

编写你的第一个 C++ 程序——在 Windows 下的 VSCode 编辑器 + clang 编译器安装教程

若能有一定 C++ 语法 / bash 命令行的理论基础，阅读会轻松许多；没有也没关系，按步骤配置即可实现你的第一个 C++ 程序 =w=

你应当如何阅读本文

本文在行文过程中，会有一些提示帮助你以正确的方式阅读。因此在此先介绍一下一些标记。

1. 强调：使用 **粗体** 与「」意味如果忽略掉这些文字，你可能在 **编辑器&编译器的概念** 上很难理解后面某处的知识；
2. 引用：使用引用，

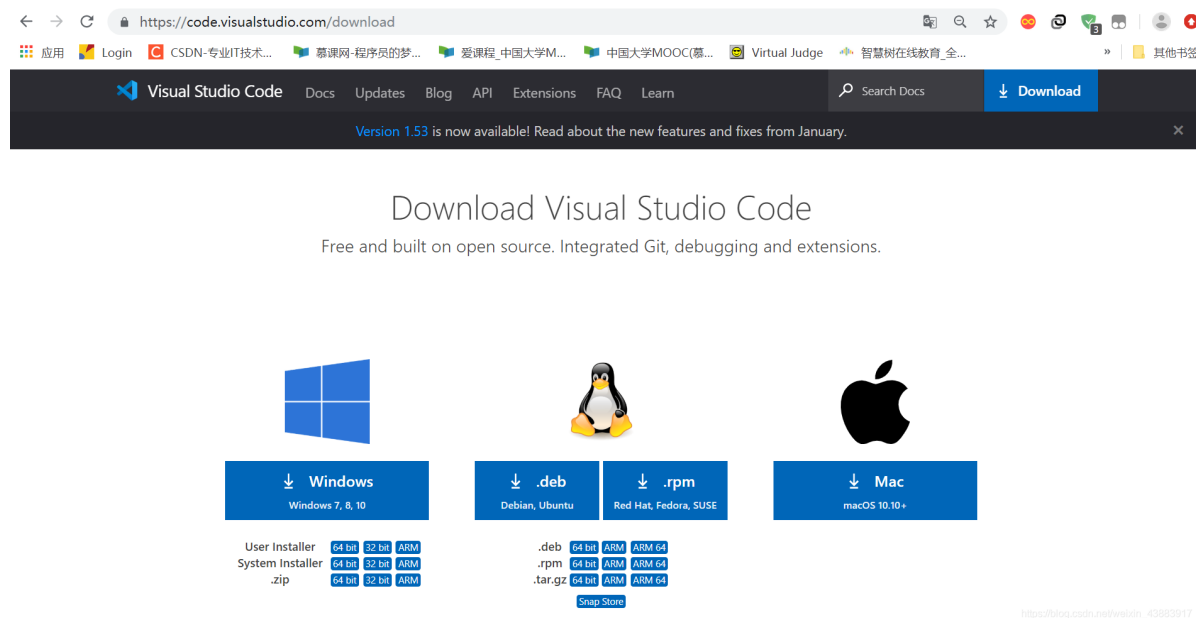
表明这些文字在你第一次阅读本文的时候 不需要 深入了解其中的内容，仅作参考即可，在后续编辑器与编译器使用多了后再深入去探索。

VSCode 介绍与安装

VSCode 全称 Visual Studio Code，是微软出的一款轻量级代码编辑器，免费、开源而且功能强大。它支持几乎所有主流的程序语言的语法高亮、智能代码补全、自定义热键、括号匹配、代码片段、代码对比 Diff、GIT 等特性，支持插件扩展，并针对网页开发和云端应用开发做了优化。

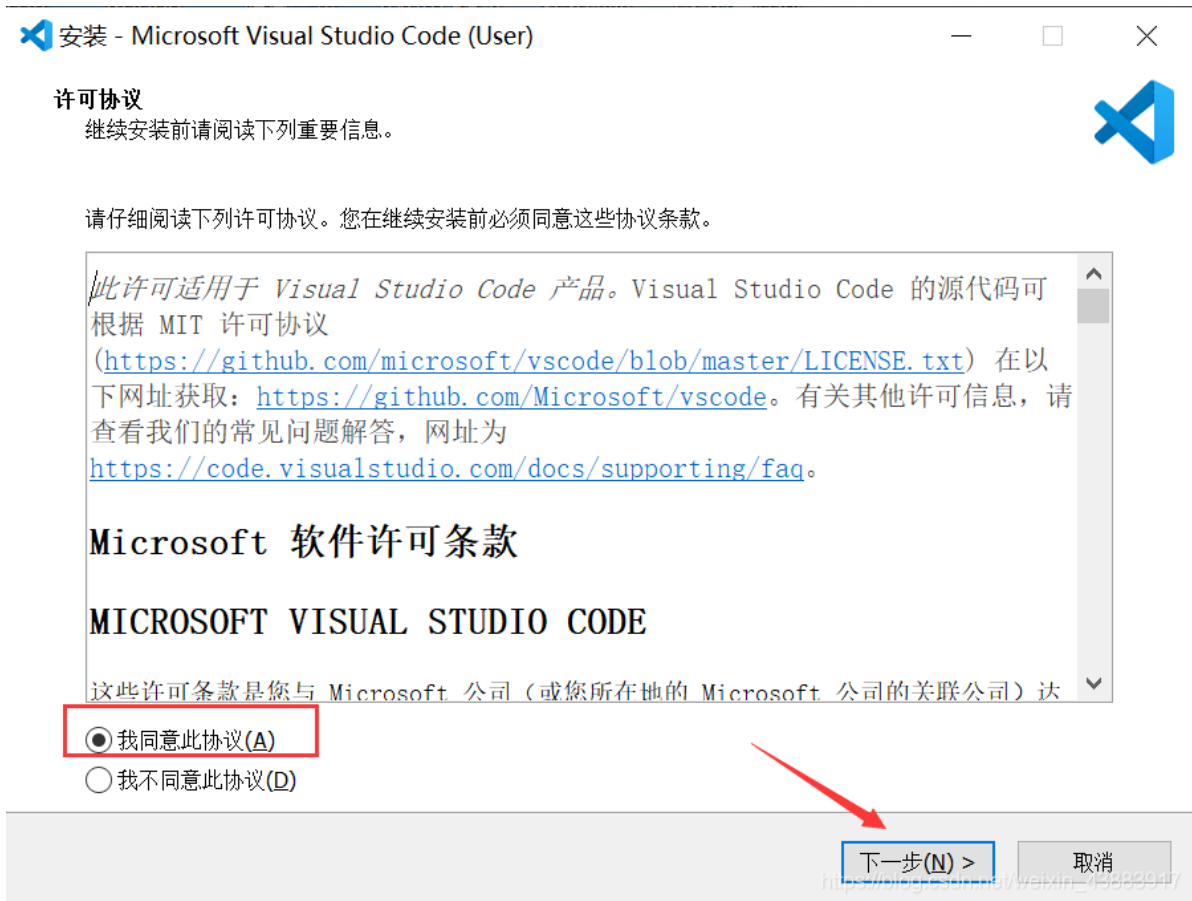
换言之，一切轻量级的代码脚本，只要安装了恰当的「插件」，即可在 VSCode 上编辑编译并运行。

让我们开始下载&安装吧。VSCode 下载地址：[Download Visual Studio Code - Mac, Linux, Windows](https://code.visualstudio.com/download)，根据自己的操作系统选择相应的版本下载即可（以下以 Windows 为例）。

