

Python 项目（用以解决实际问题）

一、在终端配置 Conda 初始化

1. 确定命令行能找到 Conda 命令

2. 执行 `conda init bash`

执行 `conda init` 的作用

1. **修改 Shell 配置文件**: `conda init` 会修改 `.bashrc`、`.zshrc`、`.bash_profile` 等终端配置文件（根据你使用的 Shell）。
2. **自动激活 base 环境**: 执行后，打开终端时会默认激活 Conda 的 `base` 环境。
3. **调整 PATH 变量**: 确保 Conda 相关命令在终端可用，而不需要手动添加 `PATH`。

3. 查看被刚刚被修改的启动文件（用 `code` 启动 vs `code` 打开当前文件夹，或在 vs `code` 手动打开）

```
$.bash_profile X
$.bash_profile
1
2 # >>> conda initialize >>>
3 # !! Contents within this block are managed by 'conda init' !!
4 if [ -f '/c/Users/71970/anaconda3/Scripts/conda.exe' ]; then
5     eval "$('/c/Users/71970/anaconda3/Scripts/conda.exe' 'shell.bash' 'hook')"
6 fi
7 # <<< conda initialize <<<
8
9
```

主要作用是在终端启动时自动初始化 Conda，使其环境管理功能生效。

4. 再次打开终端查看效果：

```
(base)
71970@132 MINGW64 ~
$
```

5. 修复 `(base)` 位置、添加换行（根据老师提供的链接）



joelostblom on Aug 5, 2020

I would be happy to try to submit a PR on this if someone can provide guidance on where this should go, e.g. [the update_prompt python function](#), in [activate.bat](#), or maybe in [conda.sh](#)?

And what type of fix would be accepted here? Something simple as if the OS is windows and `PS1` starts with a newline, the conda environment should be inserted after and not before the newline. Or actually trying to determine if Git Bash is installed and only change the prompt then?

In the meantime, a workaround is to edit `"/c/Program Files/Git/etc/profile.d/git-prompt.sh"` to delete the line that reads the following (it should be line 13):

```
PS1="$PS1""\n" # new line
```

用注释# 的方式删除行

And then if you want a newline before `(base)` in your prompt string, append `PS1="\n$PS1"` to `~/.bash_profile`.

在启动文档中添加内容

```
(base) 71970@132 MINGW64 ~
$ pwd
/c/Users/71970

(base) 71970@1 MINGW64 ~
$
```

二、创建 Conda 环境

1. 用 conda info 查看本机配置信息

```
$ conda info

active environment : base
active env location : C:\Users\71970\anaconda3
shell level : 1
user config file : C:\Users\71970\.condarc
```

2. 用 conda env list 查看已有的 Conda 环境的名称和路径

```
$ conda env list
# conda environments:
#
base * C:\Users\71970\anaconda3
```

3. 理解 Conda 环境

定义

Conda 环境是一个独立的空间，在这个空间里可以安装特定版本的 Python 解释器和各种软件包。不同的环境之间相互隔离，互不干扰。

作用

- **隔离项目依赖**：不同的项目可能会依赖不同版本的库，比如项目 A 依赖 TensorFlow 1.x 版本，而项目 B 依赖 TensorFlow 2.x 版本。借助 Conda 环境，能够为每个项目构建独立的环境，分别安装所需的库版本，从而避免版本冲突。
- **复现开发环境**：你可以把环境的配置信息导出为一个文件，其他人能够利用这个文件在他们的机器上创建出相同的环境，这样就保证了项目在不同机器上的一致性。
- **管理 Python 版本**：在不同的 Conda 环境中可以安装不同版本的 Python。比如，你既可以创建一个 Python 3.7 的环境来运行旧项目，又可以创建一个 Python 3.10 的环境来开发新项目。

4. 用 conda create 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 和 pandas 、 statsmodels 软件包）

```
(base) 71970@13. MINGW64 ~
$ conda create -n task2 python=3.9 pandas statsmodels
```

创建新环境：使用 conda create 命令，如 conda create -n my_env python=3.8 创建名为 my_env 的 Python 3.8 环境。也可在创建时指定安装特定的包，如 conda create -n my_env python=3.8 numpy=1.21。

5. 再次查看环境

```
$ conda env list
# conda environments:
#
base * C:\Users\71970\anaconda3
task1 C:\Users\71970\anaconda3\envs\task1
task2 C:\Users\71970\anaconda3\envs\task2
```

