一、Git安装

[Git (git-scm.com)](https://git-scm.com/)

安装都选择默认

二、使用git clone

1、在本地建立要存放项目的文件夹

2、右键打开Open Git Bash here

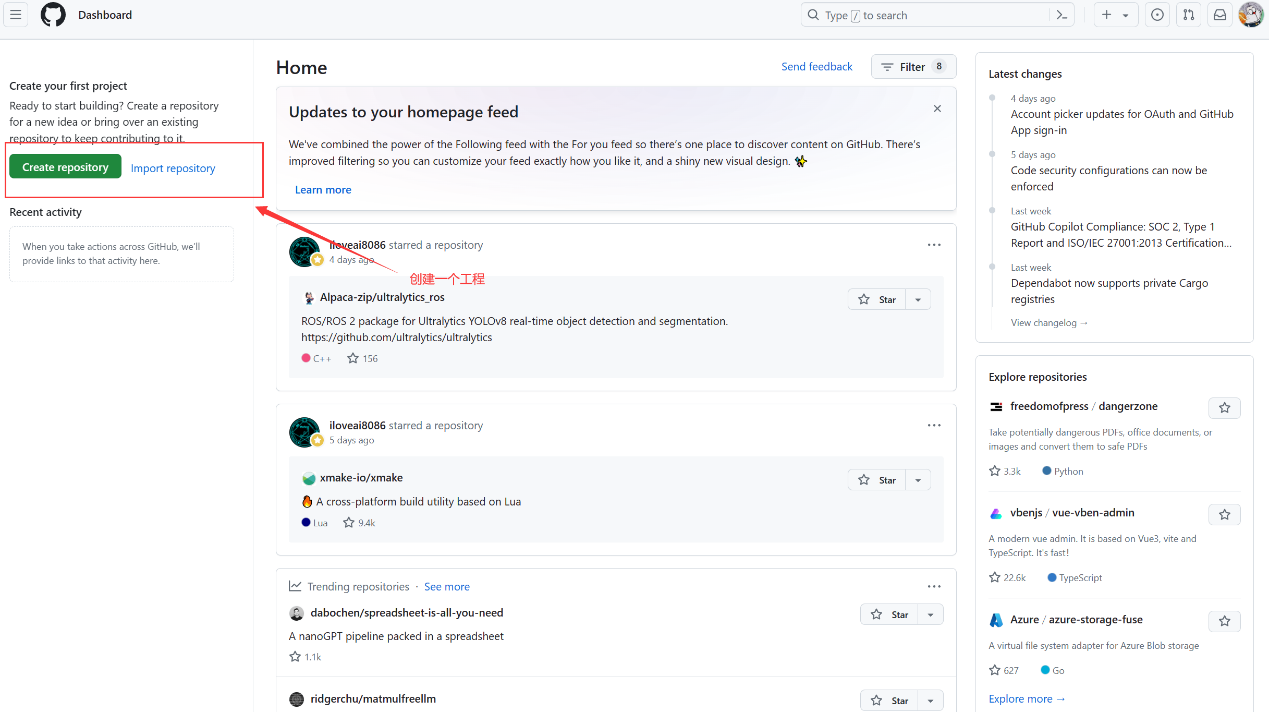
3、git clone （HTTPS）[<https://github.com/用户名/>项目名.git]

