**Git---版本管理工具（VCS）**

1. 分布式版本控制；
2. 多个开发人员协调工作；
3. 有效监听谁做的修改；
4. 本地及远程操作。

**Git Basic Commands**

1. git init // 初始化本地git仓库（本地空的仓库）
2. git and <file> // 添加文件（形成文件队列）
3. git status // 查看状态（查看被添加、修改的文件）
4. git commit // 提交（提交刚刚add的文件队列）

// -m ‘memo’ -- 提交时一并填写备注信息

1. git push // 推送到仓库（把本地仓库的文件推送到远程仓库，用于分享、保存）
2. git pull // 从远程仓库拉取数据（从远程仓库中拉取文件）
3. git clone // 从远程仓库拷贝数据

**Git安装：**

Win：<https://git-scm.com/download/win>

Mac：<https://git-scm.com/download/mac>

**Git使用：**

安装后，目录下会有一个Git bash工具，此为Linux终端，使用Linux命令。

Git –version查看git版本确认是否安装成功

$ git --version

git version 2.31.1.windows.1

下面进入具体操作：

1. 创建本地仓库目录，位于E:\工作\GitNativeRepositories\MyFirstRepository

由于Linux下的文件系统不同于Windows，无法在命令行下切换盘符，所以在目录中右键，点击Git Bash Here，打开bash命令行



l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MyFirstRepository (master)

$

1. 当前目录下创建文件

touch index.html

touch app.js

1. 初始化git本地仓库

git init

1. 查看当前git状态

$ git status

On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

app.js

index.html

nothing added to commit but untracked files present (use "git add" to track)

即当前两个文件待加入队列

1. 添加文件

######

git add <file> // 添加单个文件

git add \* // 添加所有文件

git add . // 添加所有文件

git add \*.<type> // 添加某类文件，例如：git add \*.html

查看状态：

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: app.js

new file: index.html

1. 将文件从队列中移除

git rm --cached <file>

1. 将队列中的文件提交

git commit

之后会自动进入VIM，进行备注信息的编写。结束后按Esc键，:wq保存后退出。

Git commit -m ‘<memo>’

此命令不会进入VIM，会在提交的同时，使用’<memo>’进行备注。

1. 忽略相关文件

本地仓库中的txt文件一般不需要上传，所以在提交文件时需要忽略部分文件

例如：在本地仓库新建log.txt文件，查看git状态

Untracked files:

(use "git add <file>..." to include in what will be committed)

log.txt

在当前目录下创建文件.gitignore，将需要忽略的文件夹名、文件名写入

此时，再查看git状态，如下：

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

由此可见，log.txt文件即被文件队列忽略掉了，之后添加并提交.gitignore文件。

关于忽略文件夹，新建目录dir1、dir2，如果需要忽略dir1整个目录及其中的文件，则在.gitignore文件中添加“/dir1”，查看git状态，如下：

Untracked files:

(use "git add <file>..." to include in what will be committed)

dir2/

1. 创建分支branch

为了在工作过程中不影响主线文件，则需要使用分支进行更改，确认无误后，再合并到主线文件中。

git branch temp

git status，结果如下：

On branch master

可以看到，当前在默认的master主线下。之后切换分支

git checkout temp

切换前，提示符为：

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MyFirstRepository (master)

切换后，提示符为：

$ git checkout temp

Switched to branch 'temp'

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MyFirstRepository (temp)

在分支temp中创建文件login.html，并修改内容。之后修改原有的index.html内容并保存。查看git状态，如下：

$ git status

On branch temp

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

Untracked files:

(use "git add <file>..." to include in what will be committed)

login.html

可见，由于更改了login.html、index.html两个文件，所以当前需要更新的有两个文件。添加队列并提交

$ git add .

$ git commit -m 'branch temp'

此时切换回主线

$ git checkout master

Switched to branch 'master'

会发现本地仓库中没有了在temp分支下新建的login.html文件，且index.html的内容并没有出现在temp分支下的改动。

当再切换回分支temp后，本地仓库又出现了login.html文件，且index.html的内容发生了实时的变化。

所以，在分支中所做的修改，不会影响到主线文件的结构以及内容。

1. 将分支合并到主线

首先回到主线

git checkout master

git merge temp

$ git merge temp

Merge made by the 'recursive' strategy.

index.html | 2 +-

login.html | 12 ++++++++++++

2 files changed, 13 insertions(+), 1 deletion(-)

