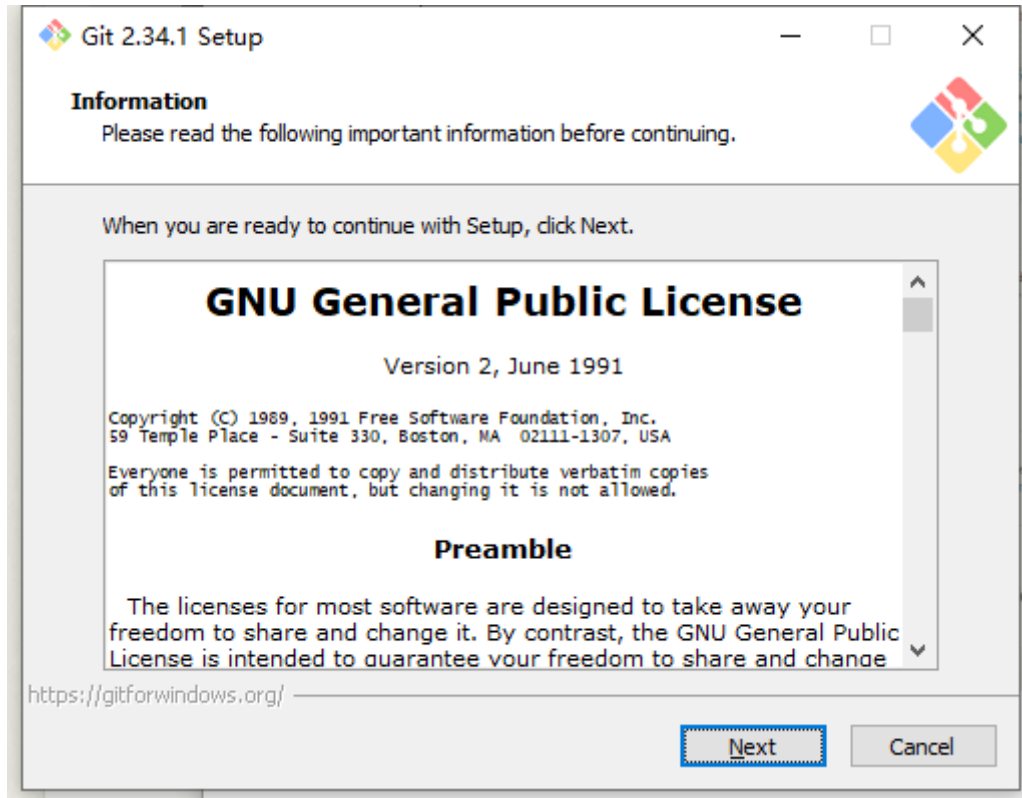
[./pages/link/index.md](#)为Links界面的主要内容

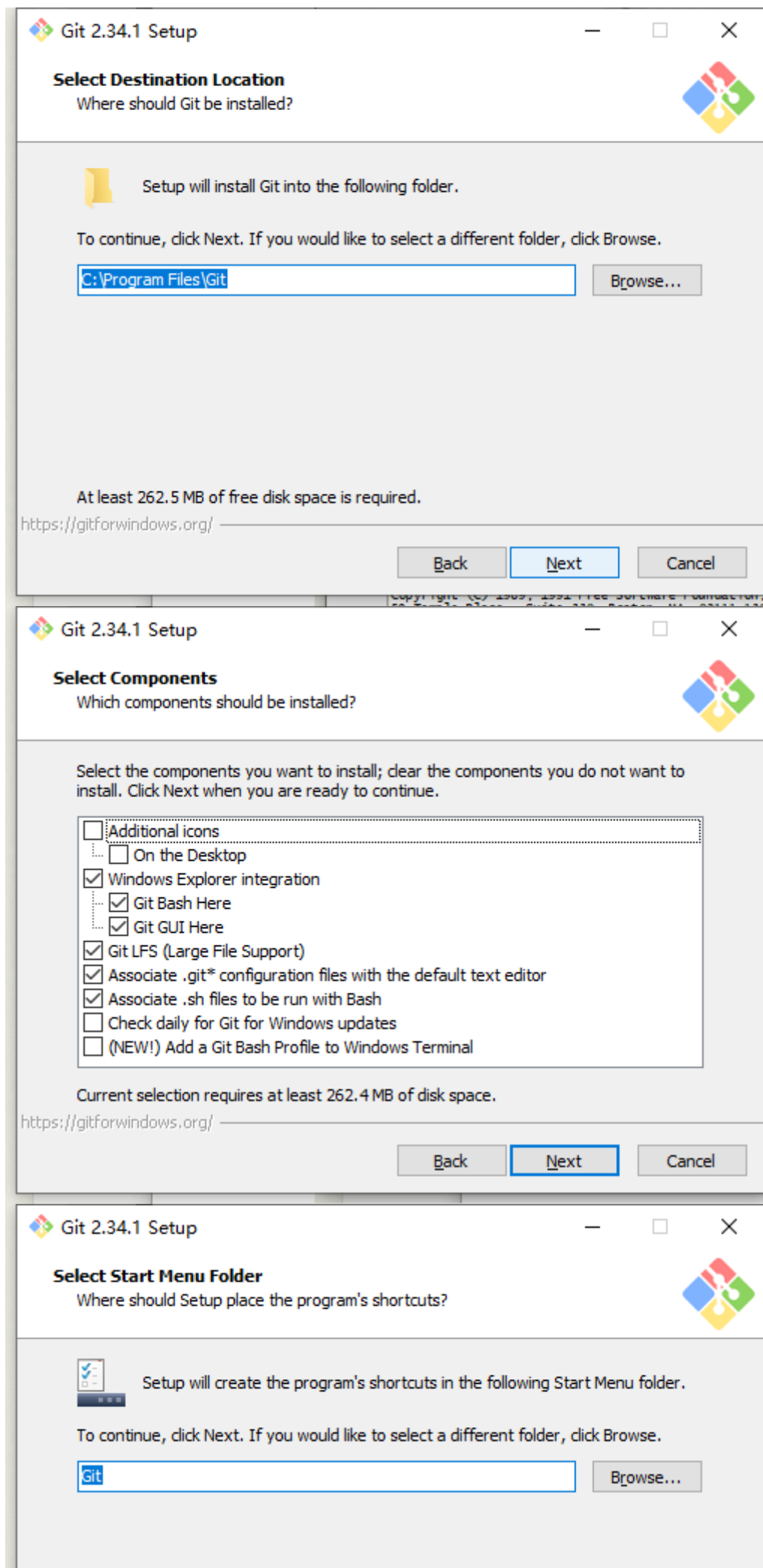
## 2.Github网站搭建

1. Git安装，git官网：<https://git-scm.com/>，点击Download for Windows



2. Git安装，按照图示步骤一直Next即可





☐ Don't create a Start Menu folder

<https://gitforwindows.org/>

Back

Next

Cancel

## Git 2.34.1 Setup

### Choosing the default editor used by Git

Which editor would you like Git to use?

Use Vim (the ubiquitous text editor) as Git's default editor

The [Vim editor](#), while powerful, [can be hard to use](#). Its user interface is unintuitive and its key bindings are awkward.

**Note:** Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.

**Note:** This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

<https://gitforwindows.org/>

Back

Next

Cancel

## Git 2.34.1 Setup

### Adjusting the name of the initial branch in new repositories

What would you like Git to name the initial branch after "git init"?

☒ Let Git decide

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project [intends](#) to change this default to a more inclusive name in the near future.