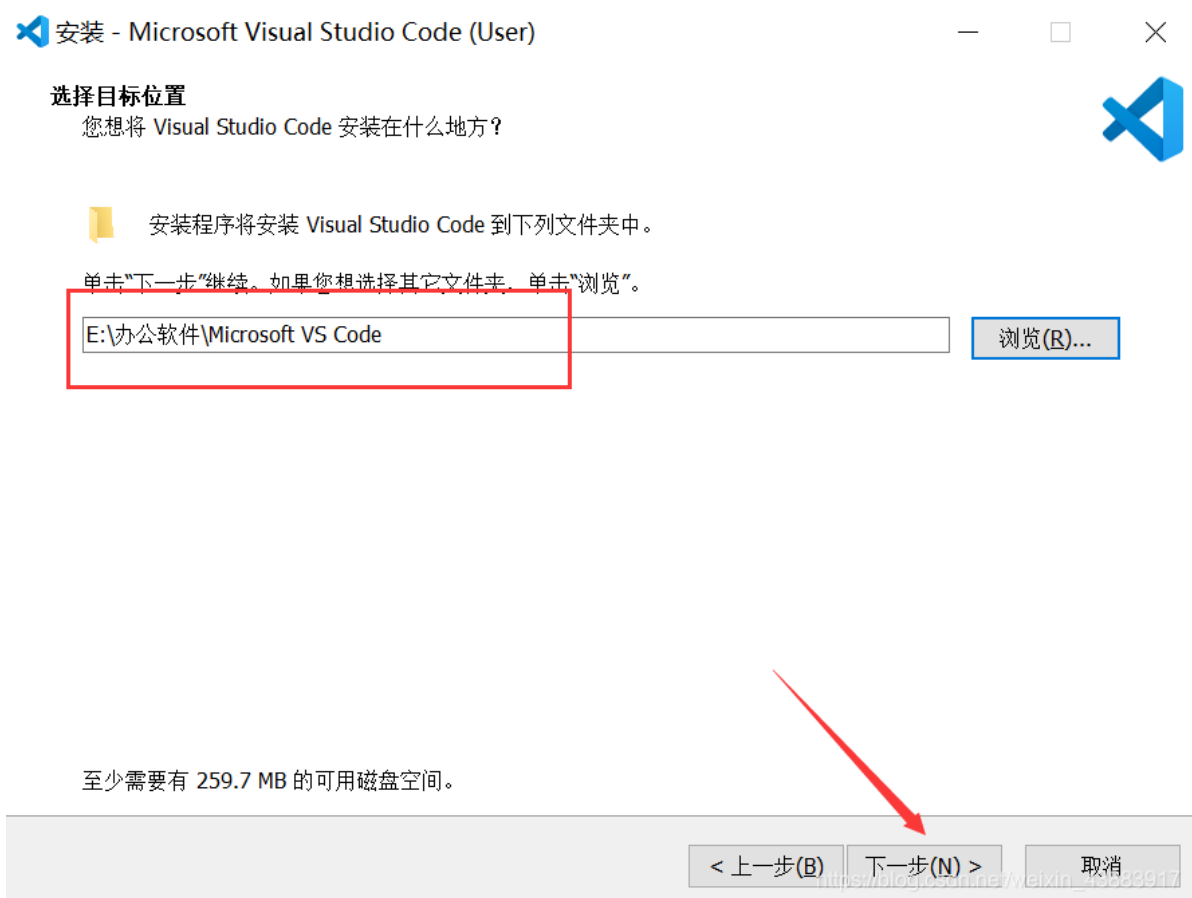


安装

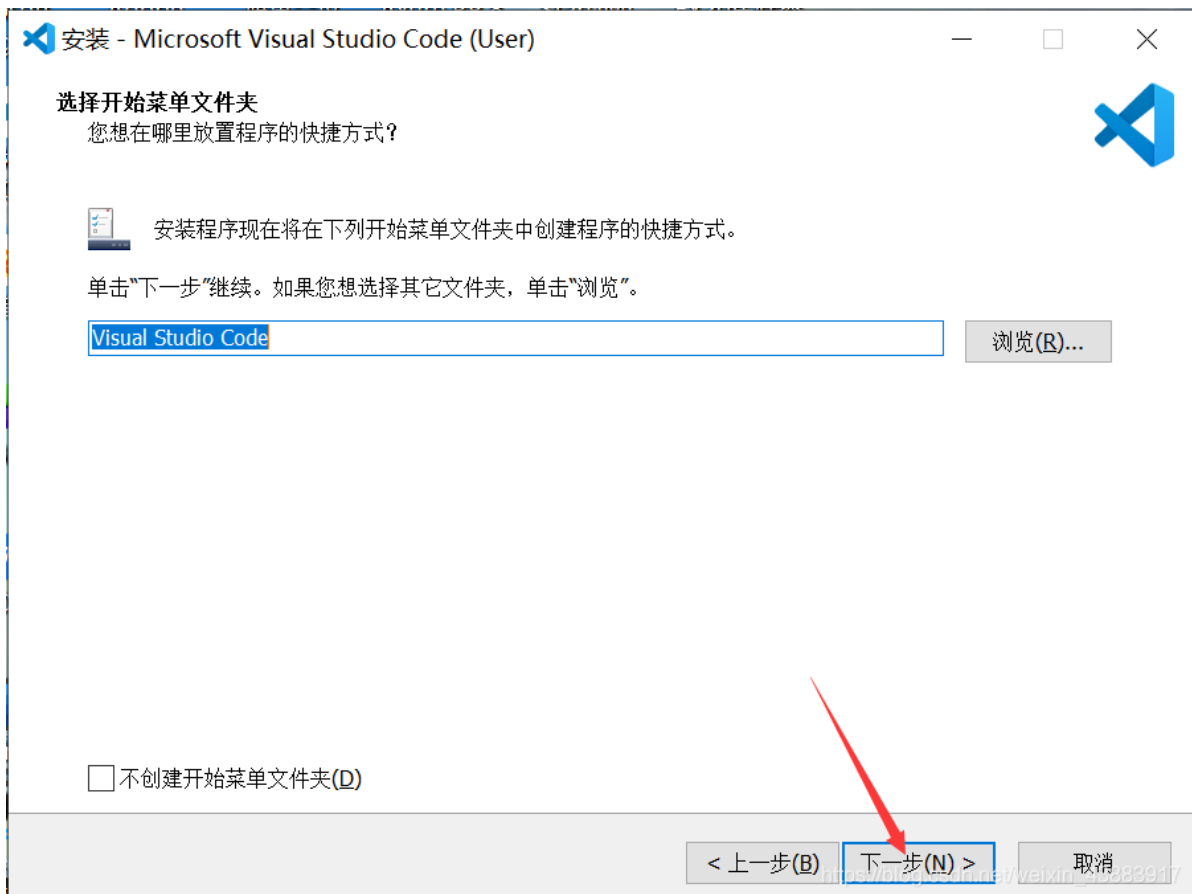
1. 同意协议：



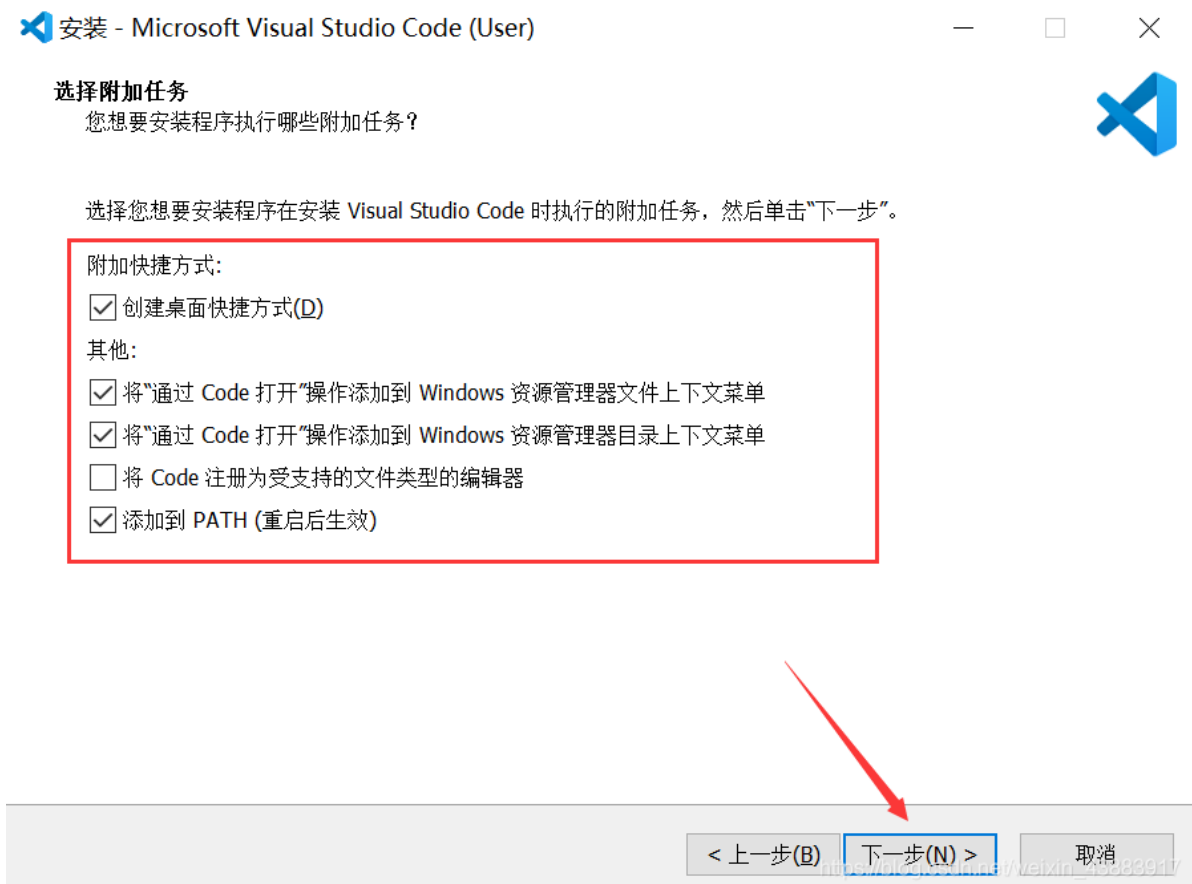
2. 选择安装的位置。默认系统盘，不过放其他位置不影响使用：



3. 快捷图标名称设置：

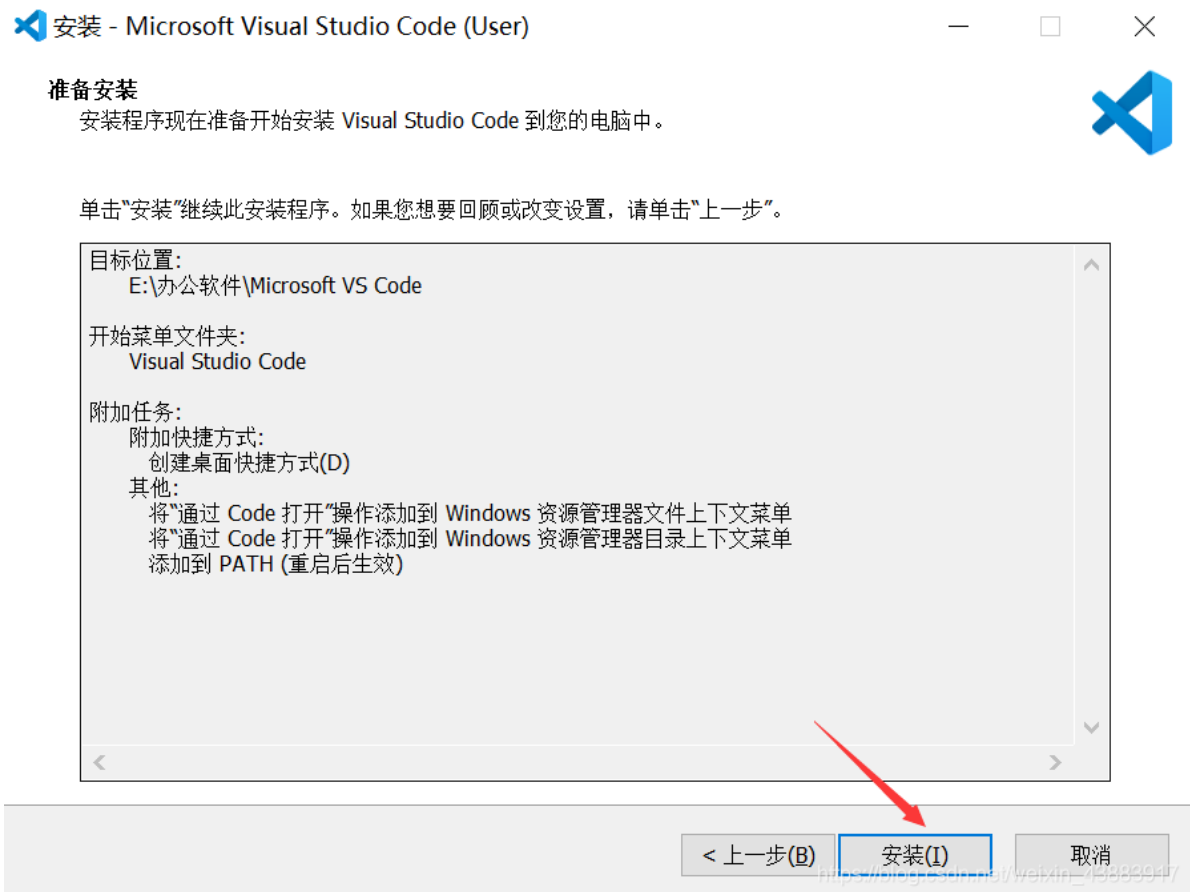


4. 这里注意下，进行相关的选择：

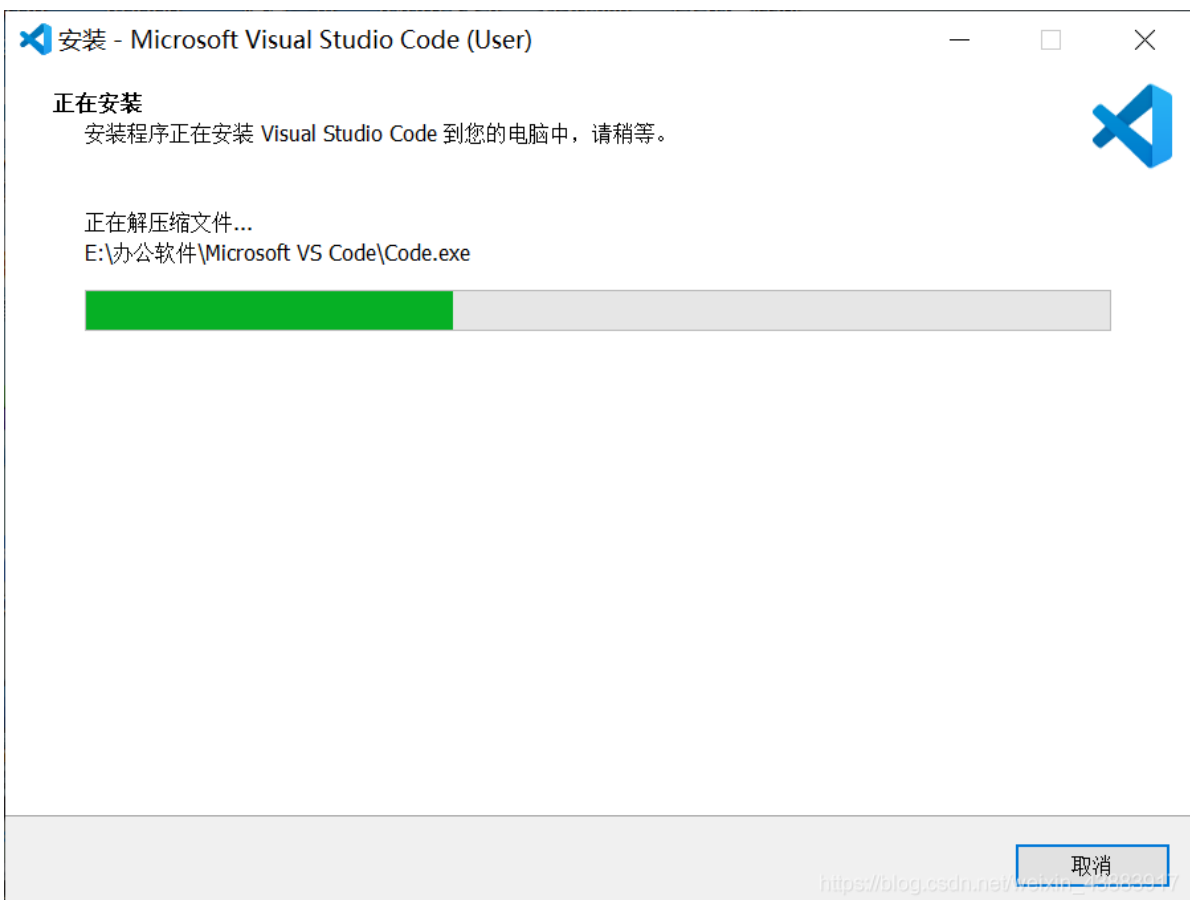


