# GitHub and git 简明教程

## 前言

说白了就是一段简单的废话：

首先**简单阐述**一下GitHub以及git的作用以及其方便之处，

**GitHub**是一个远程仓库，其作用可以看作是在网络中找到一个锚钉，只要有**网络**你就可以通过**命令行**访问你的远程仓库！

（git命令行操作其实蛮重要的，因为我们学校的超算并没有一个图形界面，你本来也就只能通过命令行传输文件和管理文件：想象一下用命令行创建好多级文件多痛苦）

**Git**是一个本地的代码版本控制软件，如同它的名字一样，他能做的就是修改你的本地仓库，同时做好备注（git会要求你对你每一次提交修改的代码做注释）

到时候回头想到啥在写点啥吧！

文档作者：涂彦声 [tys0321@outlook.com](mailto:tys0321@outlook.com)

## 目录

[GitHub and git 简明教程](#_Toc136941670)

[前言](#_Toc136941671)

[目录](#_Toc136941672)

[安装git](#_Toc136941673)

[命令行conda安装git](#_Toc136941674)

[简述官网安装git其他方式](#_Toc136941675)

[GitHub](#_Toc136941676)

[实际操作](#_Toc136941677)

[配置本地git和远程连接到GitHub](#_Toc136941678)

[本地git配置](#_Toc136941679)

[本地git配置远程仓库](#_Toc136941680)

[GitHub SSH设置](#_Toc136941681)

[实际操作](#_Toc136941682)

[本地修改](#_Toc136941683)

[上传（下载）远程仓库](#_Toc136941684)

[通过vscode简化推送、拉取流程](#_Toc136941685)

## 安装git

Git是一个本地的**代码版本**控制软件，

每次提交代码都会要求你对修改进行注释，

同时支持回溯代码等功能。

优先放一个git他们官方网站的下载地址[**https://git\_scm.com/downloads**](https://git_scm.com/downloads)，

