

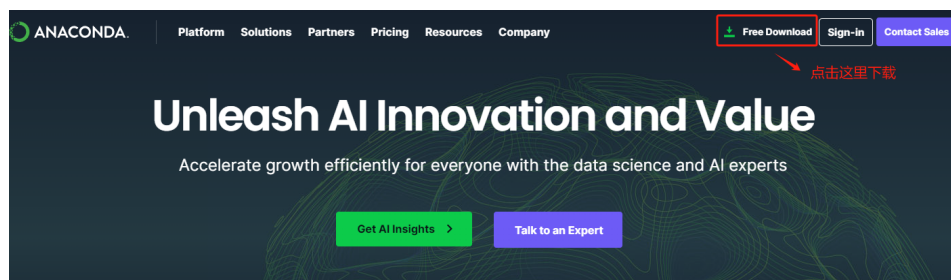
Ch 2. 在Windows 系统下配置 Python 环境

大多数Python项目依赖于特定的库或模块来正常运行。例如，一个网页爬虫可能依赖于requests库来发送网络请求，而数据分析项目可能依赖于pandas或numpy。这些依赖不会随着项目代码一起被包含在GitHub仓库中。所以我们下载的自然也不会包含这部分的内容。所以需要安装这个依赖环境。

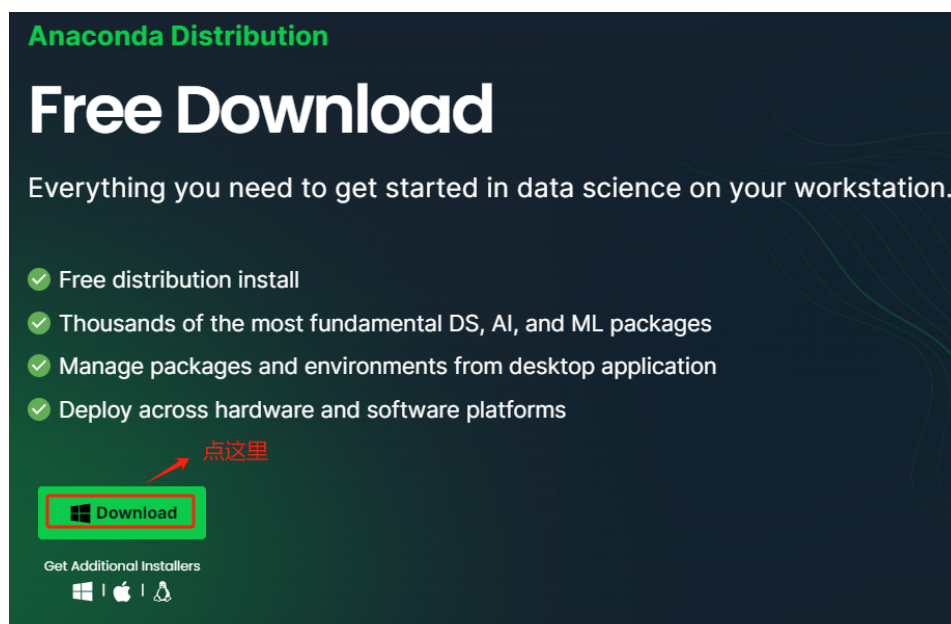
1. 使用Anaconda创建项目依赖环境（推荐）

Anaconda 是一个流行的开源Python发行版，用于科学计算。它包含了数据科学和机器学习领域中常用的一系列工具和库。其安装的方式也非常简单。具体安装过程如下：

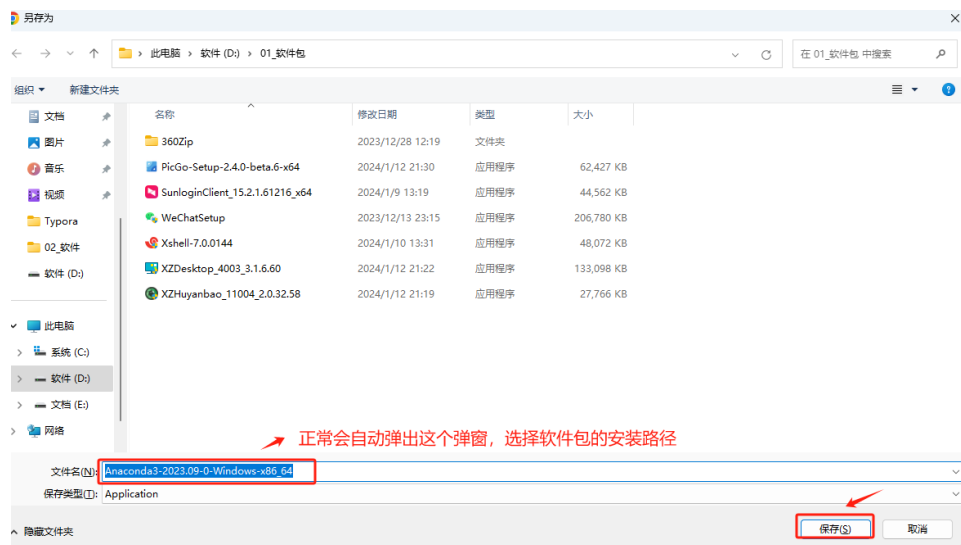
- Step 1. 进入Anaconda3的官网：<https://www.anaconda.com/>



