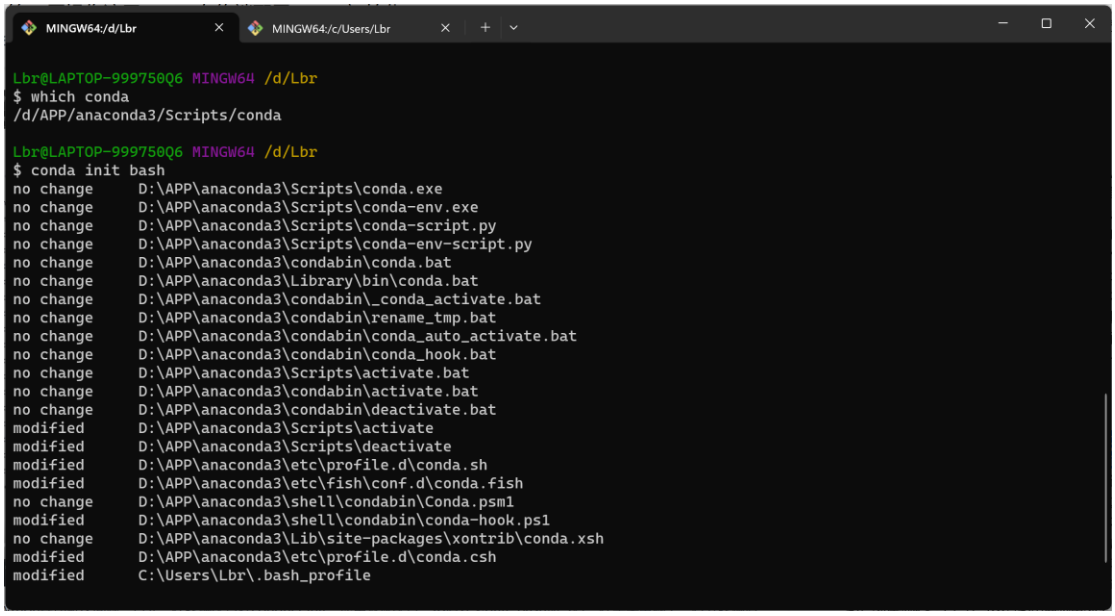
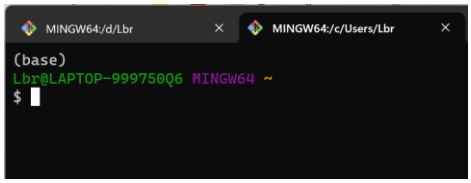
在终端配置 Conda 初始化