在这里能有各种系统（MacOs, Windows, Linux）安装git的方法！

### 命令行conda安装git

由于个人习惯问题，我喜欢用mimiconda管理我的各种python包和软件。

其中git就可以通过conda安装：

**conda search git # 查看conda库里面的git版本**

**conda install git # 默认安装最新版本git**

由于miniconda可以在MacOs, Linux, Windows上运行，

所以这个方法命令基本上可以在**全平台运行**。

### 简述官网安装git其他方式

Windwos：直接下载安装包安装

MacOs：通过包管理器HomeBrew安装

**brew install git**

Linux/Unix：

**apt-get install git #** Debian/**Ubuntu**

**yum install git #** Fedora, **CentOs7**

还有其他就不列了，学校超算系统为CentOs7，

但是好像需要管理者权限才能用yum安装东西（学生拿不到权限）

### GitHub

GitHub是一个**远程仓库**，

你可以通过网络访问你在互联网上的GitHub仓库。

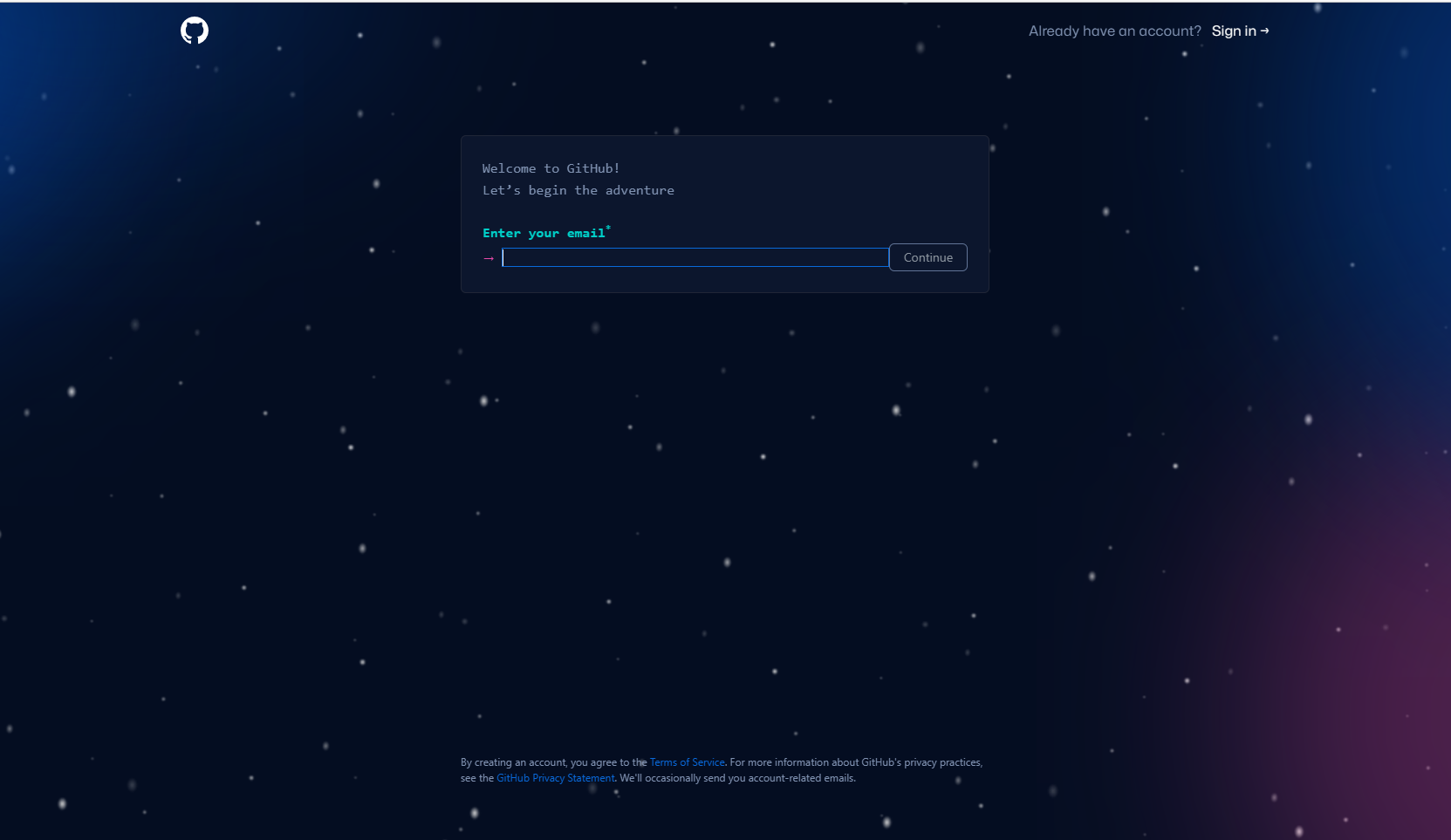
可以在**任意设备**复制你远程仓库中的代码或者上传

还是首先贴GitHub的官方网址：[**https://github.com/**](https://github.com/)