- Step 2. 下载Windows版本的Anaconda3

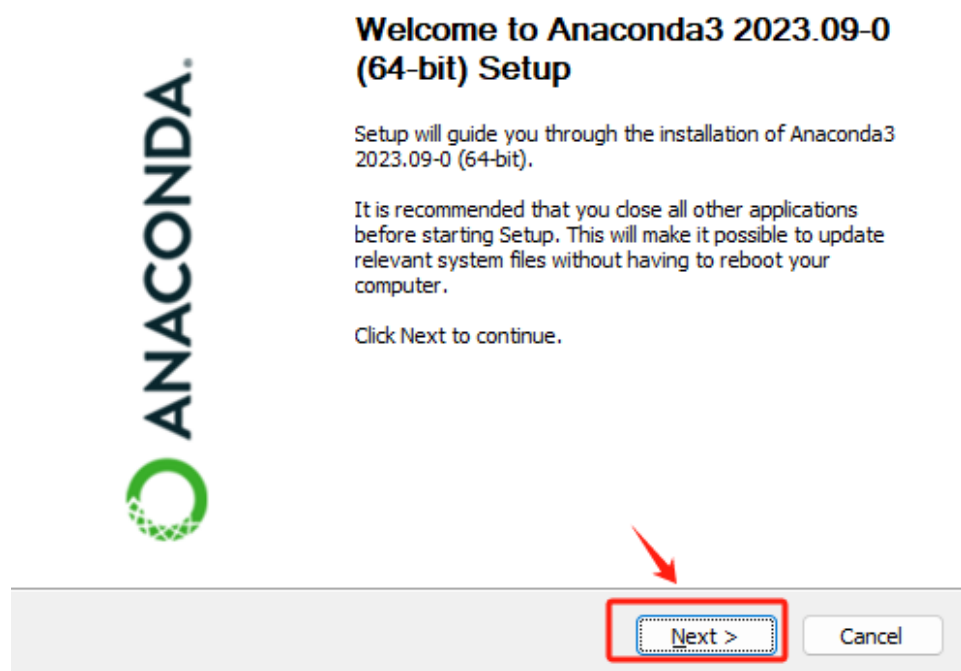


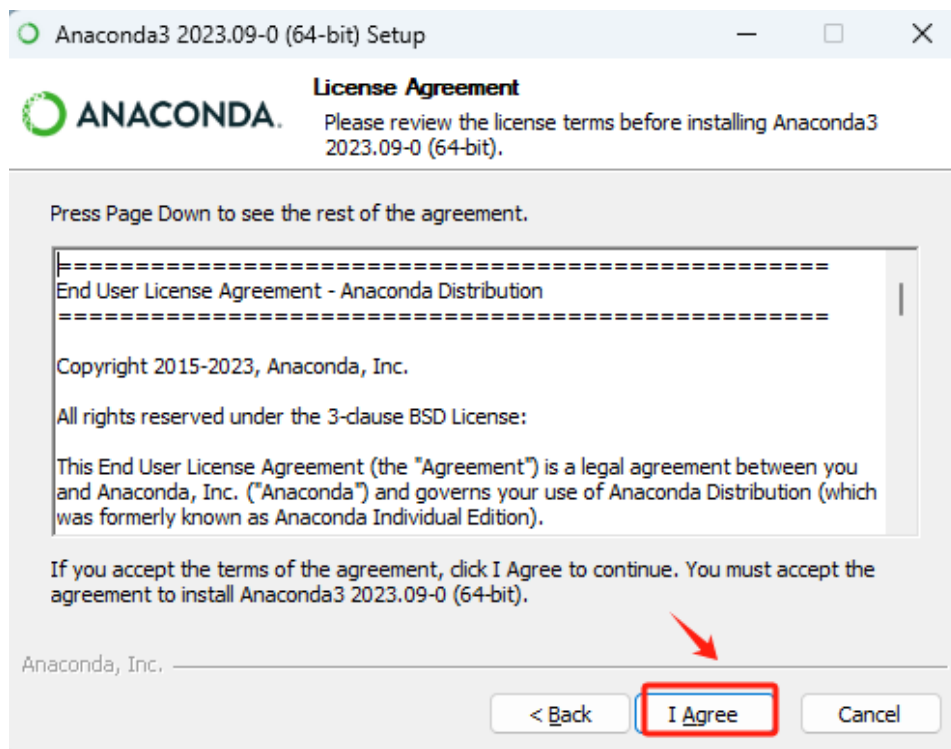
如果网络正常，会自动弹出下载弹窗，自行选择下载路径。



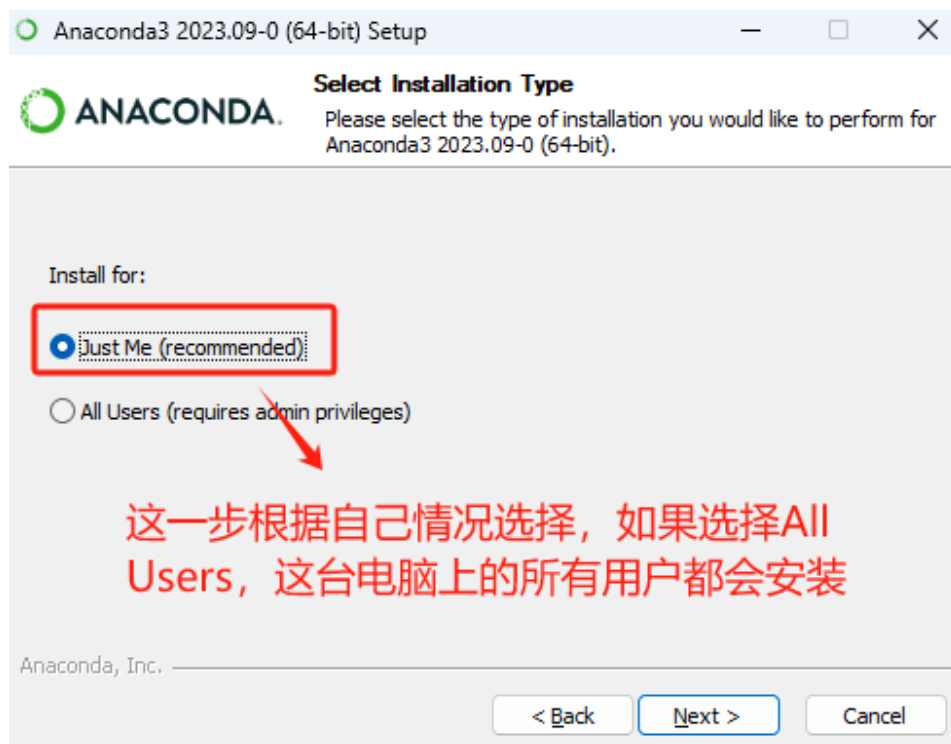
• Step 3. 安装Windows版本的Anaconda3

直接鼠标双击运行安装程序，执行傻瓜式安装。



