



浙大宁波理工学院
NINGBOTech UNIVERSITY

信号与系统分析

课程实验报告

专业：电子信息工程

班别：212 班

姓名：王振宇

学号：3210435016

浙大宁波理工学院实验报告

姓名 王振宇 学号 3210435016 学院 信息科学与工程学院
专业 电子信息工程 班级 212 班 课程 信号与系统分析
实验时间 2022-9-14 实验地点 SI302 指导教师 王一刚

实验一

一、实验目的

PYTHON 和 GITHUB 开发环境搭建

二、实验内容

1. 申请 github 账号
2. 安装 VCCODE
3. 能够实现在电脑端和云端的数据同步
4. 记录中间实现过程中的每一步的关键细节

三、实验过程

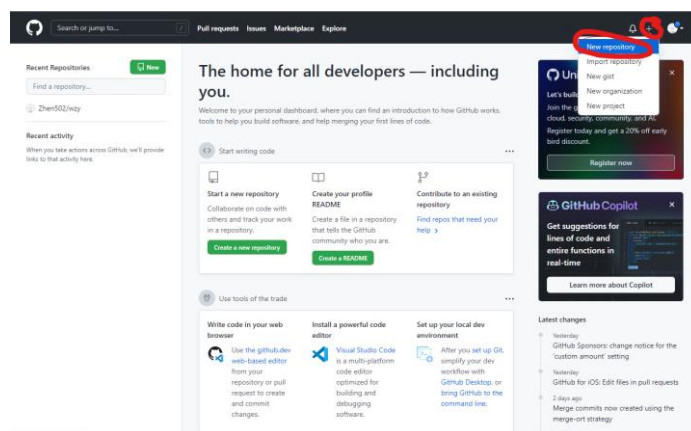
1. GitHub 账号注册



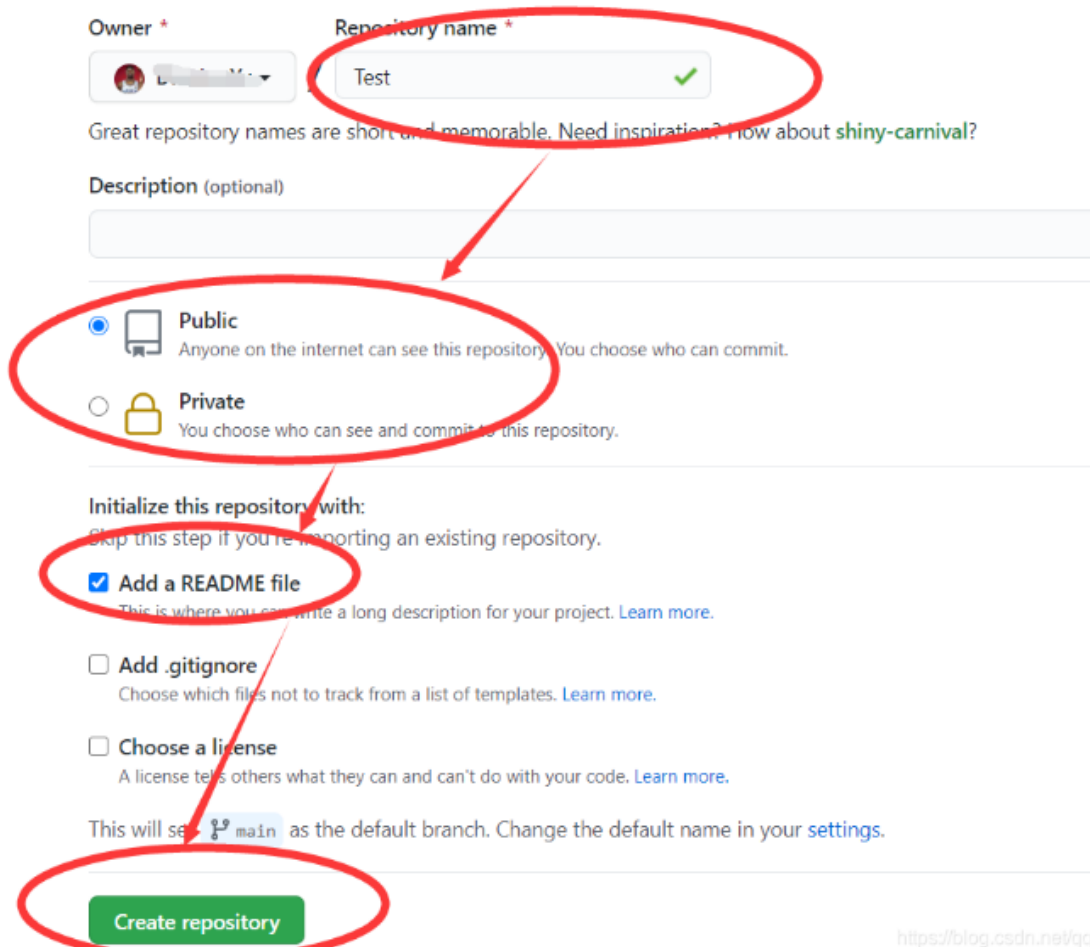
