CS100-Ray-Tracing 环境配置

CS100-Ray-Tracing **环境配置**
- 分门别类地管理文件
- 项目在 Windows/Linux 中用 VSCode+ GCC 的配置过程
  - 第一步：创建项目文件夹、写入源代码
  - 第二步：准备脚本，生成图片
- 常见问题 FAQ
  - Python 相关问题
  - PowerShell 相关问题
- 附录
  - 源代码

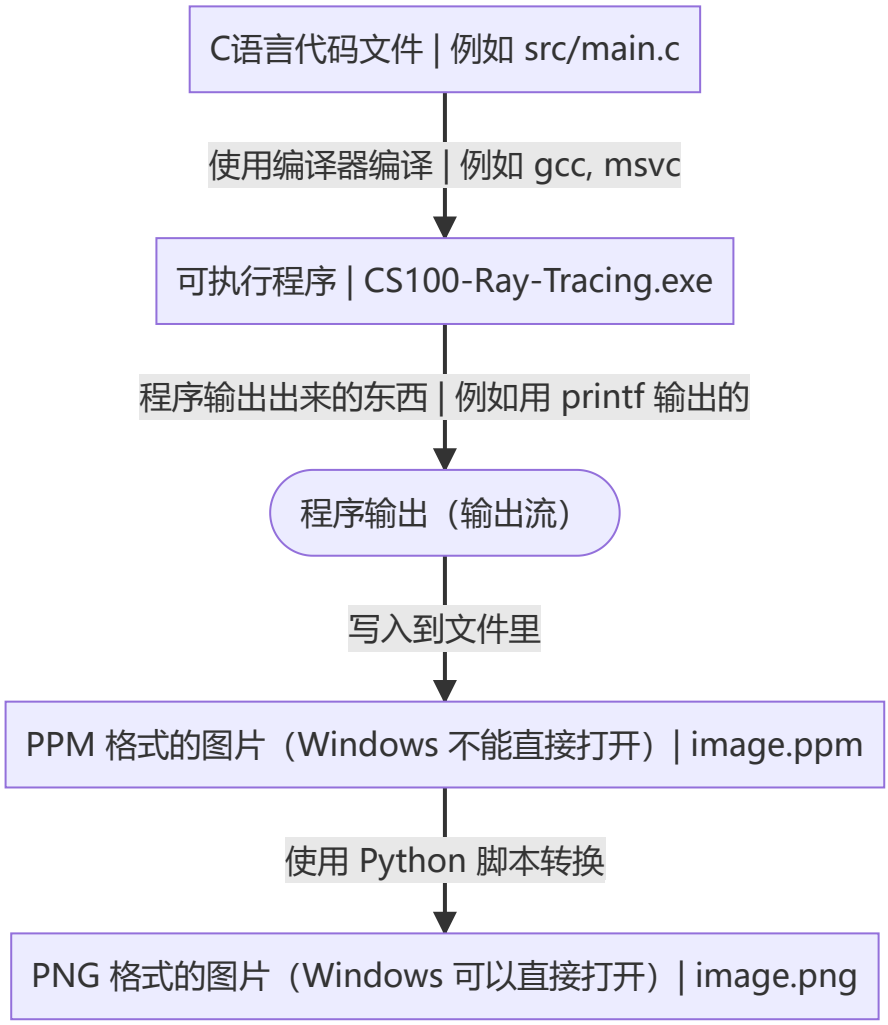
分门别类地管理文件

区别于 CS100 平时的作业都是单文件的开发，我们的项目是在多文件的背景下开发的。项目一开始开发的时候文件数量并不多，不会影响到开发过程。但是随着项目的开发，我们一定会遇到：