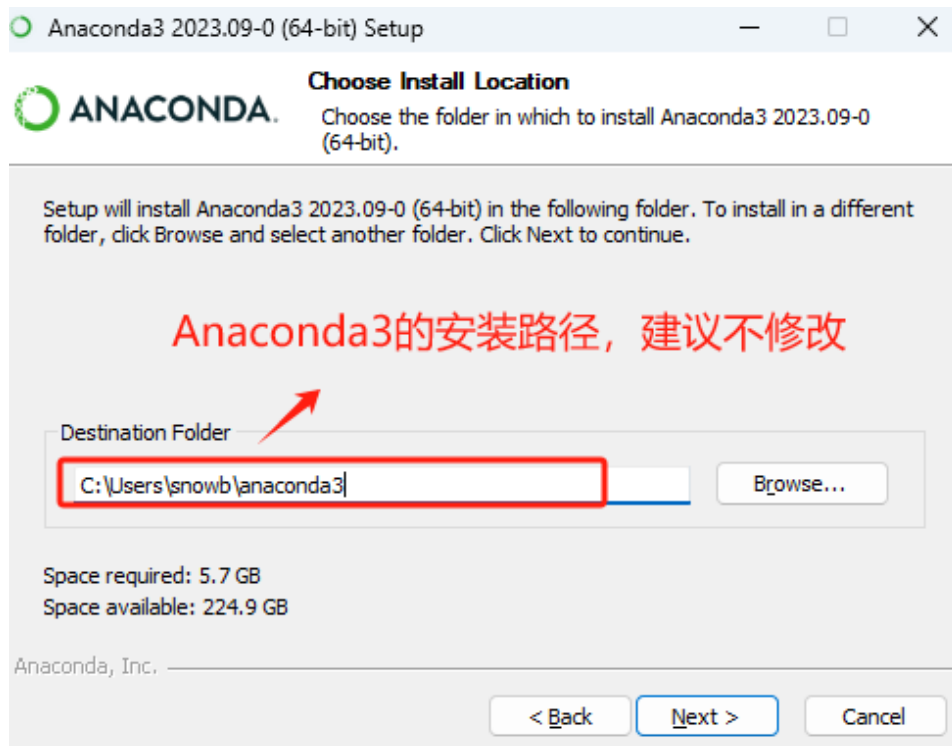


这里根据情况自行选择，选择哪一个都不会影响Windows当前用户的使用。

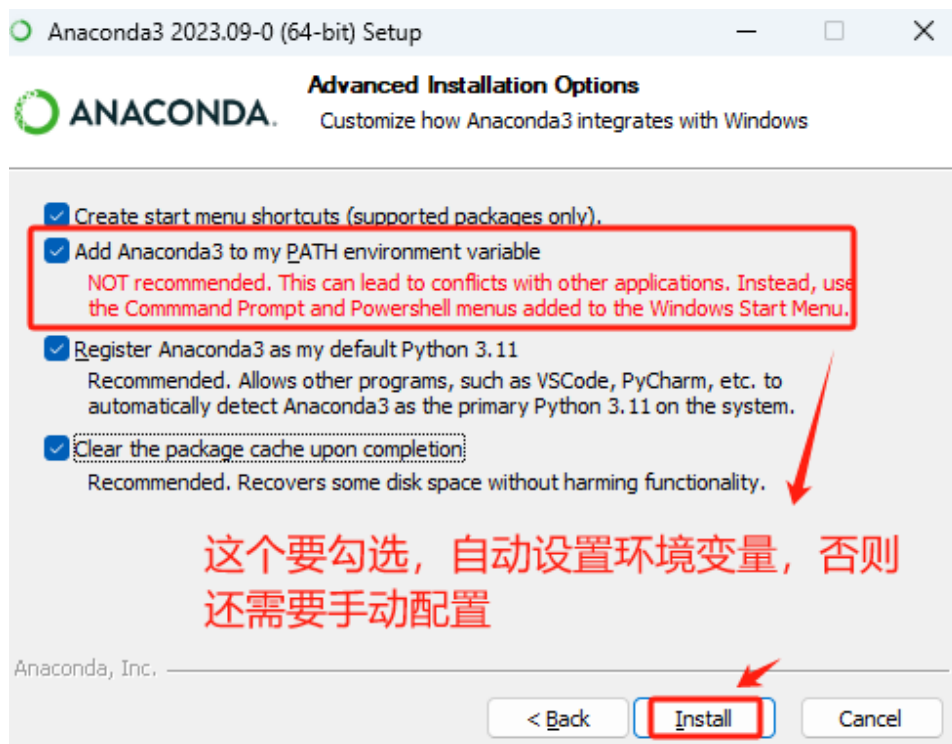


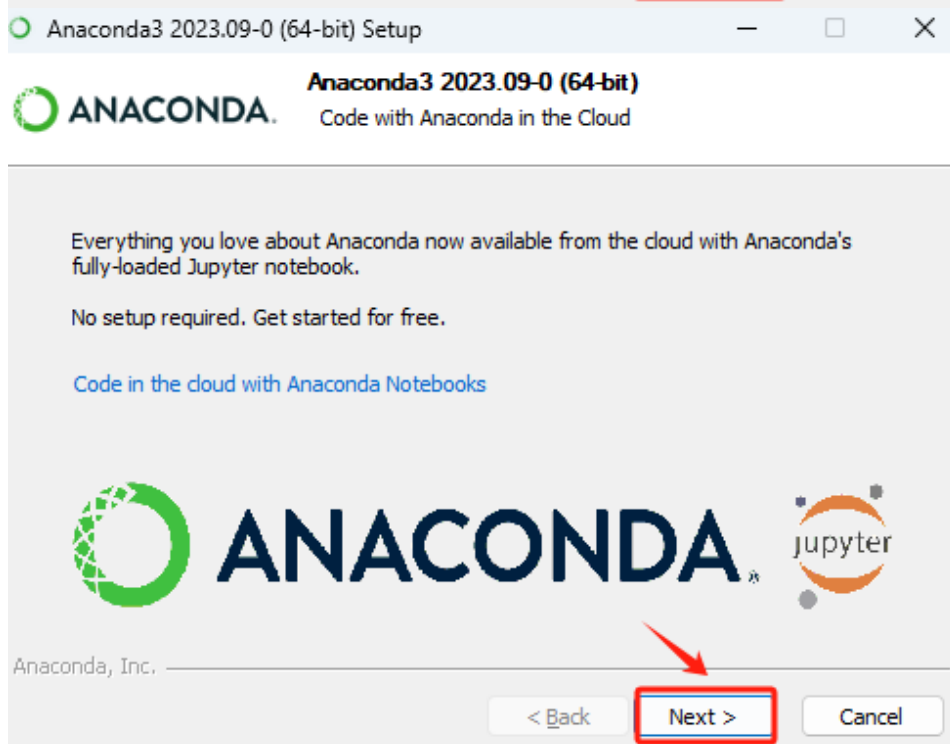
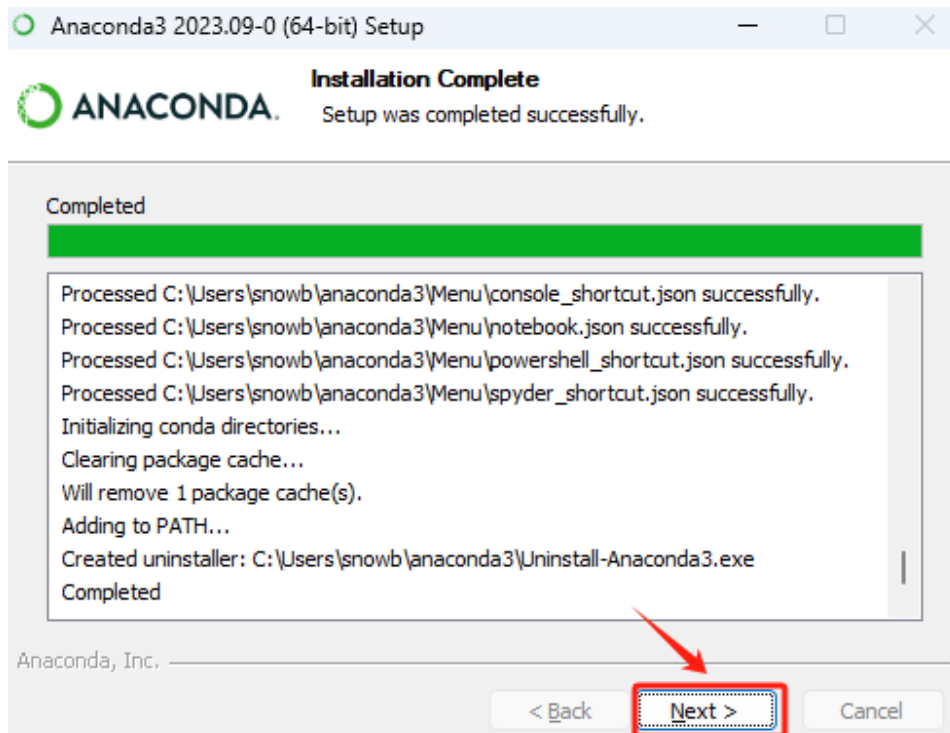
这一步根据自己情况选择，如果选择All Users，这台电脑上的所有用户都会安装