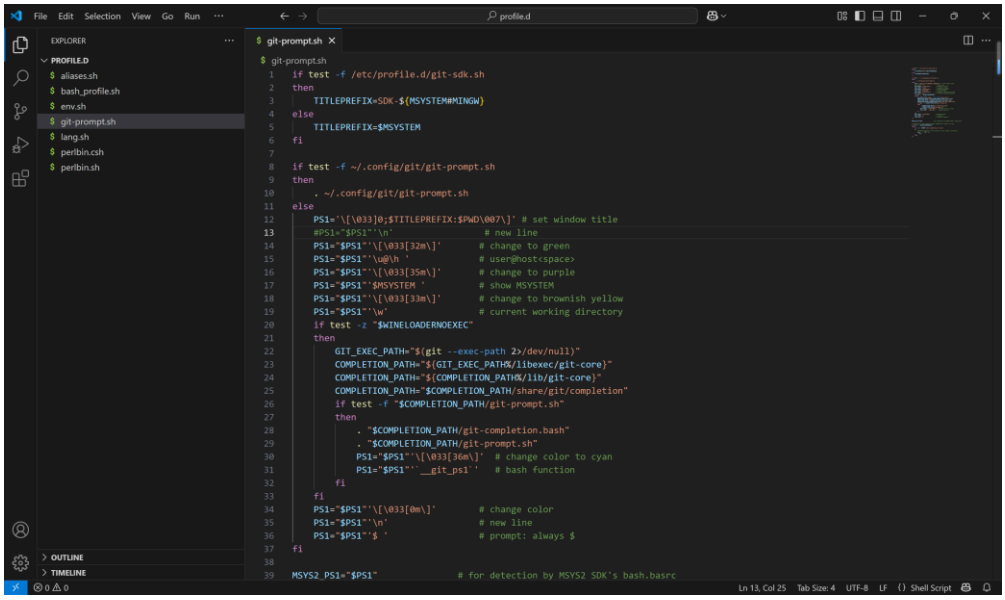
终端（Git Bash）配置好 Conda Init

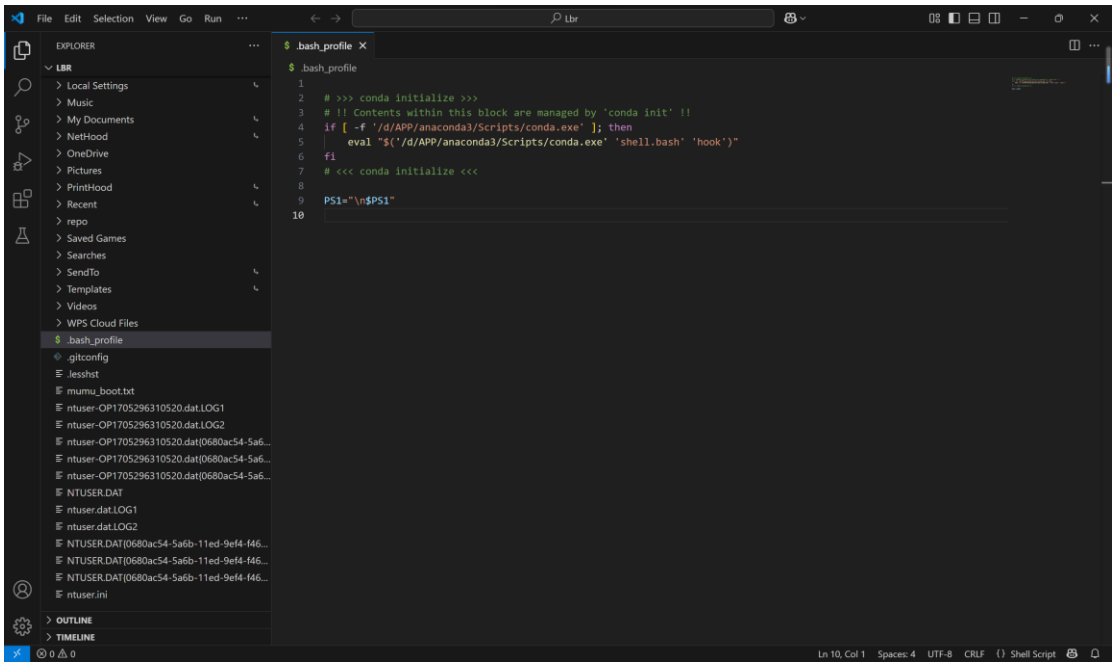
命令行输入 conda init bash

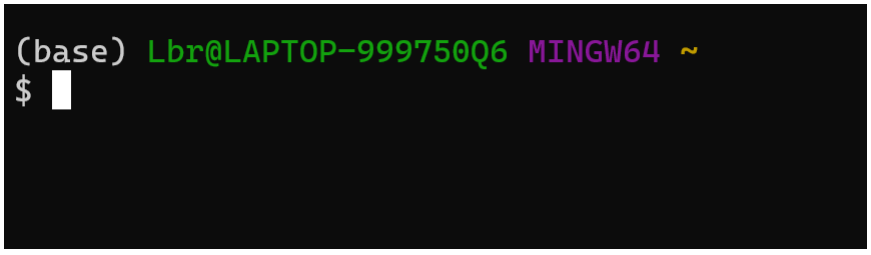




修改启动脚本


File Edit Selection View Go Run
profile.d
EXPLORER
PROFILE.D
aliases.sh
bash_profile.sh
env.sh
git-prompt.sh
lang.sh
perlbin.csh
perlbin.sh
git-prompt.sh
OUTLINE
TIMELINE
Ln 13, Col 25 Tab Size: 4 UTF-8 LF Shell Script

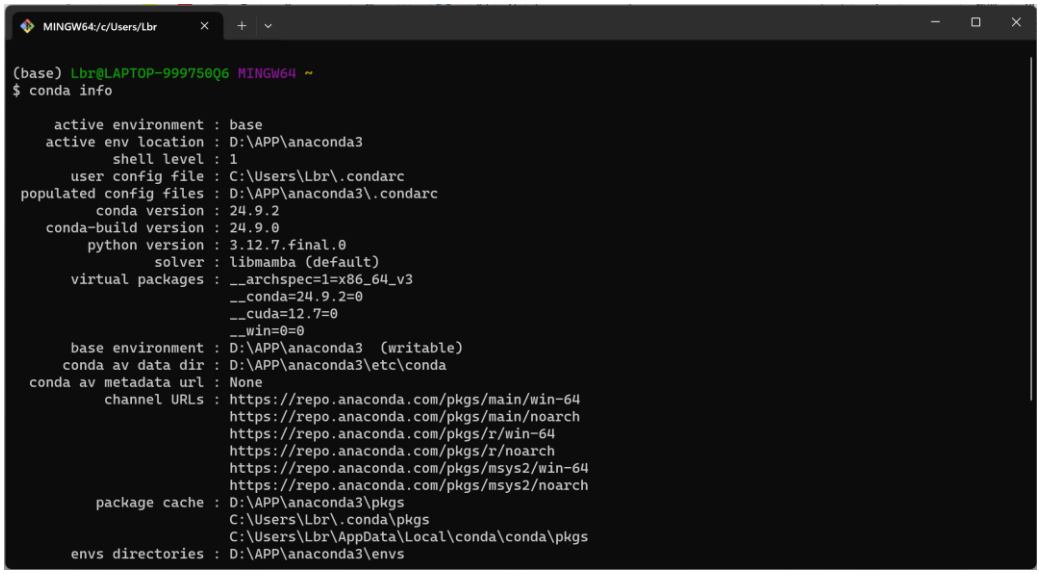

File Edit Selection View Go Run
Lbr
EXPLORER
LBR
Local Settings
Music
My Documents
NetHood
OneDrive
Pictures
PrintHood
Recent
repo
Saved Games
Searches
SendTo
Templates
Videos
WPS Cloud Files
.bash_profile
.gitconfig
.lesshst
mumu_boot.txt
ntuser-OP1705296310520.dat.LOG1
ntuser-OP1705296310520.dat.LOG2
NTUSER.DAT
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
.bash_profile
OUTLINE
TIMELINE
Ln 10, Col 1 Spaces: 4 UTF-8 CRLF Shell Script


