

一、在自己的终端(比如 GitBash、Zsh 等)配置好 CondaInit, 使得启动终端后, 在提示符(比如\$, %)前能够看到(base)

配置 conda

Whichconda 来看 conda 的位置

Condainitbash

重启可以看见配置了 base

```
(base)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ |
```

二、使用 conda info 命令查看本机 Conda 的配置信息

```
(base)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda info

      active environment : base
      active env location : D:\STUDY\Anaconda3
            shell level : 1
      user config file : C:\Users\HUAWEI\.condarc
populated config files : D:\STUDY\Anaconda3\.condarc
         conda version : 24.9.2
    conda-build version : 24.9.0
         python version : 3.12.7.final.0
            solver type : libmamba (default)
virtual packages : __archspec=1=skylake
                  __conda=24.9.2=0
                  __cuda=10.2=0
                  __win=0=0
base environment : D:\STUDY\Anaconda3 (writable)
```

```
(base)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda env list
# conda environments:
#
base                                * D:\STUDY\Anaconda3

(base)
```

三、使用 conda create 命令创建两个 Conda 环境, 一个里面安装 Python3.12 和 requests 软件包, 另一个里面安装 Python3.9、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

-n 是在命名, 把这个起名为 prj1 (意思是 project1)



```
(base)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda activate prj1
(prj1)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ python --version
Python 3.12.9
```

同理，用 `conda activate prj2`，切换 `prj2` 的环境（可以想象不同的生产车间）

使用 `conda list` 命令显示 Conda 环境里的软件包列表及其版本信息  
（像更换到 `prj2` 还是用 `activate` 命令）

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

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

## 项目简介

Anaconda 是一个用于科学计算的 Python 发行版，支持 Linux, Mac, Windows, 包含了众多流行的科学计算、数据分析的 Python 包。

Anaconda 安装包可以到以下链接下载。

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>

## 使用方法

镜像站提供了 Anaconda 仓库与第三方源（`conda-forge`、`msys2`、`pytorch` 等，各镜像站镜像的第三方源并不相同，可以参考下方「第三方镜像源」一节）的镜像，各系统都可以通过修改用户目录下的 `.condarc` 文件来使用镜像站。

不同系统下的 `.condarc` 目录如下：

- Linux: `${HOME}/.condarc`
- macOS: `${HOME}/.condarc`
- Windows: `C:\Users\<YourUserName>\.condarc`

注：

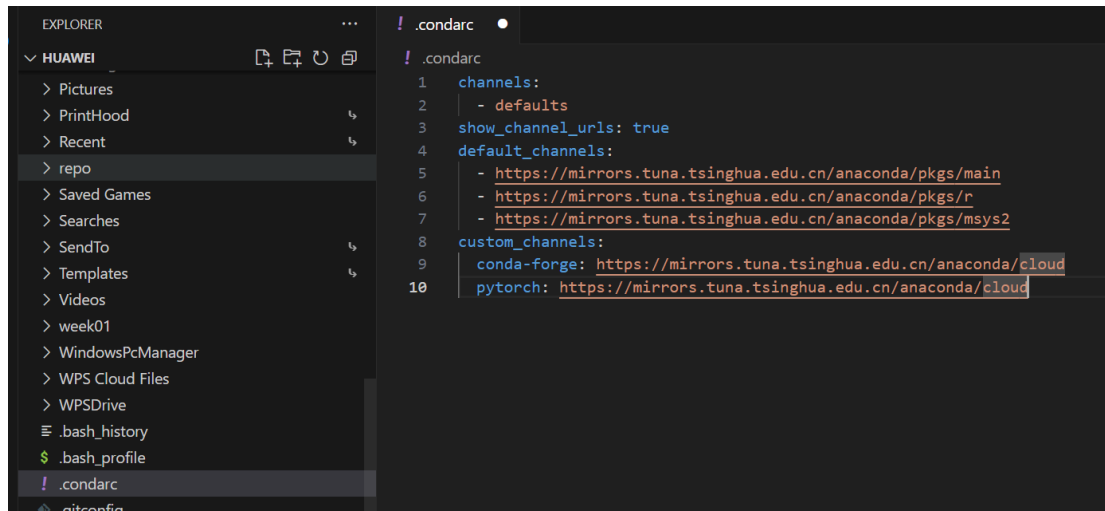
- \* Windows 用户无法直接创建名为 `.condarc` 的文件，可先执行 `conda config --set show_channel_urls yes` 生成该文件之后再修改。
- \* 由于更新过快难以同步，TUNA 等镜像站不同步 `pytorch-nightly`，`pytorch-nightly-cpu`，`ignite-nightly` 这三个包。
- \* 如果您正在从某一镜像源切换到另一镜像源，请检查镜像源是否同步了您所需要的 repo，以及该 repo 是否支持您使用的平台 (e.g. linux-64)。
- \* 为了保证以下配置在所有镜像站可用，配置中只加入了少量必须的第三方源，您可以在下方的列表中自行寻找并添加其他第三方源。