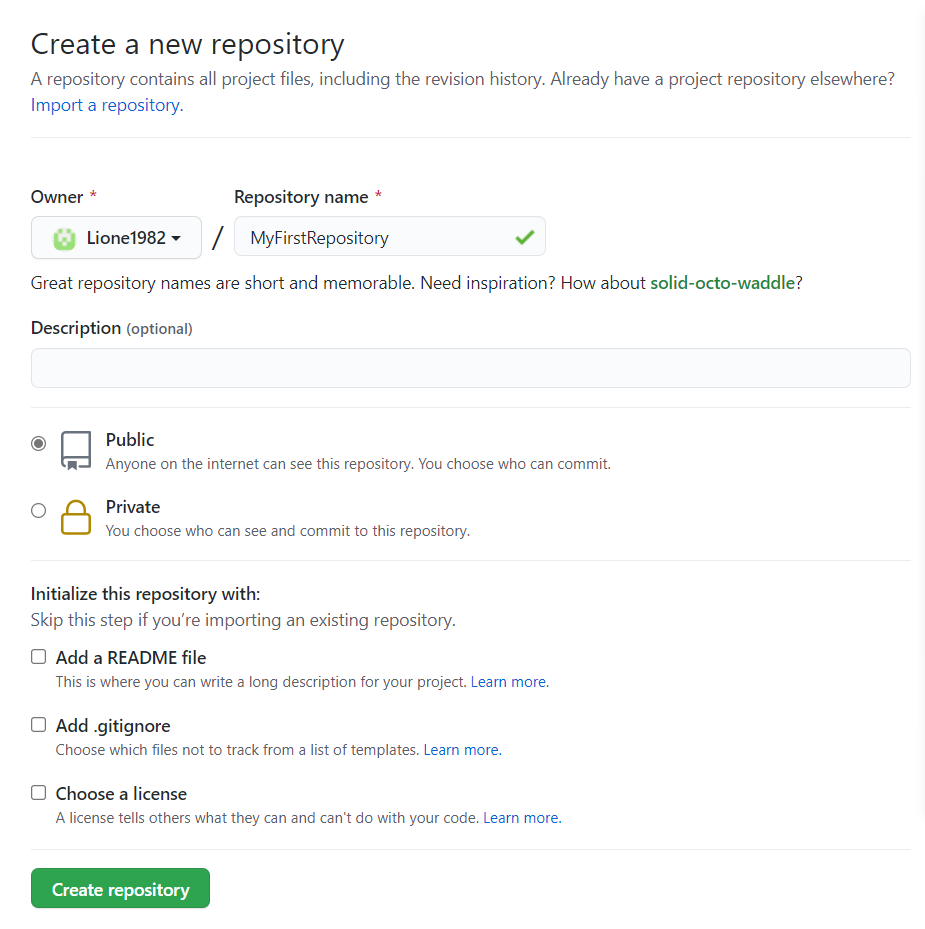
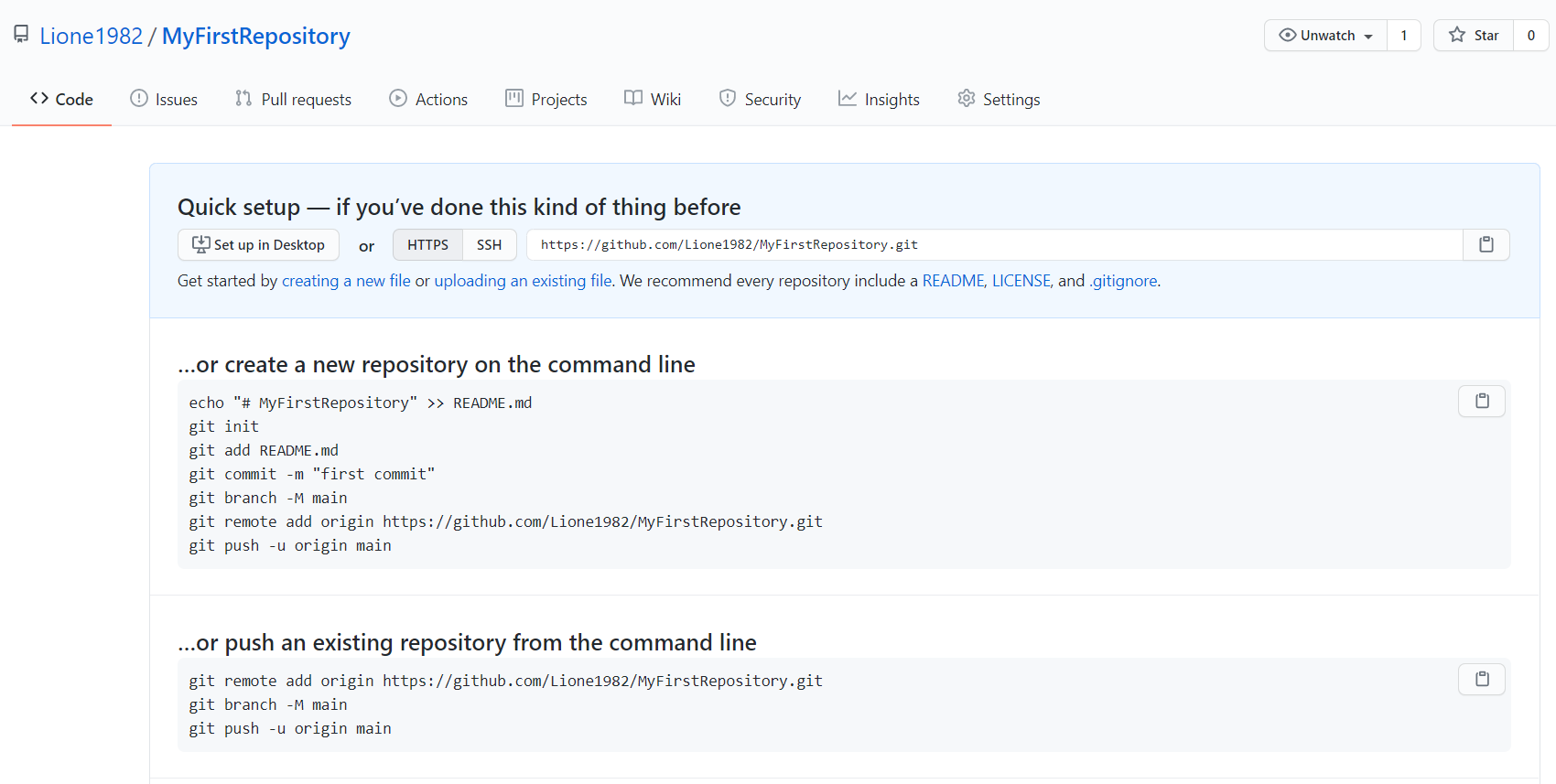
create mode 100644 login.html

此时，分支内容已经合并到主线master，查看本地仓库目录，login.html文件出现，且index.html的内容已经变化。

1. 将本地仓库同步到远程仓库GitHub

登录GitHub，<https://github.com>

新建仓库New Repository，Owner默认为当前用户名，设置仓库名Repository name，默认使用Public，Private是收费的，Create repository。

使用git remote命令对本地仓库与远程仓库进行对接

首先查看本地仓库是否已经存在与远程仓库的对接

git remote

为空表示尚无对接

使用如下命令：

git remote add origin <https://github.com/Lione1982/MyFirstRepository.git>

再次调用git remote查看远程对接

注意，已经创建了remote之后，在下次push之前不需要再次remote。

$ git remote

origin

之后将本地仓库的文件推送至远程仓库

git push -u origin master

push过一次之后，下次push不需要配置，直接git push即可。

git push时遇到用户验证的问题

查看所有git配置项

$ git config --list

配置用户名

$ git config --global user.name Lione1982

配置用户邮箱

$ git config --global user.email [mydream.1982@hotmail.com](mailto:mydream.1982@hotmail.com)

网络上的解决办法是使用如下命令，清除本地账户缓存，push时输入

$ git config --system --unset credential.helper

push成功，显式结果如下：

$ git push -u origin master

Enumerating objects: 25, done.