(base) Lbr@LAPTOP-999750Q6 MINGW64 ~
$

初始化完成

创建 Conda 环境

conda info 命令查看本机 Conda 的配置信息



conda env list 命令查看已有的 Conda 环境的名称和路径

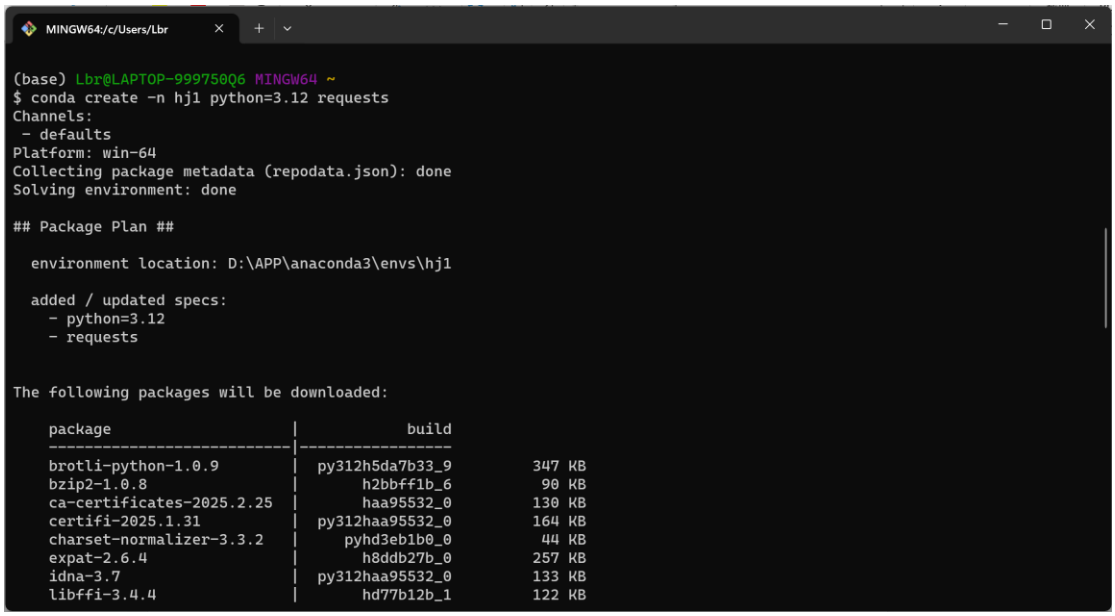
```
(base) Lbr@LAPTOP-999750Q6 MINGW64 ~
$ conda env list
# conda environments:
#
base                  *  D:\APP\anaconda3
```

Conda 环境:
独立的、隔离的工作空间，允许你在同一台计算机上管理不同项目所需的软件包和依赖项，避免版本冲突。
核心作用:
依赖隔离：每个环境有独立的 Python 解释器、第三方库（如 NumPy、Pandas）和工具，不同项目的依赖互不干扰。
版本控制：可为不同项目指定特定版本的软件（例如：项目 A 用 Python 3.8 + TensorFlow 2.4，项目 B 用 Python 3.10 + TensorFlow 2.12）。
跨平台支持：Windows/macOS/Linux 通用，适合团队协作和部署。

使用 conda create 命令创建两个 Conda 环境

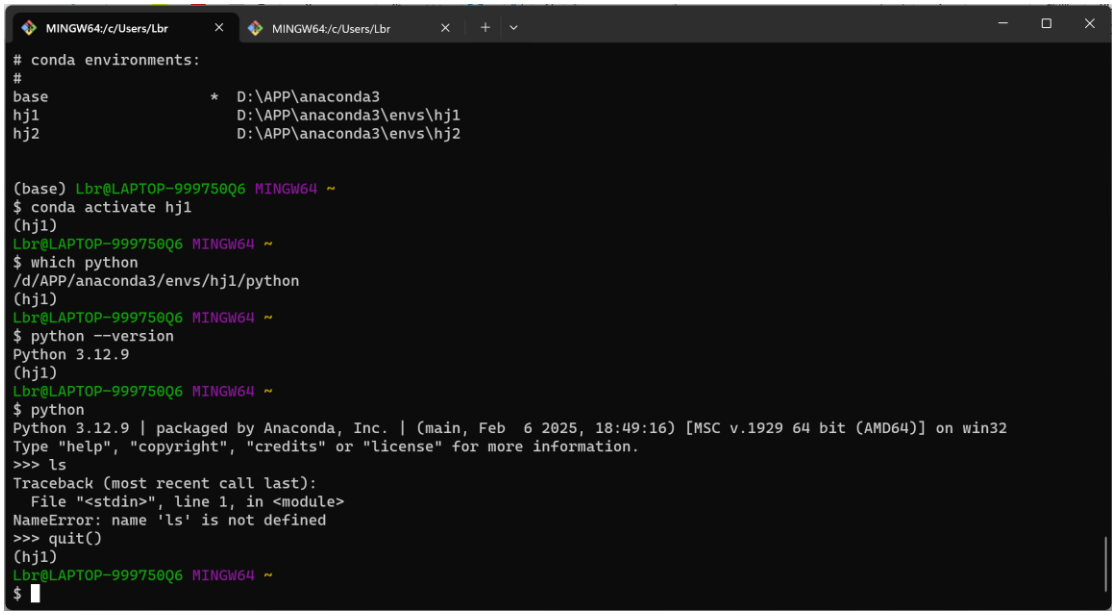
一个安装 Python 3.12 和 requests 软件包，另一个安装 Python 3.9、pandas 和 statsmodels 软件包

