

Better IDE: Coding with C/C++

Yu-Hsuan Chen

Outline

- 為何不用Dev-C++
- Visual Studio Code + GNU 介紹
 - 如何安裝GNU C Compiler
 - GCC / G++指令與終端機操作
 - VSCode上不用終端機的替代方案
- [Bonus Pack] Code::Blocks 安裝與中文化

棄用Dev-C++的幾個理由

過時的版本

- Dev-C++ 最後一次更新已經是2015年，版本號停留在5.1.1
- 內附的GNU Compiler也停留在4.9.2



不靈活的 Formatter

與其說他不靈活
不如說他笨

清楚的排版可以讓code
一目瞭然

```
[*] 新文件1
1 #include<conio.h>
2 #include<stdio.h>
3 int main()
4 {
5     float e[100][100],w[100][100],p[100],q[100],t;
6     int i,n,j,k,l,m,r,root[100][100];
7     printf("this is optimal binary search tree\n");
8     printf("enter the value of n\n");
9     scanf("%d",&n);
10    printf("enter the values of the probability\n");
11    for(i=1;i<=n;i++)
12    {
13        scanf("%f",&p[i]);
14    }
15    printf("enter the values of the probability of dummy\n");
16    for(i=0;i<=n;i++)
17    {
18        scanf("%f",&q[i]);
19    }
20    for(i=1;i<=n+1;i++)
21    {
22        e[i][i-1]=q[i-1];
23        w[i][i-1]=q[i-1];
24    }
25    for(l=1;l<=n;l++)
26    {
27        for(i=1;i<=n-l+1;i++)
28    {
29        j=i+l-1;
30        e[i][j]=100.0;
31        w[i][j]=w[i][j-1]+p[j]+q[j];
32        for(r=i;r<=j;r++)
33    {
34            t=e[i][r-1]+e[r+1][j]+w[i][j];
35            if(t<e[i][j])
36    {
37                e[i][j]=t;
38                root[i][j]=r;
39            }
40        }
41    }
42 }
```

上面的code 直接丟進垃圾桶，NOW
有Bug也不要找我，自己解決

堪用的 Debugger

Debugger的Trace功能簡
單、有時候會無法用

A screenshot of a debugger's code editor window. The file is named 'testset2.cpp'. The code contains a main function that initializes an array 'arr' with values from 0 to 1000. A for loop iterates through the array, setting each element to its index. The line 'arr[i] = i;' is highlighted in blue, indicating it is the current instruction being executed or selected. The code editor has syntax highlighting for C++ keywords and comments. Below the code editor is a toolbar with various debugger-related icons and buttons.



謎樣的
optimization
某一些非正規的寫法
Dev-C++會自己幫你修正

```
#include<iostream>

int main()
{
    int a;
    std::cout << a;
}
```

思考一下上面的a值應該會印出什麼?

```
#include<iostream>

int main()
{
    int a;
    std::cout << a;
}
```

正常來說，應該是個任意的亂數
但是Dev-C++會自己幫你初始化成0啊

```
1 // Example program
2 #include <iostream>
3 #include <string>
4
5 int main()
6 {
7     int a;
8     std::cout << a;
9 }
10 8:19: warning: 'a' is used uninitialized in this function [-Wuninitialized]
```

