

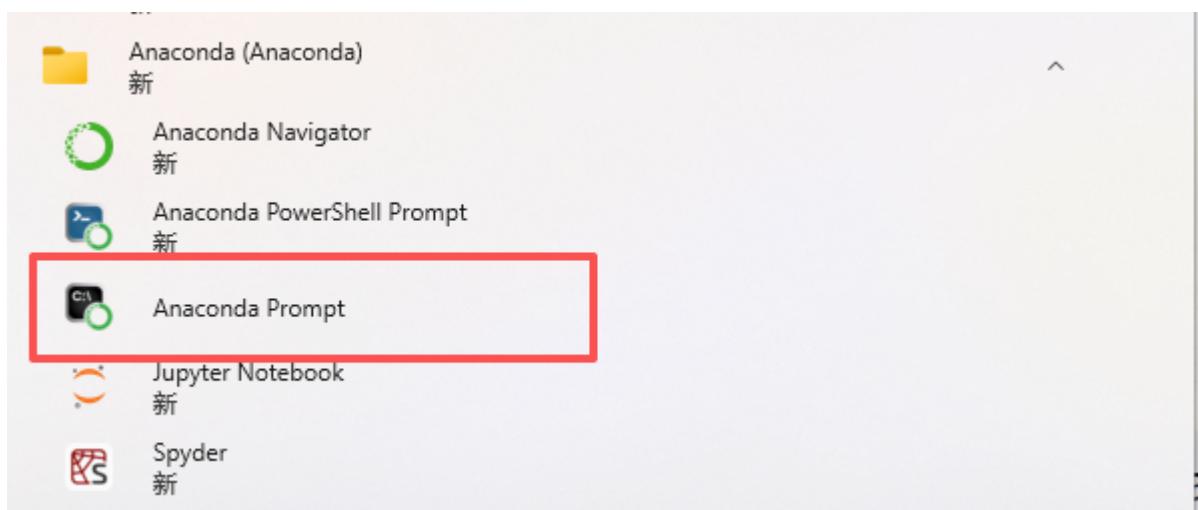
- 写在前面
- Step1 创建conda 虚拟环境
- 使用Jupyter Notebook连接虚拟环境
  - 前置设置
    - 配置Jupyter Notebook默认打开路径