创建 conda 环境 hj1 和 hj2

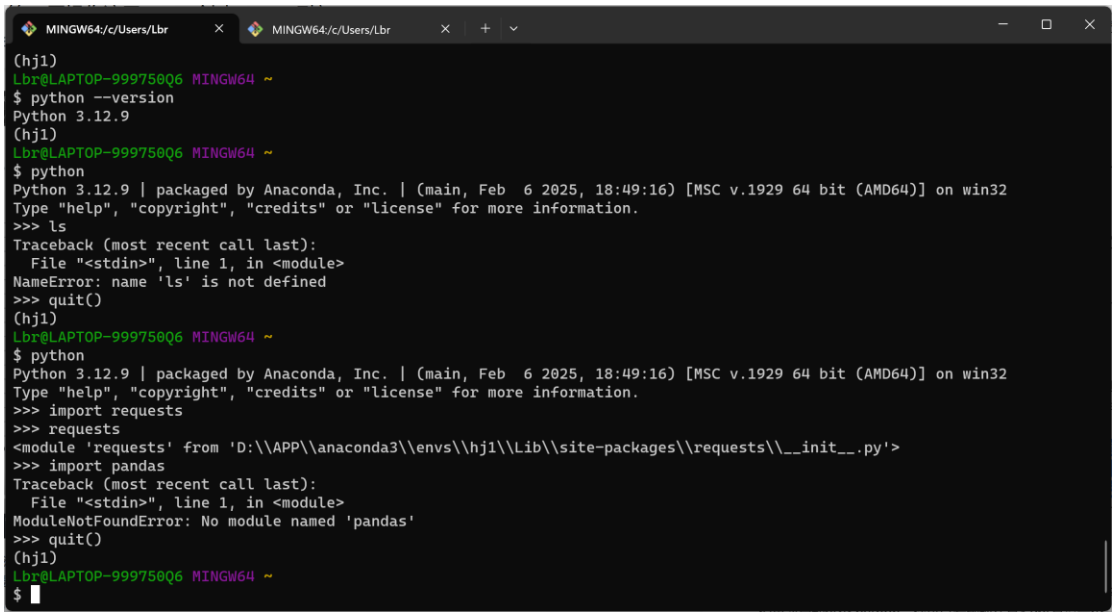


在终端里切换 Conda 环境

conda env list 查看有哪些环境
conda activate 环境名称 ： 激活某个环境
输入 python 进入 python，此时接受 python 语句，不识别 bash 语句，输入 quit()退出 python

