

# Python 开发环境安装配置

## 目录

|   |    |
|---|----|
| 1. 总说明 .....  | 2  |
| 2. 安装 Anaconda .....                                  | 3  |
| 2.1. 下载安装 Anaconda .....                              | 3  |
| 2.2. 配置系统环境变量 .....                                   | 5  |
| 2.3. 设置 Mac 上的环境变量---搜索路径 PATH .....                  | 9  |
| 2.4. 验证 python 环境的设置 .....                            | 9  |
| 3. 配置 pip 使用清华源 .....                                 | 11 |
| 4. 安装 jieba 和 emoji 包 .....                           | 13 |
| 5. 下载安装 ollama .....                                  | 16 |
| 5.1. 安装 ollama .....                                  | 16 |
| 5.2. 设置 windows 的系统变量 .....                           | 17 |
| 5.3. 设置 Mac 上的系统变量 .....                              | 21 |
| 5.4. 通过 ollama 下载 qwen3:0.6b 大模型 .....                | 21 |
| 5.5. 无网情况下使用 ollama .....                             | 23 |
| 6. Modelscope 自行下载 qwen3-embedding-0.6B .....         | 25 |
| 7. 安装大模型相关模块 .....                                    | 27 |
| 7.1. 安装 gensim .....                                  | 27 |
| 7.2. 安装 sentence_transformers 、 ollama 和 openai ..... | 28 |
| 7.3. Python 降级 .....                                  | 28 |
| 8. 获取阿里云通义千问 API Key .....                            | 30 |
| 8.1. 设置 Win 环境变量 .....                                | 30 |
| 8.2. 设置 Mac 的环境变量 .....                               | 33 |

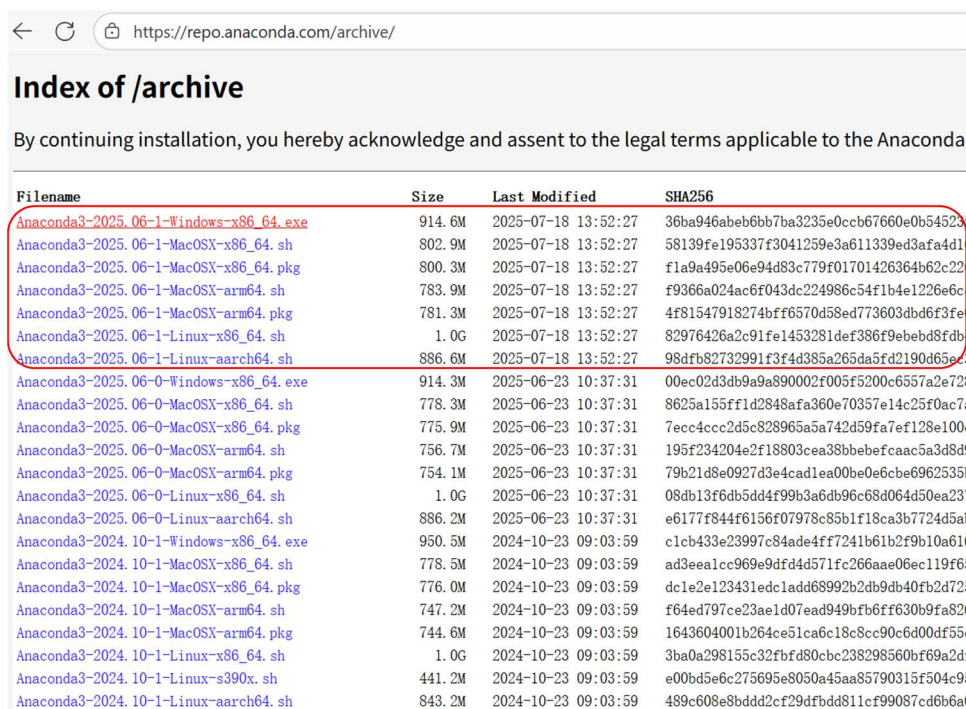
## 1. 总说明

因为不同版本的 Anaconda 对应的 Python 版本有所不同，而后续系统中使用的模块对 Python 版本有要求。如果不是资深的程序员，我们强烈建议您**首先把计算机上所有的关于 Python 系统都卸载**，得到一个完全没有 Python 的计算机，然后按照下面的流程安装配置所需的环境。

## 2. 安装 Anaconda

### 2.1. 下载安装 Anaconda

Anaconda 的所有发布版本都列在这个网页：<https://repo.continuum.io/archive/>

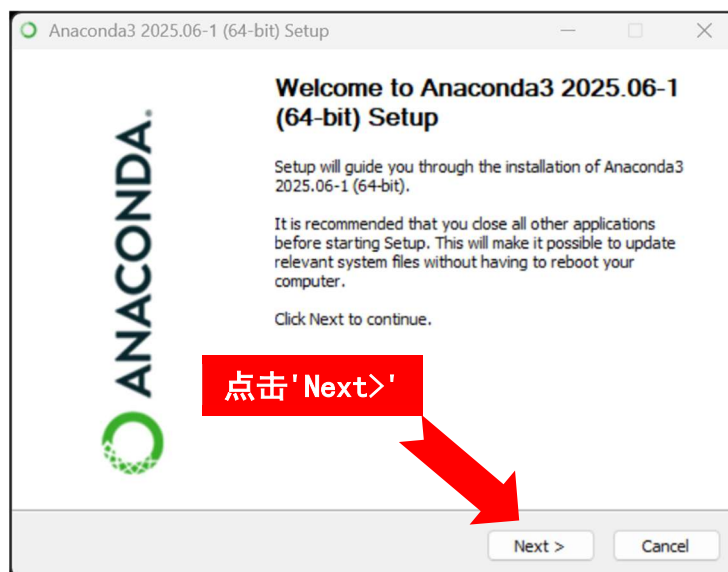


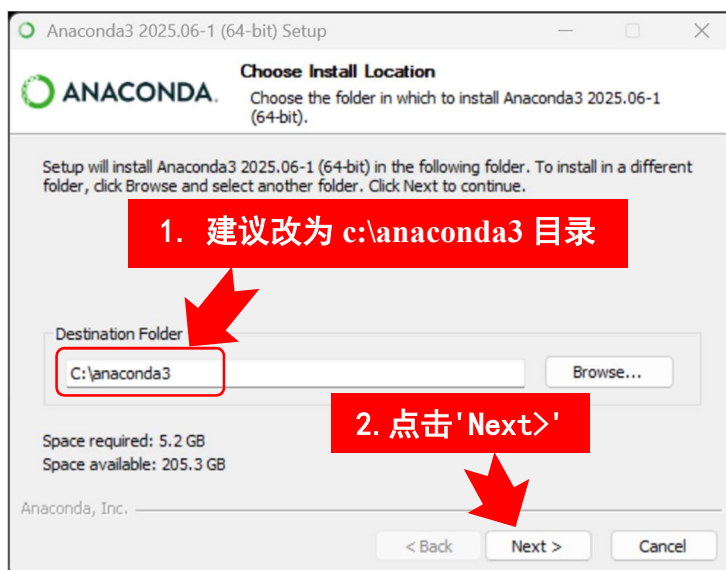
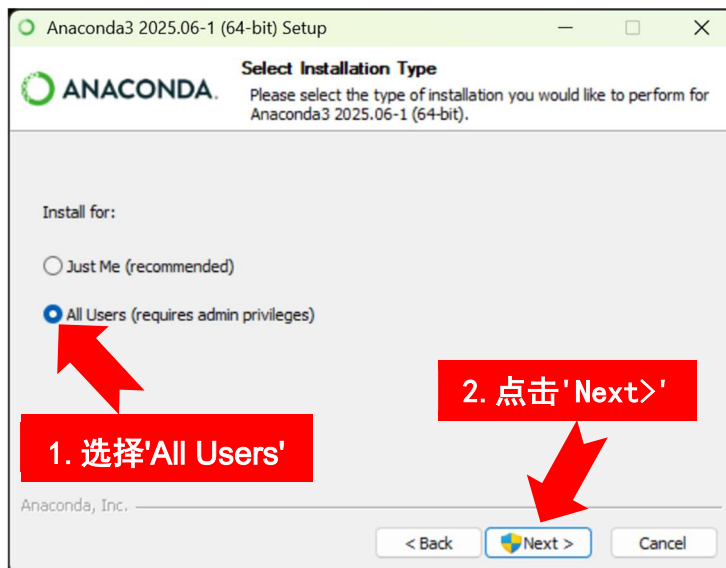
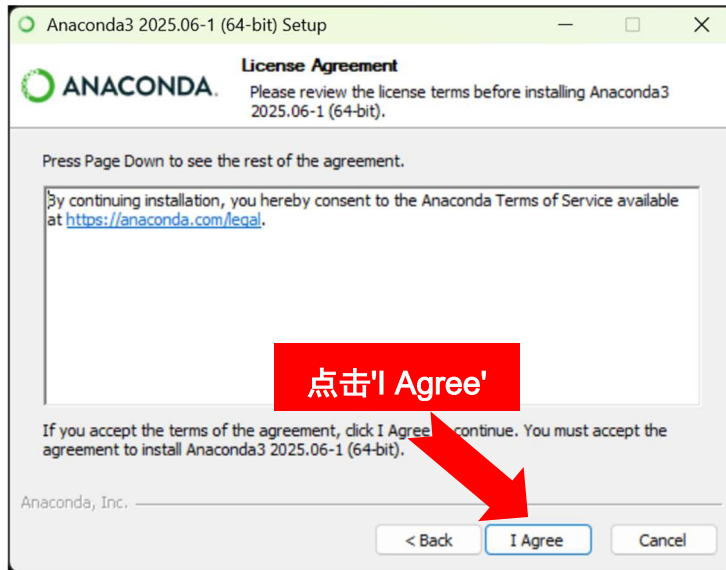
Index of /archive

