

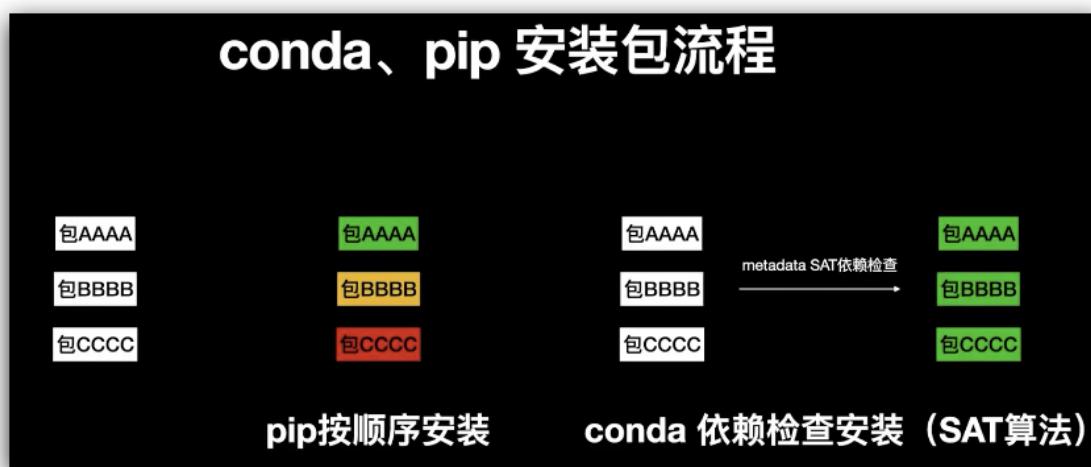
快速搞定 Python 库和环境

Conda + pip 主要好处:

- 使用简单
- 支持主流操作系统
- 支持虚拟环境，快速切换多个版本 Python、包
- 第三方库安装成功率高

	Conda	Pip
包格式	二进制 .tar.bz2 .conda	.Wheel 或 源代码
支持语言	Python、C、C++、R等	Python
Python包	✓	✓
虚拟环境	✓ 支持	✗ 不支持，需配合Virtualenv等
依赖检查	✓	✗
包源	Anaconda、cloud	PyPI