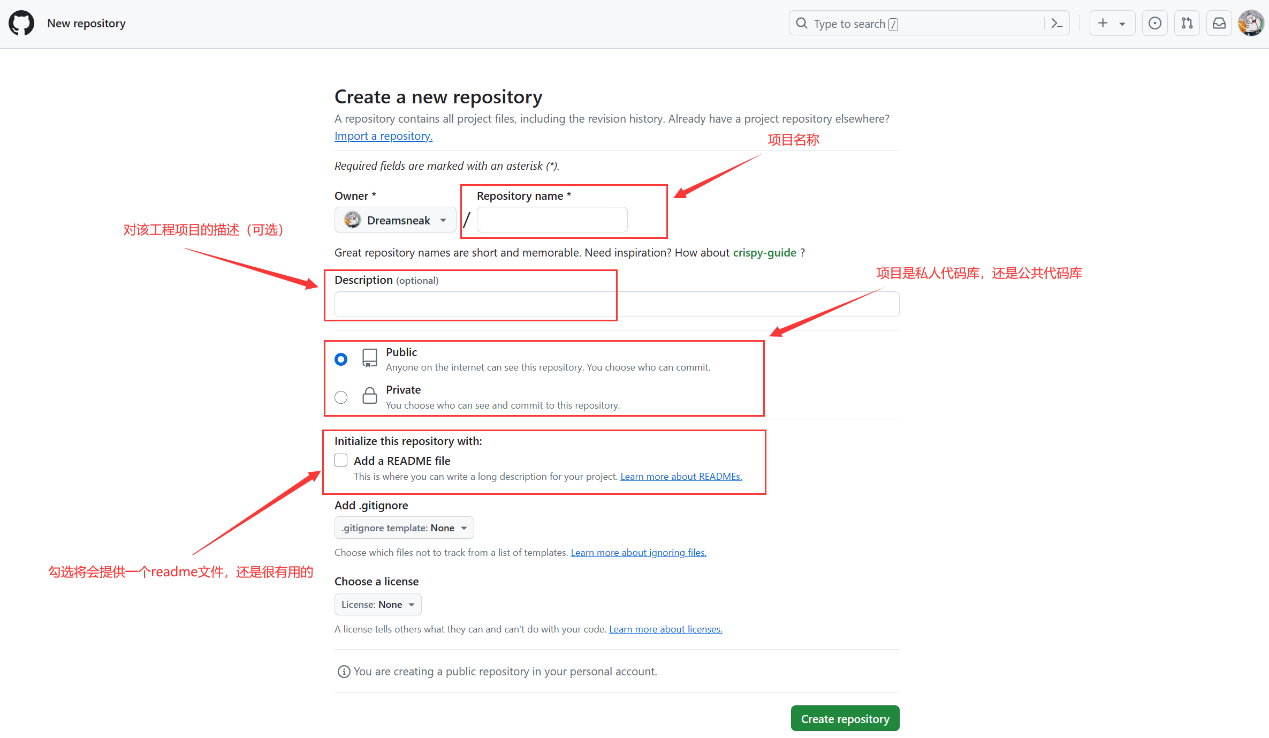
