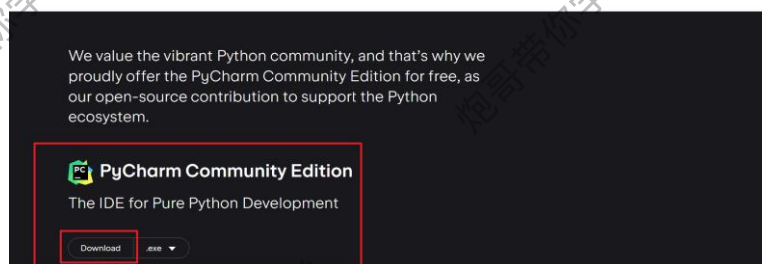
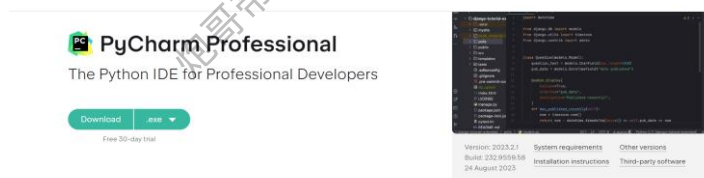


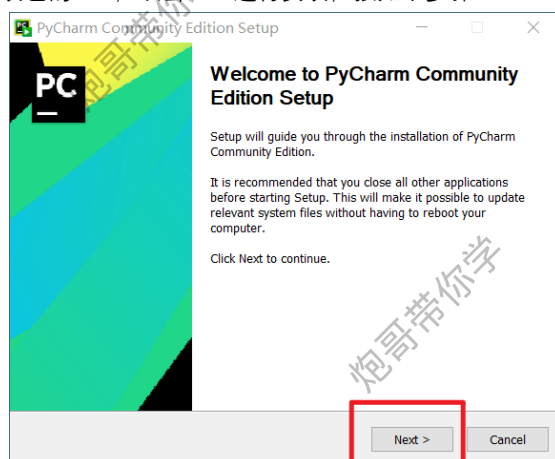
环境安装流程

1、Pycharm 安装

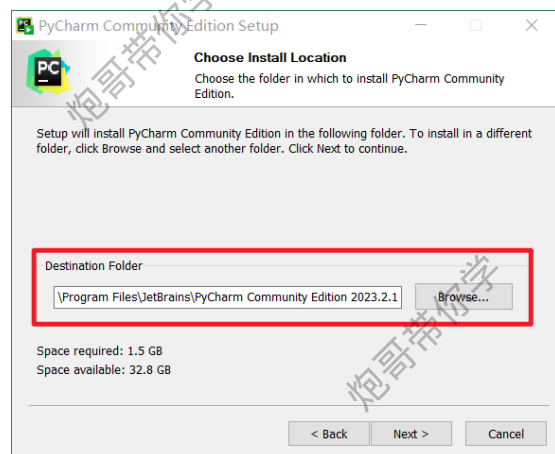
输入如下 pycharm 下载网址: <https://www.jetbrains.com/pycharm/download/?section=windows>
下载如下的 pycharm 的社区版版本, 该版本为免费版本。



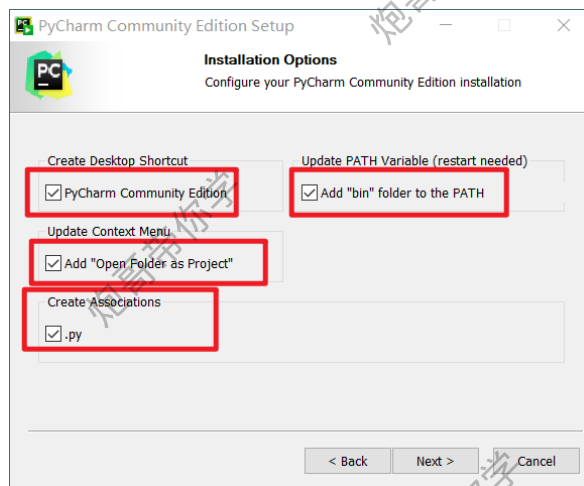
下载完成, 会有一个安装包的 exe, 双击 exe 进行安装, 按如下步骤



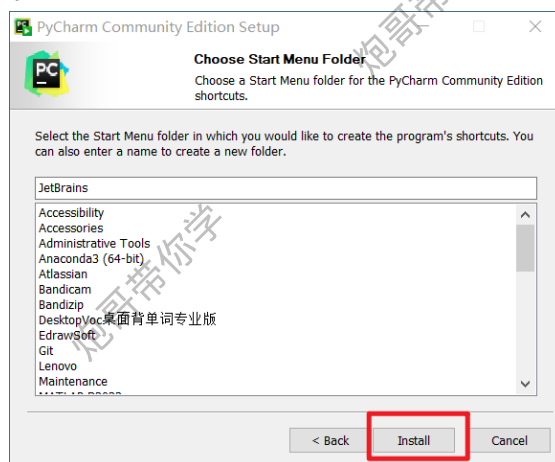
上述步骤会跳转到如图的步骤, 该步骤是安装路径, 默认路径是安装在 c 盘, 但是不建议放到 c 盘, 我是安装在 D 盘, 同时路径中不要有中文, 以免后续有各种奇怪的问题。