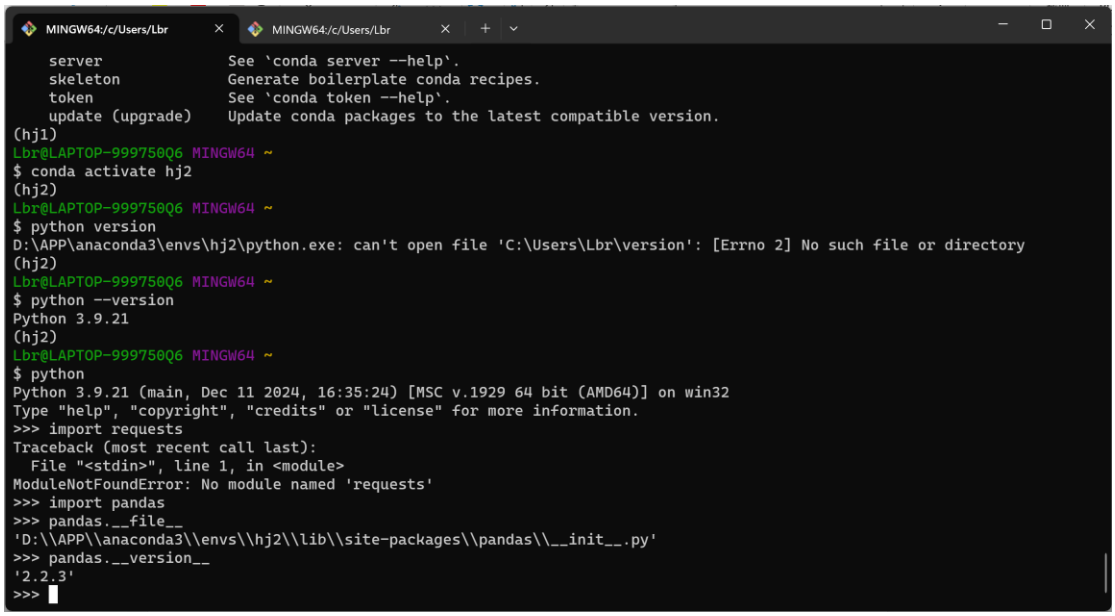


Python 中 import requests 为导入 requests 模块

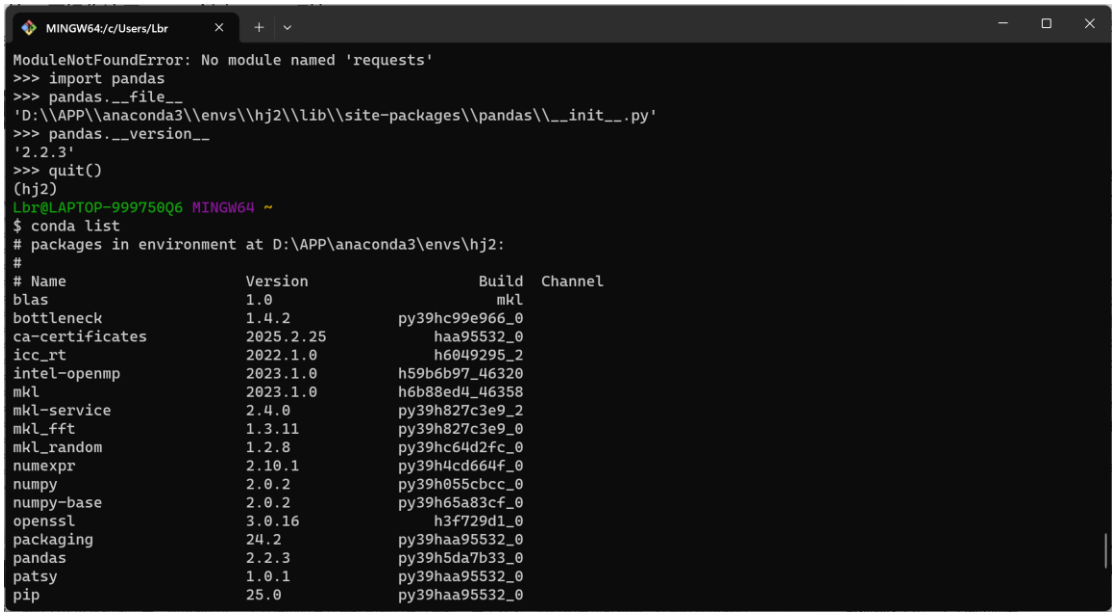


验证 Python 和软件包的版本

输入 python 查看 python 版本
Pandas.__version__为查看 pandas 版本

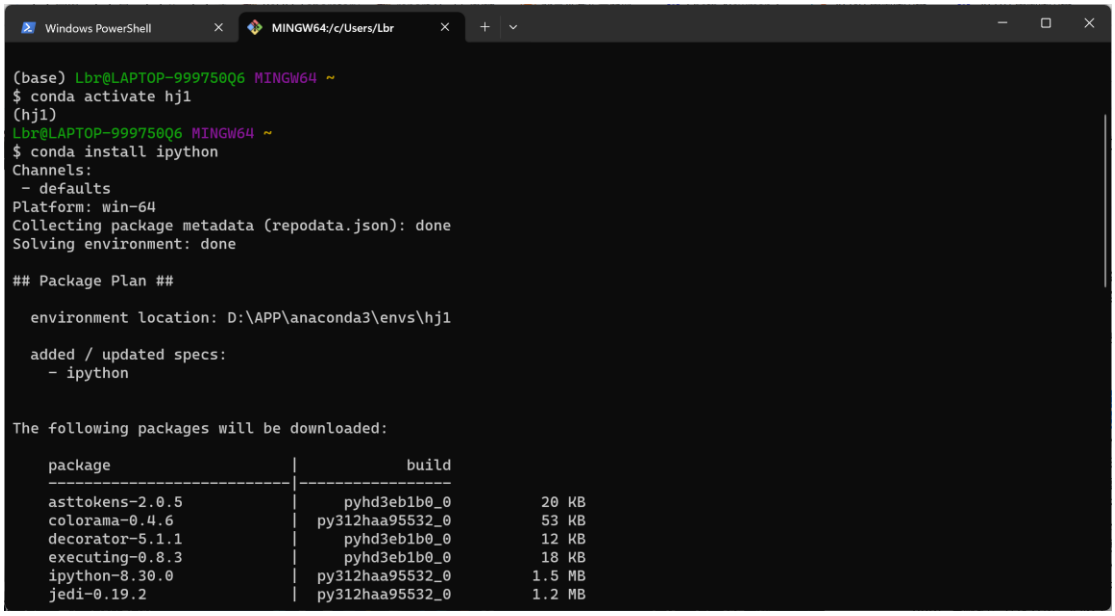


conda list 命令显示 Conda 环境里的软件包列表及其版本信息



Conda-Forge 清华镜像

命令：conda install 软件包的名称：往 Conda 环境里安装更多的软件包，并验证版本



验证版本信息

```
Lbr@LAPTOP-999750Q6 MINGW64 ~
$ conda list
# packages in environment at D:\APP\anaconda3\envs\hj1:
#
# Name                    Version                   Build  Channel
asttokens                 2.0.5              pyhd3eb1b0_0
brotli-python             1.0.9           py312h5da7b33_9
bzip2                     1.0.8               h2bbff1b_6
ca-certificates           2025.2.25           haa95532_0
certifi                   2025.1.31       py312haa95532_0
charset-normalizer        3.3.2              pyhd3eb1b0_0
colorama                  0.4.6           py312haa95532_0
decorator                 5.1.1              pyhd3eb1b0_0
executing                 0.8.3              pyhd3eb1b0_0
expat                     2.6.4               h8ddb27b_0
idna                      3.7             py312haa95532_0
ipython                   8.30.0          py312haa95532_0
```

配置 Anaconda 清华镜像

作用：保证连接和下载的稳定，加快 conda install 安装软件包的速度

可先执行 命令生成该.condarc 的文件之后再修改

```
Lbr@LAPTOP-999750Q6 MINGW64 ~
$ conda config --set show_channel_urls yes
(hj1)
Lbr@LAPTOP-999750Q6 MINGW64 ~
$
```

$ .bash_profile ! .condarc ●

! .condarc

```
channels:
  - defaults
show_channel_urls: true
default_channels:
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
custom_channels:
  conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

使用命令清除索引缓存：conda clean -i ， conda create -n myenv numpy

```
(hj1)
Lbr@LAPTOP-999750Q6 MINGW64 ~
$ conda clean -i
Will remove 1 index cache(s).
Proceed ([y]/n)? y

