

第三周主题 在 conda 下创建不同的环境，运行不同的项目

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

```
lqy0929@LAPTOP-CDK7DEUF MINGW64 ~
$ conda init bash
no change      D:\anaconda\Scripts\conda.exe
no change      D:\anaconda\Scripts\conda-env.exe
no change      D:\anaconda\Scripts\conda-script.py
no change      D:\anaconda\Scripts\conda-env-script.py
no change      D:\anaconda\condabin\conda.bat
no change      D:\anaconda\Library\bin\conda.bat
no change      D:\anaconda\condabin\_conda_activate.bat
no change      D:\anaconda\condabin\rename_tmp.bat
no change      D:\anaconda\condabin\conda_auto_activate.bat
no change      D:\anaconda\condabin\conda_hook.bat
no change      D:\anaconda\Scripts\activate.bat
no change      D:\anaconda\condabin\activate.bat
no change      D:\anaconda\condabin\deactivate.bat
modified       D:\anaconda\Scripts\activate
modified       D:\anaconda\Scripts\deactivate
modified       D:\anaconda\etc\profile.d\conda.sh
modified       D:\anaconda\etc\fish\conf.d\conda.fish
no change      D:\anaconda\shell\condabin\Conda.psm1
modified       D:\anaconda\shell\condabin\conda-hook.ps1
no change      D:\anaconda\Lib\site-packages\xontrib\conda.xsh
modified       D:\anaconda\etc\profile.d\conda.csh
modified       C:\Users\lqy0929\.bash_profile

==> For changes to take effect, close and re-open your current shell. <==
```

自启动文件夹，可自己设置命令（图中为在 VS code 中打开 .bash_profile 界面）

```
$ .bash_profile X
$ .bash_profile
1
2 # >>> conda initialize >>>
3 # !! Contents within this block are managed by 'conda init' !!
4 if [ -f '/d/anaconda/Scripts/conda.exe' ]; then
5     eval "$('/d/anaconda/Scripts/conda.exe' 'shell.bash' 'hook')"
6 fi
7 # <<< conda initialize <<<
```

重新打开终端，成功看到 base

```
(base)
lqy0929@LAPTOP-CDK7DEUF MINGW64 ~
$
```

解决 bash 不在同一行→用 VS code 打开"D:\Git\Git\etc\profile.d\git-prompt.sh"删除 13 行#new line 的命令（遇到问题找解决方法）

```
(base) lqy0929@LAPTOP-CDK7DEUF MINGW64 ~  
$
```

在 .bash_profile 文件夹中输入 PS1="n\$PS1"（在 VScode 中输入）

```
(base) lqy0929@LAPTOP-CDK7DEUF MINGW64 ~  
$ █
```

注：VS code 只是文本编辑器，打不开 dll 等一些二进制的文件，在终端中按 Ctrl I 可以清空屏幕

2. 使用 conda info 命令查看本机 Conda 的配置信息

```
$ conda info  
  
  active environment : base  
  active env location : D:\anaconda  
    shell level      : 1  
  user config file   : C:\Users\lqy0929\.condarc  
populated config files : D:\anaconda\.condarc  
                      C:\Users\lqy0929\.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_v4  
                      __conda=24.9.2=0  
                      __cuda=11.2=0  
                      __win=0=0  
  
  base environment   : D:\anaconda (writable)  
  conda av data dir  : D:\anaconda\etc\conda  
  conda av metadata url : None  
    channel URLs    : https://repo.anaconda.com/pkgs/main/win-64  
                      https://repo.anaconda.com/pkgs/main/noarch  
                      https://repo.anaconda.com/pkgs/r/win-64  
                      https://repo.anaconda.com/pkgs/r/noarch  
                      https://repo.anaconda.com/pkgs/msys2/win-64  
                      https://repo.anaconda.com/pkgs/msys2/noarch
```

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（学习使用大模型学习知识→检验并实际应用）

```
(base) lqy0929@LAPTOP-CDK7DEUF MINGW64 ~  
$ conda env list  
# conda environments:  
#  
base * D:\anaconda
```

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`（用来访问 `http` 的地址） 软件包，

```
(base) lqy0929@LAPTOP-CDK7DEUF MINGW64 /d
$ 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:\anaconda\envs\prj1

added / updated specs:
  - python=3.12
  - requests
```

5. 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 `conda activate prj1/prj2`，(`conda deactivate`)验证路径→`which Python`，之后用 `import` 可以导入之前下载的软件包（相反要移除环境：`conda env remove -n prj1`）

用 **Python** 做项目→一道菜；下载的软件包→原材料

可以建立不同的 **Python** 环境（避免依赖冲突隔离环境等）

```
(base) lqy0929@LAPTOP-CDK7DEUF MINGW64 /d
$ conda env list
# conda environments:
#
base                  *  D:\anaconda
prj1                  D:\anaconda\envs\prj1
prj2                  D:\anaconda\envs\prj2
```