By continuing installation, you hereby acknowledge and assent to the legal terms applicable to the Anaconda

| Filename   | Size   | Last Modified       | SHA256                                   |
|--|--------|---------------------|--|
| <a href="#">Anaconda3-2025.06-1-Windows-x86_64.exe</a> | 914.6M | 2025-07-18 13:52:27 | 36ba946abeb6bb7ba3235e0ccb67660e0b54523  |
| <a href="#">Anaconda3-2025.06-1-MacOSX-x86_64.sh</a>   | 802.9M | 2025-07-18 13:52:27 | 58139fe195337f3041259e3a611339ed3afa4d1  |
| <a href="#">Anaconda3-2025.06-1-MacOSX-x86_64.pkg</a>  | 800.3M | 2025-07-18 13:52:27 | f1a9a495e06e94d83c779f01701426364b62c22  |
| <a href="#">Anaconda3-2025.06-1-MacOSX-arm64.sh</a>    | 783.9M | 2025-07-18 13:52:27 | f9366a024ac6f043dc224986c54f1b4e1226e6c  |
| <a href="#">Anaconda3-2025.06-1-MacOSX-arm64.pkg</a>   | 781.3M | 2025-07-18 13:52:27 | 4f81547918274bf6570d58ed773603dbd6f3fe   |
| <a href="#">Anaconda3-2025.06-1-Linux-x86_64.sh</a>    | 1.0G   | 2025-07-18 13:52:27 | 82976426a2c91fe1453281def386f9ebeb8fdb   |
| <a href="#">Anaconda3-2025.06-1-Linux-aarch64.sh</a>   | 886.6M | 2025-07-18 13:52:27 | 98dfb82732991f3f4d385a265da5fd2190d65e   |
| <a href="#">Anaconda3-2025.06-0-Windows-x86_64.exe</a> | 914.3M | 2025-06-23 10:37:31 | 00ec02d3db9a9a890002f005f5200c6557a2e72i |
| <a href="#">Anaconda3-2025.06-0-MacOSX-x86_64.sh</a>   | 778.3M | 2025-06-23 10:37:31 | 8625a155ff1d2848afa360e70357e14c25f0ac7i |
| <a href="#">Anaconda3-2025.06-0-MacOSX-x86_64.pkg</a>  | 775.9M | 2025-06-23 10:37:31 | 7ecc4ccc2d5c828965a5a742d59fa7ef128e100i |
| <a href="#">Anaconda3-2025.06-0-MacOSX-arm64.sh</a>    | 756.7M | 2025-06-23 10:37:31 | 195f234204e2f18803cea38bbebefcaac5a3d8di |
| <a href="#">Anaconda3-2025.06-0-MacOSX-arm64.pkg</a>   | 754.1M | 2025-06-23 10:37:31 | 79b21d8e0927d3e4cad1ea00be0e6cbe6962535i |
| <a href="#">Anaconda3-2025.06-0-Linux-x86_64.sh</a>    | 1.0G   | 2025-06-23 10:37:31 | 08db13f6db5dd4f99b3a6db96c68d064d50ea23i |
| <a href="#">Anaconda3-2025.06-0-Linux-aarch64.sh</a>   | 886.2M | 2025-06-23 10:37:31 | e6177f844f6156f07978c85b1f18ca3b7724d5ai |
| <a href="#">Anaconda3-2024.10-1-Windows-x86_64.exe</a> | 950.5M | 2024-10-23 09:03:59 | c1cb433e23997c84ade4ff7241b61b2f9b10a61i |
| <a href="#">Anaconda3-2024.10-1-MacOSX-x86_64.sh</a>   | 778.5M | 2024-10-23 09:03:59 | ad3ee1cc969e9dfd4d571fc266aae06ec119f6i  |
| <a href="#">Anaconda3-2024.10-1-MacOSX-x86_64.pkg</a>  | 776.0M | 2024-10-23 09:03:59 | dc1e2e123431edc1add68992b2db9db40fb2d72i |
| <a href="#">Anaconda3-2024.10-1-MacOSX-arm64.sh</a>    | 747.2M | 2024-10-23 09:03:59 | f64ed797ce23ae1d07ead949bfb6ff630b9fa82i |
| <a href="#">Anaconda3-2024.10-1-MacOSX-arm64.pkg</a>   | 744.6M | 2024-10-23 09:03:59 | 1643604001b264ce51ca6c18c8cc90c6d00df55i |
| <a href="#">Anaconda3-2024.10-1-Linux-x86_64.sh</a>    | 1.0G   | 2024-10-23 09:03:59 | 3ba0a298155c32fbfd80bc238298560bf69a2di  |
| <a href="#">Anaconda3-2024.10-1-Linux-s390x.sh</a>     | 441.2M | 2024-10-23 09:03:59 | e00bd5e6c275695e8050a45aa85790315f504c9i |
| <a href="#">Anaconda3-2024.10-1-Linux-aarch64.sh</a>   | 843.2M | 2024-10-23 09:03:59 | 489c608e8bddd2cf29dfbdd811cf99087cd6b6ai |

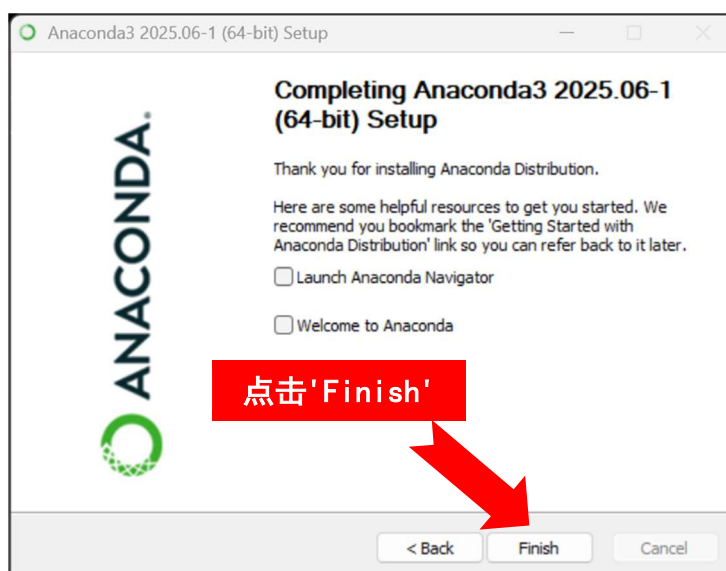
一般建议下载最新的系统，可能安装的过程中，有些设置会有部分变化，但基本差不多。下面的安装界面是 2025.06-1，也可以作为参考安装其他的版本。







建议把**三项都选中**，然后点击 **Install** 进行系统安装。



## 2.2. 配置系统环境变量

为了方便使用 `python`，需要将 `python` 相关程序所在的目录加入到系统的搜索路径中，即 `Path` 中。

需要将下面的路径加入到系统“环境变量”中的 `Path`：

`C:\anaconda3`

`C:\anaconda3\Scripts`

`C:\anaconda3\Library\usr\bin`

`C:\anaconda3\Library\bin`

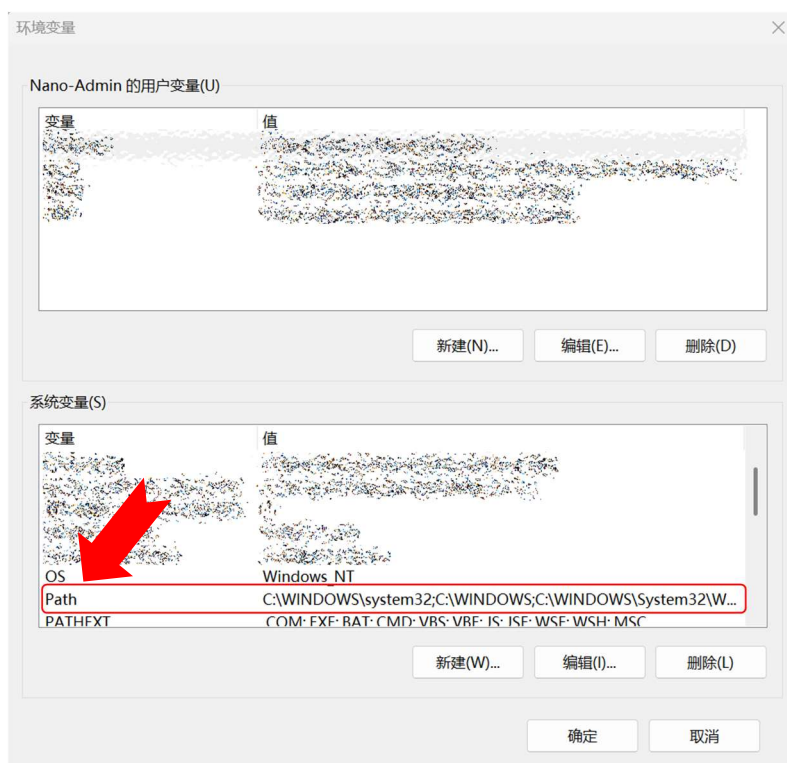
在**管理员帐号**下，windows 操作系统：开始→“设置”→“系统信息”→“高级系统设置”→在“高级”选项卡→“环境变量”。