(hj1)
Lbr@LAPTOP-999750Q6 MINGW64 ~
```

```
Lbr@LAPTOP-999750Q6 MINGW64 ~
$ conda config --set channel_priority strict
(hj1)
Lbr@LAPTOP-999750Q6 MINGW64 ~
```

将 conda-forge 设置为默认 Channel，让 conda install 能够安装更多的软件包

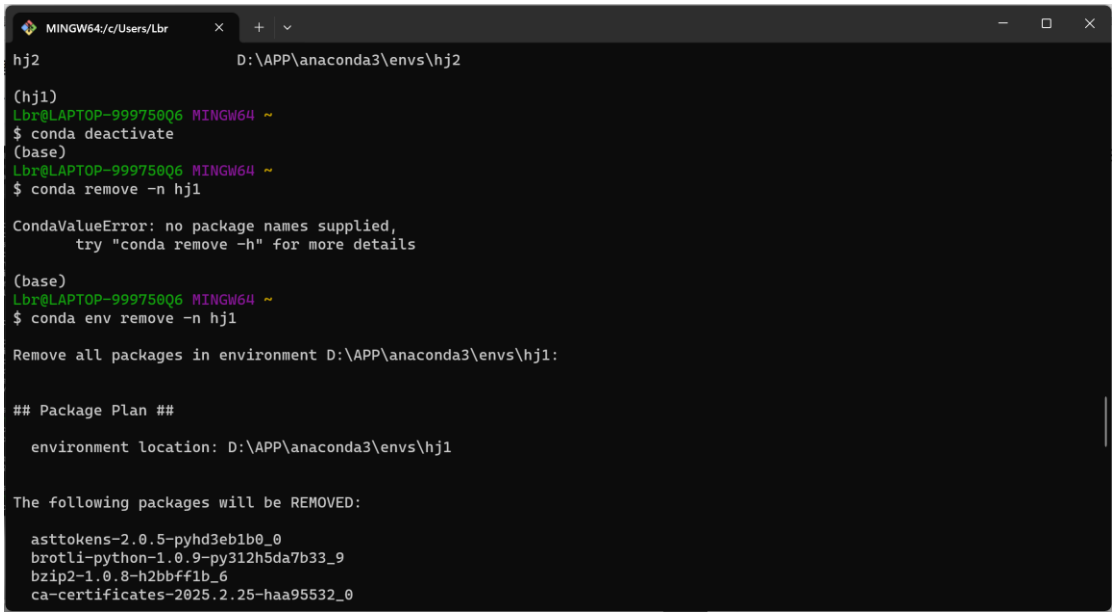
.condarc

```
channels:
  - conda-forge
  - defaults
show_channel_urls: true
default_channels:
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
custom_channels:
  conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
channel_priority: strict
```

注：软件包不可直接删除重新安装（直接删除可能导致某些依赖于他的软件包无法运行，强制删除整个环境的某个部分不可取）

正确做法：

1. 取消已经激活的环境（使用 conda deactivate 命令）
2. 删除整个环境（conda env remove -n 环境名称）



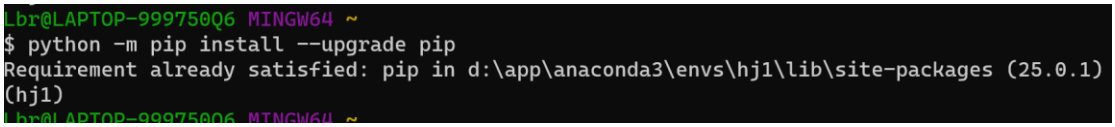
环境配置的导出与重建

使用 pip install 命令往 Conda 环境里安装 Python 软件包，并验证版本

作用：在 conda install 无法找到要安装的软件包时可使用

配置 PyPI 清华镜像，加快 pip install 安装软件包的速度

1. 升级 pip 到最新的版本后进行配置：



2. 将全局的这个索引的这个网址设置成清华的域名

```
Lbr@LAPTOP-999750Q6 MINGW64 ~
$ pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
Writing to C:\Users\Lbr\AppData\Roaming\pip\pip.ini
(hj1)
```

完成后即可在镜像网站下载软件包，而不是从国外网站下载，提升下载速度

安装软件包 tushare