进入上述所示官网：自行注册账号，并记住账号、密码以及注册所使用的邮箱。

2. GitHub 创建自己的仓库

进入自己的 github 账号，我的账号如下图所示：



点击加号，展开列表；并点击列表下的 New repository 进入到创建仓库界面。



第一个红色方框内输入你的仓库名称，自由命名，合法即可。随后按照上述箭头所示流程勾选，然后点击最下面的创建仓库的绿色按钮。仓库创建好之后，开始下载 Git。

3. 下载安装 Git

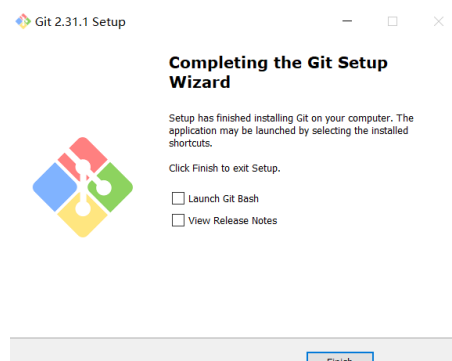
下载链接：<https://git-scm.com/>

根据你电脑的位数选择下载的版本

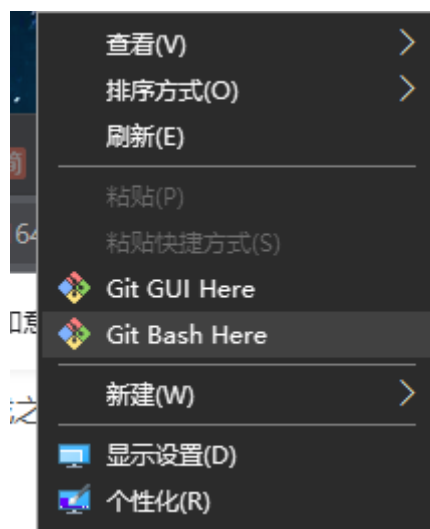
如果是 32 位则选择红色方框中对应的：32-bit Git for Windows Setup