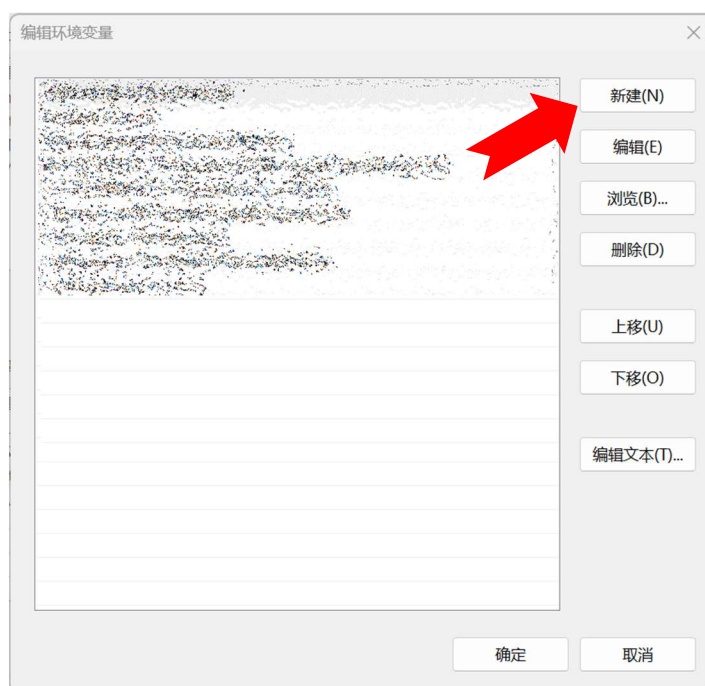
在“高级”选项卡中，点击“环境变量 ...”



双击“系统变量”中的 Path:

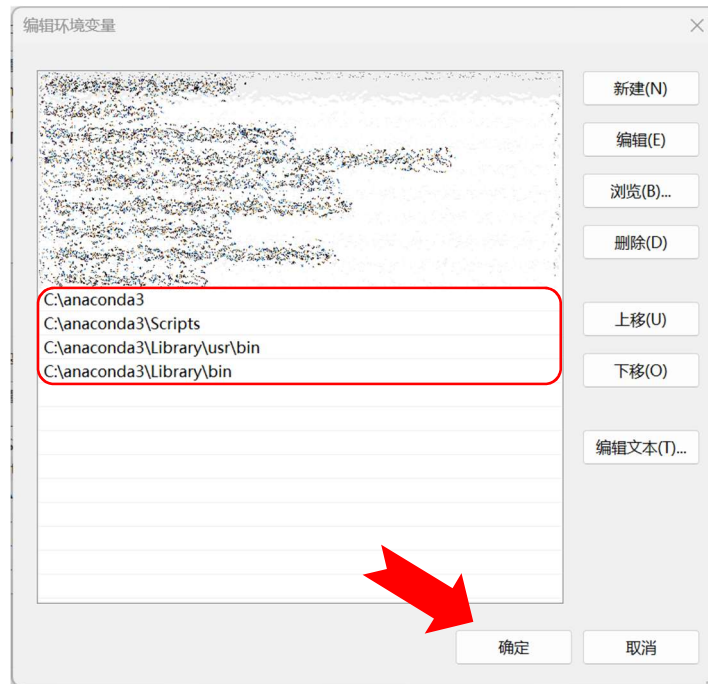


点击“新建”后，系统会增加一个条目。

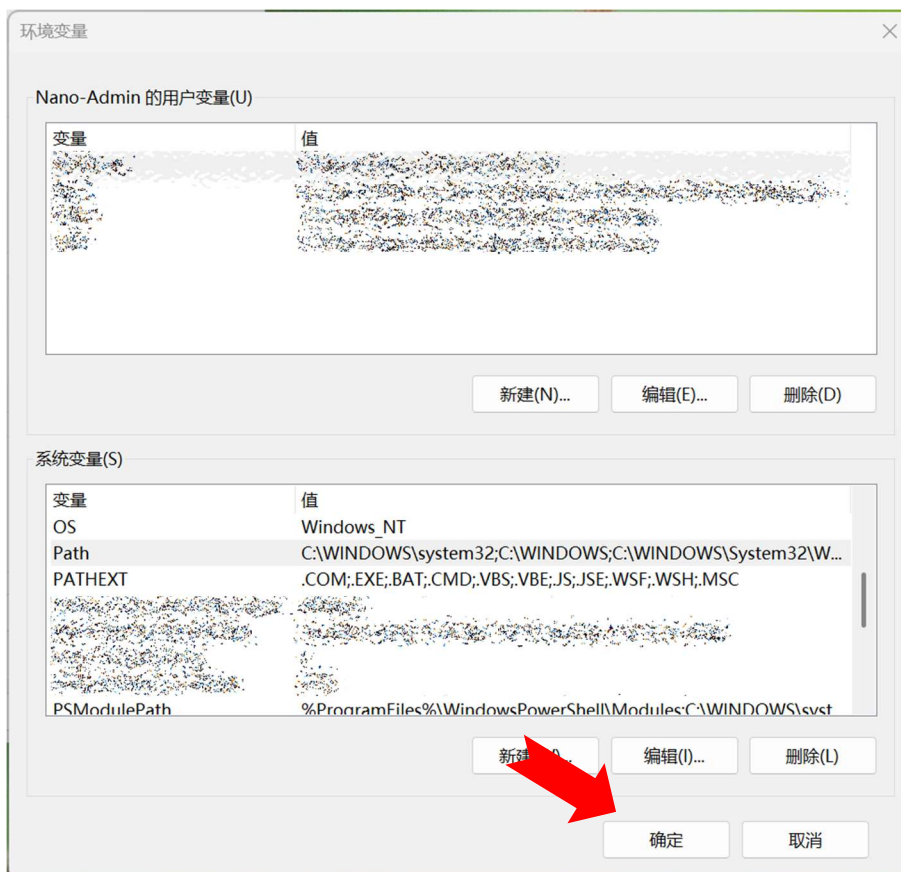


这个时候把下面四个路径逐次输入，分别拷贝后粘贴即可：

C:\anaconda3  
C:\anaconda3\Scripts  
C:\anaconda3\Library\usr\bin  
C:\anaconda3\Library\bin



将上图中的 4 个路径加入后，点击“确定”，关闭“编辑环境变量”窗口。



然后，点击“确定”，就可以关闭“环境变量”的设置。为了确保系统变量的设置有效，建议在设置以后，重启系统。

## 2.3. 设置 Mac 上的环境变量---搜索路径 PATH

在 Mac 环境下，可以通过下面方法设置环境变量，即搜索路径 PATH，或者您熟悉的命令进行搜索路径的设置。下面的命令，假设你的 anaconda3 安装在 /opt/anaconda3 下，如果是其他的目录，请做相应的修改。

```
echo 'export PATH="/path/to/anaconda3:$PATH"' >> ~/.zshrc
echo 'export PATH="/path/to/anaconda3/Scripts:$PATH"' >> ~/.zshrc
echo 'export PATH="/path/to/anaconda3/Library/usr/bin:$PATH"' >> ~/.zshrc
echo 'export PATH="/path/to/anaconda3/Library/bin:$PATH"' >> ~/.zshrc

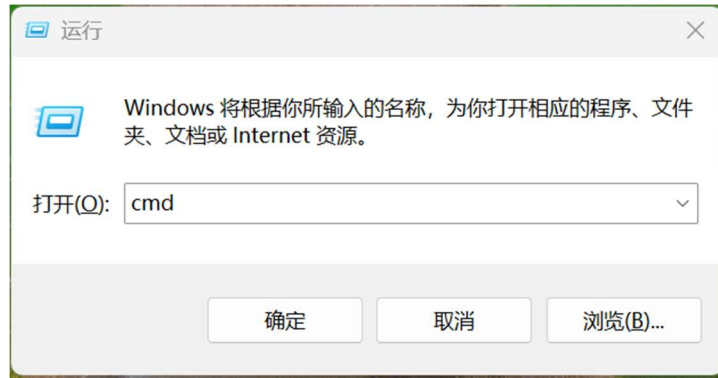
source ~/.zshrc
```

设置环境变量后，建议重启系统。重启系统后，可以通过下面的命令显示 PATH，确认 anaconda 相关的路径都在其中。

```
echo $PATH
```

## 2.4. 验证 python 环境的设置

Win +R 打开“运行”窗口，输入 cmd，打开命令提示符窗口：



```
C:\WINDOWS\system32\cmd. x + v - □ ×
Microsoft Windows [版本 10.0.26200.6899]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Nano-Admin>python
Python 3.13.5 | packaged by Anaconda, Inc. | (main, Jun 12 2025, 16:37:03
) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World!')
Hello World!
>>> exit(0)

C:\Users\Nano-Admin>
```

在打开的命令提示符窗口中，依次输入：

```
python
print('Hello World')
exit(0)
```

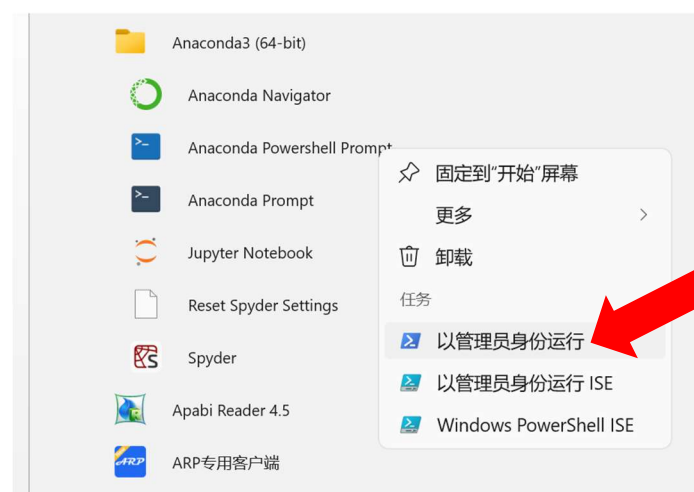
执行结果如上图所示，则表示 python 环境正常。

### 3. 配置 pip 使用清华源

在 anaconda 的使用过程中，经常需要使用 pip 下载各种模块资源。如果每次都需从国外下载的话，速度慢，并且有的时候下载会被中断。因此，建议在安装 anaconda 后，把 pip 的下载源改为国内。