1. 代码的结构越来越复杂，源代码文件数量越来越多。源代码文件这部分包括
   1. `.c`，`.cpp`：`C/C++` 源文件
   2. `.h`，`.hpp`：`C/C++` 头文件
2. 与代码运行相关的文件也会越来越多。相关文件这部分包括但不限于
   1. `.exe`，`.o`，`.a`，`.so`，`.lib`：可执行文件和组成可执行文件的相关文件
   2. `.py`，`.sh`，`.ps1`：方便代码运行的脚本文件
   3. `.ppm`、`.png`：图片文件，代码最后的运行结果
   4. `json`：代码配置相关的文件

因此我们需要未雨绸缪地规划好项目文件排布，分门别类的管理文件。针对当前项目，我们采取如下的管理方法：

1. **使用自动化的方式生成运行结果，即使用构建脚本**，替代我们的手动操作。因此我们的脚本流程具体如下：



2. **将不同种类的文件放入对应的文件夹归类整理，即使用文件目录**。具体如下：

   📁 `CS100-Ray-Tracing/`，我们项目的**根目录**。

   - 📁 `src/`，**源代码文件夹**，包括 📄 `main.c`等源文件。
   - 📁 `script/`，**脚本文件夹**。
     - ➡️ `run-debug.ps1`，使用 `Windows` 的命令行工具 `PowerShell` 生成我们的运行结果。
     - ➡️ `run-debug.sh`，使用 `Linux`（~~或者更准确的叫做POSIX环境~~）的命令行工具（例如 `bash`）生成我们的运行结果。
     - 📄 `ppm_to_png.py`，`Python` 脚本，将 `PPM` 文件转化成可以直接打开的 `PNG` 图片文件。
   - 📁 `windows-build/` 或者 `linux-build/`，**项目文件夹**，包含**可执行文件**。
     - `windows-build/`，本意为用 `VS2022` 生成产项目文件夹，现在指用 `.ps1` 文件生成的项目文件夹。
     - `linux-build/`，本意为用在 `Linux` 系统中用 `gcc` 等工具链生成的项目文件夹，现在指用 `.sh` 文件生成的项目文件夹。
     - 📁 `Debug/`，存储有在 `Debug` 模式下产生的可执行文件。

        - 🏃 `CS100-Ray-Tracing.exe`，编译产生的可执行文件。
    - 📁 `result/`，**运行结果**
        - 🖼️ `image.ppm`，PPM 格式图片，Windows 系统中不能直接打开，Ubuntu 系统中可以直接打开。
        - 🖼️ `image.png`，PNG 格式图片，几乎所有操作系统都可以直接打开。

项目在 Windows/Linux 中用 VSCode+ GCC 的配置过程

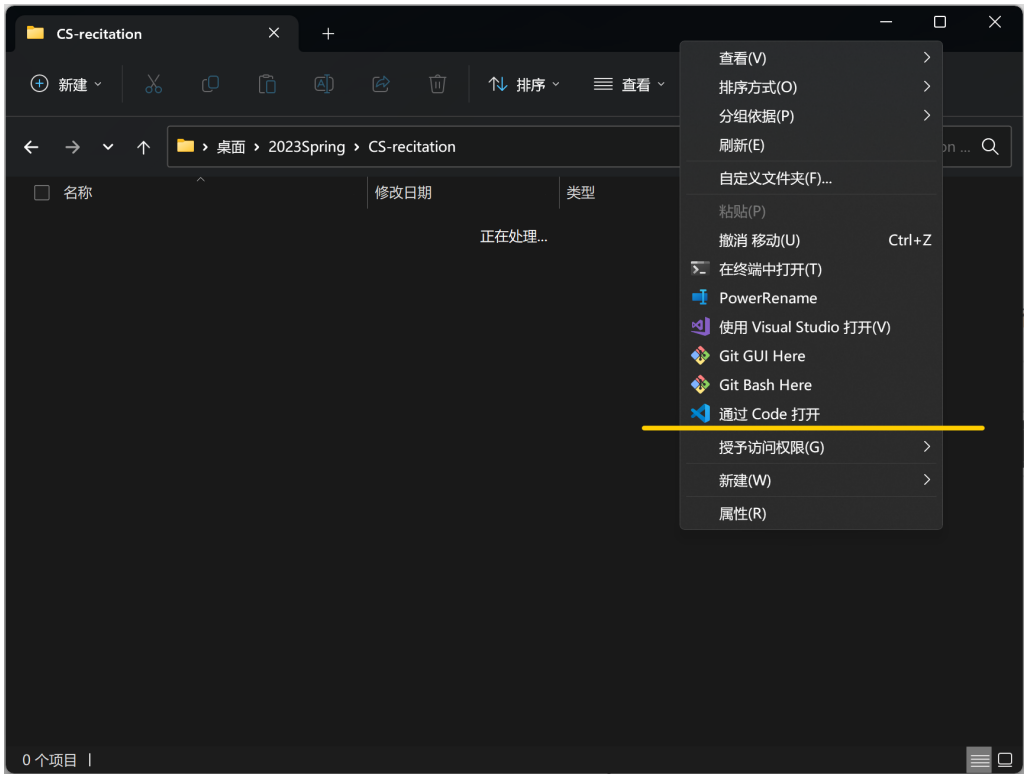
前置条件：`VSCode` 和 `gcc`（或者 `clang`）已经安装完成。