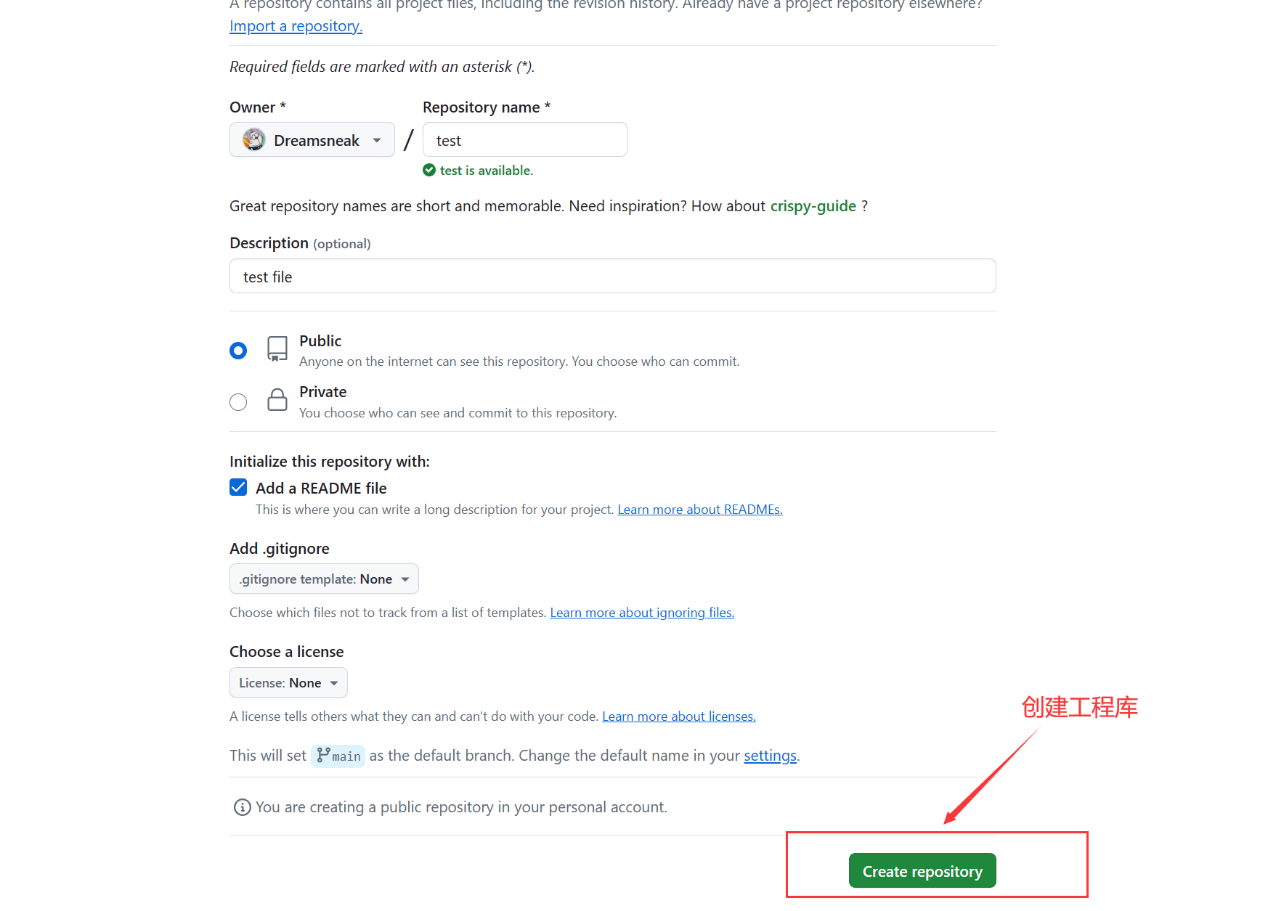
三、建立Github仓库

