

Git

Git属于分散性版本管理系统，是为版本管理而设计的软件。

Linux--Linus在05年开发了git原型程序

SVN，CVS集中式的版本控制系统，集中存放中央服务器的

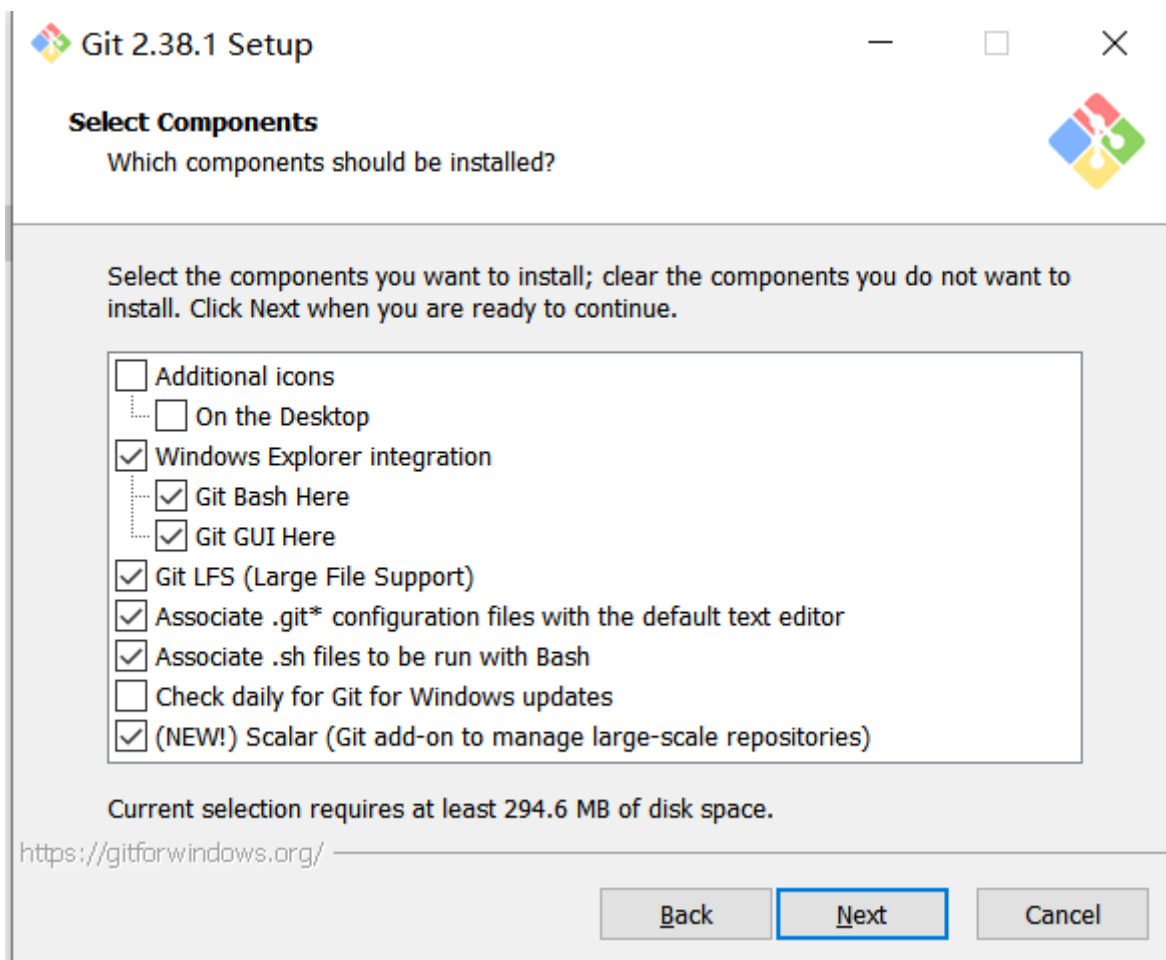
Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Git 是一个免费的开源分布式版本控制系统，旨在快速高效地处理从小型到超大型项目的所有内容。Git 易于学习，占用空间很小，性能快如闪电。它超越了Subversion，CVS，Perforce和ClearCase等SCM工具，具有廉价的本地分支，方便的暂存区域和多个工作流程等功能。