```
MINGW64:/c/Users/Lbr
requests                  2.32.3          pyhd8ed1ab_1    conda-forge
setuptools                75.8.2          pyhff2d567_0    conda-forge
tbb                       2021.13.0         h62715c5_1    conda-forge
tk                        8.6.13            h5226925_1    conda-forge
tzdata                    2025b             h78e105d_0    conda-forge
ucrt                      10.0.22621.0      h57928b3_1    conda-forge
urllib3                   2.3.0           pyhd8ed1ab_0    conda-forge
vc                        14.3             h2b53caa_26    conda-forge
vc14_runtime              14.42.34438       hfd919c2_26    conda-forge
wheel                     0.45.1          pyhd8ed1ab_1    conda-forge
win_inet_pton             1.1.0           pyh7428d3b_8    conda-forge
zstandard                 0.23.0          py312h4389bb4_1    conda-forge
(hj1)
Lbr@LAPTOP-999750Q6 MINGW64 ~
$ pip install tushare
Looking in indexes: https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
Collecting tushare
  Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/38/33/0b4ed09bff3b69887c2545f1c587420c89685d36c377095e465cc000759b/tushare-1.4.21-py3-none-any.whl (142 kB)
Collecting pandas (from tushare)
  Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/29/d4/1244ab8edf173a10fd601f7e13b9566c1b525c4f365d6bee918e68381889/pandas-2.2.3-cp312-cp312-win_amd64.whl (11.5 MB)
     ---------------------------------------- 11.5/11.5 MB 7.3 MB/s eta 0:00:00
Requirement already satisfied: requests in d:\app\anaconda3\envs\hj1\lib\site-packages (from tushare) (2.32.3)
Collecting lxml (from tushare)
  Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/2b/9c/8abe21585d20ef70ad9cec7562da4332b764ed69ec29b7389d23dfabcea0/lxml-5.3.1-cp312-cp312-win_amd64.whl (3.8 MB)
     ---------------------------------------- 3.8/3.8 MB 5.6 MB/s eta 0:00:00
Collecting simplejson (from tushare)
  Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/bc/5d/4e243e937fa3560107c69f6f7c2eed8589163f5ed1432
```

进入 python：直接在命令行中输入 pyhton
在 python 中导入 tushare：import tushare
查看 tushare 版本：tushare.__version__
查看 tushare 文件路径：tushare.__file__
退出 python：quit()

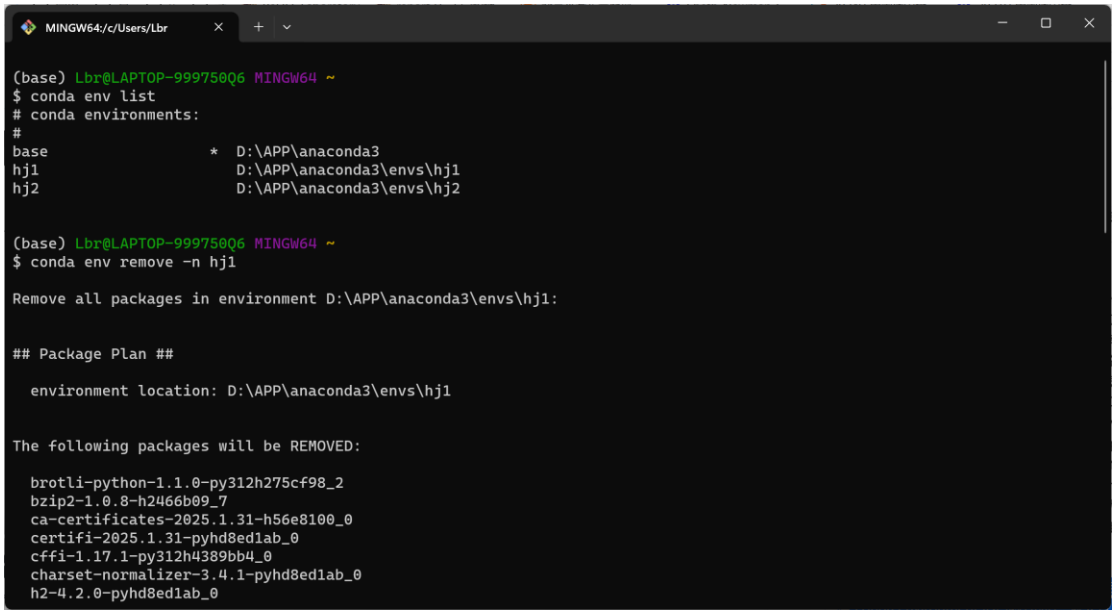
```
Lbr@LAPTOP-999750Q6 MINGW64 ~
$ python
Python 3.12.9 | packaged by conda-forge | (main, Mar  4 2025, 22:37:18) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tushare
>>> tushare.__version__
'1.4.21'
>>> tushare.__file__
'D:\\APP\\anaconda3\\envs\\hj1\\Lib\\site-packages\\tushare\\__init__.py'
>>>
```

导出 environment.yml Conda 环境配置文件

环境中有一些的软件包和他们的版本信息，可以导出
作用：导出后的环境配置可以在缺乏适合的环境时可直接生成和原来一样的环境
导出命令：conda env export -f 文件名.yml

```
Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo
$ conda env export -f huanjing1.yml
(hj1)
```

删除 Conda 环境



用 environment.yml 配置文件重建 Conda 环境

将环境文件移动到 repo 中

```
(base) Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo
$ mkdir hj1

(base) Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo
$ cd ./hj1/

(base) Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo/hj1
$ mv /d/Lbr/repo/huanjing1.yml .
```

用环境文件在原环境已经删除的情况下重建删除的环境

使用命令 conda env create -f /路径/环境文件.yml

```
(base) Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo/hj1
$ conda env list
# conda environments:
#
base                  *  D:\APP\anaconda3
hj1                      D:\APP\anaconda3\envs\hj1
hj2                      D:\APP\anaconda3\envs\hj2
```

理解 Conda 与 Python 的关系

多语言支持：conda 是跨语言的包管理系统，不仅可以管理 python，还可以管理其他编程语言（如 R,C++等），但是 python 主要专注于 python 语言包的管理，其他语言的包支持能力

有限
版本管理与依赖处理：可以创建和管理不同版本 python 的环境，同时分析依赖项，确保依赖项的版本兼容
环境隔离：可以做到不同 python 版本在不同环境中的隔离，但是 python 仅仅是在同一版本下的隔离。
跨平台兼容性：在不同系统上面都可以使用
环境共享和复制：可将环境导出为文件并可通过文件复制相同的环境

理解 Conda-Forge 与 Conda 的关系

conda 是一个包管理和环境管理的一个系统，提供了一系列的命令，而 conda-forge 是基于康大的一个社区驱动的软件包仓库
功能：conda 是核心组件，是一个创建，管理虚拟环境的系统，也可以在环境中安装，更新，卸载软件包，切换环境，conda-forge 是一个软件包仓库，里面有各类软件包
优势互补：conda-forge 丰富了 conda 软件包的资源，conda 为 conda-forge 提供了分发渠道

理解 Python 解释器

Python 解释器是一种能够读取并执行 Python 代码的程序。它将你编写的 Python 代码转化为计算机可以理解和执行的机器语言指令。

Conda-forge 和 PyPI 的区别

Conda-forge 不仅包括 python 包，也包含其他语言的软件包，以及一些依赖项（超出了 python 以外的东西，但是又需要用 python 去调用的）
PyPI：python 官方的软件包仓库，主要聚焦于 python 包，python 包的种类更丰富全面

理解第三方软件包

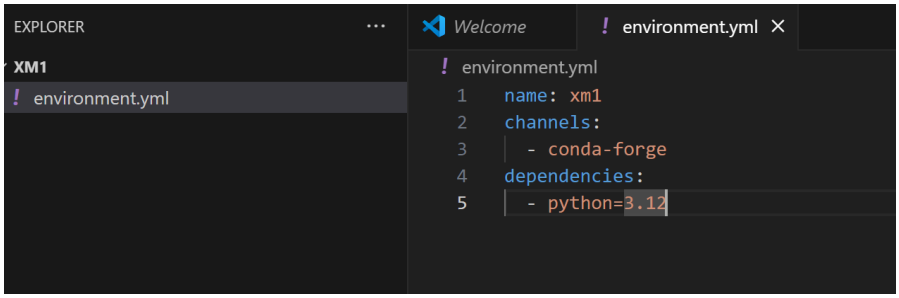
第三方软件包（Third-party Packages）是由 Python 社区开发者或组织（非 Python 官方团队）创建的、可复用的代码模块，用于扩展 Python 的功能，帮助开发者快速实现特定任务（如数据分析、Web 开发、机器学习等）。
优势:
快速开发：避免重复造轮子，直接使用成熟解决方案。
功能强大：覆盖几乎所有领域（如 AI、图像处理、数据库连接）。
社区支持：开源包通常有文档、示例和社区答疑（如 GitHub Issues）

编写和运行 Python 程序

创建项目目录

创建本地项目文件夹（最好为小写字母和数字构成，尽量不用大写字母，空格，特殊符号，中文，容易产生编解码的兼容问题）
命令：mkdir 文件夹名称

在文件夹中创建 environment.yml 文件（使用 VS code）