Counting objects: 100% (25/25), done.

Delta compression using up to 12 threads

Compressing objects: 100% (21/21), done.

Writing objects: 100% (25/25), 2.29 KiB | 782.00 KiB/s, done.

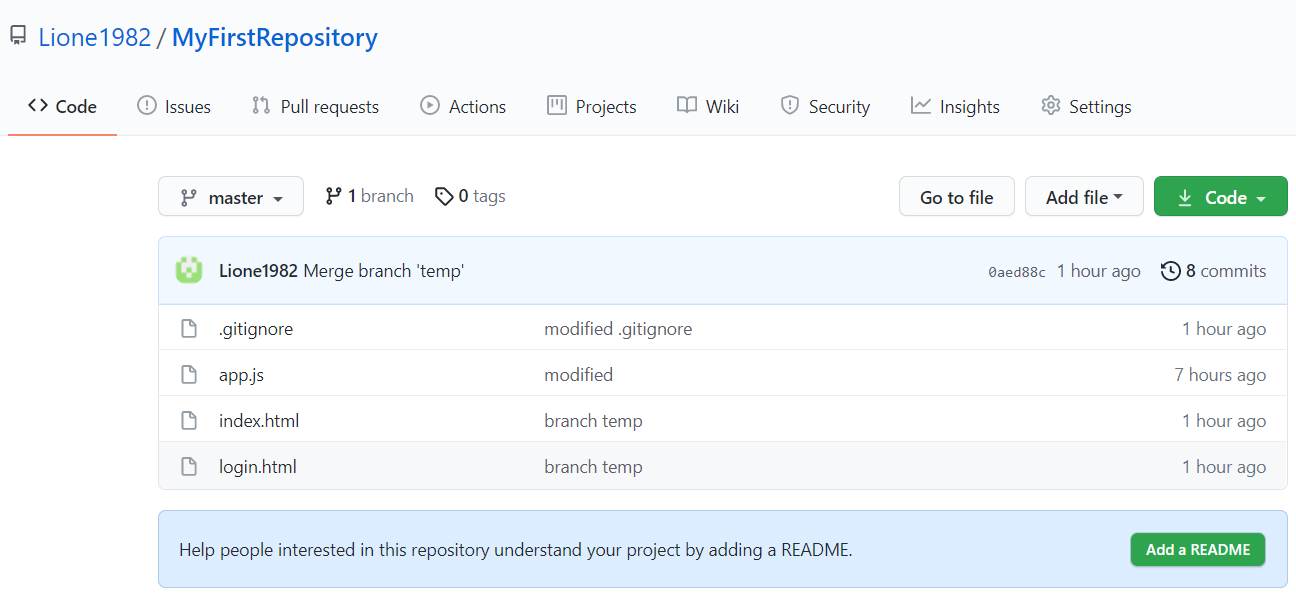
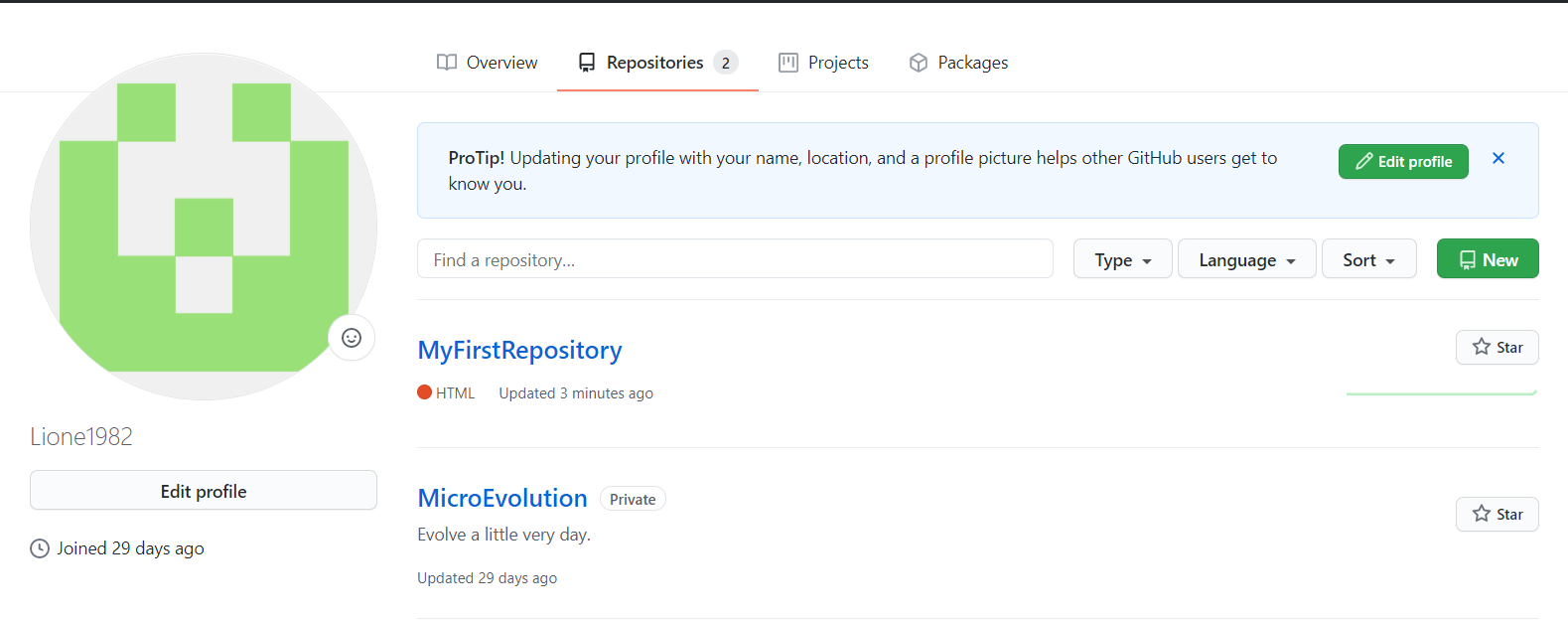
Total 25 (delta 7), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (7/7), done.

