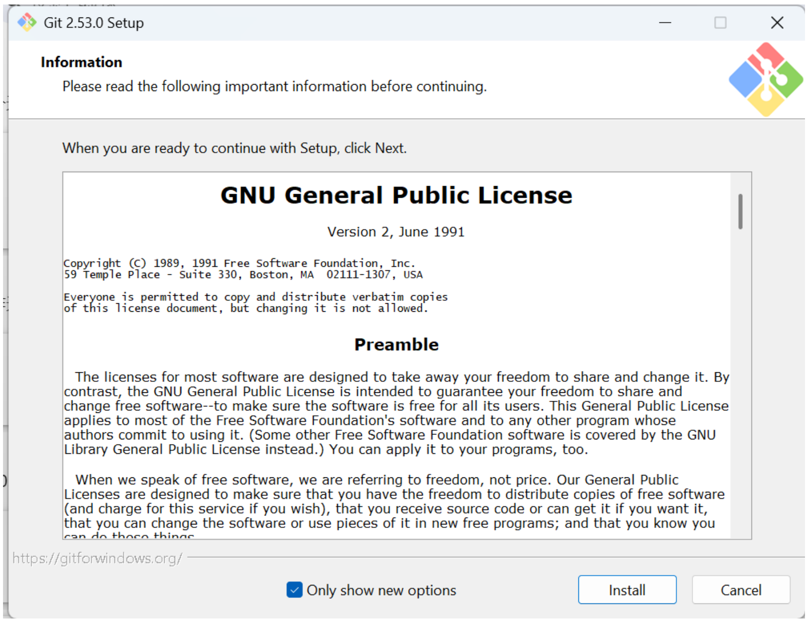
Git使用

- 官网下载
  - https://git-scm.com/install/windows
- 安装，定义好后，一直next



Git 2.53.0 Setup

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

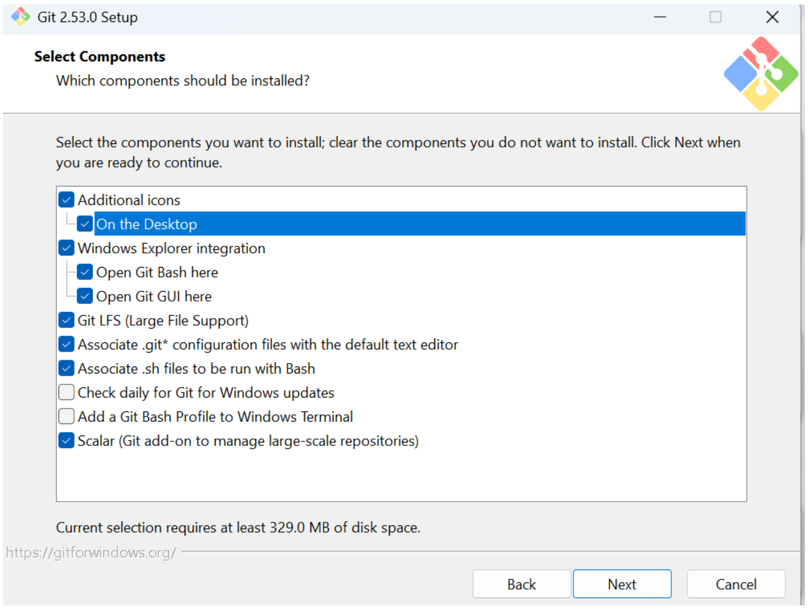
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

https://gitforwindows.org/

☑ Only show new options



Git 2.53.0 Setup

Select Components

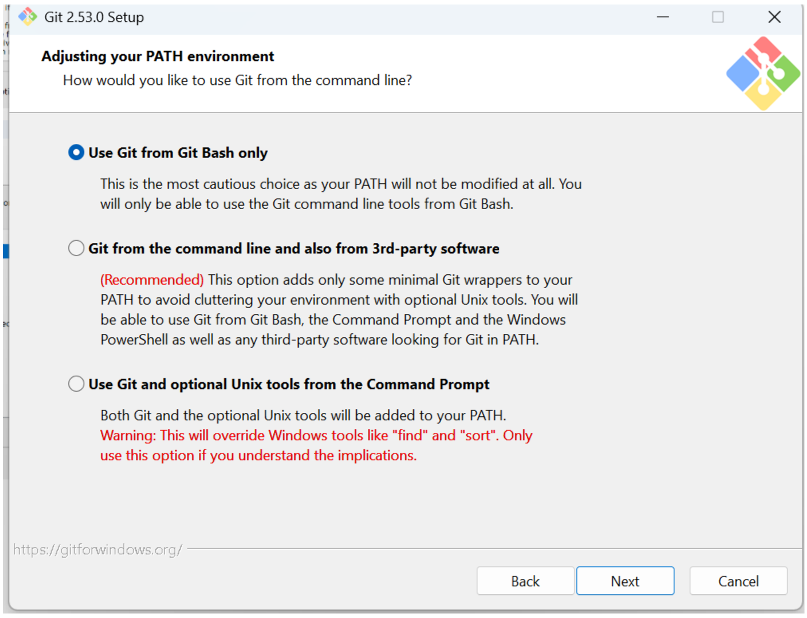
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Additional icons
  - ☑ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Open Git Bash here
  - ☑ Open Git GUI here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- □ Check daily for Git for Windows updates
- □ Add a Git Bash Profile to Windows Terminal
- ☑ Scalar (Git add-on to manage large-scale repositories)

Current selection requires at least 329.0 MB of disk space.

https://gitforwindows.org/



Git 2.53.0 Setup

Adjusting your PATH environment

How would you like to use Git from the command line?