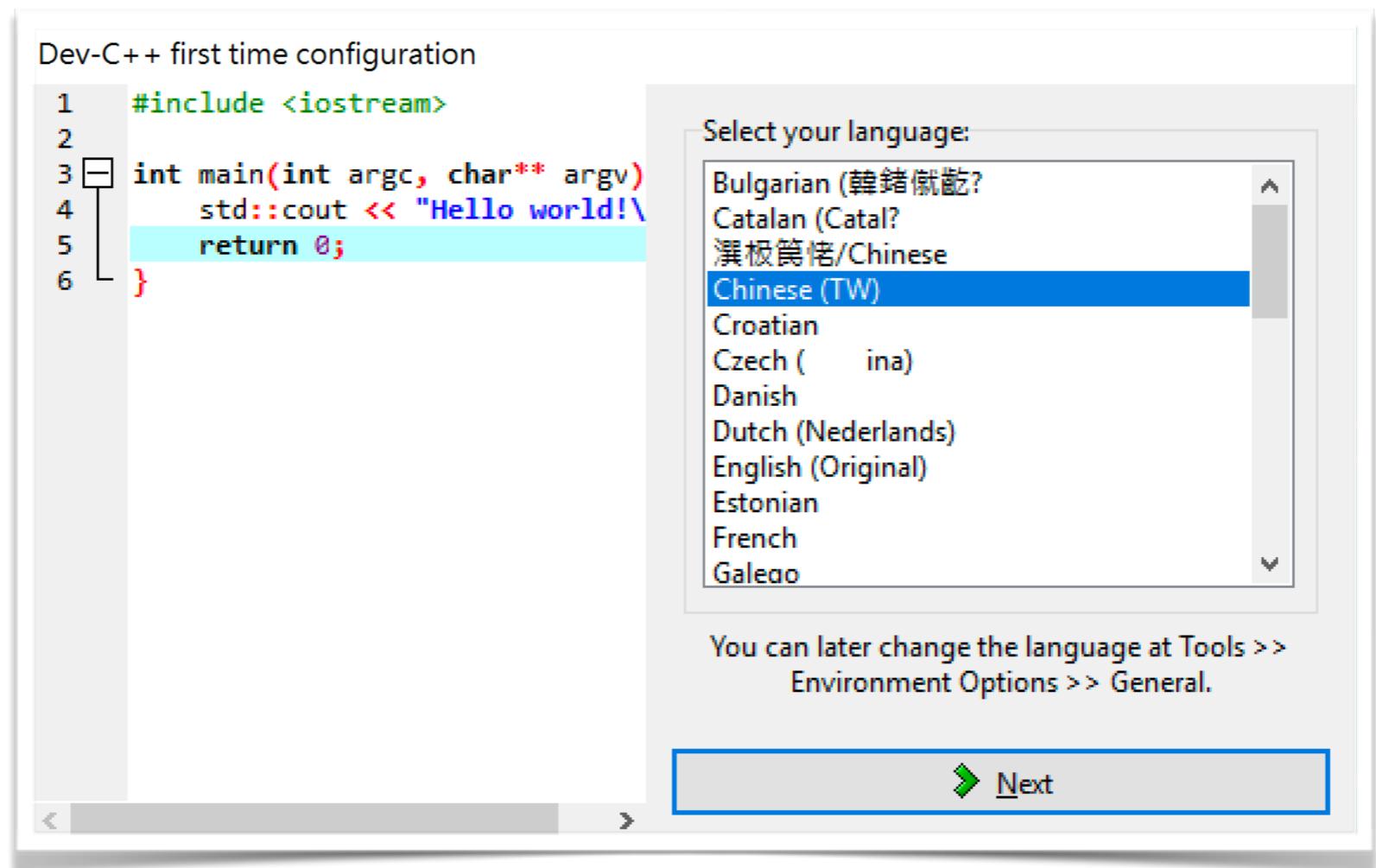
這種用法Compiler通常會
提出Warning的

因為Dev-C++幫你了一把
也沒有主動告訴你出問題了
然後你就覺得答案正確，開開心心上傳E-tutor

通過	執行結果
是	AC: 完全正確
是	AC: 完全正確
是	AC: 完全正確
否	WA: 錯誤的結果
是	AC: 完全正確

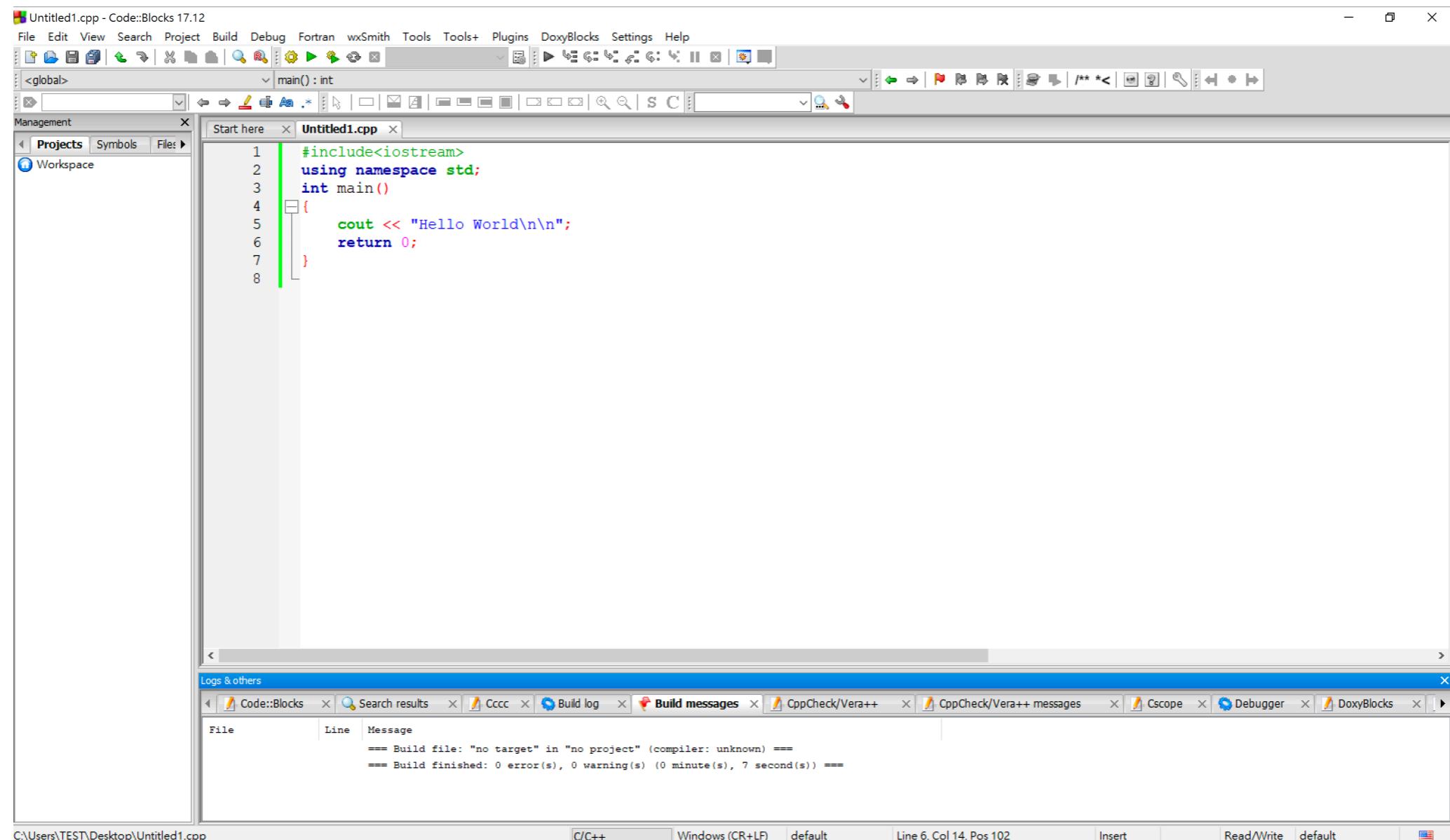
看吧，這下WA了吧，而且是保密測資

那為什麼教學現場還是這麼多老師喜歡用？



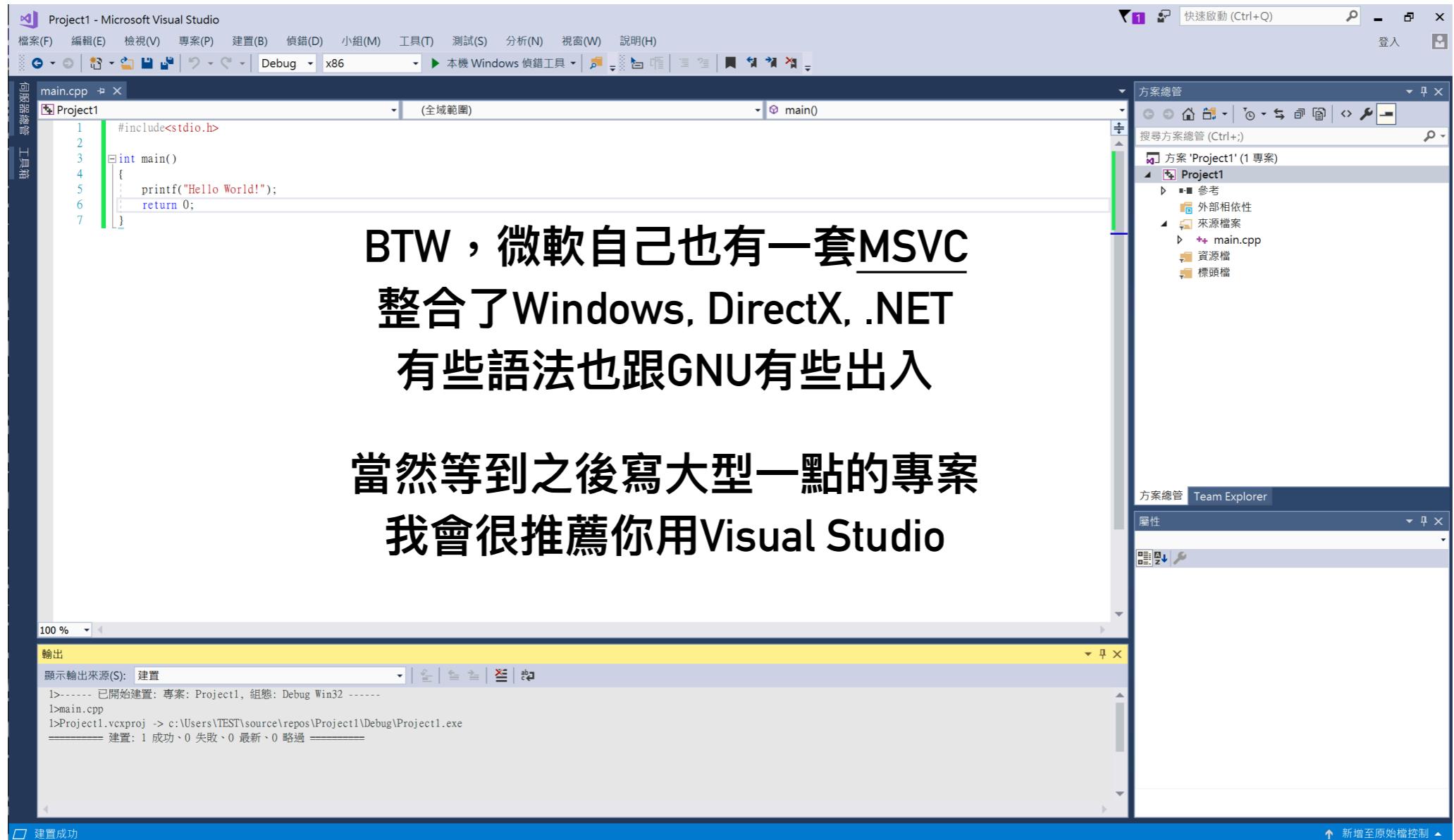
因為他是少數簡單好上手、介面有中文的IDE

