

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)

```
11 else
12     PS1='\[\033]0;$TITLEPREFIX:$PWD\007\]' # set window title
13     #PS1="$PS1"\n' # new line
14     PS1="$PS1"'\[\033[32m\' # change to green
15     PS1="$PS1"'\u@\h ' # user@host<space>
16     PS1="$PS1"'\[\033[35m\' # change to purple
17     PS1="$PS1"'\$MSYSTEM ' # show MSYSTEM
```

```
(base) Administrator@PC-20241018DCUL MINGW64 ~
$
```

```
(base) Administrator@PC-20241018DCUL MINGW64 ~
$
```

2. 使用 conda info 命令查看本机 Conda 的配置信息

```
(base) Administrator@PC-20241018DCUL MINGW64 ~
$ conda info

active environment : base
active env location : C:\Users\Administrator\anaconda3
shell level : 1
user config file : C:\Users\Administrator\.condarc
populated config files : C:\Users\Administrator\anaconda3\.condarc
                        C:\Users\Administrator\.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
                  __win=0=0
base environment : C:\Users\Administrator\anaconda3 (writable)
conda av data dir : C:\Users\Administrator\anaconda3\etc\conda
conda av metadata url : None
channel URLs : https://repo.anaconda.com/pkgs/main/win-64
               https://repo.anaconda.com/pkgs/main/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

```
(base) Administrator@PC-20241018DCUL MINGW64 ~
$ conda env list
# conda environments:
#
base * C:\Users\Administrator\anaconda3
```

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

```
(base) Administrator@PC-20241018DCUL MINGW64 ~
$ conda create -n prj1 python=3.12 requests
Channels:
- defaults
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\Administrator\anaconda3\envs\prj1

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

```
(base) Administrator@PC-20241018DCUL MINGW64 ~
$ conda create -n prj2 python=3.9 pandas statsmodels
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\Administrator\anaconda3\envs\prj2

added / updated specs:
- pandas
- python=3.9
- statsmodels
```

```
(base) Administrator@PC-20241018DCUL MINGW64 ~
$ conda env list
# conda environments:
#
base * C:\Users\Administrator\anaconda3
prj1 C:\Users\Administrator\anaconda3\envs\prj1
prj2 C:\Users\Administrator\anaconda3\envs\prj2
```

```
Administrator@PC-20241018DCUL MINGW64 ~
$ which python
/c/Users/Administrator/anaconda3/envs/prj1/python
(prj1)
Administrator@PC-20241018DCUL MINGW64 ~
$ python --version
Python 3.12.9
(prj1)
```

```
Administrator@PC-20241018DCUL MINGW64 ~
$ python
Python 3.9.21 (main, Dec 11 2024, 16:35:24) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import requests
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'requests'
>>> import pandas
>>> pandas.__file__
'C:\Users\Administrator\anaconda3\envs\prj2\lib\site-packages\pandas\__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