6. 激活、切换环境，验证 python 和软件包版本

```
(base) 71970@1326 MINGW64 ~
$ conda activate task1
(task1)
71970@1326 MINGW64 ~
$ which python
/c/Users/71970/anaconda3/envs/task1/python
(task1)
71970@1326 MINGW64 ~
$ python --version
Python 3.12.9
(task1)
71970@1326 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 'C:\\Users\\71970\\anaconda3\\envs\\task1\\Lib\\site
-packages\\requests\\__init__.py'>
>>> import pandas
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'pandas'
>>> quit()
(task1)
71970@1326 MINGW64 ~
$ conda activate task2
(task2)
71970@1326 MINGW64 ~
$ which python
/c/Users/71970/anaconda3/envs/task2/python
(task2)
71970@1326 MINGW64 ~
$ python --version
Python 3.9.21
71970@1326 MINGW64 ~
$ python
Python 3.9.21 (main, Dec 11 2024, 16:35:24) [MSC v.1929 64 bit (AMD64)] on w
in32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> import pandas
>>> pandas.__file__
'C:\\Users\\71970\\anaconda3\\envs\\task2\\lib\\site-packages\\pandas\\__ini
t__.py'
>>> pandas.__version__
'2.2.3'
>>> import statsmodels
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'statsmodels'
>>> import statsmodels
>>> statsmodels.__version__
'0.14.4'
>>> quit
Use quit() or Ctrl-Z plus Return to exit
>>> quit()
```

7. 用 conda list 查看环境中软件包列表和版本信息

```
71970@1326 MINGW64 ~
$ conda list
# packages in environment at C:\Users\71970\anaconda3\envs\task1:
#
# Name                                Version                                Build      Channel
brotli-python                         1.0.9                                py312h5da7b33_9
bzip2                                 1.0.8                                h2bbff1b_6
```

8.向 Conda 环境中添加软件包并验证版本

```
71970@1326 MINGW64 ~  
$ conda activate task1  
(task1)  
71970@1326 MINGW64 ~  
$ conda install ipython  
Channels:  
- defaults  
Platform: win-64  
Collecting package metadata (repodata.json): done  
Solving environment: done  
  
done  
(task1)  
71970@1326 MINGW64 ~  
$ conda list  
# packages in environment at C:\Users\71970\anaconda3\envs\task1:  
#  
# 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  
jedi                                   0.19.2                               py312haa95532_0
```

三、Conda-Forge 清华镜像

1.前往清华大学开源软件镜像站-anaconda 查看方法

2.在终端用命令生成名为 .condarc 的文件

```
$ conda config --set show_channel_urls yes
```

3.将文件内容替换为：（修改为通过镜像站下载，更快）

```
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
```

4.Channel 改为 conda-forge 优先（社区驱动，提供数量众多的软件包）

```
! .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
```

5. 安装 polars 软件包

```
(task1)
71970@1326 MINGW64 ~
$ conda install polars
Channels:
- conda-forge
- defaults
- https://repo.anaconda.com/pkgs/main
- https://repo.anaconda.com/pkgs/r
- https://repo.anaconda.com/pkgs/msys2
Platform: win-64

(task1)
71970@1326 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
>>>
```

PS: 下载某个软件包发现不可用, 不要单独 remove 它, 先 deactivate 回到 base, conda env list 查看环境, 用 conda env remove -n ** 的方式删除有问题环境。

6. 安装 tushare 软件包: 用 conda install 是找不到的, 需要用 pip install(直接装很慢, 同样可以配置 PyPI 清华镜像来加速)

```
(task1)
71970@1326 MINGW64 ~
$ python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\71970\anaconda3\envs\task1\lib\site-packages (25.0)
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 1.0 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
(task1)
71970@1326 MINGW64 ~
$ pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
Writing to C:\Users\71970\AppData\Roaming\pip\pip.ini
(task1)
71970@1326 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/2b/b1/8aa06e934b778920624047770cc8fd5518a1e9d3066043150a691f71edc1/tushare-1.4.19-p
```

```
71970@1326 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__
'C:\\Users\\71970\\anaconda3\\envs\\task1\\Lib\\site-packages\\tushare\\__init__.py'
>>>
```