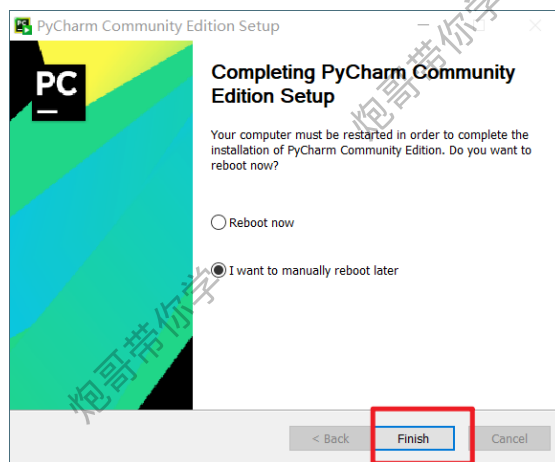
上述步骤后, 跳转到如下步骤, 将所有的框全部勾选上



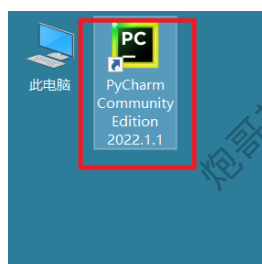
点击安装，开始安装即可



安装完毕，点击完成即可



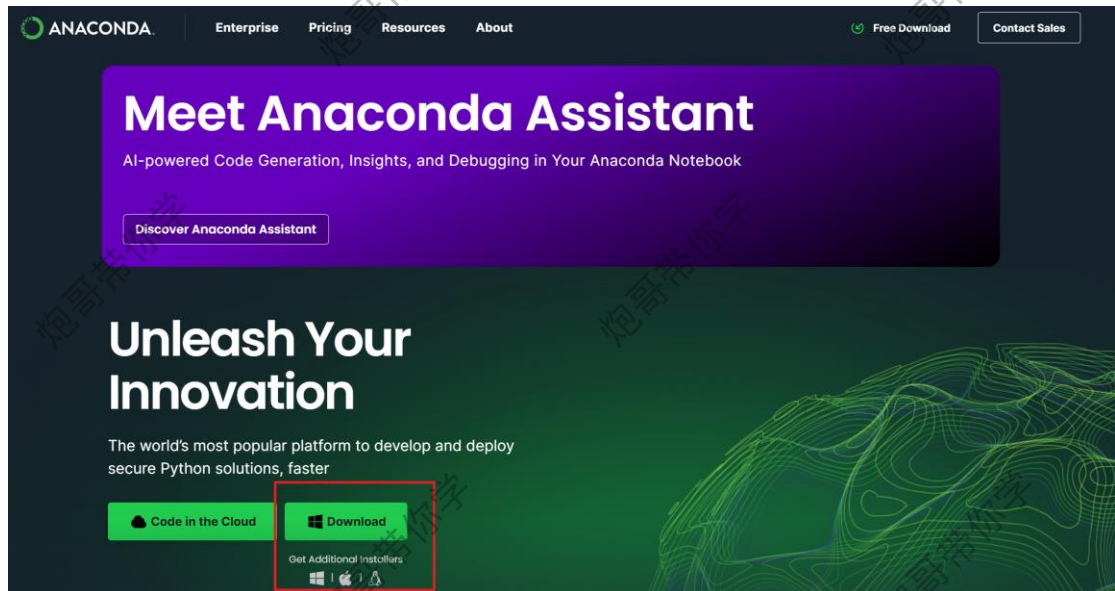
此时桌面会有一个 pycharm 的图标，表格该软件安装完成。



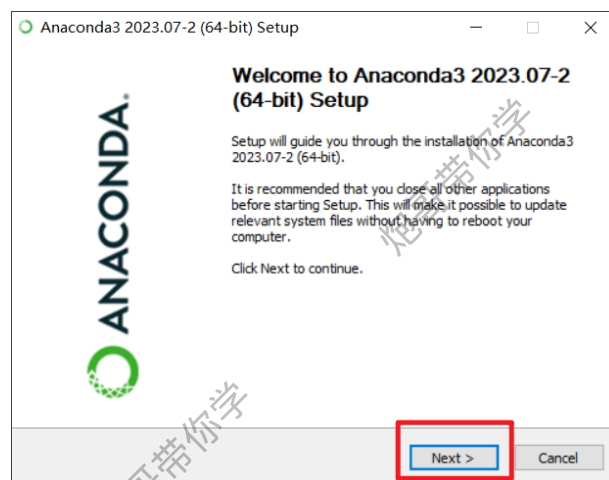
2、anaconda 安装与 Python 环境的创建

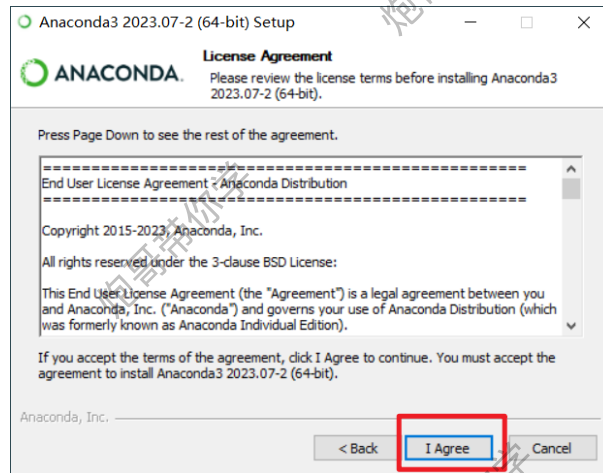
2.1 anaconda 安装

输入如下 anaconda 下载网址：<https://www.anaconda.com/>，打开的网址如下图所示（随着时间的流程官网页面也会更新）。点击如下红框的下载界面：