第一步：创建项目文件夹、写入源代码

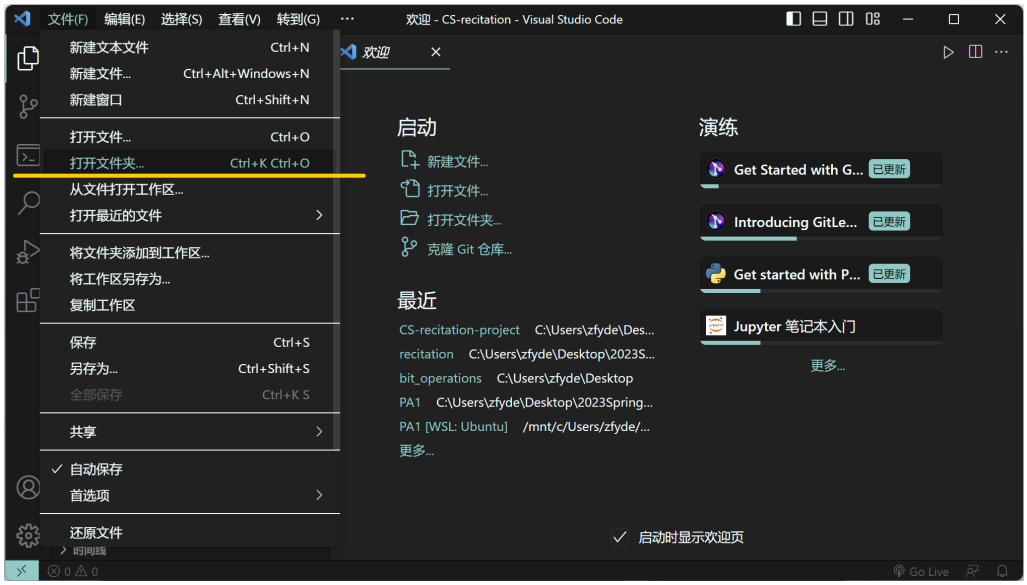
1. 在一个适当的路径下创建项目文件夹。（注意路径名尽量不要含有中文）
    - ⛔不推荐，含有中文