## 写在前面

使用IDE连接conda虚拟环境之前完成步骤[Anaconda安装](#)并完成环境变量配置，国内镜像源配置与安装包存放位置更改

## Step1 创建conda 虚拟环境

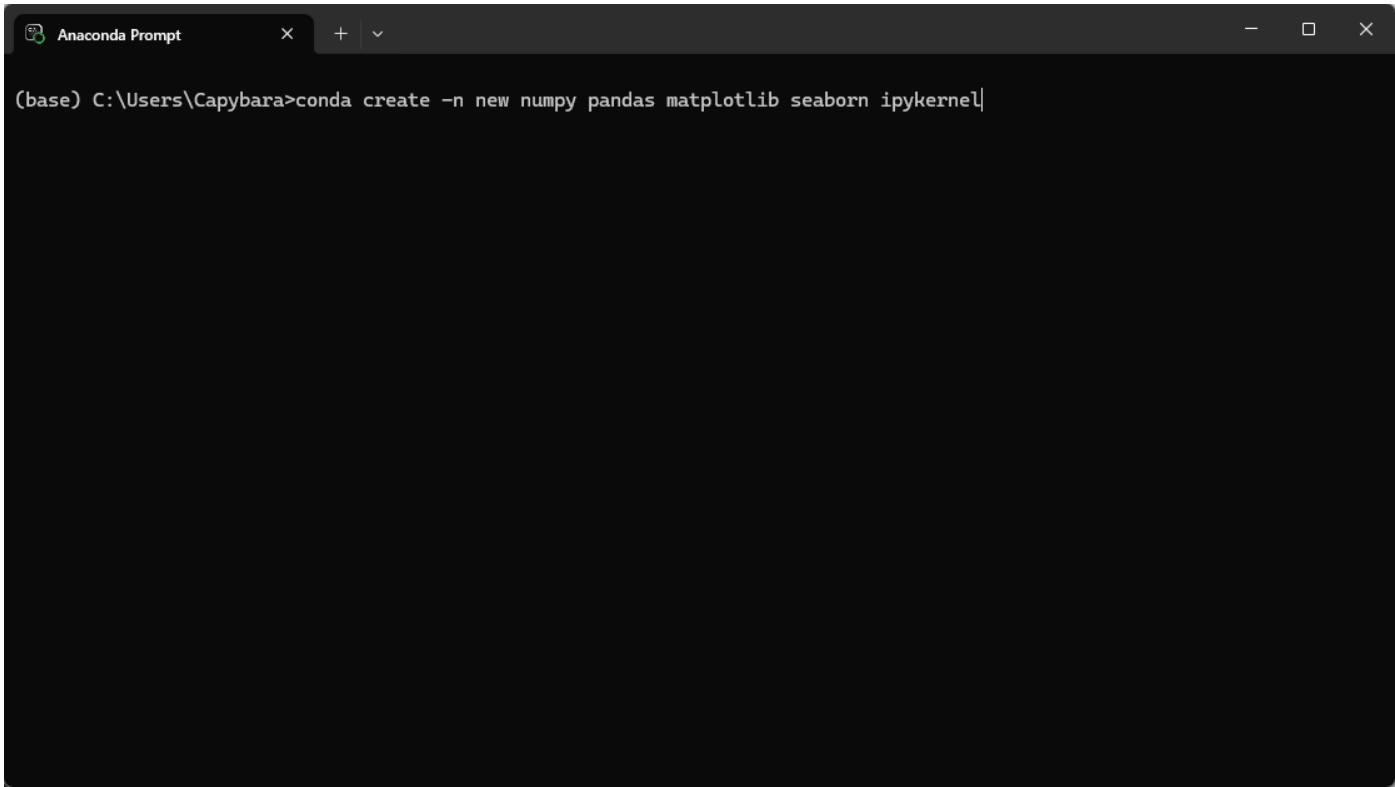
打开Anaconda Prompt



在Anaconda Prompt 或运行过conda init命令的Cmd或Powershell终端中执行（回车Enter执行）

即在 **(base)C:\User\用户名** 这个提示符下运行命令（MacOS同理，只要在这个提示符下运行即可）

```
conda activate -n 此处写自定义环境名称 numpy pandas matplotlib seaborn ipykernel
```



(base) C:\Users\Capybara>conda create -n new numpy pandas matplotlib seaborn ipykernel