这个强烈不建议修改，如修改后可能会导致一下环境变量出现问题。



这里强烈建议勾选 Add Anaconda3 to my_path environment variable, 这会自动设置环境变量, 否则还需要后续再手动设置。



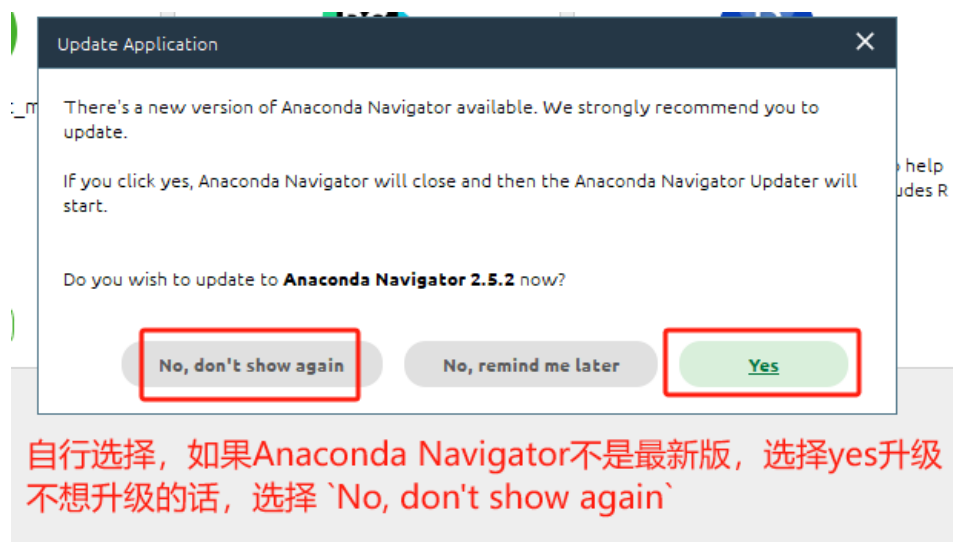




- Step 4. 启动Anaconda3进行验证



根据个人情况，自行选择是否需要升级到最新版本。



如果选择了升级的话，等待完成即可。

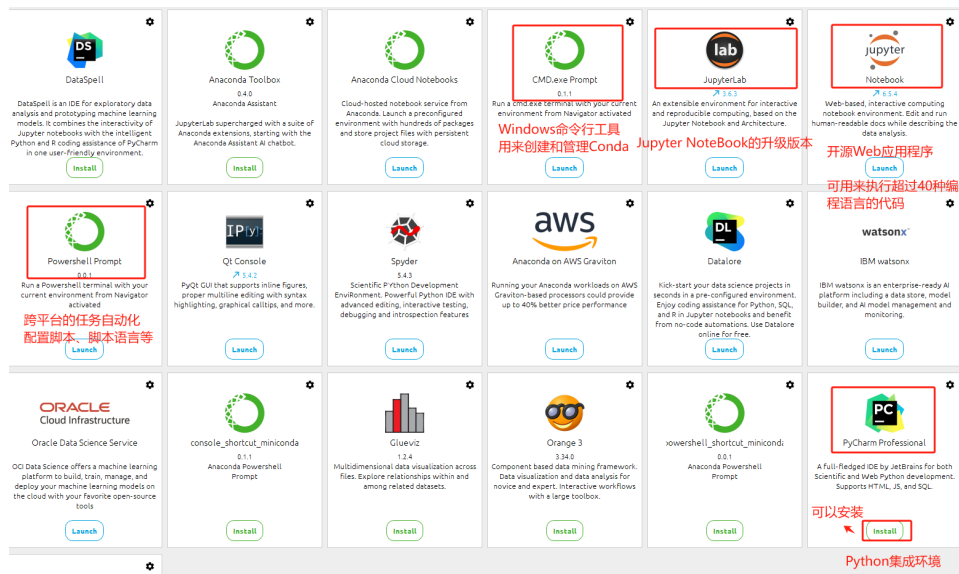


• Step 5. Anaconda3简要说明

Anaconda Navigator 是 Anaconda 发行版中的一个图形用户界面（GUI），提供了一个直观的界面，通过这个界面，用户可以管理不同的环境、安装和更新包，以及启动各种数据科学和机器学习工具，例如 Jupyter Notebook、Spyder、RStudio 等。此外，它还提供了对 Anaconda Cloud 和其他资源的访问，使得分享工作和发现其他人的工作变得更加容易。Navigator 适合那些不习惯使用命令行界面的用户。其中比较关键的应用程序已经做了标注：

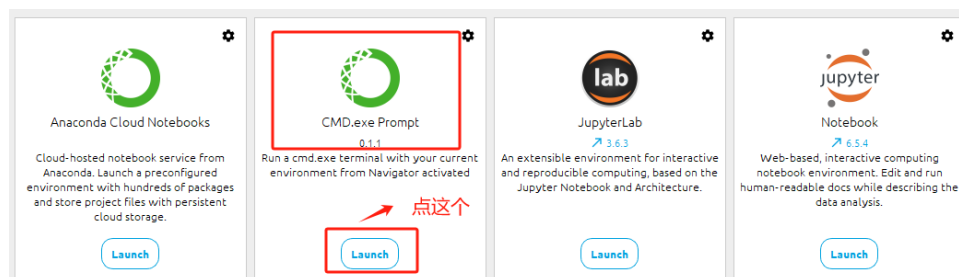
1. **Anaconda CMD.exe Prompt**：一个基于 Windows 命令行界面的工具，它预配置了 Anaconda 发行版的环境变量。使用 Anaconda Prompt，可以执行各种与 Anaconda 相关的命令，例如创建和管理 Conda 环境、安装、更新和卸载包等。这是一个非常强大的工具，特别是对于那些熟悉命令行操作的用户。
2. **Powershell Prompt**：类似于 Anaconda CMD.exe Prompt，Powershell Prompt 是另一种命令行界面，但它基于 Windows 的 PowerShell。PowerShell 是一种跨平台的任务自动化解决方案，包含命令行壳、脚本语言和配置管理框架。Anaconda 的 Powershell Prompt 也被配置为可以直接执行与 Anaconda 和 Conda 相关的操作。
3. **PyCharm**：一个流行的 Python 集成开发环境（IDE），由 JetBrains 开发。它提供了代码完成、调试、测试、版本控制等多种功能，非常适合 Python 项目的开发。
4. **Jupyter Lab**：它是 Jupyter Notebook 的下一代界面，提供了一个灵活和强大的工具集，用于交互式数据科学和科学计算。Jupyter Lab 不仅包括了 Jupyter Notebook 的所有功能，还增加了许多新特性，如更灵活的窗口布局、更好的编辑器、实时预览、终端、文件浏览器等。它支持多种类型的文档和活动，包括文本编辑器、Jupyter 笔记本、数据视图等，是一个更为综合的数据科学工作环境。