- “通过 Code 打开”操作添加到 windows 资源管理器文件上下文菜单：把这个两个勾选上，**可以对文件使用鼠标右键，选择 VSCode 打开。**
- 将 Code 注册为受支持的文件类型的编辑器：勾选的话，这样会默认使用 VSCode 打开支持的相关文件，**文件图标也会改变。**（见仁见智，可勾可不勾）
- 添加到 PATH（重启后生效）：建议勾选，这样可以使用控制台打开 VSCode 了。

5. 点击安装：



6. 等待安装完成：



编辑器的插件配置

1. 安装好后启动程序，配置中文界面：

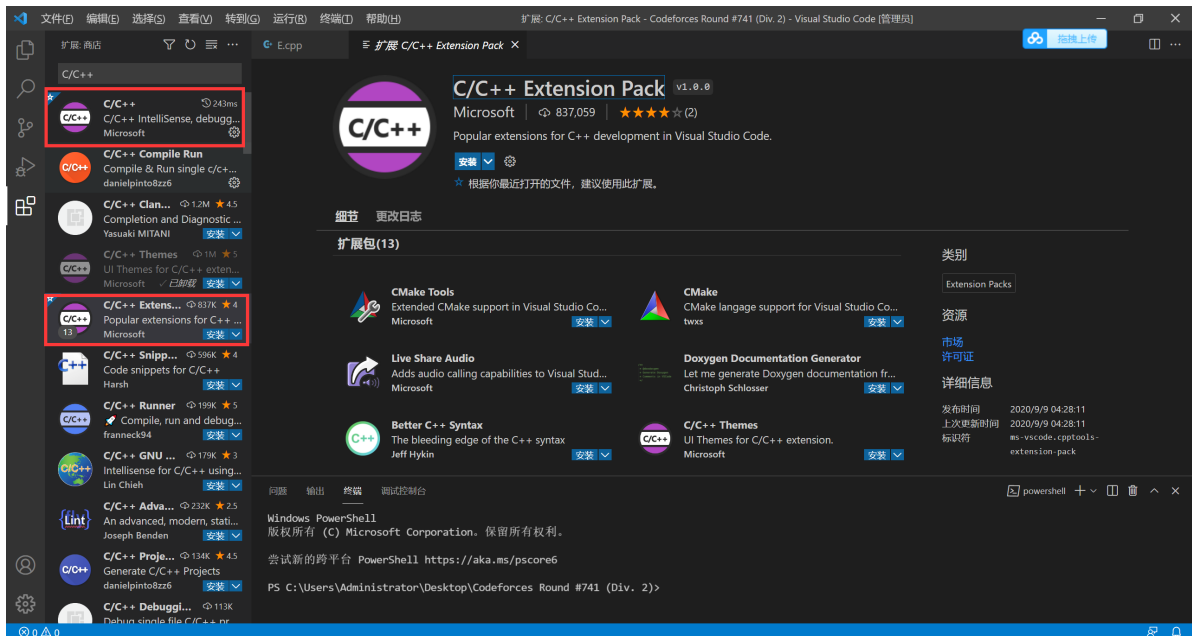


上面安装完成后会出现下面的界面，我们搜索 Chinese，点击 install。安装好后 Restart，重启后就是中文界面了。



上面的Chinese (Simplified) Language 就是 VSCode 众多插件的一个。插件可以更改主题，改变编辑器的外观；可以支持一种新的编程语言，以及配置相应的调试器；还可以管理代码格式，让代码更规范与可观看。为了不跑偏主题，仅介绍与语言直接相关的几个插件，之后对于插件的添加与移除，全交给你自己去探索。