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

使用 `conda install` 命令往 **Conda** 环境里安装更多的软件包，并验证版本，注意版本是否一致

根据 文档，配置 **Anaconda** 清华镜像，加快 `conda install` 安装软件包的速度，在`.condaec`中设置途中语句将 `conda-forge` 设置为默认 **Channel**，让 `conda install` 能够安装更多的软件

```
$ conda list
# packages in environment at D:\anaconda\envs\prj1:
#
# Name                                Version                                Build      Channe
l
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
jedi                                  0.19.2                               py312haa95532_0
```



```

! .condarc
1 channels:
2   - conda-forge
3   - defaults
4 show_channel_urls: true
5 default_channels:
6   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main
7   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/r
8   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/msys2
9 custom_channels:
10  conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
11  pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
12  channel_priority: strict
13

```

Conda forge 开源社区，更新及时，跨平台支持，广泛的软件包选择

注：环境是一个整体，不要单独删除软件包

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

安装 tushare（金融相关数据）

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

设为默认

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

```

导出 `environment.yml` Conda 环境配置文件，这样换电脑，或者重装系统都可以和之前一样运行

EXPLORER

LQY0929

Pictures

PrintHood

Recent

repo

Saved Games

Searches

SendTo

Templates

Videos

WPS Cloud Files

WPSDrive

新建文件夹

.bash_profile

.condarc

.gitconfig

.lessht

environment.yml

ntuser-OP1704515239125.dat.LOG1

ntuser-OP1704515239125.dat.LOG2

ntuser-OP1704515239125.dat{8a61832f-a29...

ntuser-OP1704515239125.dat{8a61832f-a29...

ntuser-OP1704515239125.dat{8a61832f-a29...

ntuser-OP1704580859736.dat.LOG1

ntuser-OP1704580859736.dat.LOG2

OUTLINE

TIMELINE

environment.yml

environment.yml

1 name: prj1

2 channels:

3 - conda-forge

4 - defaults

5 - https://repo.anaconda.com/pkg/main

6 - https://repo.anaconda.com/pkg/r

7 - https://repo.anaconda.com/pkg/msys2

8 dependencies:

9 - asttokens=2.0.5=pyhd3eb1b0_0

10 - brotli-python=1.0.9=py312h5da7b33_9

11 - bzip2=1.0.8=h2bbff1b_6

12 - ca-certificates=2025.2.25=h55532_0

13 - certifi=2025.1.31=pyhd8ed1ab_0

14 - charset-normalizer=3.3.2=pyhd3eb1b0_0

15 - colorama=0.4.6=py312haa95532_0

16 - decorator=5.1.1=pyhd3eb1b0_0

17 - executing=0.8.3=pyhd3eb1b0_0

18 - expat=2.6.4=h8ddb27b_0

19 - idna=3.7=py312haa95532_0

20 - intel-openmp=2024.2.1=h57928b3_1083

21 - ipython=8.30.0=py312haa95532_0

22 - jedi=0.19.2=py312haa95532_0

23 - libblas=3.9.0=31_h641d27c_mk1

24 - libcbblas=3.9.0=31_h5e41251_mk1

25 - libexpat=2.6.4=he0c23c2_0

26 - libffi=3.4.4=hd77b12b_1

27 - libhwloc=2.11.2=default_ha69328c_1001

28 - libiconv=1.18=h135ad9c_1

29 - liblapack=3.9.0=31_h1aa476e_mk1

30 - liblzma=5.6.4=h2466b09_0

这样换电脑，或者重装系统都可以和之前一样运行

```
lqy0929@LAPTOP-CDK7DEUF MINGW64 /d/repo/prj1
$ mv /d/repo/environment.yml ./
```

将环境文件移入新的文件夹

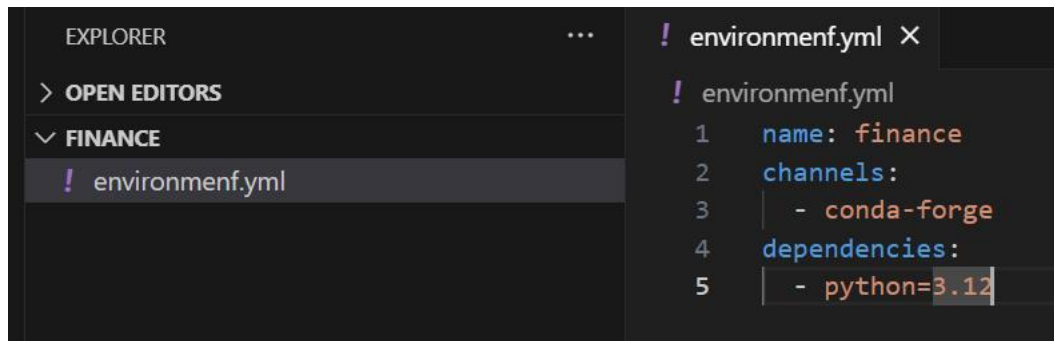
因为在当前地址下就有 environment.yml 所以就可以直接用 conda env create 就可以自动配置，如下图已经重新变回 prj1 和 prj2

