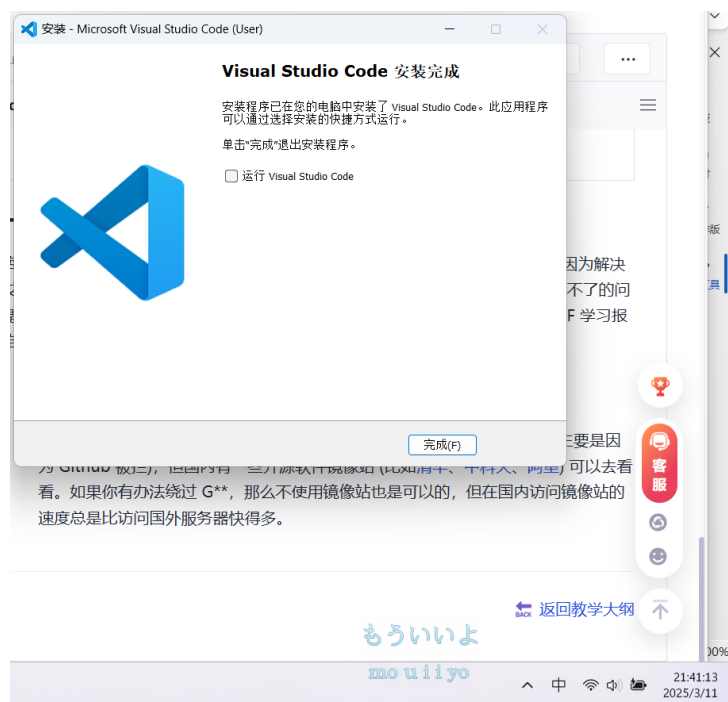


本周任务：

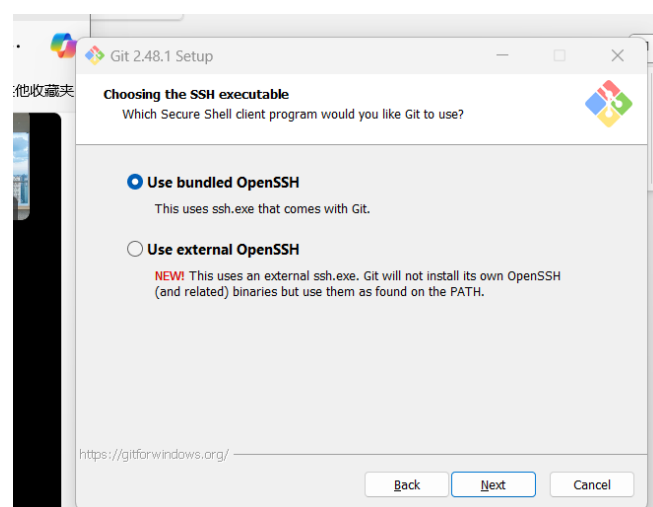
1. 安装 [VS Code](#) (代码编辑器)
2. 安装 [Git](#) (代码版本管理软件)
3. 安装 [Anaconda](#) (Python 解释器)
4. 访问 [GitCode](#) (代码托管平台) 注册用户，然后保持登录
5. 生成自己设备的 SSH 密钥 (公钥-私钥对)，将公钥添加到自己的 GitCode 安全设置里 ([参考](#))
6. 去 [第01周打卡](#) 仓库阅读说明，完成学习报告的提交

任务 1: 安装 [VS Code](#) (代码编辑器)

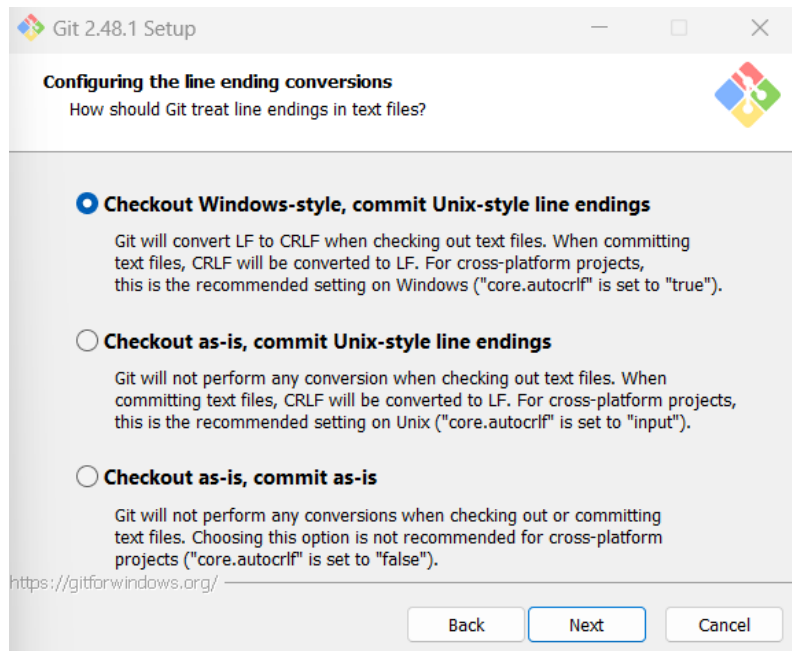


任务二：安装 [Git](#) (代码版本管理软件)