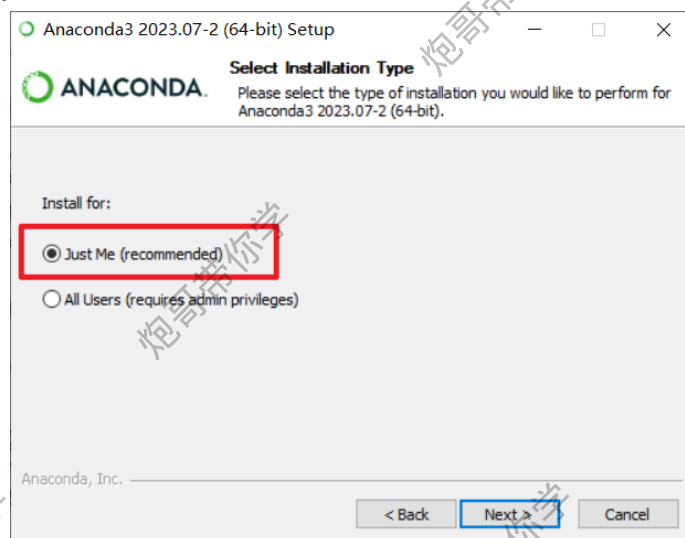


下载完成，会有一个安装包的 exe，双击 exe 进行安装，按如下步骤

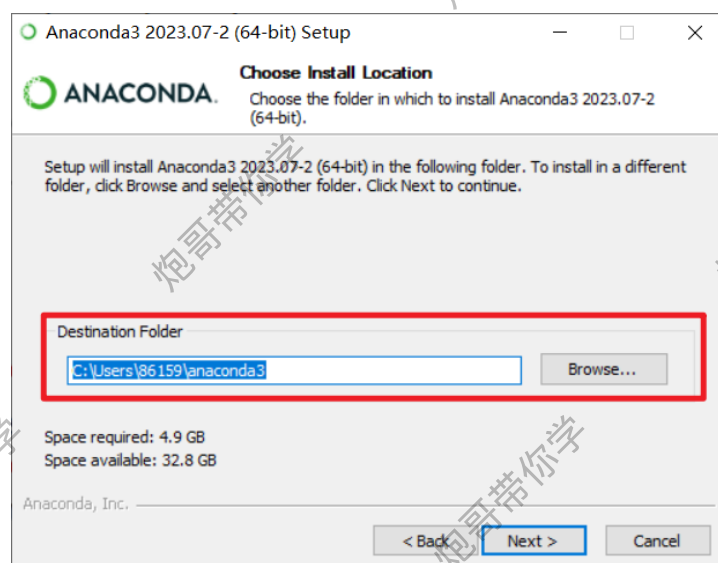




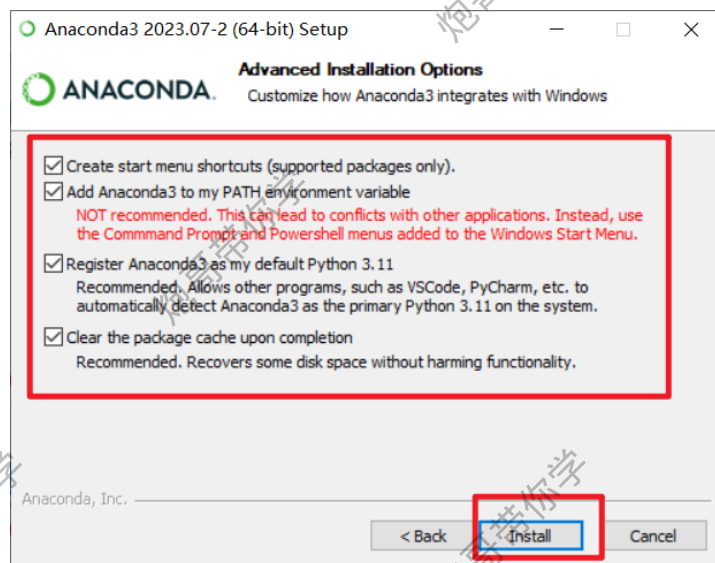
这里一定要选择 just me 否则在后续创建环境的时候会出各种各样的问题



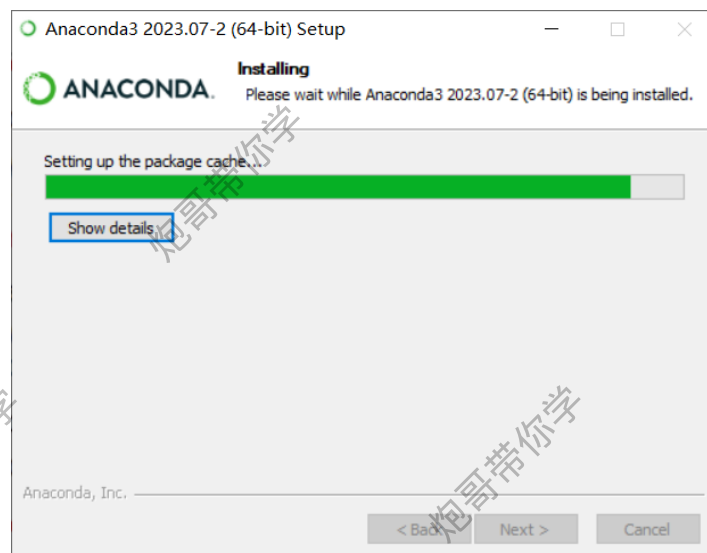
这里安装路径的选择，一定最好不要放在默认路径，默认路径在 C 盘，最好在其他盘中创建路径，并记住按路径。因为后续在安装各种环境的时候环境占用空间很大，放在 c 盘空间会不够。