    - ✅推荐

2. 用 `VSCode` 打开这一个文件夹
    - 选项一：文件夹下 `Shift+右键` 打开



- 选项二：通过 VSCode 的菜单打开



3. 点开左侧第一个图标，资源管理器，在其中鼠标右键新建一个 `src` 文件夹，用于存放源代码。在 `src` 文件夹里面新建 `main.c` 文件。

4. 在 `main.c` 里面写入我们的源代码（见#源代码）。

   源代码的意思是——将一个 `ppm` 格式的渐变色图片 `printf` 到我们的命令行输出里。

5. （可选）测试一下

   1. 打开终端

   2. 在终端里输入以下命令，按回车。

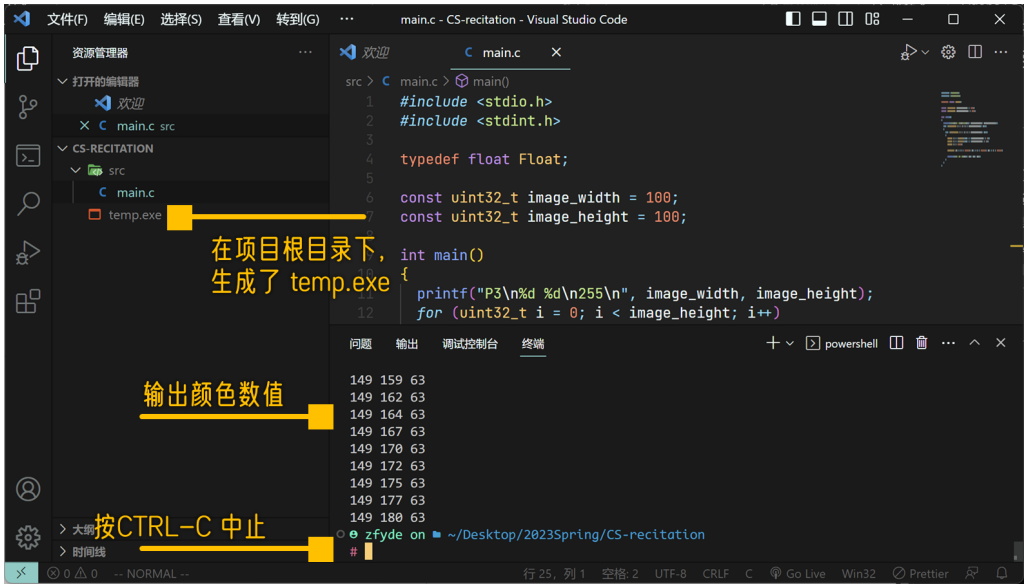
   如果你使用 `Windows`:

```
powershell
PS .../CS100-Ray-Tracing/> gcc .\src\main.c -o .\temp.exe
PS .../CS100-Ray-Tracing/> .\temp.exe
```

   如果你使用 `Linux`:

```bash
.../CS100-Ray-Tracing/ $ gcc ./src/main.c -o temp
.../CS100-Ray-Tracing/ $ ./temp
```

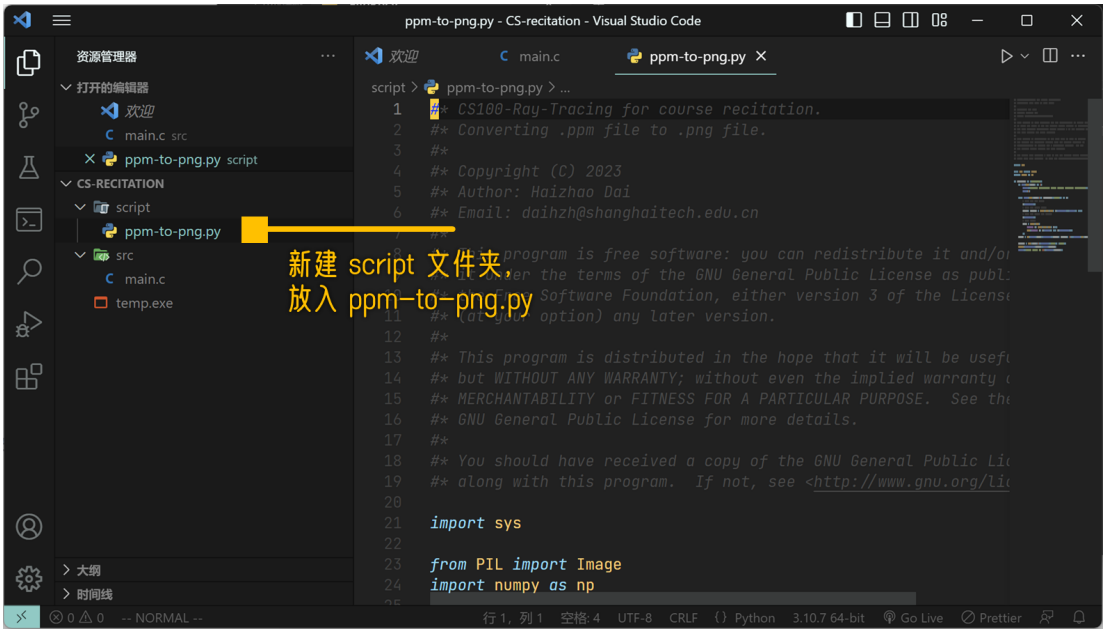
3. 会看见在项目根目录下生成了 `temp.exe`，终端界面不断输出数字（按 `Ctrl-C` 终止程序，~~将刷屏行为绳之以法~~）。