```
(base) Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo/xm1
$ cat environment.yml
name: xm1
channels:
  - conda-forge
dependencies:
  - python=3.12
```

环境名称是 xm1，环境优先的 channel 是 conda-forge，环境中安装 python3.12

激活环境 命令：conda activate 环境名称

```
(base) Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo/xm1
$ conda env list
# conda environments:
#
base                  *  D:\APP\anaconda3
hj1                      D:\APP\anaconda3\envs\hj1
hj2                      D:\APP\anaconda3\envs\hj2
xm1                      D:\APP\anaconda3\envs\xm1


(base) Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo/xm1
$ conda activate xm1
(xm1)
Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo/xm1
```

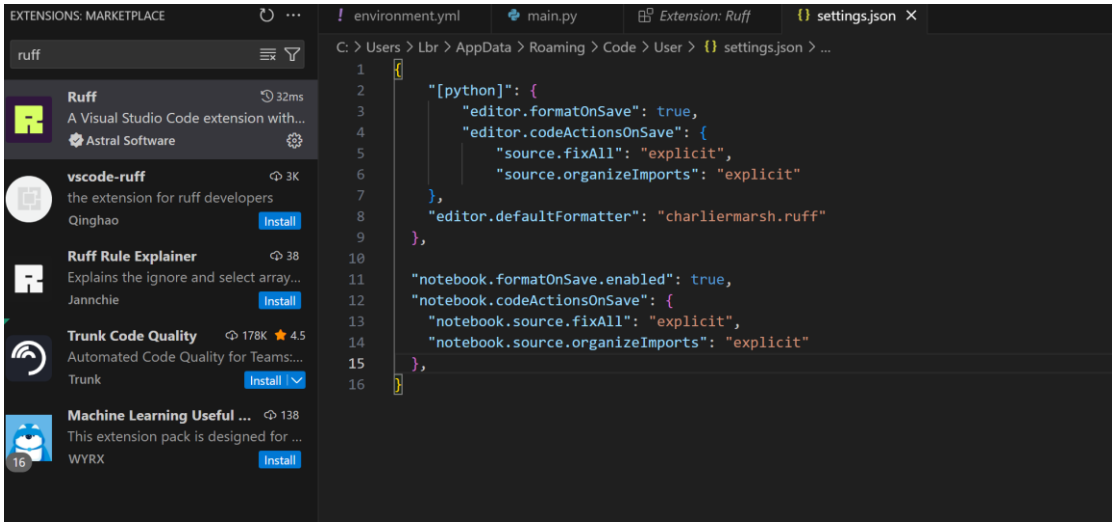
此时环境已准备完成

创建简易 python 程序（VS code 创建）

在 VS Code 文本编辑器里安装一些支持 Python 开发的常用扩展

作用：在 vs code 里边去编写 python 的代码时会增加很多专门针对 python 的功能

安装和设置 ruff



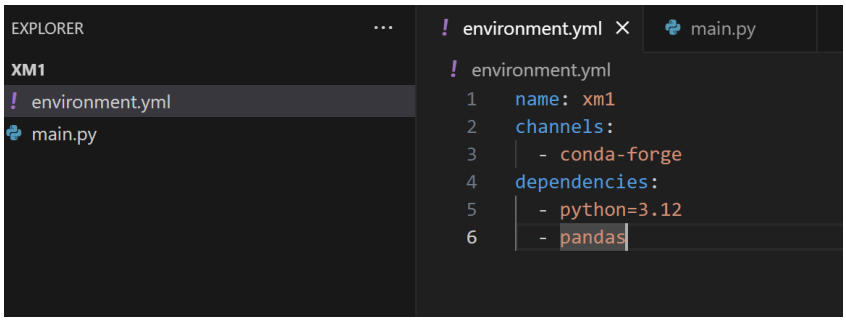
选择 python 解释器

CTRL+shift+p 输入 python interpre，从文件夹找到当前环境的文件夹，选择里面的 python 的 exe 文件（报错，未完成配置）

在环境中运行该脚本

```
(base) Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo/xm1
$ conda activate xm1
(xm1)
Lbr@LAPTOP-999750Q6 MINGW64 /d/Lbr/repo/xm1
$ python main.py
Hello, conda!
(xm1)
```

问题：在脚本中导入环境中不存在的软件包 pandas 可以从环境文件中加入软件包 pandas 解决无法找到软件包 pandas 的问题



环境文件更改完成后在终端中进行环境更新 命令：conda env update

把网址所指向的 CSV 的文件下载在当前文件夹中

在命令行中粘贴 curl -O https://edg.epa.gov/EPADataCommons/public/OA/EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv

-o：输出成文件