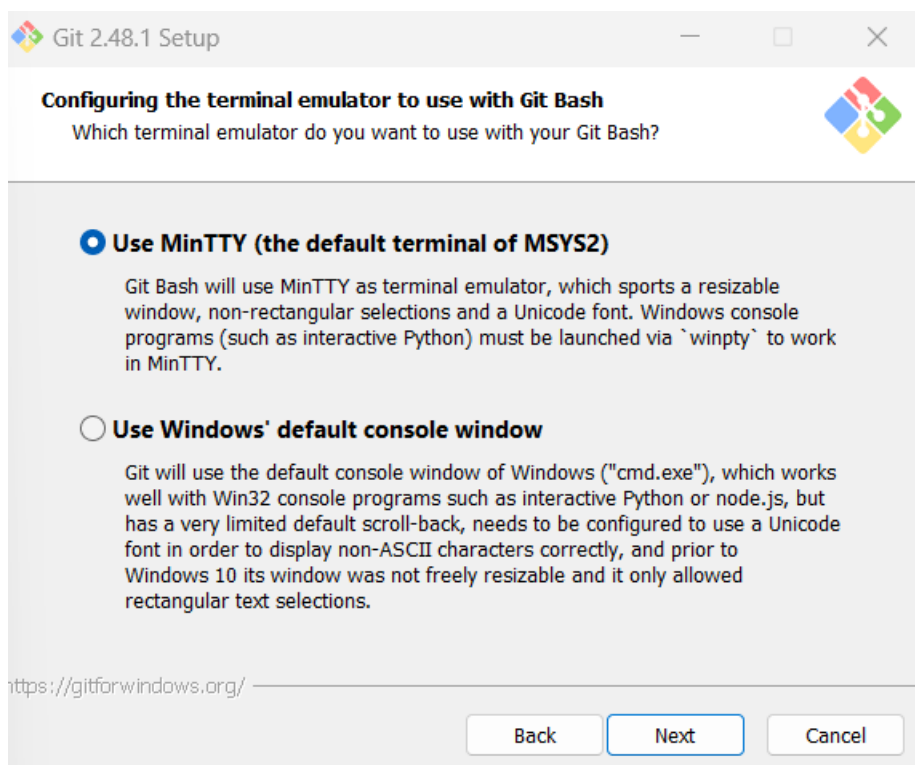
SSH 它是在互联网上加密通讯的一个工具，两台设备见通过互联网通讯走公共管道，不加密会被窃听，SSH 是加密工具中的一种。