按照提示进行修改

```

(prj1)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda config --set show_channel_urls yes
(prj1)

```



将 conda-forge 设置为默认 Channel

```

! .condarc
! .condarc
1 channels:
2   - conda-forge
3 show_channel_urls: true
4 default_channels:
5   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
6   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
7   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/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
11

```

使用下列命令清除索引缓存，然后下载 polarsconda clean -i  
conda create -n myenv numpy

```

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda config --set channel_priority strict
(prj1)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda install polars
Channels:
- defaults
Platform: win-64
collecting package metadata (repodata.json): / |

```

删除仓库 1 (prj1) ,这个时候仓库就只有 prj2。也可以再把 prj1 创建,这次多了 polars

```
(prj1)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda deactivate
(prj2)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda env remove -n prj1

Remove all packages in environment D:\STUDY\Anaconda3\envs\prj1:

## Package Plan ##

  environment location: D:\STUDY\Anaconda3\envs\prj1
```

```
(prj2)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda env list
# conda environments:
#
base                                D:\STUDY\Anaconda3
prj2                                * D:\STUDY\Anaconda3\envs\prj2

(prj2)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda create -n prj1 python=3.12 requests polars-lts-cpu
Channels:
 - conda-forge
 - https://repo.anaconda.com/pkgs/main
 - https://repo.anaconda.com/pkgs/r
 - https://repo.anaconda.com/pkgs/msys2
Platform: win-64
Collecting package metadata (repodata.json): \ |
```

```
(prj2)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda env list
# conda environments:
#
prj1                                C:\Users\HUAWEI\.conda\envs\prj1
base                                D:\STUDY\Anaconda3
prj2                                * D:\STUDY\Anaconda3\envs\prj2

(prj2)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda activate prj1
(prj1)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ python
Python 3.12.9 | packaged by conda-forge | (main, Mar  4 2025, 22:37:18) [M
943 64 bit (AMD64)] on win32
Type 'help', 'copyright', 'credits' or 'license' for more information.
>>> import polars
>>>
```

```
(prj1)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ pip install tushare
Collecting tushare
  Downloading tushare-1.4.21-py3-none-any.whl.metadata (3.3 kB)
Collecting pandas (from tushare)
  Downloading pandas-2.2.3-cp312-cp312-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: requests in c:\users\huawei\.conda\envs\prj1\lib\
site-packages (from tushare) (2.32.3)
Collecting lxml (from tushare)
  Downloading lxml-5.3.1-cp312-cp312-win_amd64.whl.metadata (3.8 kB)
Collecting simplejson (from tushare)
  Downloading simplejson-3.20.1-cp312-cp312-win_amd64.whl.metadata (3.4 kB)
Collecting bs4 (from tushare)
  Downloading bs4-0.0.2-py2.py3-none-any.whl.metadata (411 bytes)
Collecting websocket-client>=0.57.0 (from tushare)
  Downloading websocket_client-1.8.0-py3-none-any.whl.metadata (8.0 kB)
Collecting tqdm (from tushare)
  Downloading tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)
Collecting beautifulsoup4 (from bs4->tushare)
```

```
(prj1)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi/web/
simple
Writing to C:\Users\HUAWEI\AppData\Roaming\pip\pip.ini
(prj1)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 /
$ conda env export
name: prj1
channels:
- conda-forge
- https://repo.anaconda.com/pkgs/main
- https://repo.anaconda.com/pkgs/r
- https://repo.anaconda.com/pkgs/msys2
dependencies:
- brotli-python=1.1.0=py312h275cf98_2
- bzip2=1.0.8=h2466b09_7
- ca-certificates=2025.1.31=h56e8100_0
- certifi=2025.1.31=pyhd8ed1ab_0
```