**这一步骤，强烈建议大家一定要做！**

在 windows 环境下，右键点击“Anaconda Powershell Prompt”，然后“以管理员身份运行”。



在命令窗口中，依次输入下面两个命令：

```
pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple/
```

```
pip config set install.trusted-host pypi.tuna.tsinghua.edu.cn
```

为了验证设置是否成功，可以输入：

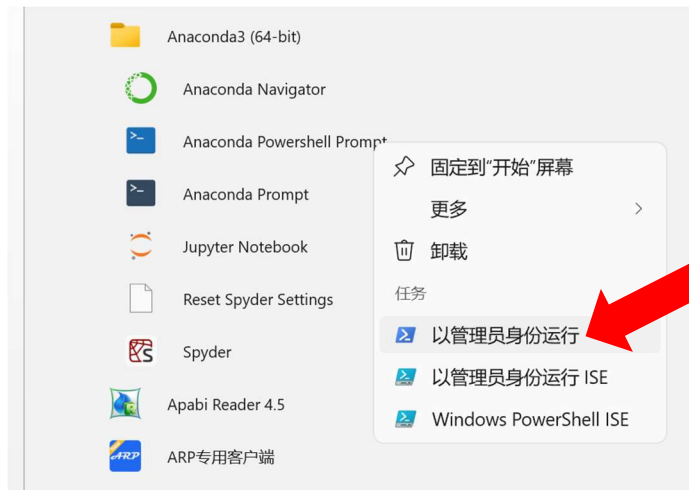
```
pip config list
```

如果输出中包含 `index-url = https://pypi.tuna.tsinghua.edu.cn/simple` 则表明配置成功。

```
Anaconda Powershell Prompt x + v - □ x
(base) PS C:\Users\Tingshao Zhu> pip config list
(base) PS C:\Users\Tingshao Zhu> pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple/
Writing to C:\Users\Tingshao Zhu\AppData\Roaming\pip\pip.ini
(base) PS C:\Users\Tingshao Zhu> pip config set install.trusted-host pypi.tuna.tsinghua.edu.cn
Writing to C:\Users\Tingshao Zhu\AppData\Roaming\pip\pip.ini
(base) PS C:\Users\Tingshao Zhu> pip config list
global.index-url='https://pypi.tuna.tsinghua.edu.cn/simple/'
install.trusted-host='pypi.tuna.tsinghua.edu.cn'
(base) PS C:\Users\Tingshao Zhu>
```

## 4. 安装 jieba 和 emoji 包

在 windows 环境下，右键点击“Anaconda Powershell Prompt”，并以管理员身份运行。



在命令窗口中，依次输入下面两个命令：

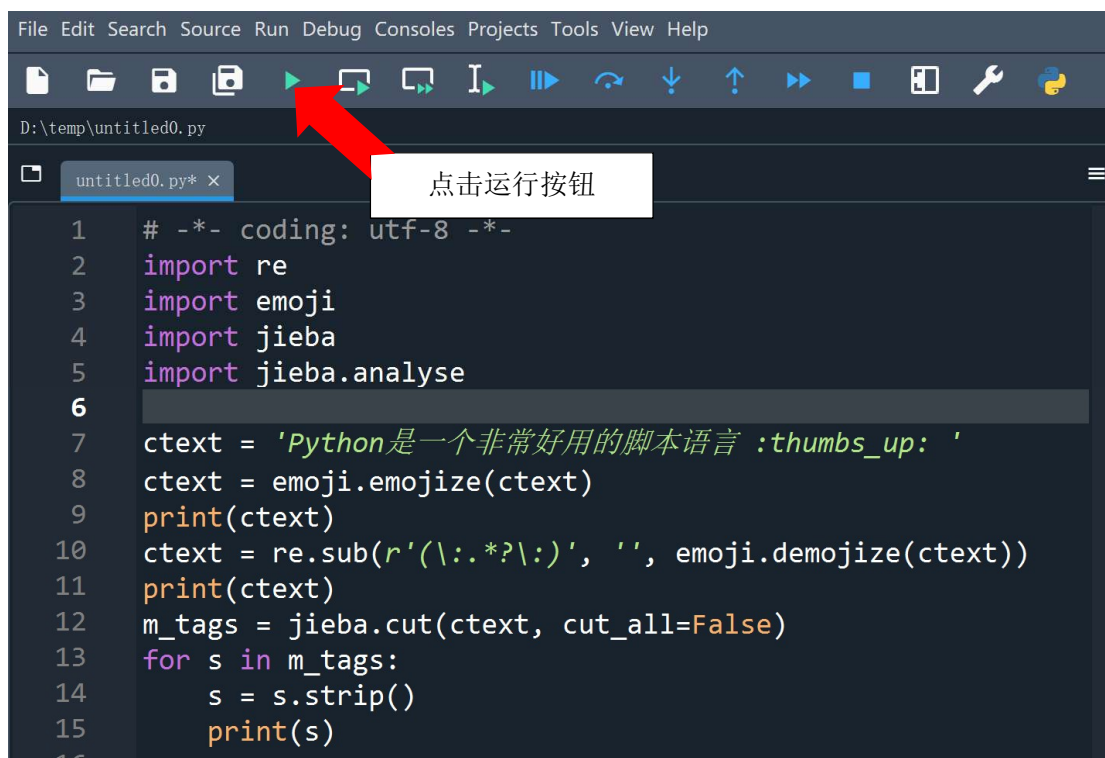
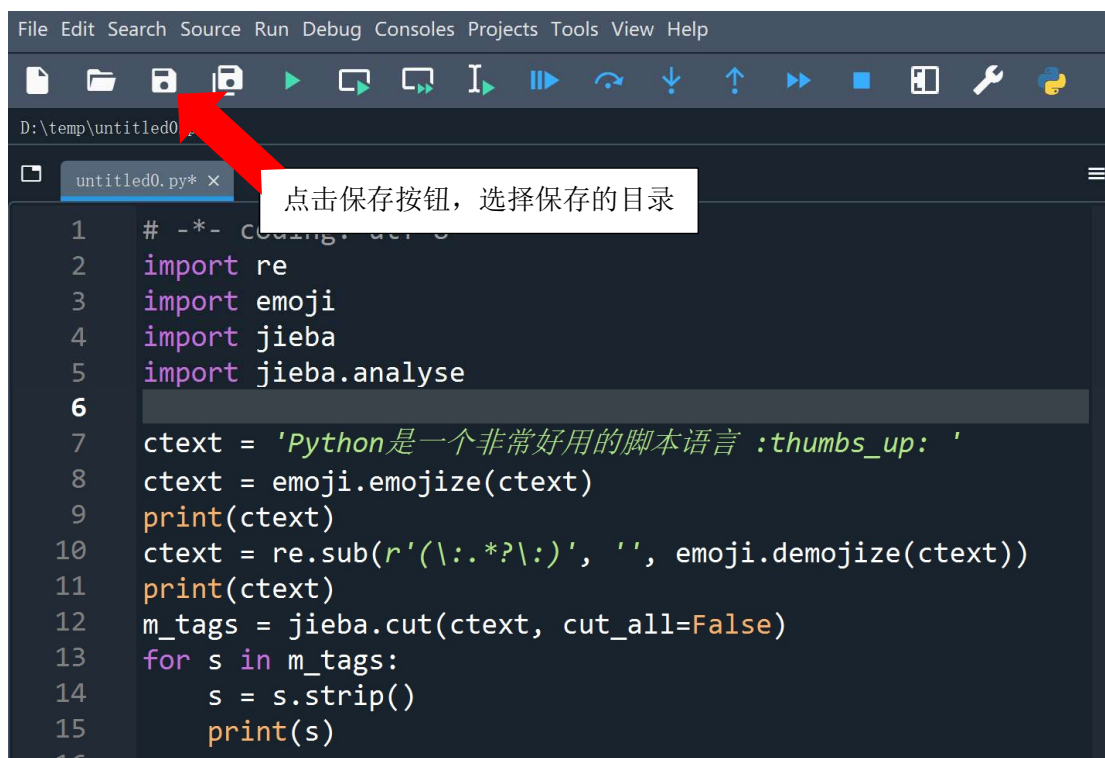
```
pip install jieba
```

```
pip install emoji
```

在成功安装 jieba 和 emoji 后，打开 spyder，把下面的程序粘贴保存到文件并运行：

```
# -*- coding: utf-8 -*-
import re
import emoji
import jieba
import jieba.analyse

ctext = 'Python 是一个非常好用的脚本语言 :thumbs_up:'
ctext = emoji.emojize(ctext)
print(ctext)
ctext = re.sub(r'(\:.*?:)', '', emoji.demojize(ctext))
print(ctext)
m_tags = jieba.cut(ctext, cut_all=False)
for s in m_tags:
    s = s.strip()
    print(s)
```