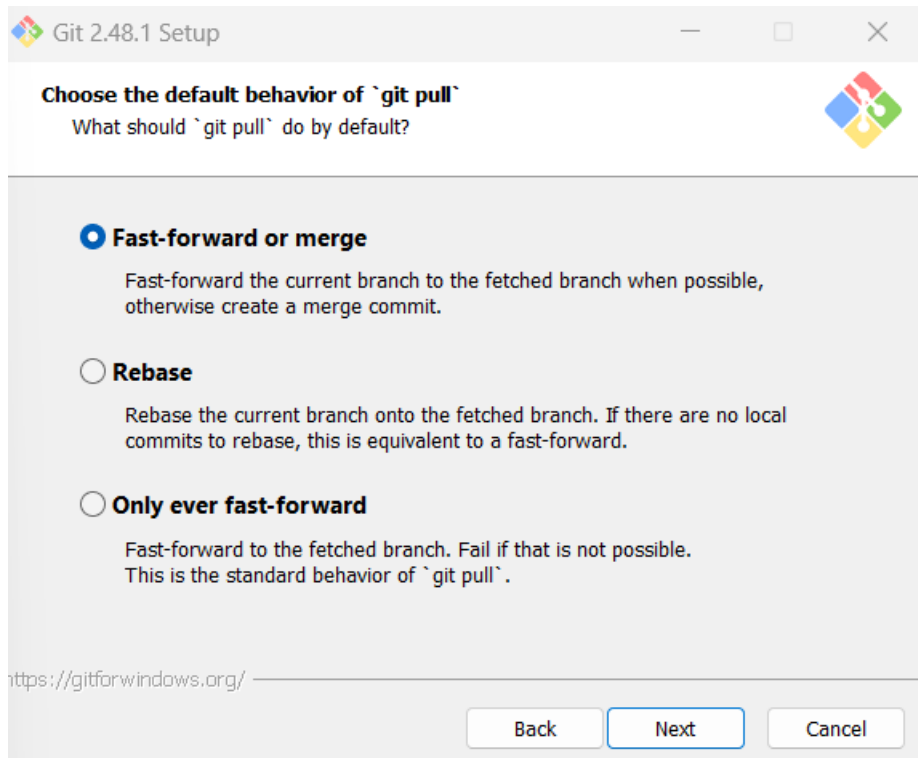
SSL 是网页在加密通讯时的一个协议



Checkout windows-style, 在从代码仓库里面把代码迁出时, 如果是 windows 电脑, 那么取出的时候, 就是按照 windows 的换行, windows 的换行符是 CRLF。这样在 windows 的软件里面就能看到正常的换行, 但是往里面去提交代码时, 也就是 commit 代码, 往代码仓库里面去提交代码的时候, 它会自动的给你换成 unix 的换行。

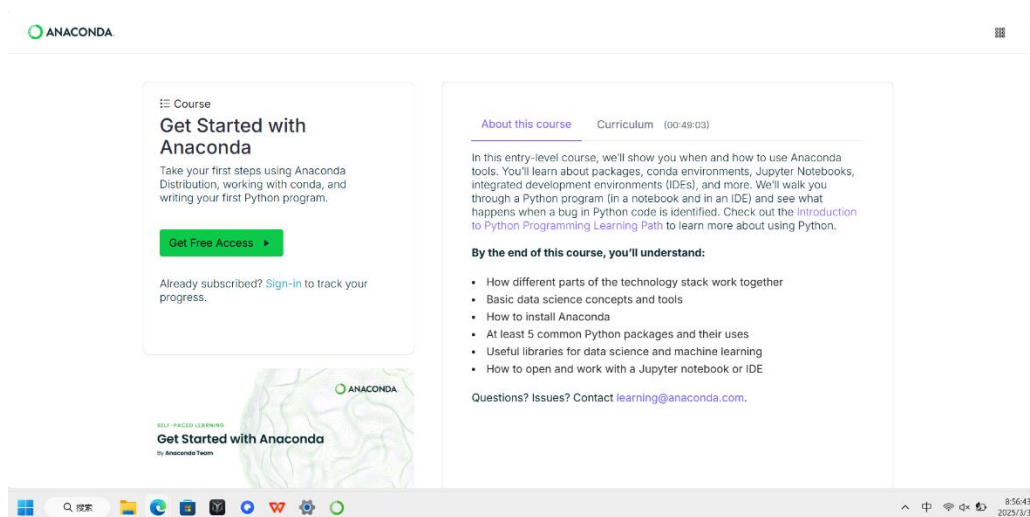


配置终端模拟器: MinTTY 是在 windows 下面模拟苹果或乌班图的 unix 终端的一个模拟品, 它是一个终端模拟器, 它虽然能在 windows 上面运行, 但是它的行为看起来跟在乌班图或苹果里一样, 能符合各种跨平台标准。



将代码从平台拉取到本地时的合并方式，选择 rebase 方式后在往平台推代码时会破坏别人的代码。

任务三：安装 [Anaconda](#) (Python 解释器)



命令：

```
which conda
echo $PATH
python
    quit()
which python
ipython
```

安装了 miniconda，通过命令“conda install ipython”在其中安装了 ipython，使用 ipython 可以显示语法高亮。

ymq@LAPTOP-J3NTEO80 MINGW64 ~/repo

~代表用户主目录，如:/c/Users/ymq/

命令 (git bash):

Pwd #显示当前工作目录的完整路径

mkdir repo #新建一个目录 repo

cd repo #在 Git Bash 中，cd repo 是一个命令，用于更改当前工作目录为名为 repo 的子目录。具体来说：cd 是 "change directory" 的缩写，是用于切换目录的命令。

ls #查看当前目录

ls -a #列出当前目录下的所有文件和文件夹，包括隐藏的

code #打开 vs code

code . #用 VS Code 打开当前目录，其中 "." 代表当前目录，

cat filename #显示文件内容

python xx.py #运行 xx.py 这一脚本

按上键会显示历史命令，连续按会依次显示所有历史命令

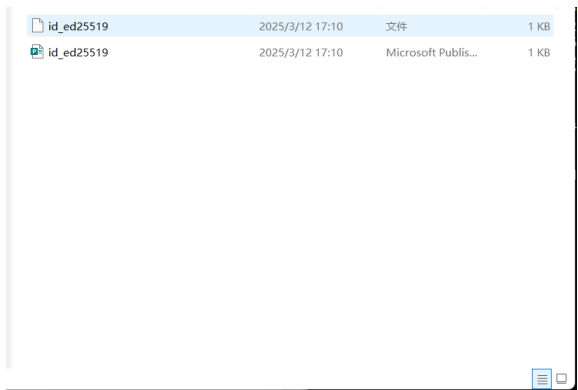
任务四：访问 [GitCode](https://gitcode.com) (代码托管平台) 注册用户，然后保持登录



