

## 第三周学习笔记

Conda 与 Python 的关系：

Conda 是独立于 Python 的环境管理器，可管理不同版本的 Python 及其依赖。

Conda-Forge

社区驱动的 Conda 软件源，提供比官方渠道更丰富的包。

路径问题

每个 Conda 环境有独立的 Python 解释器和包路径，激活环境后自动切换。

## 任务目标

我们解决任何实际问题，基本上都是以 **项目** 为工作方式。所以我们首先应该了解如何 **创建** Python 项目，然后是如何 **使用** 她，最后是如何 **逐步地改进** 她。

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)
2. 使用 `conda info` 命令查看本机 Conda 的配置信息
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
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
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
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
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
8. 使用 `pip install` 命令往 Conda 环境里安装 Python 软件包，并验证版本
9. 根据 [文档](#) 配置 PyPI 清华镜像，加快 `pip install` 安装软件包的速度
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

```
连续循环嵌套.py • python • # 初始化 Conda (根据你的 shell 类型选择) Untitled-1 •
1 # 初始化 Conda (根据你的 shell 类型选择)
2 conda init bash # 适用于 Git Bash、bash
3 conda init zsh # 适用于 Zsh
4
5 # 重启终端后, 提示符前应显示 (base)
6 conda info # 查看 Conda 版本、环境路径等信息
7 conda env list # 查看所有环境 (带 * 号的是当前激活环境)
8 # 创建 Python 3.12 + requests 环境
9 conda create -n py312 python=3.12 requests
10
11 # 创建 Python 3.9 + pandas + statsmodels 环境
12 conda create -n py39 python=3.9 pandas statsmodels
13
14 # 激活环境
15 conda activate py312 # 切换到 py312 环境
16 conda activate py39 # 切换到 py39 环境
17
18 # 验证 Python 和包版本
19 python --version
20 python -c "import requests; print(requests.__version__)"
21 python -c "import pandas; print(pandas.__version__)"
22 # 查看当前环境的所有包
23 conda list
24
25 # 安装额外包 (以 numpy 为例)
26 conda install numpy=1.24 # 指定版本
27 conda install matplotlib # 不指定版本则安装最新
28
29 # 验证安装
30 python -c "import numpy; print(numpy.__version__)"
31 # 添加清华镜像源
32 conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
33 conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free
34 conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
35 conda config --set show_channel_urls yes
```

Do you want to install from Microsoft

```
39 # 临时使用清华镜像
40 pip install -i https://pypi.tuna.tsinghua.edu.cn/simple some-package
41
42 # 永久配置
43 pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
44 # 导出环境配置 (在激活环境后执行)
45 conda env export > environment.yml
46
47 # 删除环境
48 conda env remove -n py312
49
50 # 从配置文件重建环境
51 conda env create -f environment.yml
52 # 创建项目目录
53 mkdir my_project && cd my_project
54
55 # 创建项目专用环境
56 conda create -n my_project python=3.11
57
58 # 激活环境并安装依赖
59 conda activate my_project
60 conda install numpy pandas matplotlib
61
62 # 在 VS Code 中打开项目
63 code .import pandas as pd
64 import numpy as np
65
66 def main():
67     # 创建示例数据
68     data = {
69         'Name': ['Alice', 'Bob', 'Charlie'],
70         'Age': [25, 30, 35]
71     }
72     'Score': np.random.randint(0, 100, 3)
73
74     # 创建 DataFrame
75     df = pd.DataFrame(data)
```

```

66 def main():
67     # 创建示例数据
68     data = {
69         'Name': ['Alice', 'Bob', 'Charlie'],
70         'Age': [25, 30, 35]
71         'Score': np.random.randint(0, 100, 3)
72     }
73
74     # 创建 DataFrame
75     df = pd.DataFrame(data)
76
77     # 打印数据
78     print("数据统计: ")
79     print(df.describe())
80
81 if __name__ == "__main__":
82     main()
83     # 在已激活的环境中执行
84 python main.py

```

1.问题：高强老师第三周第五个视频 34:15，在 git bash 中输入 python main.py 无法出现 “Hello, conda”

2.源代码及图片：

```

(base) lyj29@LAPTOP-HRH4STI0 MINGW64 ~/repo/myproject
$ conda activate myproject
(myproject)
lyj29@LAPTOP-HRH4STI0 MINGW64 ~/repo/myproject
$ python main.py
(myproject)

```

3.出现原因：这个错误是由于 Windows PowerShell 的执行策略 (Execution Policy) 限制了脚本的运行。默认情况下，Windows 系统可能禁止运行脚本，以防止恶意脚本的执行。

```

C: > Users > 86189 > Desktop > 连续循环嵌套.py
1  √ for i in range(1,10):
2  √     for j in range(1,i+1):
3      |     print("{}*{} = {}".format(i,j,i*j),end = "\t")
4      |     print()
5

```





# Anaconda 软件仓库

线路选择 自动 ▾

☒ 是否使用 HTTPS

☐ 是否使用 sudo

## 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`



## 第三方源列表

您可以遵循上述配置文件中的格式添加第三方源（推荐），或者通过以下命令添加第三方源：

第三方源 auto ▾

```
conda config --set custom_channels.auto https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/
```

## Miniconda

Miniconda 是一个 Anaconda 的轻量级替代，默认只包含了 python 和 conda，但是可以通过 pip 和 conda 来安装所需要的包。

Miniconda 安装包可以到以下链接下载。

```
https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/
```