四、环境配置的导出与重建

1. 导出环境配置文件: `conda env export -f environment.yml`

```
! environment.yml
1  name: task1
2  channels:
3    - conda-forge
4    - defaults
5    - https://repo.anaconda.com/pkgs/main
6    - https://repo.anaconda.com/pkgs/r
7    - https://repo.anaconda.com/pkgs/msys2
8  dependencies:
9    - asttokens=2.0.5=pyhd3eb1b0_0
10   - brotli-python=1.0.9=py312h5da7b33_9
```

2. 删除上述导出过的环境: `conda env remove -n **`

```
$ conda env list
# conda environments:
#
base                  * C:\Users\71970\anaconda3
task2                 C:\Users\71970\anaconda3\envs\task2
```

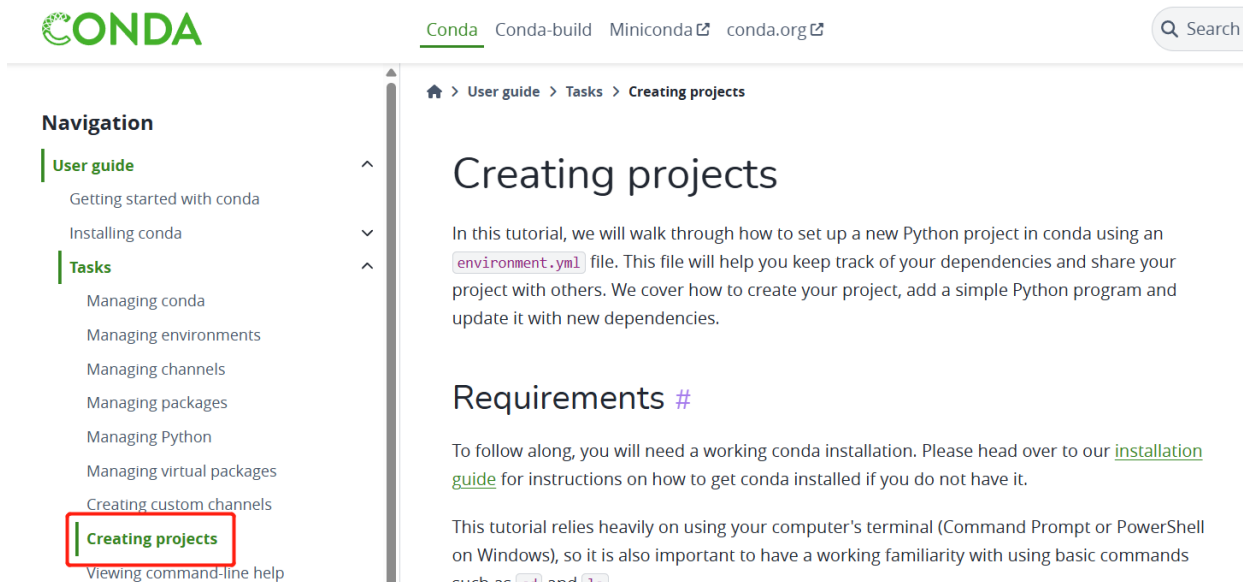
3. 用配置文件重建环境: `conda env creat`(在配置文件所在的路径下运行)

```
$ conda env list
# conda environments:
#
base                  * C:\Users\71970\anaconda3
task1                 C:\Users\71970\anaconda3\envs\task1
task2                 C:\Users\71970\anaconda3\envs\task2
```