运行结果在右下角：

```
Console 1/A x
In [1]: runfile('C:/temp/untitled0.py', wdir='C:/temp')
Building prefix dict from the default dictionary ...
Loading model from cache C:
\Users\TINGSH~1\AppData\Local\Temp\jieba.cache
Python是一个非常好用的脚本语言 🍊
Python是一个非常好用的脚本语言
Loading model cost 0.527 seconds.
Prefix dict has been built successfully.
Python
是
一个
非常
好用
的
```

出现上述运行结果，说明 jieba 和 emoji 安装成功！

## 5. 下载安装 ollama

林萃分析系统需要调用本地部署的大语言模型。为了方便部署，我们使用 ollama 进行大语言模型的安装和部署。

ollama 是一个开源工具，主要用于在本地计算机上高效部署和运行大语言模型(Large Language Models, LLMs)。ollama 提供了丰富的功能和特性，通过简化的命令行和可视化界面，帮助用户快速下载、管理和交互多种大语言模型(如 Qwen3、DeepSeek、Llama 等)，无需依赖云端服务，特别适合开发者、研究人员及对数据隐私有高要求的用户。

### 5.1. 安装 ollama

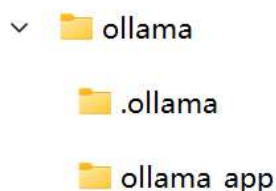
一般情况下，ollama 安装在当前用户帐号下，例如："C:\Users\Tingshao Zhu\AppData\Local\Programs\Ollama"。这使得其他用户在使用的时候，很不方便。为此，建议将 ollama 安装在一个指定的目录下。请按照下列的顺序操作：

#### (一) . 卸载已安装的 ollama 并删除 .ollama 目录

首先，请把已经安装的 ollama 卸载，不管是安装在哪个帐号下的，都全部卸载。卸载之后，请检查 c:\users 的当前用户下，删除 .ollama 目录。

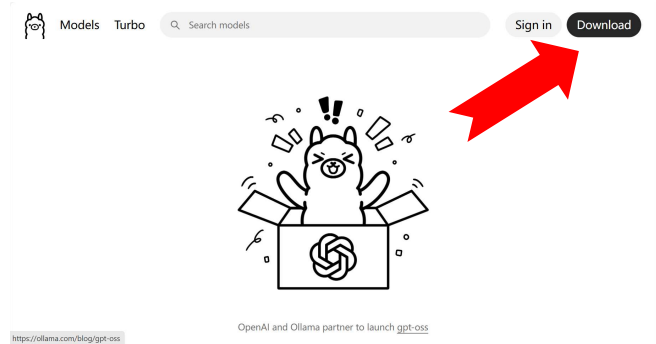
#### (二) . 新建 ollama 本地文件夹

这里我们假设把 ollama 都放在 c:\ollama 文件夹(也可以是其他地方，如 d:\ollama)，在这个文件夹下新建两个文件夹：一个是 .ollama 为了保存下载的本地大语言模型，另一个 ollama\_app 做为 ollama 的安装目录。



#### (三) . 下载并安装 ollama

在浏览器上打开下面网站，<https://www.ollama.com/>，点击“Download”按钮下载 ollama 的安装程序。



这里，我们假设将安装程序下载到 `c:\temp`，然后在**管理员帐号**下，打开“命令提示符”窗口（在 `win+R` 中输入 `CMD` 就可以打开命令提示符窗口），然后在命令行输入：

```
cd c:\temp  
OllamaSetup.exe /DIR="c:\ollama\ollama_app"
```

在弹出的 ollama 安装的窗口中，点击 “Install”，完成安装。然后，在 “`c:\users\当前管理员`” 文件夹下，删除 `.ollama` 目录。

## 5.2. 设置 windows 的系统变量

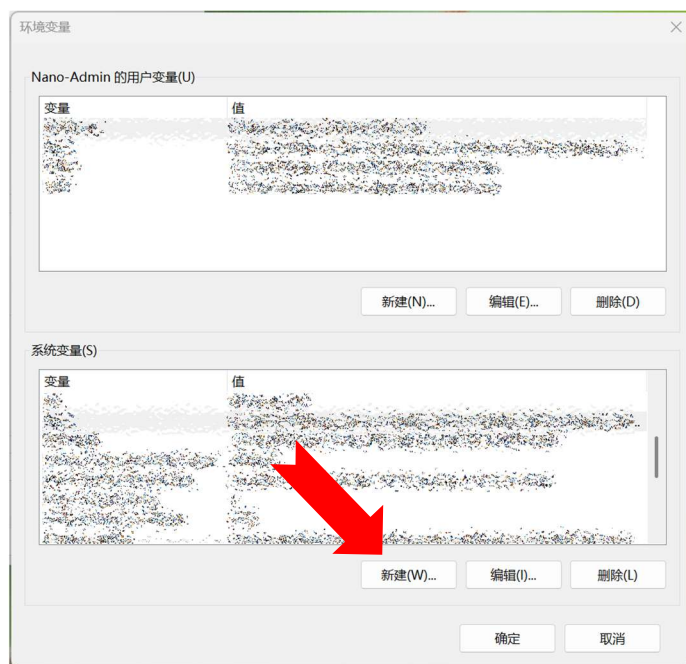
在**管理员帐号**下，windows 环境下，在“系统信息”，点击“高级系统设置”。



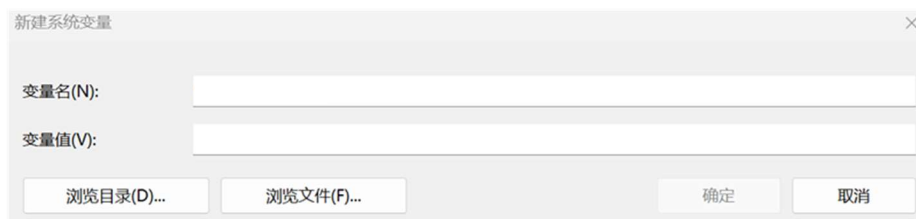
在“高级”标签下，点击“环境变量 ...”