● **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

○ **Git from the command line and also from 3rd-party software**

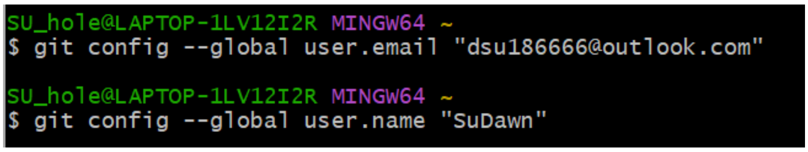
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

○ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH. Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

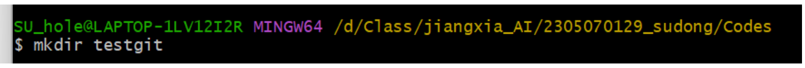
https://gitforwindows.org/

- 安装完成后，打开git，进行最后一步设置



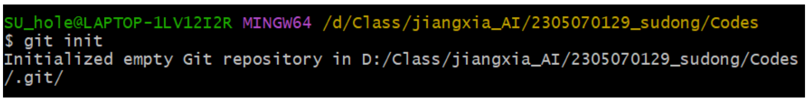
```
SU_hole@LAPTOP-1LV12I2R MINGW64 ~
$ git config --global user.email "dsu186666@outlook.com"

SU_hole@LAPTOP-1LV12I2R MINGW64 ~
$ git config --global user.name "SuDawn"
```

  - Git是分布式版本控制系统，所以需要填写用户名和邮箱作为一个标识
    - `git config --global user.email "dsu186666@outlook.com"`
    - `git config --global user.name "SuDawn"`
- 创建版本库
  - 版本库又名仓库，英文名repository,你可以简单的理解一个目录，这个目录里面的所有文件都可以被Git管理起来，每个文件的修改，删除，Git都能跟踪，以便任何时刻都可以追踪历史，或者在将来某个时刻还可以将文件”还原”。
  - -

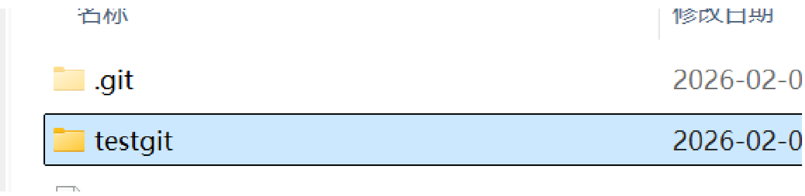


```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes
$ mkdir testgit
```

  - 通过命令 git init 把这个目录变成git可以管理的仓库

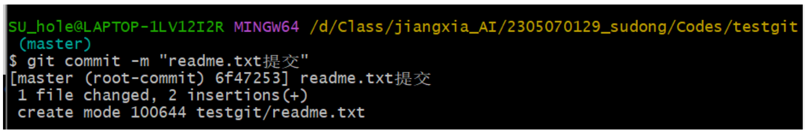


```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes
$ git init
Initialized empty Git repository in D:/Class/jiangxia_AI/2305070129_sudong/Codes
/.git/
```



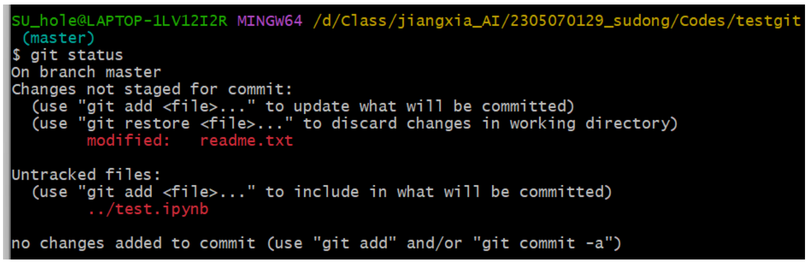
名称	修改日期
.git	2026-02-0
testgit	2026-02-0

- 把文件添加到版本库中
  - 在testgit下创建read.txt
  - `git add readme.txt`
  - `git commit` 告诉Git，把文件提交到仓库。



```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git commit -m "readme.txt提交"
[master (root-commit) 6f47253] readme.txt提交
 1 file changed, 2 insertions(+)
 create mode 100644 testgit/readme.txt
```

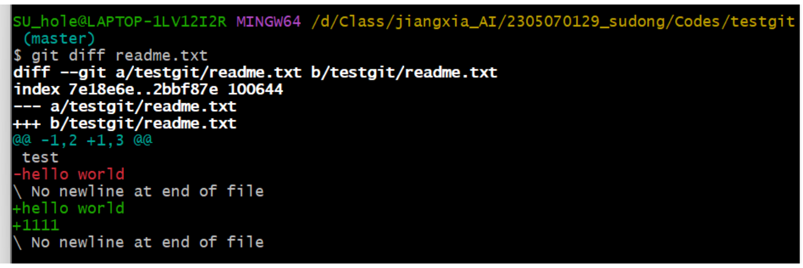
- `git status`



```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ 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:   readme.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ../test.ipynb

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

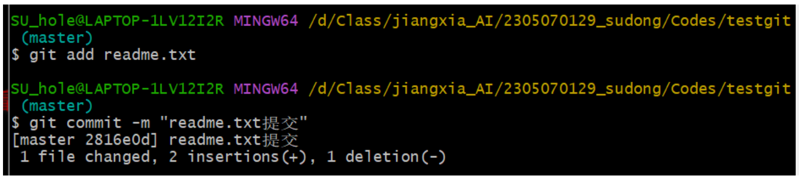
  - 文件已被修改，但是未被提交的修改

- `git diff readme.txt`



```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git diff readme.txt
diff --git a/testgit/readme.txt b/testgit/readme.txt
index 7e18e6e..2bbf87e 100644
--- a/testgit/readme.txt
+++ b/testgit/readme.txt
@@ -1,2 +1,3 @@
 test
-hello world
\ No newline at end of file
+hello world
+1111
\ No newline at end of file
```

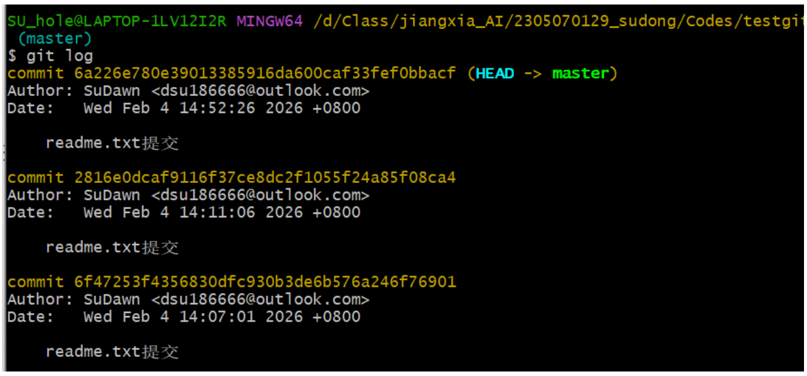
- 知道了对readme.txt文件做了什么修改后，我们可以放心的提交到仓库了，提交修改和提交文件是一样的2步



