附件 2

现场任务挑战主题

一、主题内容

（一）AI 水生态智能诊断

生态环境数字化治理与生物多样性保护是国家“十五五”生态治理的重点发展方向，AI 智能分析技术凭借高效解析、精准研判的优势，将成为水生态研究与治理的核心技术手段。在水环境改善与河流生态修复的发展背景下，AI 技术可进行水环境与生物多样性耦合关系的量化分析，助力破解河流生态风险识别难、治理成效评估不精准的难题。本任务立足环境工程建设需求，依托现代生态学原理与多源水环境监测数据，机器学习、计算机建模等技术方法，系统探究水环境动态演变对物种组成、结构特征及耐污适应性的影响机制，挖掘动态响应的内在关联，以此实现生态风险因子识别与生态工程治理成效的科学化、量化评估。

（二）智能追日

根据“十五五”规划纲要，积极稳妥推进和实现碳达峰。统筹发展和减排，扎实开展碳达峰行动，加快产业结构、能源结构、交通运输结构等调整优化，确保如期实现碳达峰目标。全面实施碳排放总量和强度双控制度。推动重点领域节能降碳。提升应对全球气候变化能力。我国幅员辽阔，具备十分充足的光照资源。光伏发电做为我国清洁能源中的重要组成部分，其发电效率的提

升被列为国家重点发展项目。本任务基于电路、机械结构设计、软件编程等技术，旨在提升光伏板对准的精度，从而提高发电效率。

二、硬件要求

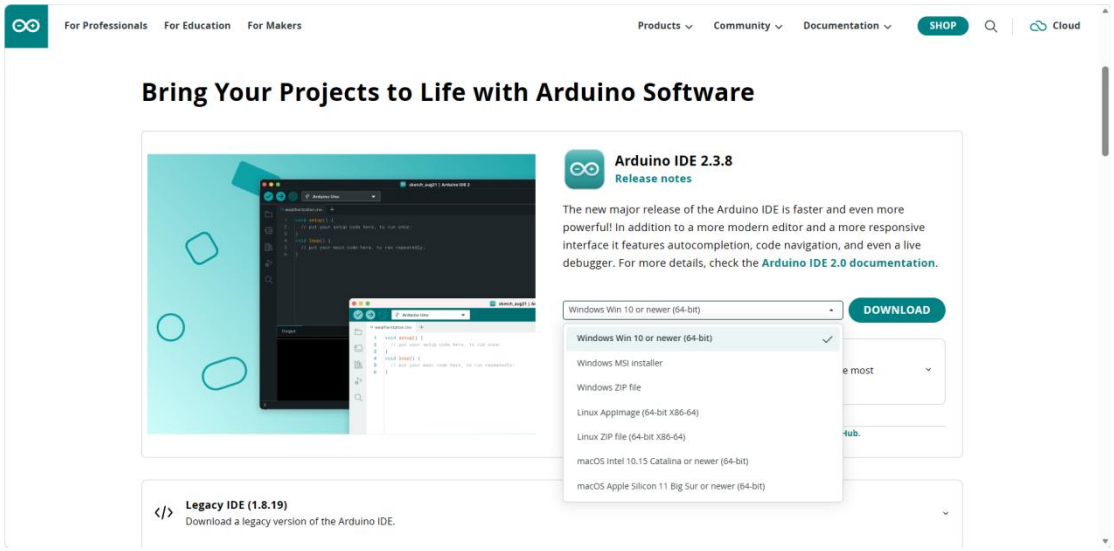
请选手自带电脑、鼠标、电源、插排。若电脑无 usb 口需要自带 typec-typec 数据线。

三、软件要求

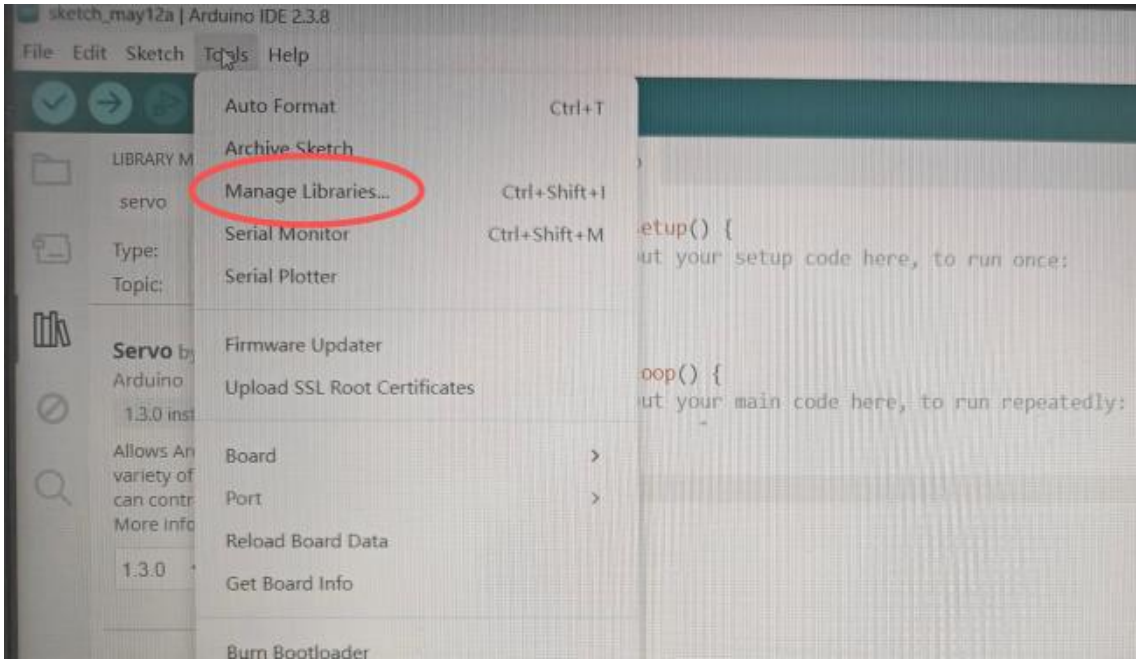
选手需要提前在电脑上下载好 Arduino 编程软件、Python、PyCharm 和常用科学计算包，检查环境是否可用，且在软件内下载好相关库函数。

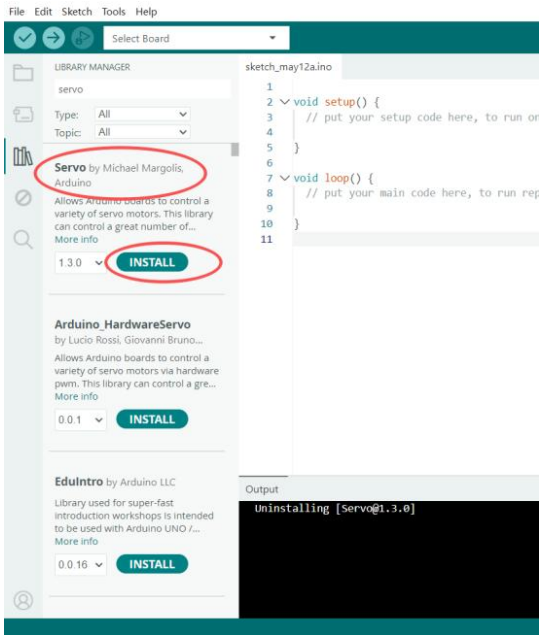
（一）Arduino

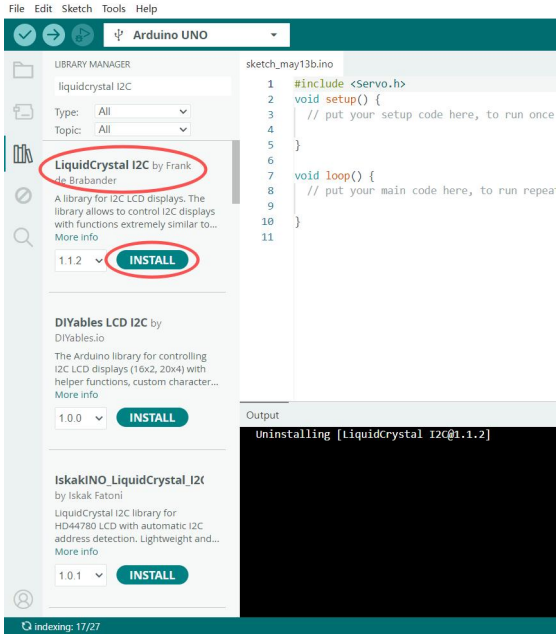
1. Arduino 下载地址：arduino.cc/en/software。根据电脑系统选择合适的软件（如图所示）。