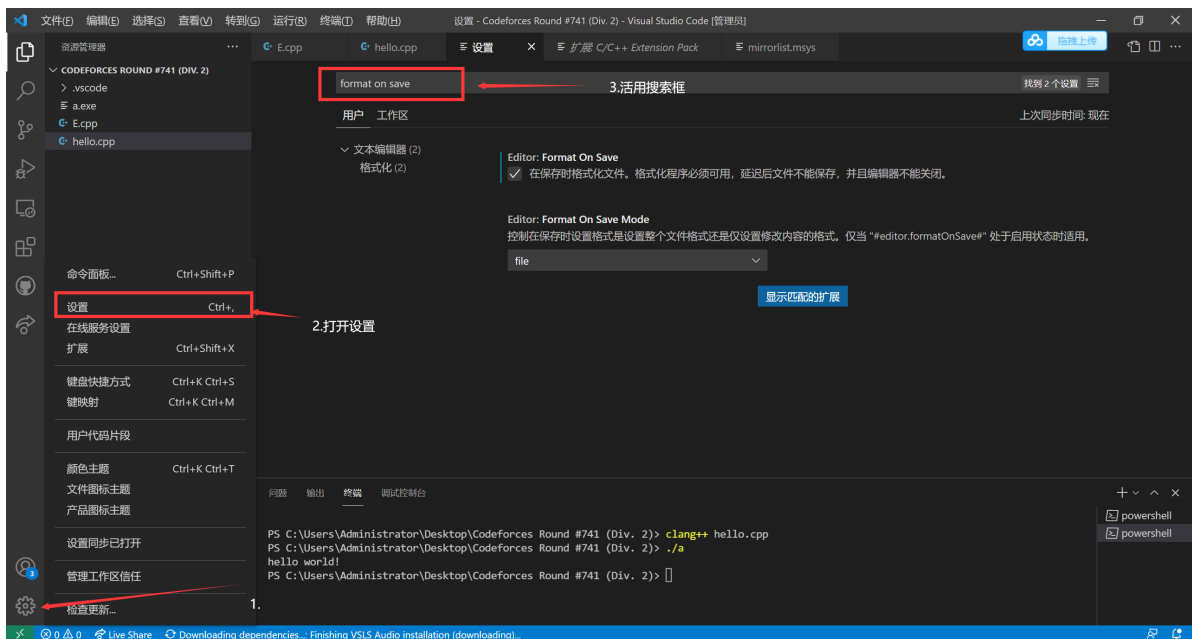
安装 C/C++ 基础&扩展插件：



作为 VSCode 官方发布的 C/C++ 插件，C/C++ Extension 提供了丰富的代码格式化设置与其他功能，在下面的格式化配置需要用到。鉴于笔者能力有限，尚且对该插件的其他功能了解不深，就不多做介绍了。

代码格式化配置

打开 VSCode，进行代码格式化配置：



推荐配置：

保存时格式化：Format On Save 勾上

打完分号后格式化：Format On Type 勾上

滚轮缩放：Mouse Wheel Zoom 勾上

自动保存：Auto Save: onFocusChange

飘红可疑编译错误：Error Squiggles = Enabled

以下是个人爱好：

大括号不换行：将 "{ BasedOnStyle: Chromium, IndentWidth: 4}"（没有""） 键入 clang_format_style 中

编译器 Clang 的安装

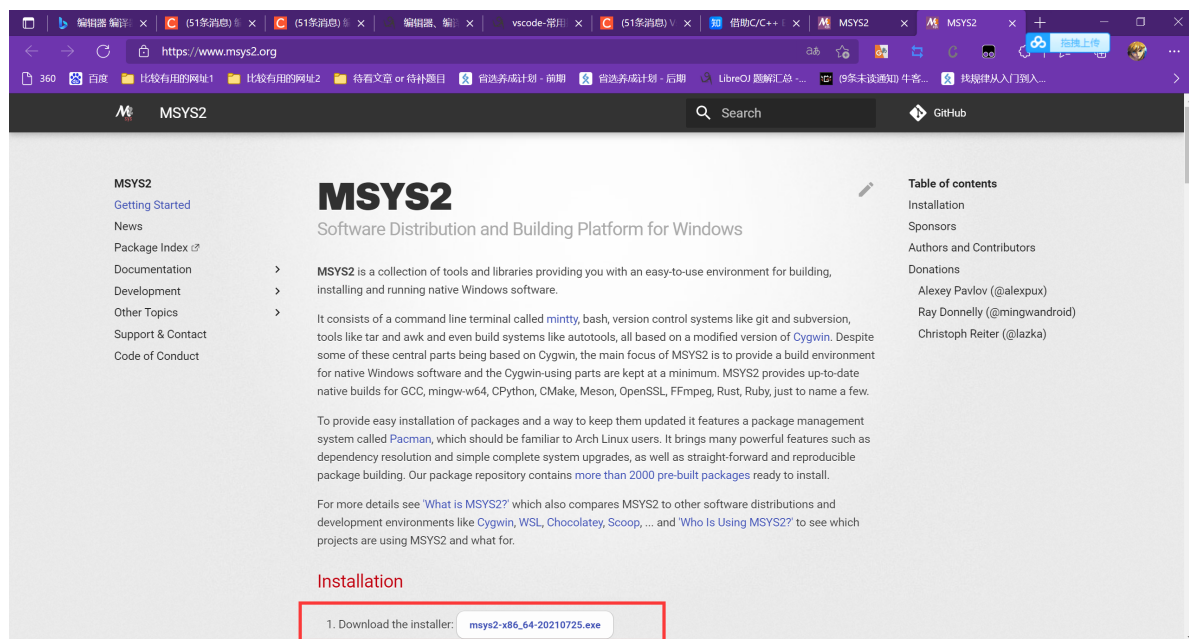
至此，你已经有了一个相对完整的编辑器配置了，不过尚且不能“运行”任何一个代码脚本，原因在于还没有编译器。关于编辑器和编译器的区别如下：

编辑器，就是 **编辑所用的软件**，比如windows自带的记事本就是一个简单的编辑器软件，你要是喜欢，也可有用记事本来写代码，只不过记事本没有语法高亮，不显示行号等等，所以少有人用记事本去写代码。本文介绍的 VSCode 就是一个编辑器，诸如此类的还有 vim，sublime，notepad++，emacs，atom 等。

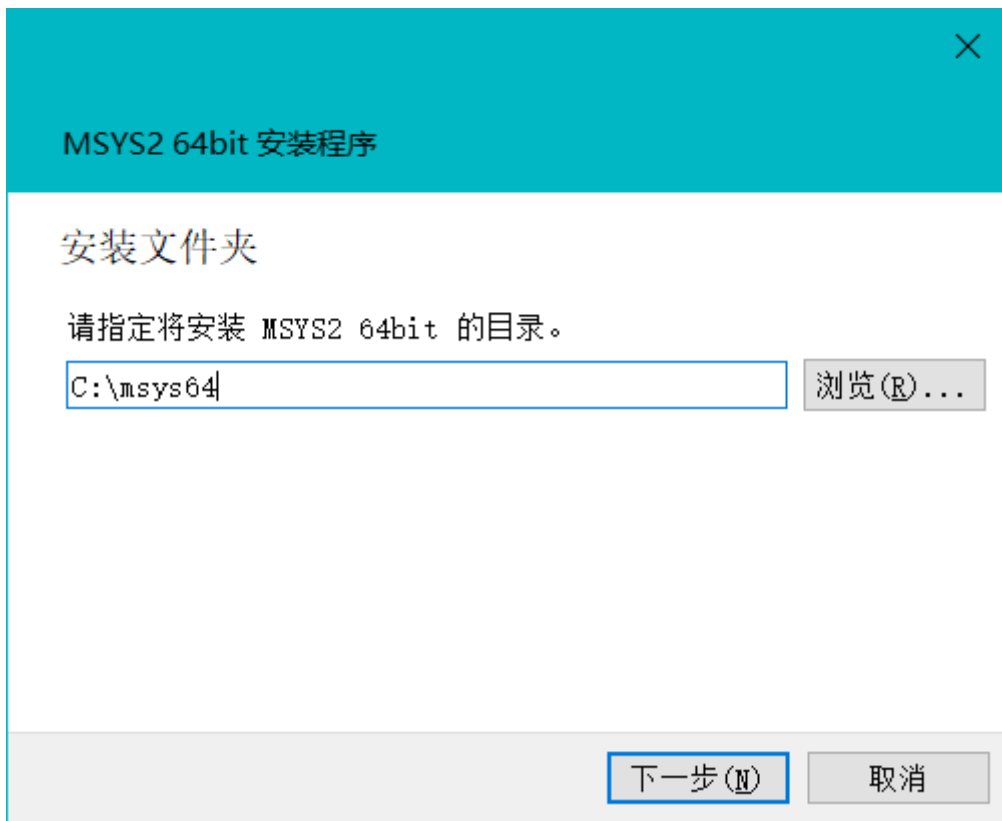
编译器：广义的编译器是指把一种语言翻译为另外一种语言，一般是把高级语言转换为低级语言。我们在编辑器里所写的代码不管是c，c++，java抑或是python，这些是想要让计算机执行的指令，但是计算机本身是看不懂这些源代码的，所以这时就需要编译器将我们写的程序源代码转换为计算机真正能执行的目标代码。主流的 C/C++ 编译器有 GCC 与 Clang，我们选择 Clang 一款即可编译轻量的 Cpp 文件了。