安装

组件的选择，默认就可以



注册账号

[Gitee - 基于 Git 的代码托管和研发协作平台](#)

<https://github.com/>

设置

设置姓名和邮箱

```
git config --global user.name "你的名字"  
git config --global user.email "你的邮箱"
```

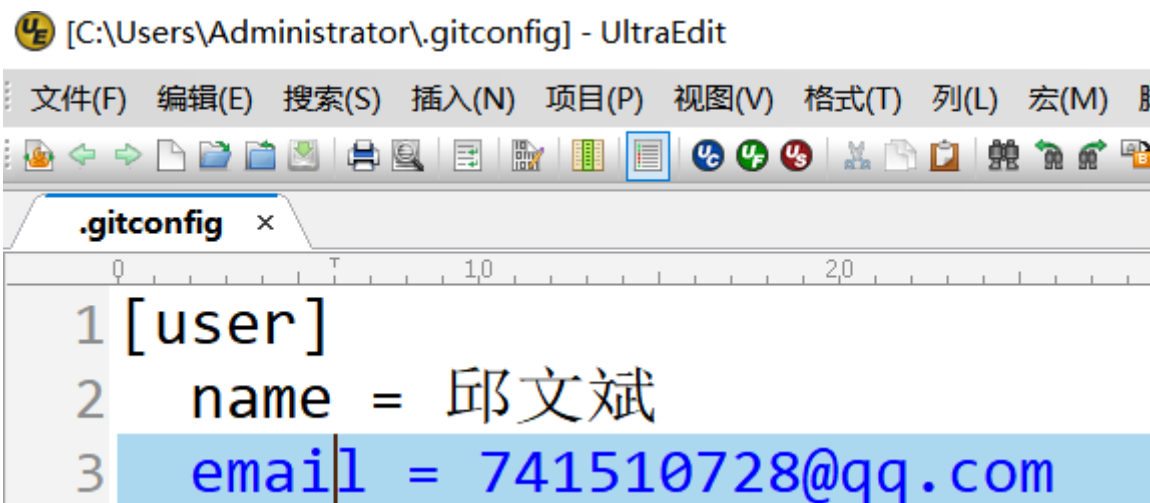
git config命令指定与git相关的配置

config配置级别：

- system：系统
- global：用户
- local：当前仓库