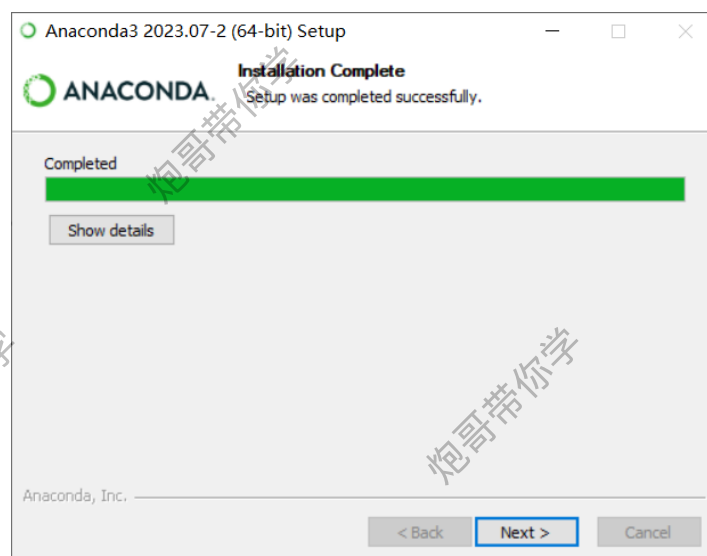
和我一样将这里的 √ 全部框上，点击安装

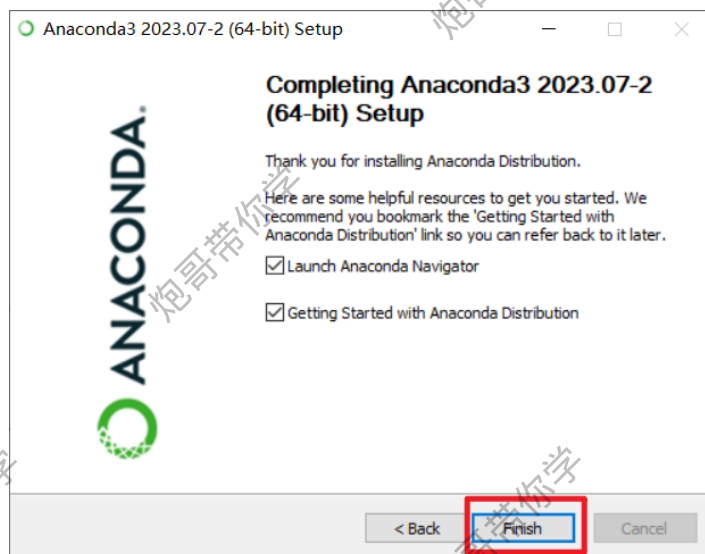


这里下载安装在最后一段可能需要一些时间，请安心等待

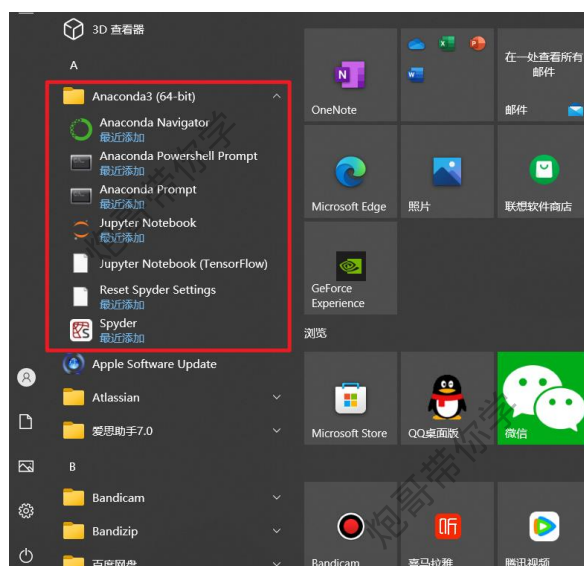


安装完毕点击完成即可



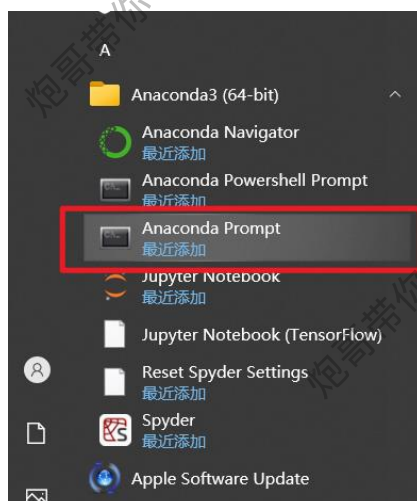


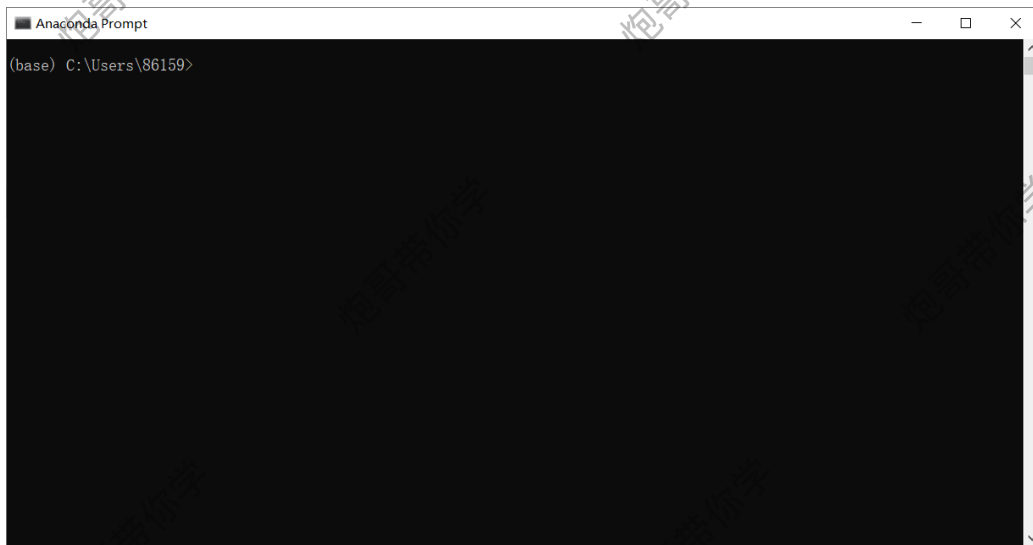
安装完毕以后，在电脑的开始键里面就有安装好的 anaconda



2.2、Anaconda 中 Python 环境的创建

点击图标 anaconda prompt，会弹出一个黑框





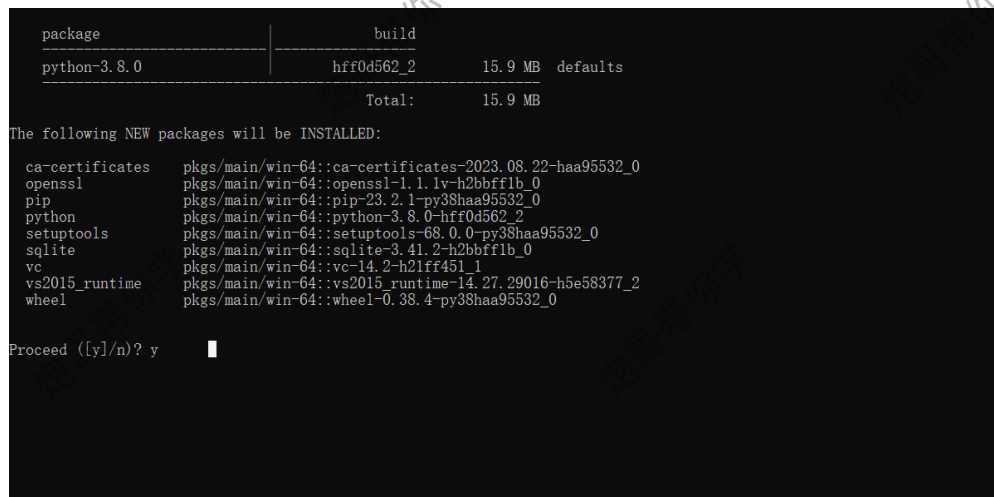
输入如下命令：

```
conda create -n pytorch python==3.8
```

上述的命令的意思是，创建一个名字为 pytorch 的环境，其中 Python 版本为 3.8，其中名字可以指定，只要不是中文都行，不一定是 pytorch，Python 的版本也可以，3.5、3.6、3.7 都可以。



会跳出如下界面，输入 y，表示下载对应的环境依赖包



所有的包安装完毕输入如下命令查看环境数量：

conda env list，可以看到现在环境中除了基础的 base 环境，还有刚刚创建的名字为 pytorch 的环境

```
(base) C:\Users\86159>conda env list
# conda environments:
#
base                * D:\code\anaconda
pytorch             D:\code\anaconda\envs\pytorch

(base) C:\Users\86159>_
```

输入命令，activate pytorch，进入到 pytorch 环境中

```
(base) C:\Users\86159>conda env list
# conda environments:
#
base                * D:\code\anaconda
pytorch             D:\code\anaconda\envs\pytorch

(base) C:\Users\86159>activate pytorch
(pytorch) C:\Users\86159>_
```