conda 会进行依赖检查，检查不同的包之间的依赖关系，避免包之间的互相影响。



但是 conda 中包的数量远不及 pip 包的数量，所以对于一些冷门的资源，可能还是需要使用 pip。

Python 运行环境：用 Conda 创建虚拟环境

Python 包安装三板斧：Conda>pip>编译安装

1. 下载安装 Conda 软件

<https://docs.conda.io/en/latest/miniconda.html>

macOS系统：双击打开 `Miniconda3-latest-MacOSX-x86_64.pkg` 按流程安装

macOS系统（M1）：终端（terminal）中输入 `bash Miniconda3-latest-MacOSX-arm64.sh`

Windows系统：双击打开 `Miniconda3-latest-Windows-x86_64.exe`（64位）安装

2. 创建 Python 虚拟环境

列出所有环境：`conda env list`

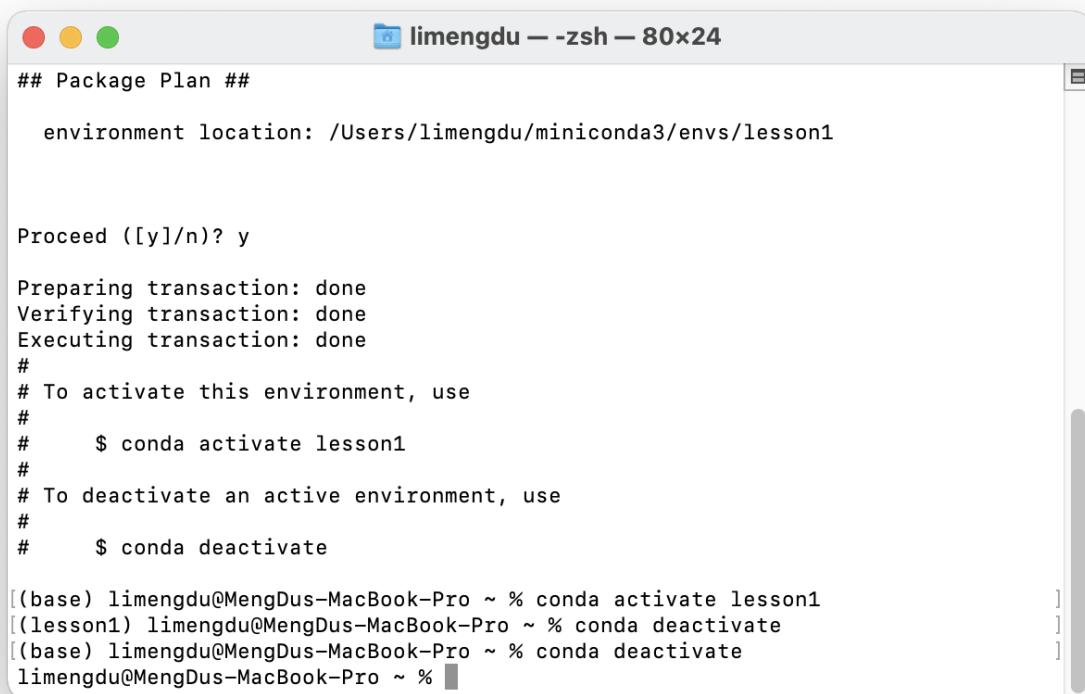
创建环境：`conda creat --name 环境名称英文`

进入环境：`conda activate 环境名称`

退出环境：`conda deactivate`

删除环境：`conda remove --name 环境名称 --all`

创建指定 Python 版本环境：`conda create --name 环境名称 python=3.7`



```
## Package Plan ##

environment location: /Users/limengdu/miniconda3/envs/lesson1

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate lesson1
#
# To deactivate an active environment, use
#
#     $ conda deactivate

[(base) limengdu@MengDus-MacBook-Pro ~ % conda activate lesson1
[(lesson1) limengdu@MengDus-MacBook-Pro ~ % conda deactivate
[(base) limengdu@MengDus-MacBook-Pro ~ % conda deactivate
limengdu@MengDus-MacBook-Pro ~ % ]]
```

3. Conda/pip 安装第三方包

国内加速镜像

windows 及其他系统 conda 换源方法:

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

TUNA 还提供了 Anaconda 仓库与第三方源（conda-forge、msys2、pytorch 等，[查看完整列表](#)）的镜像，各系统都可以通过修改用户目录下的 .condarc 文件。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。

注：由于更新过快难以同步，我们不同步 pytorch-nightly, pytorch-nightly-cpu, ignite-nightly 这三个包。

```
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
  msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  pytorch-lts: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

即可添加 Anaconda Python 免费仓库。

运行 `conda clean -i` 清除索引缓存，保证用的是镜像站提供的索引。

运行 `conda create -n myenv numpy` 测试一下吧。

windows 及其他系统 pip 换源方法：

Windows

- 临时使用：`pip install -i https://pypi.tuna.tsinghua.edu.cn/simple {包名}`
- 永久使用：

第一步：在 `C:\Users\Administrator` 目录下创建 `pip` 文件夹

第二步：在第一步创建的文件夹下 (`C:\Users\Administrator\pip`) 创建 `pip.ini` 文件

第三步：记事本编辑保存 `pip.ini` 文件内容为以下部分：

```
[global]
index-url = https://pypi.tuna.tsinghua.edu.cn/simple/
[install]
trusted-host = pypi.tuna.tsinghua.edu.cn
```

MacOS 系统

- 临时使用: pip install -i <https://pypi.tuna.tsinghua.edu.cn/simple> {包名}
- 永久使用:

执行以下语句

```
cd ~

mkdir .pip

cd .pip

nano pip.conf
```

pip.conf 写入

```
[global]

index-url = https://pypi.tuna.tsinghua.edu.cn/simple/

[install]

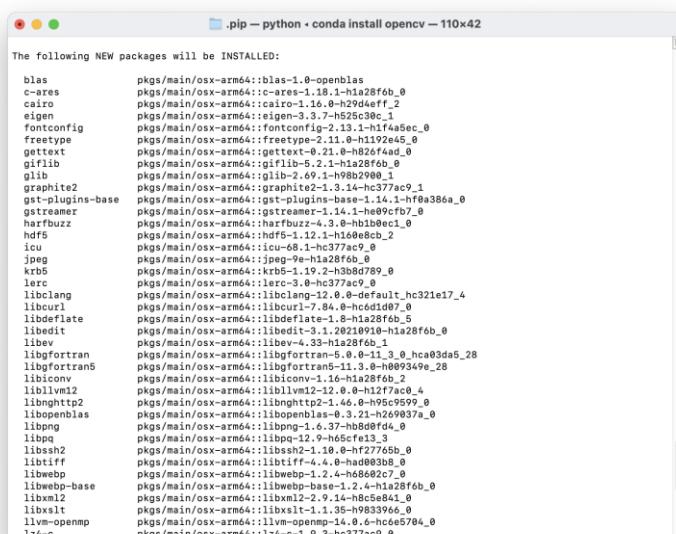
trusted-host = pypi.tuna.tsinghua.edu.cn
```

保存 pip.conf

安装 OpenCV

```
conda install opencv
```

⚠ 注意: 此时如果你没有进入 conda 的环境, 那么就是在 base 里面安装 opencv!



```
conda install mediapipe
```

```
limengdu@MengDus-MacBook-Pro .pip % conda install mediapipe
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.

