

GitHub 的官网: <https://www.github.com>

Github 的简易使用方法

如果你使用github仅仅是为了另一篇文档中提到的存储个人代码或者查找其他人的代码,我们完全可以避开Github的使用中极为麻烦的一部分的学习.即:Git的安装与常见指令学习,这一部分甚至涵盖了VScode的配置和远程Github仓库连接等一系列的问题。我们可以把重点只放在仓库的搭建和维护上。

(长时间不使用或维护的仓库Github官方会自动清理,在此之前你会收到一封来自San Francisco的邮件提醒你codespace的清理事宜,你完全拥有充足的时间对收到提醒的仓库进行一些小的修改来避免被自动清除)

Step1 注册

访问 <https://www.github.com>

点击"Sign up"注册

填写用户名、邮箱、密码

完成邮箱验证

Step2 个人资料的完善

单击右上角个人头像,点击 Your profile 即可设置自己的个人信息(同 Wechat,QQ 等常见社交媒体的个人信息设置)

The screenshot displays the GitHub profile of a user named 'Echo-GG' (username: Ech_0). The profile includes a circular avatar, the name 'Ech_0', and the location 'ShenZhen China'. Below the profile information, there is a section for 'Popular repositories' with a grid of repository cards. These cards include 'start_up' (a simple start), 'blog', 'Echo-GG.github.io', 'Algorithm-learning' (Notes about algorithm learning), 'SZU-Introduction-to-Computer-System-Organization' (SZU Introduction to Computer System Organization), and 'Github-guidance' (A github guidance specially designed for szuers). A 'Contribution graph' shows '118 contributions in the last year' with a calendar grid where green squares represent contributions. On the right side, a dropdown menu is open, showing options like 'Set status', 'Your profile' (highlighted with a red box), 'Your repositories', 'Your Copilot', 'Your projects', 'Your stars', 'Your gists', 'Your organizations', 'Your enterprises', 'Your sponsors', 'Try Enterprise', 'Feature preview', 'Settings', 'GitHub Website', 'GitHub Docs', 'GitHub Support', 'GitHub Community', and 'Sign out'.