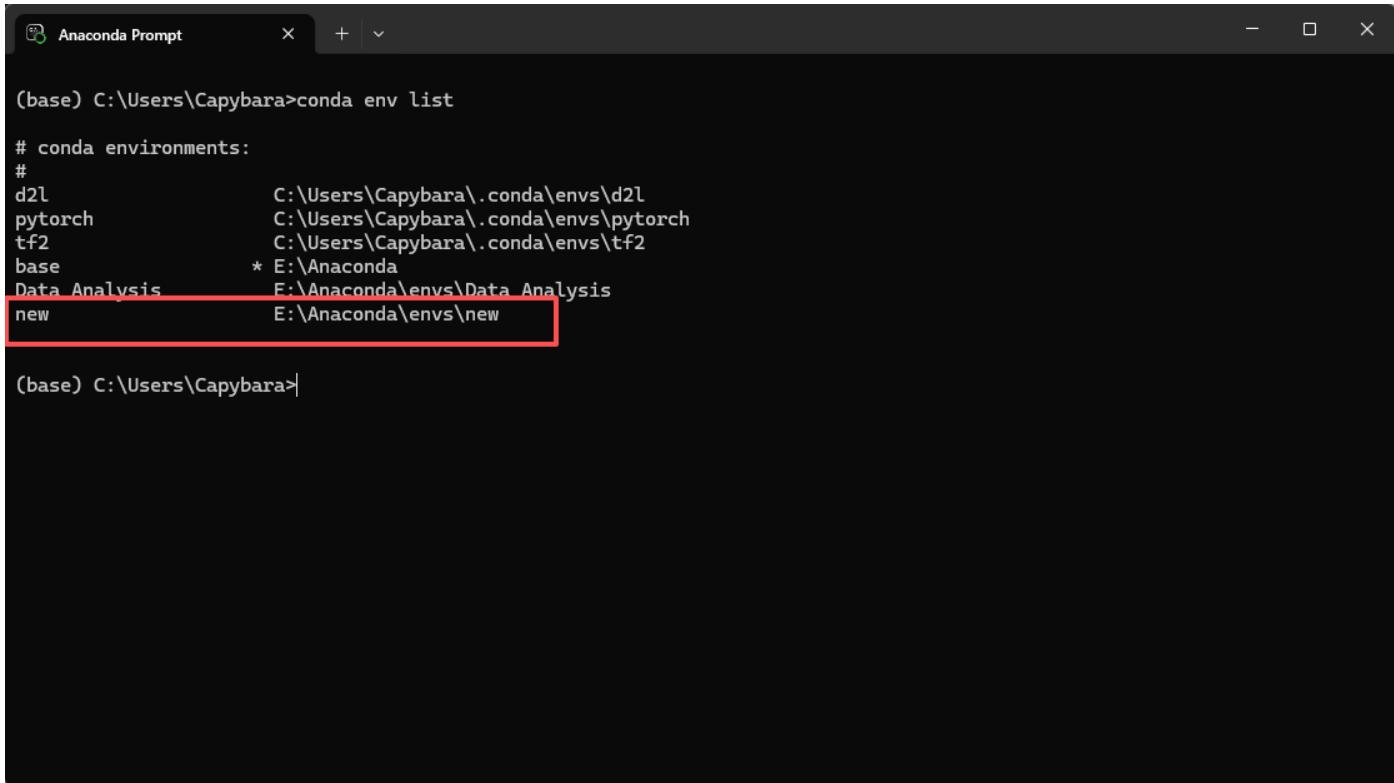
输入y，自动安装（此处可能花费一段时间请耐心等待），等再次出现提示符

(base)C:\User\用户名 新环境即创建成功

在 (base)C:\User\用户名 运行命令

```
conda env list
```

结果出现刚才创建的conda虚拟环境即为创建成功，该环境中预装了numpy pandas matplotlib seaborn ipykernel这些包



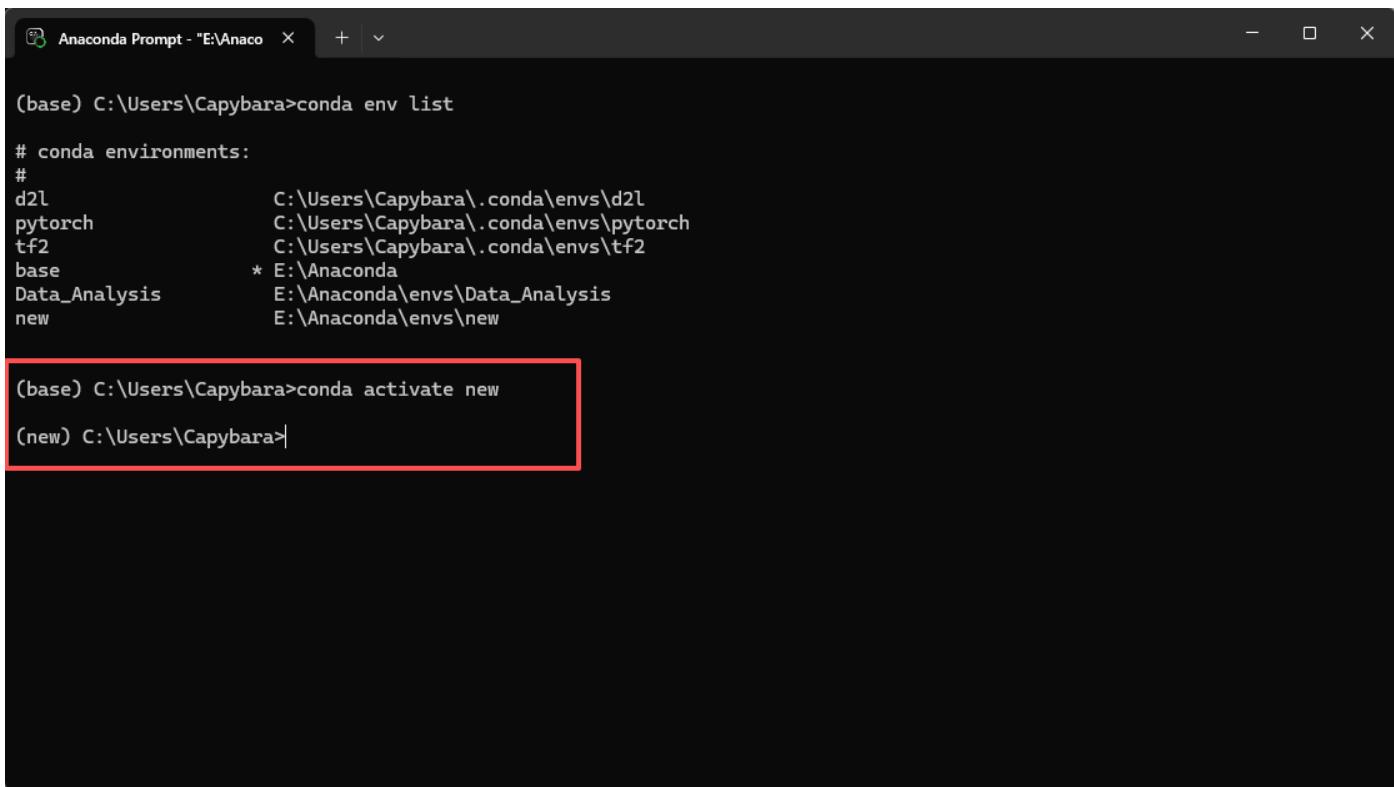
```
(base) C:\Users\Capybara>conda env list

# conda environments:
#
d2l          C:\Users\Capybara\.conda\envs\d2l
pytorch      C:\Users\Capybara\.conda\envs\pytorch
tf2          C:\Users\Capybara\.conda\envs\tf2
base          * E:\Anaconda
Data_Analysis E:\Anaconda\envs\Data_Analysis
new           E:\Anaconda\envs\new
```