1、登入自己的 Github，点击创建 Create repository

2、对创建的项目工程内容进行选择性的填写

用户名[Owner]、项目名[Repository name]、勾选[Add a README file]



四、git配置[本地邮箱、用户名、密码]

1、查看当前配置

查看全部配置：git config -l

git config user.email

git config user.name

git config user.password

2、修改全局配置

git config --global user.name "xxx(新的用户名)"

git config --global user.email "123456@163.com(新的邮箱)"

git config --global user.password "123456(新的密码)"

3、修改局部配置

git config user.name "xxx(新的用户名)"

git config user.email "123456@163.com(新的邮箱)"

git config user.password "123456(新的密码)"

出现错误：fatal: not in a git directory

原因：没有进入git目录或者仓库，即文件中需要有’.git’文件

解决办法：git init

五、Git简单生成公钥和私钥

1、打开Git Bash

2、执行生成公钥和私钥的命令：ssh-keygen -t rsa -C “your\_email@example.com”

-t rsa：指定密钥类型为RSA

-C “your\_email@example.com”：添加注释（通常是您的电子邮件地址），这有助于识别密钥

执行上述命令后，系统会提示您输入文件名以保存密钥（默认情况下保存在~/.ssh/id\_rsa中）。

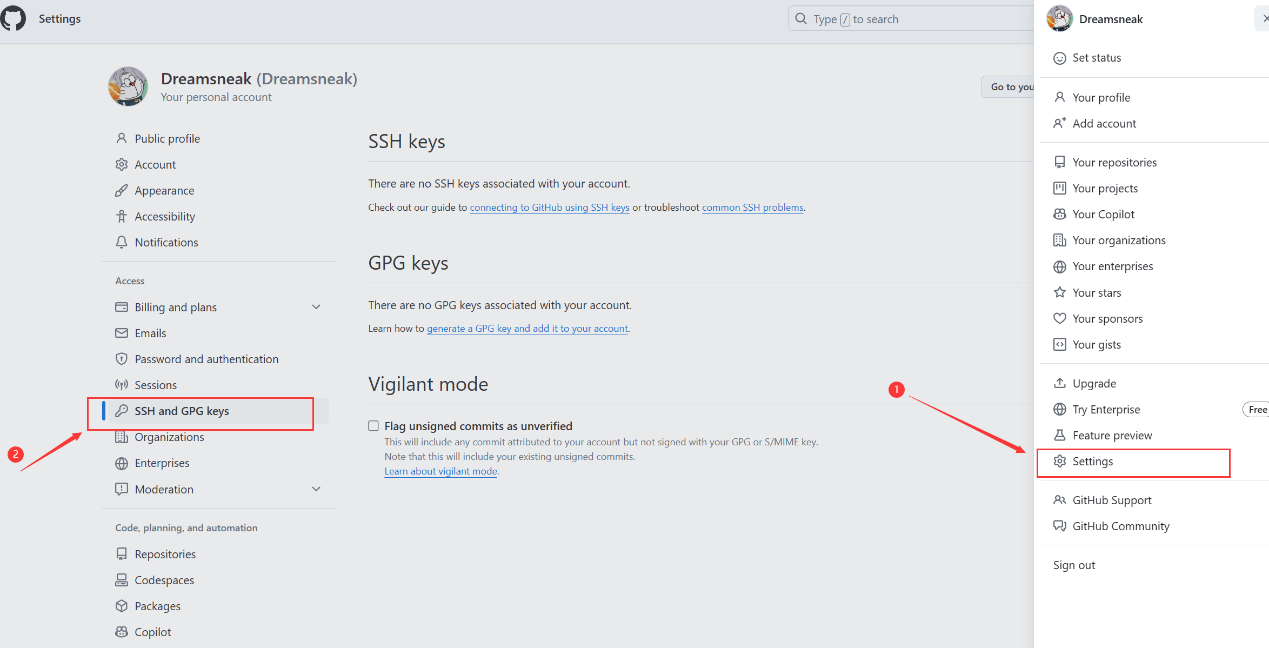
[以下需要按三个回车，后两个是为私钥设置密码，我设置了1]

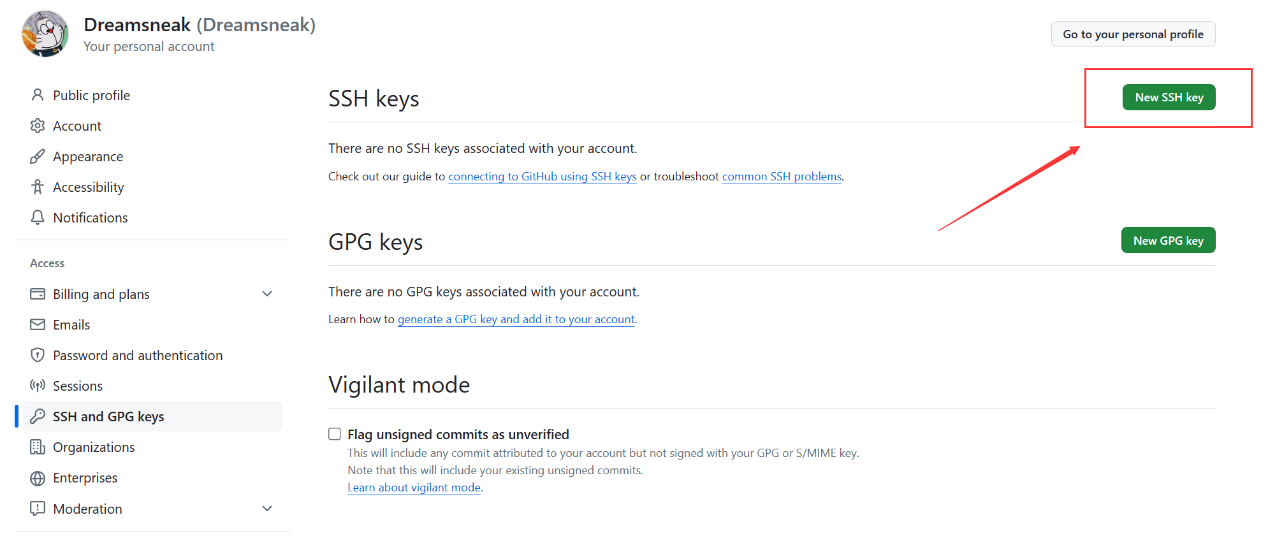
3、SSH在C:\Users\huawei\.ssh中，id\_rsa：私钥文件，id\_rsa.pub：公钥文件