```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git add readme.txt

SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git commit -m "readme.txt提交"
[master 2816e0d] readme.txt提交
 1 file changed, 2 insertions(+), 1 deletion(-)
```

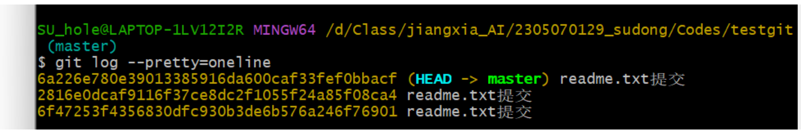
  - `git add`
  - `git commit`

- 版本回退
  - `git log`



```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git log
commit 6a226e780e39013385916da600caf33fef0bbacf (HEAD -> master)
Author: SuDawn <dsu186666@outlook.com>
Date:   Wed Feb 4 14:52:26 2026 +0800

    readme.txt提交

commit 2816e0dcaf9116f37ce8dc2f1055f24a85f08ca4
Author: SuDawn <dsu186666@outlook.com>
Date:   Wed Feb 4 14:11:06 2026 +0800

    readme.txt提交

commit 6f47253f4356830dfc930b3de6b576a246f76901
Author: SuDawn <dsu186666@outlook.com>
Date:   Wed Feb 4 14:07:01 2026 +0800

    readme.txt提交
```

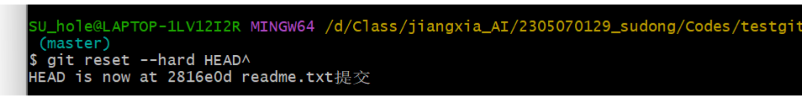
    - `git log -pretty=oneline`



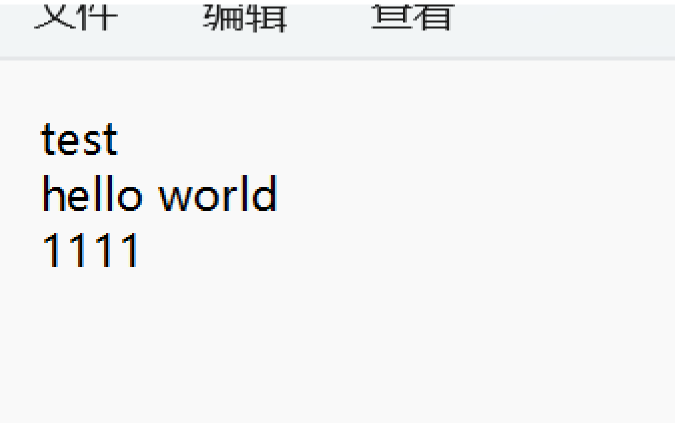
```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git log --pretty=oneline
6a226e780e39013385916da600caf33fef0bbacf (HEAD -> master) readme.txt提交
2816e0dcaf9116f37ce8dc2f1055f24a85f08ca4 readme.txt提交
6f47253f4356830dfc930b3de6b576a246f76901 readme.txt提交
```

  - 使用版本回退操作，把当前的版本回退到上一个版本

  - `git reset --hard HEAD^`

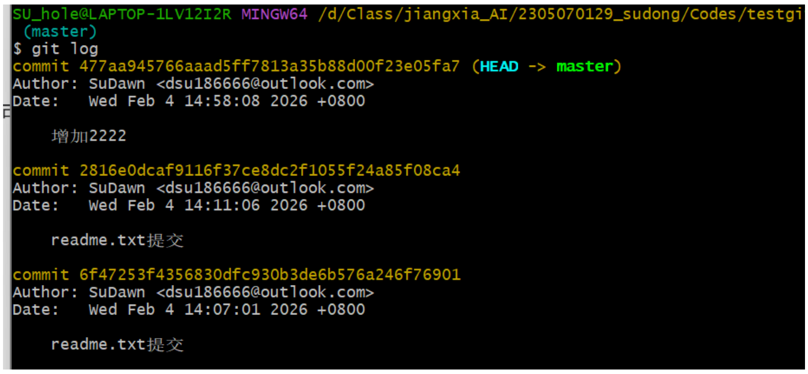
    ```
    SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
     (master)
    $ git reset --hard HEAD^
    HEAD is now at 2816e0d readme.txt提交
    ```

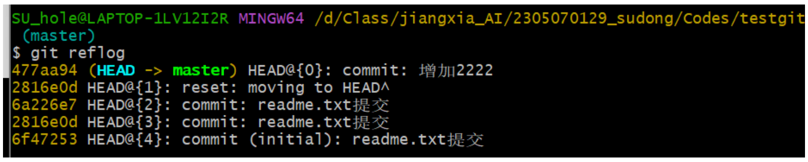
    ```
    test
    hello world
    1111
    ```

    - 那么如果要回退到上上个版本只需把HEAD^ 改成 HEAD^^ 以此类推。
    - 那如果要回退到前100个版本的话，使用上面的方法肯定不方便，我们可以使用下面的简便命令操作： `git reset --hard HEAD~100` 即可。

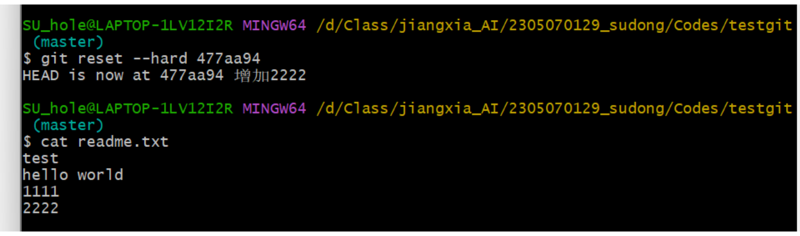
  - `git reset --hard 版本号`

    ```
    SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgi
     (master)
    $ git log