```
(base)
lqy0929@LAPTOP-CDK7DEUF MINGW64 /d/repo/prj1
$ conda env list
# conda environments:
#
base                  *  D:\anaconda
prj1                  D:\anaconda\envs\prj1
prj2                  D:\anaconda\envs\prj2
(base)
```

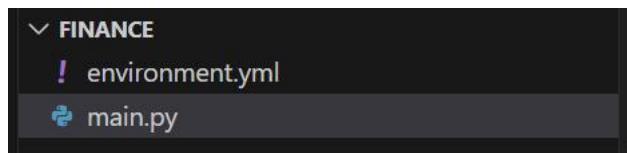
概念类问题可使用大模型了解，Conda 与 Python 的关系，理解 Python 解释器、第三方软件包、PyPI 软件仓库、以及程序/软件包的路径问题

使用 conda 而不使用 Python 环境原因：多语言支持（支持其他编程语言）；conda 可以使用不同版本的 python，环境隔离；扩平台兼容性（IOS,linux,windows）;环境共享和复制；pypi 只是 Python 下的软件包，conda-forge 包含其他语言的软件包

按照创建项目目录，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
创建项目目录最好都是小写字母不要有大写字母也不要有空格、特殊字符，在新创建的文件夹下新建 environment.yml 创建环境



之后在终端确定该文件下有相应的 environment.yml 文件再运行 conda env create
在 VScode 中新建一个 main.py



常用扩展：Python，ruff:纠正代码的语法错误；下 VS code 中编辑 main.py

```
(base) lqy0929@LAPTOP-CDK7DEUF MINGW64 ~
$ cd /d/repo/finance

(base) lqy0929@LAPTOP-CDK7DEUF MINGW64 /d/repo/finance
$ ls -l
total 1
-rw-r--r-- 1 lqy0929 197121 73  3月 25 15:25 environmentf.yml

(base) lqy0929@LAPTOP-CDK7DEUF MINGW64 /d/repo/finance
$ ls -l
total 1
-rw-r--r-- 1 lqy0929 197121 73  3月 25 15:26 environment.yml

(base) lqy0929@LAPTOP-CDK7DEUF MINGW64 /d/repo/finance
$ conda env create
```

```

main.py > ...
1  def main():
2      print("Hello, conda!")
3
4
5  if __name__ == "__main__":
6      main()
7

```

```

lqy0929@LAPTOP-CDK7DEUF MINGW64 /d/repo/finance
$ python main.py
Hello, conda!
(finance)

```

在 environment.yml 中可以设置下载想用的软件包，也可以 conda install

```

! environment.yml x main.py
! environment.yml
1  name: finance
2  channels:
3      - conda-forge
4  dependencies:
5      - python=3.12
6      - pandas

```

在 VS code 中输入

```

main.py > ...
1  import pandas as pd
2
3
4  def main():
5      print("Hello, conda!")
6      print(pd.__version__)
7      print(pd.__file__)
8
9
10 if __name__ == "__main__":
11     main()
12

```

之后在终端中运行 Python main.py

```

lqy0929@LAPTOP-CDK7DEUF MINGW64 /d/repo/finance
$ python main.py
Hello, conda!
2.2.3
D:\anaconda\envs\finance\Lib\site-packages\pandas\__init__.py
(finance)

```

在网站中下载数据

```
curl -O https://edg.epa.gov/EPADDataCommons/public/OA/EPA\_SmartLocationDatabase\_V3\_Jan\_2021\_Final.csv
```

```
lqy0929@LAPTOP-CDK7DEUF MINGW64 /d/repo/finance
$ ls -l
total 2946
-rw-r--r-- 1 lqy0929 197121      85  3月 25 16:02 environment.yml
-rw-r--r-- 1 lqy0929 197121 3014656  3月 25 16:30 EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv
-rw-r--r-- 1 lqy0929 197121      161  3月 25 16:08 main.py
(finance)
```

Curl 可以像互联网一样地发出请求

之后再运行下图中的命令，得到“美国居民中有多少百分比生活在高度适合步行的社区？”这一问题的答案

```
! environment.yml  main.py  Settings  settings.json
main.py > main
1  import pandas as pd
2
3
4  def main():
5      """
6      Answers the question:
7
8      What percentage of U.S. residents live highly walkable neighborhoods?
9
10     "15.26" is the threshold on the index for a highly walkable area.
11     """
12     csv_file = "./EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv"
13     highly_walkable = 15.26
14
15     df = pd.read_csv(csv_file)
16
17     total_population = df["TotPop"].sum()
18     highly_walkable_pop = df[df["NatWalkInd"] >= highly_walkable]["TotPop"].sum()
19
20     percentage = (highly_walkable_pop / total_population) * 100.0
21
22     print(f"{percentage:.2f}% of U.S. residents live in highlywalkable neighborhoods.")
23
24
25 if __name__ == "__main__":
26     main()
27
```

得到结果

```
lqy0929@LAPTOP-CDK7DEUF MINGW64 /d/repo/finance
$ python main.py
10.69% of U.S. residents live in highlywalkable neighborhoods.
(finance)
```