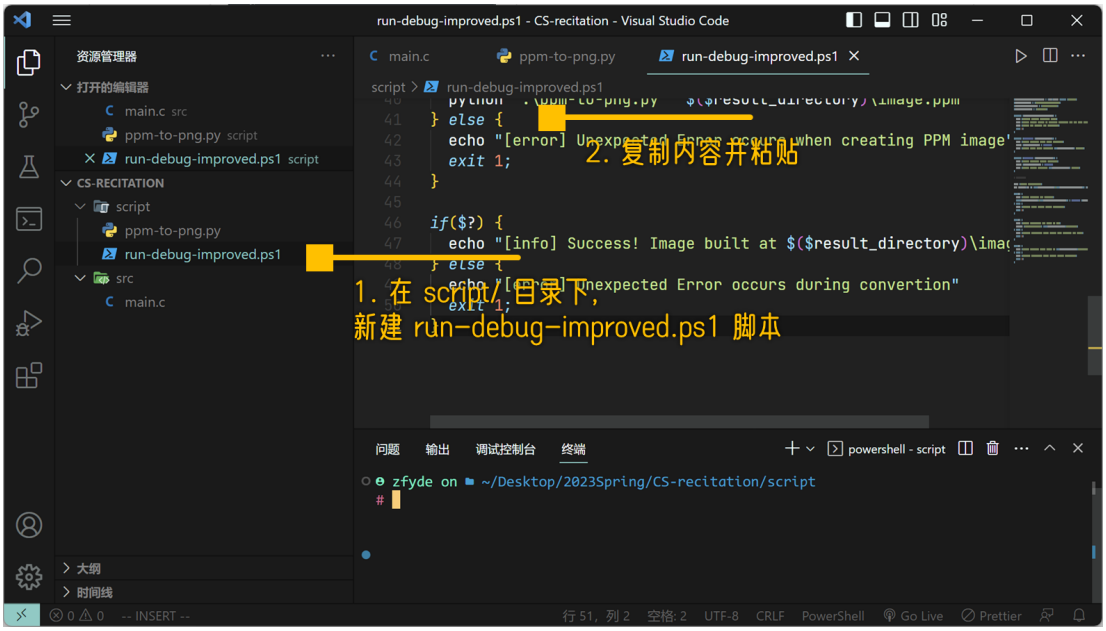
4. 🎉 第一步完成了，记得删除 `temp.exe`。

第二步：准备脚本，生成图片

1. 在项目根目录下，新建 `script` 文件夹。下载一个 [ppm-to-png.py]。



2. 下载一个 [run-debug.ps1]。



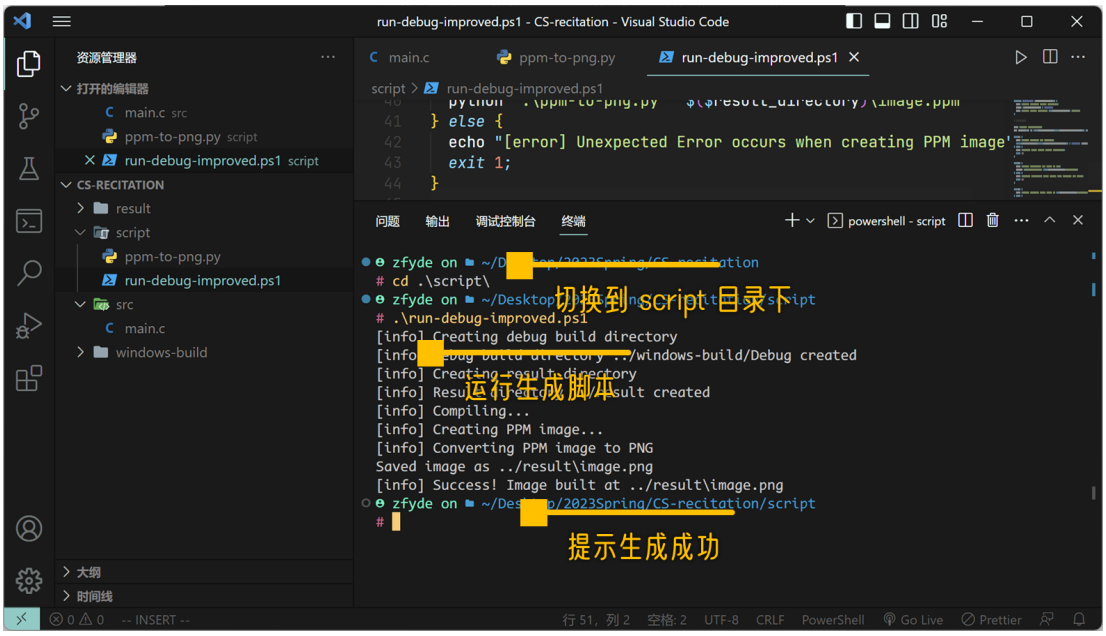