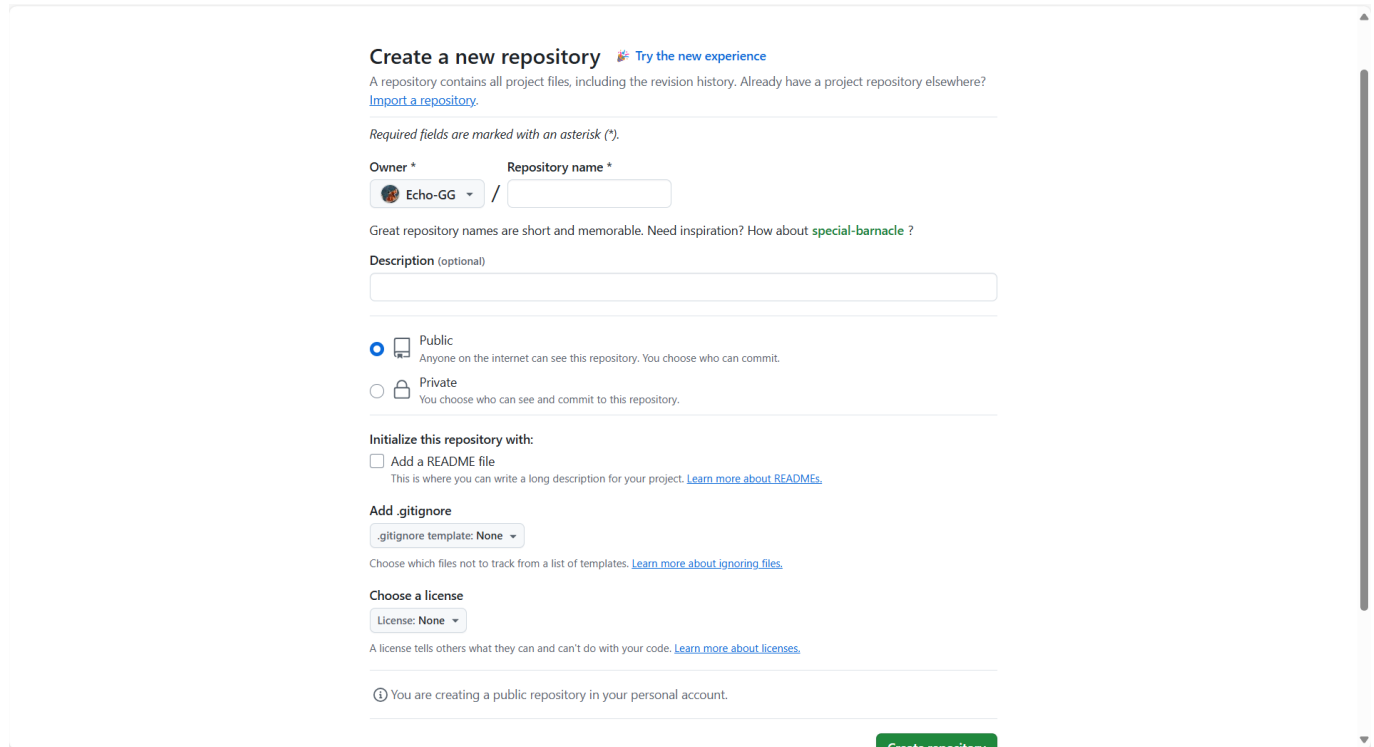
可以看到这张图的下半部分对应的深浅绿色小方块就是自己的contributions.这也是另一个我们要使用Github的原因: 即用户在Github上提交的所有代码,都会被系统自动记录,根据贡献量的多少和提交贡献的日期,形成深深浅浅的绿色小方块代表对应日期的代码贡献量(绿色越深代表当日提交的代码量越多)。这一贡献量的可视化呈

现,一方面能够提升自己在编写提交代码时的成就感与获得感;另一方面,在未来的求职简历中,这部分数据的呈现被HR尤为看重和考量(相当于完整的记录你的计算机学习生涯).

再次声明:由于这部分我们只简要介绍一些基础的法,故不涉及Git指令的部分,而是注重Github平台的使用介绍.

Step3 创建你的第一个仓库(Repository)

单击上方从左至右第二个选项"Repositories",点击右上方的绿色按钮"New"即可完成新仓库的创建.



Create a new repository [Try the new experience](#)

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

Required fields are marked with an asterisk (*).

Owner * Echo-GG / Repository name *

Great repository names are short and memorable. Need inspiration? How about [special-barnacle](#)?

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:

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

Add .gitignore
.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

① You are creating a public repository in your personal account.

[Create repository](#)

一般地我们需要给自己新建的仓库取一个名字,然后勾选Public即可(Private的仓库会额外收取费用,这也是github的主要收入来源之一)

需要注意的是隐私代码不宜选用public公开出来;