PS: 第一次重建失败, 显示受制于优先级设定, 将`.condarc` 文件中 `channel_priority: strict` 注释掉, 再次重建成功。

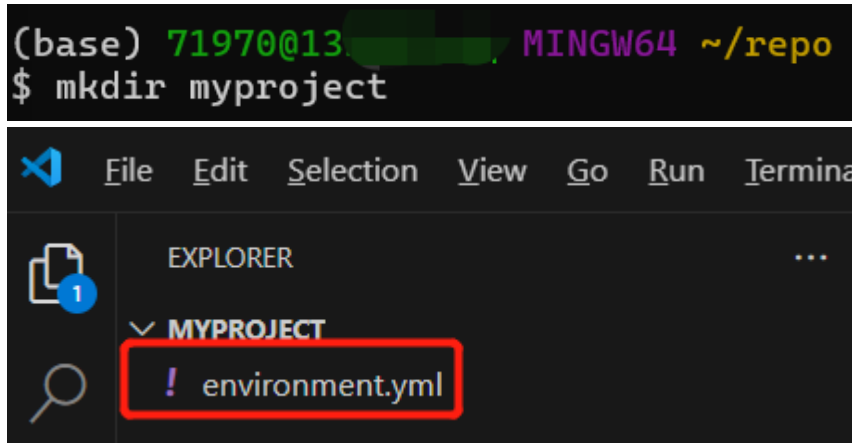
五、编写和运行 Python 程序

1. 参考官方网站

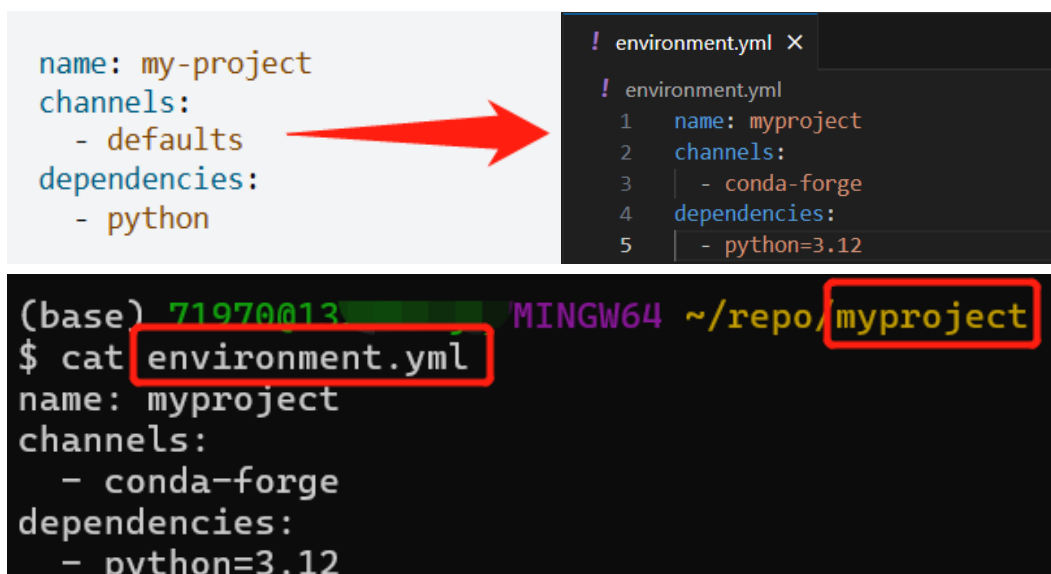


The screenshot shows the Conda website's 'Creating projects' tutorial page. The navigation sidebar on the left lists 'User guide' and 'Tasks'. Under 'Tasks', 'Creating projects' is highlighted with a red box. The main content area is titled 'Creating projects' and contains introductory text about setting up a new Python project in Conda using an `environment.yml` file. It also includes a 'Requirements' section with a heading and introductory text.

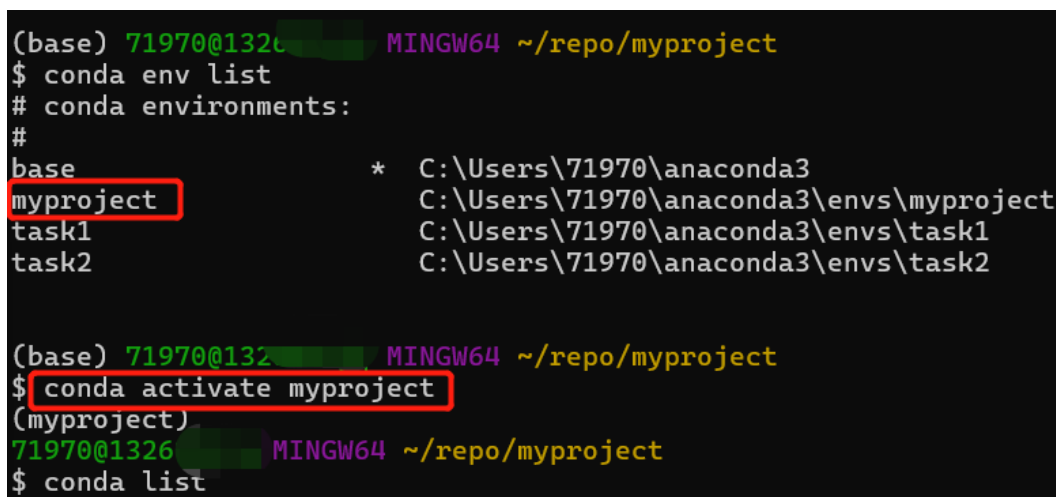
2.用终端新建一个文件夹 myproject，用 vs code 打开，并在其中新建一个文件 environment.yml



