

第三周学习报告

任务目标

我们解决任何实际问题，基本上都是以 **项目** 为工作方式。所以我们首先应该了解如何 **创建** Python 项目，然后是如何 **使用** 她，最后是如何 **逐步地改进** 她。

1. 在自己的终端 (比如 Git Bash、Zsh 等) 配置好 Conda Init, 使得启动终端后, 在提示符 (比如 \$、%) 前能够看到 (base)
2. 使用 `conda info` 命令查看本机 Conda 的配置信息
3. 使用 `conda env list` 命令查看已有的 Conda 环境的名称和路径, 理解 **Conda 环境** 的概念
4. 使用 `conda create` 命令创建两个 Conda 环境, 一个里面安装 Python 3.12 和 requests 软件包, 另一个里面安装 Python 3.9、pandas 和 statsmodels 软件包, 能够在终端里切换 Conda 环境, 验证 Python 和软件包的版本
5. 使用 `conda list` 命令显示 Conda 环境里的软件包列表及其版本信息
6. 使用 `conda install` 命令往 Conda 环境里安装更多的软件包, 并验证版本
7. 根据 [文档](#), 配置 Anaconda 清华镜像, 加快 `conda install` 安装软件包的速度, 将 `conda-forge` 设置为默认 Channel, 让 `conda install` 能够安装更多的软件包
8. 使用 `pip install` 命令往 Conda 环境里安装 Python 软件包, 并验证版本
9. 根据 [文档](#) 配置 PyPI 清华镜像, 加快 `pip install` 安装软件包的速度
10. 能够导出 `environment.yml` Conda 环境配置文件, 能够删除 Conda 环境, 能够用 `environment.yml` 配置文件重建 Conda 环境
11. 理解 Conda 与 Python 的关系, 理解 Conda-Forge 与 Conda 的关系, 理解 Python 解释器、第三方软件包、PyPI 软件仓库、以及程序/软件包的路径问题
12. 按照 [教程](#) 创建项目目录, 在 VS Code 文本编辑器里安装一些支持 Python 开发的常用扩展, 编写 `main.py` 脚本, 创建该项目专用的 Conda 环境, 在终端里激活该环境并成功运行该脚本
13. 去 [第03周打卡](#) 仓库, 按照「操作步骤」说明, 用 Fork-PR 方法提交本周学习报告

1. 在自己的终端 (比如 Git Bash、Zsh 等) 配置好 Conda Init, 使得启动终端后, 在提示符 (比如 \$、%) 前能够看到 (base)

Control +L 清空屏幕

2. 使用 `conda info` 命令查看本机 Conda 的配置信息

```
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如
$ conda info

active environment : base
active env location : D:\anaconda3
  shell level      : 1
user config file   : C:\Users\李意如\.condarc
populated config files : D:\anaconda3\.condarc
conda version      : 24.9.2
conda-build version : 24.9.0
python version     : 3.12.7.final.0
solver            : libmamba (default)
virtual packages   : __archspec=1=x86_64_v3
                   __conda=24.9.2=0
                   __win=0
base environment   : D:\anaconda3 (writable)
conda av data dir  : D:\anaconda3\etc\conda
conda av metadata url : None
channel URLs       : https://repo.anaconda.com/plugs/main/win-64
                   https://repo.anaconda.com/plugs/main/noarch
                   https://repo.anaconda.com/plugs/t/win-64
                   https://repo.anaconda.com/plugs/t/noarch
                   https://repo.anaconda.com/plugs/nsys2/win-64
                   https://repo.anaconda.com/plugs/nsys2/noarch
package cache      : D:\anaconda3\plugs
                   C:\Users\李意如\.conda\plugs
                   C:\Users\李意如\AppData\Local\conda\conda\plugs
envs directories   : D:\anaconda3\envs
                   C:\Users\李意如\.conda\envs
                   C:\Users\李意如\AppData\Local\conda\conda\envs
platform          : win-64
user-agent         : conda/24.9.2 requests/2.32.3 CPython/3.12.7 Windows/11 Windows/10.0.22631 solver/libmamba conda-libmamba-solver/24.9.0 libmambapy/1.5.8 aau/0.4.4 c/. s/. e/.
administrator      : False
netrc file         : None
offline mode       : False
```