范围system>global>local

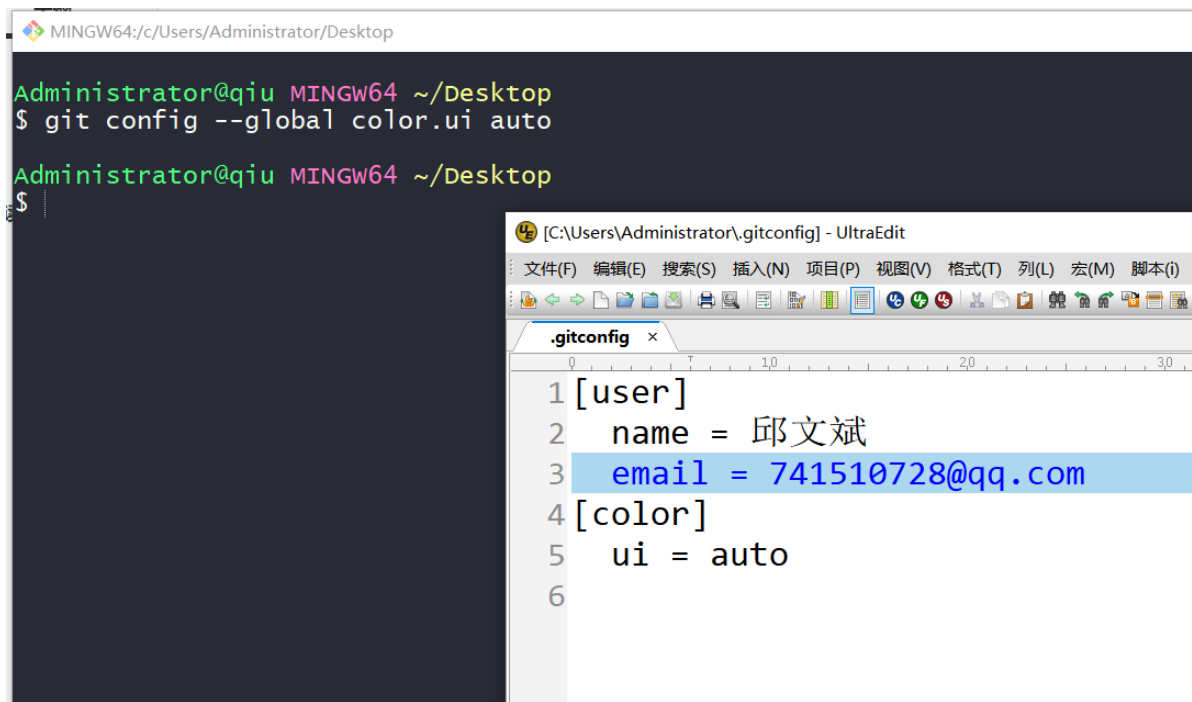
```
# 查看git配置  
git config --list
```



提高命令输出的可读性

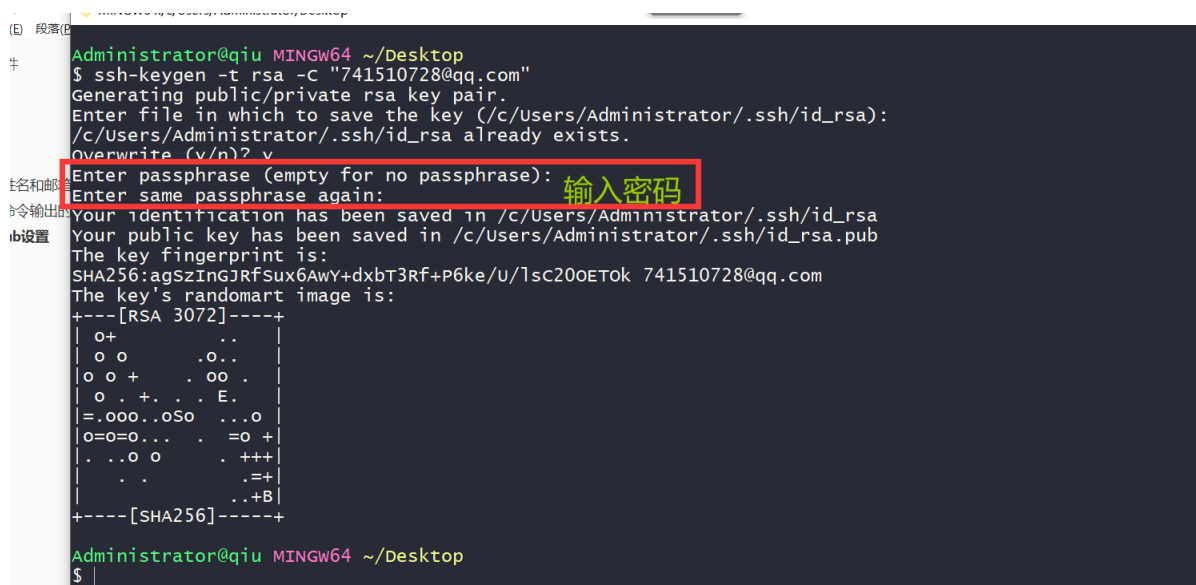
设置color.ui为auto，让输出有更高的可读性

```
git config --global color.ui auto
```



GitHub设置

```
# 1、打开git bash
# 2、生成sshkey
ssh-keygen -t rsa -C "741510728@qq.com"
```