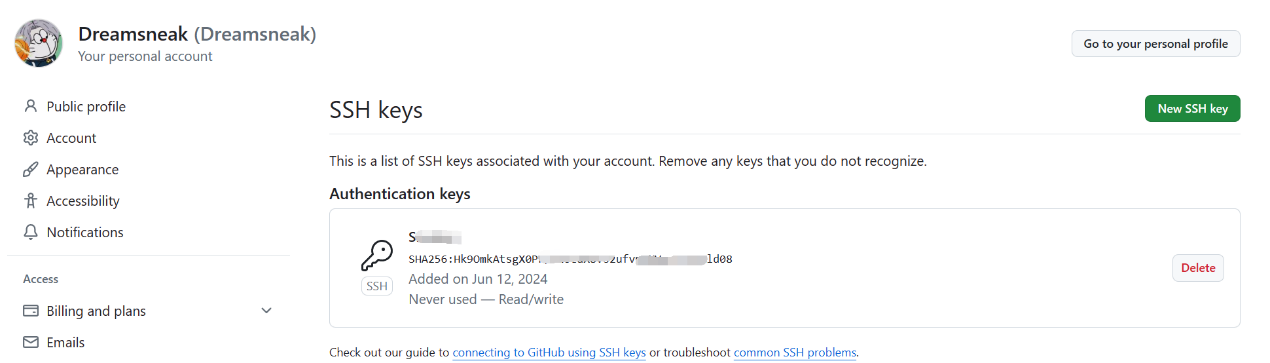
六、Github配置SSH

1、主页中选择Setting，点击SSH and GPG keys

2、提交公钥：找到.ssh文件[C:\Users\huawei\.ssh\id\_rsa.pub]，用文本编辑器打开，复制内容到Github的Key中，点击Add SSH Key







七、SSH连接

1、复制Repository的SSH链接

2、回到Git Bash页面，输入

git remote add origin [git@github.com:用户名/文件名PytorchLearning.git](mailto:git@github.com:用户名/文件名PytorchLearning.git)

3、查看连接的远程仓库链接：git remote -v

八、推送文件git push

1、选择自己的想要放置的位置，克隆仓库

git clone （HTTPS）[<https://github.com/用户名/>项目名.git]