3. 使用 `conda env list` 命令查看已有的 Conda 环境的名称和路径, 理解 **Conda 环境** 的概念

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ conda env list
# conda environments:
#
base * D:\anaconda3
```

名称: base 具体路径: D: \anaconda3

作用

- **依赖管理**：在不同的项目中，可能会使用到不同版本的 Python 以及各种不同的第三方库。通过 Conda 环境，可以为每个项目创建一个独立的环境，在这个环境中安装项目所需的特定版本的 Python 和相关包，而不会影响其他项目的运行环境。
- **环境隔离**：避免不同项目之间因依赖冲突而导致的问题。例如，项目 A 依赖于某个库的 1.0 版本，而项目 B 需要该库的 2.0 版本，在不同的 Conda 环境中可以分别安装这两个版本，使两个项目都能正常运行，而不会出现因版本冲突导致的错误。

4. 使用 `conda create` 命令创建两个 Conda 环境，一个里面安装 Python 3.12 和 `requests` 软件包，另一个里面安装 Python 3.9、`pandas` 和 `statsmodels` 软件包，能够在终端里切换 Conda 环境，验证 Python 和软件包的版本

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~  
$ conda create -n prj1 python=3.12 requests  
Channels:  
- defaults  
Platform: win-64  
Collecting package metadata (repodata.json): done  
Solving environment: done  
  
## Package Plan ##  
  
environment location: D:\anaconda3\envs\prj1  
added / updated specs:
```

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~  
$ conda create -n prj2 python=3.9 pandas statsmodels  
Channels:  
- defaults  
Platform: win-64  
Collecting package metadata (repodata.json): done  
Solving environment: done
```

看一下有哪些环境

```
(base)  
李意如@LAPTOP-9J8HOMDD MINGW64 ~  
$ conda env list  
# conda environments:  
#  
base * D:\anaconda3  
prj1 D:\anaconda3\envs\prj1  
prj2 D:\anaconda3\envs\prj2
```

更改，三个之间都用 `activate` 进行修改

```
(base)  
李意如@LAPTOP-9J8HOMDD MINGW64 ~  
$ conda activate prj1  
(prj1)  
李意如@LAPTOP-9J8HOMDD MINGW64 ~  
$
```

查看版本

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ which python
/d/anaconda3/envs/prj1/python
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ python --version
Python 3.12.9
```

python

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ python
Python 3.12.9 | packaged by Anaconda, Inc. | (main, Feb 6 2025, 18:49:16) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

quit()退掉

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ python
Python 3.12.9 | packaged by Anaconda, Inc. | (main, Feb 6 2025, 18:49:16) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()
```

requests 是一个软件包，pandas、statsmodels 的打开方式同理，

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ python
Python 3.12.9 | packaged by Anaconda, Inc. | (main, Feb 6 2025, 18:49:16) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import requests
>>> requests
<module 'requests' from 'D:\\anaconda3\\envs\\prj1\\Lib\\site-packages\\requests\\__init__.py'>
>>> requests.__file__
'D:\\anaconda3\\envs\\prj1\\Lib\\site-packages\\requests\\__init__.py'
>>>
```