3. 切换终端目录到 `script/` 文件夹下，运行脚本，进行生成。

   如果你使用 Windows：

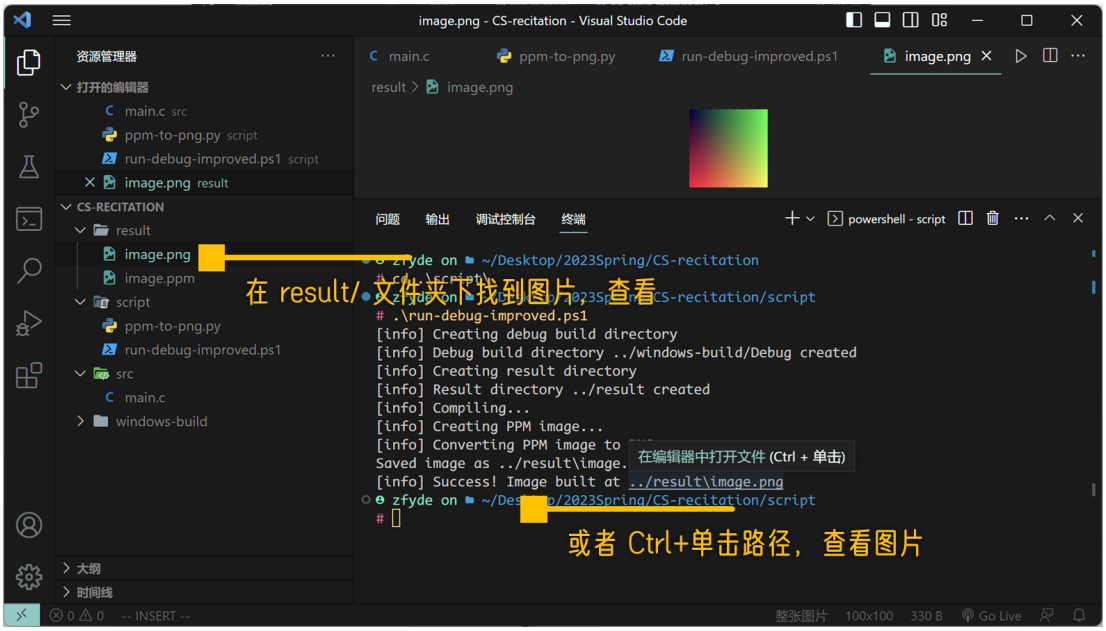
```powershell
PS .../CS100-Ray-Tracing>cd script
PS .../CS100-Ray-Tracing/script> ./run-debug.ps1
```