然后，在“系统变量”中新建变量



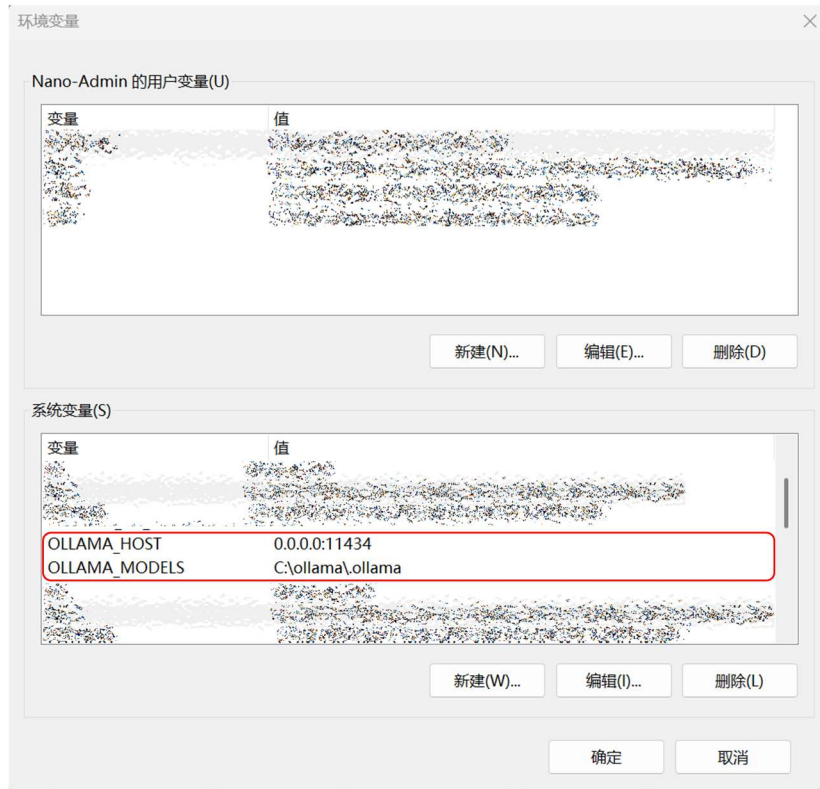
在“新建系统变量”的窗口中，分别输入变量名和变量值，然后点击“确定”。



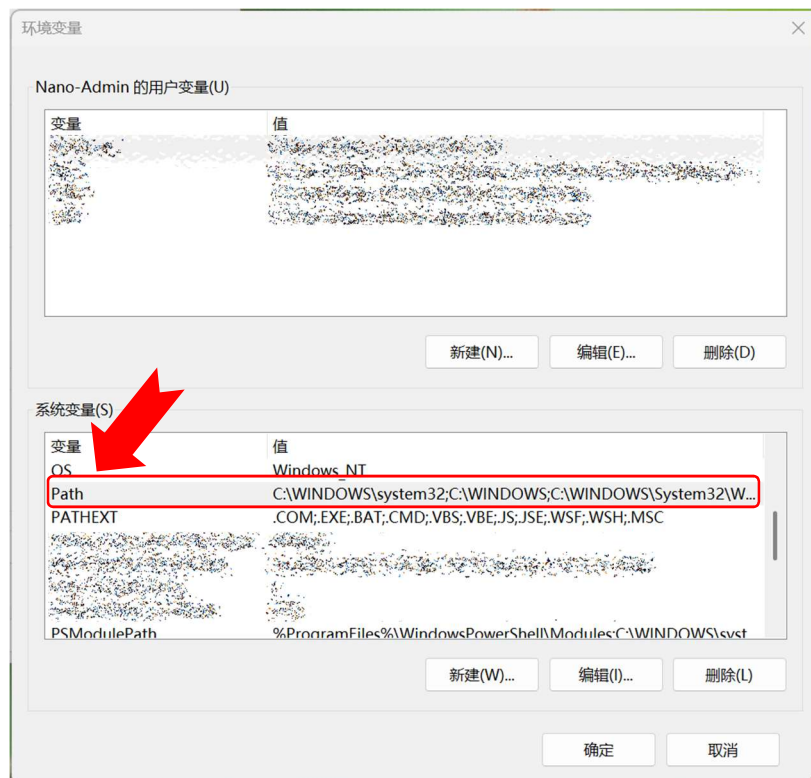
新建两个系统变量，

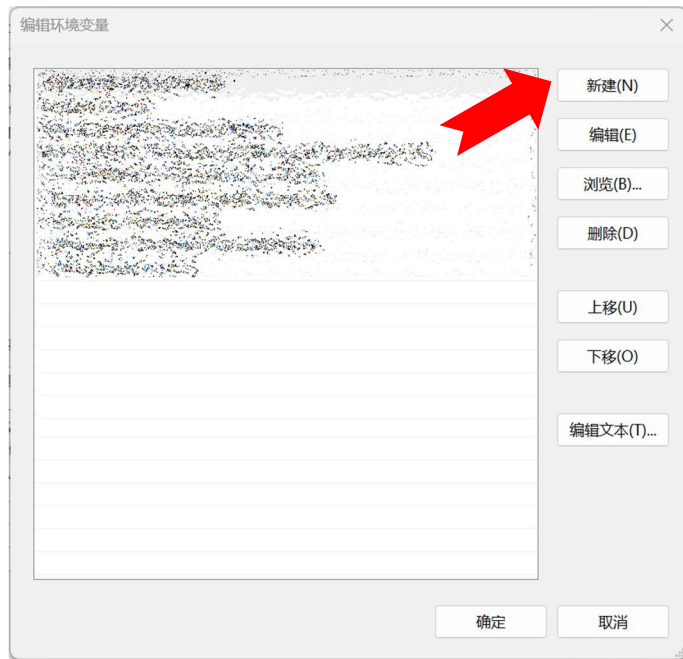
变量名：OLLAMA\_HOST，变量值：0.0.0.0:11434

变量名：OLLAMA\_MODELS，变量值：c:\ollama\ollama

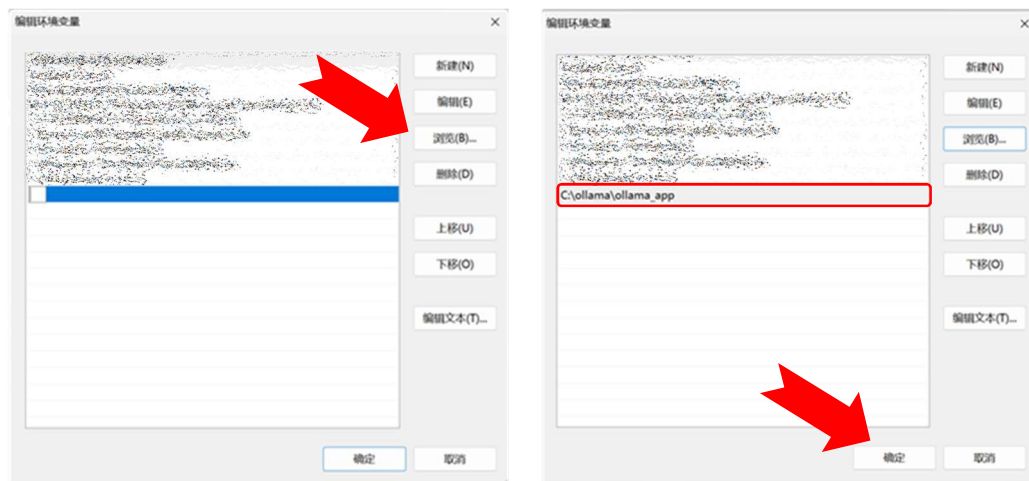


然后，双击 系统变量 中的 Path:

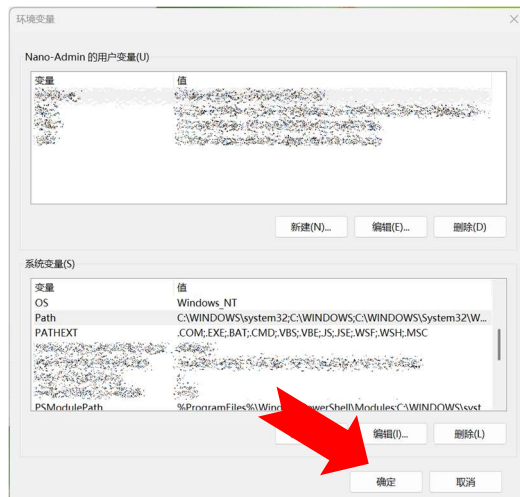




点击“新建”后，系统会增加一个条目，这个时候可以点击“浏览”，定位到 ollama\_app，如上则为: c:\ollama\ollama\_app



点击“确定”，关闭“编辑环境变量”窗口。



然后，点击“确定”，就可以关闭“环境变量”的设置。为了确保系统变量的设置有效，建议在设置以后，重启系统。

### 5.3. 设置 Mac 上的系统变量

在 Mac 环境下，可以通过下面方法设置 ollama 需要的系统变量，或者您熟悉的命令进行系统变量和搜索路径的设置。

```
echo 'export OLLAMA_HOST="0.0.0.0:11434"' >> ~/.zshrc
echo 'export OLLAMA_MODELS="/path/to/.ollama"' >> ~/.zshrc
echo 'export PATH="/path/to/ollama_app:$PATH"' >> ~/.zshrc

source ~/.zshrc
```

设置环境变量后，建议重启系统。

### 5.4. 通过 ollama 下载 qwen3:0.6b 大模型

在安装并设置好环境变量后，可以在浏览器中输入：<http://localhost:11434>

- 如果打开的网页中显示：Ollama is running，表明 ollama 正常运行。
- 如果显示网页不可访问，说明 ollama 的 serve 没有启动，需要先启动 ollama 服务，可以打开“命令提示符”窗口（在搜索框中输入 CMD 就可以打开命令提示符窗口），在命令行依次输入：

```
cd c:\ollama\ollama_app
ollama serve
```

在启动 ollama 服务后，可以保留命令窗口，用浏览器访问 <http://localhost:11434>，以确保 ollama 服务正常运行。

在启动 ollama 服务后，我们可以拉取并运行大模型。在“命令提示符”窗口依次输入拉取和运行命令。大模型运行后，就可以接收输入，在“>>>”后输入文本，就可以与大模型进行对话。如果想退出对话，输入“/bye”即可。

```
命令提示符
Microsoft Windows [版本 10.0.26100.6725]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Tingshao Zhu>cd c:\ollama\ollama_app

c:\ollama\ollama_app>ollama pull qwen3:0.6b
pulling manifest
pulling 7f4030143c1c: 100% ██████████ 522 MB
pulling ae370d884f10: 100% ██████████ 1.7 KB
pulling d18a5cc71b84: 100% ██████████ 11 KB
pulling cff3f395ef37: 100% ██████████ 120 B
pulling b0830f4ff6a0: 100% ██████████ 490 B
verifying sha256 digest
writing manifest
success

c:\ollama\ollama_app>ollama run qwen3:0.6b
>>> 你好
Thinking...
好的，用户发来的是“你好”，我需要回应。首先，用户可能是在测试我的反应，或者想进行一些互动。我应该保持友好和开放的态度，比如打招呼，同时可以询问一些问题，促进交流。要注意语气要自然，不要太生硬，让用户感到舒适。此外，还要考虑用户可能的意图，是想聊天、寻求帮助，还是其他用途。最后，确保回应简洁明了，符合自然的交流方式。
...done thinking.

你好！有什么有趣的事情想和你聊聊吗？ 😊

>>> /bye

c:\ollama\ollama_app>
```

也可以使用 ollama 窗口，在 ollama\_app 目录下，双击“ollama app.exe”

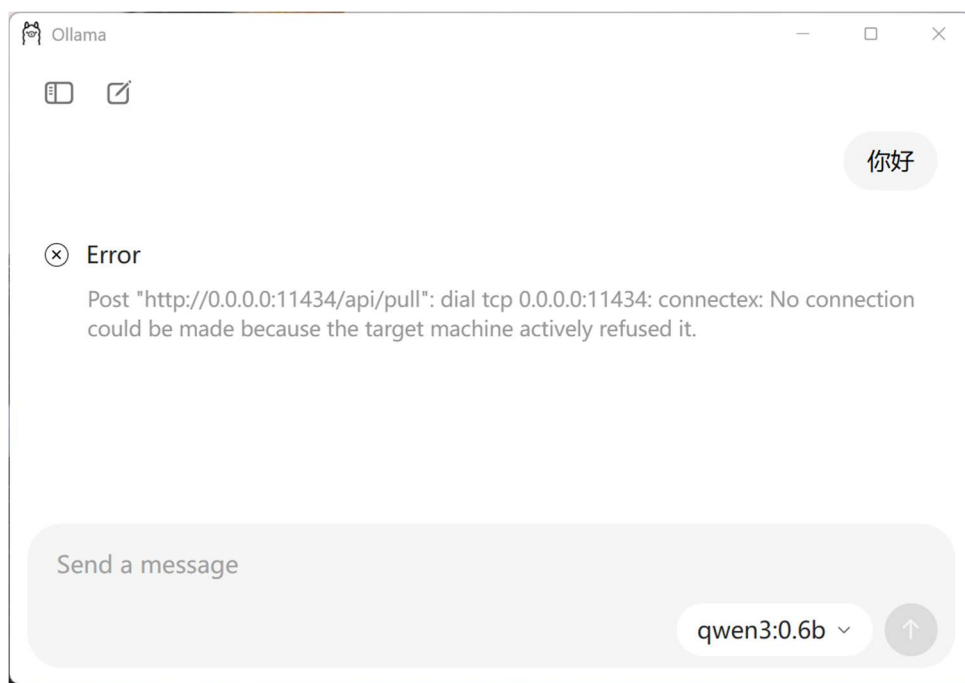


出现这样的回复信息，说明大语言模型成功部署并可以使用。

我们在资料 U 盘的 llm\_models 中已经下载了两个 0.6B 的模型，一个是 qwen3-embedding-0.6B；另外一个 qwen3-0.6b，可以把这个目录下的文件夹和文件，都拷贝到 c:\ollama\ollama 的目录下就可以，**也可自行下载！**