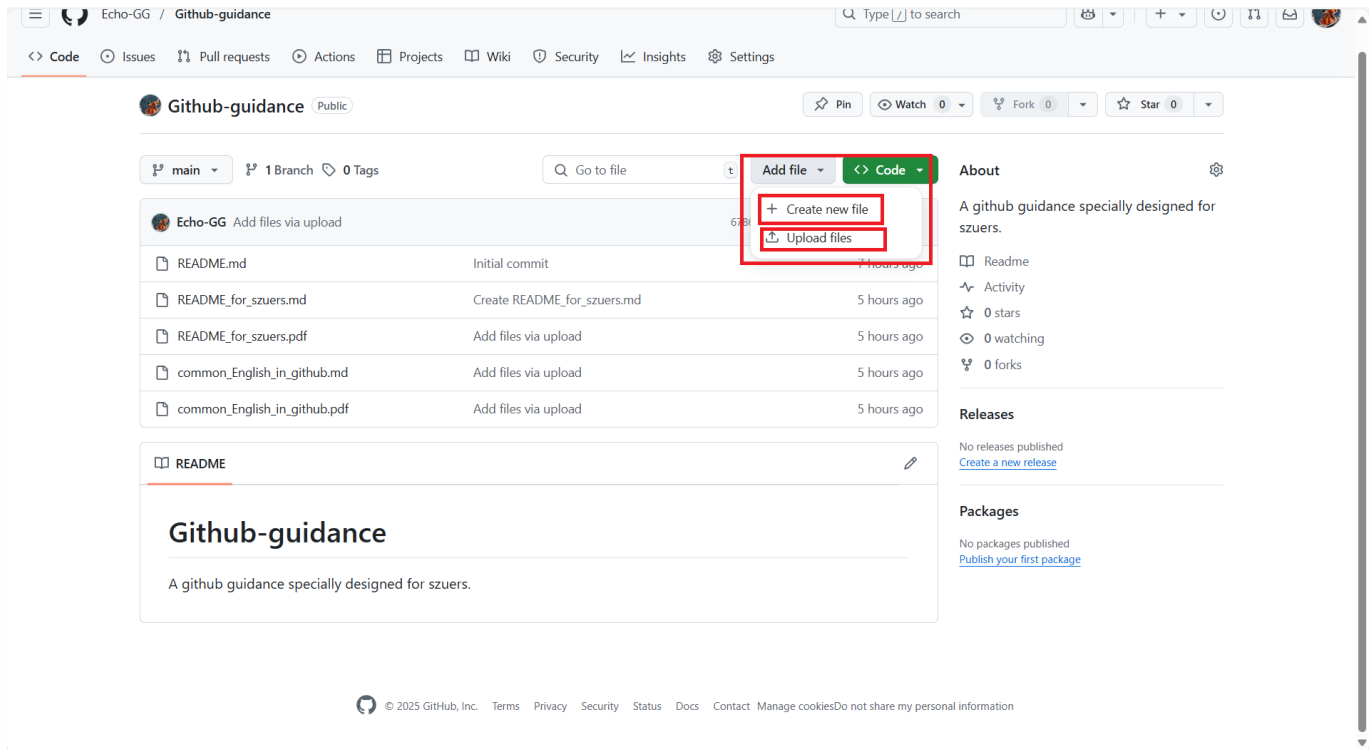
private的含义是除非你手动邀请或添加特定人员,否则你的这个私有仓库不会被任何其他人直接访问到,而且使用github中的search功能也无法搜索到有关你的私有仓库的任何信息.

我们推荐在创建仓库的同时自动添加一份README.md file

有关于 README: 这会是你今后使用github的过程中经常接触到的一种文件.几乎所有的开源项目都会配上一份 README file.在这个文件中,contributor 通常会对该项目/仓库进行一定程度的介绍和说明.我们通过 README file 可以快速获取这部分代码的基本用处和相关信息.

从这里我们不难看出一份好的 README file 的重要性.事实上,我们也可以主动为github上的知名开源项目贡献 README file.一份好的README file 有时候比代码本身更让人看重.

创建好仓库后即可向仓库中添加文件了.



添加文件的方式主要有两种:

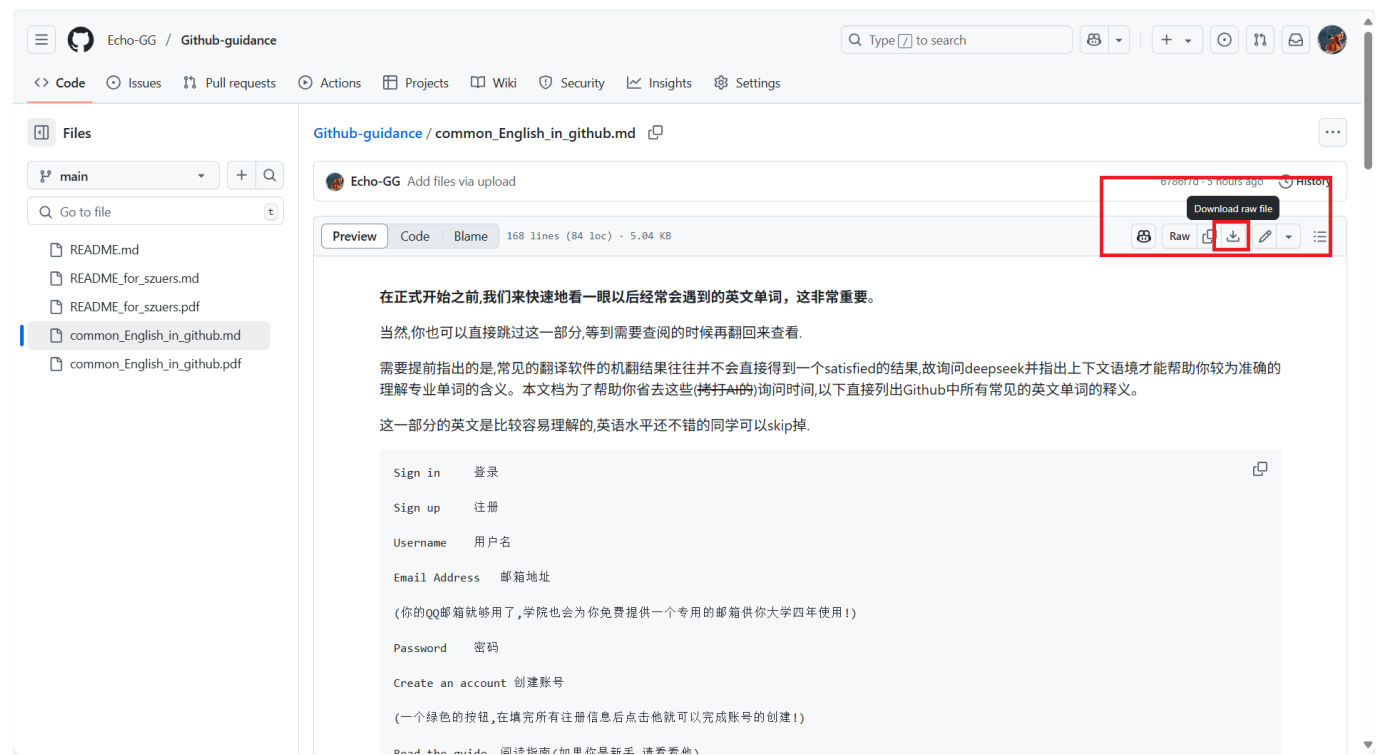
1. Create new file

这种方法主要适用于在线编写或者拷贝一些不太长的代码,例如README file 等.特点是支持实时修改和在线编辑.

2. Upload files

这种方法主要适用于本地已写好的文件向Github这一远程代码托管平台上传.我们可以采用直接拖动桌面文件夹的方式,将写好在一个文件夹里的代码,图片等项目组成文件一并上传到Github上,同理在Github上也会出现一个具有完全相同层次结构的对应的文件夹.