2. 下载并安装文件后在“tools”中选择“manage libaries”，搜索并安装 “servo” 和 “liquidcrystal I2C” 库函数（如图所示）。







（二）Python

Python + PyCharm 环境安装指导手册

适用对象：中学生、Python 初学者
适用系统：Windows 10、Windows 11、macOS Intel、macOS Apple Silicon
用途：从零安装 Python、PyCharm 和常用科学计算包，并学会检查环境是否可用

目录

- 1. 需要安装什么
- 2. 推荐电脑配置
- 3. Windows 10 / Windows 11 安装步骤
- 4. macOS 安装步骤
- 5. 创建测试项目并安装 Python 第三方包
- 6. 检查 Python 包是否安装成功
- 7. 编写 hello_world.py 并运行测试
- 8. 完整安装检查清单
- 9. 常见问题
- 10. 官方参考资料

---

1. 需要安装什么

需要安装三类内容：

1. Python
2. PyCharm
3. Python 第三方包

常用第三方包如下：

```
numpy
pandas
matplotlib
scipy
scikit-learn
openpyxl
```

这些包常用于数据处理、表格读取、绘图和简单机器学习。

---

2. 推荐电脑配置

项目	最低要求	推荐配置
系统	Windows 10/11 64 位；macOS	Windows 10/11 64 位；macOS 12 或以上
CPU	4 核，x86_64 或 arm64	Intel i5 / Ryzen 5 / Apple M1 或以上
内存	8 GB	16 GB
硬盘	20 GB 可用空间	SSD，50 GB 可用空间
GPU	不需要	不需要

说明：本环境用于学习 Python、数据分析和基础机器学习，不需要独立显卡。

---

3. Windows 10 / Windows 11 安装步骤

Windows 10 和 Windows 11 的安装步骤基本相同。

3.1 下载 Python

1. 打开 Python 官网：

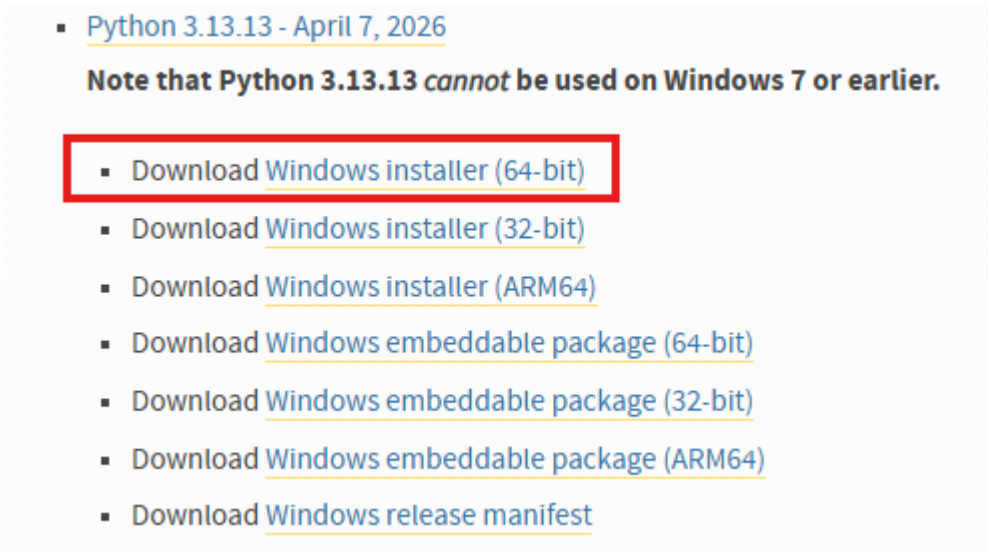
   https://www.python.org/downloads/

2. 点击页面上的 Python 下载按钮。

3. 建议选择：

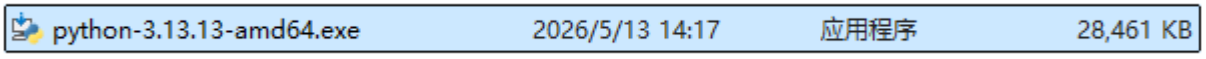
```
Python 3.13 或 Python 3.14
64-bit
```