☐ Override the default branch name for new repositories

**NEW!** Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

main

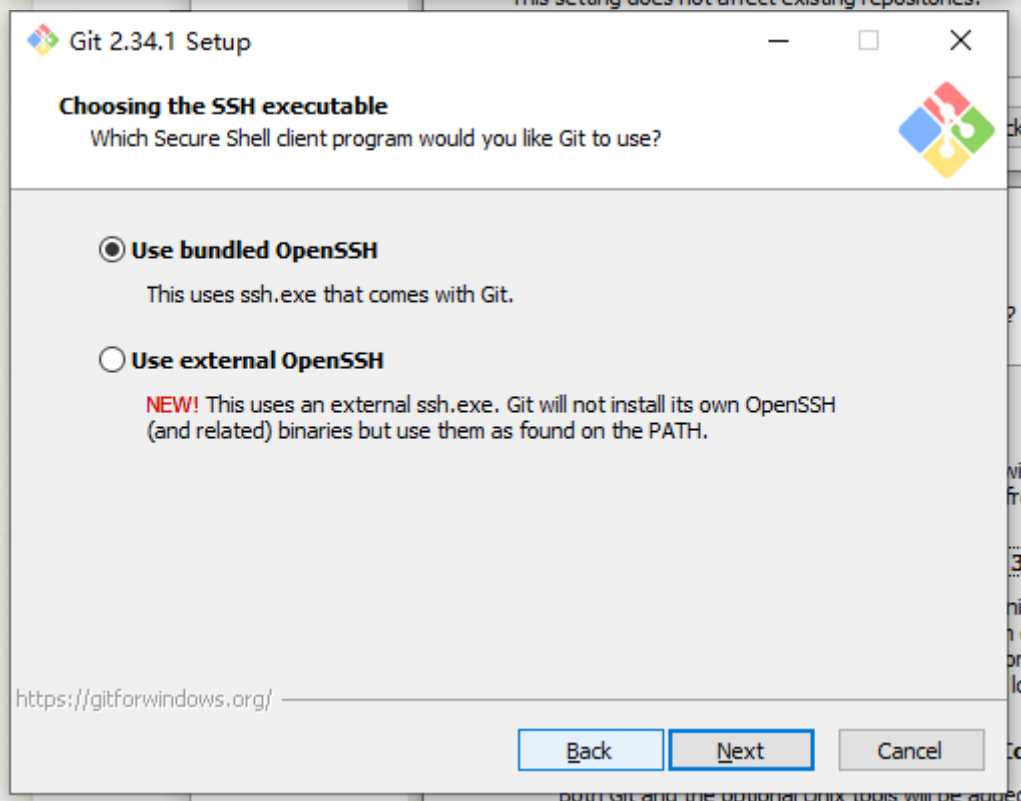
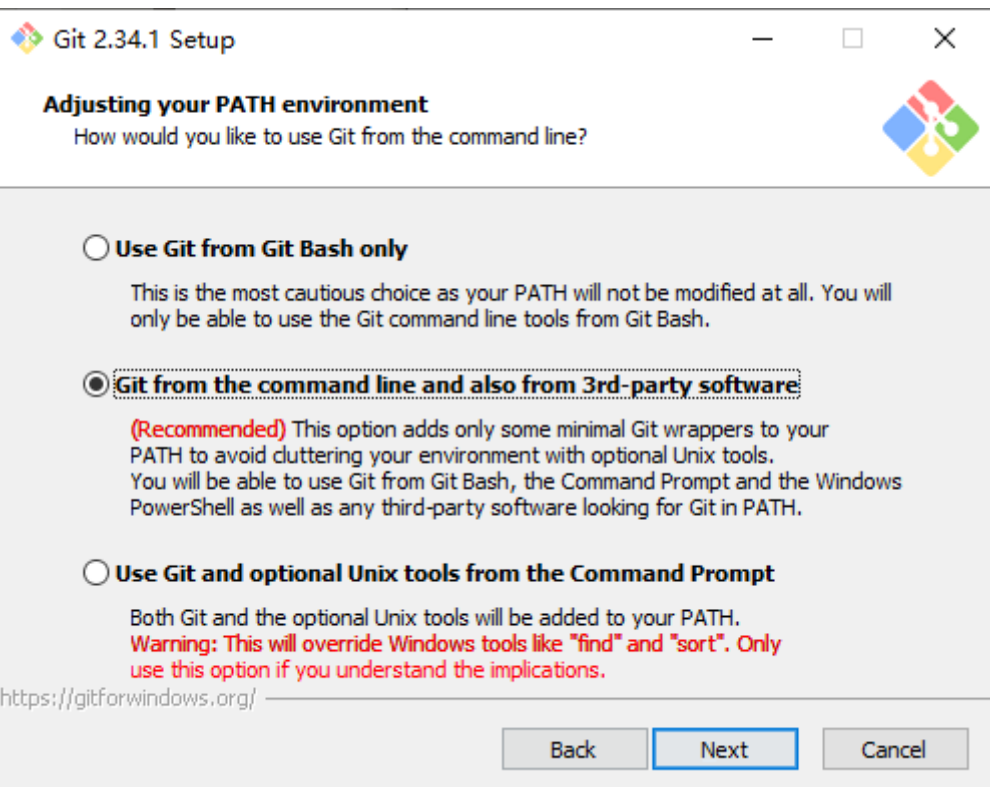
This setting does not affect existing repositories.

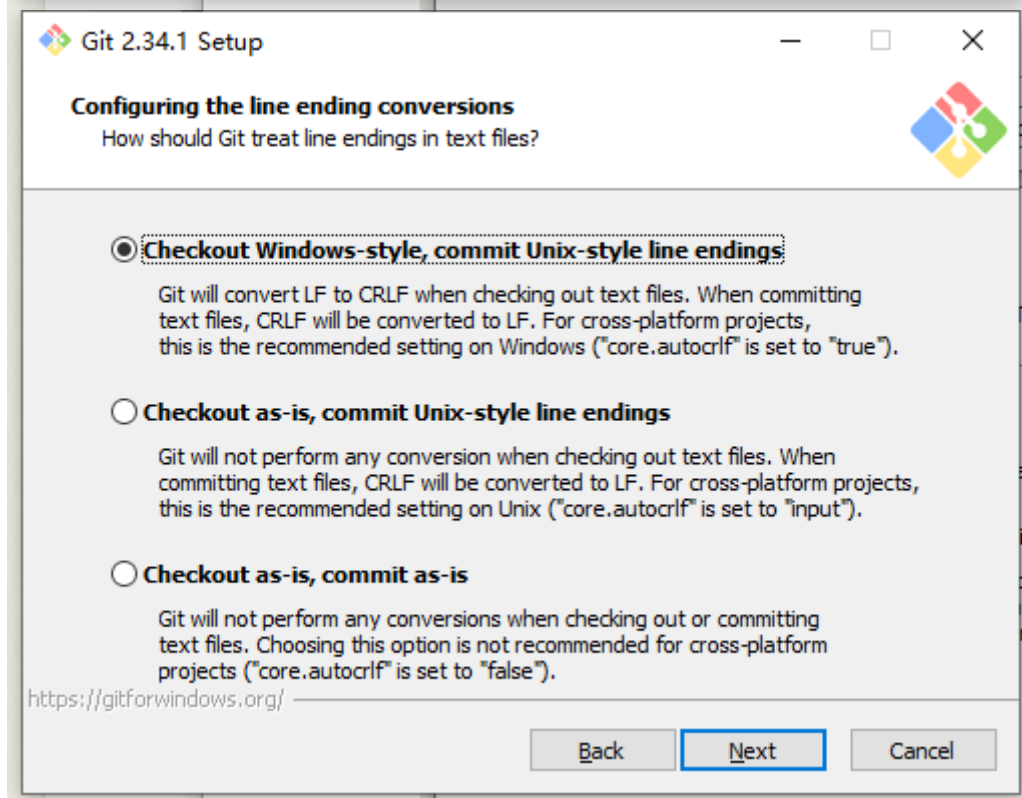
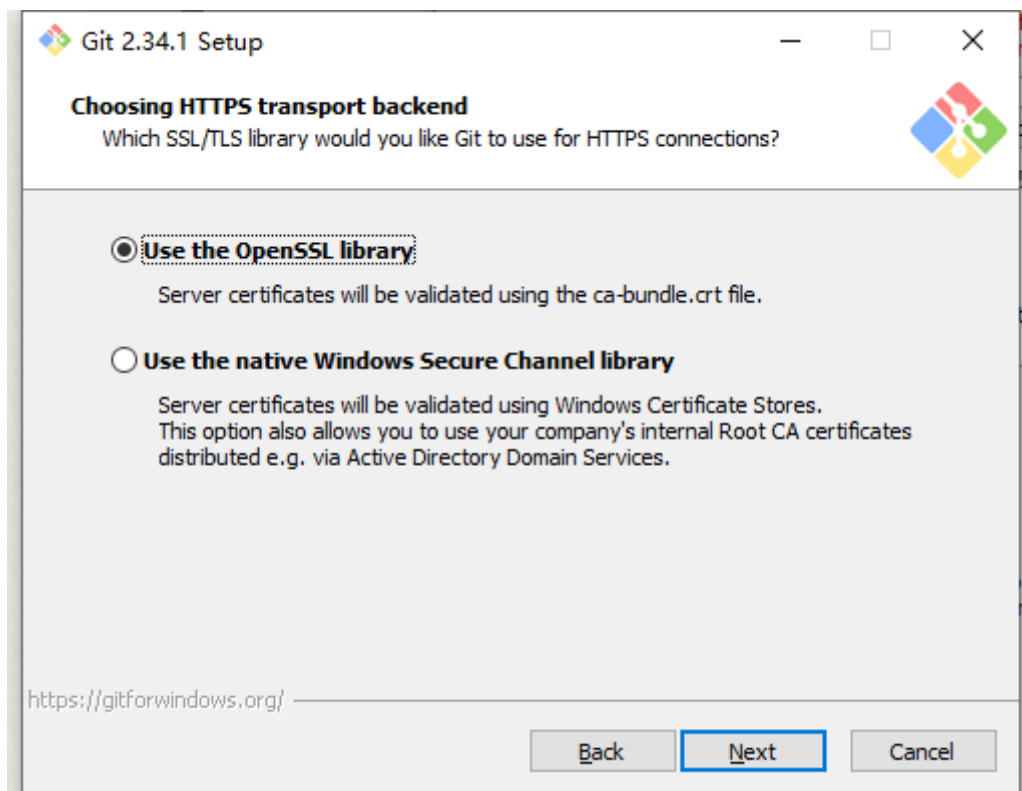
<https://gitforwindows.org/>

Back

Next

Cancel





Git 2.34.1 Setup

Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?