讓我們看看其他IDE的長相...



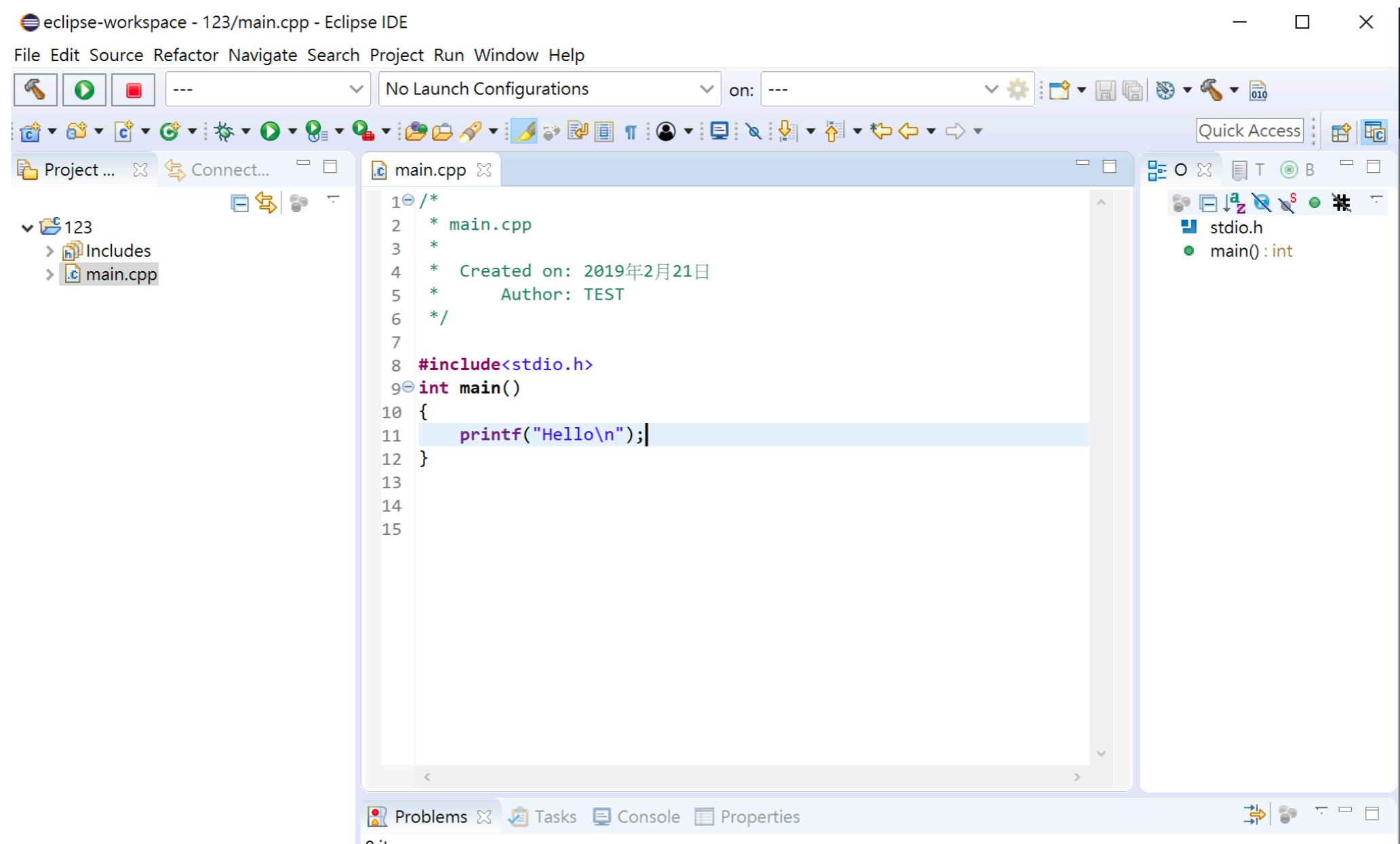
Code::Blocks

- 最新版本20.03
- 沒有內建中文介面
需要額外下載語系檔套用
- 具自動完成功能



Visual Studio

- 最新版本2019
- 內建正體中文
- 功能最強大、介面較複雜
- 占空間(VC環境約需6GB)



Eclipse

- 最新版本2020-03
- 部分中文化(有語言包)
- 不好說(裝Eclipse之前還需要裝Java)

綜觀下來的結論就是

要中文介面，要馬簡單到爆、不然就很巨大

沒有中文 就會讓大家對程式設計感到害怕

所以...