激活虚拟环境new(你自定义环境的名称)

```
conda activate new
```

看到提示符变为 (new)C:\User\用户名 即切换环境成功



```
(base) C:\Users\Capybara>conda env list

# conda environments:
#
d2l          C:\Users\Capybara\.conda\envs\d2l
pytorch      C:\Users\Capybara\.conda\envs\pytorch
tf2          C:\Users\Capybara\.conda\envs\tf2
base          * E:\Anaconda
Data_Analysis E:\Anaconda\envs\Data_Analysis
new           E:\Anaconda\envs\new

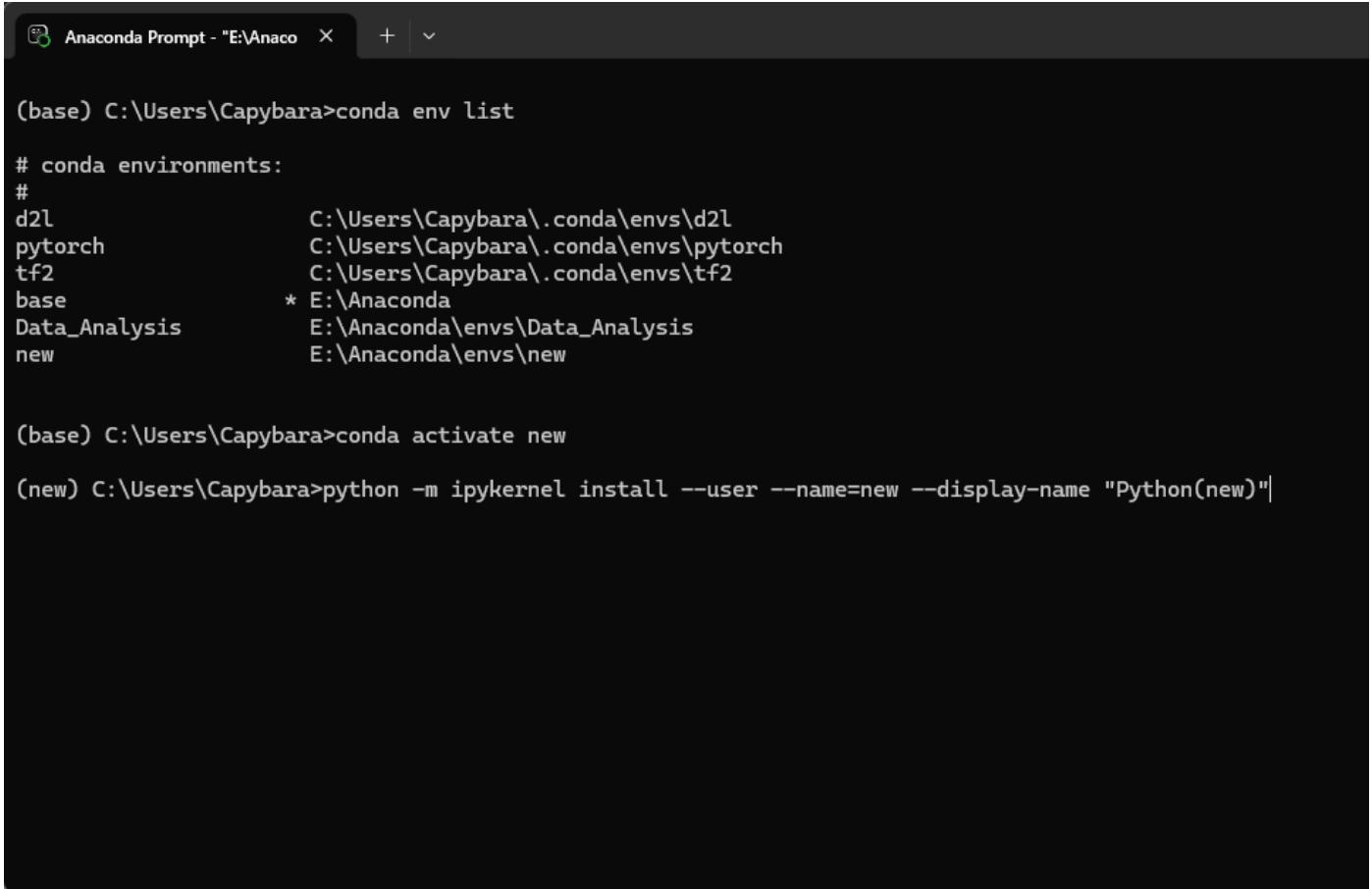
(base) C:\Users\Capybara>conda activate new

(new) C:\Users\Capybara>
```

注册notebook 内核

在 (new)C:\User\用户名 下执行命令(注意将new替换成你自定义环境名称)

```
python -m ipykernel install --user --name=new --display-name "Python (new)"
```



Anaconda Prompt - "E:\Anaconda" + ▾

```
(base) C:\Users\Capybara>conda env list
# conda environments:
#
d2l                  C:\Users\Capybara\.conda\envs\d2l
pytorch               C:\Users\Capybara\.conda\envs\pytorch
tf2                  C:\Users\Capybara\.conda\envs\tf2
base                 * E:\Anaconda
Data_Analysis         E:\Anaconda\envs\Data_Analysis
new                  E:\Anaconda\envs\new

(base) C:\Users\Capybara>conda activate new
(new) C:\Users\Capybara>python -m ipykernel install --user --name=new --display-name "Python(new)"
```