☒ Use MinTTY (the default terminal of MSYS2)

Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via ``winpty`` to work in MinTTY.

☐ Use Windows' default console window

Git will use the default console window of Windows (`"cmd.exe"`), which works well with Win32 console programs such as interactive Python or `node.js`, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

<https://gitforwindows.org/>

Back

Next

Cancel

Checkout as-is, commit as-is

Choose the default behavior of ``git pull``

What should ``git pull`` do by default?

☒ Default (fast-forward or merge)

This is the standard behavior of ``git pull``: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

☐ Rebase

Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

☐ Only ever fast-forward

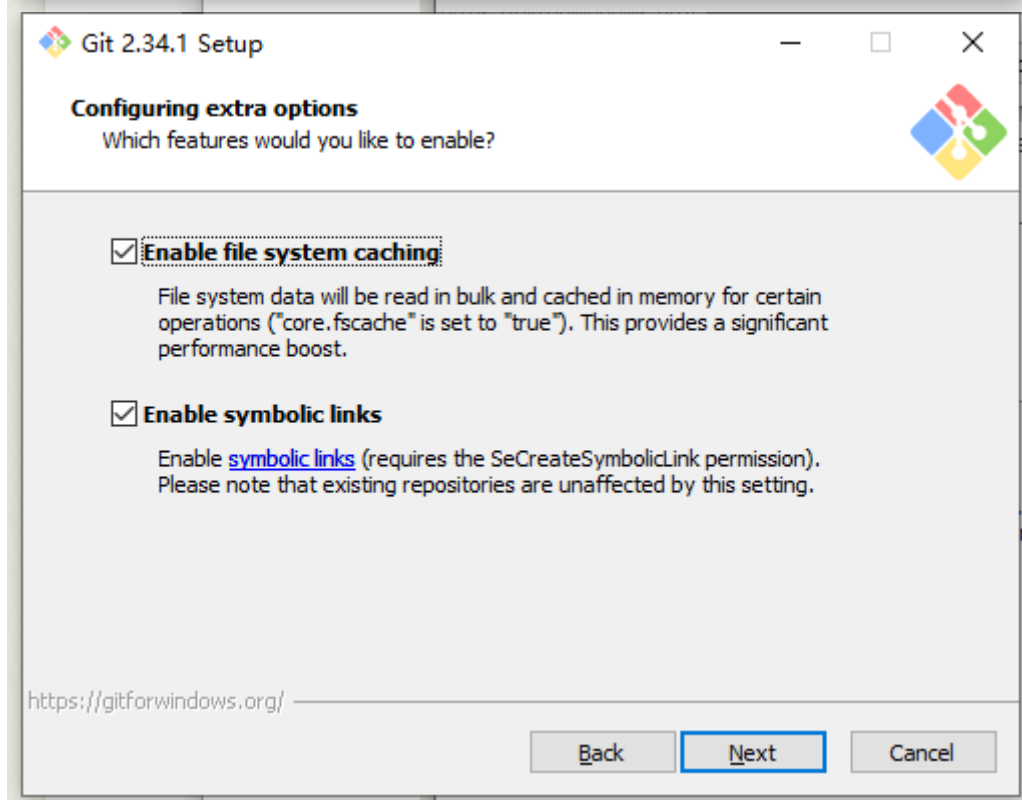
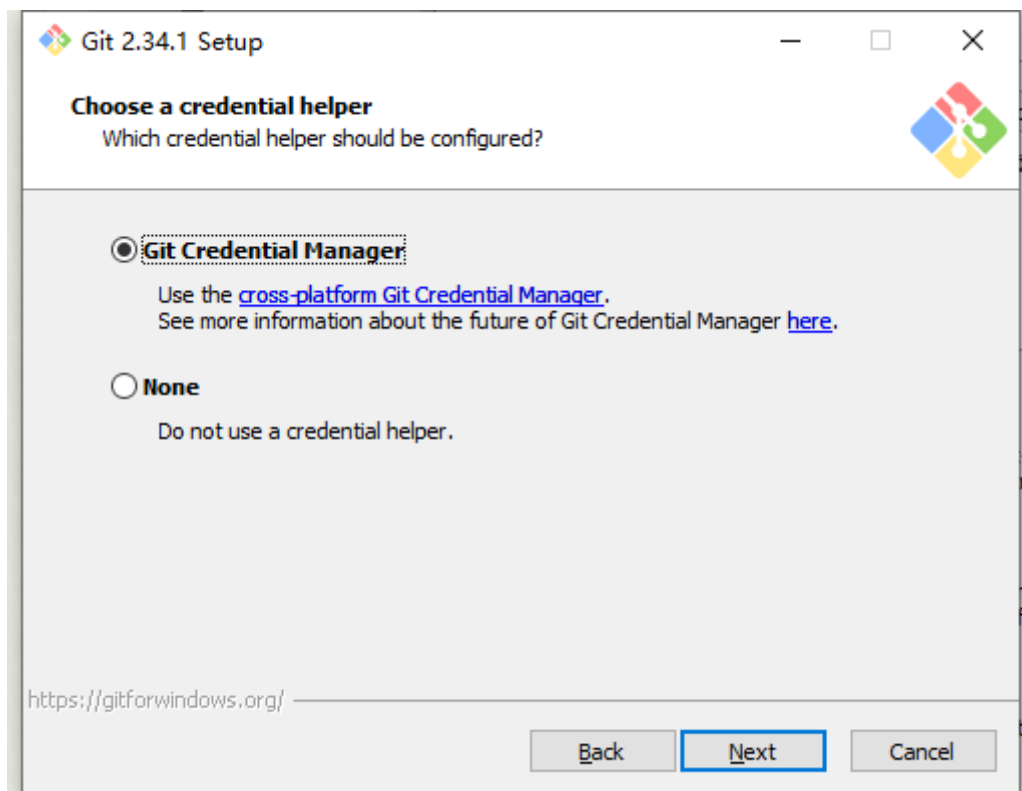
Fast-forward to the fetched branch. Fail if that is not possible.

<https://gitforwindows.org/>

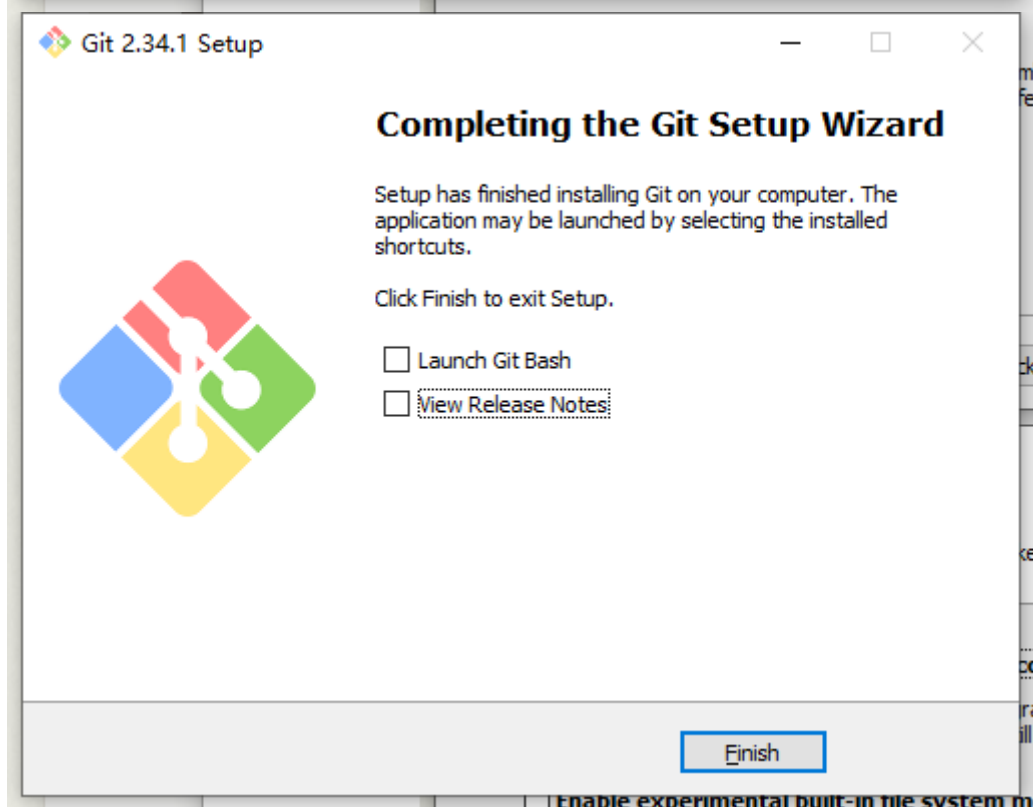
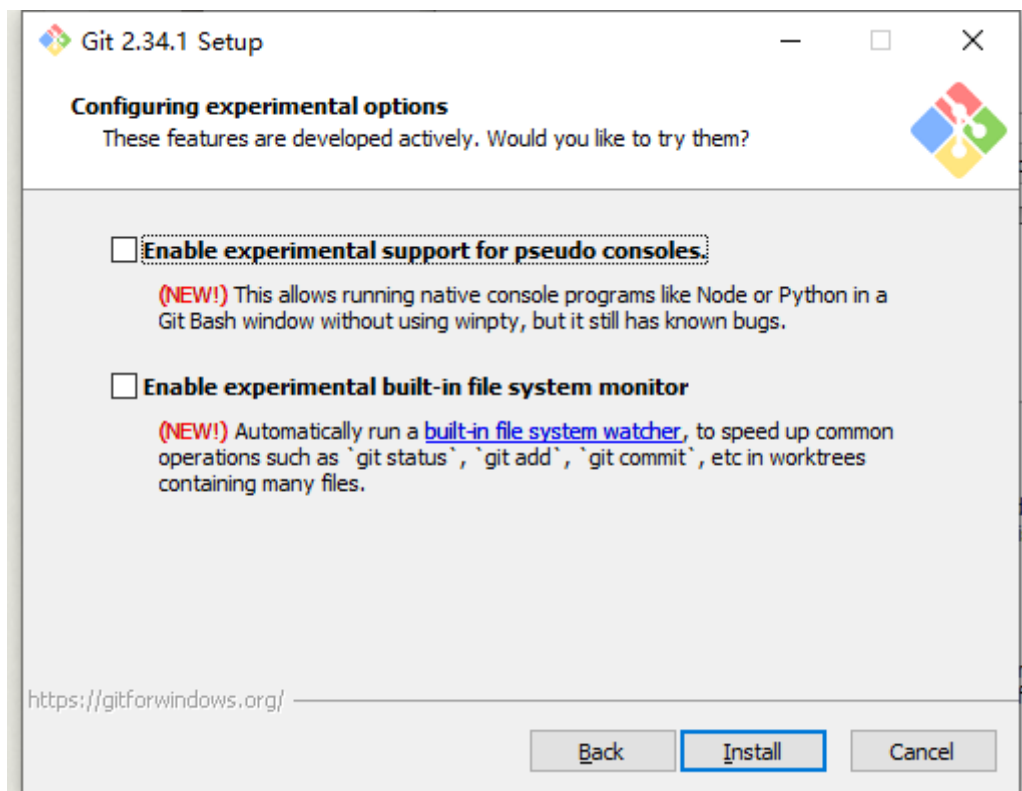
Back