**如果沒有強大的IDE
何不自己組一個？**



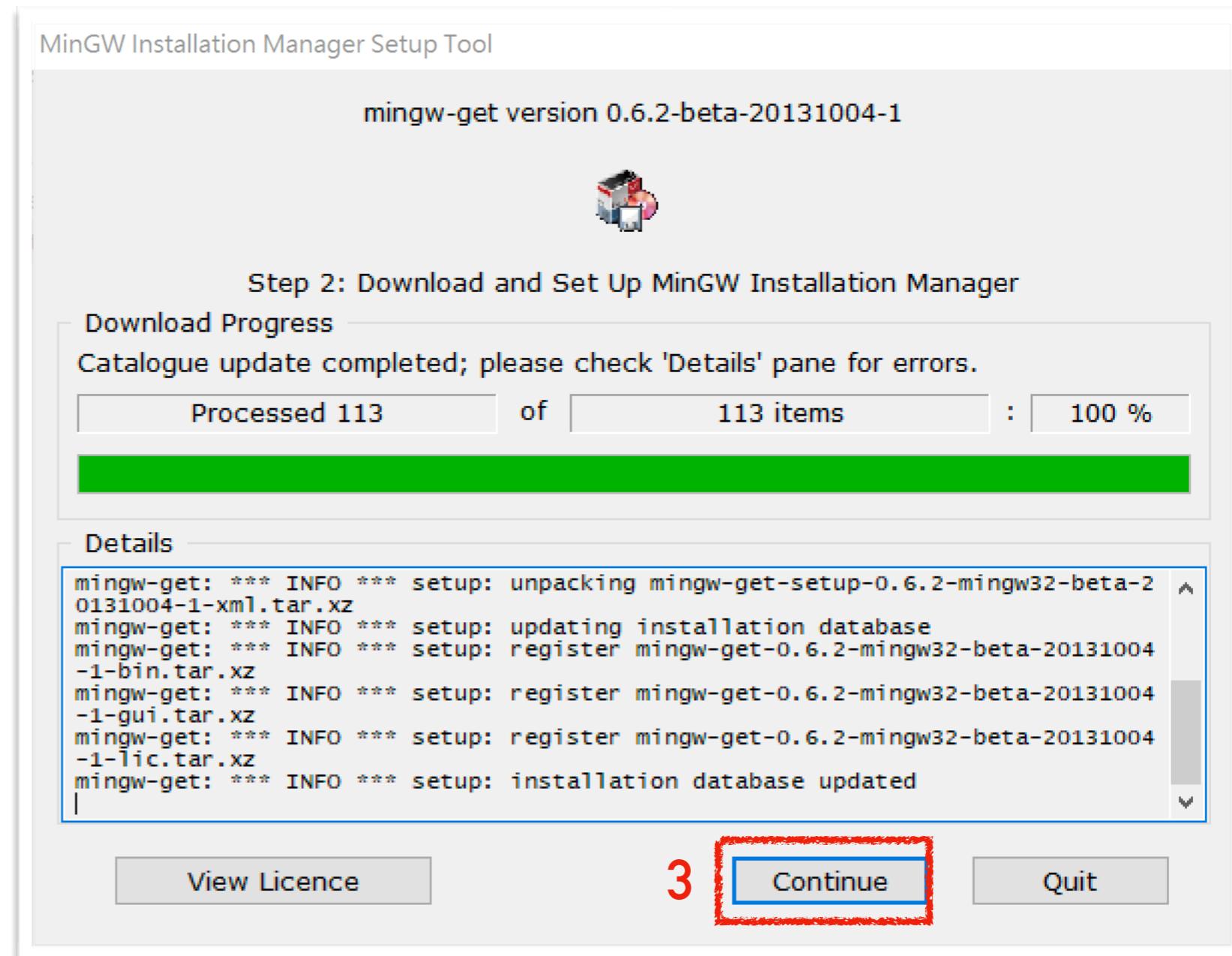
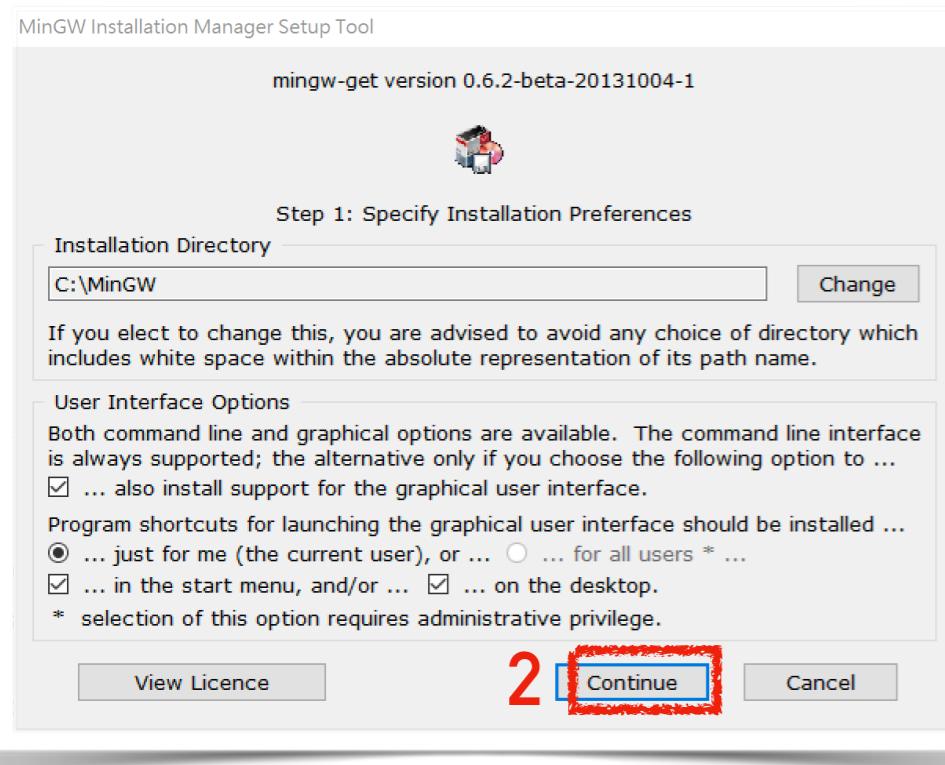
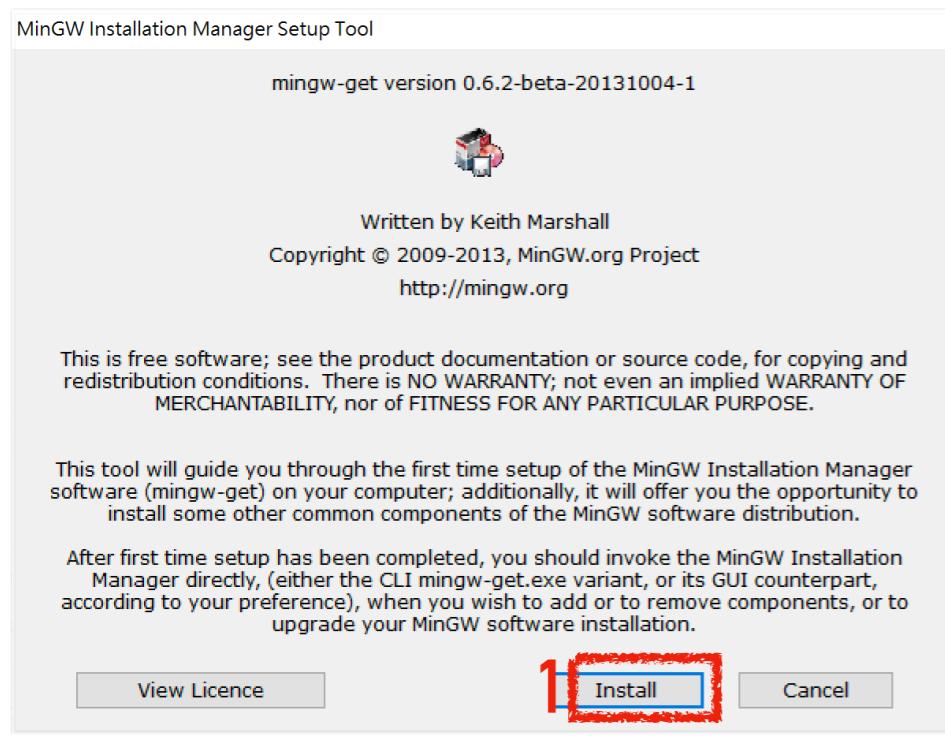
今日兩大主角