5. 使用 conda list 命令显示 Conda 环境里的软件包列表及其版本信息

(非常重要，可以知道具体有哪些软件包)

```
(prj2)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ conda list
# packages in environment at D:\anaconda3\envs\prj2:
#
# Name                   Version           Build    Channel
blas                     1.0               mkl
bottleneck               1.4.2            py39hc99e966_0
ca-certificates          2025.2.25         haa95532_0
icc_rt                   2022.1.0          h6049295_2
intel-openmp             2023.1.0          h59b6b97_46320
mkl                      2023.1.0          h6b88ed4_46358
mkl-service              2.4.0            py39h827c3e9_2
mkl_fft                  1.3.11           py39h827c3e9_0
mkl_random               1.2.8            py39hc64d2fc_0
numexpr                  2.10.1           py39h4cd664f_0
numpy                    2.0.2            py39h655cbbc_0
numpy-base               2.0.2            py39h65a83cf_0
openssl                  3.0.16           h3f729d1_0
packaging                24.2             nv39haa95532_0
pandas                   2.2.3            py39h5da7b33_0
patsy                    1.0.1            py39haa95532_0
pip                      25.0             py39haa95532_0
pybind11-abi             5                hd3eb1b0_0
python                   3.9.21           h8205438_1
python-dateutil          2.9.0post0       py39haa95532_2
python-tzdata            2023.3           pyhd3eb1b0_0
pytz                     2024.1           py39haa95532_0
scipy                    1.13.1           py39h8640f81_1
setuptools               72.1.0           py39haa95532_0
six                      1.16.0           pyhd3eb1b0_1
sqlite                   3.45.3           h2bbff1b_0
statsmodels              0.14.4           py39h827c3e9_0
tzdata                   2025a            h04d1e81_0
vc                       14.42            haa95532_4
vs2015_runtime           14.42.34433      he0abc0d_4
wheel                    0.45.1           py39haa95532_0
```

剩下的有些是软件包的依赖项

6. 使用 `conda install` 命令往 Conda 环境里安装更多的软件包，并验证版本安装

```
(prj2)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ conda install ipython
Retrieving notices: ...working... done
Channels:
- defaults
```

验证版本 `conda list`

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ conda list
# packages in environment at D:\anaconda3\envs\prj2:
#
# Name                    Version           Build
asttokens                 2.0.5            pyhd3eb1b0_0
backcall                  0.2.0            pyhd3eb1b0_0
blas                      1.0              mkl
bottleneck                1.4.2            py39hc99e966_0
ca-certificates           2025.2.25        haa95532_0
colorama                  0.4.6            py39haa95532_0
decorator                 5.1.1            pyhd3eb1b0_0
exceptiongroup            1.2.0            py39haa95532_0
executing                 0.8.3            pyhd3eb1b0_0
icc_rt                    2022.1.0          h6049295_2
intel-openmp              2023.1.0          h59b6b97_46320
ipython                   8.15.0            py39haa95532_0
jedi                      0.19.2           py39haa95532_0
matplotlib-inline         0.1.6            py39haa95532_0
mkl                       2023.1.0          h6b88ed4_46358
```

7. 根据 [文档](#)，配置 Anaconda 清华镜像，加快 `conda install` 安装软件包的速度，将 `conda-forge` 设置为默认 Channel，让 `conda install` 能够安装更多的软件包

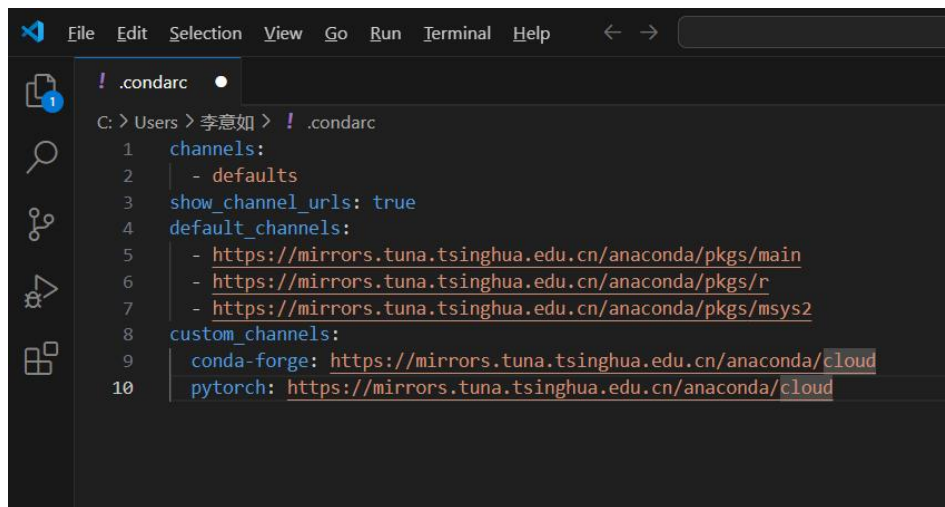
先创建 `.condarc` 文件夹，`conda config --set show_channel_urls yes`