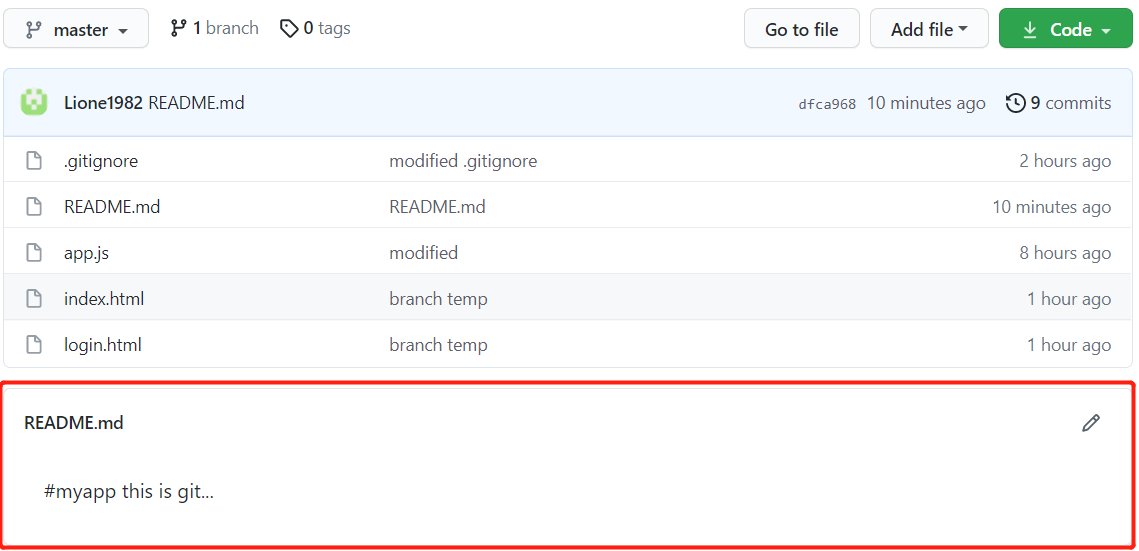
To https://github.com/Lione1982/MyFirstRepository.git

\* [new branch] master -> master

Branch 'master' set up to track remote branch 'master' from 'origin'.



添加README.md，用于项目描述。



md文件---markdown，轻量级标记语言，使用纯文本格式编写文档，之后转换成有效的XHTML（或者HTML）文档。

$ touch README.md

$ git add .

$ git commit -m 'README.md'

$ git push：

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 12 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 289 bytes | 289.00 KiB/s, done.

Total 3 (delta 1), reused 0 (delta 0), pack-reused 0

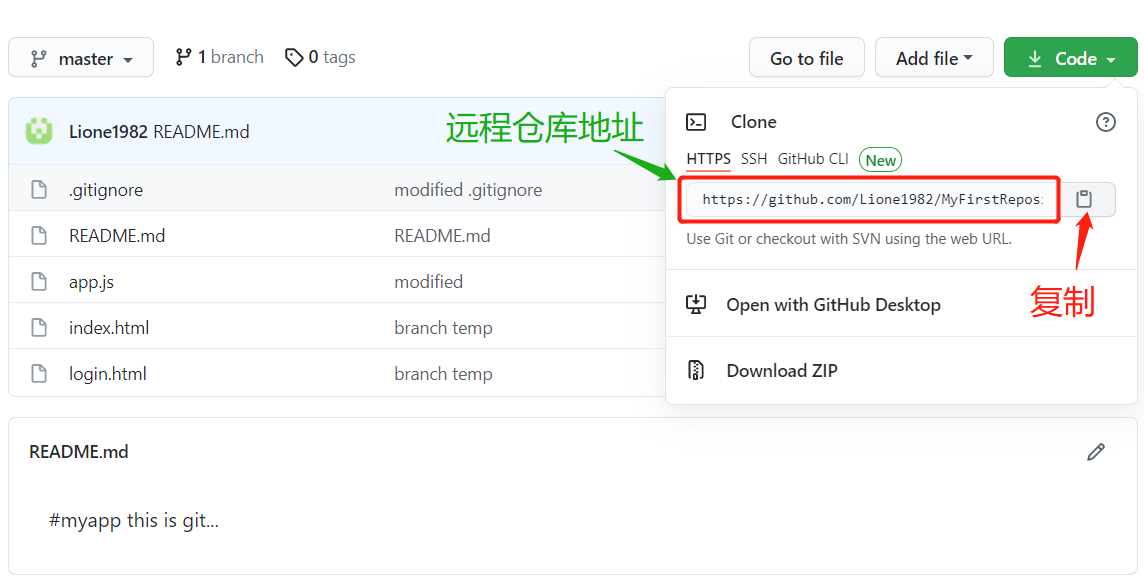
remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To https://github.com/Lione1982/MyFirstRepository.git

0aed88c..dfca968 master -> master

1. 项目克隆

用于将远程仓库中的文件克隆到本地仓库



在本地仓库同级目录新建文件夹CloneMyFirstRepository，右键Git Bash Here，打开命令行。

或者使用Windows PowerShell进入Git Bash，运行-powershell

PS C:\Users\l> cd E:\工作\GitNativeRepositories\CloneMyFirstRepository

PS E:\工作\GitNativeRepositories\CloneMyFirstRepository> git --version

git version 2.31.1.windows.1

PS E:\工作\GitNativeRepositories\CloneMyFirstRepository> git clone https://github.com/Lione1982/MyFirstRepository.git

Cloning into 'MyFirstRepository'...

remote: Enumerating objects: 28, done.

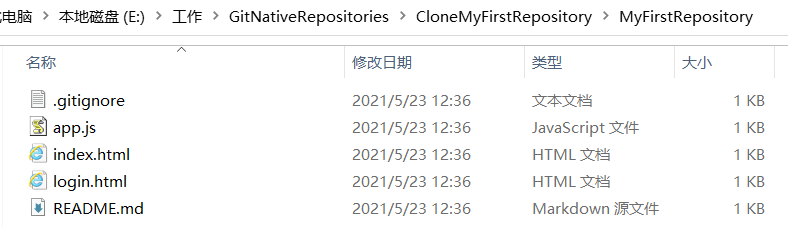
remote: Counting objects: 100% (28/28), done.

remote: Compressing objects: 100% (15/15), done.

remote: Total 28 (delta 8), reused 28 (delta 8), pack-reused 0

Receiving objects: 100% (28/28), done.