```
Administrator@PC-20241018DCUL MINGW64 ~
$ conda list
# packages in environment at C:\Users\Administrator\anaconda3\envs\prj2
#
# Name Version Build Channel
blas 1.0 mkl defaults
bottleneck 1.4.2 py39hc99e966_0 defaults
ca-certificates 2025.2.25 haa95532_0 defaults
icc_rt 2022.1.0 h6049295_2 defaults
intel-openmp 2023.1.0 h59b6b97_46320 defaults
mkl 2023.1.0 h6b88ed4_46358 defaults
mkl-service 2.4.0 py39h827c3e9_2 defaults
mkl_fft 1.3.11 py39h827c3e9_0 defaults
mkl_random 1.2.8 py39hc64d2fc_0 defaults
numexpr 2.10.1 py39h4cd664f_0 defaults
numpy 2.0.2 py39h055cbcc_0 defaults
numpy-base 2.0.2 py39h65a83cf_0 defaults
openssl 3.0.16 h3f729d1_0 defaults
packaging 24.2 py39haa95532_0 defaults
pandas 2.2.3 py39h5da7b33_0 defaults
patsy 1.0.1 py39haa95532_0 defaults
pip 25.0 py39haa95532_0 defaults
pybind11-abi 5 hd3eb1b0_0 defaults
python 3.9.21 h8205438_1 defaults
python-dateutil 2.9.0post0 py39haa95532_2 defaults
python-tzdata 2023.3 pyhd3eb1b0_0 defaults
pytz 2024.1 py39haa95532_0 defaults
```

##### 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

expat	2.6.4	h8ddb27b_0	defaults
idna	3.7	py312haa95532_0	defaults
ipython	8.30.0	py312haa95532_0	defaults
jedi	0.19.2	py312haa95532_0	defaults
libffi	3.4.4	hd77b12b_1	defaults
matplotlib-inline	0.1.6	py312haa95532_0	defaults
openssl	3.0.16	h3f729d1_0	defaults
parso	0.8.4	py312haa95532_0	defaults
six	25.0	py312haa95532_0	defaults

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

```
! .condarc
1 channels:
2   - conda-forge
3   - defaults
4 show_channel_urls: true
5 default_channels:
6   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
7   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
8   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/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
```

libinpthread	12.0.0.r4.gg4f2fc60ca	h57928b3_9	conda-forge
libxml2	2.13.6	he286e8c_0	conda-forge
libzlib	1.3.1	h2466b09_2	conda-forge
matplotlib-inline	0.1.6	py312haa95532_0	https://repo.anaconda.com/pkgs/main
mk1	2024.2.2	h66d3029_15	conda-forge
numpy	2.2.4	py312h50c54_0	conda-forge
openssl	3.4.1	ha4e3fda_0	conda-forge
packaging	24.2	pyhd8ed1ab_2	conda-forge
parso	0.8.4	py312haa95532_0	https://repo.anaconda.com/pkgs/main
pip	25.0	py312haa95532_0	https://repo.anaconda.com/pkgs/main
polars	1.24.0	py312haa95532_0	conda-forge
prompt-toolkit	3.0.43	py312haa95532_0	https://repo.anaconda.com/pkgs/main
prompt_toolkit	3.0.43	hd3eb1b0_0	https://repo.anaconda.com/pkgs/main
pure_eval	0.2.2	pyhd3eb1b0_0	https://repo.anaconda.com/pkgs/main
pygments	2.15.1	py312haa95532_1	https://repo.anaconda.com/pkgs/main
pysocks	1.7.1	py312haa95532_0	https://repo.anaconda.com/pkgs/main
python	3.12.9	h3f84c4b_1_cpython	conda-forge
python_abi	3.12	5_cp312	conda-forge
requests	2.32.3	py312haa95532_1	https://repo.anaconda.com/pkgs/main
setuptools	75.8.0	py312haa95532_0	https://repo.anaconda.com/pkgs/main
six	1.16.0	pyhd3eb1b0_1	https://repo.anaconda.com/pkgs/main

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

```
Administrator@PC-20241018DCUL MINGW64 ~
$ pip install tushare
Collecting tushare
  Downloading tushare-1.4.19-py3-none-any.whl.metadata (3.
Collecting pandas (from tushare)
  Downloading pandas-2.2.3-cp312-cp312-win_amd64.whl.metad
Requirement already satisfied: requests in c:\users\admini
2.32.3)
Collecting lxml (from tushare)
  Downloading lxml-5.3.1-cp312-cp312-win_amd64.whl.metadat
Collecting simplejson (from 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

```
Administrator@PC-20241018DCUL MINGW64 ~
$ python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\administrator\anaconda3\envs\
Collecting pip
  Downloading pip-25.0.1-py3-none-any.whl.metadata (3.7 kB)
  Downloading pip-25.0.1-py3-none-any.whl (1.8 MB)
  1.8/1.8 MB 4.2 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 25.0
    Uninstalling pip-25.0:
      Successfully uninstalled pip-25.0
  Successfully installed pip-25.0.1
(prj1)
Administrator@PC-20241018DCUL MINGW64 ~
$ pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi
Writing to C:\Users\Administrator\AppData\Roaming\pip\pip.ini
(prj1)
```