既然缺少了编译器，那就让我们一起安装编译器吧：

1. 在 <https://www.msys2.org>（如果官网打不开，直接 google msys2，在 sourceforge 上下载）下载 msys2（我们通过 msys2 来下载并安装 Clang）：



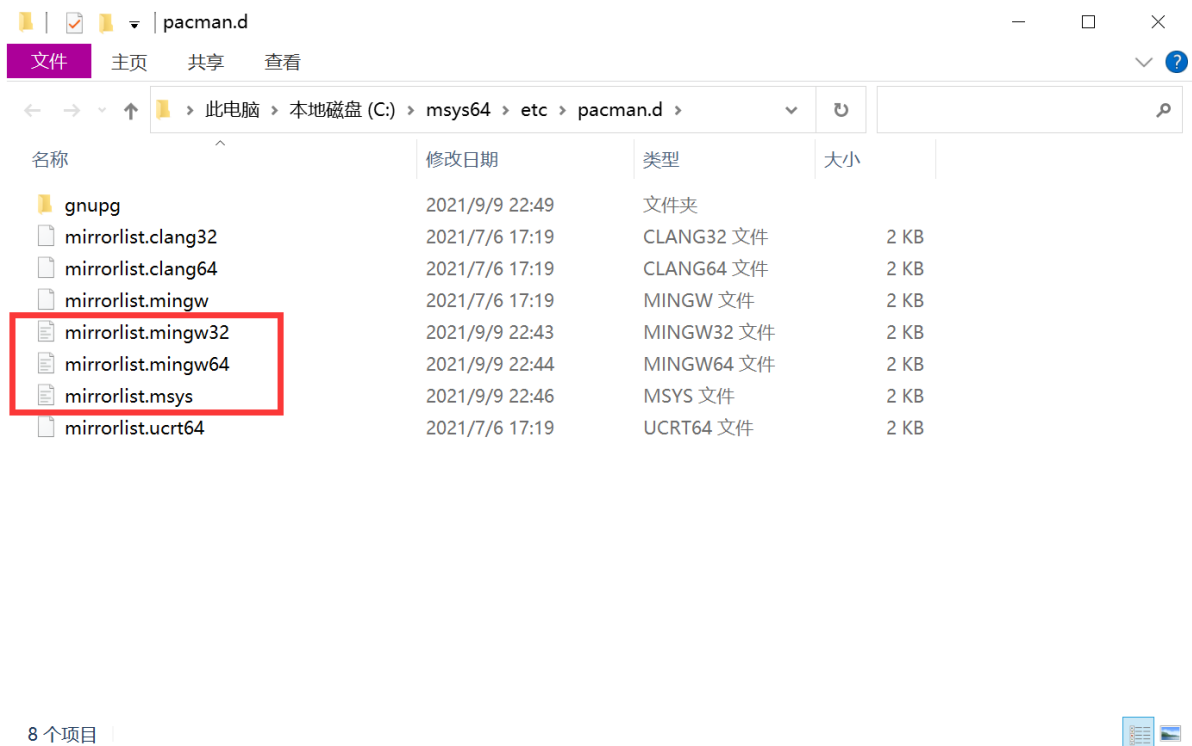
2. 启动 .exe 程序，进行安装：



msys2 默认安装在 C 盘，在这里我自己不做修改，安装路径读者可自行决定。

3. 安装完成后，配置 msys2:

打开 `C:\msys64\etc\pacman.d` 可以看到 `mirrorlist.mingw32`、`mirrorlist.mingw64`、`mirrorlist.msys` 三个文件



分别打开这三个文件，在文件末尾分别加入这几行：

`mirrorlist.mingw32`:

```
Server = https://mirrors.tuna.tsinghua.edu.cn/msys2/mingw/i686
```

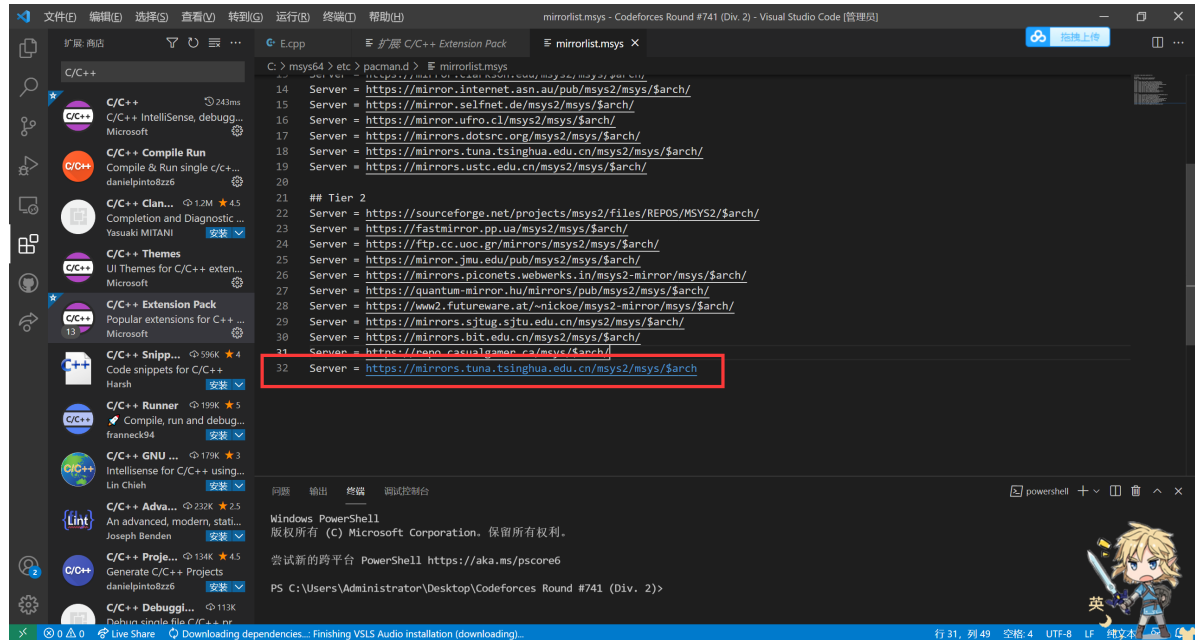

mirrorlist.mingw64:

```
Server = https://mirrors.tuna.tsinghua.edu.cn/msys2/mingw/x86_64
```

mirrorlist.msys:

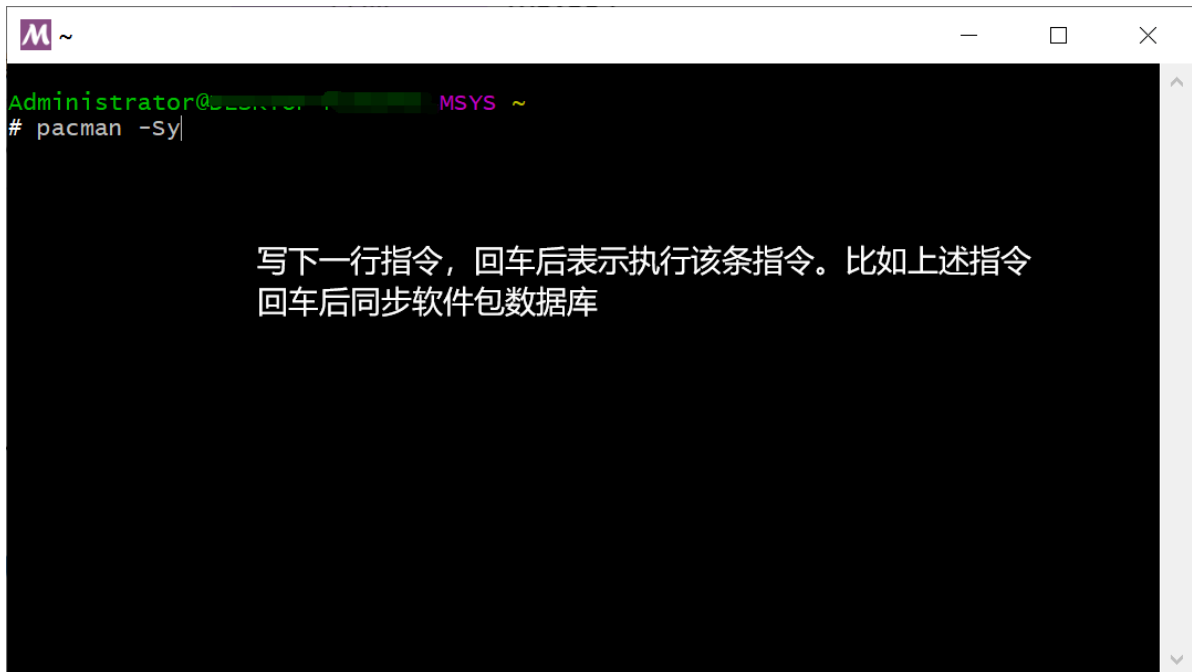
```
Server = https://mirrors.tuna.tsinghua.edu.cn/msys2/msys/$arch
```

mirrorlist.msys 示例:



4. 随后打开 msys2，输入 `pacman -sy` 同步软件包数据库，然后再输入 `pacman -su` 更新软件包，期间 msys2 提示你输入 y 之后会关闭，重新打开再输入一遍 `pacman -su` 即可（注意，你可以选择复制指令后，在 msys2.exe 黑框右键选择 paste，就可以粘贴进命令行中）：





A terminal window with a black background and white text. The title bar shows a purple 'M' icon and a tilde '~'. The prompt is 'Administrator@... MSYS ~'. The command '# pacman -Sy' is entered. Below the command, there is a large block of white text: '写下一行指令，回车后表示执行该条指令。比如上述指令回车后同步软件包数据库'.

```
Administrator@... MSYS ~
# pacman -Sy
```

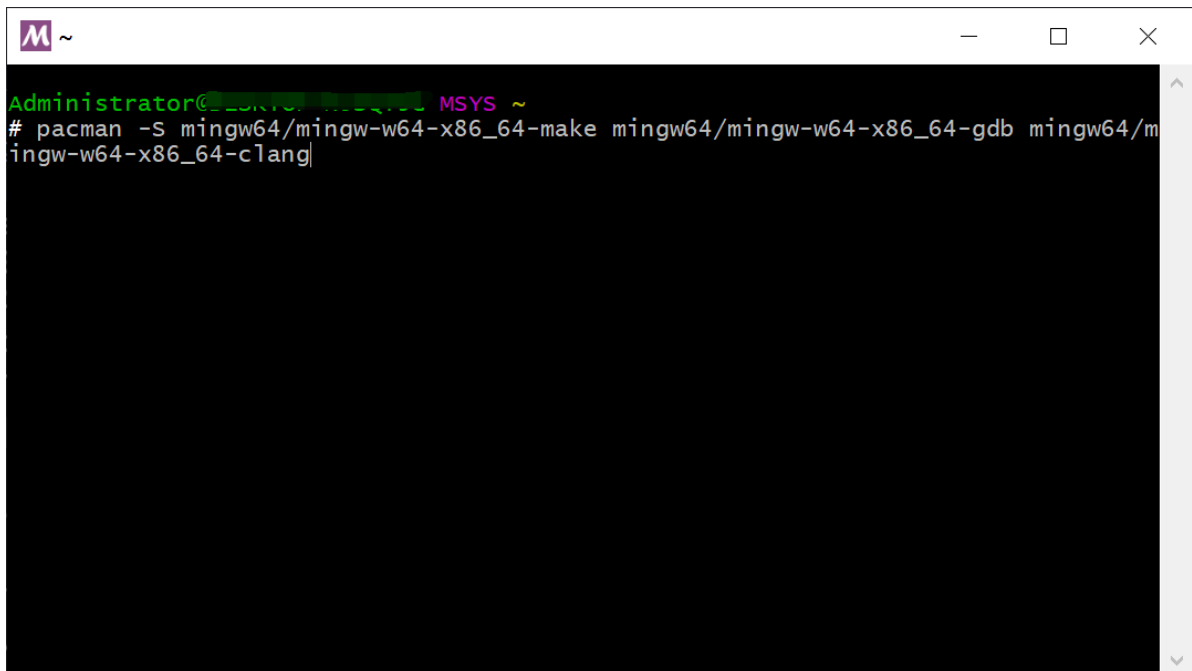
写下一行指令，回车后表示执行该条指令。比如上述指令回车后同步软件包数据库

(主机名打码了。不影响演示)

5. 打开 msys2，分别输入三条指令

```
pacman -S mingw64/mingw-w64-x86_64-make
pacman -S mingw64/mingw-w64-x86_64-gdb
pacman -S mingw64/mingw-w64-x86_64-clang
```

按照提示，就可以完成 Clang 的安装了：



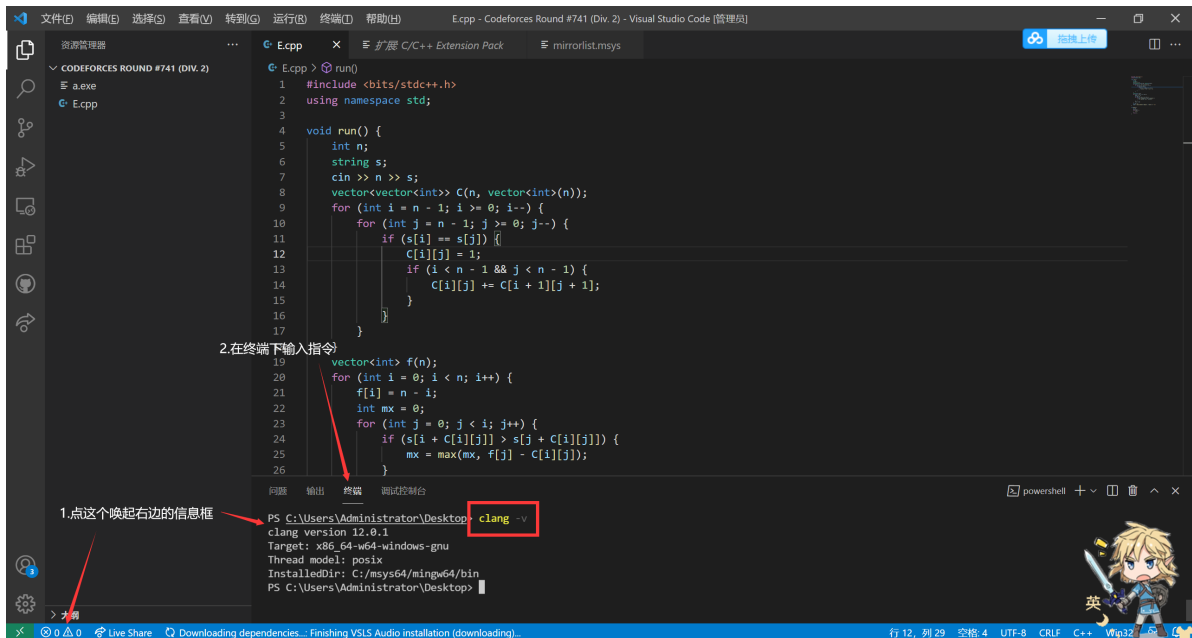
A terminal window with a black background and white text. The title bar shows a purple 'M' icon and a tilde '~'. The prompt is 'Administrator@... MSYS ~'. Three commands are entered on separate lines: '# pacman -S mingw64/mingw-w64-x86_64-make', 'mingw64/mingw-w64-x86_64-gdb', and 'mingw64/mingw-w64-x86_64-clang'.

```
Administrator@... MSYS ~
# pacman -S mingw64/mingw-w64-x86_64-make mingw64/mingw-w64-x86_64-gdb mingw64/mingw-w64-x86_64-clang
```