5. **Jupyter Notebook**: 它是一个开源的 Web 应用程序，允许创建和共享包含实时代码、方程、可视化和叙述文本的文档。它广泛用于数据清洗和转换、数值模拟、统计建模、数据可视化、机器学习等领域。Jupyter Notebook 支持超过 40 种编程语言，包括 Python、R、Julia 和 Scala。它是一个非常流行的工具，特别是在数据科学和学术研究中。

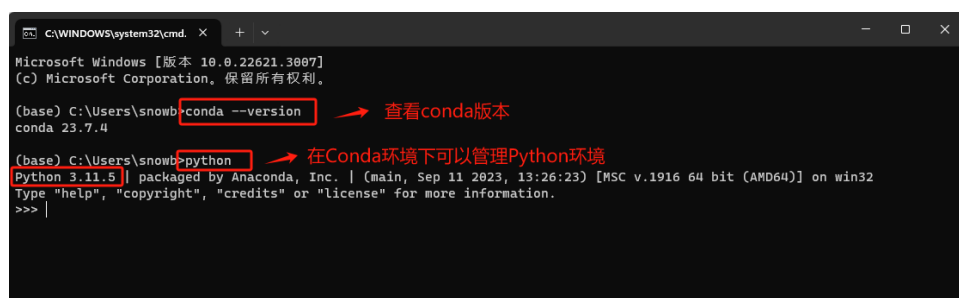


• Step 6. 验证Conda

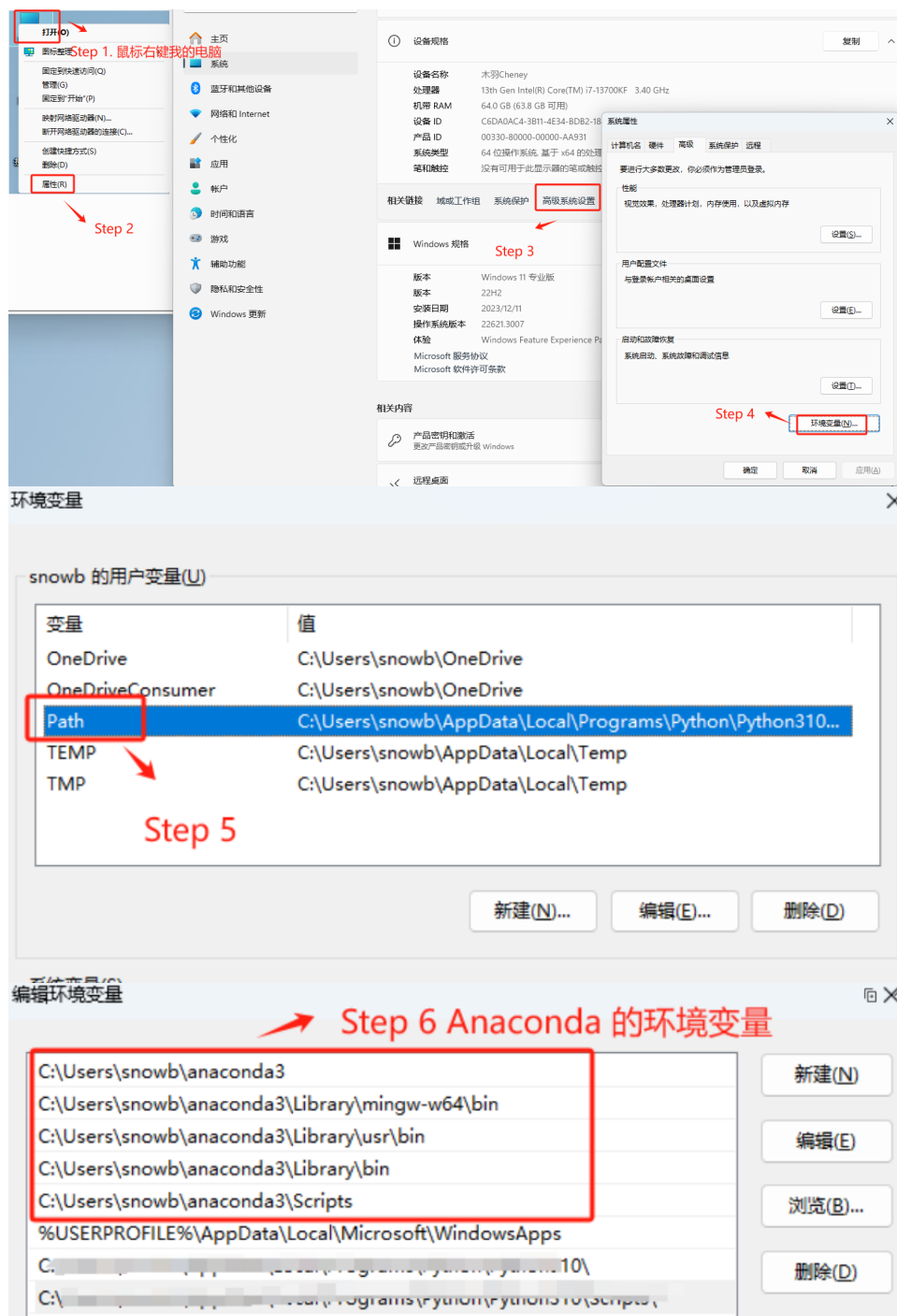
在Anaconda环境中，conda 是一个开源的包管理器和环境管理器，用于安装、运行和升级包和环境。验证方式如下：



点击后，会进入如下命令行终端，输入 `conda --version` 查看conda版本。



该命令会生效的原因，是因为我们在安装Anaconda3的过程中，自动添加了conda的环境变量。



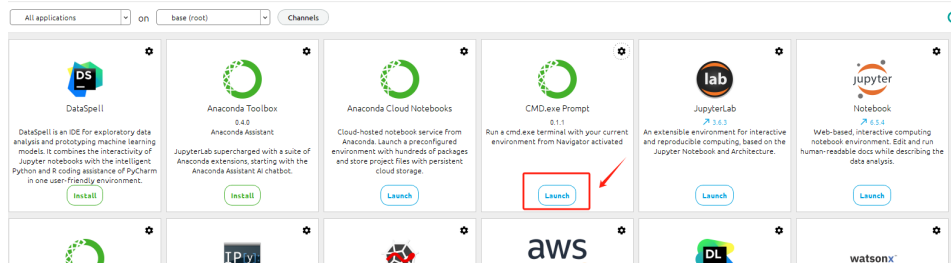
• Step 7. 升级conda到最新版本

强烈建议将conda 升级到最新版本，主要原因是最新版本的 conda 更有可能与其他最新的软件和库兼容。比如对最新版本的Python的支持，以及与最新发布包的兼容性，会在解析和冲突处理方面变得更加精确和高效。升级非常简单，执行如下命令：

```
# 首先更新conda工具
conda update -n base -c defaults conda

# 再更新各库
conda update --all
```