cornado	0.4.1	py312h027c5e9_0	https://repo.anaconda.
tqdm	4.66.5	py312hfc267ef_0	https://repo.anaconda.
traitlets	5.14.3	py312haa95532_0	https://repo.anaconda.
truststore	0.8.0	py312haa95532_0	https://repo.anaconda.
tushare	1.4.19	pypi_0	pypi
twisted	23.10.0	py312haa95532_0	https://repo.anaconda.
twisted-iocpsupport	1.0.2	py312h2bbff1b_0	https://repo.anaconda.
typing-extensions	4.11.0	py312haa95532_0	https://repo.anaconda.
typing_extensions	4.11.0	py312haa95532_0	https://repo.anaconda.
tzdata	2024b	h04d1e81_0	https://repo.anaconda.
uc-micro-py	1.0.1	py312haa95532_0	https://repo.anaconda.
ujson	5.10.0	py312h5da7b33_0	https://repo.anaconda.
unicodedata2	15.1.0	py312h2bbff1b_0	https://repo.anaconda.



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

```
! enviroment.yml
1  name: base
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    - _anaconda_depends=2024.10=py312_mkl_0
10   - aiobotocore=2.12.3=py312h9a95532_0
11   - aiohappyeyeballs=2.4.0=py312h9a95532_0
12   - aiohttp=3.10.5=py312h827c3e9_0
13   - aioitertools=0.7.1=pyhd3eb1b0_0
14   - aiosignal=1.2.0=pyhd3eb1b0_0
```

```
Administrator@PC-20241018DCUL MINGW64 ~
$ conda env list
# conda environments:
#
base                  C:\Users\Administrator\anaconda3
prj2                  C:\Users\Administrator\anaconda3
```

```
Administrator@PC-20241018DCUL MINGW64 ~/repo/prj1
$ conda env create --help
usage: conda-script.py env create [-h] [-f FILE] [-n ENVIRONMENT] [-p PATH] [-c] [-k] [--no
[no-default-packages] [--json] [-v] [-q] [-d] [-y] [--subdir SUBDIR]
[remote_definition]

Create an environment based on an environment definition file.

If using an environment.yml file (the default), you can name the
environment in the first line of the file with 'name: envname' or
you can specify the environment name in the CLI command using the
-n/--name argument. The name specified in the CLI will override
the name specified in the environment.yml file.

Unless you are in the directory containing the environment definition
file, use -f to specify the file path of the environment definition
file you want to use.
```

```
Administrator@PC-20241018DCUL MINGW64 ~
$ conda env list
# conda environments:
#
base                  C:\Users\Administrator\anaconda3
prj1                  C:\Users\Administrator\anaconda3\envs\prj1
prj2                  C:\Users\Administrator\anaconda3\envs\prj2
```

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

#### Conda 与 Python 的关系

Conda 是 Python 的包和环境管理工具：Conda 可以用于安装、管理 Python 的第三方软件包，还能创建和管理不同的 Python 环境，每个环境可以有独立的 Python 版本以及软件包组合，这有助于解决不同项目对 Python 版本和软件包依赖不同的问题。支持多语言：虽然 Conda 常用于 Python 项目，但它也可以管理其他语言的软件包和环境，如 R、Java 等，不过在 Python 生态系统中应用最为广泛。

#### Conda - Forge 与 Conda 的关系

Conda - Forge 是 Conda 的软件包渠道：Conda 本身是一个包管理系统，而 Conda - Forge 是众多软件包渠道中的一个，且是非常重要的一个。它提供了大量的开源软件包，涵盖了科学计算、数据分析、机器学习等多个领域。