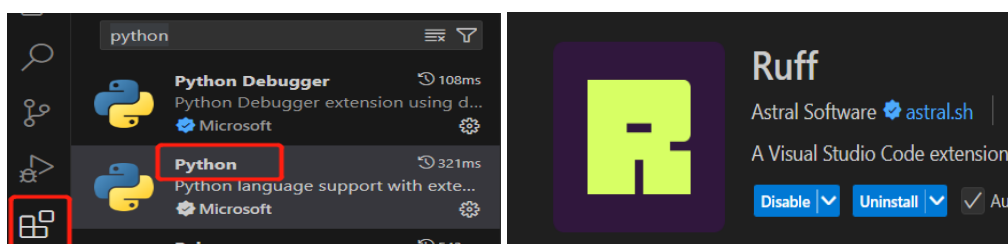
3.向上述文件中添加内容并做相应修改:



4.创建环境（在 myproject 下执行 `conda env create`）并激活:



5.在 vs code 安装拓展（python 和 ruff）



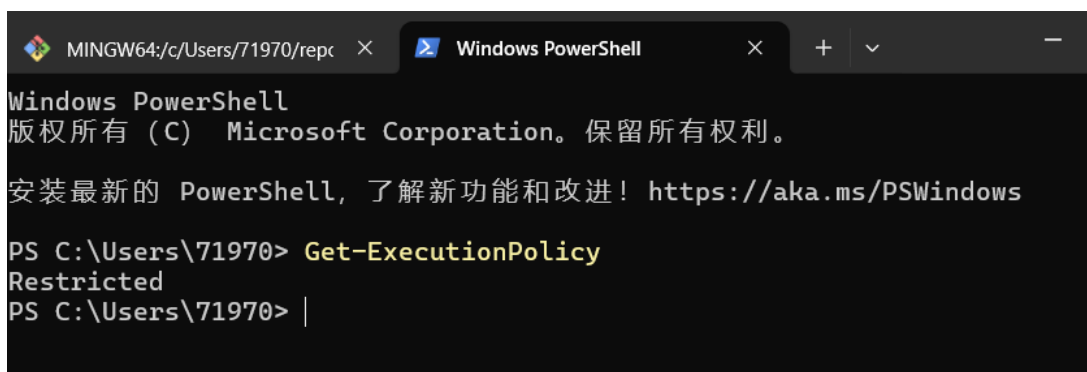
6. 在 vs code 新建文件 main.py 并添加内容

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

7. 在终端运行:

```
71970@1326 MINGW64 ~/repo/myproject
$ python main.py
(myproject)
71970@1326 MINGW64 ~/repo/myproject
$ python main.py
Hello, conda!
```

PS:第一次运行失败，解决方案如图



```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

PS C:\Users\71970> Get-ExecutionPolicy
Restricted
PS C:\Users\71970> |
```

在当前非管理员权限的 PowerShell 终端界面下，可以先输入以下命令并回车：

```
plaintext ^
start-process PowerShell -verb runas
```

这条命令会弹出用户账户控制（UAC）提示框，点击“是”，就会打开一个以管理员身份运行的新 PowerShell 窗口。

然后在新打开的管理员身份的 PowerShell 窗口中，输入以下命令修改执行策略：

```
plaintext ^
Set-ExecutionPolicy RemoteSigned -Force
```

其中，RemoteSigned表示允许运行本地编写的脚本，从互联网下载的脚本需要签名才能运行；-Force参数会跳过确认提示，直接应用策略。

8. 安装 pandas(可以用 conda install,但这里介绍一种较优的方式)

①在 vs code 中添加 pandas 至 environment.yml

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

②终端更新: conda env update

9. 编辑脚本, 添加 pandas 相关内容并运行

```
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
```

```
71970@13269 MINGW64 ~/repo/myproject
$ python main.py
Hello, conda!
2.2.3
C:\Users\71970\anaconda3\envs\myproject\Lib\site-packages\pandas\__init__.py:24: DeprecationWarning:
The distutils package is deprecated and slated for removal in Python 3.12. Use setuptools or
Cython instead to build and distribute packages.
  from distutils.command.build import build
(myproject)
```