Resolving deltas: 100% (8/8), done.



至此，文件克隆成功。

***20210529，今天由GitHub上克隆项目时，使用git clone <url> cloning into ‘<dirname>’，遇到问题，提示【fatal: Too many arguments.】，于是删除cloning into，保留git clone <url> ‘<dirname>’后正常。***

**GitHub客户端**

Github是借助Git管理代码的平台

1. Git是代码管理工具
2. Github是基于Git实现的代码管理平台

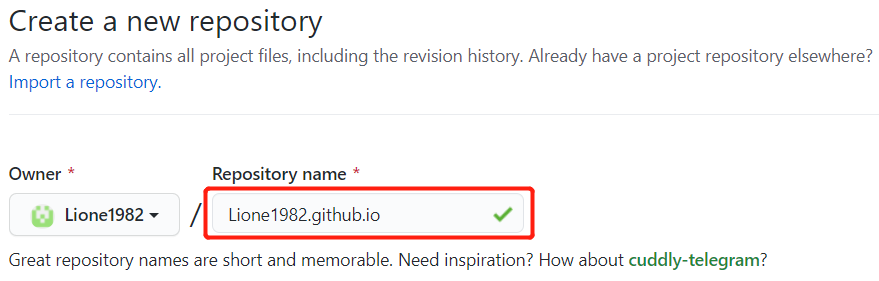
Github的操作流程：

1. 在Github官网创建账号
2. 创建远程仓库（存储代码）
3. 安装Github客户端
4. 登录客户端并克隆仓库
5. 在本地仓库中存储数据
6. 提交数据并备注信息
7. 同步本地数据到远程仓库

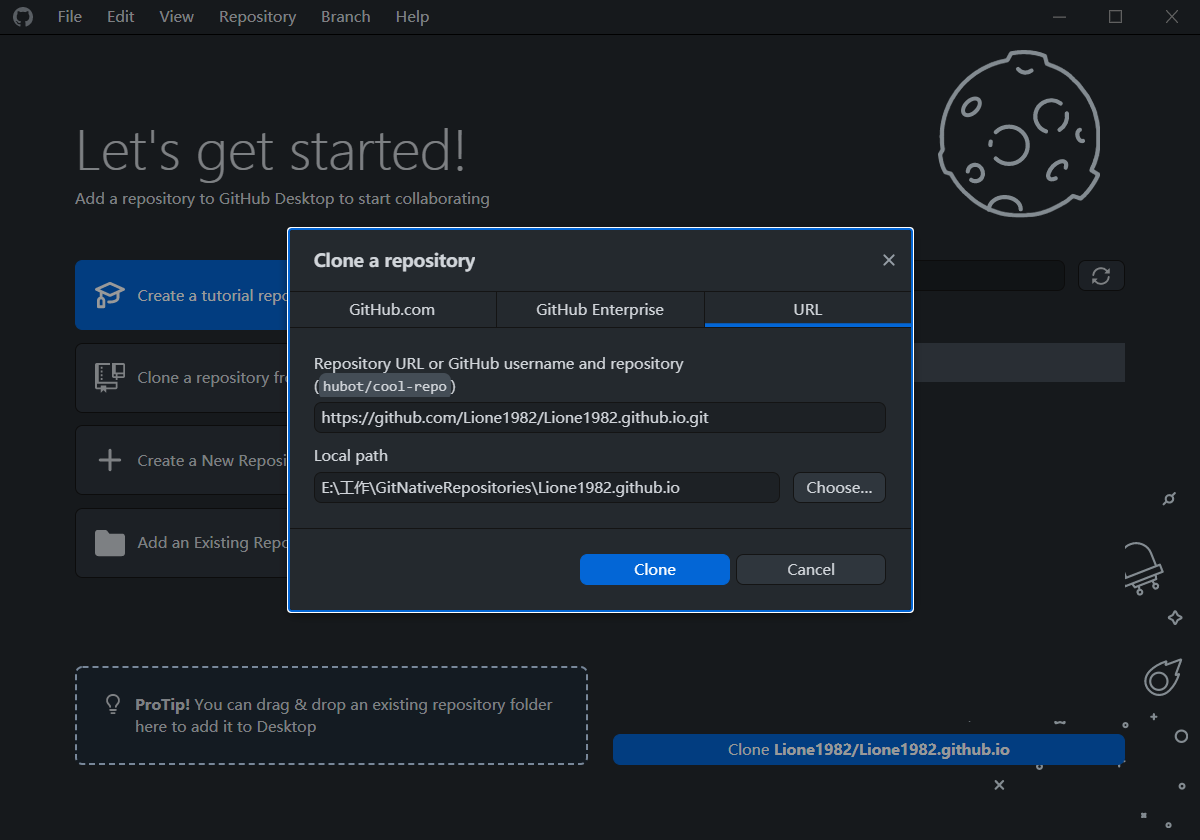
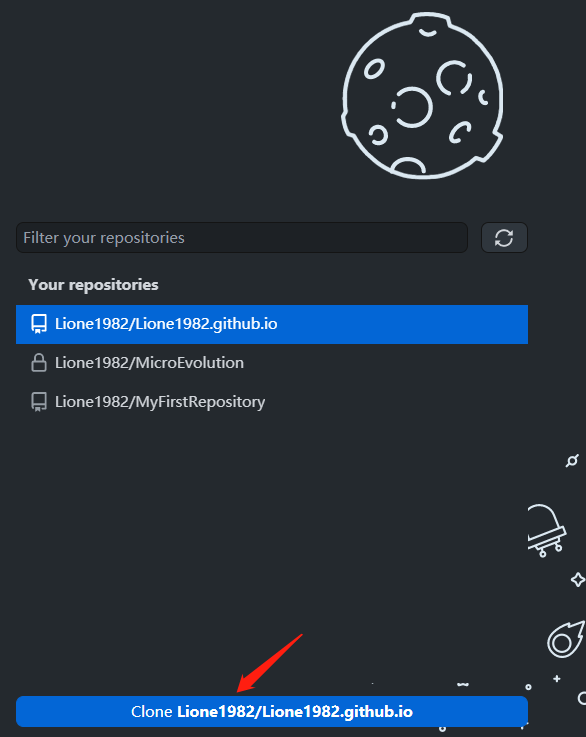
安装Github客户端