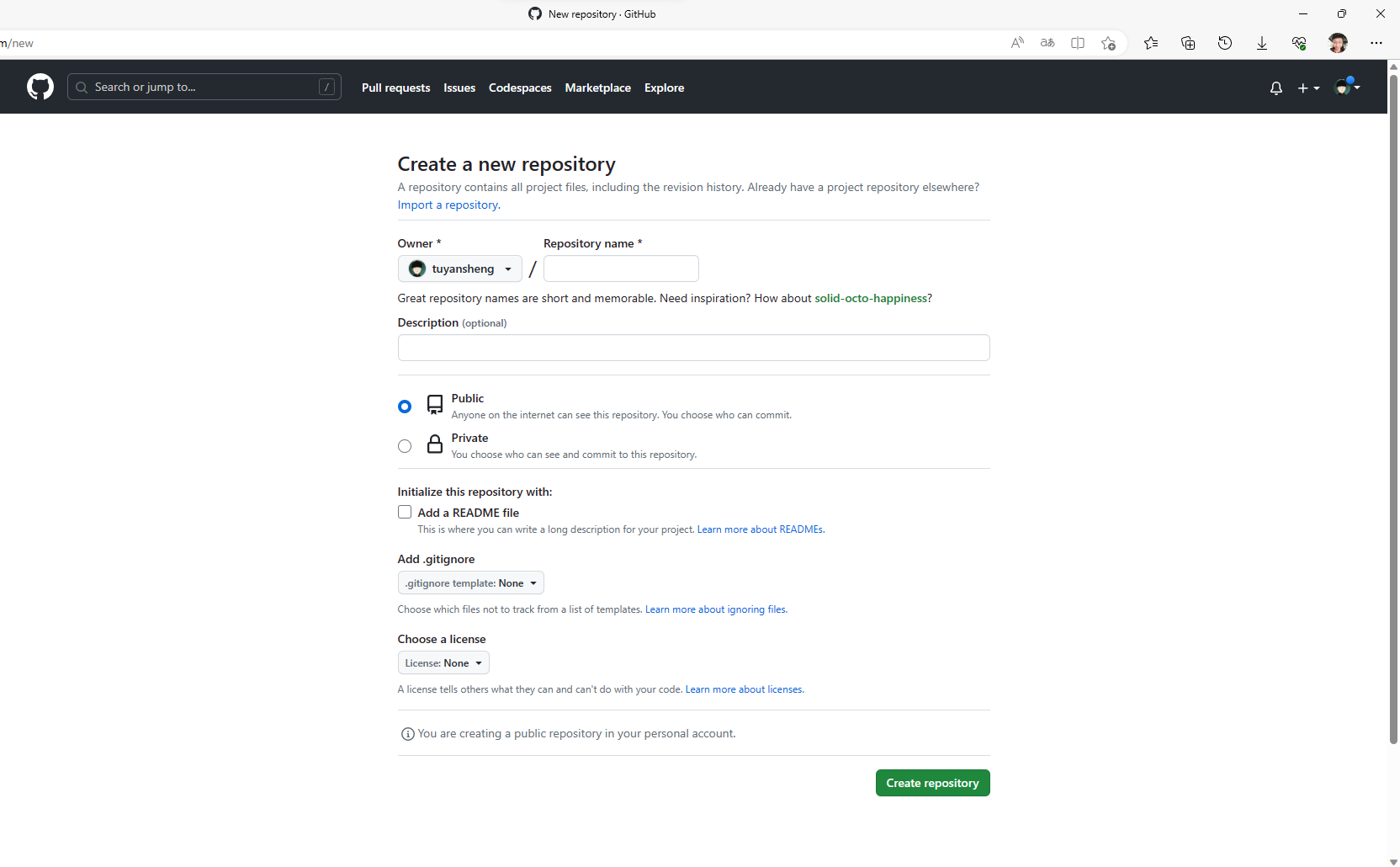
### 实际操作

首先需要创建你的GitHub账户

（如果以前没创建过的话）



其次创建一个新的库



需要输入你的远程仓库名字

（随便取就行，可以修改的）

然后可以设置这个仓库是否对公众开放等设置

接下来还有github如何连接到你的个人电脑的问题等。我会放在后面讲清楚

## 配置本地git和远程连接到GitHub

首先这里我假设我们本地都部署好了git，

同时我们已经有了GitHub的远程仓库。

**Git运行逻辑**：修改代码（workspace） -> 提交缓存到缓存区（index） -> 正式提交修改到本地仓库（local repository）-> 提交到远程仓库

（其实vscode安装一个插件这一套操作只需要两步！）

图示

描述已自动生成

### 本地git配置

最开始我们需要在我们的电脑里初始化我们的仓库

（最方便的就是直接选取我们的代码文件夹的第一级文件夹，它下面的所有文件都会被放在库里面）

**cd ~/your\_codespace\_path**

**git init**

然后我们要配置本地git的用户名和email地址

这一步是证明你的身份，

（例如你上传到远程仓库，而仓库又有多个作者同时修改，知道你是谁就很重要了）

**git config --global user.name “your\_name”**

**git config --global user.email “your\_email”**

（尽量使用和GitHub同样的名字和邮箱，我记得好像是通过邮箱地址判断是哪个用户的）

**这时候就可以说本地git的功能基本上都可以使用了，**

常用指令有：

**git add . # 提交修改的文件到缓存区（index）**

**git commit -a # 正式提交到本地仓库**，注意-a参数会调用vim，然后在vim打开的文件第一行输入你的修改注释，然后保存退出就行了

### 本地git配置远程仓库

首先要在**本地**设置git的远程信息，