输入 conda list 命令，查看该环境下的所有依赖包，可以看到有一个 Python 版本为 3.8，到此 anaconda 中的 Python 环境就安装好了，我们有了对应版本的 Python 解释器可以进行 Python 代码的撰写了。

```
(base) C:\Users\86159>activate pytorch
(pytorch) C:\Users\86159>conda list
# packages in environment at D:\code\anaconda\envs\pytorch:
# Name                    Version            Build    Channel
ca-certificates           2023.08.22         haa95532_0 defaults
openssl                   1.1.1v             h2bbff1b_0 defaults
pip                       23.2.1             py38haa95532_0 defaults
python                    3.8.0              hff0d562_2 defaults
setuptools                68.0.0             py38haa95532_0 defaults
sqlite                    3.41.2             h2bbff1b_0 defaults
ve                         14.2               h21ff451_1 defaults
vs2015_runtime            14.27.29016        h5e58377_2 defaults
wheel                     0.38.4             py38haa95532_0 defaults

(pytorch) C:\Users\86159>
```

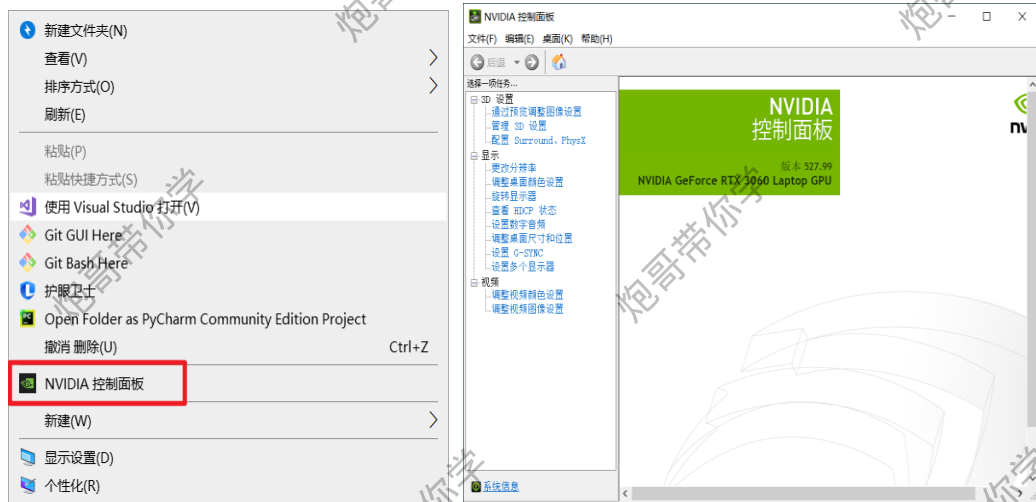

3、pytorch 与深度学习环境安装

3.1、NVIDIA 驱动安装与更新

按如下步骤找到自己显卡的型号，可以看到我的显卡是 3060



有显卡驱动的，可以直接在桌面右键，找到英伟达驱动控制面板打开就好了。



没有显卡驱动去英伟达驱动官网 <https://www.nvidia.cn/Download/index.aspx?lang=cn> 打开驱动官网找到适合自己设备的选项就行。