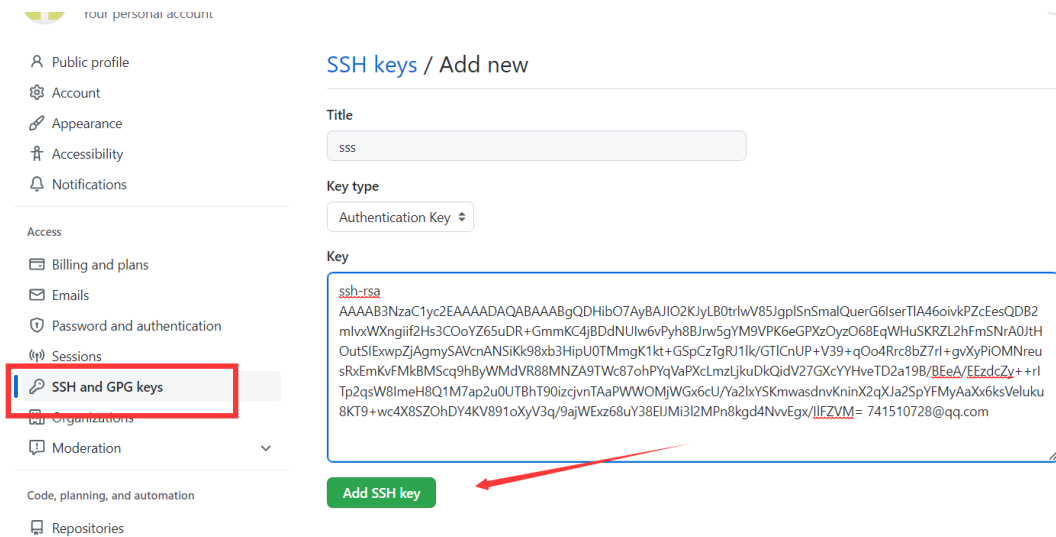
然后打开路径

```
C:\Users\Administrator\.ssh
```

Acer (C:) > 用户 > Administrator > .ssh			
名称	修改日期	类型	大小
<input type="checkbox"/> id_rsa	2022/11/26 10:17	文件	3 KB
<input checked="" type="checkbox"/> id_rsa.pub	2022/11/26 10:17	PUB 文件	1 KB

分别是私钥和公钥

我们用记事本打开公钥id_rsa.pub，然后全选复制所有内容



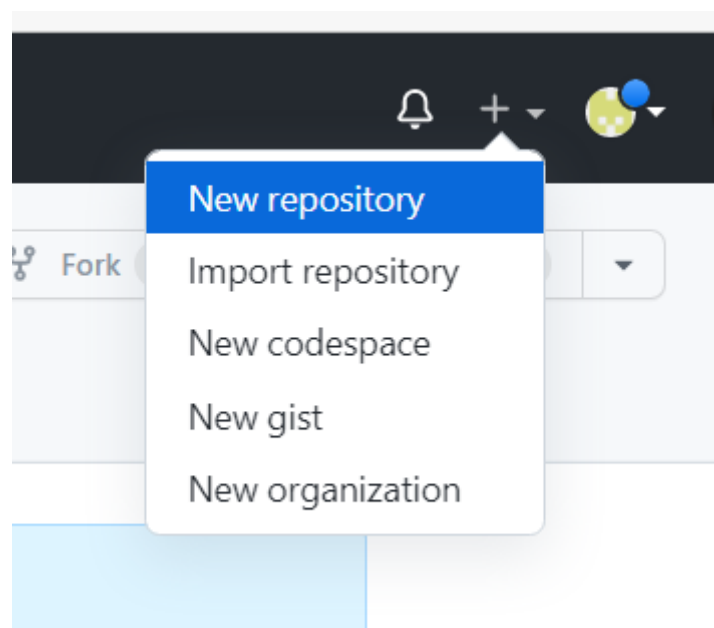
完成设置后，邮箱会收到提示邮件，这个时候，我们就可以使用私钥和GitHub进行认证和通信了。

执行命令验证

```
ssh -T git@github.com
```

```
Administrator@qiu MINGW64 ~/Desktop
$ ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3vVVV6TuJJhpZisF/zLDA0zPMSvHdkr4UvCoQu.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enter passphrase for key '/c/Users/Administrator/.ssh/id_rsa':
Hi waqwb! You've successfully authenticated, but GitHub does not provide shell access.
```

创建仓库



Create a new repository

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

Owner *

waqwb

Repository name *

Hello-World

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

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:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

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

.gitignore template: None

Choose a license

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

License: None

You are creating a public repository in your personal account.