让你的电脑知道你要从网络中的哪个远程仓库下载（上传）代码。

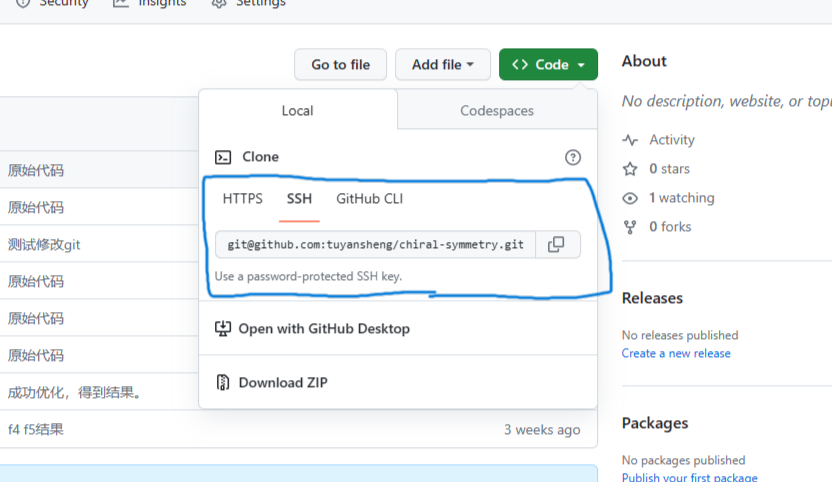
**git remote add <my\_remote\_name>** [**git@github.com:github\_name/repository\_name.git**](mailto:git@github.com:github_name/repository_name.git) **# 添加远程信息，其中<my\_remote\_name>是自己取的名字，用来快捷标记哪个远程仓库**

**git remote -v # 查询已有远程信息**

对于**GitHub**来说，

通常我们可以使用https://网址或者ssh来远程登陆GitHub

我们可以在GitHub上复制远程连接网址



但是非常**不推荐**使用https登陆

因为代理经常出问题！

git config —global —unset http.proxy

git config —global —unset https.proxy

这是代理出问题之后的重制代理的方法。

### GitHub SSH设置

如果使用SSH登陆远程仓库，

我们需要生成SSH密钥对，

（如果本地有请跳过这一步）

默认密钥保存路径为用户主文件夹中的隐藏文件.ssh中

**cd ~/.ssh # 其中id\_rsa.pub为公钥**

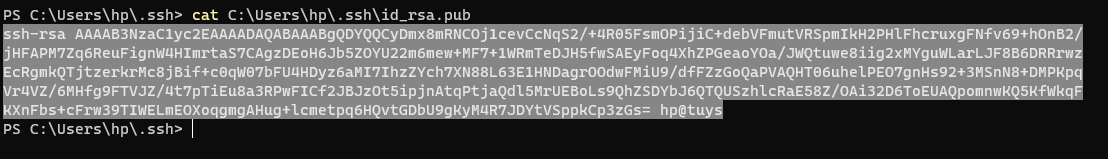
**ssh-keygen -t rsa # 生成密钥对，其中-t为设置密钥解密方式为rsa**

然后提取出id\_rsa.pub文件中的密钥

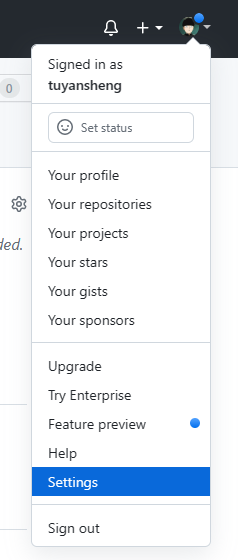
**cat ~/.ssh/id\_rsa.pub #这两种选择一种查看就行了**