Next

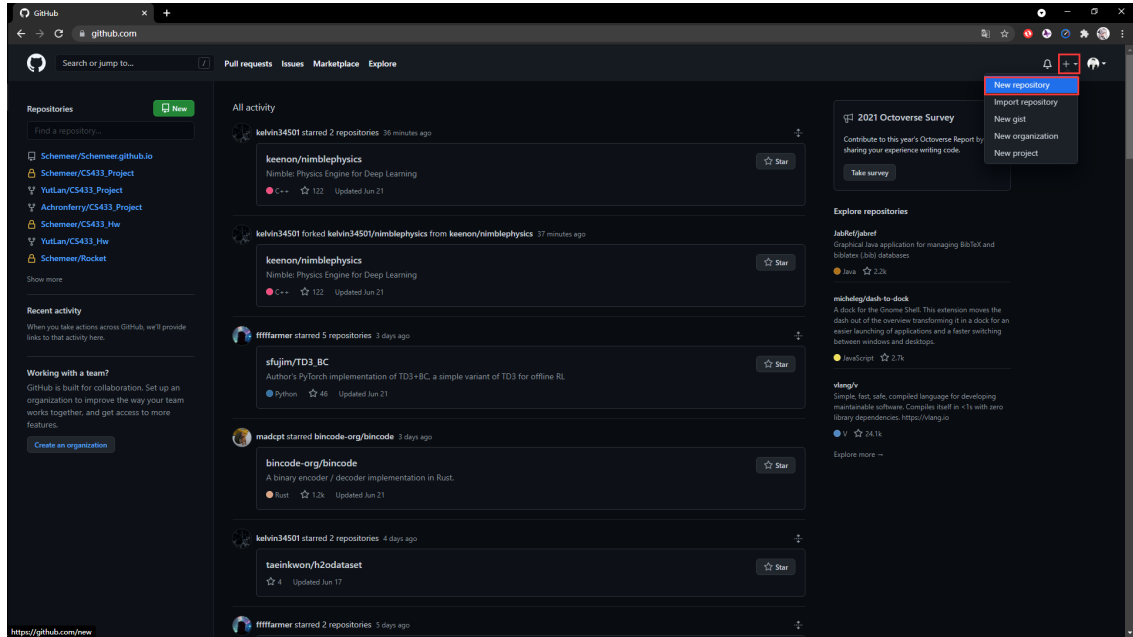
Cancel



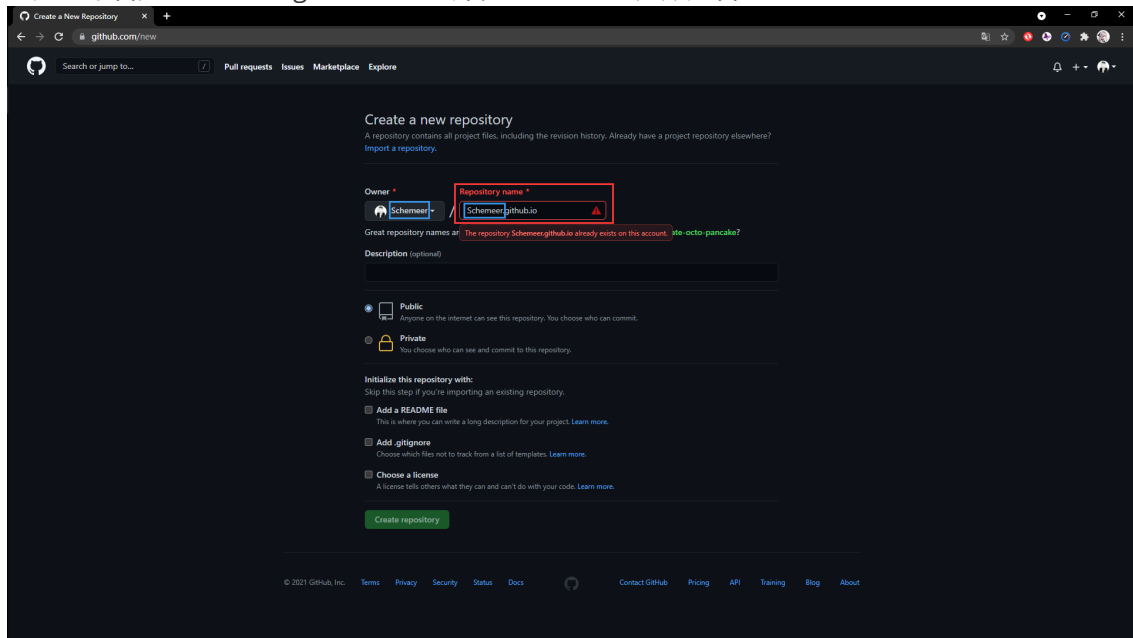




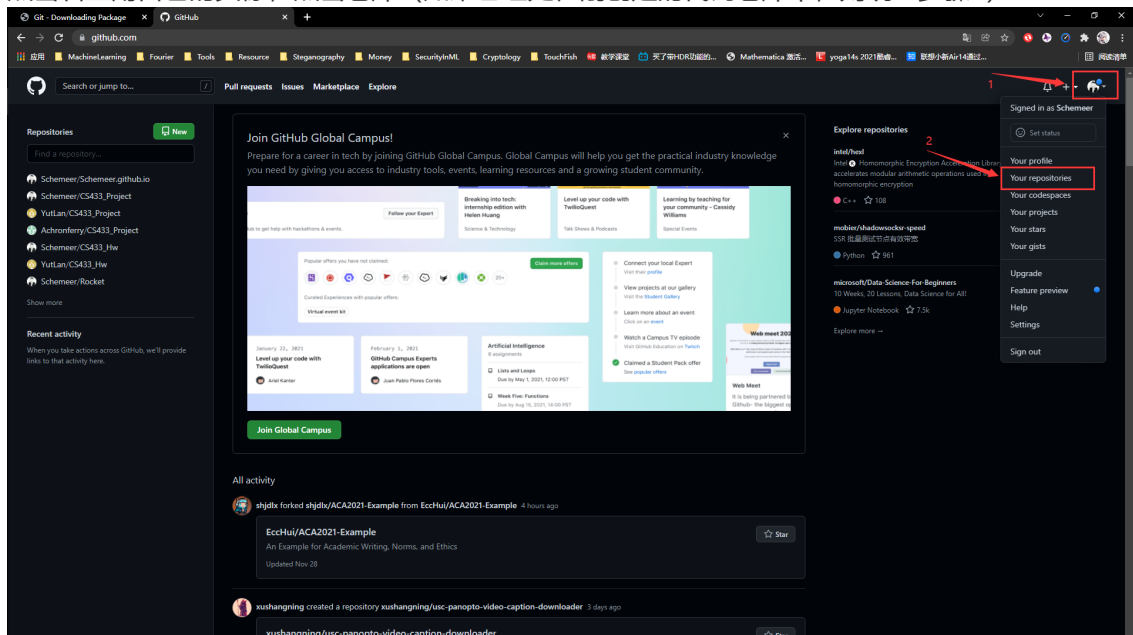
### 3. 登录Github，点击右上角加号，选择新建仓库