如果是 64 位则选择红色方框中对应的 64 位：64-bit Git for Windows Setup。

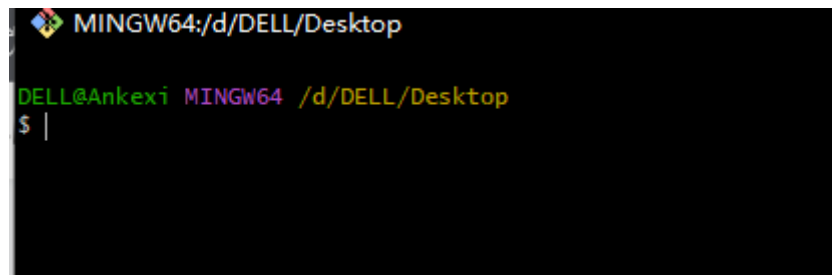
等待其安装完毕即可，最后得到下图所示，点击 finish，完成即可：



点击完成之后，测试 git 是否安装成功，在桌面空白处右键，找到 Git，并点击 Git Bash：



得到如下图所示界面之后，表示安装成功：



随便关闭界面，以上就算 Git 的安装过程。

4. Git 连接 GitHub

4.1 初始化 git

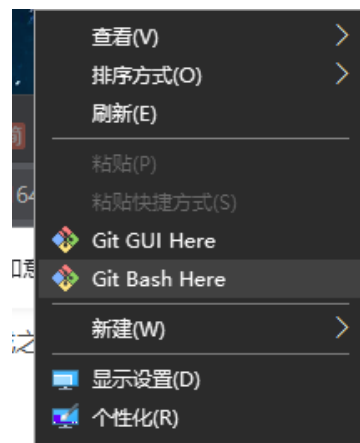
GitHub 支持两种同步方式“https”和“ssh”。

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 GitHub 的服务器上。

直接演示 ssh 方式

首先找到，你的项目文件下面。然后右键，得到下图所示界面：



随后点击 git bash here 弹出了 git 命令控制台！

因为 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。

刚开始在控制命令台输入以下命令：

`git config --global user.name "Zhen502"` 用户名标识 ---- 实际也可以填写您的 GitHub 仓库的名称

`git config --global user.email "pellam@163.com"` 邮箱标识 --- ----可以填写 GitHub 仓库的邮箱

如下图所示：

```
DELL@Ankexi MINGW64 /d/DELL/Desktop/wzy (main)
$ git config --global user.name "Zhen502"

DELL@Ankexi MINGW64 /d/DELL/Desktop/wzy (main)
$ git config --global user.email "pellam@163.com"

DELL@Ankexi MINGW64 /d/DELL/Desktop/wzy (main)
$
```

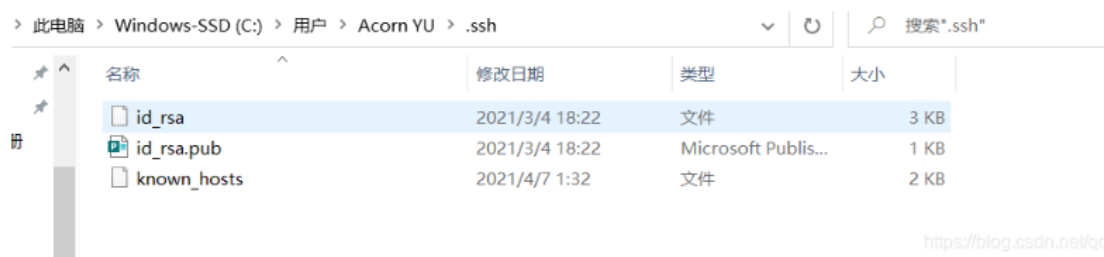
注意：`git config --global` “参数”，有了这个参数，表示你这台机器上所有的 Git 仓库都会使用这个配置，当然你也可以对某个仓库指定的不同的用户名和邮箱。