NVIDIA 驱动程序下载

在下方的下拉列表中进行选择，针对您的 NVIDIA 产品确定合适的驱动。

产品类型:	GeForce
产品系列:	GeForce RTX 30 Series (Notebooks)
产品家族:	GeForce RTX 3060 Laptop GPU
操作系统:	Windows 10 64-bit
下载类型:	Game Ready 驱动程序 (GRD)
语言:	Chinese (Simplified)

搜索

GeForce Game Ready 驱动程序

版本:	537.13 WHQL
发布日期:	2023.8.22
操作系统:	Windows 10 64-bit, Windows 11
语言:	Chinese (Simplified)
文件大小:	644.27 MB

下载

下载安装完以后，我们查看支持 cuda 驱动的支持最高版本 cuda toolkit。

按下 win+R 组合键，打开 cmd 命令窗口。输入如下的命令：

nvidia-smi

得到如下的结果，可以看到我的显卡驱动支持的最高版本 cuda toolkit 是 12.0，版本是向下兼容的，因此只要安装向下兼容的 cuda toolkit 版本即可。

Microsoft Windows [版本 10.0.19045.3324]
(c) Microsoft Corporation。保留所有权利。
C:\Users\86159>nvidia-smi
Wed Sep 6 22:51:54 2023

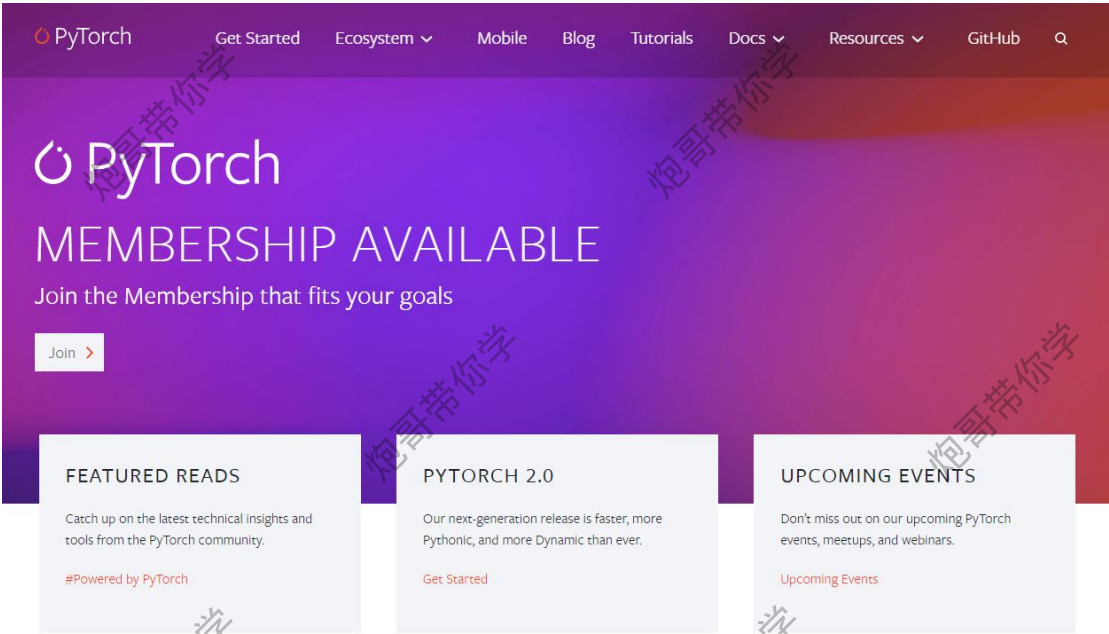
NVIDIA-SMI 527.99				Driver Version: 527.99		CUDA Version: 12.0		
GPU	Name	TCC/WDDM	Bus-Id	Disp. A	Memory-Usage	Volatile GPU-Util	Uncorr. Compute M.	ECC MIG M.
Fan	Temp	Perf Pwr:Usage/Cap						
0	NVIDIA GeForce ...	WDDM	00000000:01:00.0	On				
N/A	55C	P8 16W / 129W	1058MiB / 6144MiB			10%	Default	N/A
								N/A

Processes:

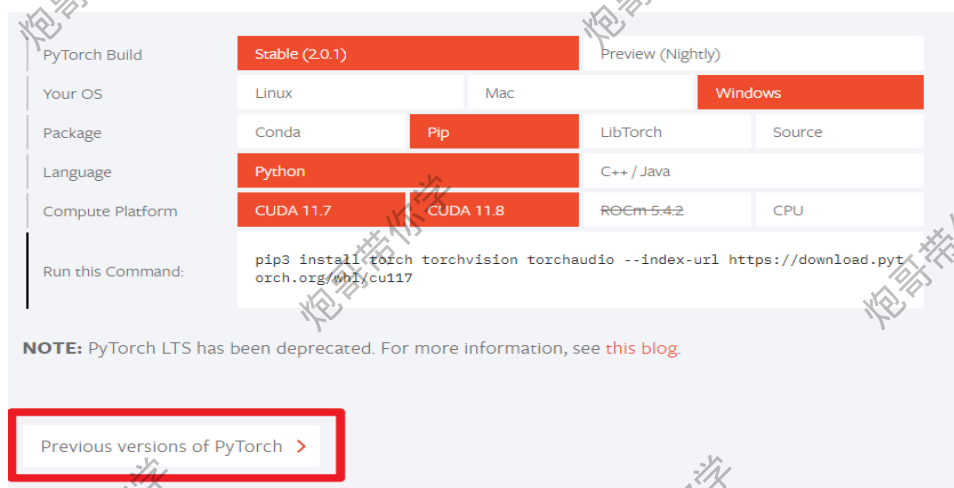
GPU	GI ID	CI ID	PID	Type	Process name	GPU Memory Usage
0	N/A	N/A	9356	C+G	...ge\Application\msedge.exe	N/A
0	N/A	N/A	9476	C+G	...5nlh2txyewy\SearchApp.exe	N/A
0	N/A	N/A	12456	C+G	C:\Windows\explorer.exe	N/A
0	N/A	N/A	13604	C+G	...cw5nlh2txyewy\LockApp.exe	N/A
0	N/A	N/A	13816	C+G	...ChatStore\WeChatStore.exe	N/A
0	N/A	N/A	15276	C+G	...wekyb3d8bbwe\Video.UI.exe	N/A
0	N/A	N/A	15536	C+G	...e\PhoneExperienceHost.exe	N/A
0	N/A	N/A	17932	C+G	...bbwe\Microsoft.Photos.exe	N/A

3.2、pytorch 的 GPU 深度学习环境安装

输入如下 pytorch 下载网址：<https://pytorch.org/>，打开的网址如下图所示（随着时间的流程官网页面也会更新）。



将页面往下滑动，出现如下界面，可以看到当前最新的版本为 2.0.1 版本，对应的 cuda toolkit 有 11.7 和 11.8，安装方式有 pip 和 cuda，还有对应的电脑系统。由于我们要用的是 1.10.1 版本的 pytorch 和 cuda toolkit 有 11.3，因此点图中红框的连接，找之前的版本。



不断往下滑动，找到对应的 1.9.1 版本，复制红框中的链接

v1.10.1

Conda

OSX

```
# conda
conda install pytorch==1.10.1 torchvision==0.11.2 torchaudio==0.10.1 -c pytorch
```

Linux and Windows

```
# CUDA 10.2
conda install pytorch==1.10.1 torchvision==0.11.2 torchaudio==0.10.1 cudatoolkit=10.2 -c pytorch

# CUDA 11.3
conda install pytorch==1.10.1 torchvision==0.11.2 torchaudio==0.10.1 cudatoolkit=11.3 -c pytorch -c conda-forge

# CPU Only
conda install pytorch==1.10.1 torchvision==0.11.2 torchaudio==0.10.1 cpuonly -c pytorch
```

进入 anaconda 自己创建的环境，复制上述的命令，加载对应的依赖包，输入 y，进行对应的 pytorch 和 cudatoolkit 安装。中间可能会因为网络的原因导致下载对应的安装包失败，就需要从新输入该命令，使其恢复下载，已经下载好的包不会在次下载，只会下载没有安装好的包。

```
(torch) C:\Users\wu>conda install pytorch==1.10.1 torchvision==0.11.2 torchaudio==0.10.1 cudatoolkit=11.3 -c pytorch -c conda-forge
```

```

libuv conda-forge/win-64::libuv-1.44.2-h8ffe710_0
libwebp-base conda-forge/win-64::libwebp-base-1.2.4-h8ffe710_0
libxcb conda-forge/win-64::libxcb-1.13-hcd874cb_1004
libzlib conda-forge/win-64::libzlib-1.2.12-h8ffe710_2
lz4-c conda-forge/win-64::lz4-c-1.9.3-h8ffe710_1
m2w64-gcc-libgfortran conda-forge/win-64::m2w64-gcc-libgfortran-5.3.0-6
m2w64-gcc-libs conda-forge/win-64::m2w64-gcc-libs-5.3.0-7
m2w64-gcc-libs-core conda-forge/win-64::m2w64-gcc-libs-core-5.3.0-7
m2w64-gmp conda-forge/win-64::m2w64-gmp-6.1.0-2
m2w64-libwinpthread-git conda-forge/win-64::m2w64-libwinpthread-git-5.0.0.4634.697f757-2
mkl conda-forge/win-64::mkl-2022.1.0-h6a75c08_874
mkl-devel conda-forge/win-64::mkl-devel-2022.1.0-h57928b3_875
mkl-include conda-forge/win-64::mkl-include-2022.1.0-h6a75c08_874
msys2-conda-epoch conda-forge/win-64::msys2-conda-epoch-20160418-1
numpy conda-forge/win-64::numpy-1.23.2-py38h223ccf5_0
openjpeg conda-forge/win-64::openjpeg-2.5.0-hc9384bd_1
pillow conda-forge/win-64::pillow-9.2.0-py38h37aa274_2
pthread-stubs conda-forge/win-64::pthread-stubs-0.4-hcd874cb_1001
python_abi conda-forge/win-64::python_abi-3.8-2_cp38
pytorch pytorch/win-64::pytorch-1.10.1-py3.8_cuda11.3_cudnn8_0
pytorch-mutex pytorch/noarch::pytorch-mutex-1.0-cuda
tbb conda-forge/win-64::tbb-2021.5.0-h2d74725_1
tk conda-forge/win-64::tk-8.6.12-h8ffe710_0
torchaudio pytorch/win-64::torchaudio-0.10.1-py38_cu113
torchvision pytorch/win-64::torchvision-0.11.2-py38_cu113
typing_extensions conda-forge/noarch::typing_extensions-4.7.1-pyha770c72_0
xorg-libxau conda-forge/win-64::xorg-libxau-1.0.11-hcd874cb_0
xorg-libxdmcp conda-forge/win-64::xorg-libxdmcp-1.1.3-hcd874cb_0
xz conda-forge/win-64::xz-5.2.6-h8d14728_0
zstd conda-forge/win-64::zstd-1.5.2-h6255e5f_4