<https://desktop.github.com>

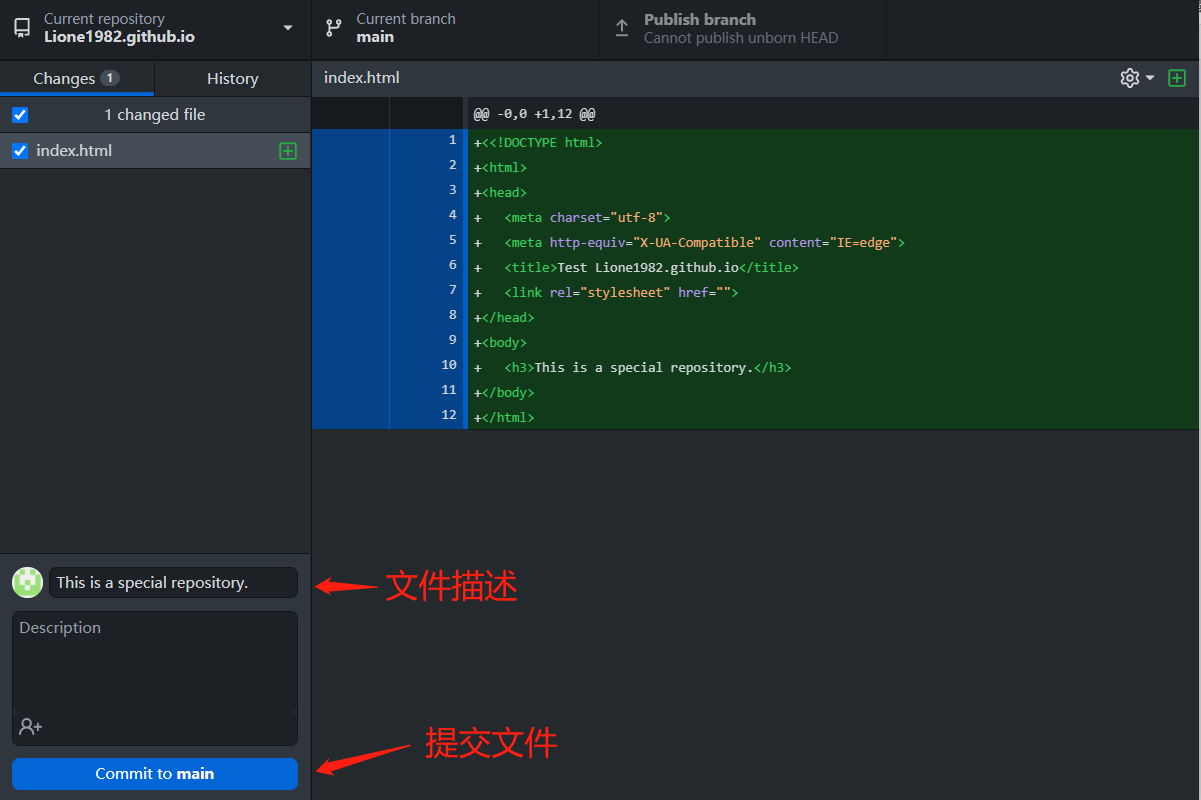
在Github官网创建一个特殊的仓库，Repository name使用<Owner>.github.io，优势是可以将此仓库作为一个服务器供他人访问。



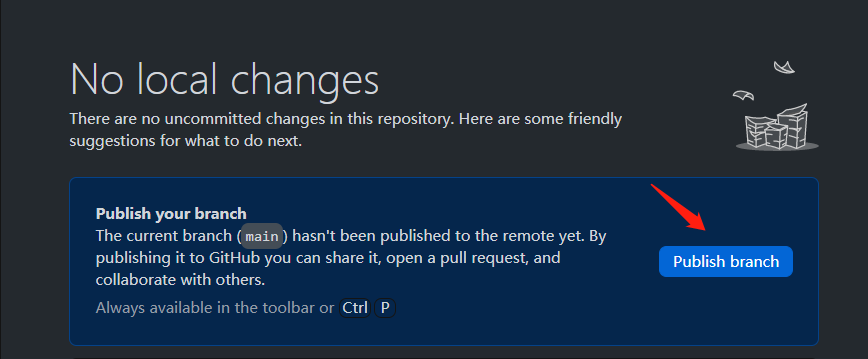
进入客户端，克隆此特殊仓库到本地仓库



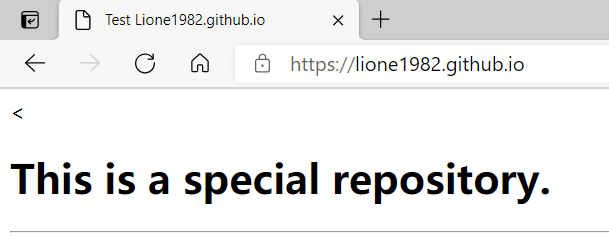
在本地仓库Lione1982.github.io中新建index.html文件，此时回到Github客户端，可以看到新建的文件及内容。



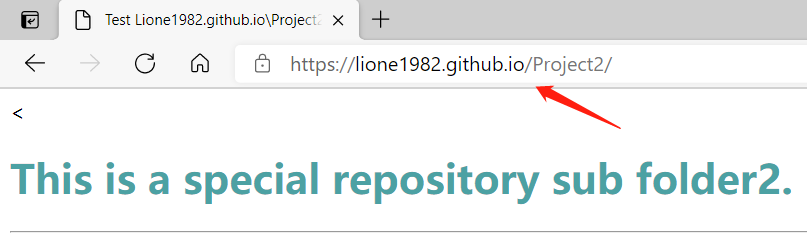
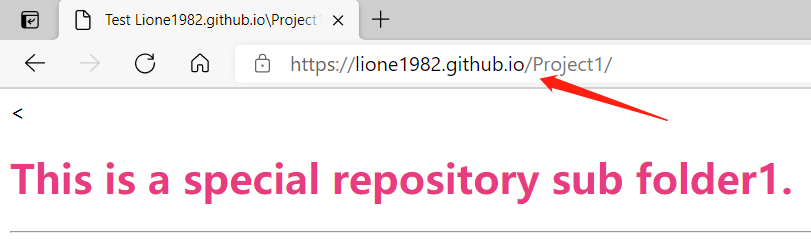
填写文件描述后提交，之后发布，开始网络传输