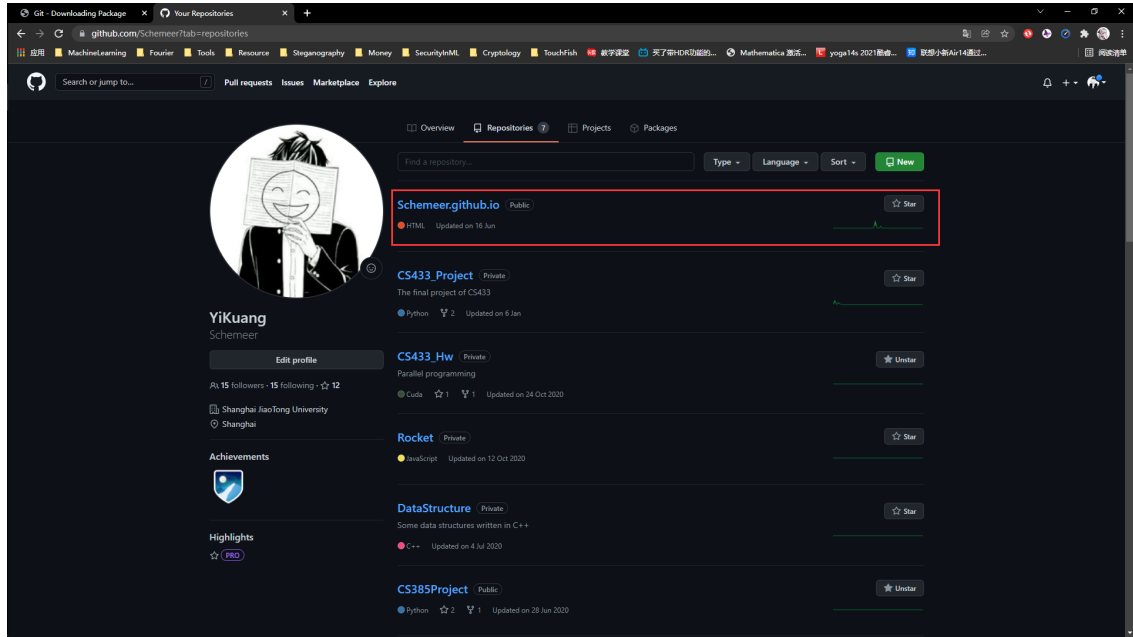
### 4. 创建一个名为username.github.io的仓库，username为账户名



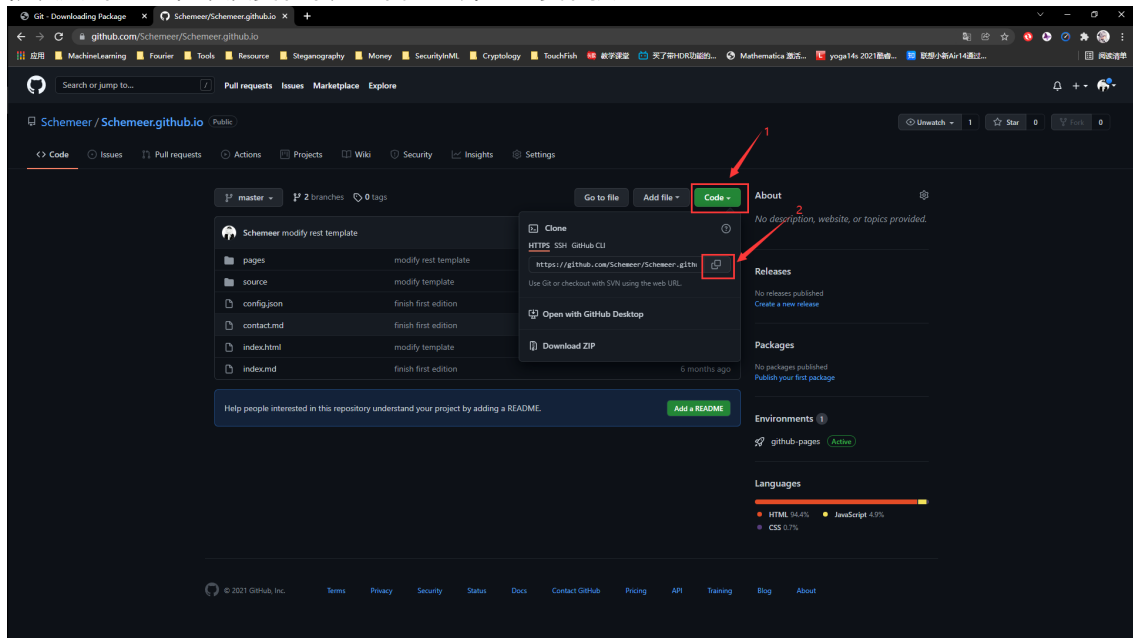
### 5. 点击右上角自己的头像，点击仓库（如果已经处在刚创建的代码仓库中，则跳至步骤7）