   如果你使用 Linux：

```bash
.../CS100-Ray-Tracing/ $ cd script
.../CS100-Ray-Tracing/ $ ./run-debug.sh
```



4. 🎉 看到渐变图片，说明运行成功。



常见问题 FAQ

Python 相关问题

检查 python 和 `numpy`，`PIL` 是否已安装。

- 在终端里输入 `python`，如果弹出微软商店就是没有装好，需要进入微软商店或者下载 安装。
- 在终端里继续尝试 `import numpy` 和 `import PIL`，出现报错则需要使用 `pip` 安装两个包。

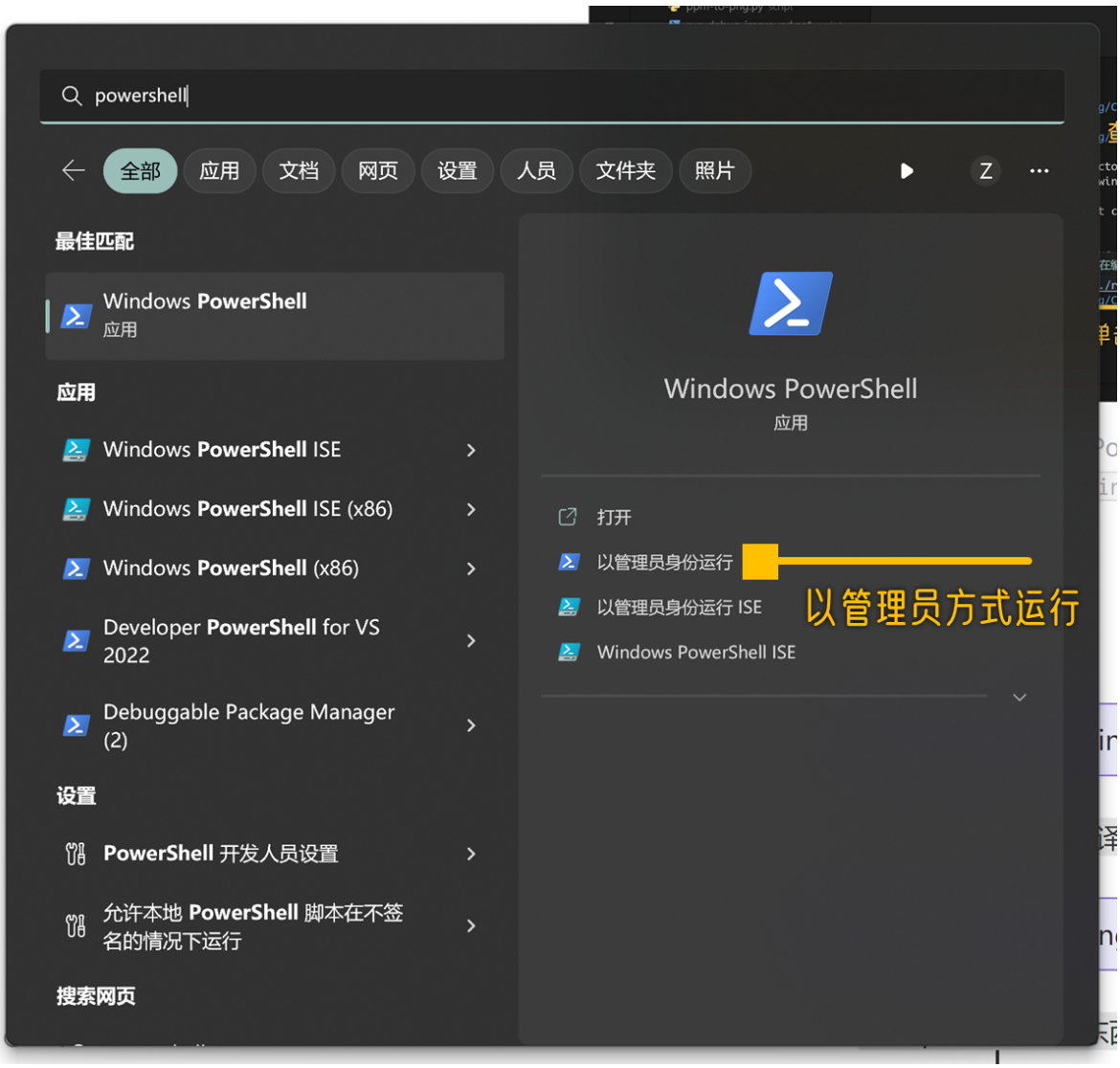
具体如下图所示：

状态	图例
✅ 配置好 Python 了	

状态	图例
⛔没有安装 Python	
✐解决方案	去微软商店下载，或者点此跳转官网安装，或者下载 Anaconda。
⛔ 没有安装好 `numpy` 和 `PIL`	
✐解决方案	输入 `exit` 退出 Python 后，在命令行里输入：`pip install numpy pillow -i https://pypi.tuna.tsinghua.edu.cn/simple`

PowerShell 相关问题

如果运行时提示 "无法在本地运行脚本" "Execution Policy" 字样，请用管理员身份打开 Power shell，输入 `Set-ExecutionPolicy -Scope LocalMachine -ExecutionPolicy RemoteSigned` 后回车。更改 PowerShell 执行网络下载脚本的安全策略。


powershell
全部
应用
文档
网页
设置
人员
文件夹
照片
最佳匹配
Windows PowerShell
应用
应用
Windows PowerShell ISE
Windows PowerShell ISE (x86)
Windows PowerShell (x86)
Developer PowerShell for VS 2022
Debuggable Package Manager (2)
设置
PowerShell 开发人员设置
允许本地 PowerShell 脚本在不签名的情况下运行
搜索网页
Windows PowerShell
应用
打开
以管理员身份运行
以管理员身份运行 ISE
Windows PowerShell ISE
以管理员方式运行

管理员: Windows PowerShell