Visual Studio Code + GCC Compiler

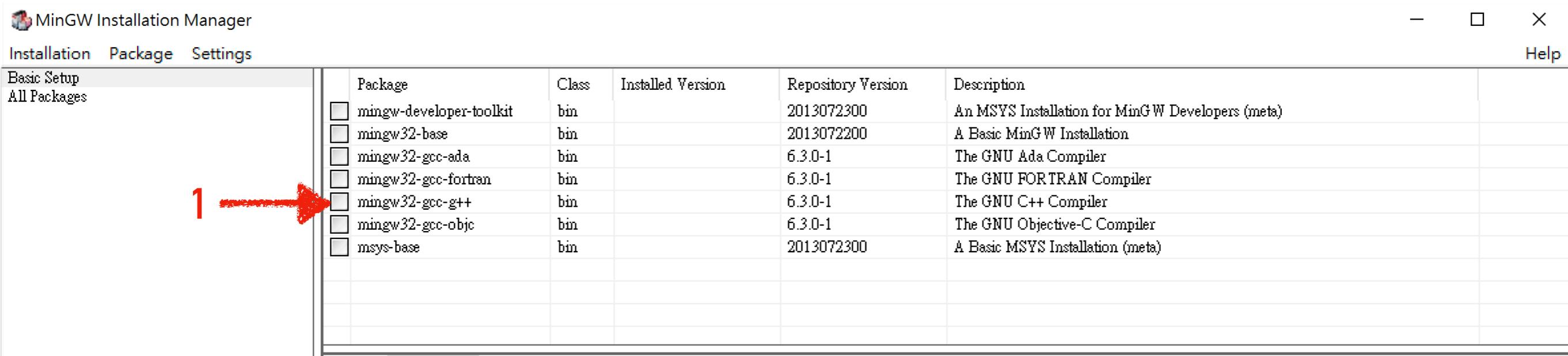
安裝MinGW 1

- MinGW是將GCC編譯器和GNU Binutils移植到Windows底下的產物。我們要透過MinGW下載GNU C的編譯器
- 下載MinGW Installer：
<https://sourceforge.net/projects/mingw/>