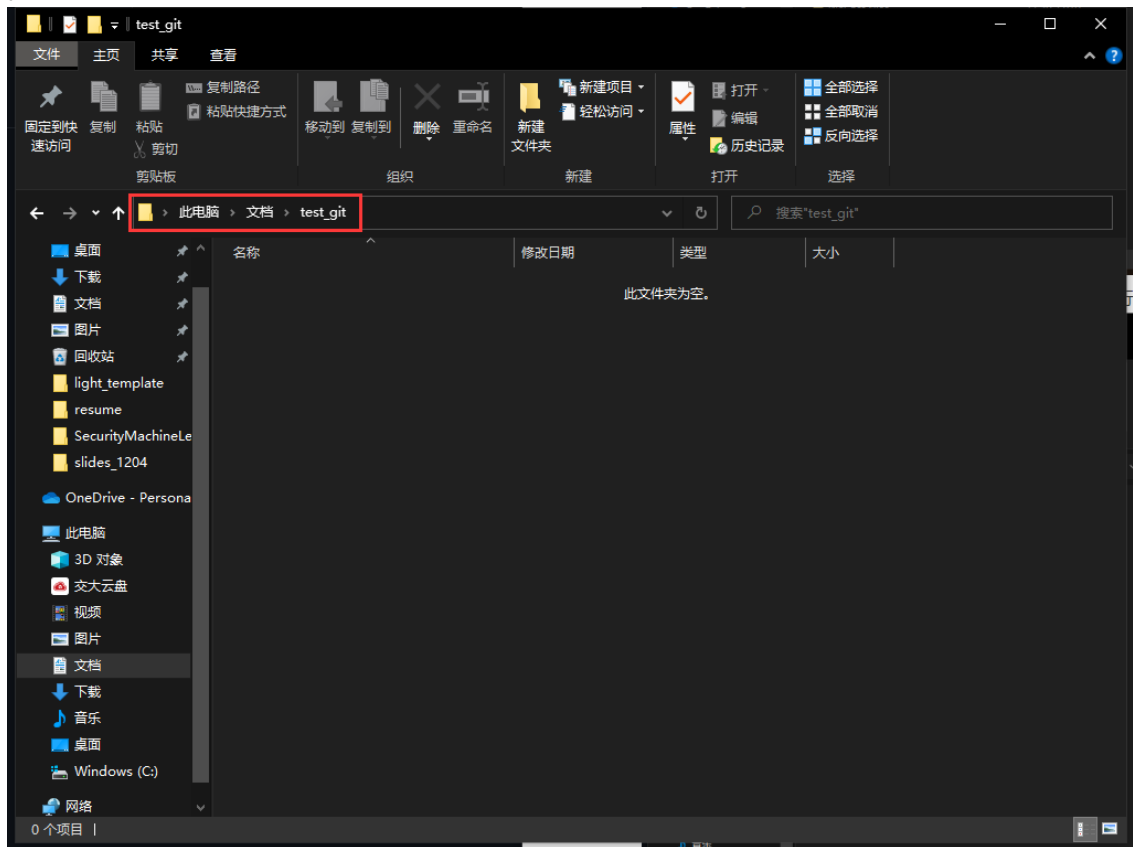
## 6. 选择刚刚创建的代码仓库



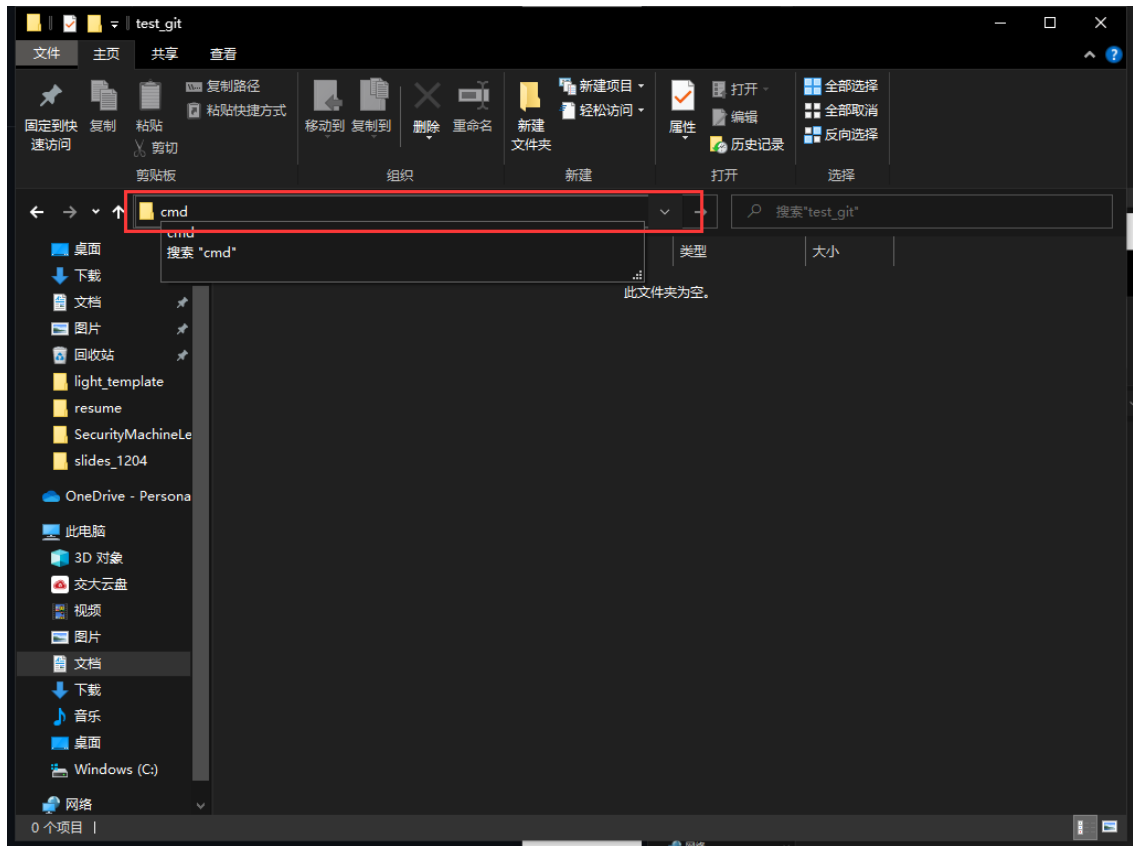
## 7. 依次点击Code，以及复制下方的代码仓库地址复制按钮



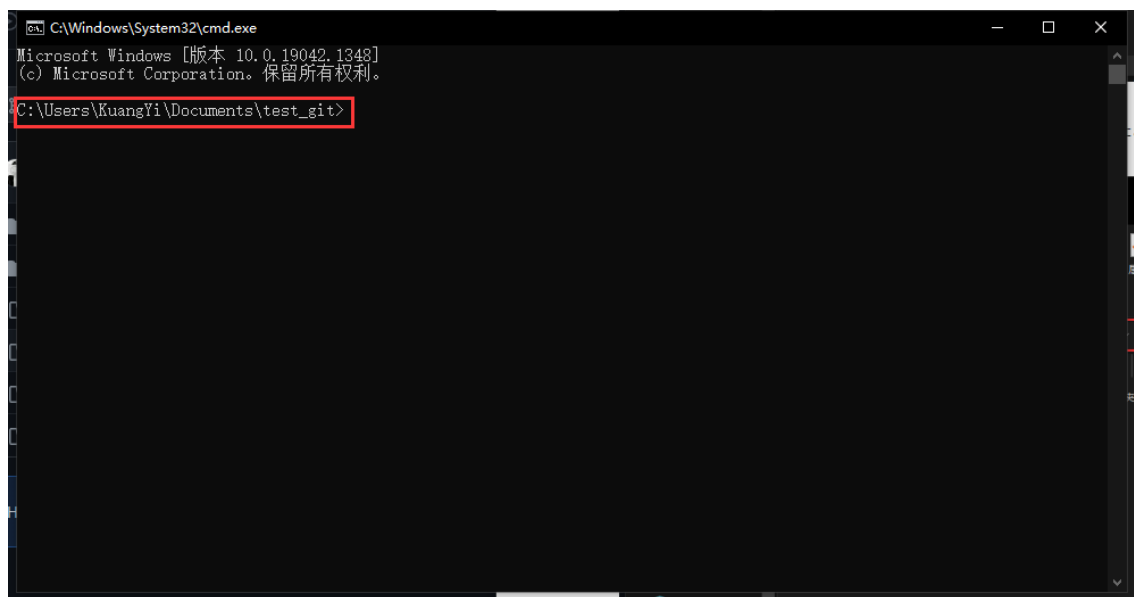
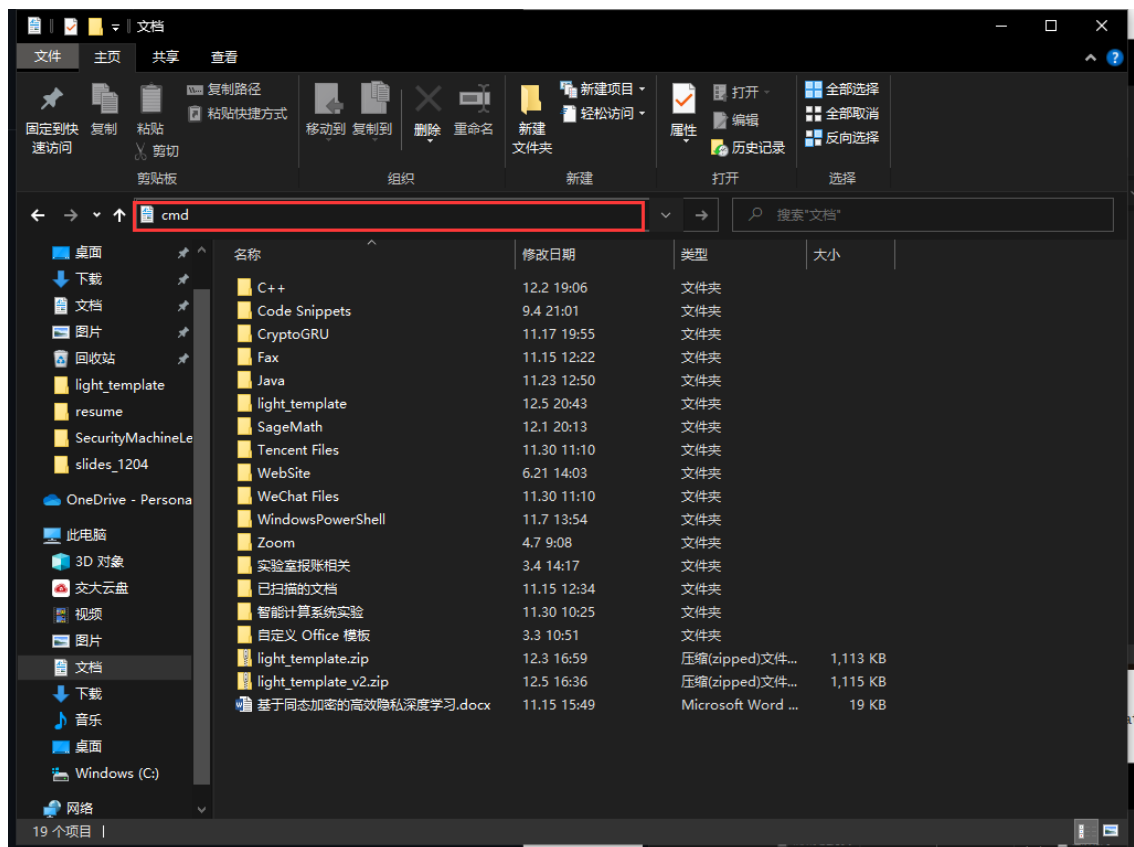
8. 在电脑中打开希望放置本地代码文件的地方，例如我选择将代码文件放在文档中的test\_git文件夹内