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。
安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows
zfyde on C:/WINDOWS/system32
Set-ExecutionPolicy -Scope LocalMachine -ExecutionPolicy RemoteSigned
输入命令后回车
执行策略更改
执行策略可帮助你防止执行不信任的脚本。更改执行策略可能会产生安全风险，如 https://go.microsoft.com/fwlink/?LinkID=135170 中的 about_Execution_Policies 帮助主题所述。是否要更改执行策略?
[Y] 是(Y) [A] 全是(A) [N] 否(N) [L] 全否(L) [S] 暂停(S) [?] 帮助 (默认值为“N”): Y
zfyde on C:/WINDOWS/system32
^S
输入 Y 回车确认

附录

源代码

```c
#include <stdio.h>
#include <stdint.h>

typedef float Float;

int main() {
    uint32_t image_width = 256, image_height = 256;

    printf("P3\n%u %u\n255\n", image_width, image_height);
    for (uint32_t j = 0; j < image_height; ++j) {
        for (uint32_t i = 0; i < image_width; ++i) {
            Float r = (Float)i / (Float)(image_width - 1);
            Float g = (Float)j / (Float)(image_height - 1);
            Float b = (Float)0.25;

            uint8_t ir = (uint8_t)(255.0 * r);
            uint8_t ig = (uint8_t)(255.0 * g);
            uint8_t ib = (uint8_t)(255.0 * b);

            printf("%u %u %u\n", ir, ig, ib);
        }
    }
    return 0;
}
```