## 5.5. 无网情况下使用 ollama

有的时候，在没有互联网连接的情况下，使用大语言模型，就会出现下面的错误信息。



在这种情况下，打开一个命令提示符窗口，输入：



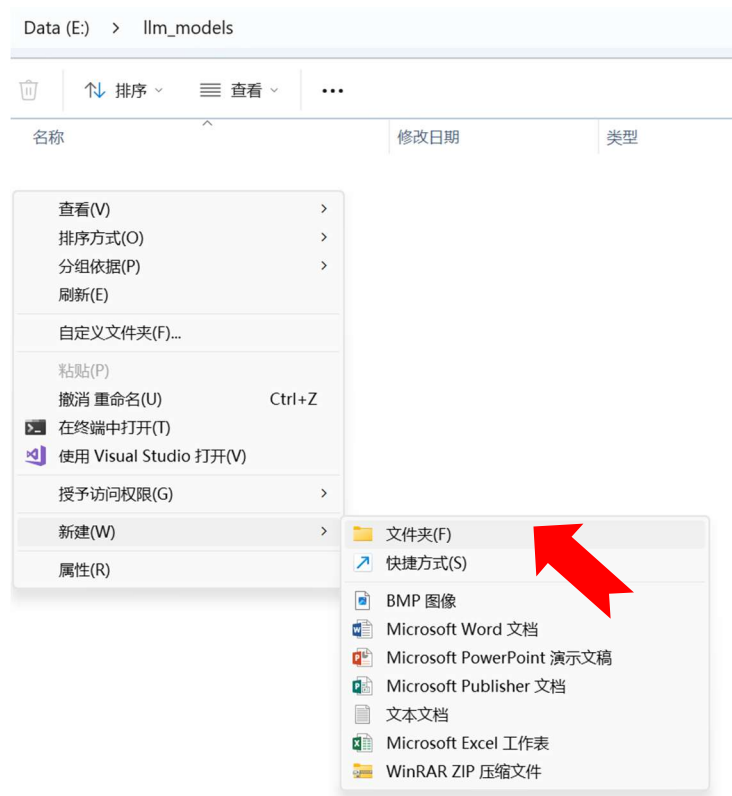
然后，在 ollama 的窗口中，就可以对话了。

在无网的情况下使用大语言模型，需要先启动大模型的服务，`ollama serve`，然后就可以通过窗口、命令行或 `python` 程序调用大模型的服务。

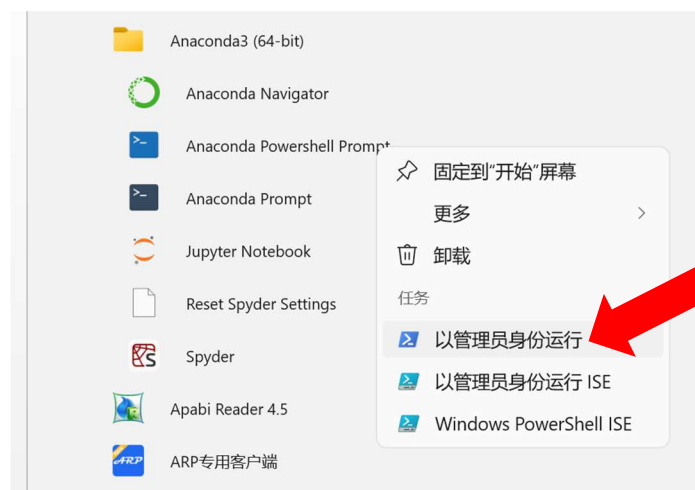
**注意：** `ollama` 可以在一个模型文件夹下保存多个大语言模型，可以打开 `ollama.com` ，查询可以下载的大模型，然后使用 “`ollama pull XXXX:YY`” 命令下载大模型文件到 `ollama\.ollama` 的文件夹下。可以使用 “`ollama list`” 查看系统已经下载的大模型。如果需要运行某个特定版本的大模型，可以在窗口上选择，或者命令行中指定，如：`ollama run qwen2:32b`

## 6. Modelscope 自行下载 qwen3-embedding-0.6B

我们在资料 U 盘上放置了下载的 qwen3 大模型，qwen3-embedding-0.6B。用户也可以通过 modelscope 自行下载，过程如下。



在本地的大模型的目录下，新建一个目录，qwen3-embedding-0.6B。右键点击“Anaconda Powershell Prompt”，并以管理员身份运行。



如果计算机里还没有安装 modelscope，可以在命令行窗口，输入下面的命令：

`pip install modelscope`

```
(base) PS C:\WINDOWS\system32> pip install modelscope
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple/
Collecting modelscope
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/a2/f1/f3e84271e115da91d9520e65846a7176ad356f71d021c134f872687d4ffc/modelscope-1.29.2-py3-none-any.whl (5.9 MB)
    5.9/5.9 MB 8.2 MB/s eta 0:00:00
Requirement already satisfied: filelock in c:\anaconda3\lib\site-packages (from modelscope) (3.13.1)
Requirement already satisfied: requests>=2.25 in c:\anaconda3\lib\site-packages (from modelscope) (2.31.0)
Requirement already satisfied: setuptools in c:\anaconda3\lib\site-packages (from modelscope) (68.2.2)
Requirement already satisfied: tqdm=4.64.0 in c:\anaconda3\lib\site-packages (from modelscope) (4.65.0)
Requirement already satisfied: urllib3=1.26 in c:\anaconda3\lib\site-packages (from modelscope) (2.0.7)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\anaconda3\lib\site-packages (from requests>=2.25->modelscope) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\anaconda3\lib\site-packages (from requests>=2.25->modelscope) (3.4)
Requirement already satisfied: certifi=2017.4.17 in c:\anaconda3\lib\site-packages (from requests>=2.25->modelscope) (2024.2.2)
Requirement already satisfied: colorama in c:\anaconda3\lib\site-packages (from tqdm=4.64.0->modelscope) (0.4.6)
Installing collected packages: modelscope
Successfully installed modelscope-1.29.2
```

上面表示成功安装。

如果系统安装了 modelscope，在命令行窗口，输入下面的命令，下载大模型。

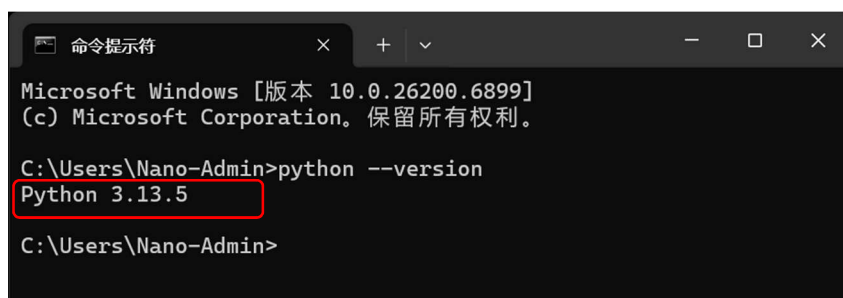
```
modelscope download --model Qwen/Qwen3-Embedding-0.6B --local_dir
'C:\workshop\llm_models\qwen3-embedding-0.6B'
```

## 7. 安装大模型相关模块

为了运行大模型，需要下载并安装相关的包。一般情况下，都可以通过 `pip install` 安装，也可以下载.whl 文件以后，从 whl 安装。

### 7.1. 安装 gensim

在数据分析过程中，我们会使用 `gensim` 进行文本分析。在管理员身份运行的命令提示符窗口中，请输入：`python --version`，以获取当前的 python 版本。



```
命令提示符
Microsoft Windows [版本 10.0.26200.6899]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Nano-Admin>python --version
Python 3.13.5

C:\Users\Nano-Admin>
```

打开下面这个网页，从中选择适合 python 版本的 `gensim` 的 whl 文件。

<https://pypi.org/project/gensim/#files>



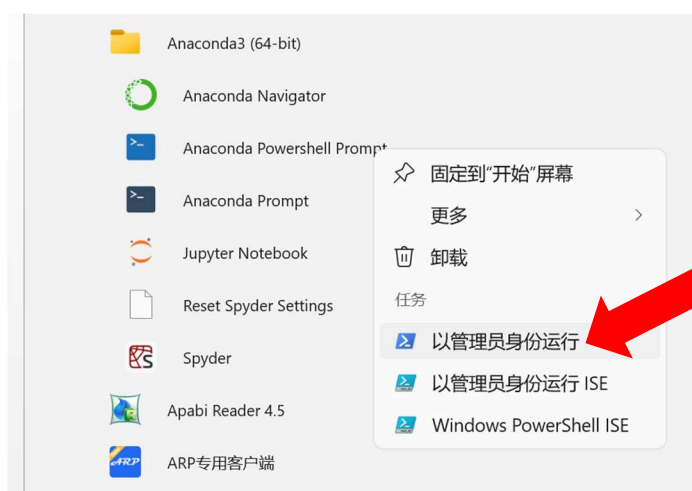
根据 Python 的版本（3.13.5），选择适合 cp313-cp313 的 whl，windows 系统就可以下载：`gensim-4.4.0-cp313-cp313-win_amd64.whl`