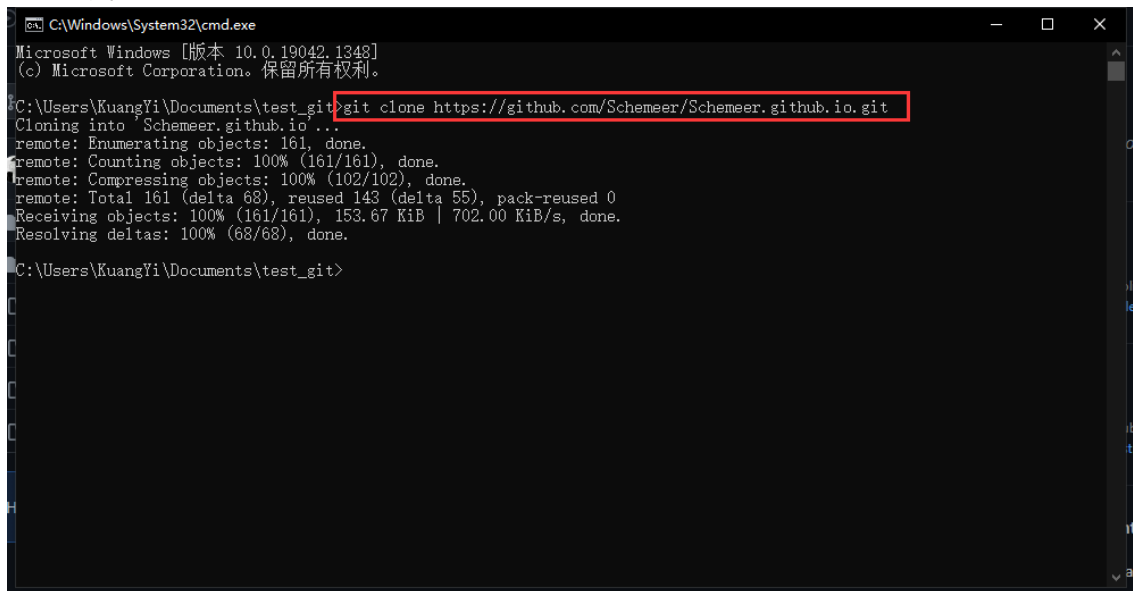
9. 单击地址栏



10. 输入 `cmd` 并敲击回车，此时会打开在当前路径下打开一个命令行



11. 在命令行中输入 `git clone repo_path` , `repo_path` 为步骤6中拷贝的代码仓库地址, 将该项目 clone到本地

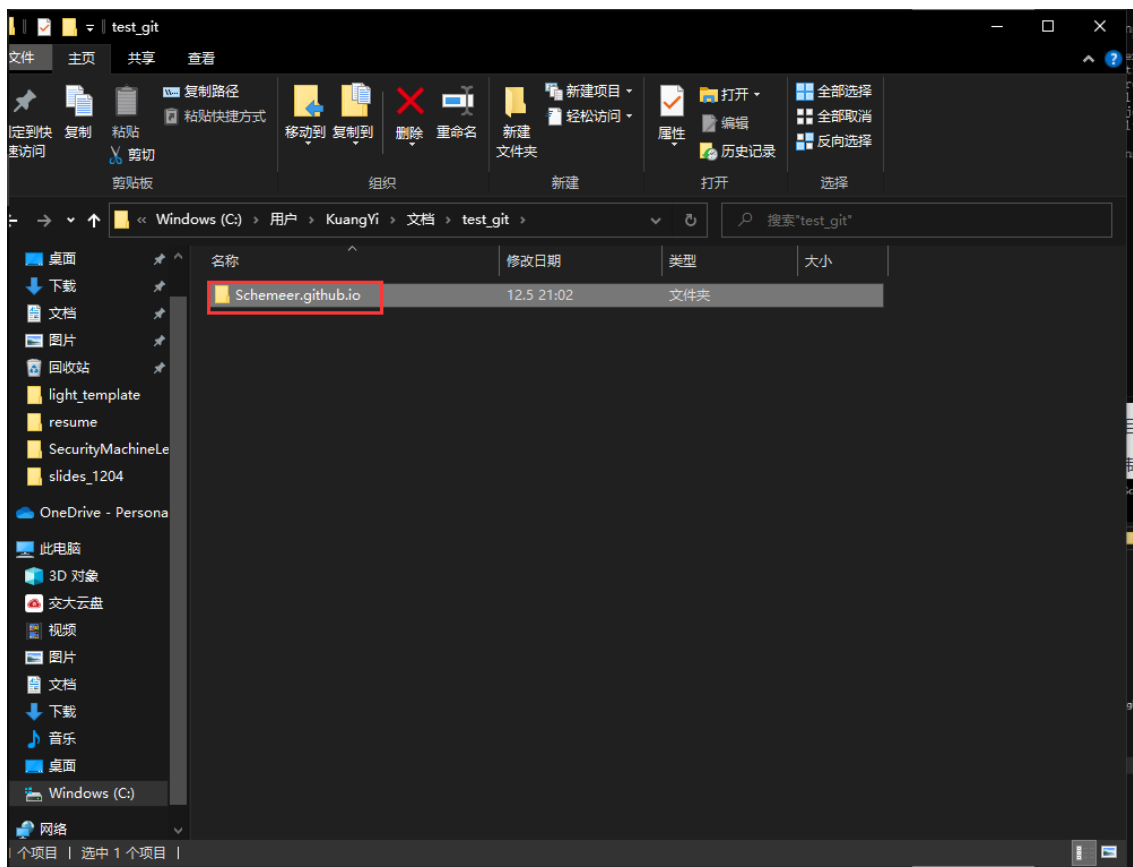


```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19042.1348]
(c) Microsoft Corporation. 保留所有权利。

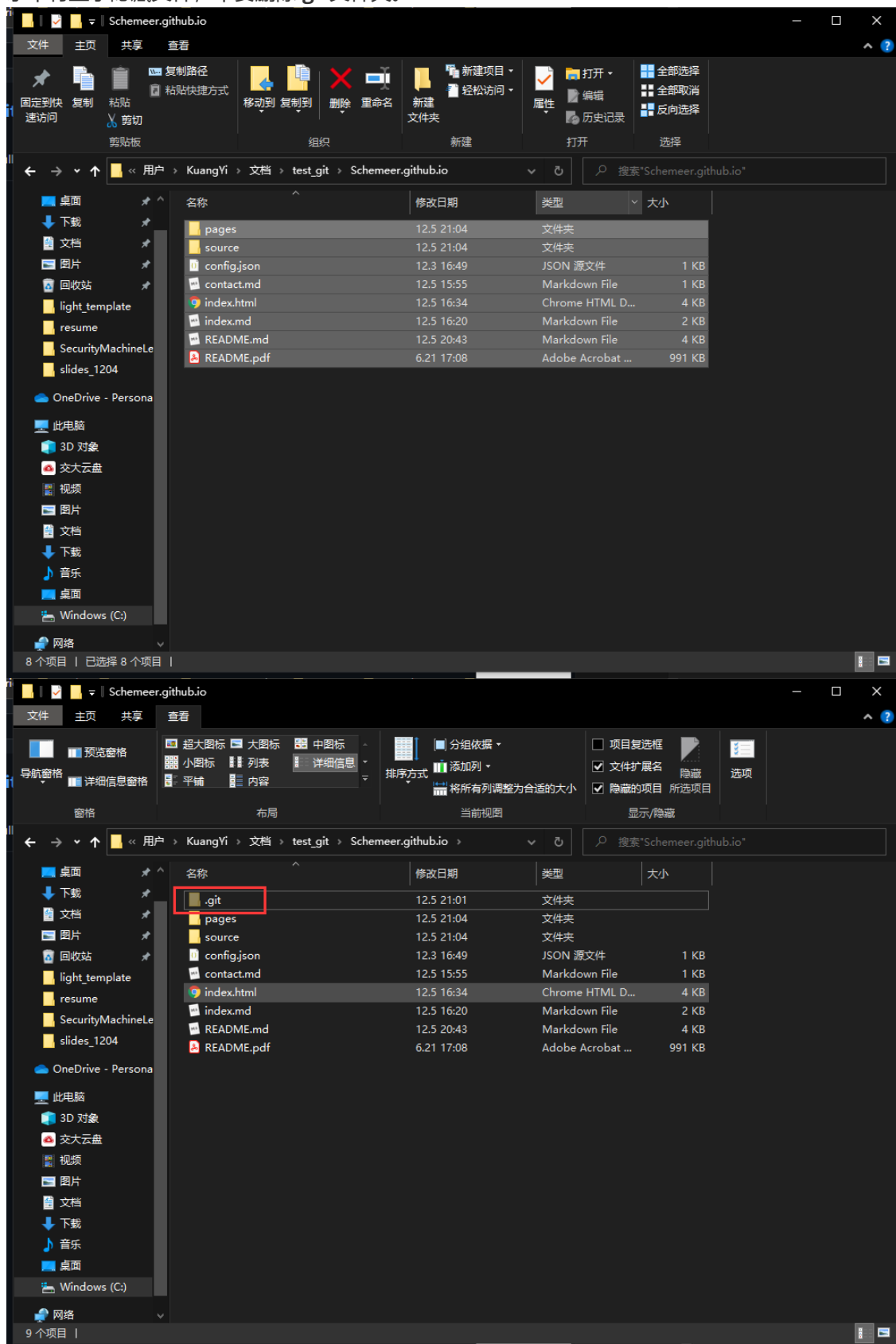
C:\Users\KuangYi\Documents\test_git>git clone https://github.com/Schemeer/Schemeer.github.io.git
Cloning into 'Schemeer.github.io'...
remote: Enumerating objects: 161, done.
remote: Counting objects: 100% (161/161), done.
remote: Compressing objects: 100% (102/102), done.
remote: Total 161 (delta 68), reused 143 (delta 55), pack-reused 0
Receiving objects: 100% (161/161), 153.67 KiB | 702.00 KiB/s, done.
Resolving deltas: 100% (68/68), done.

C:\Users\KuangYi\Documents\test_git>
```