这时，回到Github官网，就可以看到Lione1982.github.io仓库下有了index.html，下面通过浏览器地址Lione1982.github.io就可以访问到index.html页面了。



可以添加子目录，其中新建html文件，则可以使用[https://Lione1982.github.io/<subpath](https://Lione1982.github.io/%3csubpath)>访问。【注意：子目录大小写敏感，主域名可大写或小写】



更换域名

由于上面的链接，使用了Github提供的服务器域名，为了安全起见，使用自己申请的域名。

使用阿里云注册域名【淘宝账号登录】哈哈宝贝很开心

<https://wanwang.aliyun.com/domain/tld?spm=5176.22941746.J_2447957890.13.5d90a5f6Cp3GaS#.com>

右上角控制台-域名-点击持有的域名右侧解析按钮-添加解析-手机验证码

记录类型：CNAME（Connect Name）

主机记录：

1. www的
2. @的（没有www的）

解析线路使用默认值

记录值：Lione1982.github.io

保存结束后等待生效。

回到本地仓库中index.html同级目录下创建文件，写入注册的域名【www.domainname.com】。保存的名字必须为CNAME。

2022-09-09 GitHub使用记录：

创建如下目录E:\工作\GitNativeRepositories\MySecondRepository，新建文件hit.txt

PowerShell下切换到如上目录

1. 将目录置为Git仓库---**git init**

PS E:\工作\GitNativeRepositories\MySecondRepository> git init

Initialized empty Git repository in E:/工作/GitNativeRepositories/MySecondRepository/.git/

Warning: Your console font probably doesn't support Unicode. If you experience strange characters in the output, consider switching to a TrueType font such as Consolas!

1. 查看git状态---**git status**

PS E:\工作\GitNativeRepositories\MySecondRepository> git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

hit.txt

nothing added to commit but untracked files present (use "git add" to track)

1. 添加文件追踪---**git add** <filename>

PS E:\工作\GitNativeRepositories\MySecondRepository> git add hit.txt

PS E:\工作\GitNativeRepositories\MySecondRepository> git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: hit.txt

1. 提交更改---**git commit -m** ‘<memo>’

PS E:\工作\GitNativeRepositories\MySecondRepository> git commit -m 'first commit'

[master (root-commit) 0c40fc4] first commit

1 file changed, 1 insertion(+)

create mode 100644 hit.txt

1. 查看git状态---**git status**

PS E:\工作\GitNativeRepositories\MySecondRepository> git status

On branch master

nothing to commit, working tree clean

1. 查看git日志---**git log**

PS E:\工作\GitNativeRepositories\MySecondRepository> git log

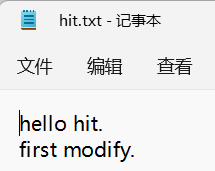
commit 0c40fc41b847df2011f6a7b5cc683f6ecee46f9c (HEAD -> master)

Author: Lione1982 <mydream.1982@hotmail.com>

Date: Fri Sep 9 10:07:47 2022 +0800

first commit

1. 手动更新hit.txt文件内容后查看git状态，如下：



PS E:\工作\GitNativeRepositories\MySecondRepository> git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: hit.txt

no changes added to commit (use "git add" and/or "git commit -a")

1. 添加追踪到缓冲区并提交

PS E:\工作\GitNativeRepositories\MySecondRepository> git add hit.txt

PS E:\工作\GitNativeRepositories\MySecondRepository> git commit -m 'first modify'

[master aef9aad] first modify

1 file changed, 2 insertions(+), 1 deletion(-)

1. 查看git日志---**git log**

PS E:\工作\GitNativeRepositories\MySecondRepository> git log

commit aef9aad885883407b346a25c306a6d4aa9a953bf (HEAD -> master)

Author: Lione1982 <mydream.1982@hotmail.com>

Date: Fri Sep 9 10:10:32 2022 +0800

first modify

commit 0c40fc41b847df2011f6a7b5cc683f6ecee46f9c

Author: Lione1982 <mydream.1982@hotmail.com>

Date: Fri Sep 9 10:07:47 2022 +0800

first commit

1. 分支-新建---**git branch** <branchname>

PS E:\工作\GitNativeRepositories\MySecondRepository> git branch b1

1. 分支-查看状态---**git branch**

PS E:\工作\GitNativeRepositories\MySecondRepository> git branch

b1

\* master

1. 分支-切换---**git checkout** <branchname>

PS E:\工作\GitNativeRepositories\MySecondRepository> git branch b1

PS E:\工作\GitNativeRepositories\MySecondRepository> git branch

b1

\* master

1. 分支-合并---**git merge** <branchname>

将其他分支合并到master分支，合并前需切换回master

PS E:\工作\GitNativeRepositories\MySecondRepository> git merge b1

Already up to date.

1. 分支-删除---**git branch -d** <branchname>

PS E:\工作\GitNativeRepositories\MySecondRepository> git branch -d b1

Deleted branch b1 (was aef9aad).

PS E:\工作\GitNativeRepositories\MySecondRepository> git branch

b2

\* master

1. 分支-一次性新建并切换---**git checkout -b** <branchname>

PS E:\工作\GitNativeRepositories\MySecondRepository> git checkout -b b2

Switched to a new branch 'b2'

PS E:\工作\GitNativeRepositories\MySecondRepository> git branch

b1

\* b2

Master

1. 标签-添加并查看状态---**git tag** <tagname>

PS E:\工作\GitNativeRepositories\MySecondRepository> git tag v1.0

PS E:\工作\GitNativeRepositories\MySecondRepository> git tag

v1.0

1. 标签-删除并查看状态---**git tag -d** <tagname>

PS E:\工作\GitNativeRepositories\MySecondRepository> git tag -d v1.0

Deleted tag 'v1.0' (was aef9aad)

1. 删除仓库文件后恢复

PS E:\工作\GitNativeRepositories\MySecondRepository> git status

On branch master

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

deleted: hit.txt

no changes added to commit (use "git add" and/or "git commit -a")