```
(prj2)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ conda config --set show_channel_urls yes
```

用 `vscode` 打开文件夹并进行复制粘贴

channels:

- defaultsshow_channel_urls: truedefault_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/msys2custom_channels:
- conda-forge: <https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud>
- pytorch: <https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud>



```
! .condarc
C: > Users > 李意如 > ! .condarc
1 channels:
2 | - defaults
3 show_channel_urls: true
4 default_channels:
5 | - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main
6 | - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/r
7 | - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/msys2
8 custom_channels:
9 | conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
10 | pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

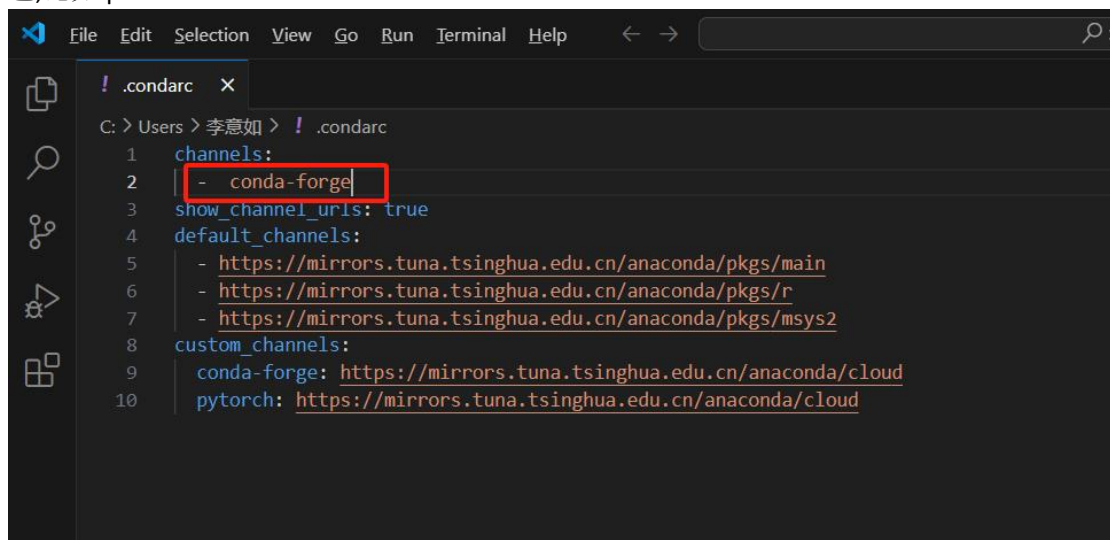
使用下列命令清除索引缓存，并安装常用包测试一下。

```
conda clean -i
conda create -n myenv numpy
```

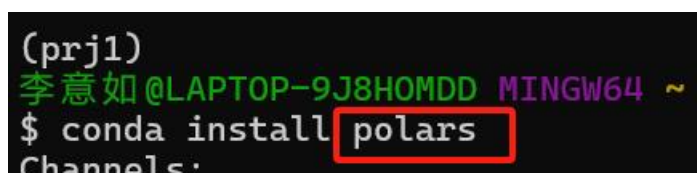
conda clean -i

conda create -n myenv numpy

怎么设置 conda-forge 为默认 channels。可以安装更多的软件包，很前沿很高级很酷的软件包,比如 polars



```
! .condarc X
C: > Users > 李意如 > ! .condarc
1 channels:
2 | - conda-forge
3 show_channel_urls: true
4 default_channels:
5 | - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main
6 | - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/r
7 | - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/msys2
8 custom_channels:
9 | conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
10 | pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```