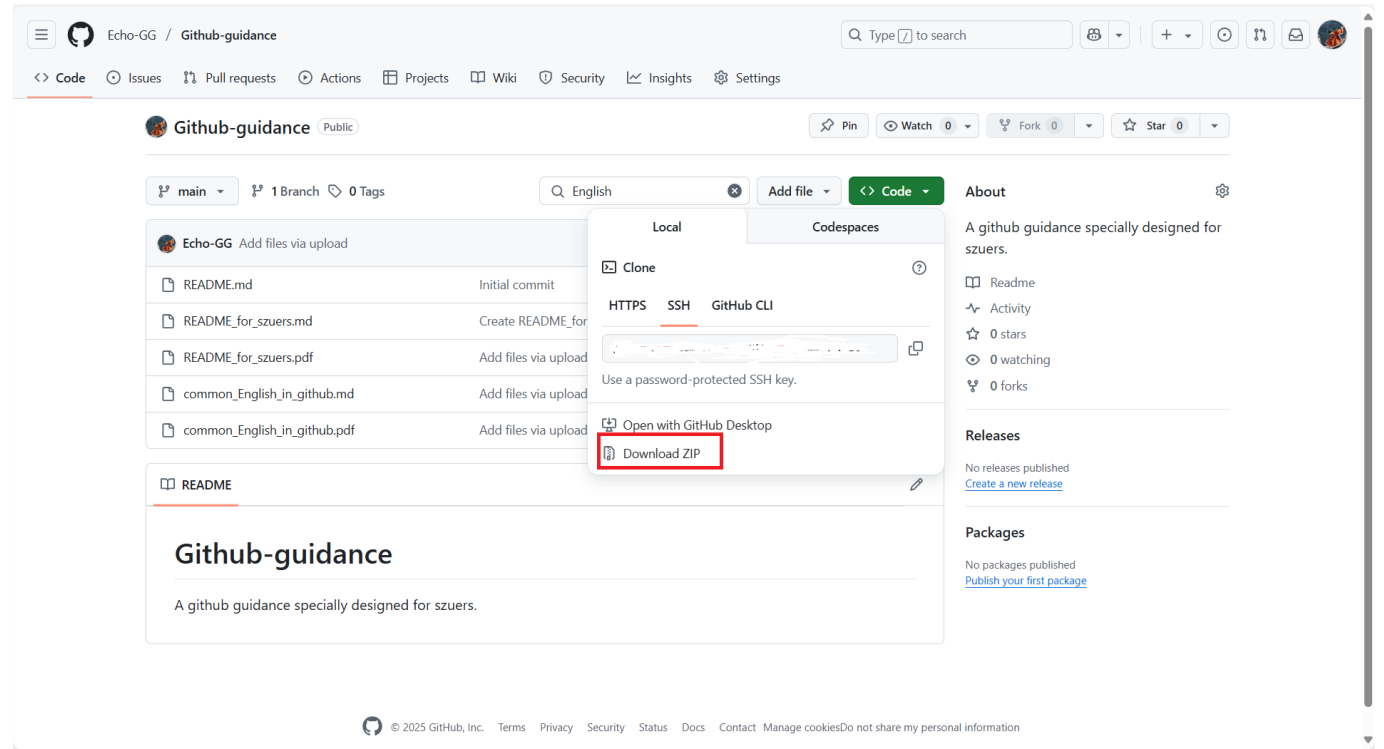
Step4 资源的搜索与获取



我们可以直接在搜索框中搜索该仓库里的内容(全站搜索就退出当前所在的仓库界面,直接使用Search功能即可),例如提示词English,找到对应文件后,可以点击右上角直接下载源文件(raw file).

这种方法适用于查找特定的单个文件并下载(源码)

另一种更常见的用法是,在搜索到开发者对应的仓库后,直接克隆或下载对应仓库文件的压缩包.(Github会自动请求仓库内的所有开放文件,并打包好成一个压缩包供其他查阅者直接下载使用)