**vim ~/.ssh/id\_rsa.pub**

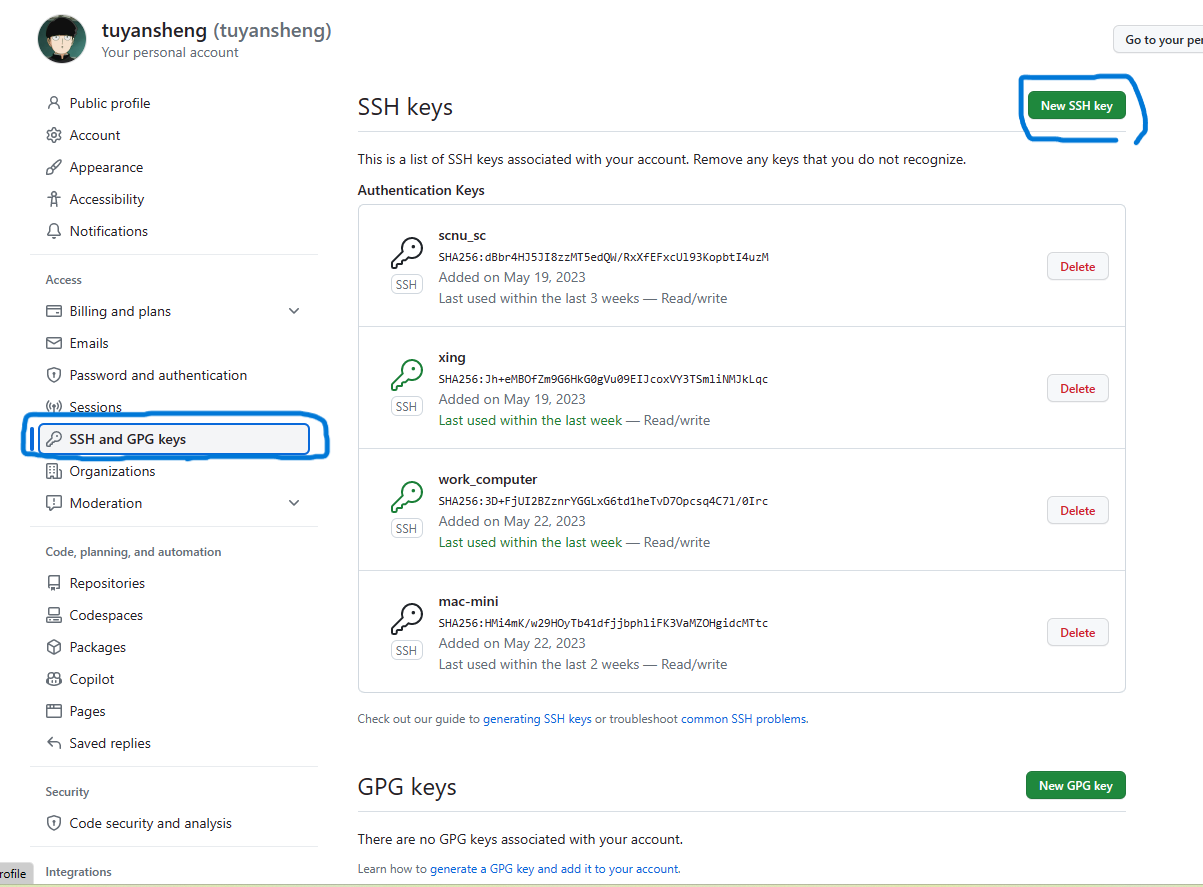
然后复制到剪贴板，

****

打开GitHub网页，



**右上角头像设置 -> SSH and GPG keys -> New SSH key**



最后加入你刚刚复制的公钥密码，确认

**这样你就有了远程登陆GitHub的权限**

（注意这个密钥对是依赖于你的个人电脑的，你上传了公钥，在你生成密钥对的路径还有对应的私钥用来“开锁”，所以不同的设备需要重新上传公钥到GitHub来获取登陆权限）

## 实际操作

下面这段简单展示怎么使用git以及同步到远程仓库

### 本地修改

根据我们修改代码的流程

首先我们在本地仓库**修改文件**（新增，修改，删除）

提交修改

**git add . # . 意味着保存所有修改到缓存区(index)**

**git status # 查看缓存区信息**

**git commit -a # 提交修改并保存到本地仓库**

（注意这里可能需要vim）

### 上传（下载）远程仓库

首先查询本地仓库信息

**git branch -v # 本地分支信息**

**git remote -v # 本地配置的远程仓库信息**

推送到远程仓库

**git push <remote\_name> <local\_branch>:<remote\_branch> # 输入你设置的远程名字，以及选择具体本地分支到远程分支**

拉取远程仓库

**git clone** [**git@github.com:github\_name/rep\_name.git**](mailto:git@github.com:github_name/rep_name.git) **#从远程仓库直接克隆下来（本地没有仓库）**