4. 不要下载带有 `pre-release`、`alpha`、`beta`、`rc` 字样的测试版。



3.2 安装 Python

1. 双击下载好的 Python 安装包。

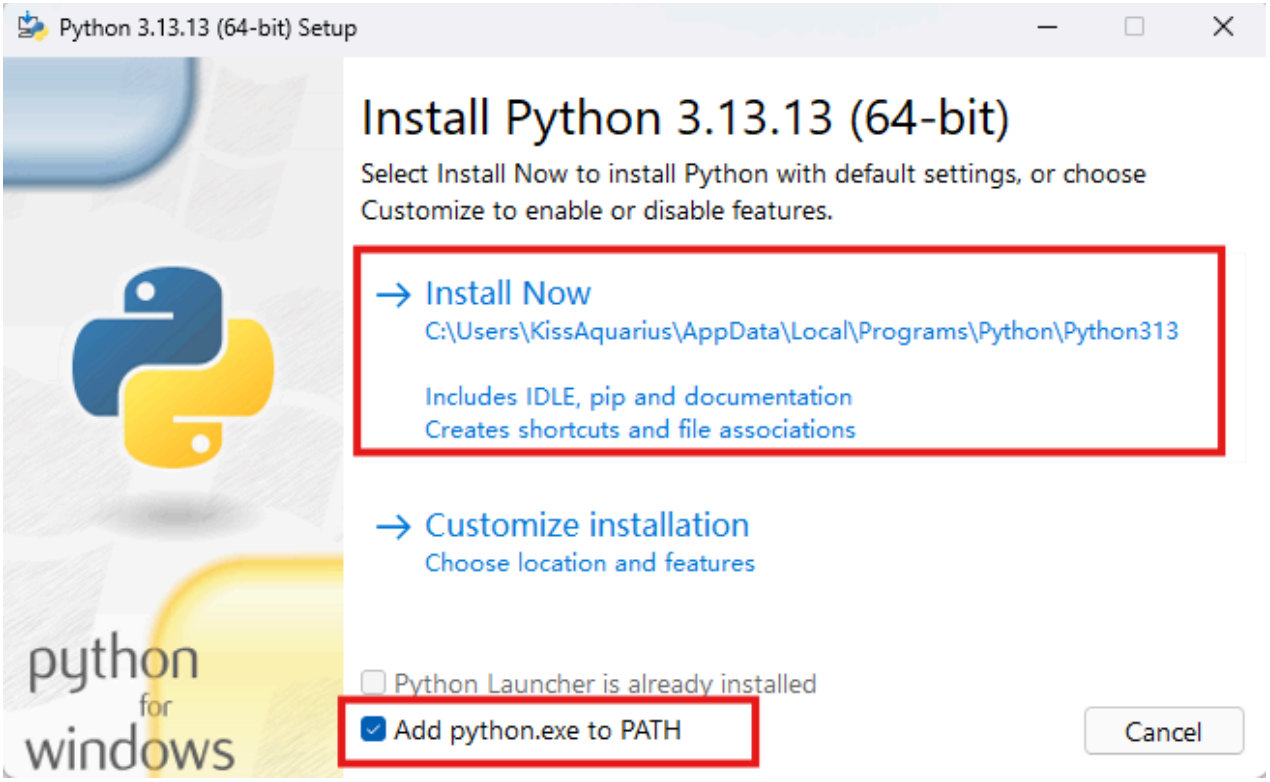


2. 在第一个安装界面，勾选：

```
Add python.exe to PATH
```

3. 点击：

```
Install Now
```



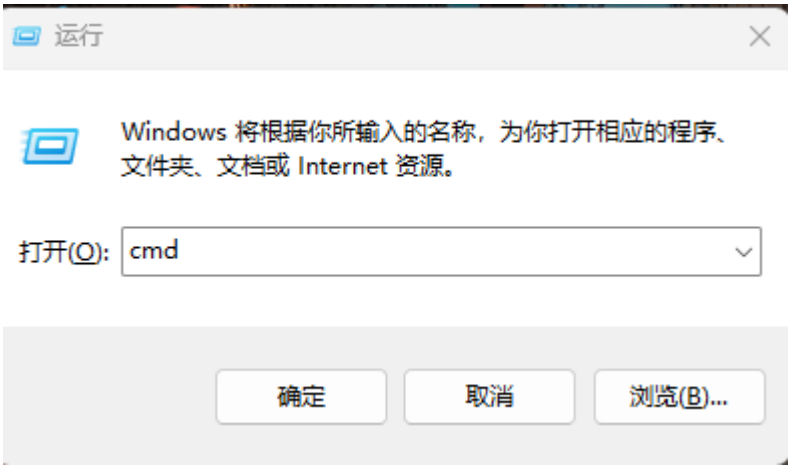
4. 等待安装完成。
5. 点击：

```
Close
```

3.3 检查 Python 是否安装成功

1. 按 `Win + R`。
2. 输入：

```
cmd
```



3. 按回车，打开命令提示符。
4. 输入：

```
python --version
```

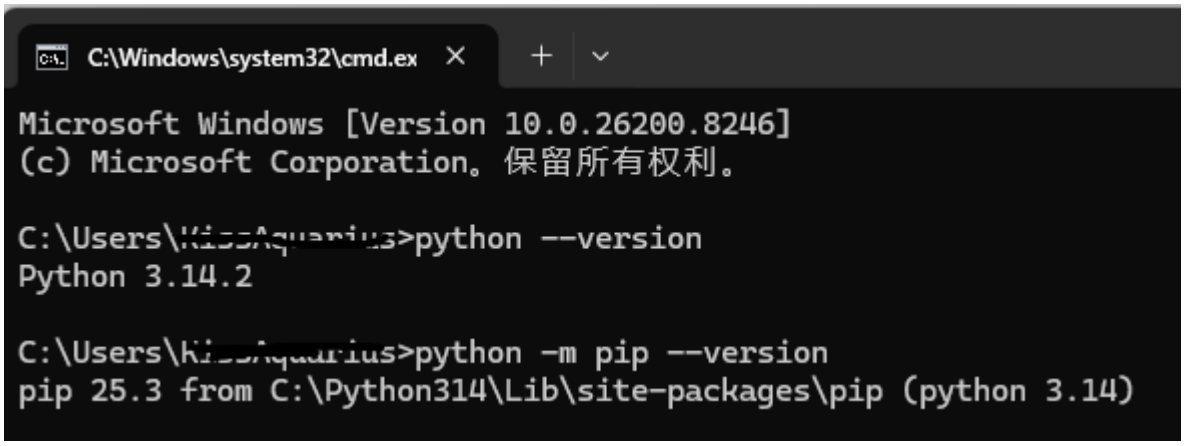
5. 如果看到类似下面的内容，说明 Python 安装成功：

```
Python 3.14.5
```

6. 再输入：

```
python -m pip --version
```

7. 如果看到 `pip` 的版本信息，说明包管理工具也安装成功。



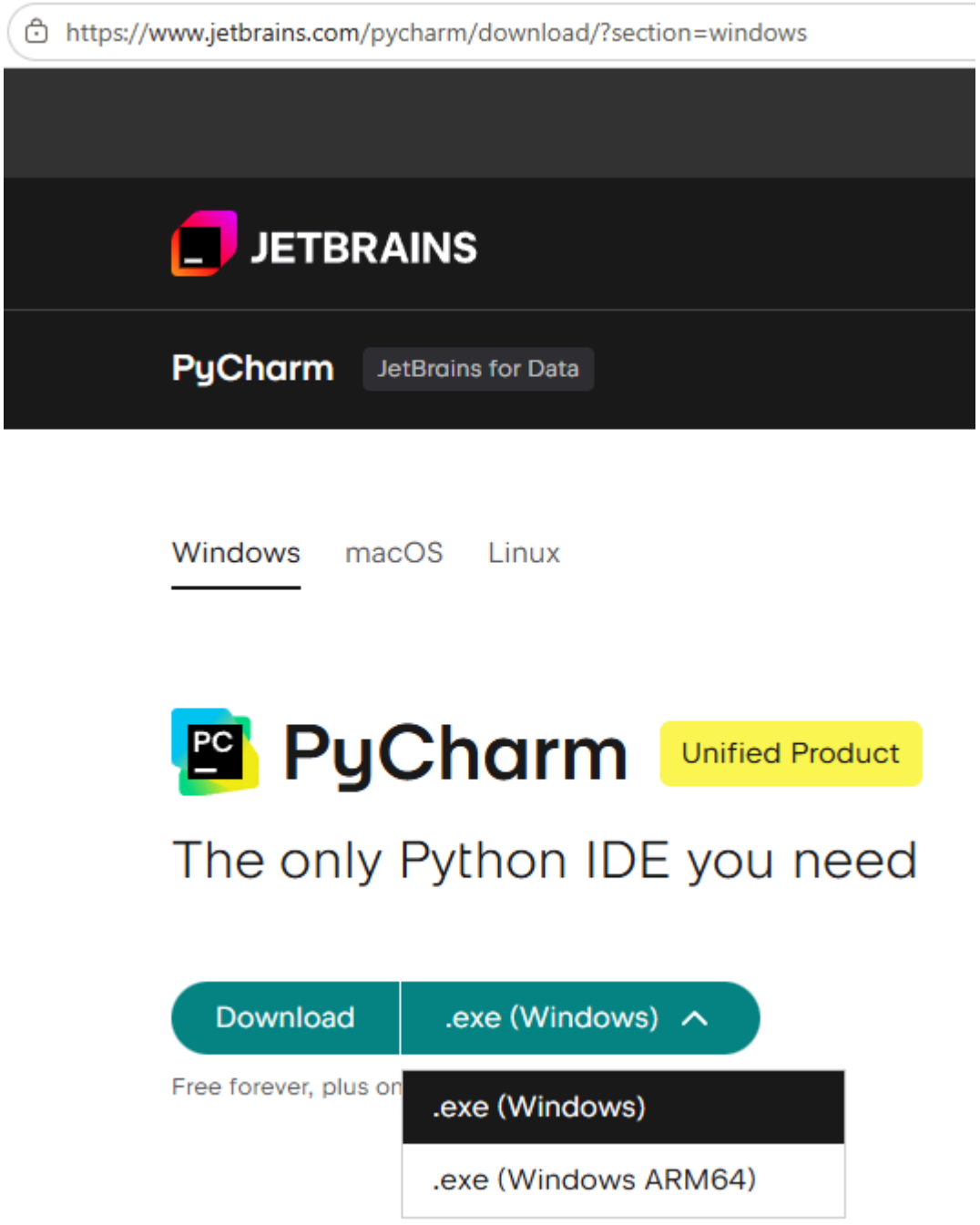
3.4 下载 PyCharm

1. 打开 PyCharm 官网：

   https://www.jetbrains.com/pycharm/download/

2. 下载 Windows 版 PyCharm。
3. 普通 Intel / AMD 电脑选择默认 Windows 版本。
4. 如果是 Windows ARM64 电脑，选择 ARM64 版本。