4.2 github 连接 git

4.2.1 创建 SSH Key

在 git 命令界面输入如下命令：`ssh-keygen -t rsa`

然后按回车键即可。进入这个文件下面 `C:\Users\Administrator.ssh`，得到下图所示：

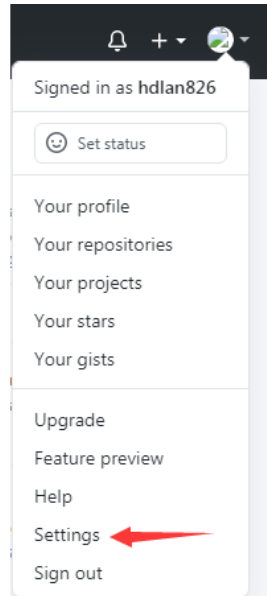


用记事本打开上述图片中的公钥，复制里面的内容，随后开始在 github 中配置 ssh 密钥。

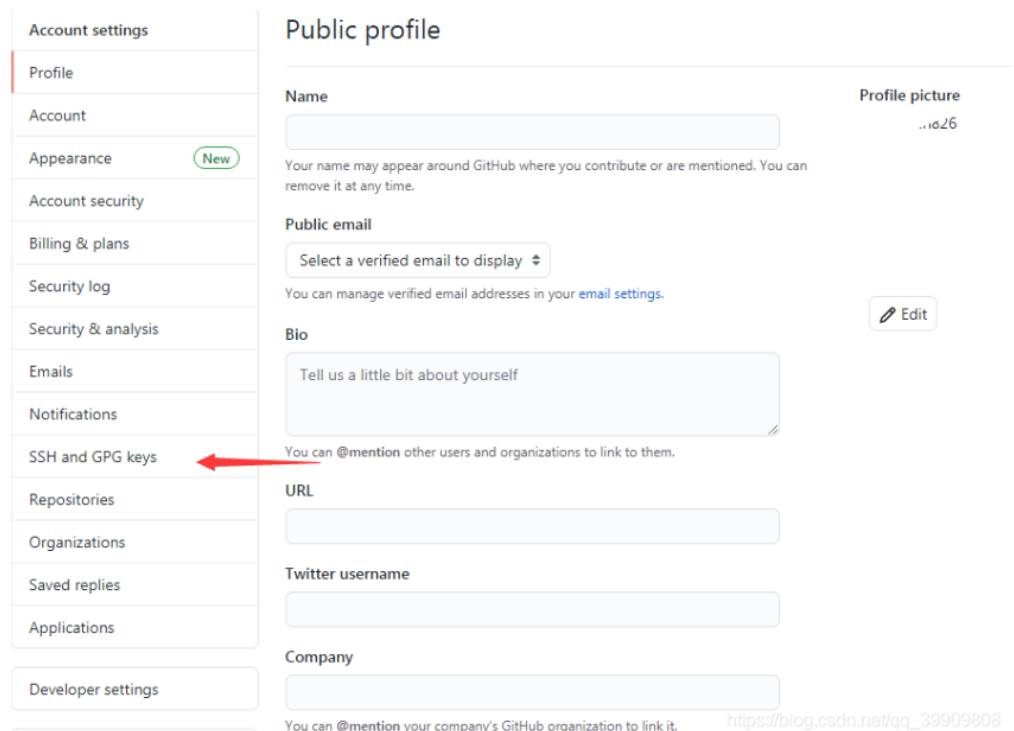
4.2.2 远程 github 配置 ssh 秘钥

密钥生成后需要在 github 上配置密钥本地才可以顺利访问。

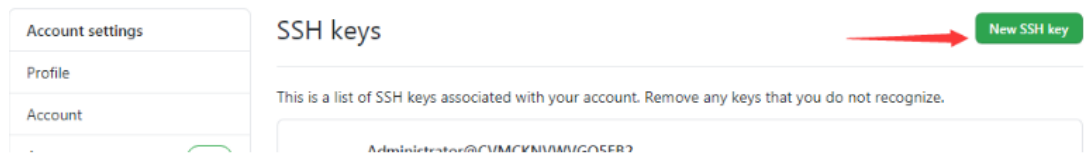
进入 github 右上角你账号的头像选择 settings



进去之后选择 SSH and GPG keys

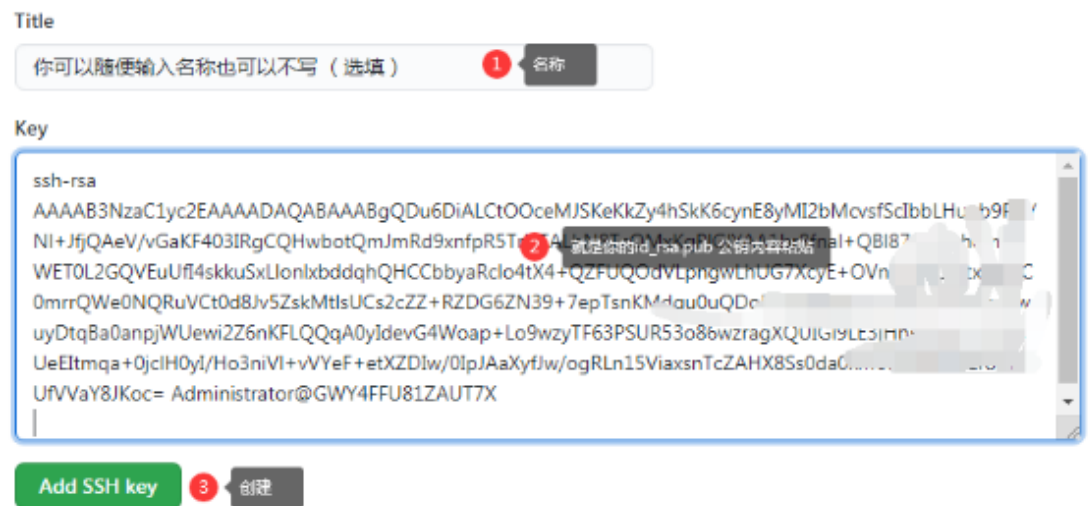


点击之后出现这个界面：



点击 New SSH key

SSH keys / Add new



4.2.4 连接指定仓库

进入你的项目下面，右键 git，进入到 git bash here 的界面，然后使用以下命令：
git remote
看是否有远程仓库源，如图所示：

```
DELL@Ankexi MINGW64 /d/DELL/Desktop/wzy (main)
$ git remote
origin
```

出现 origin 说明我的仓库是有信息的
如果出现这种情况：

```
Administrator@GWY4FFU81ZAUT7X MINGW64 /e/2.编程/3.git/zsysblog
$ git remote
fatal: not a git repository (or any of the parent directories): .git
```

出现以上错误就是文件夹没有 git init 他不是一个仓库文件夹，在重新输入 git init:

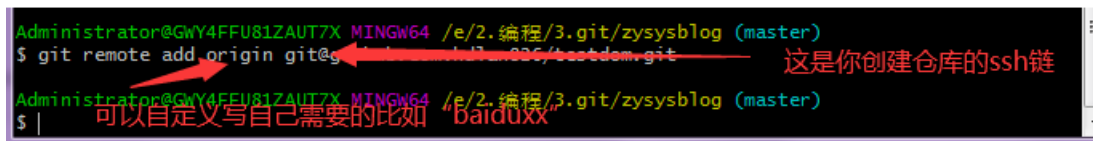
```
Administrator@GWY4FFU81ZAUT7X MINGW64 /e/2.编程/3.git/zsysblog