    commit 477aa945766aaad5ff7813a35b88d00f23e05fa7 (HEAD -> master)
    Author: SuDawn <dsu186666@outlook.com>
    Date:   Wed Feb 4 14:58:08 2026 +0800

        增加2222

    commit 2816e0dcaf9116f37ce8dc2f1055f24a85f08ca4
    Author: SuDawn <dsu186666@outlook.com>
    Date:   Wed Feb 4 14:11:06 2026 +0800

        readme.txt提交

    commit 6f47253f4356830dfc930b3de6b576a246f76901
    Author: SuDawn <dsu186666@outlook.com>
    Date:   Wed Feb 4 14:07:01 2026 +0800

        readme.txt提交
    ```

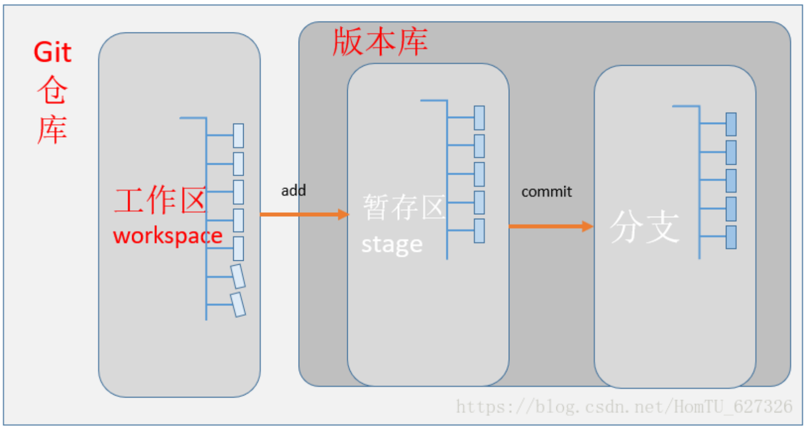
    

    ```
    SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
     (master)
    $ git reflog
    477aa94 (HEAD -> master) HEAD@{0}: commit: 增加2222
    2816e0d HEAD@{1}: reset: moving to HEAD^
    6a226e7 HEAD@{2}: commit: readme.txt提交
    2816e0d HEAD@{3}: commit: readme.txt提交
    6f47253 HEAD@{4}: commit (initial): readme.txt提交
    ```

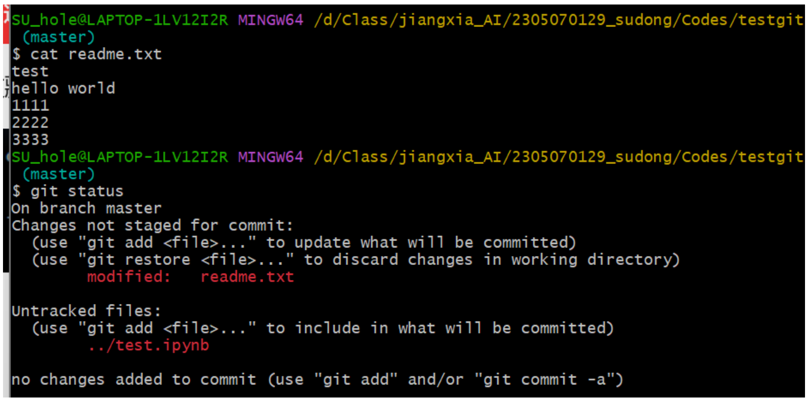
    ```
    SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
     (master)
    $ git reset --hard 477aa94
    HEAD is now at 477aa94 增加2222

    SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
     (master)
    $ cat readme.txt
    test
    hello world
    1111
    2222
    ```

- 工作区与暂存区
  - 工作区：就是你在电脑上看到的目录，比如目录下testgit里的文件(.git隐藏目录版本库除外)。或者以后需要再新建的目录文件等等都属于工作区范畴。
  - 版本库(Repository)：工作区有一个隐藏目录.git,这个不属于工作区，这是版本库。其中版本库里面存了很多东西，其中最重要的就是stage(暂存区)(或者叫index)，还有Git为我们自动创建了第一个分支master,以及指向master的一个指针HEAD。
  - 使用Git提交文件到版本库有两步：

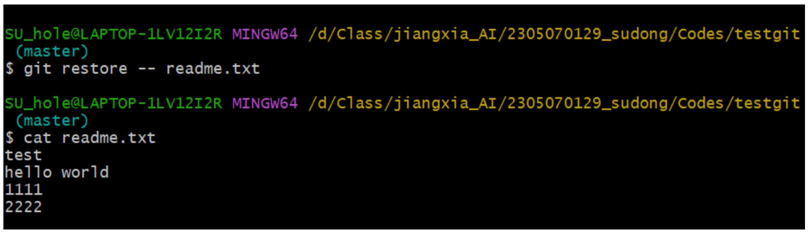


  - 第一步：是使用 git add 把文件添加进去，实际上就是把文件添加到暂存区。
  - 第二步：使用git commit提交更改，实际上就是把暂存区的所有内容提交到当前分支上。

- 撤销修改
  - 撤销修改

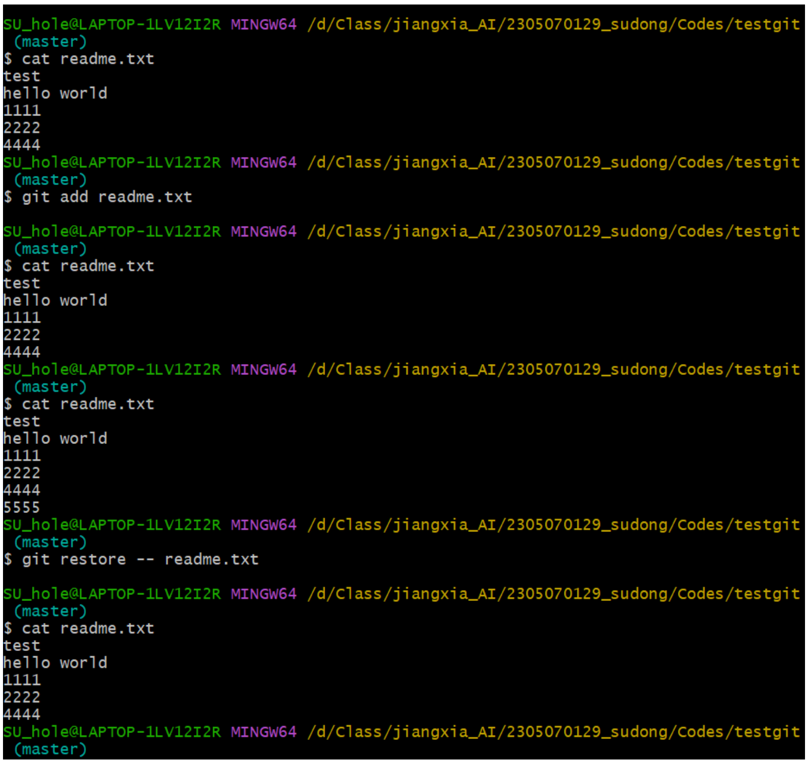


```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ cat readme.txt
test
hello world
1111
2222
3333
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   readme.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ../test.ipynb