社区驱动：Conda - Forge 由社区维护和管理，社区成员积极参与软件包的维护、更新和添加，保证了软件包的质量和及时更新，为 Conda 用户提供了丰富的资源。

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。

**Python 解释器：**是执行 Python 代码的程序，它负责将人类编写的 Python 代码解释成计算机能够理解的机器语言。当安装 Python 时，会将解释器安装到指定路径，例如在 Windows 上默认路径可能是 C:\PythonXX，在 Linux 上通常是 /usr/bin/python 等。不同版本的 Python 解释器可以安装在不同路径，通过 Conda 创建的虚拟环境也会有独立的 Python 解释器，位于虚拟环境的特定路径下，如 C:\Users\Administrator\anaconda3\envs\prj1\python.exe。

**第三方软件包：**是由其他人或组织开发的，用于扩展 Python 功能的代码库。安装第三方软件包后，它们的文件会被放置在 Python 的安装目录下的特定位置，具体取决于安装方式。如果使用 Conda 安装，软件包会被安装到 Conda 环境的 site - packages 目录中，例如 C:\Users\Administrator\anaconda3\envs\prj1\Lib\site - packages。如果直接使用 pip 安装，通常会安装到 Python 全局的 site - packages 目录，或者虚拟环境对应的 site - packages 目录（如果在虚拟环境中使用 pip）。

**PyPI 软件仓库：**是 Python 官方的软件包仓库，包含了大量的第三方软件包。当使用 pip 安装软件包时，默认会从 PyPI 仓库下载。Conda 也可以从 PyPI 获取软件包，但通常更推荐使用 Conda 的软件包渠道，因为 Conda 对软件包的管理更全面，包括依赖关系处理等。PyPI 仓库的地址是 <https://pypi.org/>，在安装软件包时，pip 会连接到这个地址查找并下载所需的软件包。

**程序 / 软件包的路径问题：**在 Python 中，程序在运行时需要知道在哪里找到所需的模块和软件包。Python 解释器会按照一定的顺序搜索路径，包括当前目录、Python 安装目录、site - packages 目录等。可以通过 sys.path 变量查看 Python 解释器搜索路径的列表。当导入模块时，解释器会在这些路径中查找对应的模块文件。如果软件包不在默认搜索路径中，需要将其所在路径添加到 sys.path 中，或者使用相对路径、绝对路径来正确导入模块。在使用 Conda 环境时，由于环境的独立性，每个环境都有自己的路径设置，确保了不同环境中的软件包和程序不会相互干扰。

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) Administrator@PC-20241018DCUL MINGW64 ~/repo/myproject
$ cat environment.yml
name: my-project
channels:
  - conda-forge
dependencies:
  - python=3.12
```

```
users / Administrator / AppData / Roaming / Code / User / settings.json / ...
{
  "workbench.startupEditor": "none",
  "[python]": {
    "editor.formatOnSave": true,
    "editor.codeActionsOnSave": {
      "source.fixAll": "explicit",
      "source.organizeImports": "explicit"
    },
    "editor.defaultFormatter": "charliermarsh.ruff"
  },
  "notebook.formatOnSave.enabled": true,
  "notebook.codeActionsOnSave": {
    "notebook.source.fixAll": "explicit",
    "notebook.source.organizeImports": "explicit"
  },
}
```

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

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

```
repo > myproject > 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
```

```
Administrator@PC-20241018DCUL MINGW64 ~/repo/myproject
$ python main.py
Hello, conda!
2.2.3
C:\Users\Administrator\anaconda3\envs\my-project\Lib\site-packages\pandas\__init__.py:24: DeprecationWarning: Importing from the pandas package is deprecated, please import from the sub-packages instead.
(my-project)
```

```
Administrator@PC-20241018DCUL MINGW64 ~/repo/myproject
$ python main.py
10.69% of U.S. residents live in highly walkable neighborhoods.
(my-project)
Administrator@PC-20241018DCUL MINGW64 ~/repo/myproject
```