6. 添加环境变量

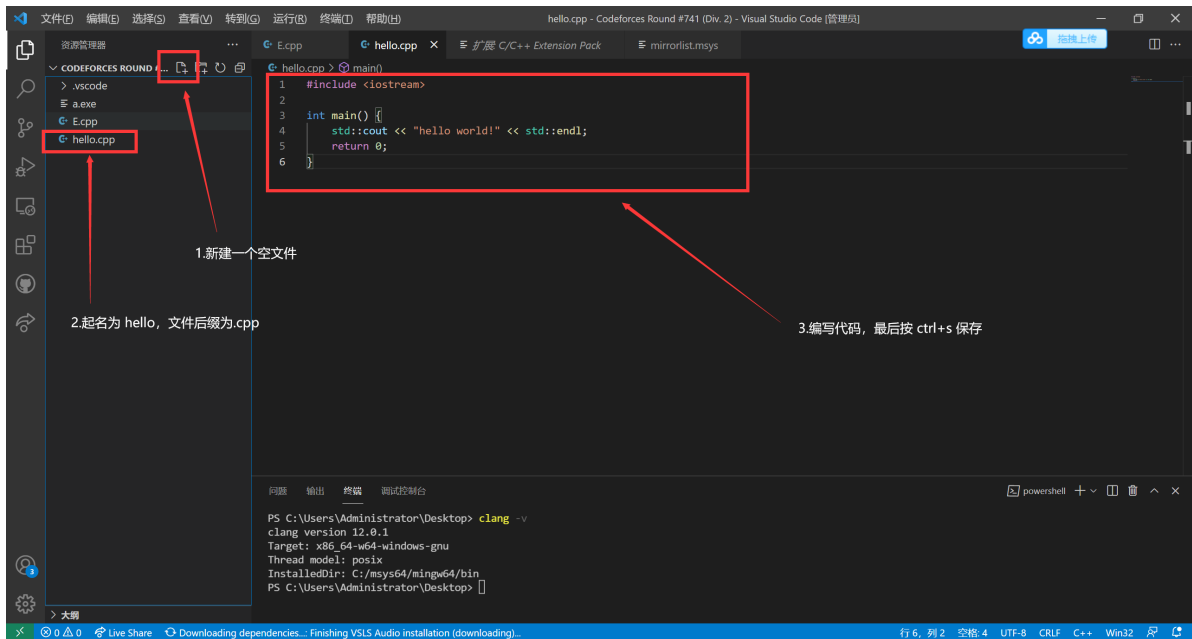
([关于如何将路径添加到系统变量 Path](#))

将 `C:\msys64\mingw64\bin` 添加到 Path 中（这路径根据前面 msys2 安装的位置决定。如果位置不一样，找到你安装目录下 `\msys64\mingw64\` 的 bin，将上面的路径换成 bin 文件的路径即可），在终端（可以在 VSCode 的终端下，以下图为例）输入 `clang -v`，看是否与下图返回的信息一致：

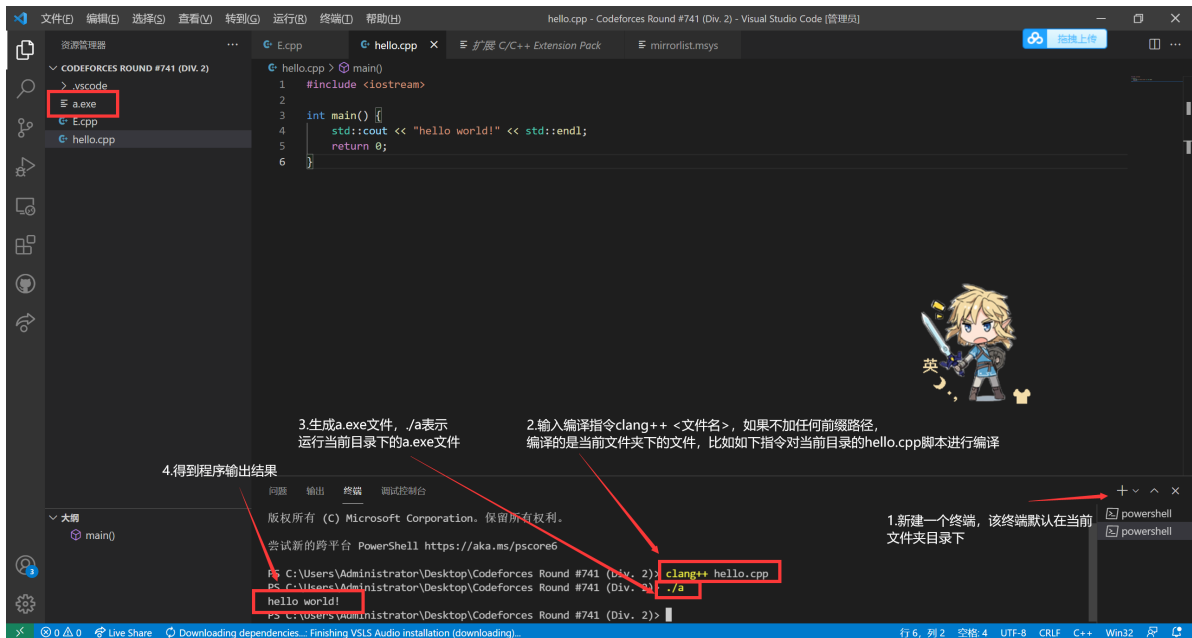


若信息基本一致（可能版本不同），那么你就已经成功安装 Clang 了！

让我们写下第一个程序吧：



接下来进行编译：



默认编译生成a.exe文件，想要给编译后的程序命名，可以加入 `-o` 指令。例如想生成 `b.exe`，则 `clang++ b.cpp -o "b.exe"`。

至此，你已经写下第一个程序并成功编译运行了，撒花 $\star, \circ, \star(\neg \nabla \neg) / \$. \circ \star$

后记

关于上述安装所遇到的问题，可以找开发区2021群 / 2021XCPC新生群提问，好心的学长学姐会帮你们解答。

提问之前请注意提问的艺术（如果您实在很直白，可以参照XCPC新生群的 提问的智慧-CPC版.pdf）

另外，本篇文章是开源的！有热心的学长学姐若对 C/C++ 编辑器编译器有更深入的理解，欢迎 Pull Request [Falicitas/write-your-first-Cpp-Code](https://github.com/kineXense/write-your-first-Cpp-Code)（gitee同源仓库：<https://gitee.com/kineXense/write-your-first-Cpp-Code>）。