## 使用Jupyter Notebook连接虚拟环境

### 前置设置

### 配置Jupyter Notebook默认打开路径

Step1:新建专门存放代码文件的文件夹,记住此路径

Step2: 在 base 环境下运行命令

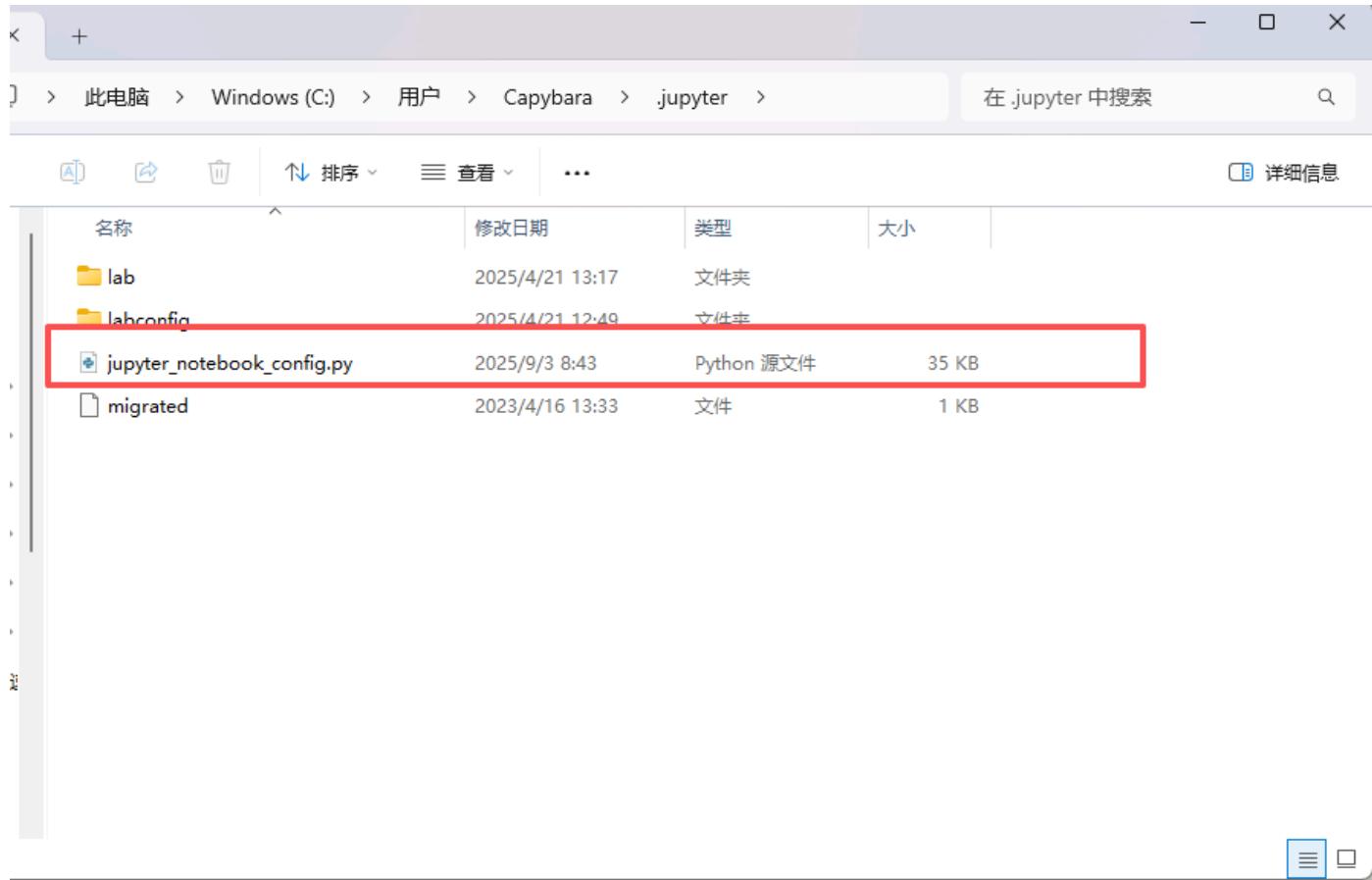
```
jupyter notebook --generate-config
```

该命令返回结果

```
Writing default config to: C:\Users\fireapproval\.jupyter\jupyter_notebook_config.py
```

Step3: 使用记事本打开该文件，这个文件一般在家目录（C:\User\用户名）的.jupyter文件夹下

(MacOS此文件夹为隐藏状态 使用cd [用户家目录]/.jupyter 进入)