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



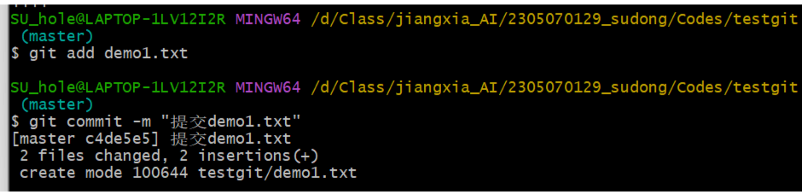
```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git restore -- readme.txt

SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ cat readme.txt
test
hello world
1111
2222
```

  - 通过撤销命令让其回到暂存区后的状态

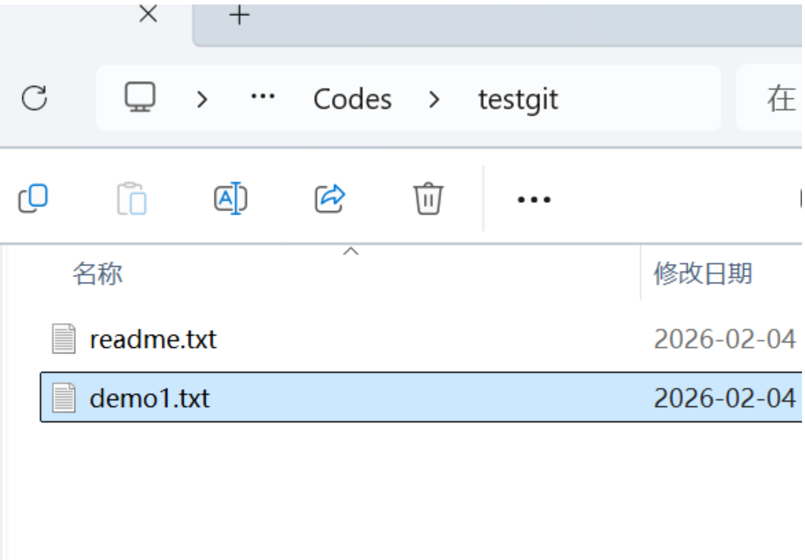


```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ cat readme.txt
test
hello world
1111
2222
4444
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git add readme.txt

SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ cat readme.txt
test
hello world
1111
2222
4444
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ cat readme.txt
test
hello world
1111
2222
4444
5555
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git restore -- readme.txt

SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ cat readme.txt
test
hello world
1111
2222
4444
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
```

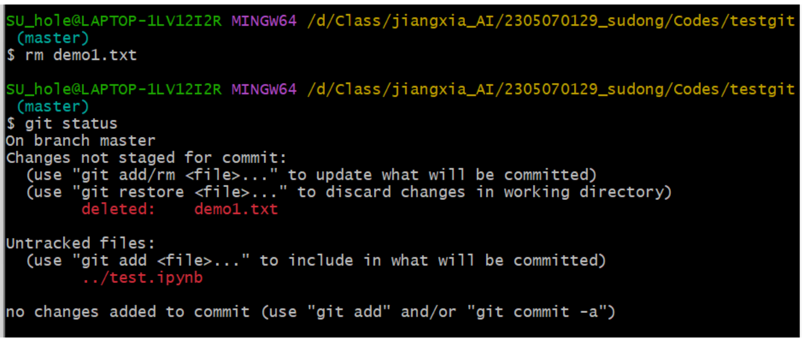
- 删除文件



```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git add demo1.txt

SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git commit -m "提交demo1.txt"
[master c4de5e5] 提交demo1.txt
 2 files changed, 2 insertions(+)
 create mode 100644 testgit/demo1.txt
```