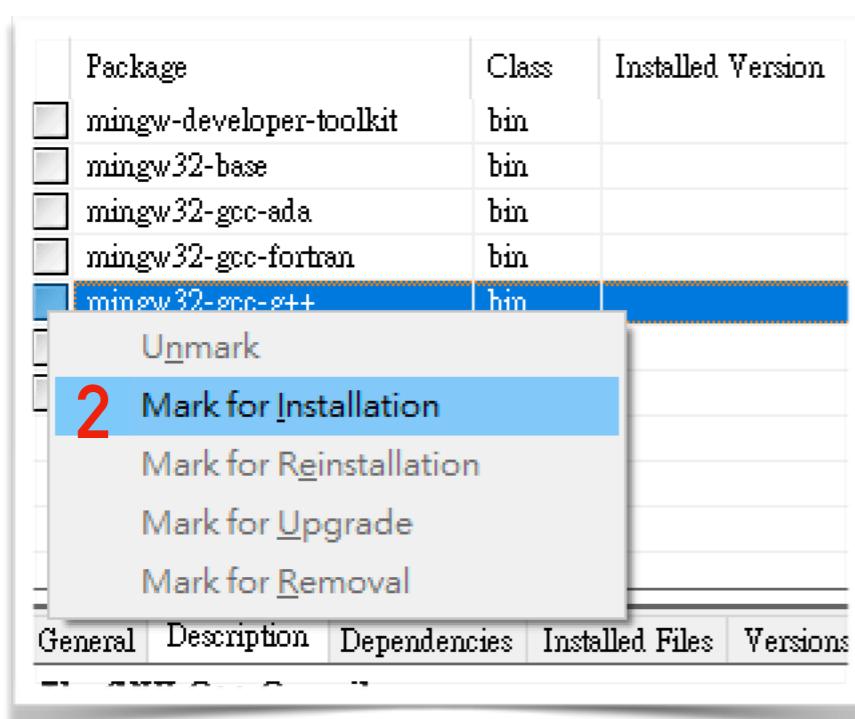
安裝MinGW 2



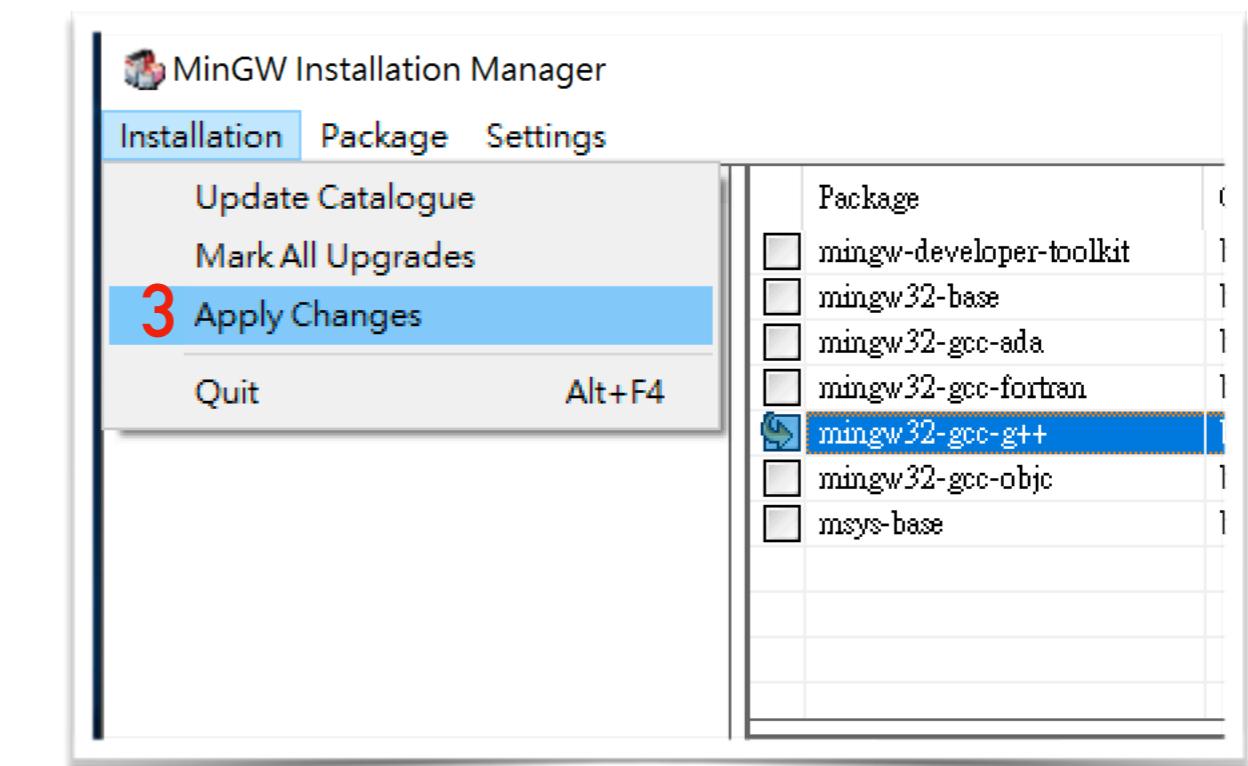
安裝MinGW 3



1



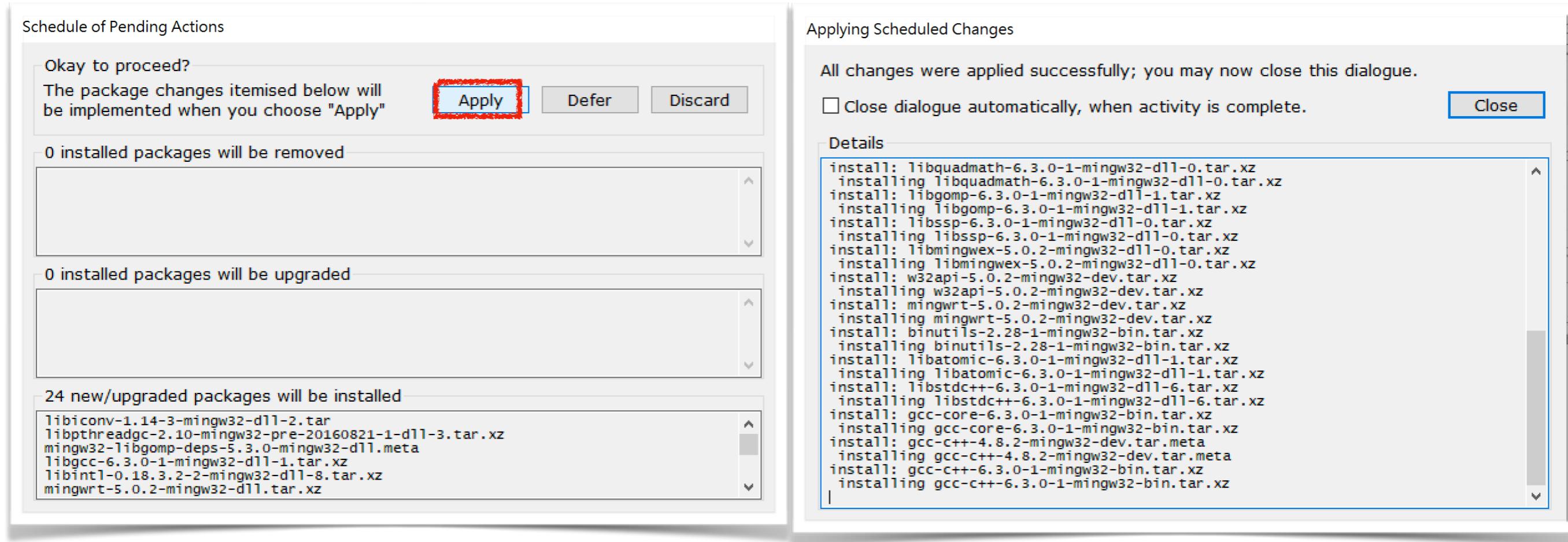
2



3

安裝MinGW 4

- 點擊Apply開始下載必要元件
完成之後就可以點擊Close關閉



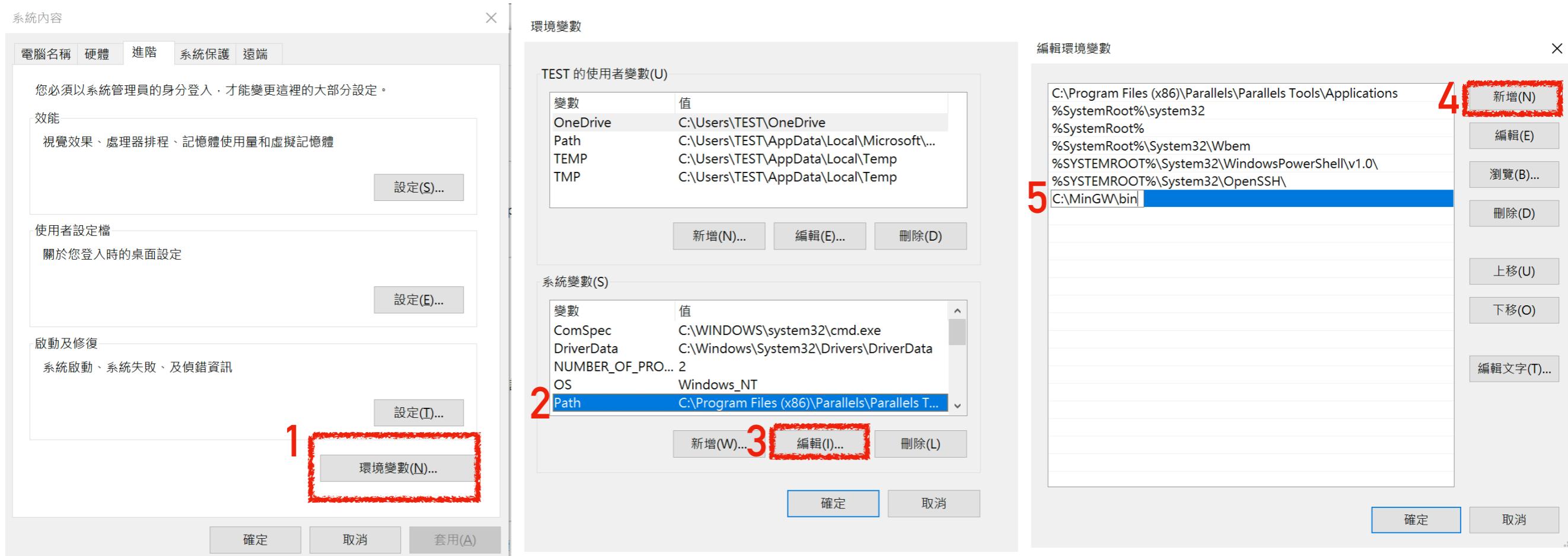
設定GCC PATH 1

- 控制台 > 系統與安全性 > 系統 > 進階系統設定



設定GCC PATH 2

- 在系統變數中加入 **C:\MinGW\bin**
加入完成點擊確定關閉所有對話框



設定GCC PATH 3

- 打開命令提示字元，輸入gcc -v 跟 g++ -v
如果可以正確看到GCC版本就表示成功



```
命令提示字元
Microsoft Windows [版本 10.0.17763.1]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

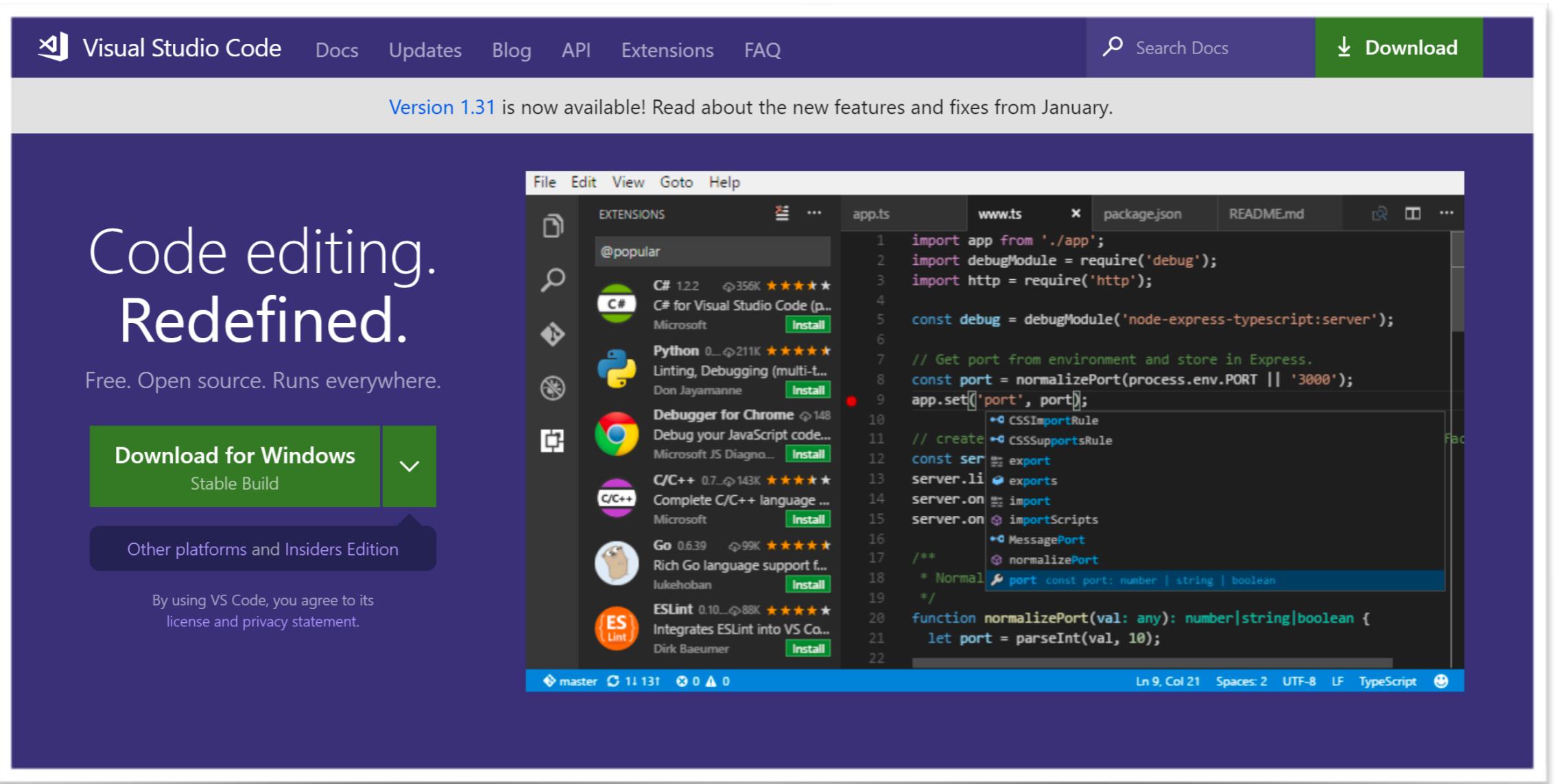
C:\Users\TEST>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=c:/mingw/bin/../libexec/gcc/mingw32/6.3.0/lto-wrapper.exe
Target: mingw32
Configured with: ./src/gcc-6.3.0/configure --build=x86_64-pc-linux-gnu --host=mingw32 --target=mingw32 --with-gmp=/mingw --with-mpfr --with-mpc=/mingw --with-isl=/mingw --prefix=/mingw --disable-win32-registry --with-arch=i586 --with-tune=generic --enable-languages=c,c++,objc,obj-c++,fortran,ada --with-pkgversion='MinGW.org GCC-6.3.0-1' --enable-static --enable-shared --enable-threads --with-dwarf2 --disable-sjlj-exceptions --enable-version-specific-runtime-libs --with-libiconv-prefix=/mingw --with-libintl-prefix=/mingw --enable-stdcxx-debug --enable-libgomp --disable-libvtv --enable-nls
Thread model: win32
gcc version 6.3.0 (MinGW.org GCC-6.3.0-1)

C:\Users\TEST>g++ -v
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=c:/mingw/bin/../libexec/gcc/mingw32/6.3.0/lto-wrapper.exe
Target: mingw32
Configured with: ./src/gcc-6.3.0/configure --build=x86_64-pc-linux-gnu --host=mingw32 --with-gmp=/mingw --with-mpfr=/mingw --with-mpc=/mingw --with-isl=/mingw --prefix=/mingw --disable-win32-registry --target=mingw32 --with-arch=i586 --enable-languages=c,c++,objc,obj-c++,fortran,ada --with-pkgversion='MinGW.org GCC-6.3.0-1' --enable-static --enable-shared --enable-threads --with-dwarf2 --disable-sjlj-exceptions --enable-version-specific-runtime-libs --with-libiconv-prefix=/mingw --with-libintl-prefix=/mingw --enable-stdcxx-debug --with-tune=generic --enable-libgomp --disable-libvtv --enable-nls
Thread model: win32
g++ version 6.3.0 (MinGW.org GCC-6.3.0-1)

C:\Users\TEST>
```