```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ conda install polars
Channels:
```

检验

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ which python
d/anaconda3/envs/prj1/python
(prj1)
李意如@LAPTOP-9J8HOMDD 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 polars
>>> polars
<module 'polars' from 'D:\anaconda3\envs\prj1\Lib\site-packages\polars\__init__.py'>
```

怎么删掉 整个环境都删掉 `conda env remove -n prj1`。删掉以后继续重新下

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ conda activate base
(base)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ conda env remove -n prj1
```

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

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ pip install tushare
```

9. 根据 文档 配置 PyPI 清华镜像，加快 `pip install` 安装软件包的速度

升级 pip 到最新的版本后进行配置：

```
python -m pip install --upgrade pip
pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
```

`python -m pip install --upgrade pip`

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ python -m pip install --upgrade pip
```

`pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple`

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
Writing to C:\Users\李意如\AppData\Roaming\pip\pip.ini
```

再重新安装测试

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ pip install tushare
Looking in indexes: https://mirrors.t
```

```
(prj1)
李意如@LAPTOP-9J8HOMDD 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__
'1.4.19'
>>> tushare.__file__
'D:\\anaconda3\\envs\\prj1\\Lib\\site-packages\\tushare\\__init__.py'
>>>
```

10. 能够导出 environment.yml Conda 环境配置文件，能够删除 Conda 环境，能够用 environment.yml 配置文件重建 Conda 环境

conda env export 放在 C 盘下面了

```
(prj1)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如
$ conda env export -f environment.yml
```

移动例：move /c/Users/李意如/environment.yml /c/Users/李意如/Documents/

重建 先删：conda activate base

conda env list

conda env remove -n prj1

重装 cd 到 environment.yml 文件夹

输入：conda env create

or：conda env create -f /path/to environment.yml

11. 理解 Conda 与 Python 的关系，理解 Conda-Forge 与 Conda 的关系，理解 Python 解释器、第三方软件包、PyPI 软件仓库、以及程序/软件包的路径问题

12. 按照 教程 创建项目目录，在 VS Code 文本编辑器里安装一些支持 Python 开发的常用扩展，编写 main.py 脚本，创建该项目专用的 Conda 环境，在终端里激活该环境并成功运行该脚本

```
(base)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo
$ mkdir myproject
```

用 vscode 打开 myproject

新建，编写

```
environment.yml
1 name: myproject
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.12
```



```
(base)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo
$ cd myproject
(base)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$ cat environment.yml
name: myproject
channels:
  - conda-forge
dependencies:
  - python=3.12(base)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$
```

```
(base)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$ conda env create
```

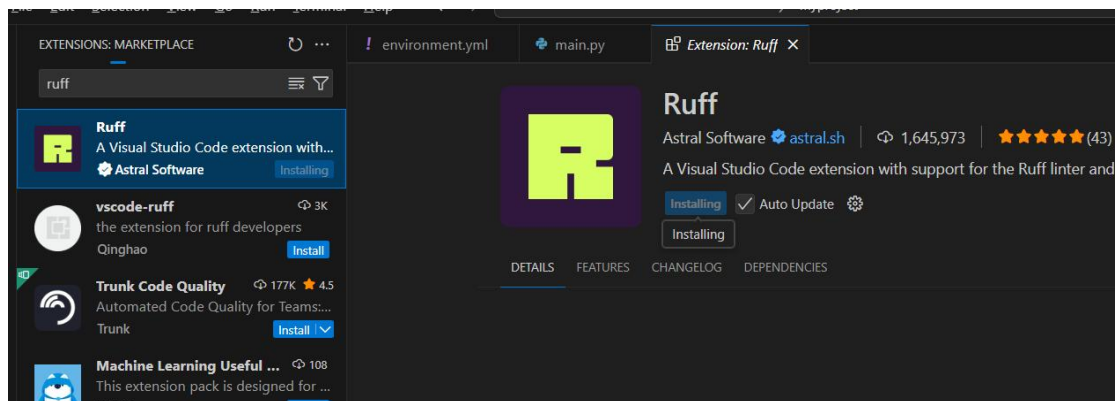
装好了