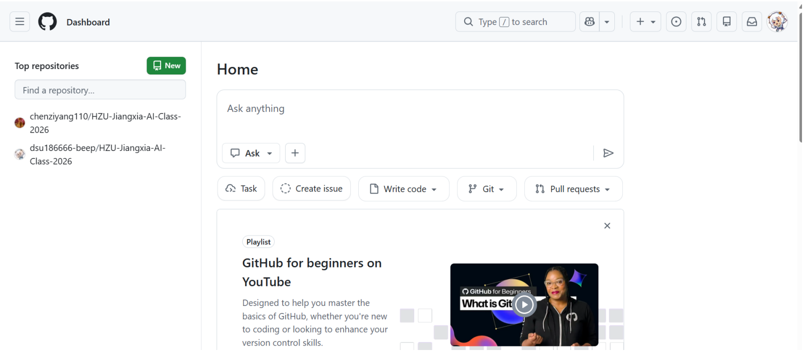
名称	修改日期
readme.txt	2026-02-04
demo1.txt	2026-02-04



```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ rm demo1.txt

SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ 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:    demo1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ../test.ipynb

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

- 远程仓库

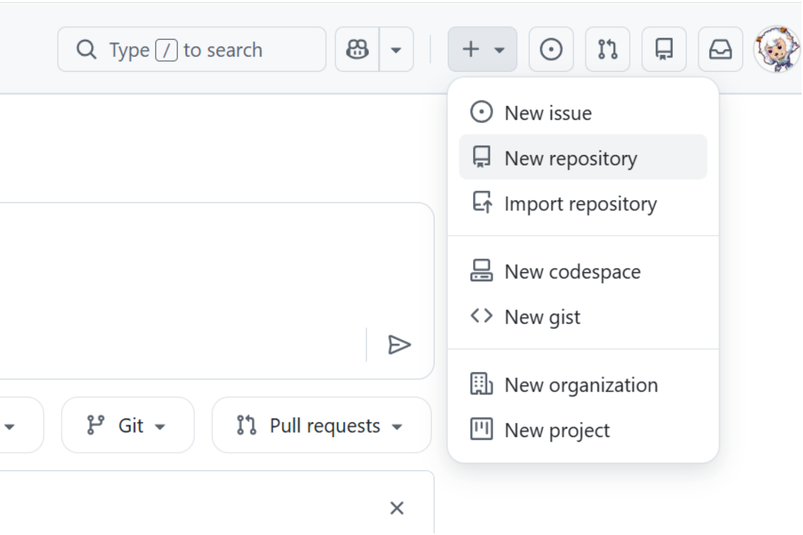
- 现在我们已经在本地创建了一个Git仓库，又想让其他人来协作开发，此时就可以把本地仓库同步到远程仓库，同时还增加了本地仓库的一个备份。
- 常用的远程仓库就是github：https://github.com/，接下来演示如何将本地代码同步到github。

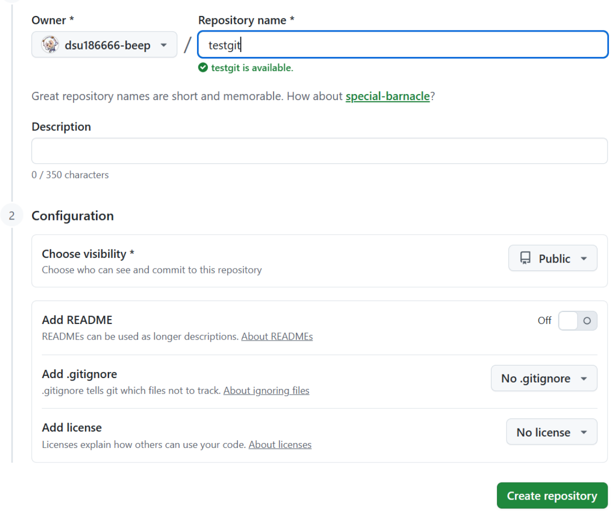


GitHub for beginners on YouTube

Designed to help you master the basics of GitHub, whether you're new to coding or looking to enhance your version control skills.

- 创建仓库





Owner * dsu186666-beep / Repository name * testgit

testgit is available.

Great repository names are short and memorable. How about special-barnacle?

Description

0 / 350 characters

2 Configuration

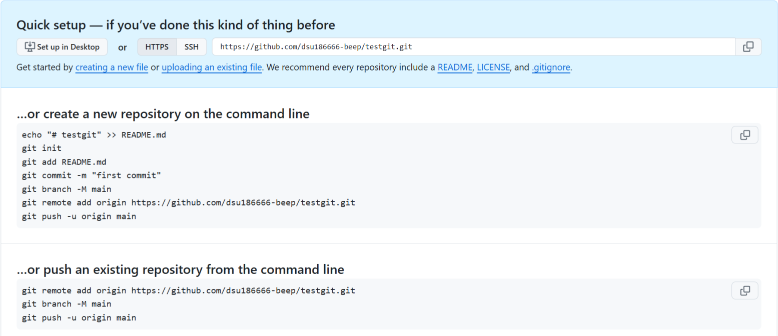
Choose visibility * Public
Choose who can see and commit to this repository

Add README Off
READMEs can be used as longer descriptions. About READMEs

Add .gitignore No .gitignore
.gitignore tells git which files not to track. About ignoring files

Add license No license
Licenses explain how others can use your code. About licenses

- Github支持两种同步方式“https”和“ssh”。



Quick setup — if you've done this kind of thing before

https://github.com/dsu186666-beep/testgit.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

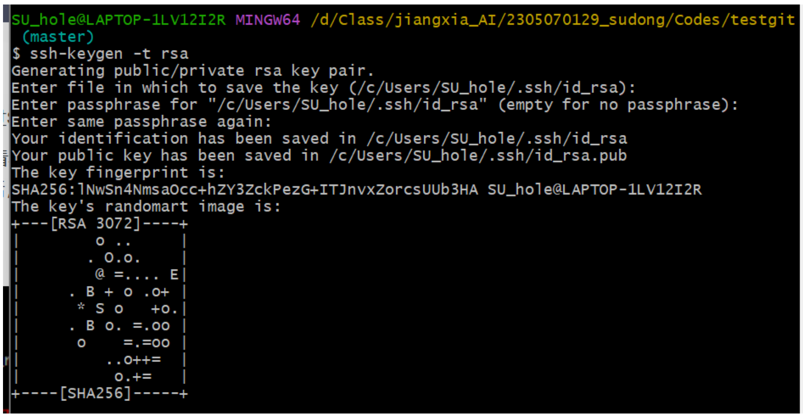
```
echo "# testgit" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/dsu186666-beep/testgit.git
git push -u origin main
```

...or push an existing repository from the command line

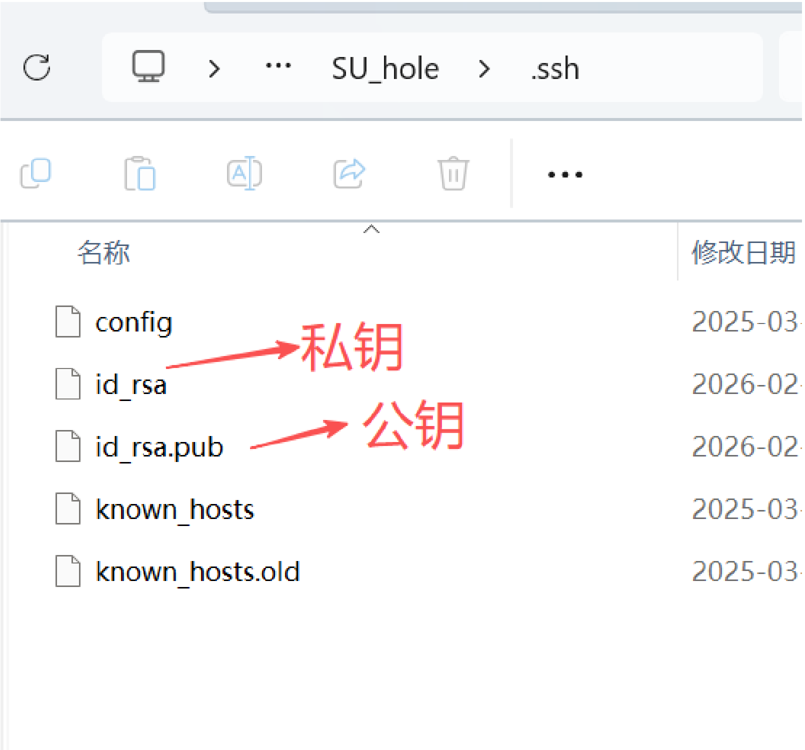
```
git remote add origin https://github.com/dsu186666-beep/testgit.git
git branch -M main
git push -u origin main
```

- 如果使用https很简单基本不需要配置就可以使用，但是每次提交代码和下载代码时都需要输入用户名和密码。
- 如果使用ssh方式就需要客户端先生成一个密钥对，即一个公钥一个私钥。然后还需要把公钥放到githib的服务器上。
    - SSH 为 Secure Shell（安全外壳协议）的缩写，由 IETF 的网络小组（Network Working Group）所制定。SSH 是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。
    - 使用ssh协议通信时，推荐使用基于密钥的验证方式。你必须为自己创建一对密匙，并把公用密匙放在需要访问的服务器上。如果你要连接到SSH服务器上，客户端软件就会向服务器发出请求，请求用你的密匙进行安全验证。服务器收到请求之后，先在该服务器上你的主目录下寻找你的公用密匙，然后把它和你发送过来的公用密匙进行比较。如果两个密匙一致，服务器就用公用密匙加密“质询”（challenge）并把它发送给客户端软件。客户端软件收到“质询”之后就可以用你的私人密匙解密再把它发送给服务器。

- ssh密钥
    - 生成 `ssh-keygen -t rsa`

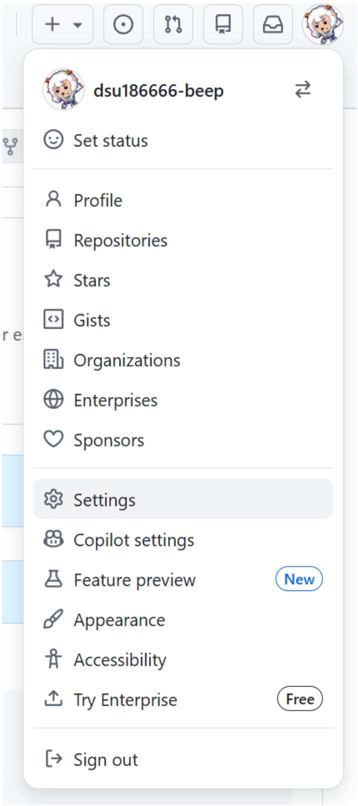