我们可以下载到本地，如：`c:\temp` 目录下，或者也可以把资料 U 盘中的 whl 文件拷贝到 `c:\temp` 下。在管理员身份运行的命令窗口中，请输入：

```
pip install c:\temp\gensim-4.4.0-cp313-cp313-win_amd64.whl
```

## 7.2. 安装 `sentence_transformers`、`ollama` 和 `openai`

在 windows 环境下，右键点击“Anaconda Powershell Prompt”，并以管理员身份运行。



在命令窗口中，依次输入下面两个命令：

```
pip install sentence_transformers
```

```
pip install ollama
```

```
pip install openai
```

## 7.3. Python 降级

如果需要降级，例如从 3.13 降到 3.12，可以管理员身份运行 Anaconda Powershell Prompt，然后在命令行运行：

```
conda create -n py312 python=3.12 anaconda  
conda activate py312
```

然后，

```
python --version
```

确认已经切换为 3.12

## 8. 获取阿里云通义千问 API Key

打开阿里云百炼，<https://bailian.console.aliyun.com/>，



点击“立即开通”。请按照网站要求，完成注册流程。登录进系统后，在 API Key 管理中，创建 API-KEY：



拷贝 API Key



一般不建议把 API Key 直接放在程序里，容易被其他人偷用。建议将 API Key 设置为环境变量。

### 8.1. 设置 Win 环境变量

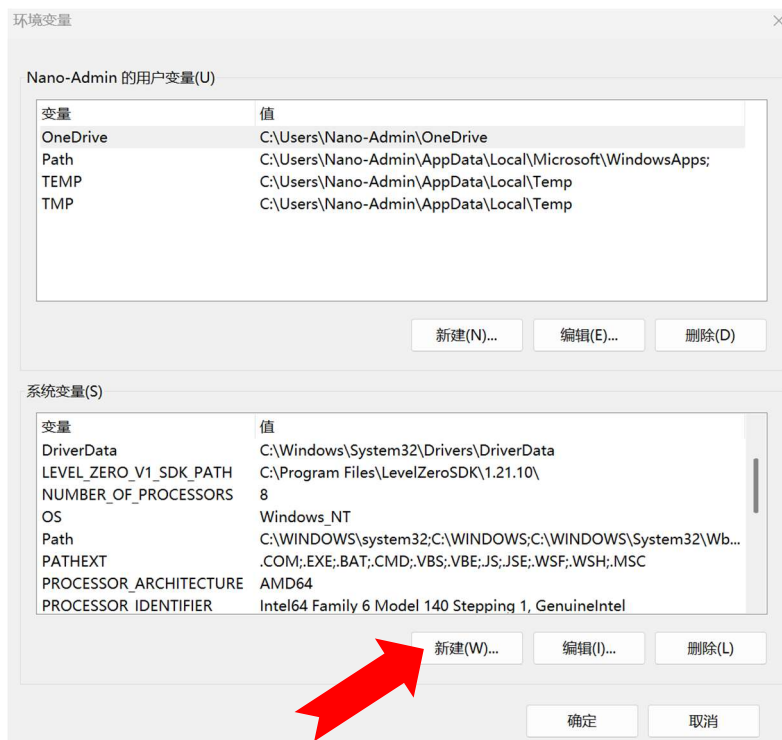
在 windows 环境下，在“系统信息”，点击“高级系统设置”。

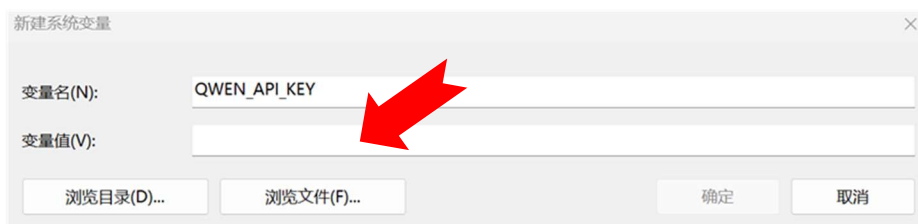


在“高级”标签下，点击“环境变量 ...”



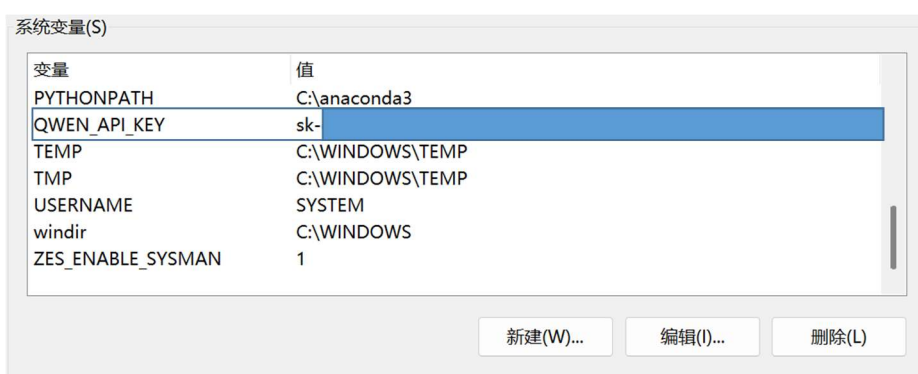
然后，在“系统变量”中新建变量



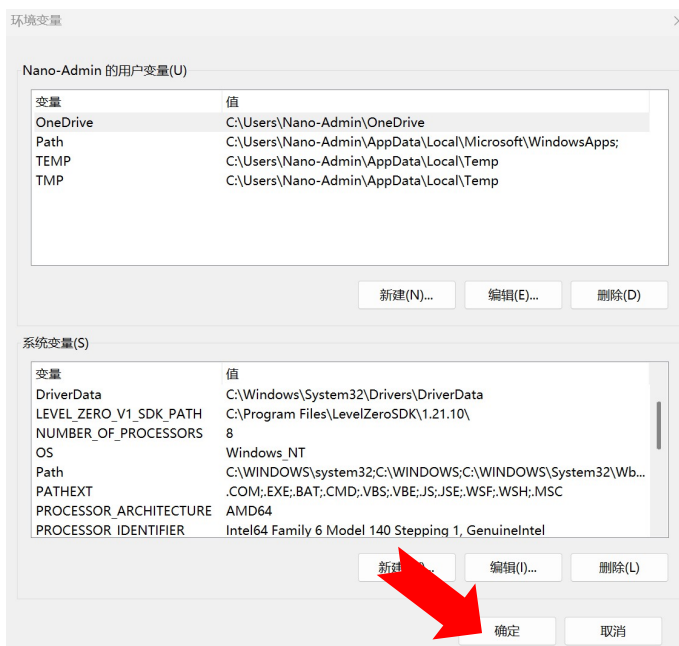


变量名为 QWEN\_API\_KEY，在“变量值”的输入框中，输入您的 API Key，并点击“确定”按钮。

在新建这个变量后，可以在列表框中看到它：



然后，点击“确定”后，就可以关闭“环境变量”的设置。



为了确保系统变量的设置有效，建议在设置以后，重启系统。

为了测试环境变量是否设置成功，把下面的程序粘贴保存到文件并运行：

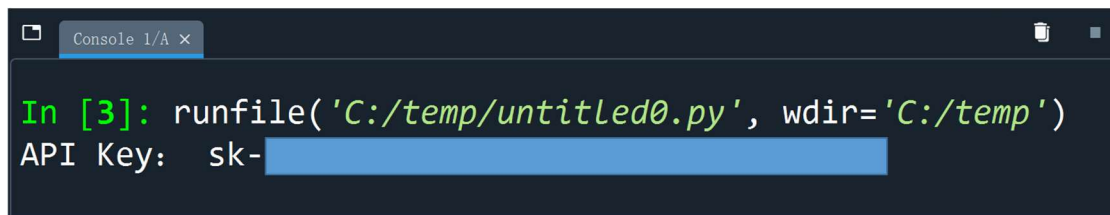
```
# -*- coding: utf-8 -*-

import os

# 获取环境变量
access_key_id = os.getenv('QWEN_API_KEY')

print("API Key: ", access_key_id)
```

运行后，能看到自己设置的这个环境变量，则说明已成功设置。



```
Console 1/A x
In [3]: runfile('C:/temp/untitled0.py', wdir='C:/temp')
API Key: sk- [REDACTED]
```

## 8.2. 设置 Mac 的环境变量

在"启动台"->"其他"中打开"终端"(Terminal)应用。

执行以下命令，用系统自带的 nano 编辑器打开配置文件：

```
nano ~/.zshrc
```

如果文件是空的或有其他内容，在文件末尾另起一行，添加以下内容。请务必将 `your_api_key_here` 替换为你真实的 API Key.

```
export QWEN_API_KEY="your_api_key_here"
```

修改后，保存并退出：

保存：按 `Ctrl+X`。确认：按 `Y`。退出：按 `Enter`

然后，运行以下命令，让刚才的改动立即在当前终端中生效，而无需重启。

```
source ~/.zshrc
```