在Anaconda Navigator 中进入命令行终端。



在命令行终端先更新conda工具

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [版本 10.0.22621.3007]
(c) Microsoft Corporation. 保留所有权利。

(base) C:\Users\snowb>conda update -n base -c defaults conda → 先更新conda工具
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\snowb\anaconda3

  added / updated specs:
    - conda

The following packages will be downloaded:
```

在更新各依赖包。

```
C:\WINDOWS\system32\cmd. x + v
(base) C:\Users\snowb>conda update --all → 再更新依赖库
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\snowb\anaconda3

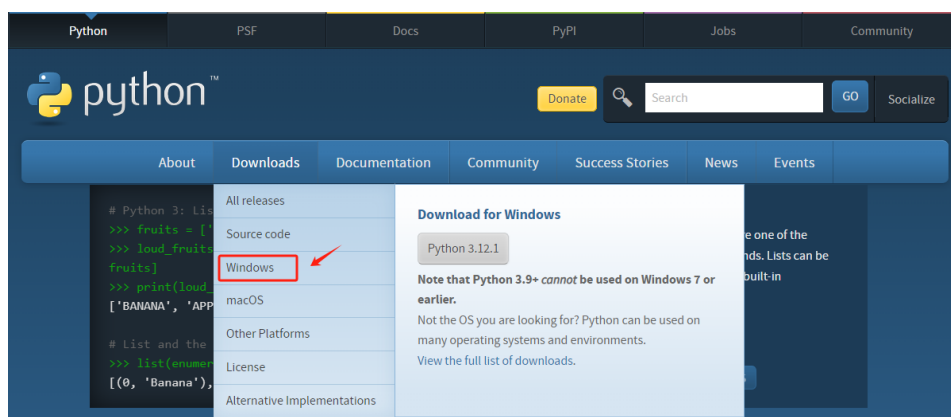
The following packages will be downloaded:

package | build
-----|-----
anaconda-anon-usage-0.4.3 | py311hfc23b7f_100 28 KB
anaconda-client-1.12.2 | py311haa95532_0 227 KB
menuinst-2.0.1 | py311hd77b12b_1 133 KB
numpy-1.24.3 | py311hdab7c0b_1 11 KB
```

2. 直接使用Python创建项目依赖环境

如果不想使用Anaconda3来管理虚拟环境，可以直接在Windows系统上安装指定版本的Python包，然后再使用该版本的Python来创建虚拟环境。操作过程如下：

- Step 1. 进入Python官网下载指定版本的Python包：<https://www.python.org/>



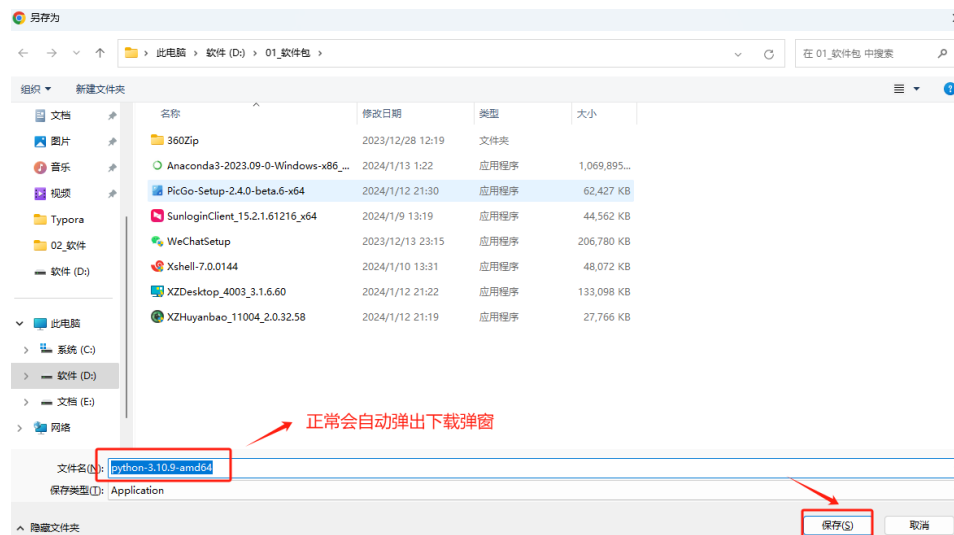
自行选择Python版本，运行大模型环境，需要Python 3.10版本及以上。我这里选择Python 3.10.9的Windows 64位安装。

Python 3.10.9 - Dec. 6, 2022

Note that Python 3.10.9 cannot be used on Windows 7 or earlier.

- Download [Windows embeddable package \(32-bit\)](#)
- Download [Windows embeddable package \(64-bit\)](#)
- Download [Windows help file](#)
- Download [Windows installer \(32-bit\)](#)
- Download [Windows installer \(64-bit\)](#) → 这里选择3.10.9版本安装

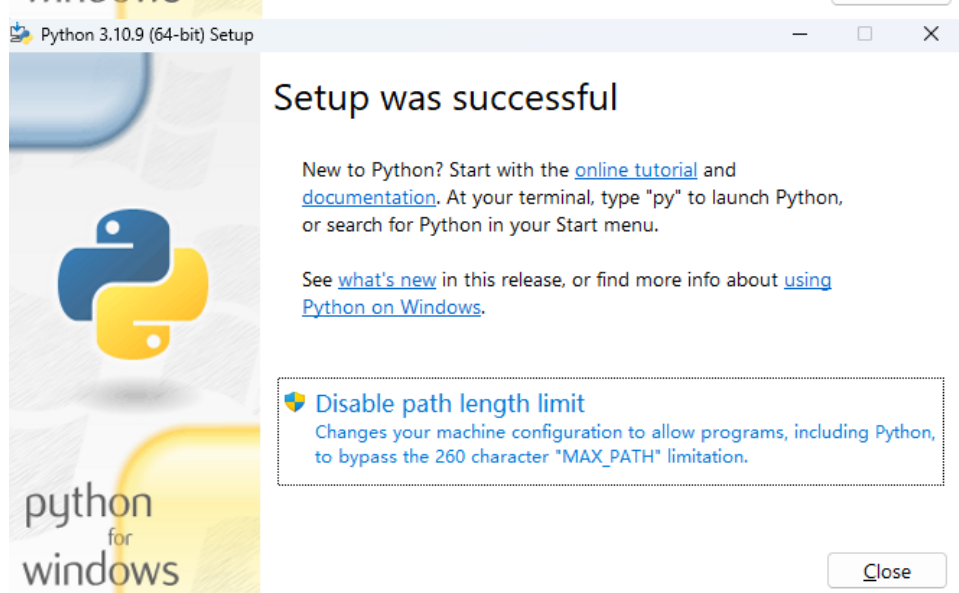
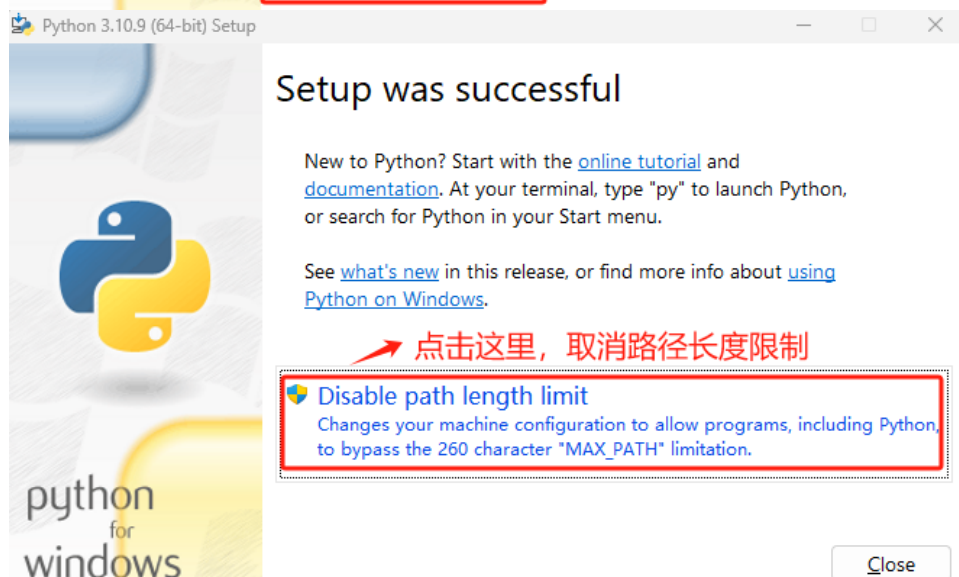
如果网络正常，会自动弹出下载弹窗，自行选择安装程序的存储路径。

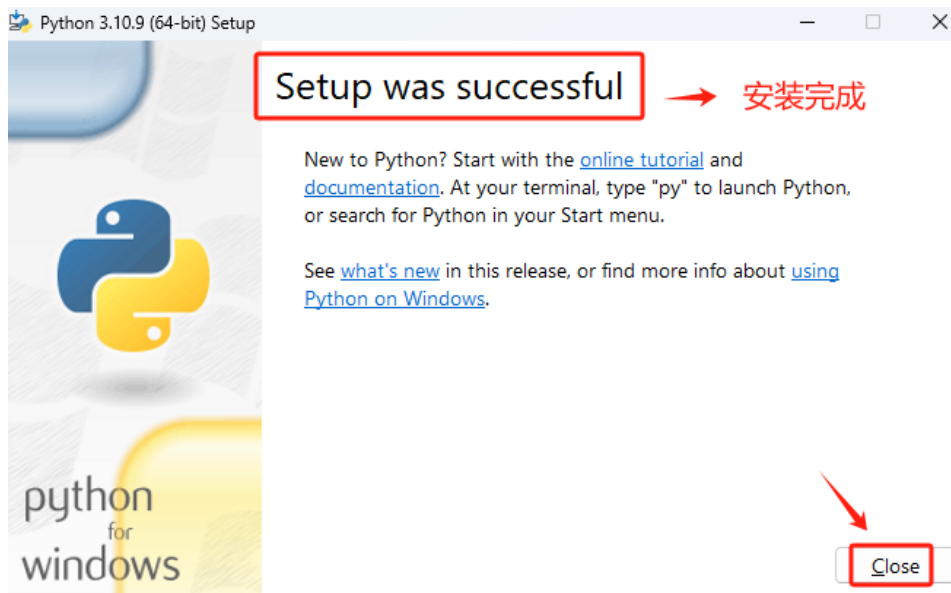


• Step 2. 下载完成后，执行傻瓜式安装

名称	修改日期	类型	大小
360Zip	2023/12/28 12:19	文件夹	
Anaconda3-2023.09-0-Windows-x86_64	2024/1/13 1:22	应用程序	1,069,895...
PicGo-Setup-2.4.0-beta.6-x64	2024/1/12 21:30	应用程序	62,427 KB
python-3.10.9-amd64 → 双击安装	2024/1/13 2:05	应用程序	28,301 KB
SunloginClient_15.2.1.61216_x64	2024/1/9 13:19	应用程序	44,562 KB
WeChatSetup	2023/12/13 23:15	应用程序	206,780 KB
Xshell-7.0.0144	2024/1/10 13:31	应用程序	48,072 KB
XZDesktop_4003_3.1.6.60	2024/1/12 21:22	应用程序	133,098 KB
XZHuyanbao_11004_2.0.32.58	2024/1/12 21:19	应用程序	27,766 KB

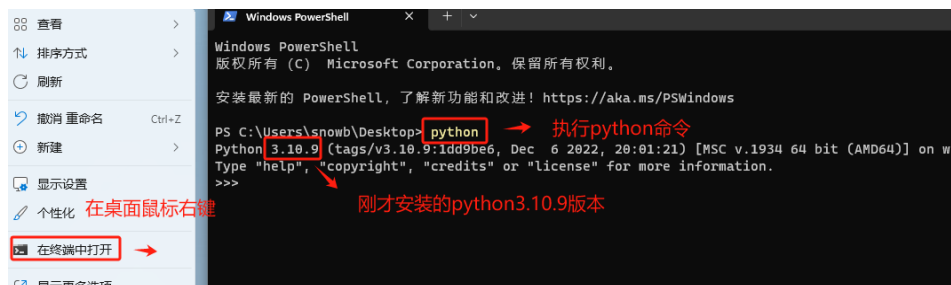
注意：这里勾选 `Add python.exe to PATH`，这会在安装过程中将Python环境设置好环境变量，否则还需要手动设置。





- Step 3. 安装完成后，验证是否安装成功

能正确输出安装的Python版本，说明安装正常。



- Step 4. 升级pip到最新版本

pip 是 Python 的官方包管理工具，用于安装和管理 Python 包。它允许用户从 Python 包索引 (PyPI) 中安装、升级和删除包。在创建新的虚拟环境前建议升级 pip，新版本的 pip 会提供更好的安装性能，更准确的依赖项解析。升级命令如下：