```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/SU_hole/.ssh/id_rsa):
Enter passphrase for "/c/Users/SU_hole/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/SU_hole/.ssh/id_rsa
Your public key has been saved in /c/Users/SU_hole/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:lNwSn4NmsaOcc+hZY3ZckPezG+ITJnvxZorcsUUb3HA SU_hole@LAPTOP-1LV12I2R
The key's randomart image is:
+---[RSA 3072]----+
|        o ..     |
|       . o.o.    |
|        @ =.... E|
|      . B + o .o+ |
|       * S o   +o.|
|      . B o. =.oo |
|       o    =.=oo |
|         ..o++=  |
|          o.+=   |
+----[SHA256]-----+
```

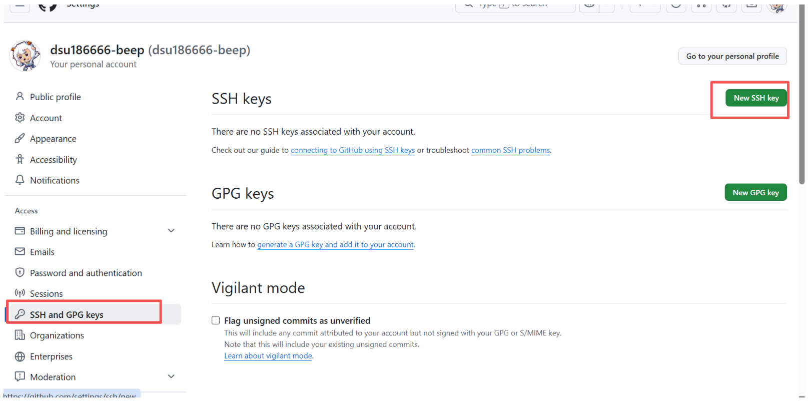


SU_hole > .ssh
名称
修改日期
config
id_rsa
id_rsa.pub
known_hosts
known_hosts.old
2025-03
2026-02
2026-02
2025-03
2025-03
私钥
公钥

- 配置



dsu186666-beep
Set status
Profile
Repositories
Stars
Gists
Organizations
Enterprises
Sponsors
Settings
Copilot settings
Feature preview
New
Appearance
Accessibility
Try Enterprise
Free
Sign out



dsu186666-beep (dsu186666-beep)
Your personal account

SSH keys

There are no SSH keys associated with your account.

Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

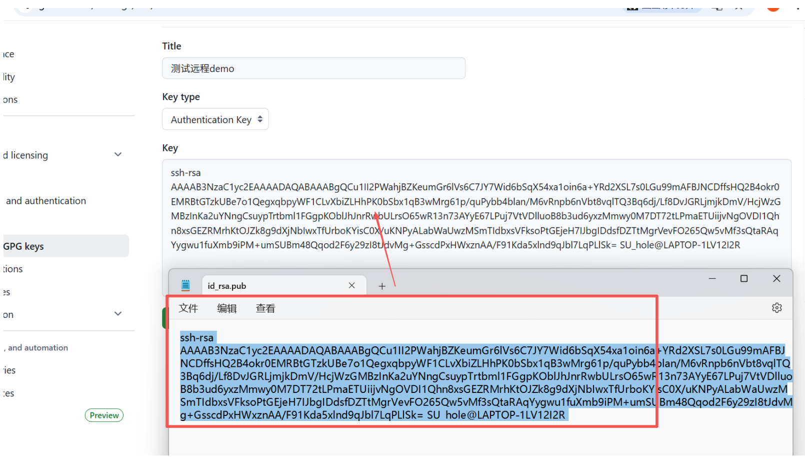
GPG keys

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

Vigilant mode

□ Flag unsigned commits as unverified
This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.
Learn about vigilant mode.



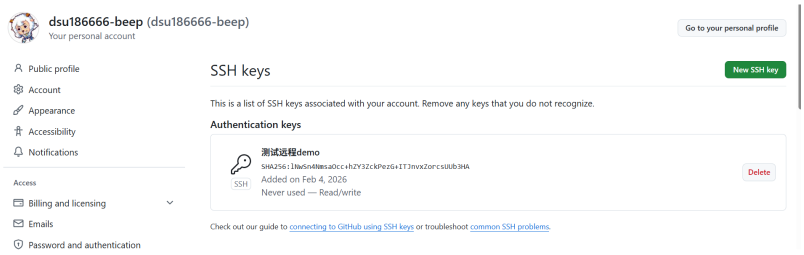
Title: 测试远程demo

Key type: Authentication Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCu1Il2PWahjBZKeumGr6lVs6C7JY7Wid6bSqX54xa1oin6a+YRd2XSL7s0LGu99mAFBJNCDffsHQ2B4okr0
EMRBtGTzkUBe7o1QegxqbpyWF1CLvXbiZLHhPK0bSbx1qB3wMrg61p/quPybb4blan/M6vRnpb6nVbt8vqlTQ3Bq6dj/Lf8DvJGRLjmjkDmV/HcjWzG
MBzInKa2uYNngCsuypTrtbml1FGgpKOblJhJnrRwbULrsO65wR13n73AYyE67LPuj7VtVDlluoB8b3ud6yxzMmwy0M7DT72tLPmaETUiijvNgOVDI1Qh
n8xsGEZRMrhKtOJZk8g9dXjNblwxTfUrboKYisC0X/uKNPyALabWaUwzMSmTIdbxsVFksoPtGEjeH7IJbgIDdsfDZTtMgrVevFO265Qw5vMf3sQtaRAq
Yygwu1fuXmb9iPM+umSUBm48Qqod2F6y29zI8tJdvMg+GsscdPxHWxznAA/F91Kda5xlnd9qJbl7LqPLlSk= SU_hole@LAPTOP-1LV12I2R
```