   不确定时，大多数 Windows 电脑都是普通 Intel / AMD 版本。



3.5 安装 PyCharm

1. 双击 PyCharm 安装包。
2. 点击：

```
Next
```

3. 建议勾选：

```
64-bit launcher
Add launchers dir to the PATH
.py
```

4. 点击：

```
Install
```

5. 安装完成后点击：

```
Finish
```

6. 第一次打开 PyCharm 时，选择默认设置即可。

说明：安装完成后重启电脑，PyCharm 不需要单独安装 Java。

---

4. macOS 安装步骤

macOS 需要先确认电脑芯片类型。

4.1 判断 Mac 芯片类型

1. 点击屏幕左上角苹果图标。
2. 点击：

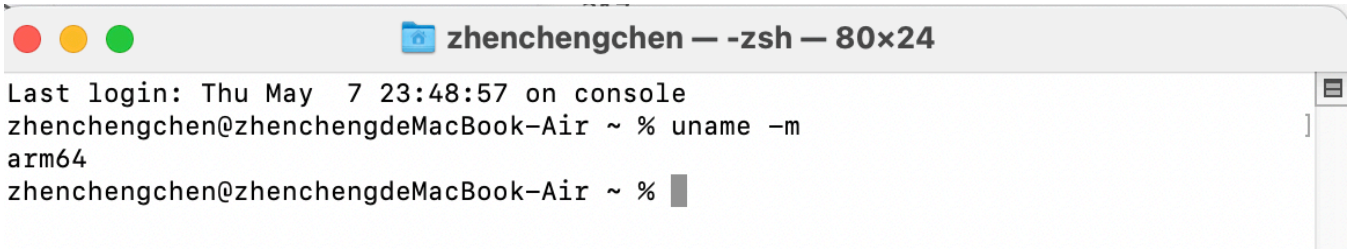
```
关于本机
```

3. 查看芯片信息：

显示内容	类型
Intel	Intel Mac
Apple M1 / M2 / M3 / M4	Apple Silicon Mac

也可以打开“终端”，输入：

```
uname -m
```



结果含义：

```
x86_64   表示 Intel Mac
arm64    表示 Apple Silicon Mac
```

4.2 下载 Python

1. 打开 Python macOS 下载页面：

   https://www.python.org/downloads/macos/

2. 下载 macOS 版 Python 安装包。
3. 建议选择：

```
Python 3.13 或 Python 3.14
macOS 64-bit universal2 installer
```

说明：`universal2` 安装包可以同时支持 Intel Mac 和 Apple Silicon Mac。

4.3 安装 Python

1. 双击下载好的 `.pkg` 文件。
2. 点击：

```
Continue
```

3. 继续按照提示操作。
4. 点击：

```
Install
```

5. 输入电脑密码。
6. 等待安装完成。

4.4 检查 Python 是否安装成功

1. command+空格，输入“终端”，回车，打开终端。
2. 输入：

```
python3 --version
```

3. 如果看到类似下面的内容，说明 Python 安装成功：

```
Python 3.14.5
```

4. 再输入：

```
python3 -m pip --version
```

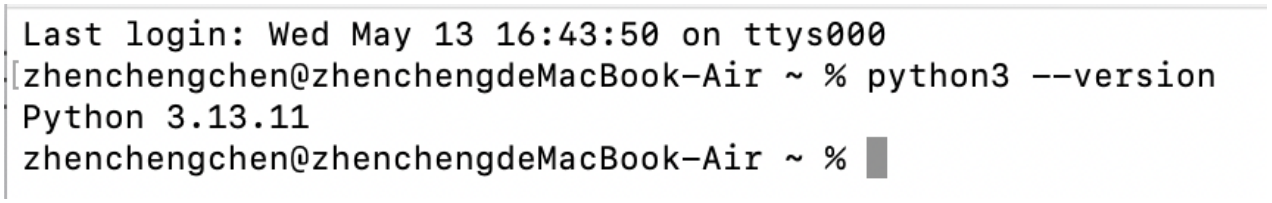
5. 如果看到 `pip` 的版本信息，说明包管理工具也安装成功。
6. 检查 Python 架构：

```
python3 -c "import platform; print(platform.machine())"
```

常见输出：

```
x86_64
arm64
```

Apple Silicon Mac 推荐输出为 `arm64`。

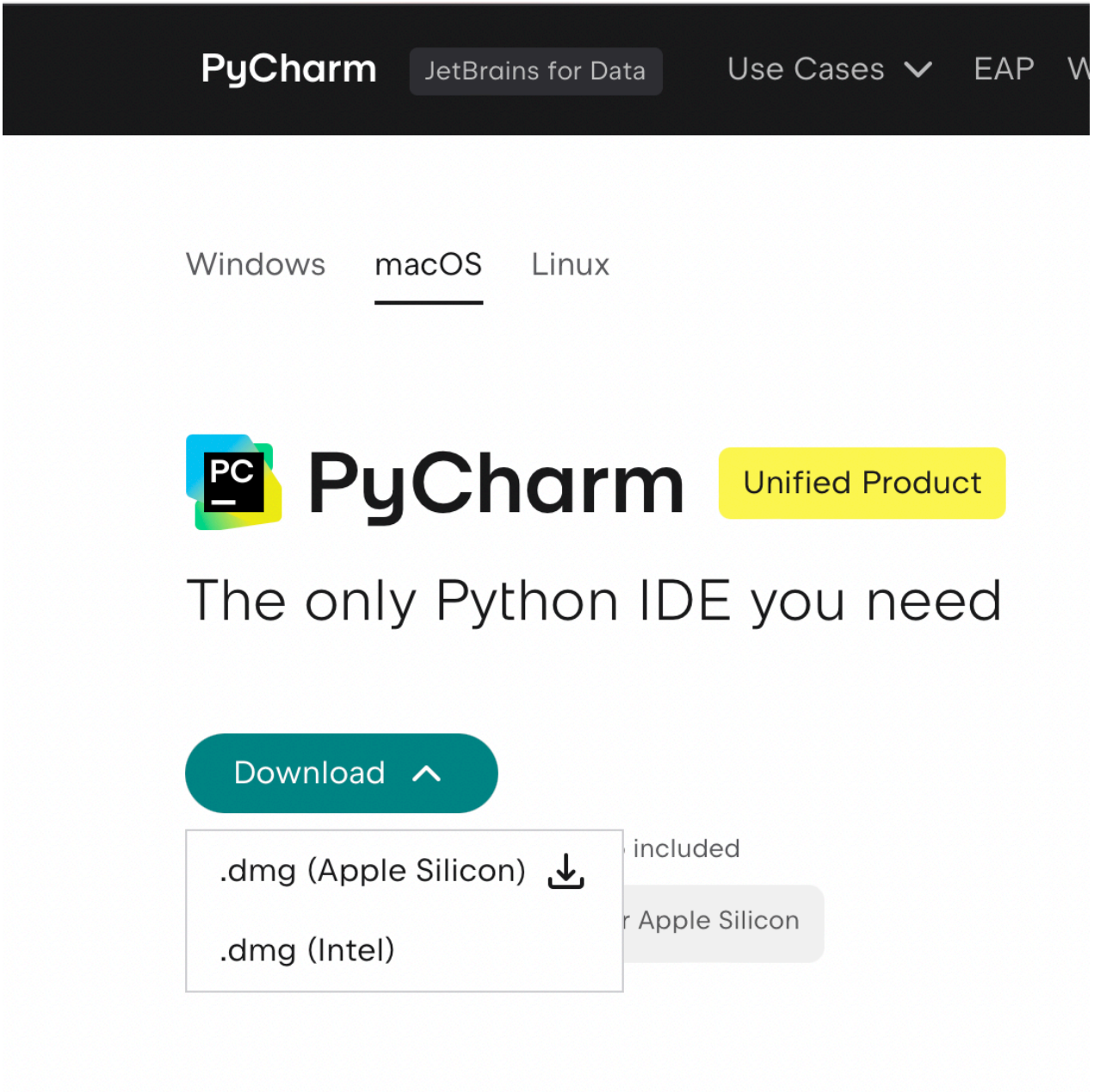


```
Last login: Wed May 13 16:43:50 on ttys000
zhenchengchen@zhenchengdeMacBook-Air ~ % python3 --version
Python 3.13.11
zhenchengchen@zhenchengdeMacBook-Air ~ % python3 -m pip --version
pip 25.3 from /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/
site-packages/pip (python 3.13)
zhenchengchen@zhenchengdeMacBook-Air ~ % python3 -c "import platform; print(plat
form.machine())"
arm64
zhenchengchen@zhenchengdeMacBook-Air ~ %
```