```
pip install --upgrade pip
```



3. 如何管理多个不同版本的Python环境？

对于直接使用Python直接创建虚拟环境的这种方式，如果需要切换python版本，则只能重新下载不同版本的Python的包，重复执行上述操作。比如我们上述安装的是Python 3.10.9，如果另外一个项目需要使用Python3.12版本，则需要重新进入<https://www.python.org/> 官网中下载。

Note that Python 3.12.1 *cannot* be used on Windows 7 or earlier.

4. 在下载一个Python3.12.1版本

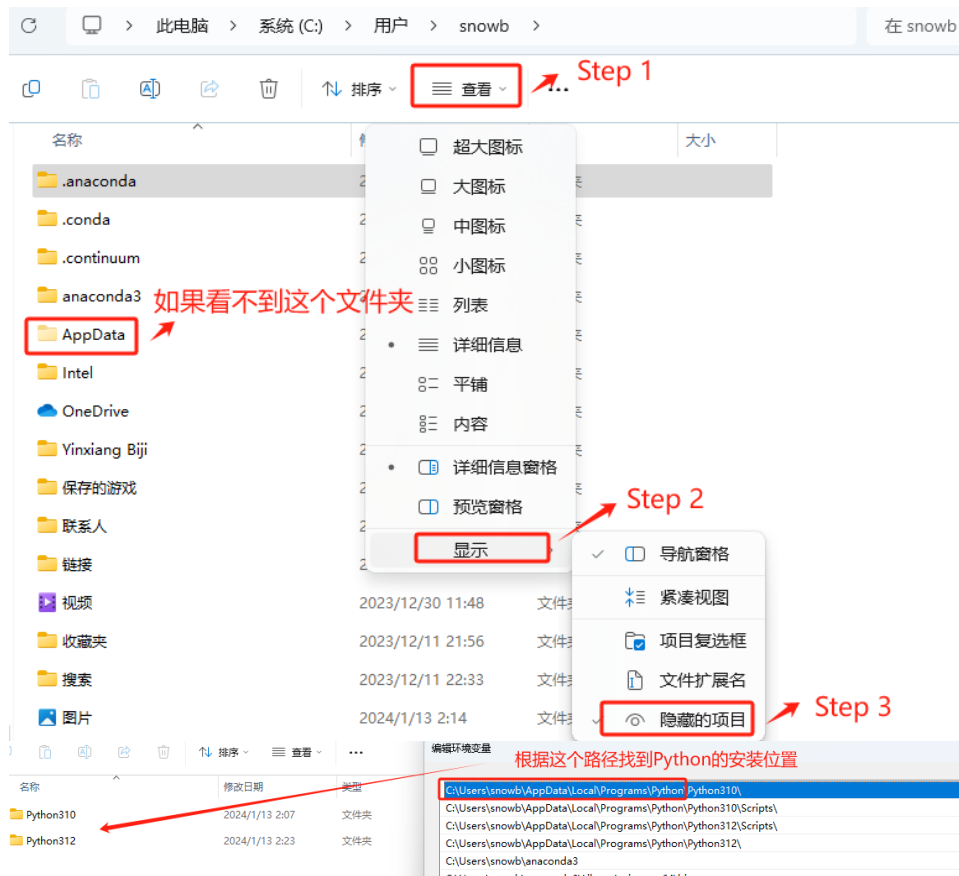
- Download [Windows embeddable package \(32-bit\)](#)
- Download [Windows embeddable package \(64-bit\)](#)
- Download [Windows embeddable package \(ARM64\)](#)
- Download [Windows installer \(32-bit\)](#)
- Download [Windows installer \(64-bit\)](#)
- Download [Windows installer \(ARM64\)](#)

本

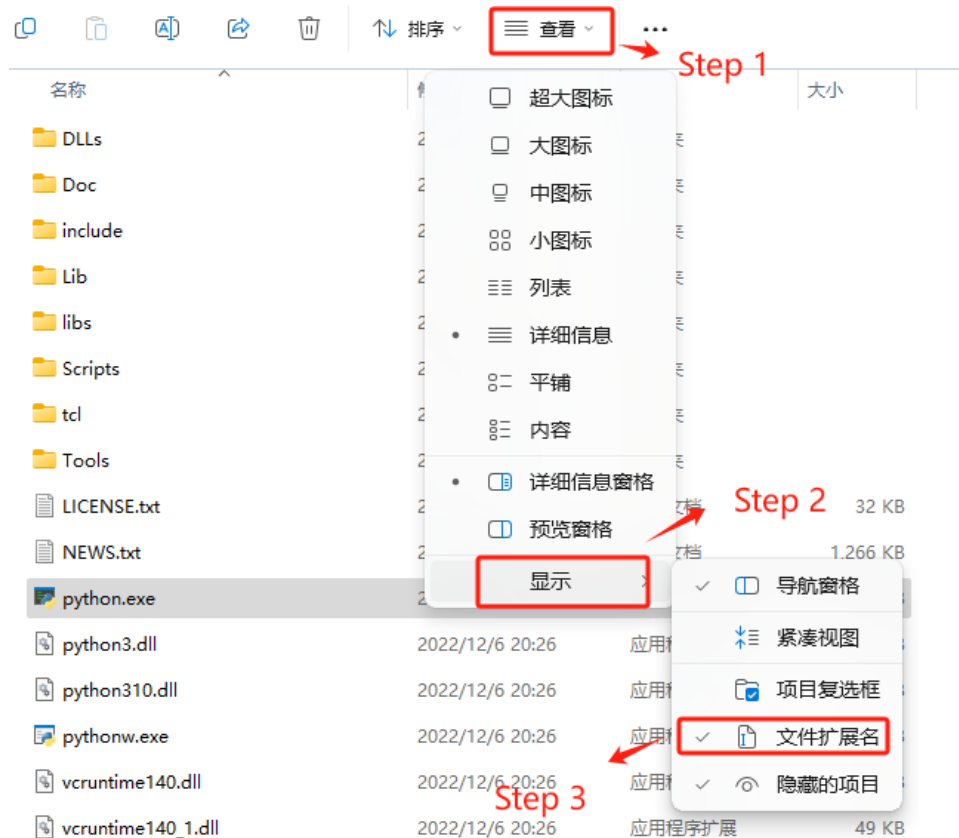
- [Python 3.13.0a1 - Oct. 13, 2023](#)
 - Download [Windows embeddable package \(32-bit\)](#)

[illegible][illegible]

如果想通过不同的指令，启动不同版本的Python，可以在系统的环境变量中找到指定版本的Python存放位置。



默认所有版本的python执行文件都是python.exe，可以进入指定文件夹中看到。



如果想区分不同版本Python，并在终端可以使用，可以通过如下方式添加版本编号。

DLLs	2024/1/13 2:07	文件夹	
Doc	2024/1/13 2:07	文件夹	
include	2024/1/13 2:07	文件夹	
Lib	2024/1/13 2:07	文件夹	
libs	2024/1/13 2:07	文件夹	
Scripts	2024/1/13 2:07	文件夹	
tcl	2024/1/13 2:07	文件夹	
Tools	2024/1/13 2:07	文件夹	
LICENSE.txt	2022/12/6 20:26	文本文档	32 KB
NEWS.txt	2022/12/6 20:27	文本文档	1,266 KB
python310.exe	2022/12/6 20:26	应用程序	100 KB
python3.dll	2022/12/6 20:26	应用程序扩展	64 KB
python310.dll	2022/12/6 20:26	应用程序扩展	4,388 KB
pythonw.exe	2022/12/6 20:26	应用程序	98 KB
vcruntime140.dll	2022/12/6 20:26	应用程序扩展	107 KB
vcruntime140_1.dll	2022/12/6 20:26	应用程序扩展	49 KB

将原有的python.exe 改为python的指定版本以区分

这里是 python --> python310

启动验证。

```

C:\Users\snowb>python → 这进入的anaconda的python环境
Python 3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Users\snowb>python310 → 这进入的python3.10的环境
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Users\snowb>python312 → 这进入的python3.12的环境
Python 3.12.1 (tags/v3.12.1:2395ca5, Dec 7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |

```

通过这种方式来灵活的管理一台机器上的多个Python环境。