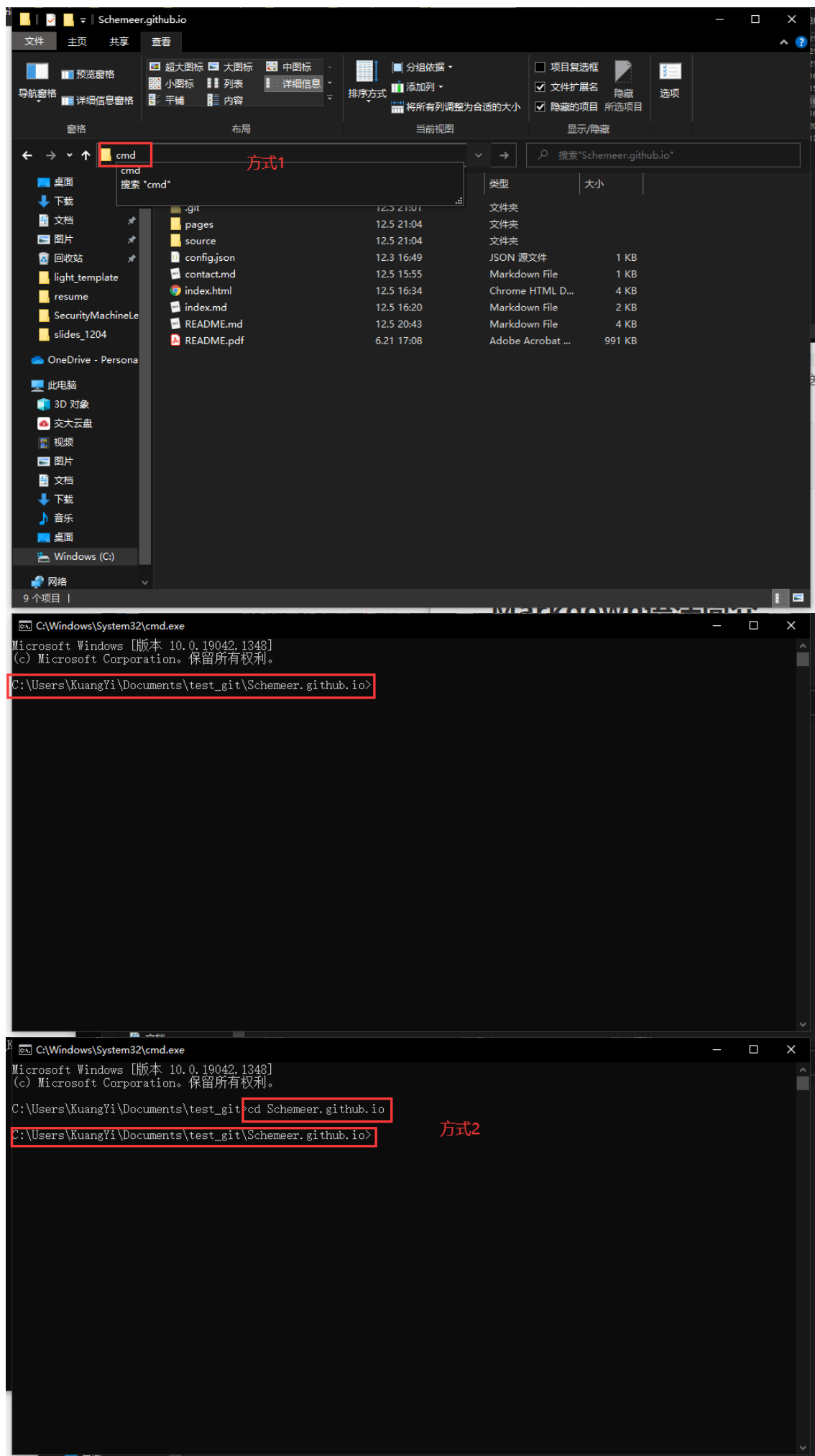
12. 此时刚才的目录下会出现一个 `username.github.io` 文件夹, 这就是刚刚创建的代码仓库



13. 双击进入该文件夹，将本项目(light\_template 文件夹内的内容)复制粘贴进仓库中，注意如果显示中有显示隐藏文件，不要删除.git文件夹。



14. 利用步骤10同样的方式或者在刚才的命令行中输入 `cd username.github.io`，将路径切换到仓库中



15. 此时，在命令行中输入以下命令，即可保存更改，提交并推送到远程仓库



这三行代码的含义分别为：

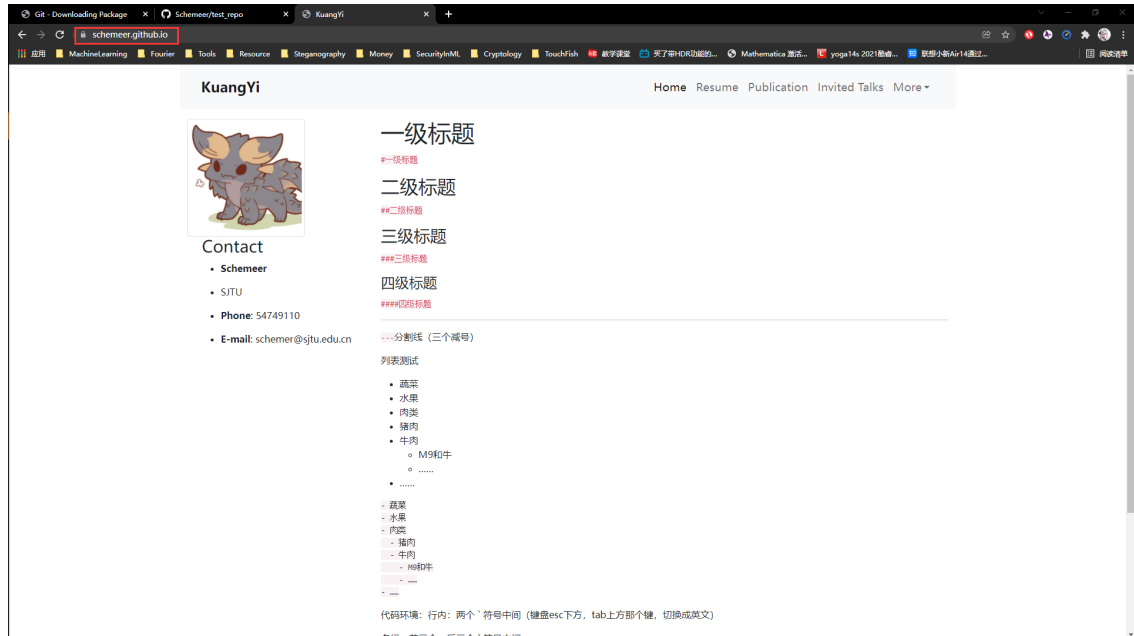
将所有文件添加到跟踪列表（即对所有文件检测其是否更新）

提交所有更新，并将这次提交命名为“first commit”

将本地的代码仓库推送到远程代码仓库（将github网站上的代码仓库同步为本地仓库的内容）

```
git add *
git commit -am "first commit"
git push
```

16. 此后等待系统刷新，便可通过 `username.github.io` 网址访问到自己的个人主页：



17. 个人主页维护与更新主要通过本地修改[文件说明](#)中的文件来实现，本地更新后，可通过如下两行代码将更新推送至远程仓库，其中command name可自己任起，一般用以简要描述此次更新的内容，然后等待服务器刷新即可看到更新后的主页。

```
git commit -am "command name"
git push
```

# Markdown语法简介

推荐使用[Typora](#)进行markdown文件的读写。

## 一级标题

#一级标题

## 二级标题

##二级标题

## 三级标题

###三级标题

## 四级标题

####四级标题

---

---分割线（三个减号）

### 无序列表

- class1
- class2
- class3
  - item1
  - item2
    - subitem1
    - .....
- .....

```
- class1
- class2
- class3
  - item1
  - item2
    - subitem1
    - .....
- .....
```

### 有序列表

1. class1
2. class2
3. class3
  1. item1
  2. item2
    1. subitem1
    2. ....
4. ....

```
1. class1
2. class2
3. class3
  1. item1
  2. item2
    1. subitem1
    2. ....
4. ....
```

代码环境：行内：两个 ` 符号中间（键盘esc下方，tab上方那个键，切换成英文）

多行：前三个，后三个 ` 符号中间

[超链接](#)

[超链接](https://www.baidu.com)

图片



![logo](./source/img/logo.jpg =100x100), 最后的=100x100是用于限制图片大小。