4.5 下载 PyCharm

1. 打开 PyCharm 官网：

   https://www.jetbrains.com/pycharm/download/



2. 根据 Mac 类型选择版本：

Mac 类型	PyCharm 版本
Intel Mac	macOS Intel
Apple Silicon Mac	macOS Apple Silicon

3. 下载 `.dmg` 文件。

4.6 安装 PyCharm

1. 双击下载好的 `.dmg` 文件。
2. 把 PyCharm 拖到：

```
Applications
```

3. 打开“应用程序”。
4. 双击 PyCharm。
5. 第一次打开时，选择默认设置即可。

---

5. 创建测试项目并安装 Python 第三方包

建议为练习项目创建独立的虚拟环境。虚拟环境可以理解为“这个文件夹自己的 Python 环境”，不会影响电脑上的其他 Python 项目。

下面统一在用户主目录下创建测试项目，不需要提前准备任何文件夹。

```
python_test/test_folder
```

5.1 Windows 安装包

1. win+R打开命令提示符cmd。
2. 进入用户主目录：

```
cd /d "%USERPROFILE%"
```

3. 创建 `python_test` 文件夹：

```
mkdir python_test
```

4. 进入 `python_test` 文件夹：

```
cd python_test
```

5. 创建 `test_folder` 文件夹：

```
mkdir test_folder
```

6. 进入 `test_folder` 文件夹：

```
cd test_folder
```

7. 创建虚拟环境：

```
python -m venv .venv
```

8. 启动虚拟环境：

```
.venv\Scripts\activate
```

9. 启动成功后，命令行前面会出现：

```
(.venv)
```

10. 升级 pip：

```
python -m pip install --upgrade pip
```

11. 安装所需包：

```
python -m pip install numpy pandas matplotlib scipy scikit-learn openpyxl
```

```
Microsoft Windows [Version 10.0.26200.8246]
(c) Microsoft Corporation。保留所有权利。

C:\Users\KissAquarius>cd /d "%USERPROFILE%"

C:\Users\KissAquarius>mkdir python_test

C:\Users\KissAquarius>cd python_test

C:\Users\KissAquarius\python_test>mkdir test_folder

C:\Users\KissAquarius\python_test>cd test_folder

C:\Users\KissAquarius\python_test\test_folder>python -m venv .venv

C:\Users\KissAquarius\python_test\test_folder>.venv\Scripts\activate

(.venv) C:\Users\KissAquarius\python_test\test_folder>python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\kissaquarius\python_test\test_folder\.venv\lib\site-packages (25.3)
Collecting pip
  Downloading pip-26.1.1-py3-none-any.whl.metadata (4.6 kB)
Downloading pip-26.1.1-py3-none-any.whl (1.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 55.8 kB/s  0:00:32
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 25.3
    Uninstalling pip-25.3:
      Successfully uninstalled pip-25.3
Successfully installed pip-26.1.1
```

```
(.venv) C:\Users\KissAquarius\python_test\test_folder>python -m pip install numpy pandas matplotlib scipy scikit-learn o
penpyxl
```

```
Successfully installed contourpy-1.3.3 cycler-0.12.1 et-xmlfile-2.0.0 fonttools-4.62.1 joblib-1.5.3 kiwisolver-1.5.0 mat
plotlib-3.10.9 numpy-2.4.4 openpyxl-3.1.5 packaging-26.2 pandas-3.0.3 pillow-12.2.0 pyparsing-3.3.2 python-dateutil-2.9.
0.post0 scikit-learn-1.8.0 scipy-1.17.1 six-1.17.0 threadpoolctl-3.6.0 tzdata-2026.2
```

5.2 macOS 安装包

1. 打开“终端”。
2. 进入用户主目录：

```
cd "$HOME"
```

3. 创建 `python_test` 文件夹：

```
mkdir python_test
```

4. 进入 `python_test` 文件夹：

```
cd python_test
```

5. 创建 `test_folder` 文件夹：

```
mkdir test_folder
```

6. 进入 `test_folder` 文件夹：

```
cd test_folder
```

7. 创建虚拟环境：

```
python3 -m venv .venv
```

8. 启动虚拟环境：

```
source .venv/bin/activate
```

9. 启动成功后，命令行前面会出现：

```
(.venv)
```

10. 升级 pip：

```
python -m pip install --upgrade pip
```

11. 安装所需包：

```
python -m pip install numpy pandas matplotlib scipy scikit-learn openpyxl
```

```
zhenchengchen@zhenchengdeMacBook-Air ~ % cd "$HOME"
zhenchengchen@zhenchengdeMacBook-Air ~ % mkdir python_test
zhenchengchen@zhenchengdeMacBook-Air ~ % cd python_test
zhenchengchen@zhenchengdeMacBook-Air python_test % mkdir test_folder
zhenchengchen@zhenchengdeMacBook-Air python_test % cd test_folder
zhenchengchen@zhenchengdeMacBook-Air test_folder % python3 -m venv .venv
```

```
zhenchengchen@zhenchengdeMacBook-Air test_folder % source .venv/bin/activate
(.venv) zhenchengchen@zhenchengdeMacBook-Air test_folder % python -m pip install
 --upgrade pip
Requirement already satisfied: pip in ./.venv/lib/python3.13/site-packages (25.3
)
Collecting pip
  Downloading pip-26.1.1-py3-none-any.whl.metadata (4.6 kB)
Downloading pip-26.1.1-py3-none-any.whl (1.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 7.8 MB/s  0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 25.3
    Uninstalling pip-25.3:
      Successfully uninstalled pip-25.3
Successfully installed pip-26.1.1
```

```
Installing collected packages: threadpoolctl, six, pyparsing, pillow, packaging,
 numpy, kiwisolver, joblib, fonttools, et-xmlfile, cycler, scipy, python-dateuti
l, openpyxl, contourpy, scikit-learn, pandas, matplotlib
Successfully installed contourpy-1.3.3 cycler-0.12.1 et-xmlfile-2.0.0 fonttools-
4.62.1 joblib-1.5.3 kiwisolver-1.5.0 matplotlib-3.10.9 numpy-2.4.4 openpyxl-3.1.
5 packaging-26.2 pandas-3.0.3 pillow-12.2.0 pyparsing-3.3.2 python-dateutil-2.9.
0.post0 scikit-learn-1.8.0 scipy-1.17.1 six-1.17.0 threadpoolctl-3.6.0
```

6. 检查 Python 包是否安装成功

请先确认命令行前面有：

```
(.venv)
```

然后运行：

```
python -c "import numpy, pandas, matplotlib, scipy, sklearn, openpyxl; print('所有包安装成功')"
```

如果输出：

```
所有包安装成功
```

说明第三方包安装完成。

```
(.venv) C:\Users\KissAquarius\python_test\test_folder>python -c "import numpy, pandas, matplotlib, scipy, sklearn, openp
yxl; print('所有包安装成功')"
所有包安装成功

(.venv) C:\Users\KissAquarius\python_test\test_folder>
```

还可以查看版本：

```
python -c "import numpy, pandas, matplotlib, scipy, sklearn, openpyxl; print('numpy',
numpy.__version__); print('pandas', pandas.__version__); print('matplotlib',
matplotlib.__version__); print('scipy', scipy.__version__); print('sklearn',
sklearn.__version__); print('openpyxl', openpyxl.__version__)"
```

```
(.venv) zhenchengchen@zhenchengdeMacBook-Air test_folder % python -c "import num
py, pandas, matplotlib, scipy, sklearn, openpyxl; print('所有包安装成功')"
所有包安装成功
(.venv) zhenchengchen@zhenchengdeMacBook-Air test_folder %
```

7. 编写 hello_world.py 并运行测试

本节会在 `python_test/test_folder` 中创建一个简单的 Python 文件。先在命令行运行一次，再用 PyCharm 打开同一个文件夹运行一次。