按下Ctrl + F 输入并找到 `root_dir`，删去前面的#和空格并在后面的字符串中输入你创建的代码存放目录

(MacOS 同理)

```
# (env: JUPYTER_PORT_RETRY)
# Default: 50
# c.ServerApp.port_retry

## Preferred starting directory to use for notebooks and kernels.
# ServerApp.preferred_dir is deprecated in jupyter-server 2.0. Use
# FileContentsManager.preferred_dir instead
# Default: ""
# c.ServerApp.preferred_dir = ""

## DISABLED: use %pylab or %matplotlib in the notebook to enable matplotlib.
# Default: 'disabled'
# c.ServerApp.pylab = 'disabled'

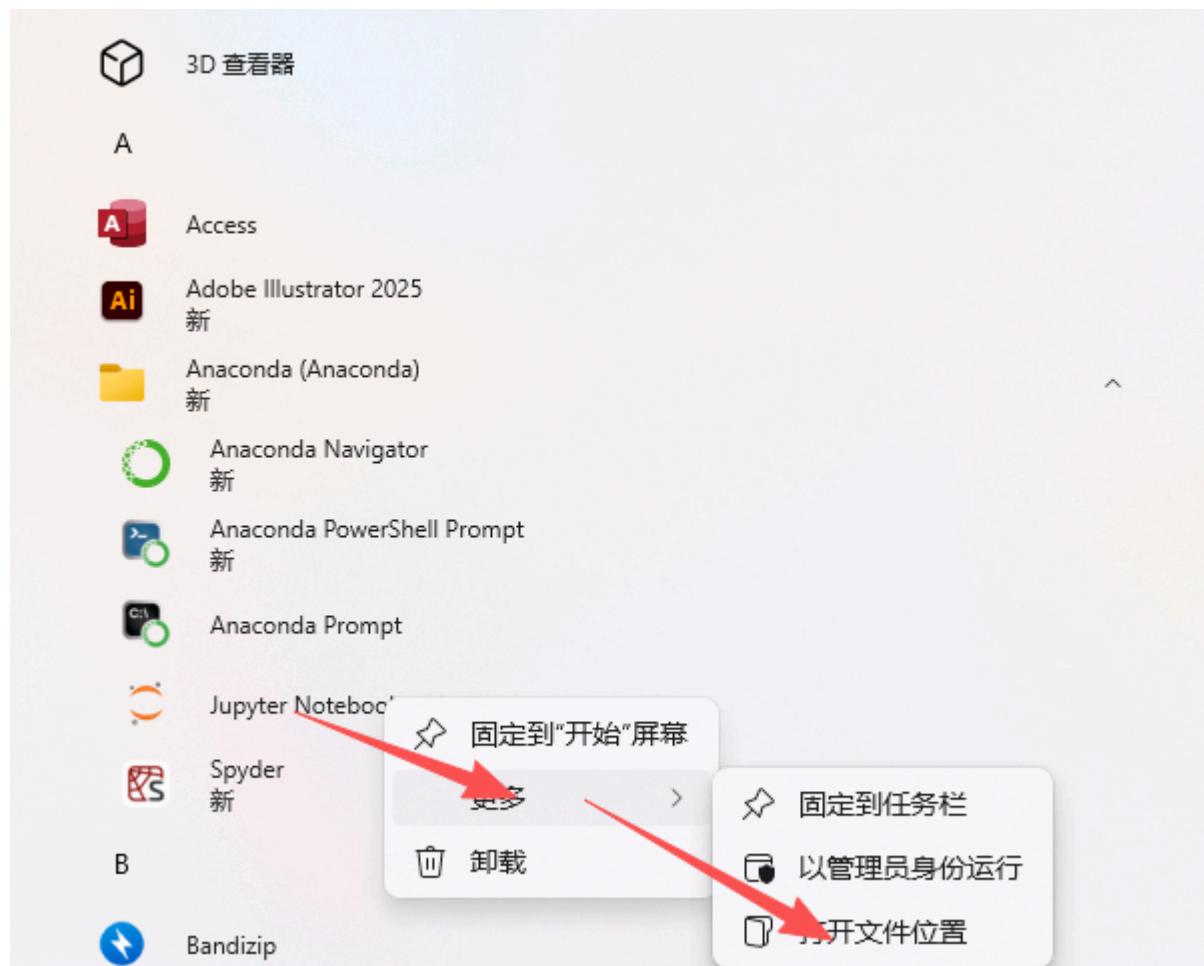