PackagesNotFoundError: The following packages are not available from current channels:

- mediapipe

Current channels:

- https://repo.anaconda.com/pkgs/main/osx-arm64
- https://repo.anaconda.com/pkgs/main/noarch
- https://repo.anaconda.com/pkgs/r/osx-arm64
- https://repo.anaconda.com/pkgs/r/noarch

To search for alternate channels that may provide the conda package you're
looking for, navigate to

    https://anaconda.org

and use the search bar at the top of the page.
```

这就属于是比较小众冷门的包，需要使用第二种方式 pip 安装。

```
python -m pip install mediapipe
```

⚠ 注意：此时如果你没有进入 conda 的环境，那么就是在 base 里面安装 mediapipe！

M1 芯片的苹果电脑，安装 mediapipe：

```
pip install mediapipe-silicon
```

要求：

Python: version 3.8 - 3.11

PIP: version 20.3+

4. 运行 Demo 程序

```
"""
```

```
演示 Demo
```

```
"""
```

```
# 导入 opencv
```

```
import cv2
```

```
import numpy as np
```

```
import math

# 导入 mediapipe: https://google.github.io/mediapipe/solutions/hands

import mediapipe as mp


mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_hands = mp.solutions.hands


hands = mp_hands.Hands(
    model_complexity=0,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5)

# 读取视频流

cap = cv2.VideoCapture(0)

# 获取画面宽度、高度

width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))

height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

while True:

    ret, frame = cap.read()

    # 镜像

    frame = cv2.flip(frame, 1)

    frame.flags.writeable = False
```

```
frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

# 识别

results = hands.process(frame)

frame.flags.writeable = True

frame = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)

# 如果有结果

if results.multi_hand_landmarks:

    # 遍历双手

    for hand_landmarks in results.multi_hand_landmarks:

        mp_drawing.draw_landmarks(
            frame,
            hand_landmarks,
            mp_hands.HAND_CONNECTIONS,
            mp_drawing_styles.get_default_hand_landmarks_style(),
            mp_drawing_styles.get_default_hand_connections_style())

    # 显示画面

    cv2.imshow('demo', frame)

if cv2.waitKey(10) & 0xFF == ord('q'):

    break

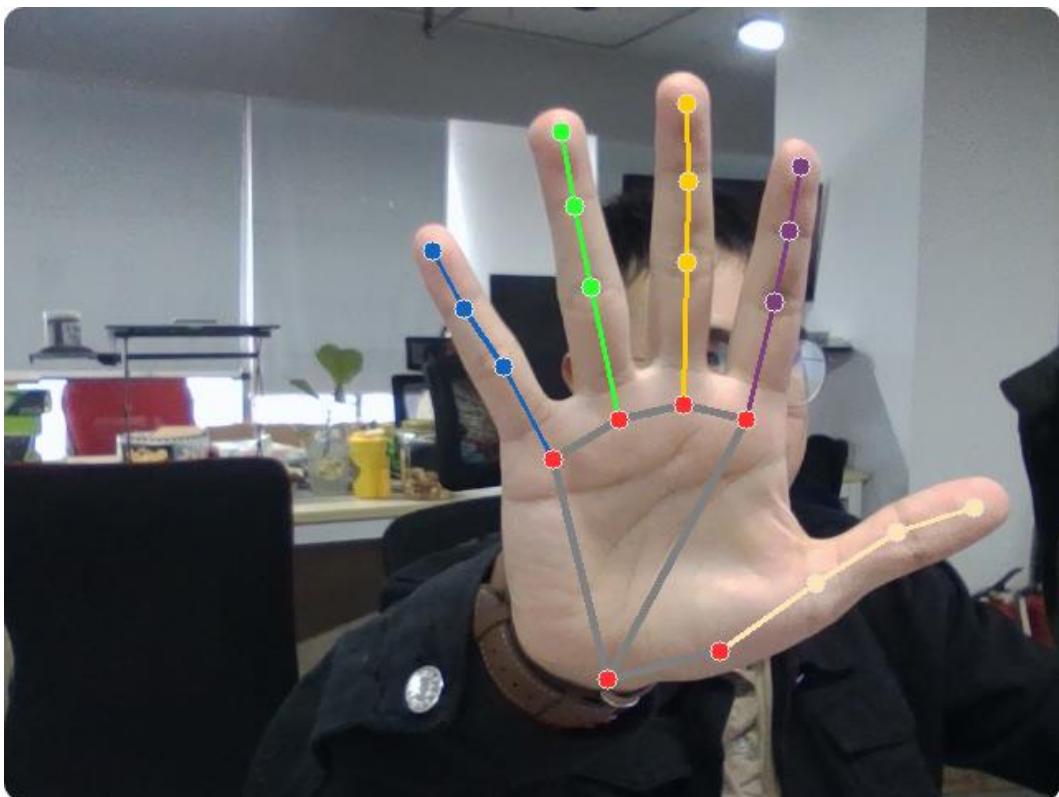
cap.release()

cv2.destroyAllWindows()
```

终端运行：

```
conda creat --name lesson1  
conda activate lesson1  
conda install opencv  
pip install mediapipe  
cd 文件夹路径  
python demo.py
```

效果展示：



思考：

Q：如何知道一个程序应该装哪些包？

A：解决办法一——报错缺什么去搜索引擎搜索需要安装什么包

解决办法二——看程序里面是否有提供说明文件

后续的课程内容请自行在自己的环境内进行。