CMake多語言編譯\_安裝與使用

[1. 前言 2](#_Toc189636570)

[2. 安裝CMake 2](#_Toc189636571)

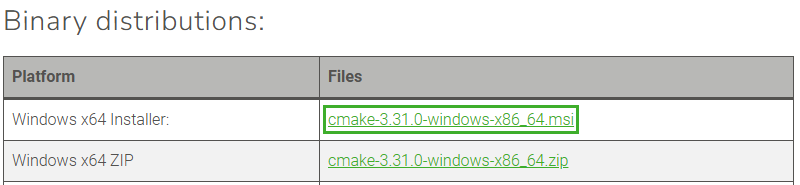
[3. 指令 2](#_Toc189636572)

[4. 編譯\*.cpp範例 2](#_Toc189636573)

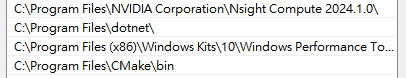
# 前言

# 安裝CMake

<https://cmake.org/download/>



得cmake-3.31.0-windows-x86\_64，雙擊安裝成功後須重啟電腦，以取得環境變數。



# 指令

* cmake –version

cmake version 3.31.0

# 編譯範例1

## 準備專案

// main.cpp

#include <iostream>

int main() {

std::cout << "Hello, CMake!" << std::endl;

return 0;

}

## 建立CMakeLists.txt檔案

在與 main.cpp 同一目錄下建立一個名為 CMakeLists.txt 的檔案，內容如下：

#---設定最低 CMake 版本要求

cmake\_minimum\_required(VERSION 3.10)

#--- 定義專案名稱，如CMakeTest1

project(CMakeTest1)

#---設定 C++版本，如使用 C++17，可改

set(CMAKE\_CXX\_STANDARD 17)

set(CMAKE\_CXX\_STANDARD\_REQUIRED True)

#---定義一個編譯指令，以便將main.cpp 編譯MakeTest1.exe

add\_executable(CMakeTest1 main.cpp)

## 建置編譯系統

建議在專案目錄下建立一個獨立的 build 目錄。請開啟終端機或命令提示字元，然後依序執行以下命令：

md D:\1\_SyscoPY\_D\0\_Basic\1\_CMake\1\_CMakeTest1

md D:\1\_SyscoPY\_D\0\_Basic\1\_CMake\1\_CMakeTest1\Build

cd D:\1\_SyscoPY\_D\0\_Basic\1\_CMake\1\_CMakeTest1\Build

cmake .. # 令 CMake 使用上層目錄中的 CMakeLists.txt 建置編譯系統檔案。

## 編譯專案

cmake --build .

這個命令會用適當的編譯工具(如在Unix/Linux上的make 或在 Windows上生成Visual Studio解決方案)來編譯程式。編譯成功後，你應該能在 build 目錄中看到生成的執行檔 CMakeTest1.exe。

## 執行生成的exe

你現在可以運行編譯後的程式：

* Windows 上：

MyProgram.exe

* Linux/macOS 上**：**

./MyProgram

你應該會看到以下輸出：

Hello, CMake!