$ git init
Initialized empty Git repository in E:/2.编程/3.git/zsysblog/.git/
```

之后在输入：git remote

```
Administrator@GWY4FFU81ZAUT7X MINGW64 /e/2.编程/3.git/zsysblog (master)
$ git remote
```


没有任何显示就是没有仓库信息，下面就需要创建仓库信息：
在界面中，输入下述命令：

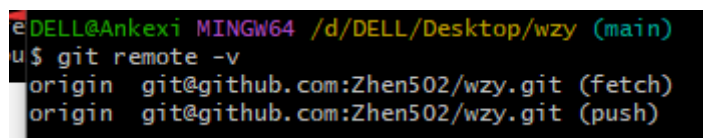
```
git remote add origin git@github.com:Zhen502/wzy.git
```



```
Administrator@GWY4FFU81ZAUT7X MINGW64 /e/2.编程/3.git/zsysblog (master)
$ git remote add origin git@github.com:Zhen502/wzy.git
Administrator@GWY4FFU81ZAUT7X MINGW64 /e/2.编程/3.git/zsysblog (master)
$ |
```

这样我们就已经连接到了，可以看看，输入以下命令

```
git remote -v
```



```
eDELL@Ankexi MINGW64 /d/DELL/Desktop/wzy (main)
u$ git remote -v
origin git@github.com:Zhen502/wzy.git (fetch)
origin git@github.com:Zhen502/wzy.git (push)
```

已经成功了。

把本地库的内容推送到远程，使用 `git push origin master` 命令，实际上是当前分支 `master` 推送到远程

由于远程库是空的，我们第一次推送 `master` 分支时，加上 `-u` 参数，Git 不但会把本地的 `master` 分支内容推送的远程新的 `master` 分支，还会把本地的 `master` 分支和远程的 `master` 分支关联起来，推送成功后，可以立刻在 `github` 页面中看到远程库的内容已经和本地一模一样了