```

The following packages will be SUPERSEDED by a higher-priority channel:

```

ca-certificates anaconda/pkgs/main::ca-certificates-2~ --> conda-forge::ca-certificates-2~
openssl anaconda/pkgs/main::openssl-1.1.1w-h2~ --> conda-forge::openssl-1.1.1w-h2~

```

Proceed ([y]/n)? y

下载完成以后，输入如下的命令查看环境中的安装包：

Conda list

会得到该环境下的所有环境包，(该图只展示了部分)

#	Name	Version	Build	Channel
	blas	2.118	mkl	conda-forge
	blas-devel	3.9.0	18_win64_mkl	conda-forge
	ca-certificates	2023.7.22	h56e8100_0	conda-forge
	cudatoolkit	11.3.1	h280eb24_10	conda-forge
	cycler	0.11.0	pypi_0	pypi
	fonttools	4.42.1	pypi_0	pypi
	freetype	2.12.1	h546665d_0	conda-forge
	intel-openmp	2023.2.0	h57928b3_49496	conda-forge
	joblib	1.3.2	pypi_0	pypi
	jpeg	9e	h8ffe710_2	conda-forge
	kiwisolver	1.4.5	pypi_0	pypi
	lcms2	2.12	h2a16943_0	conda-forge
	lerc	3.0	h0e60522_0	conda-forge
	libblas	3.9.0	18_win64_mkl	conda-forge
	libcblas	3.9.0	18_win64_mkl	conda-forge
	libdeflate	1.12	h8ffe710_0	conda-forge
	liblapack	3.9.0	18_win64_mkl	conda-forge
	liblapacke	3.9.0	18_win64_mkl	conda-forge
	libpng	1.6.37	h1d00b33_4	conda-forge
	libtiff	4.4.0	h2ed3b44_1	conda-forge
	libuv	1.44.2	h8ffe710_0	conda-forge
	libwebp-base	1.2.4	h8ffe710_0	conda-forge
	libxcb	1.13	hcd874cb_1004	conda-forge
	libzlib	1.2.12	h8ffe710_2	conda-forge
	lz4-c	1.9.3	h8ffe710_1	conda-forge
	m2w64-gcc-libgfortran	5.3.0	6	conda-forge

查看是否有安装好的对应版本的 pytorch 和 cudatoolkit

```
pandas 1.3.4 pypi_0 pypi
pillow 9.2.0 py38h37aa274_2 conda-forge
pip 23.2.1 py38haa95532_0 https://mirrors.tuna.tsinghua.edu.cn/anaconda
pthread-stubs 0.4 hcd874cb_1001 conda-forge
pyparsing 3.1.1 pypi_0 pypi
python 3.8.0 hff0d562_2 https://mirrors.tuna.tsinghua.edu.cn/anaconda
python-dateutil 2.8.2 pypi_0 pypi
python_abi 3.8 2_cp38 conda-forge
pytorch 1.10.1 py3.8_cuda11.3_cudnn8.0 pytorch
pytorch-mutex 1.0 cuda pytorch
pytz 2023.3.post1 pypi_0 pypi
scikit-learn 1.3.0 pypi_0 pypi
scipy 1.10.1 pypi_0 pypi
setuptools 68.0.0 py38haa95532_0 https://mirrors.tuna.tsinghua.edu.cn/anaconda
setuptools-scm 7.1.0 pypi_0 pypi
six 1.16.0 pypi_0 pypi
sklearn 0.0 pypi_0 pypi
sqlite 3.41.2 h2bbff1b_0 https://mirrors.tuna.tsinghua.edu.cn/anaconda
tbb 2021.5.0 h2d74725_1 conda-forge
```

```
blas 2.118 mkl conda-forge
blas-devel 3.9.0 18_win64_mkl conda-forge
ca-certificates 2023.7.22 h56e8100_0 conda-forge
cudatoolkit 11.3.1 h280eb24_10 conda-forge
cycler 0.11.0 pypi_0 pypi
fonttools 4.42.1 pypi_0 pypi
freetype 2.12.1 h546665d_0 conda-forge
intel-openmp 2023.2.0 h57928b3_49496 conda-forge
joblib 1.3.2 pypi_0 pypi
jpeg 9e h8ffe710_2 conda-forge
kiwisolver 1.4.5 pypi_0 pypi
lcms2 2.12 h2a16943_0 conda-forge
lerc 3.0 h0e60522_0 conda-forge
```

中科大源 -i https://pypi.mirrors.ustc.edu.cn/simple/

豆瓣源 -i http://pypi.douban.com/simple/ --trusted-host pypi.douban.com

```
import torch
```

```
flag = torch.cuda.is_available()
```

```
print(flag) # 返回 true 为安装成功
```

```
ngpu = 1
```

```
# Decide which device we want to run on
```

```
device = torch.device("cuda:0" if (torch.cuda.is_available() and ngpu > 0) else "cpu")
```

```
print(device)
```

```
print(torch.cuda.get_device_name(0))
```

```
print(torch.rand(3, 3).cuda())
```

```
import torch
```

```
# Check CUDA version
```

```
cuda_version = torch.version.cuda
```

```
print("CUDA Version: ", cuda_version)
```

```
# Check CuDNN version
```

```
cuda_version = torch.backends.cudnn.version()
```

```
print("CuDNN Version:", cuda_version)
```