7.1 Windows：命令行创建并运行

确认当前已经在：

```
%USERPROFILE%\python_test\test_folder
```

确认已经启动虚拟环境：

```
(.venv)
```

创建文件：

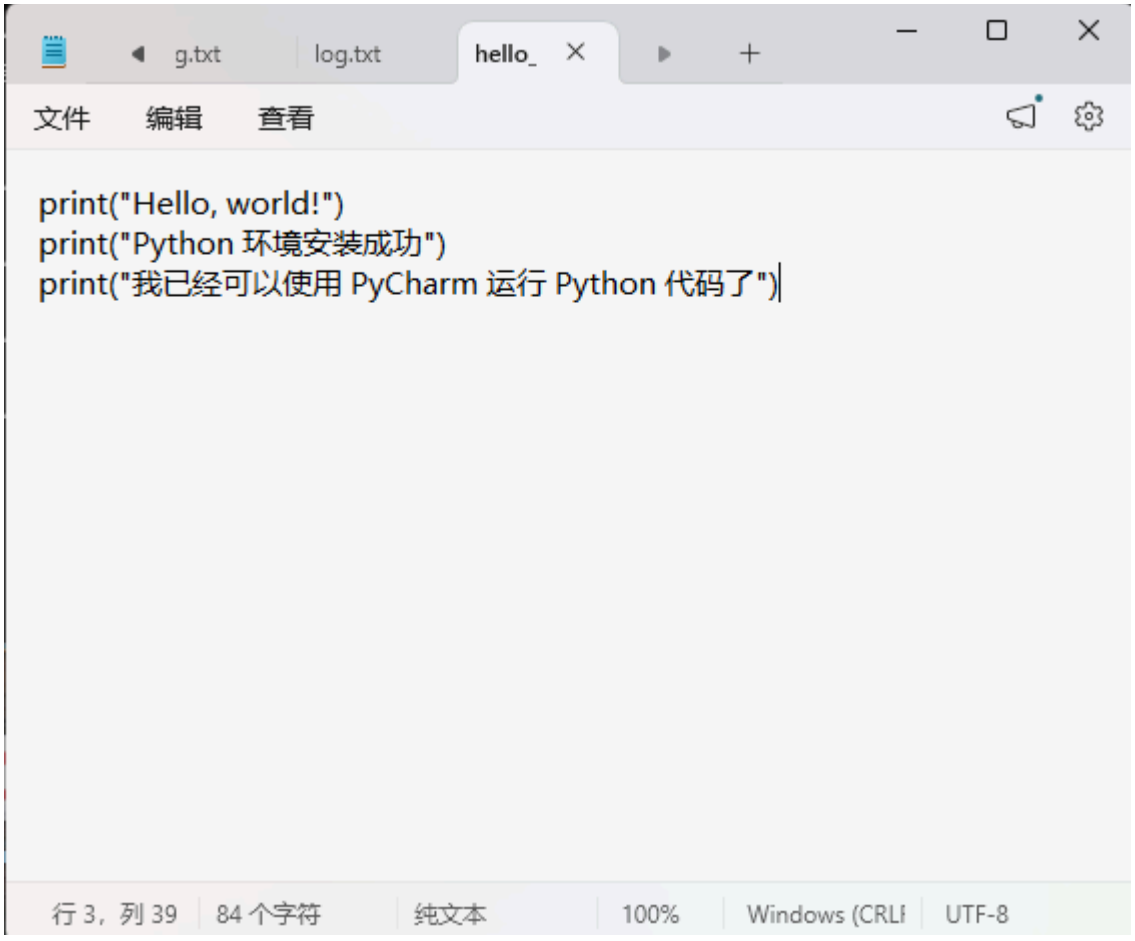
```
notepad hello_world.py
```

如果系统询问是否创建新文件，点击“是”。

在记事本中输入：

```
print("Hello, world!")
print("Python 环境安装成功")
print("我已经可以使用 PyCharm 运行 Python 代码了")
```

保存文件，然后关闭记事本。



在命令行运行：

```
python hello_world.py
```

正常输出：

```
Hello, world!
Python 环境安装成功
我已经可以使用 PyCharm 运行 Python 代码了
```

```
(.venv) C:\Users\KissAquarius\python_test\test_folder>python hello_world.py
Hello, world!
Python 环境安装成功
我已经可以使用 PyCharm 运行 Python 代码了
```

7.2 macOS：命令行创建并运行

确认当前已经在：

```
$HOME/python_test/test_folder
```

确认已经启动虚拟环境：

```
(.venv)
```

创建文件：

```
nano hello_world.py
```

在编辑器中输入：

```
print("Hello, world!")
print("Python 环境安装成功")
print("我已经可以使用 PyCharm 运行 Python 代码了")
```