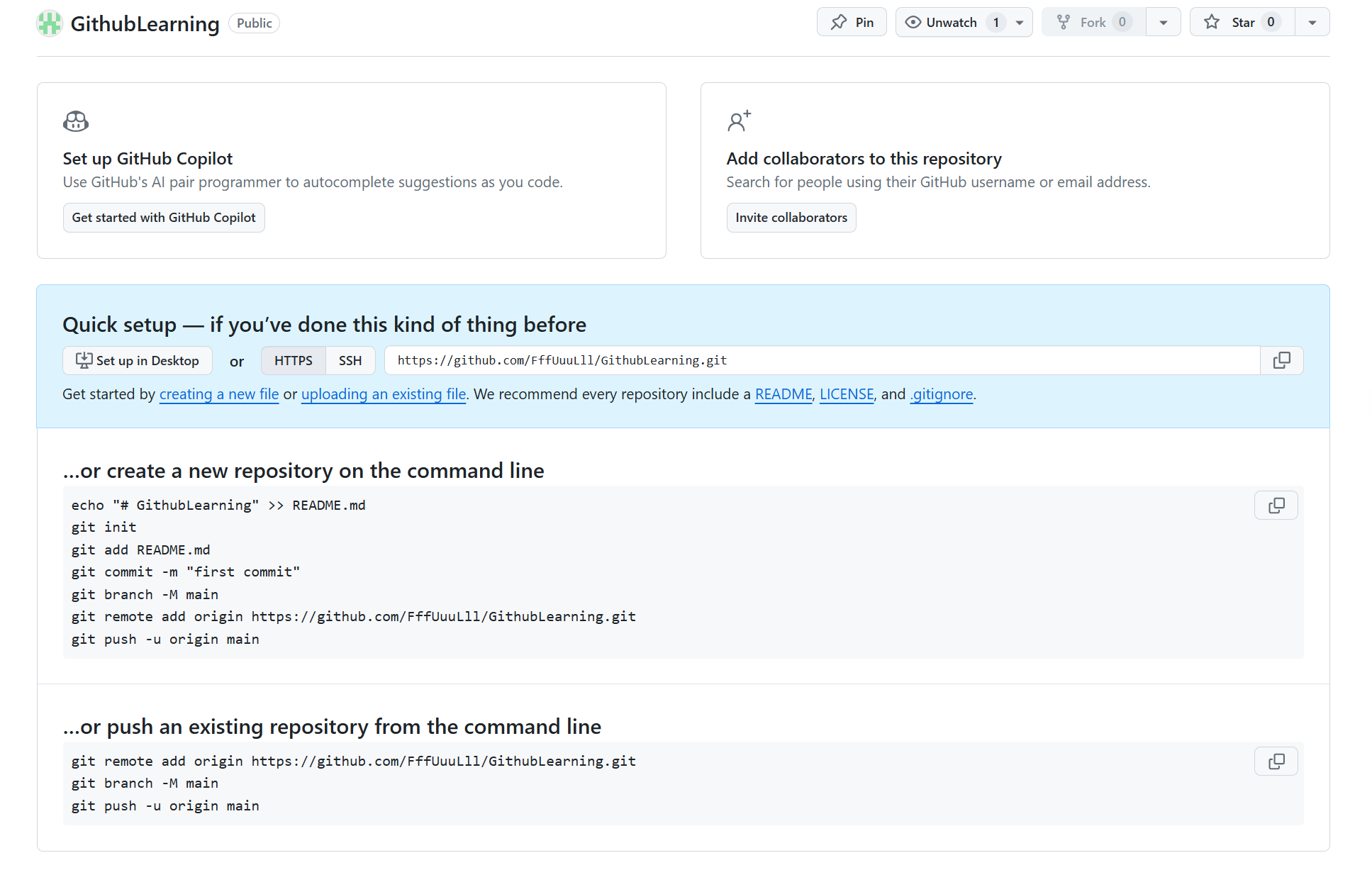
2、上传代码：git add 文件名[test.txt]

3、git commit -m “文件相关备注”，这是方便查找记忆而已

4、上传成功：git push origin main

九、Git命令汇总





九、拉取文件git pull

1、首先本地有一个文件（git clone）

2、A🡪B🡪C🡪D[主线开发]

E🡪F🡪G[副线开发]

A与E同时，E从A处git clone，等到了D和G一起处理

3、git pull的作用是从一个仓库或者本地的分支拉取并且整合代码

[git pull是拉取远程分支更新到本地仓库的操作。比如远程仓库里的文件内容有变化，需要把新内容下载下来的时候，就可以使用git pull命令。事实上，git pull是相当于从远程仓库获取最新版本，然后再与本地分支merge（合并）。]

4、

栗子1：

假设项目有以下分支，master、dev1、dev2、dev3、dev4，当前团队处于dev2分支，可能其他团队所属分支dev1或dev2修改了代码，并合并到master分支，当前团队需要从master分支获取最新的代码，合并到本地。

$ git pull origin master

执行此操作后，master分支的代码将会合并到本地当前分支，即dev2分支

栗子2

$ git pull origin master:dev4

执行此操作后，master分支的代码将会与指定的分支dev4合并

栗子3

$ git fetch origin master:dev4

$ git merge dev

先执行git fetch 再执行git merge。这种方式更安全也更符合实际要求，因为可以在merge前，我们可以查看更新情况，根据实际情况再决定是否合并