Step5 仓库的维护

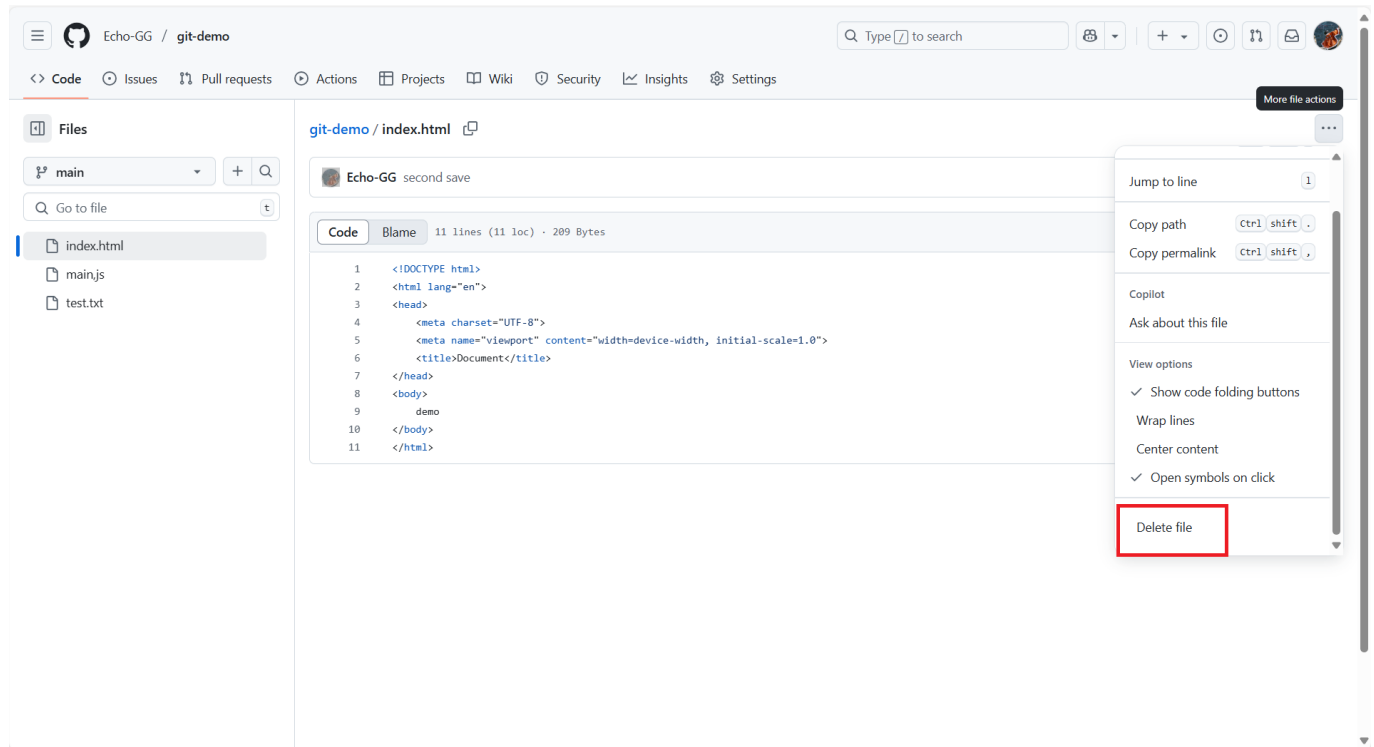
文件的添加

对于小范围的文件移动与添加,可以同上面提到的步骤一样,使用Github自带的 Add file 功能.

文件的删除

这是一个比较重要的部分,很多人会忽视这一部分.

对于已建立文件,我们如果需要删除他,最简单的一种方法就是单击进入待删除文件的界面,点击右上角...,鼠标向下滑动一下滚轮,有一个Delete file 的选项,直接单击delete即可,删除之后,网站上方会有一条蓝色的删除成功提示信息.



到这里 Github 的简单使用方法就介绍完毕了,和之前提到的一样,如果你只是希望将他作为一个远程的代码托管平台和代码查询平台来使用,以上的介绍是完全够用的。其他的进阶玩法,例如借助Github搭建属于自己的个人博客;将 Github 与自己的 vscode 远程连接,实现代码的实时保存和远程仓库同步;通过gitbash输入一系列指令来实现大规模数据移动或删除的操作(例如仓库clone)等,此处不作过多详细的介绍.感兴趣的同学可以自行查阅了解,在探索过程中遇到任何问题,也欢迎及时进行线上交流喵.

以下分享一份git的快速入门指南

<https://git-scm.com/book/en/v2>

这是官方网站给出的git上手指南,想深入理解git运作原理的同学可以自行学习,电子版pdf [progit.pdf](#)也会随本文档一并上传备用.

有关于git的新手入门视频,个人推荐:

<https://b23.tv/IIzxsRf>

虽然标题有些怪怪的,但是视频内容的确将git的入门门槛降到了最低(几乎skip掉了所有的gitbash上的命令行操作,主要依赖于vscode自带的插件功能)