保存方法：

1. 按 `Control + O`
2. 按回车
3. 按 `Control + X`

运行：

```
python hello_world.py
```

正常输出：

```
Hello, world!
Python 环境安装成功
我已经可以使用 PyCharm 运行 Python 代码了
```

7.3 用 PyCharm 打开同一个测试项目

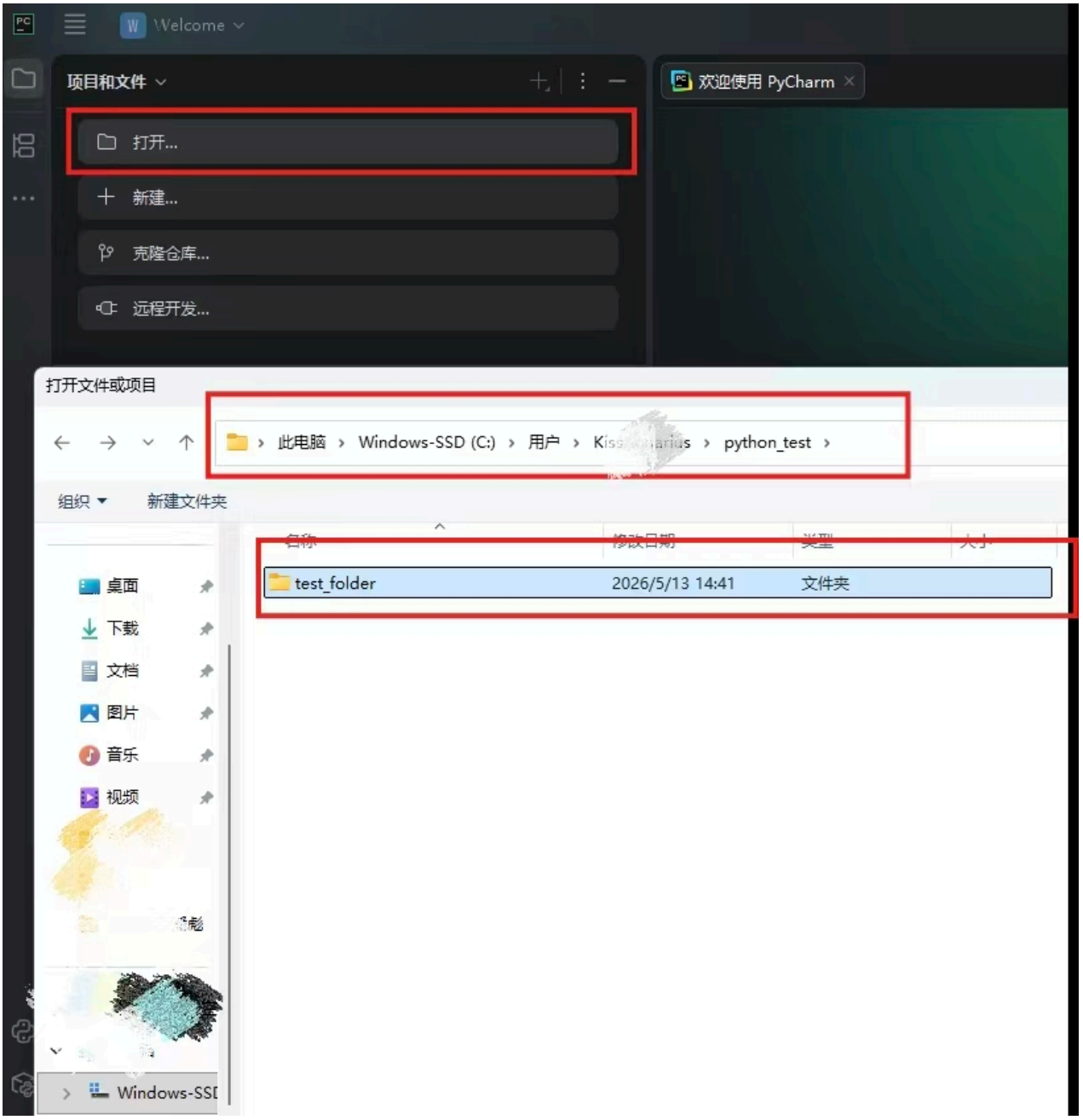
1. 打开 PyCharm。
2. 点击：

   ```
   Open
   ```

3. 选择刚才创建的文件夹：

   ```
   python_test/test_folder
   ```

4. 点击打开。



7.4 设置 Python 解释器

Windows

1. 打开设置：

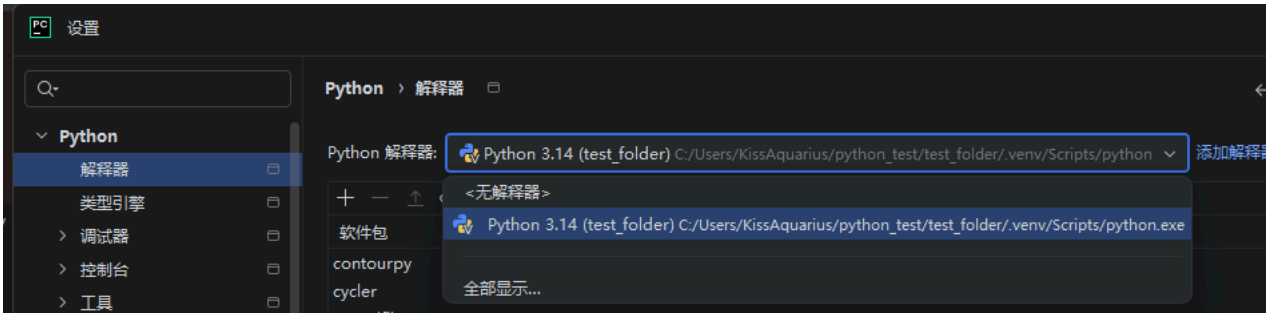
```
File -> Settings 或者文件->设置
```

2. 找到：

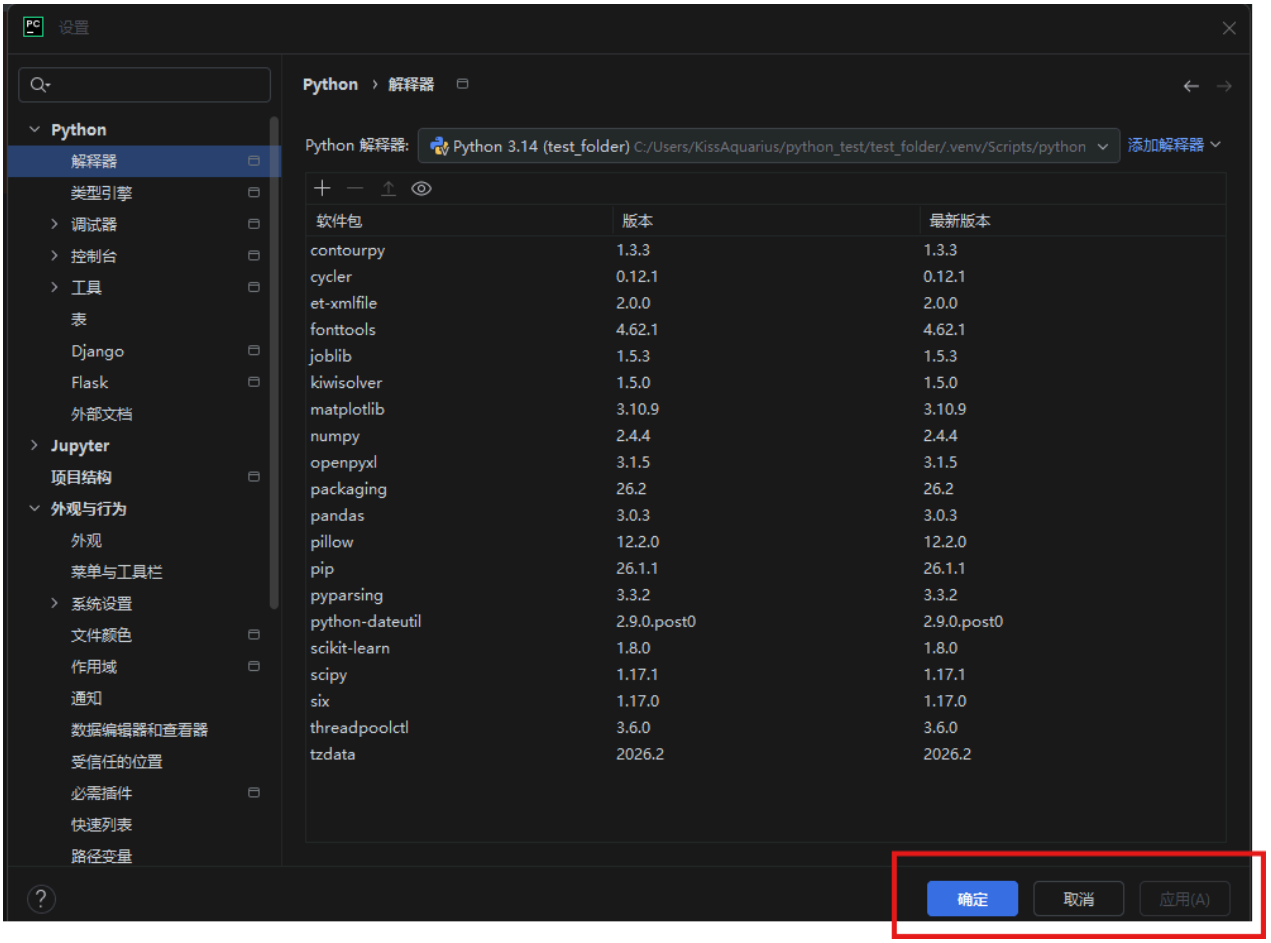
```
Project -> Python Interpreter 或者python->解释器
```

3. 选择 `python_test/test_folder` 里的虚拟环境：

```
.venv\Scripts\python.exe
```



4. 点击 `OK`。



macOS

1. 打开设置：

```
PyCharm -> Settings
```

2. 找到：

```
Project -> Python Interpreter
```

3. 选择 `python_test/test_folder` 里的虚拟环境：

```
.venv/bin/python
```

4. 点击 `OK`。

7.5 在 PyCharm 中查看或粘贴代码

如果左侧已经能看到 `hello_world.py`，直接点击它。

如果没有看到 `hello_world.py`，请按下面步骤创建：

1. 在左侧文件列表中，右键点击 `test_folder`。
2. 点击：

```
New -> Python File
```

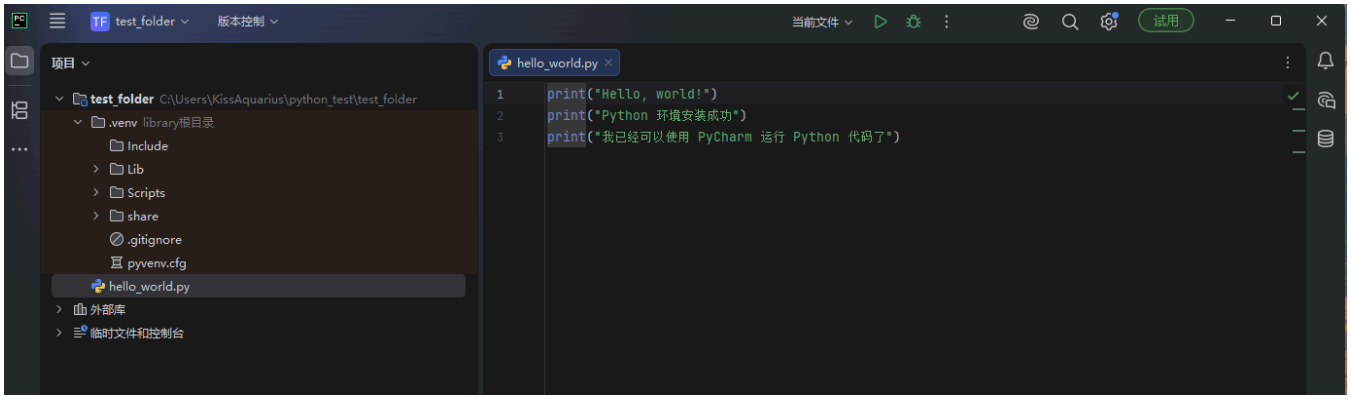
3. 文件名输入：

```
hello_world.py
```

4. 按回车，PyCharm 会创建并打开这个文件。

确认 `hello_world.py` 中有下面的代码。没有的话，请完整复制粘贴进去：

```
print("Hello, world!")
print("Python 环境安装成功")
print("我已经可以使用 PyCharm 运行 Python 代码了")
```