**git pull <remote\_name> <remote\_branch>:<local\_branch> # 从远程仓库更新代码**

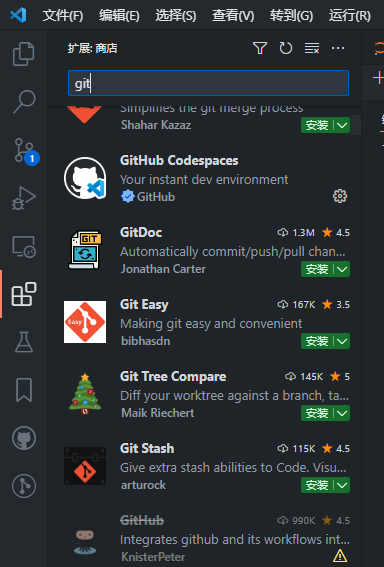
### 通过vscode简化推送、拉取流程

因为个人是vscode用户，很方便！

所以简单展示一下有多方便！

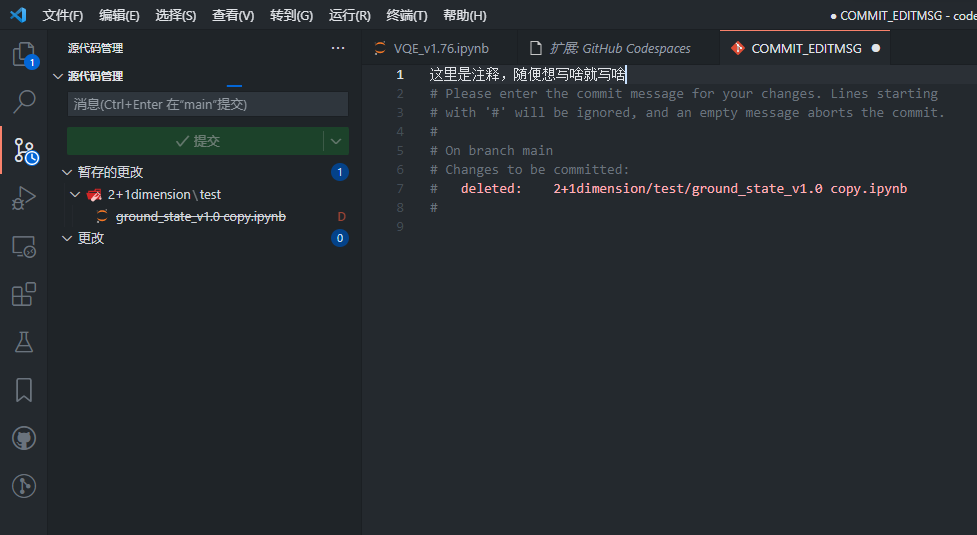
下载vscode啥的我都不讲了，git以及GitHub的配置依旧需要！

下载拓展GitHub Codespaces



然后我就只需要两步就能从本地上传修改到GitHub远程仓库啦！

**第一步：源代码管理 -> 提交 -> 写下注释 -> 右上角点击 对号 保存修改**



（在提交前如果你忘记了你修改了什么，点进左边的文件vscode还贴心的帮你做历史版本的对比来提示你）

**第二步：源代码管理 -> … -> 拉取，推送 -> 推送到 -> 选择目标远程仓库**



Ok，你通过vscode完成了一次**远程上传**

下载就直接**拉取**就行了

## 附录

git config —global —unset http.proxy

git config —global —unset https.proxy

git config —global user.name “tuyansheng”

git config —global user.mail “928235325@qq.com”

git add . 缓存修改（. 为）

git commit 提交修改

git status 查询状态

分支操作

重命名分支

git branch -m <old\_name> <new\_name>

删除分支

git branch -d <branch\_name>

切换分支

https://www.freecodecamp.org/chinese/news/git-switch-branch/

git checkout <branch\_name/SHA>

git checkout - #返回上一个分支

remote

删除remote信息

git remote remove <remote\_name>

修改remote名字

git remote rename <old\_name> <new\_name>