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCu1Il2PWahjBZKeumGr6lVs6C7JY7Wid6bSqX54xa1oin6a+YRd2XSL7s0LGu99mAFBJ
NCDffsHQ2B4okr0EMRBtGTzkUBe7o1QegxqbpyWF1CLvXbiZLHhPK0bSbx1qB3wMrg61p/quPybb4blan/M6vRnpb6nVbt8vqlTQ
3Bq6dj/Lf8DvJGRLjmjkDmV/HcjWzGMBzInKa2uYNngCsuypTrtbml1FGgpKOblJhJnrRwbULrsO65wR13n73AYyE67LPuj7VtVDlluo
B8b3ud6yxzMmwy0M7DT72tLPmaETUiijvNgOVDI1Qhn8xsGEZRMrhKtOJZk8g9dXjNblwxTfUrboKYisC0X/uKNPyALabWaUwzM
SmTIdbxsVFksoPtGEjeH7IJbgIDdsfDZTtMgrVevFO265Qw5vMf3sQtaRAqYygwu1fuXmb9iPM+umSUBm48Qqod2F6y29zI8tJdvM
g+GsscdPxHWxznAA/F91Kda5xlnd9qJbl7LqPLlSk= SU_hole@LAPTOP-1LV12I2R
```



dsu186666-beep (dsu186666-beep)
Your personal account

SSH keys

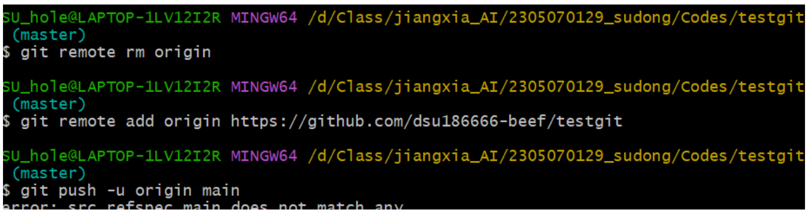
This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

测试远程demo
SHA256:lNwSn4NmsaOcc+hZY3ZckPezG+ITJnvxZorcsUUb3HA
Added on Feb 4, 2026
Never used — Read/write

Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

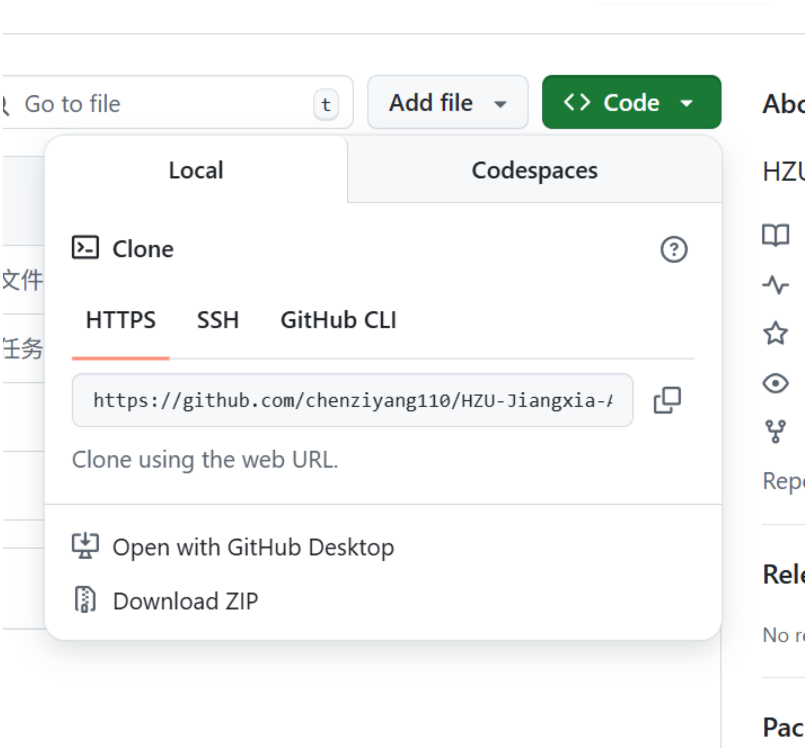
- 同步



```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git remote rm origin

SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git remote add origin https://github.com/dsu186666-beef/testgit

SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git push -u origin main
error: src refspec main does not match any
```

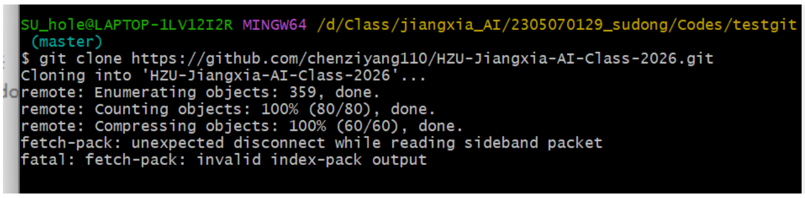
  - `git remote rm origin`

- 从远程库克隆



```
https://github.com/chenziyang110/HZU-Jiangxia-/
```

Clone using the web URL.



```
SU_hole@LAPTOP-1LV12I2R MINGW64 /d/Class/jiangxia_AI/2305070129_sudong/Codes/testgit
 (master)
$ git clone https://github.com/chenziyang110/HZU-Jiangxia-AI-Class-2026.git
Cloning into 'HZU-Jiangxia-AI-Class-2026'...
remote: Enumerating objects: 359, done.
remote: Counting objects: 100% (80/80), done.
remote: Compressing objects: 100% (60/60), done.
fetch-pack: unexpected disconnect while reading sideband packet
fatal: fetch-pack: invalid index-pack output
```

- `git clone`

-