根据提示，使用git restore放弃修改

PS E:\工作\GitNativeRepositories\MySecondRepository> git restore \*

PS E:\工作\GitNativeRepositories\MySecondRepository> git status

On branch master

nothing to commit, working tree clean

之后，目录中的文件hit.txt恢复

1. 修改仓库文件后恢复

PS E:\工作\GitNativeRepositories\MySecondRepository> git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: hit.txt

no changes added to commit (use "git add" and/or "git commit -a")

PS E:\工作\GitNativeRepositories\MySecondRepository> git restore hit.txt

之后，文件hit.txt内容恢复

1. 删除当前远程链接**git remote rm(remove) origin**

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MySecondRepository (master)

$ git remote

origin

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MySecondRepository (master)

$ git remote rm origin

或者使用【git remote remove origin】

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MySecondRepository (master)

$ git remote

1. 一次完整的更新后提交流程
2. 首先删除本地仓库，进入本地仓库目录E:\工作\GitNativeRepositories，右键git bash进入命令行
3. git remote remove origin
4. git remote add origin https://github.com/Lione1982/MySecondRepository.git
5. cd MySecondRepository/
6. 【修改E:\工作\GitNativeRepositories\MySecondRepository\hit.txt】
7. git status 【modified: hit.txt】
8. git add hit.txt
9. git status 【modified: hit.txt】
10. git commit -m 'fourth modify.'
11. git push origin master

使用logout关闭命令行，修改hit.txt后重新进入命令行，add-commit，直接push报异常，使用命令git config --system --unset credential.helper后再次push正常

异常情况处理：

1. 由于git bash菜单异常，重新安装git，之后原先git仓库的拥有者发生变化。异常信息如下：

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MySecondRepository

$ git status

fatal: detected dubious ownership in repository at 'E:/工作/GitNativeRepositories/MySecondRepository'

'E:/工作/GitNativeRepositories/MySecondRepository/.git' is owned by:

'S-1-5-32-544'

but the current user is:

'S-1-5-21-2719249276-1431798618-2504204580-1001'

To add an exception for this directory, call:

git config --global --add safe.directory

'E:/工作/GitNativeRepositories/MySecondRepository'

解决办法：根据异常提示，使用如下命令后解决

git config --global --add safe.directory

'E:/工作/GitNativeRepositories/MySecondRepository'

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MySecondRepository (master)

$ git status

On branch master

nothing to commit, working tree clean

如上，命令提示符结尾处多了（master）

1. 链接远程仓库时，用户身份校验问题，如下：

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MySecondRepository (master)

$ git push origin master

fatal: unable to access 'https://github.com/Lione1982/MySecondRepository.git/': schannel: failed to receive handshake, SSL/TLS connection failed

百度后，解决办法为：

$ git config --global http.sslBackend "openssl"

之后再次链接远程仓库

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MySecondRepository (master)

$ git push origin master

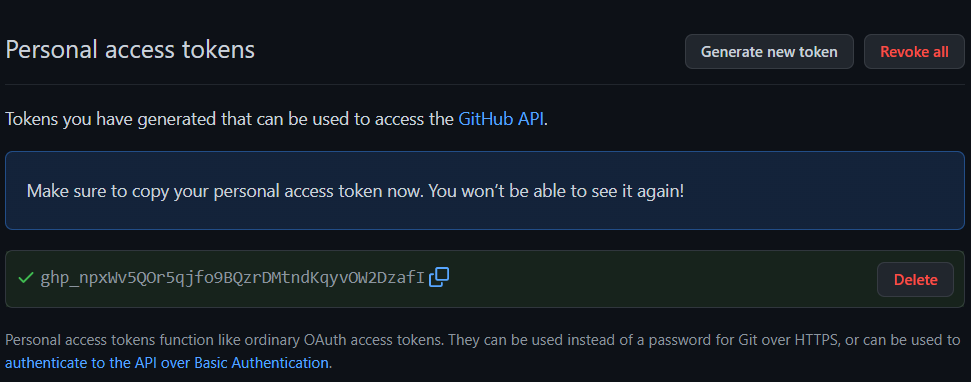
remote: Support for password authentication was removed on August 13, 2021.

remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.

fatal: Authentication failed for 'https://github.com/Lione1982/MySecondRepository.git/'

据查，此异常信息意思是自从 2021 年 8 月 13 后不再支持用户名密码的方式验证了，需要创建个人访问令牌(personal access token)。

个人令牌创建方式：Github-Settings-Developer settings-Personal access tokens-Generate new token-（confirm pwd）-set【New personal access token】-（Generate token）



Token：ghp\_npxWv5QOr5qjfo9BQzrDMtndKqyvOW2DzafI

修改现有的url，格式为：https://<token>@github.com/<USERNAME>/<REPO>.git/

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories/MySecondRepository (master)

git remote set-url origin

https://ghp\_npxWv5QOr5qjfo9BQzrDMtndKqyvOW2DzafI@github.com/Lione1982/MySecondRepository.git/

push时弹出用户名、密码，分别输入Lione1982、<token>后提交成功

1. 关于github.com的域名解析问题

使用140.82.114.3替换域名，如下：

git remote add origin

https://ghp\_npxWv5QOr5qjfo9BQzrDMtndKqyvOW2DzafI@140.82.114.3/Lione1982/MySecondRepository.git

1. 异常描述：Failed to connect to github.com port 443

问题解决：据查是由于代理问题，使用如下命令重设代理：

git config --global --unset http.proxy

git config --global --unset https.proxy

git config --global http.proxy

git config --global https.proxy

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories (master)

$ git clone https://140.82.114.3/Lione1982/MySecondRepository.git

Cloning into 'MySecondRepository'...

fatal: unable to access 'https://140.82.114.3/Lione1982/MySecondRepository.git/': Failed to connect to github.com port 443 after 21080 ms: Timed out

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories (master)

$ git config --global --unset http.proxy

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories (master)

$ git config --global --unset https.proxy

l@LAPTOP-CG1Q6IUA MINGW64 /e/工作/GitNativeRepositories (master)

$ git clone https://140.82.114.3/Lione1982/MySecondRepository.git

Cloning into 'MySecondRepository'...

warning: redirecting to https://github.com/Lione1982/MySecondRepository.git/

remote: Enumerating objects: 6, done.

remote: Counting objects: 100% (6/6), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0

Receiving objects: 100% (6/6), done.

之后，push时会弹窗要求输入用户名、<token>