## If True, display controls to shut down the Jupyter server, such as menu items
# or buttons.
# Default: True
# c.ServerApp.quit_button = True

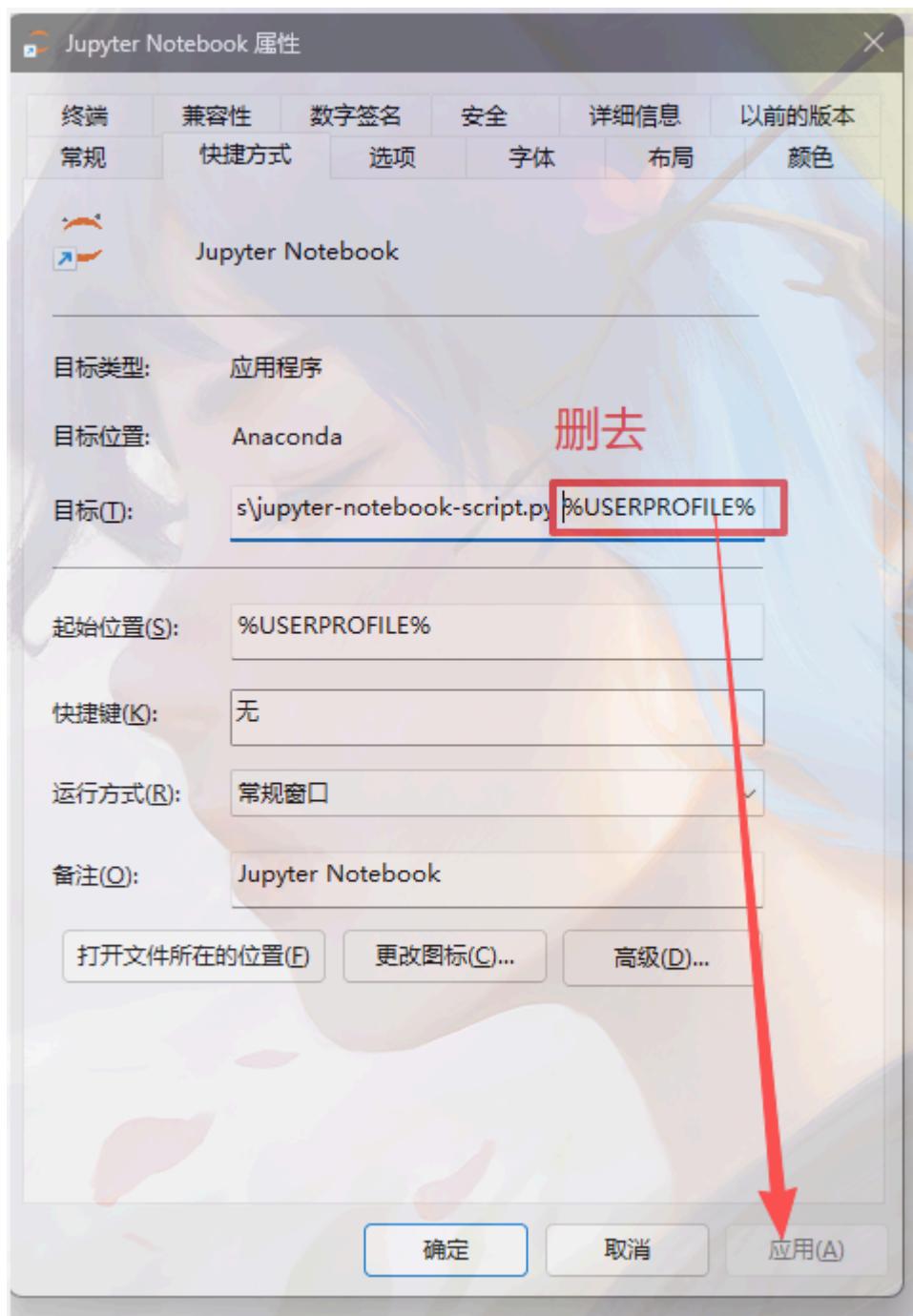
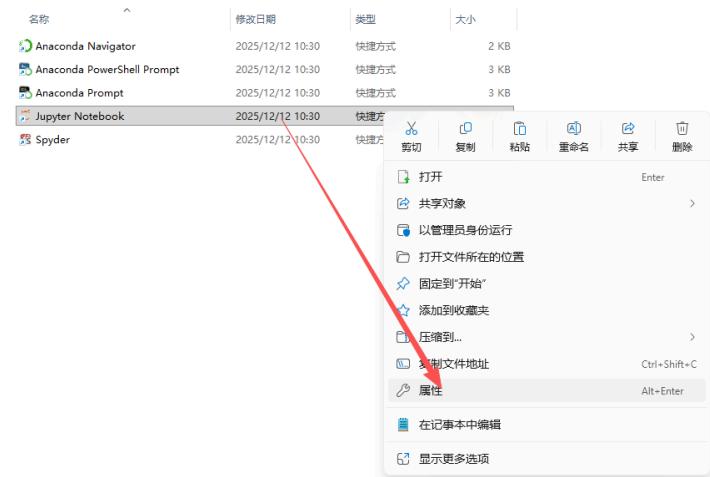
## DEPRECATED. Use ZMQChannelsWebsocketConnection.rate_limit_window
# Default: 0.0
# c.ServerApp.rate_limit_window = 0.0

## Reraise exceptions encountered loading server extensions?
# Default: False
# c.ServerApp.reraise_server_extension_failures = False

## The directory to use for notebooks and kernels
# Default: ""
c.ServerApp.root_dir = 'E:\Jupyter'
```