任务五：生成自己设备的 SSH 密钥 (公钥-私钥对)，将公钥添加到自己的 GitCode 安全设置里

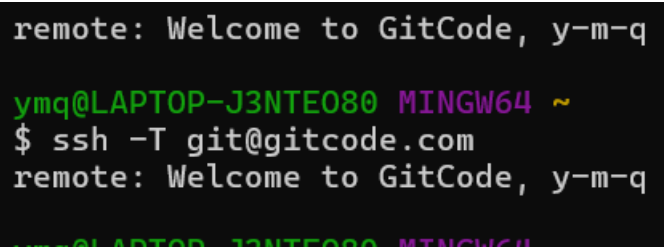
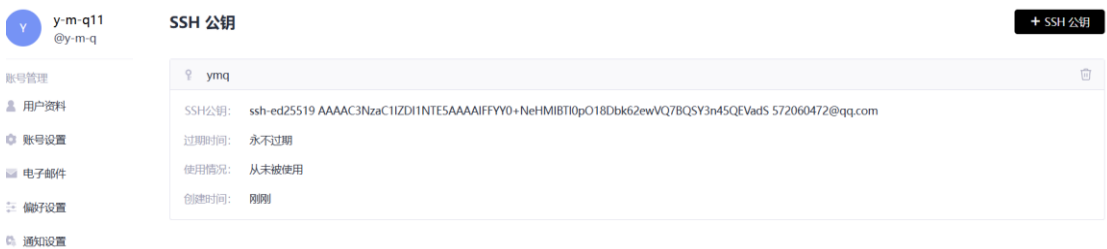
作用：在终端里配置好后，在终端打一些命令与 git code 的代码仓库，进行代码拉取同步时，不需要重复输入密码。

只要配置好了这个 SSH 这个密钥，那么你在终端里面去进行命令行的操作时，它跟 get code 这个平台进行代码的推送拉取等等这些操作的时候就不再需要输入密码了，而且是非常安全的。



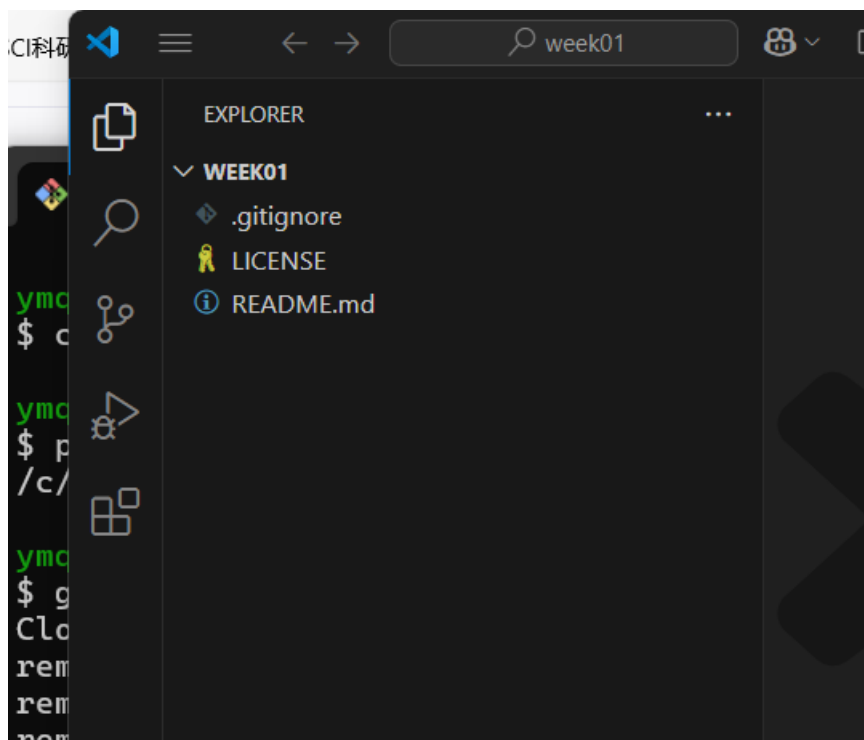
每台设备生成一个 ssh，第一个文件为私钥，第二个.pub 文件为公钥

添加公钥：



测试成功：

任务六：去第 01 周打卡 仓库阅读说明，完成学习报告的提交



运行: code week01/