只要本地作了提交，就可以通过如下命令：`git push origin master` 把本地 `master` 分支的最新修改推送到 `github` 上了。

以上就完成了 `github` 与 `git` 的连接。

3.2.5 从项目文件夹提交代码到 github

在你的项目文件夹下面，右键进入到 Git Bash Here:

进入之后，输入，git status 命令：

```
Acorn YU@LAPTOP-15UGPJ0P MINGW64 /d/paper/GraduationThesis (master)
$ git status
On branch master
Your branch is up to date with 'origin/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:   "\345\274\240\344\277\212\345\256\207-\347\254\254\344\272\2
24\347\211\210.docx"

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

我在这里第一次输入时，显示 nothing to commit, working tree clean 表示，我没有项目需要更新，上传到 github。第二次我修改文件内容之后，在输入 git status 时，提示有内容修改，需要与 GitHub 代码同步。随后输入 git add .

如下图所示：

```
Acorn YU@LAPTOP-15UGPJ0P MINGW64 /d/paper/GraduationThesis (master)
$ git add .
```

继续输入命令：git commit -m"提交更新"

```
DELL@Ankexi MINGW64 /d/DELL/Desktop/wzy (main)
$ git commit -m"提交更新"
[main fe68789] 提交更新
1 file changed, 1 insertion(+), 1 deletion(-)
```

最后，在页面中输入，git push 命令，并按回车键：