```
(base)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$ conda env list
# conda environments:
#
base * D:\anaconda3
myproject D:\anaconda3\envs\myproject
prj1 D:\anaconda3\envs\prj1
prj2 D:\anaconda3\envs\prj2
```

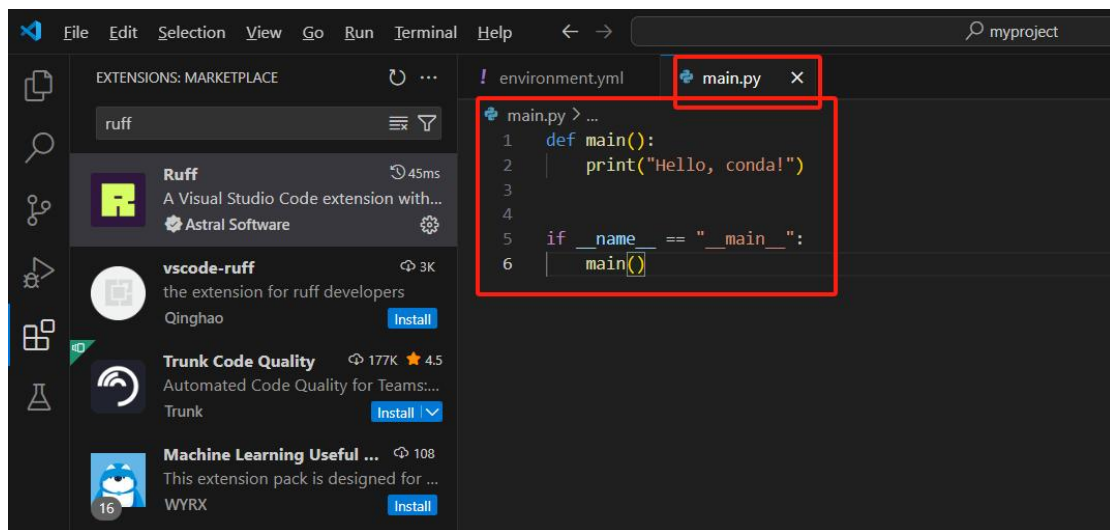
已激活

```
(base)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$ conda activate myproject
(myproject)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$
```

格式化 自动规范格式化

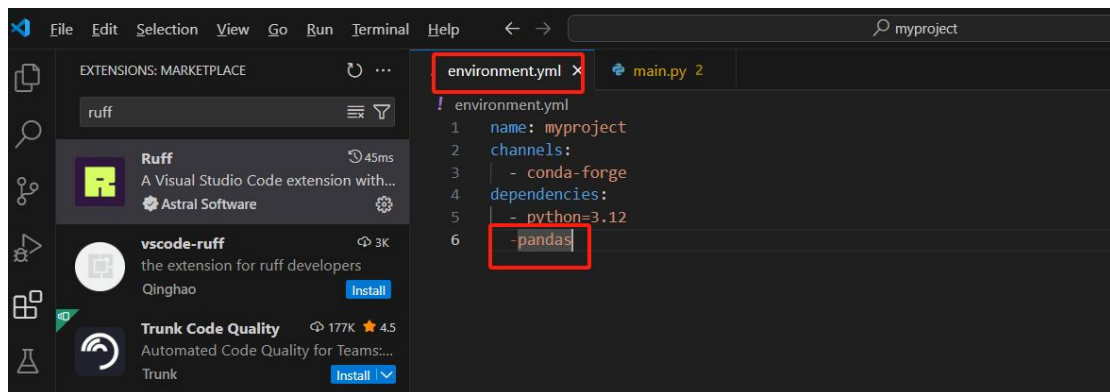


创建 main.py 并进行编写



This screenshot shows the Visual Studio Code interface. On the left, the 'EXTENSIONS: MARKETPLACE' sidebar is open, displaying a search for 'ruff'. The main editor area shows a new file named 'main.py' being created. The file content is as follows:

```
main.py > ...
1 def main():
2     print("Hello, conda!")
3
4
5 if __name__ == "__main__":
6     main()
```



This screenshot shows the Visual Studio Code interface with the 'environment.yml' file open in the editor. The file content is as follows:

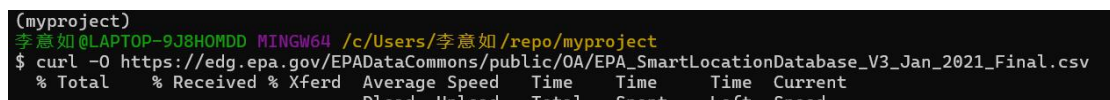
```
environment.yml
1 name: myproject
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.12
6   - pandas
```

更新



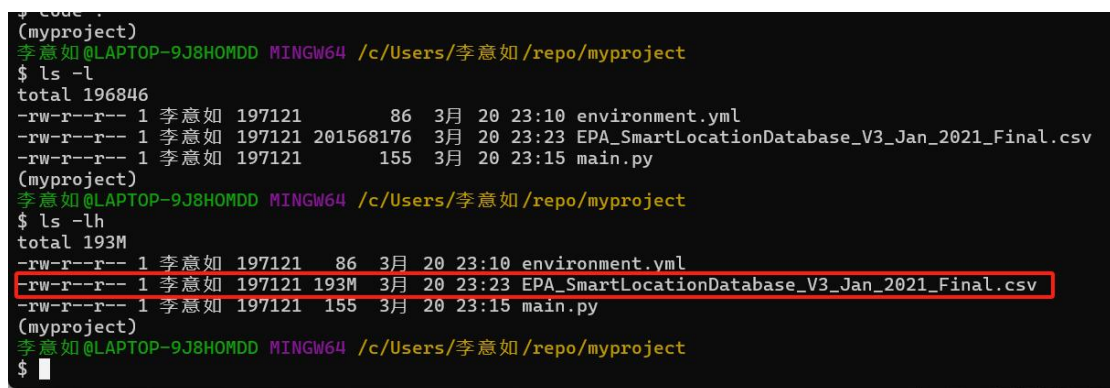
```
(myproject)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$ conda env update
```

下载 csv 文件



```
(myproject)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$ curl -O https://edg.epa.gov/EPADDataCommons/public/OA/EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv
% Total % Received % Xferd Average Speed Time Time Time Current
      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0
```

查看



```
(myproject)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$ ls -lh
total 196846
-rw-r--r-- 1 李意如 197121 86 3月 20 23:10 environment.yml
-rw-r--r-- 1 李意如 197121 201568176 3月 20 23:23 EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv
-rw-r--r-- 1 李意如 197121 155 3月 20 23:15 main.py
(myproject)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$ ls -lh
total 193M
-rw-r--r-- 1 李意如 197121 86 3月 20 23:10 environment.yml
-rw-r--r-- 1 李意如 197121 193M 3月 20 23:23 EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv
-rw-r--r-- 1 李意如 197121 155 3月 20 23:15 main.py
(myproject)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$
```

可以通过大模型进行学习

```
(myproject)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/myproject
$ conda env list
# conda environments:
#
base                D:\anaconda3
myproject            * D:\anaconda3\envs\myproject
prj1                 D:\anaconda3\envs\prj1
prj2                 D:\anaconda3\envs\prj2
```