Create repository

```
cd /f/tarena/授课/枣庄/VIP创新课程（行业）/code
# 把github上的仓库，克隆到本地
git clone https://github.com/waqwb/HelloWorld.git
```

Doc (F:) > tarena > 授课 > 枣庄 > VIP创新课程（行业） > code > HelloWorld >

名称	修改日期	类型	大小
.git	2022/11/26 10:43	文件夹	

```
Administrator@qiu MINGW64 /f/tarena/授课/枣庄/VIP创新课程（行业）/code
$ cd HelloWorld/

Administrator@qiu MINGW64 /f/tarena/授课/枣庄/VIP创新课程（行业）/code/HelloWorld
(main)
$ pwd
/f/tarena/授课/枣庄/VIP创新课程（行业）/code/HelloWorld



Administrator@qiu MINGW64 /f/tarena/授课/枣庄/VIP创新课程（行业）/code/HelloWorld
(main)
$ echo "# HelloWorld" >> README.md
```

布局

当前视图

显示/隐藏

此电脑 > Doc (F:) > tarena > 授课 > 枣庄 > VIP创新课程 (行业) > code > HelloWorld

<input type="checkbox"/> 名称	修改日期	类型	大小
<input type="checkbox"/>  .git	2022/11/26 10:43	文件夹	
<input type="checkbox"/>  README.md	2022/11/26 10:51	MD 文件	11 B

```
git add README.md
git commit -m "first commit"
```

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Hello.java
```

```
Administrator@qiu MINGW64 /f/tarena/授课/枣庄/VIP创新课程
d (main)
$ git commit -m "second commit"
[main 7e8b8cf] second commit
1 file changed, 5 insertions(+)
create mode 100644 Hello.java

Administrator@qiu MINGW64 /f/tarena/授课/枣庄/VIP创新课程
d (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

此时只是提交操作，并不会把我们目录中的内容上传至网站

```
git push -u origin main
```

第一次push的时候，会在git bash中有一个登录github的验证，大家验证即可

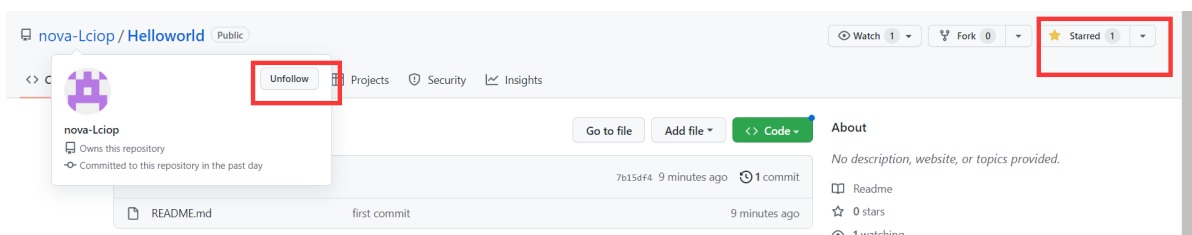
```
Administrator@qiu MINGW64 /f/tarena/授课/枣庄/VIP创新课程（行业）/code/HelloWorld (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 376 bytes | 376.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/waqwb/HelloWorld.git
   39246d3..7e8b8cf  main -> main
branch 'main' set up to track 'origin/main'.

Administrator@qiu MINGW64 /f/tarena/授课/枣庄/VIP创新课程（行业）/code/HelloWorld (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

出现以下内容证明成功了，在浏览器中刷新GitHub中的仓库

番外

关注作者和star仓库



```
cd HelloWorld
echo "# HelloWorld" >> README.md
# 当仓库没有HelloWorld仓库的时候 .git
git init
git add README.md
git commit -m "first commit"
git branch -M main
# 指定我们要上传到的仓库的地址
git remote add origin https://github.com/waqwb/HelloWorld.git
git push -u origin main
```

练习

新建一个hello.java,写一段代码添加到文件中

上传至GitHub中