```
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 257 bytes | 257.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To github.com:Zhen502/wzy.git
e029cfb..fe68789 main -> main
```

返回到你的 GitHub 账号，看看你的仓库项目有没有更新内容，更新了的话表示成功。

5. 安装 VCCODE

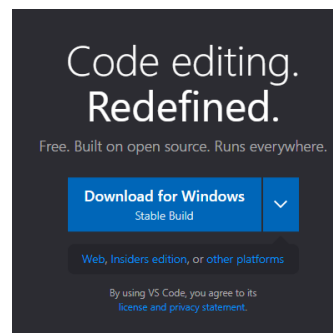
打开官网：

Visual Studio Code - Code Editing. Redefined

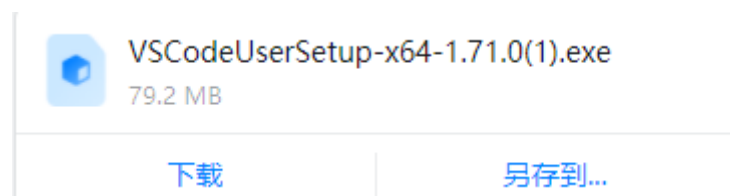
<https://code.visualstudio.com>

2022-4-7 · Visual Studio Code is a code editor redefined and optimized for built

点击下载即可

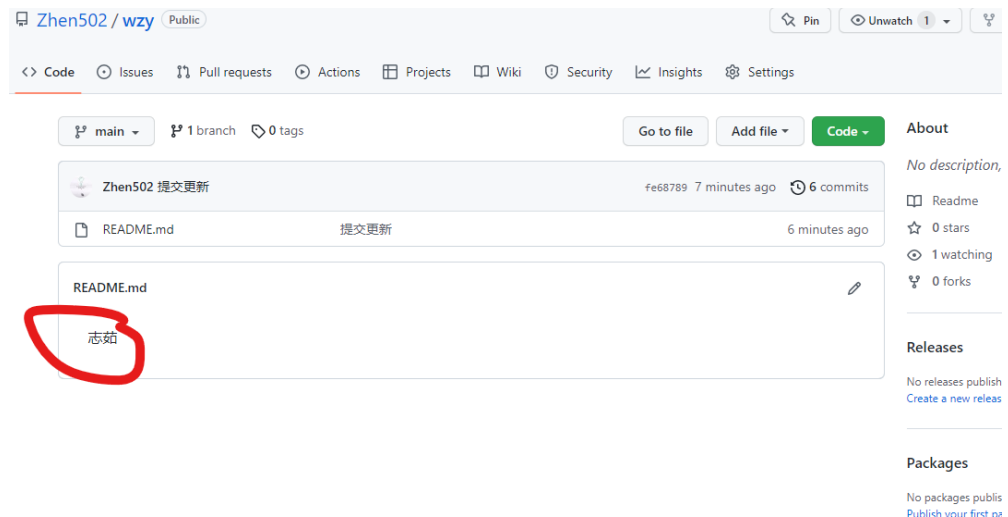


也可以使用老师给予的安装包



6. 电脑端与云端连接的前后对比

打开 github 的个人仓库，观看仓库里面的信息：



如图我仓库信息为：志茹

这是 github 仓库更改之前的信息，之后我们打开 git 对信息进行修改。

输入 `git clone git@github.com:Zhen502/wzy.git`

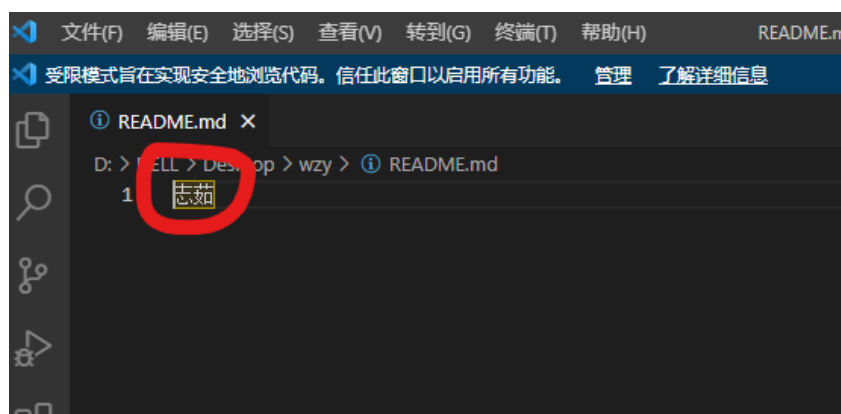
将仓库内的信息克隆下来，并保存至桌面。



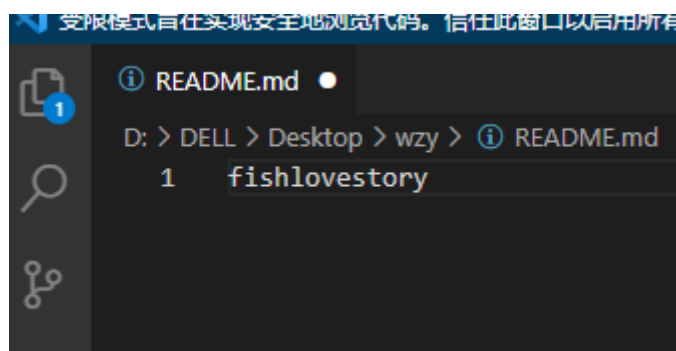
打开文件夹可发现一个隐藏文件 “.git” 以及 “README.md”

隐藏文件是必备且不可修改的，我们要对“README.md”进行需改。

用 VCCODE 打开该文件



可以看到这里的信息就是仓库里的信息，我们可以本地进行修改



这里我修改成：fishlovestory

保存退出。

在该文件夹空白处右键打开 git

重复 3 中的步骤：

输入 git status 可以发现

```
DELL@Ankexi MINGW64 /d/DELL/Desktop/wzy (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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.md

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

再次输入 git add .

继续输入命令：git commit -m"提交更新"

最后，在页面中输入，git push 命令，并按回车键：

```
DELL@Ankexi MINGW64 /d/DELL/Desktop/wzy (main)
$ git add .

DELL@Ankexi MINGW64 /d/DELL/Desktop/wzy (main)
$ git commit -m"提交更新"
[main 36ee00e] 提交更新
1 file changed, 1 insertion(+), 1 deletion(-)

DELL@Ankexi MINGW64 /d/DELL/Desktop/wzy (main)
$ git push

Enumerating objects: 5, done.
Counting objects: 20% (1/5)
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 263 bytes | 263.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To github.com:Zhen502/wzy.git
fe68789..36ee00e main -> main
```

这样就成功了，我们再回到仓库观察信息是否更改
打开个人仓库：



可以看到信息已经更改成：fishlovestory

完成。