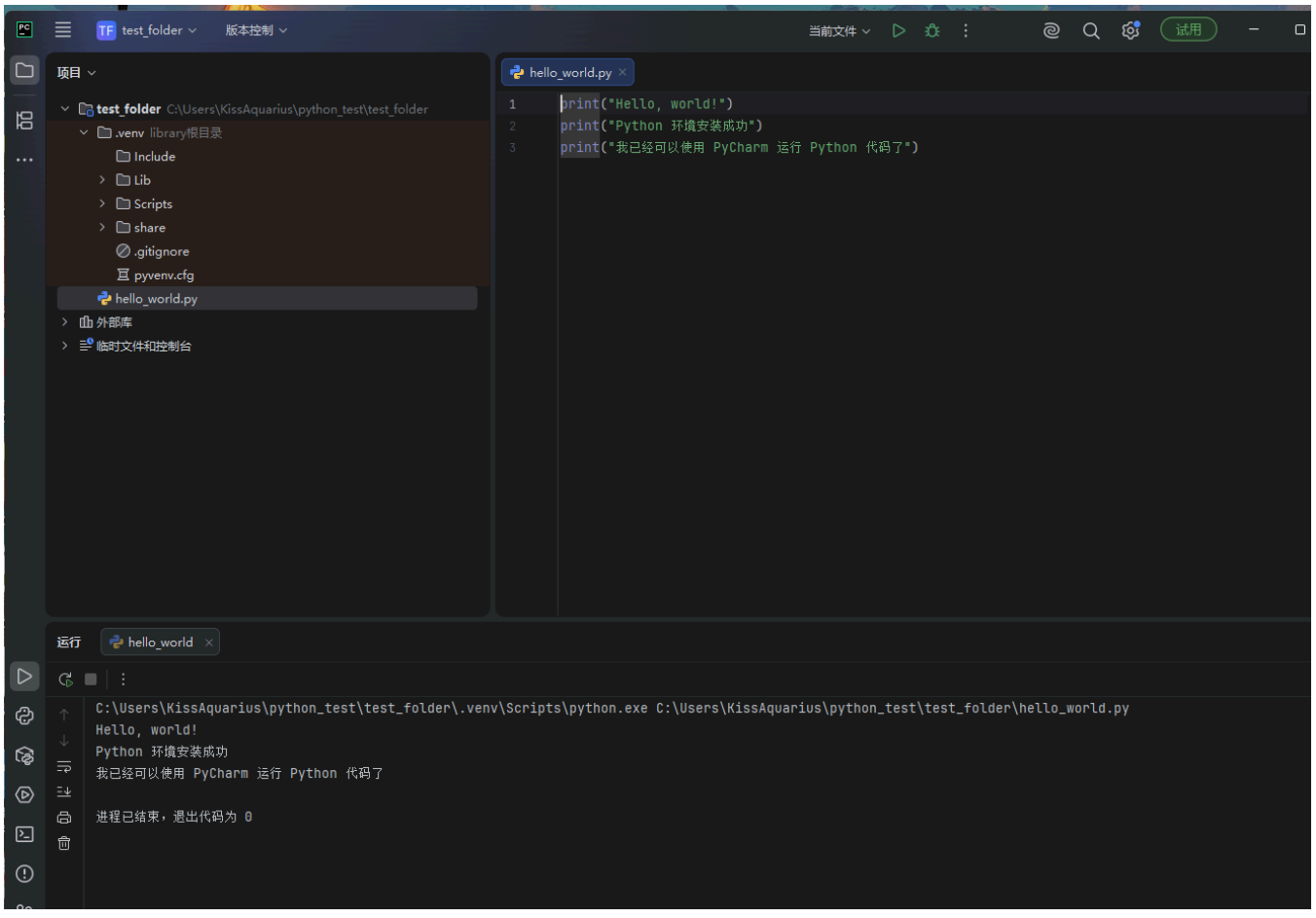
7.6 在 PyCharm 中运行 hello_world.py

1. 右键点击 `hello_world.py` 编辑区。
2. 点击：

```
Run 'hello_world'
```

3. 下方 `Run` 窗口应显示：

```
Hello, world!
Python 环境安装成功
我已经可以使用 PyCharm 运行 Python 代码了
```



看到这几行文字，说明 PyCharm 可以正常运行 Python 文件。

---

8. 完整安装检查清单

安装完成后，可以逐项检查：

检查项	Windows 命令	macOS 命令	成功标志
Python	`python --version`	`python3 --version`	显示 Python 版本
pip	`python -m pip --version`	`python3 -m pip --version`	显示 pip 版本
进入测试文件夹	`cd /d "%USERPROFILE%\python_test\test_folder"`	`cd "$HOME/python_test/test_folder"`	能进入文件夹
虚拟环境	`.venv\Scripts\activate`	`source .venv/bin/activate`	命令行出现 `(.venv)`
第三方包	`python -c "import numpy, pandas, matplotlib, scipy, sklearn, openpyxl; print('OK')"`	同左	输出 `OK`
命令行运行	`python hello_world.py`	`python hello_world.py`	输出 `Hello, world!`
PyCharm 运行	打开并运行 `hello_world.py`	打开并运行 `hello_world.py`	Run 窗口无报错

9. 常见问题

问题 1：Windows 输入 `python` 没反应或打不开

可能原因：安装 Python 时没有勾选 `Add python.exe to PATH`。

解决方法：

1. 重新运行 Python 安装包。
2. 选择 `Modify`。
3. 勾选 PATH 相关选项。
4. 安装完成后重新打开命令提示符。

也可以尝试：

```
py --version
```

如果 `py` 能用，也可以用：

```
py -m pip install numpy pandas matplotlib scipy scikit-learn openpyxl
```

问题 2：Windows PowerShell 无法启动虚拟环境

如果使用 PowerShell，启动命令是：

```
.\.venv\Scripts\Activate.ps1
```

如果提示“不允许运行脚本”，可以先使用“命令提示符 cmd”，命令是：

```
.venv\Scripts\activate
```

初学者建议优先使用 `cmd`。

问题 3：macOS 提示 `python` 找不到

macOS 安装前通常使用：

```
python3
```

创建并启动虚拟环境后，可以使用：

```
python
```

问题 4：macOS Apple Silicon 安装包失败

请检查架构：

```
python3 -c "import platform; print(platform.machine())"
```

Apple Silicon Mac 推荐输出：

```
arm64
```

如果输出是 `x86_64`，可能正在使用 Intel/Rosetta 环境。建议重新安装官网 `universal2` Python，并下载 Apple Silicon 版 PyCharm。

问题 5：下载或安装 Python 包很慢

可以重试：

```
python -m pip install numpy pandas matplotlib scipy scikit-learn openpyxl
```

如果仍然很慢，可以换一个网络后再试。

问题 6：PyCharm 找不到虚拟环境

请先确认 `python_test/test_folder` 中有 `.venv` 文件夹。

Windows 的解释器位置通常是：

```
.venv\Scripts\python.exe
```

macOS 的解释器位置通常是：

```
.venv/bin/python
```

如果没有 `.venv` 文件夹，请回到第 5 节重新创建虚拟环境。

---

10. 官方参考资料

- Python 下载页面：https://www.python.org/downloads/
- Python Windows 使用说明：https://docs.python.org/3/using/windows.html
- Python macOS 使用说明：https://docs.python.org/3/using/mac.html
- PyCharm 安装说明：https://www.jetbrains.com/help/pycharm/installation-guide.html
- scikit-learn 安装说明：https://scikit-learn.org/stable/install.html