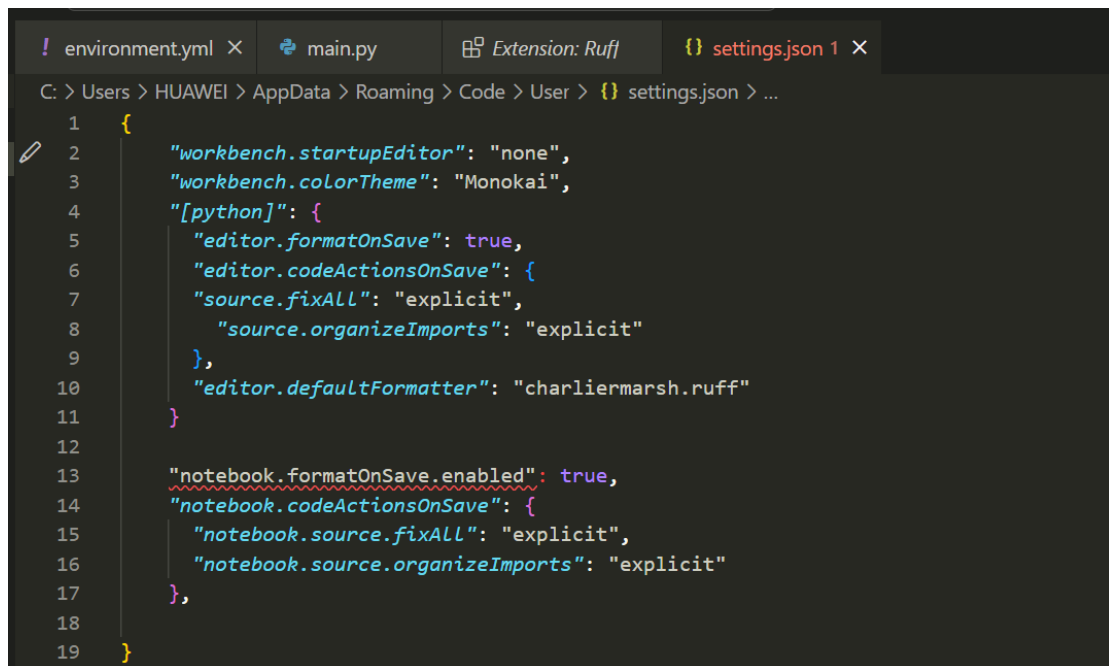
运行出 hello conda

The screenshot shows a code editor with a file named `main.py` containing the following Python code:

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

Below the code editor, a terminal window displays the output of running the script. The terminal shows the environment variables and the command `$ python main.py` being executed, resulting in the output `Hello, conda!`.

```
(myproject)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo/myproject
$ python main.py
Hello, conda!
(myproject)
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo/myproject
$
```



```
1  {
2      "workbench.startupEditor": "none",
3      "workbench.colorTheme": "Monokai",
4      "[python]": {
5          "editor.formatOnSave": true,
6          "editor.codeActionsOnSave": {
7              "source.fixAll": "explicit",
8              "source.organizeImports": "explicit"
9          },
10         "editor.defaultFormatter": "charliermarsh.ruff"
11     }
12
13     "notebook.formatOnSave.enabled": true,
14     "notebook.codeActionsOnSave": {
15         "notebook.source.fixAll": "explicit",
16         "notebook.source.organizeImports": "explicit"
17     },
18
19 }
```