删去快捷方式的"%USERPROFILE%"





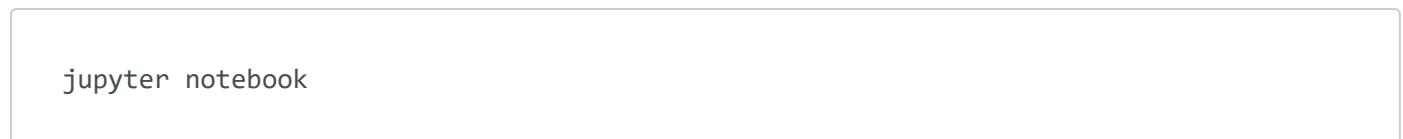
# 打开jupyter notebook 设置生效

打开方式2：

在终端中打开jupyter notebook(MacOS 适用)

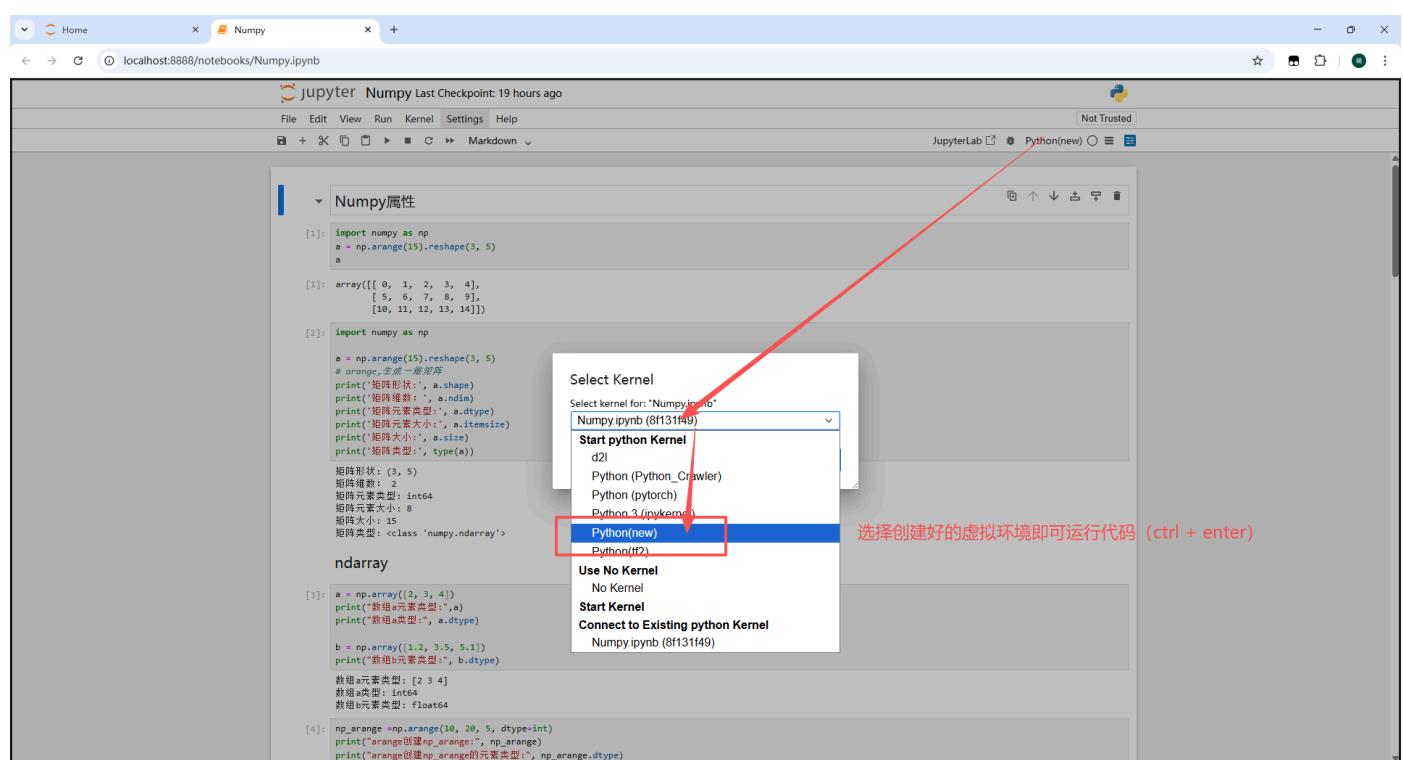
在 (base)C:\User\用户名 下运行

(注意使用 `conda deactivate` 命令切换回 `base` 环境，不要使用新创建虚拟环境，新创建虚拟环境没安装jupyter notebook)



基本使用：

切换内核：



快捷键预览：

jupyter Numpy Last Checkpoint: 19 hours ago

File Edit View Run Kernel Settings Help

About Jupyter Notebook

Show Keyboard Shortcuts... Ctrl+Shift+H

Launch Jupyter Notebook File Browser

Jupyter Reference

JupyterLab FAQ

JupyterLab Reference

Markdown Reference

About the Python(new) Kernel

Python Reference

IPython Reference

NumPy Reference

SciPy Reference

Matplotlib Reference

Sympy Reference

pandas Reference

About Jupyter

Markdown Reference

Documentation

```
[1]: import numpy as np
a = np.arange(15).reshape(3, 5)
a

[1]: array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])

[2]: import numpy as np

a = np.arange(15).reshape(3, 5)
# orange,生成一维矩阵
print("矩阵形状:", a.shape)
print("矩阵元素数:", a.size)
print("矩阵元素类型:", a.dtype)
print("矩阵元素大小:", a.itemsize)
print("矩阵大小:", a.size)
print("矩阵类型:", type(a))

矩阵形状: (3, 5)
矩阵元素数: 15
矩阵元素类型: int64
矩阵元素大小: 8
矩阵大小: 15
矩阵类型: <class 'numpy.ndarray'>

ndarray

[3]: a = np.array([2, 3, 4])
print("数组元素类型:", a)
print("数组类型:", a.dtype)

b = np.array([1, 2, 3, 5, 6])
print("数组元素类型:", b, b.dtype)

数组元素类型: [2 3 4]
数组类型: int64
数组元素类型: float64

[4]: np.arange -np.arange(10, 20, 5, dtype=int)
print("arange创建np.arange:", np.arange)
print("arange创建np.arange的元素类型:", np.arange.dtype)
```

jupyter Numpy Last Checkpoint: 19 hours ago

File Edit View Run Kernel Settings Help

Not Trusted

Keyboard Shortcuts

Redo Ctrl + Shift + Z

Undo Ctrl + Z

Run Selected Cell Shift + Enter

Find Next Ctrl + G

Find Previous Ctrl + Shift + G

Find... Ctrl + F

Activate Next Tab Ctrl + Shift + J

Activate Next Tab Bar Ctrl + Shift + .

Activate Previous Tab Ctrl + Shift + L

Activate Previous Tab Bar Ctrl + Shift + ,

Close

```
[1]: import numpy as np
a = np.arange(15).reshape(3, 5)
a

[1]: array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])

[2]: import numpy as np

a = np.arange(15).reshape(3, 5)
# orange,生成一维矩阵
print("矩阵形状:", a.shape)
print("矩阵元素数:", a.size)
print("矩阵元素类型:", a.dtype)
print("矩阵元素大小:", a.itemsize)
print("矩阵大小:", a.size)
print("矩阵类型:", type(a))

矩阵形状: (3, 5)
矩阵元素数: 15
矩阵元素类型: int64
矩阵元素大小: 8
矩阵大小: 15
矩阵类型: <class 'numpy.ndarray'>

ndarray

[3]: a = np.array([2, 3, 4])
print("数组元素类型:", a)
print("数组类型:", a.dtype)

b = np.array([1, 2, 3, 5, 6])
print("数组元素类型:", b, b.dtype)

数组元素类型: [2 3 4]
数组类型: int64
数组元素类型: float64

[4]: np.arange -np.arange(10, 20, 5, dtype=int)
print("arange创建np.arange:", np.arange)
print("arange创建np.arange的元素类型:", np.arange.dtype)
```