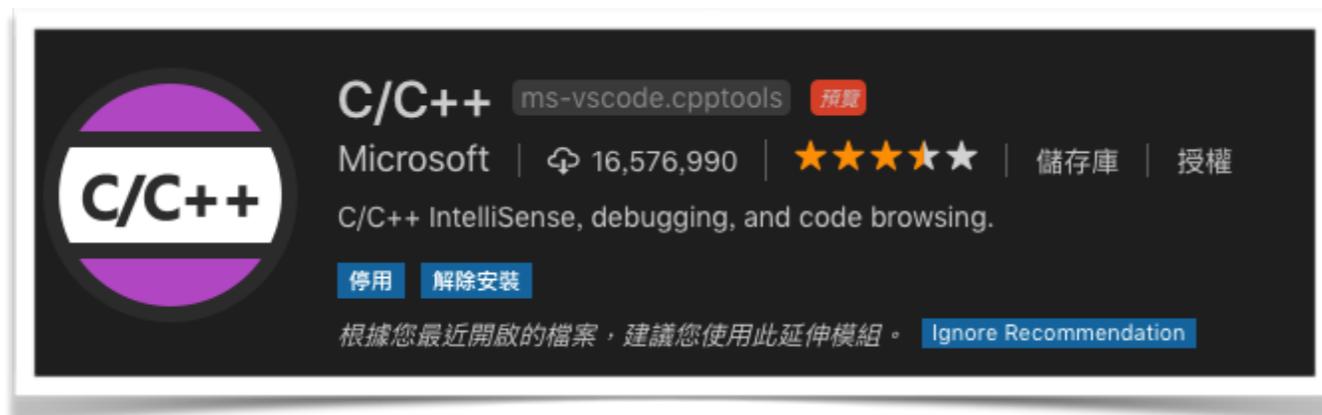
安裝Visual Studio Code

- 下載網址：<https://code.visualstudio.com/>



下載Code的額外套件

- Visual Studio Code本身是開源在Github的，有海量的第三方套件可以使用，而且VSCode本體更新的速度很勤快（大約一個月一次版本更新，有BUG幾乎不會拖太久）
- 請下載以下套件
 - Chinese (Traditional) Language Pack
 - C/C++ (ms-vscode.cpptools)



開始寫第一份Code

The screenshot shows a code editor window with two tabs: 'Untitled-1' and 'test.cpp'. The 'test.cpp' tab is active, displaying the following code:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     printf("Hello World\n!!");
6     return 0;
7 }
```

A red arrow points from the text '較可惜的地方是這些元件還無法及時檢查出語法的錯誤 (例如忘記打分號)' to the code editor area.

The code editor's status bar at the bottom indicates: 第 6 行, 第 14 欄 空格: 4 UTF-8 LF 純文字

The 'Intellisense' dropdown menu is open over the word 'int' in line 7. The menu title is '擴充功能 C++ Intellisense'. The list shows 'C++ keyword' options:

- int
- int16_t
- int32_t
- int64_t
- int8_t

Text overlay on the right side of the image reads:

保存檔案之後
自動完成的功能就會開始輔助你
當然隨時Ctrl + S是好習慣

較可惜的地方是這些元件還無法及時
檢查出語法的錯誤 (例如忘記打分號)

這裡可以切換這份檔案的語言
(快速鍵 : Ctrl + K , M)

切換至C/C++之後對應的附加元件就會啟動



編譯寫好的程式碼 1

Code裡有整合系統的終端機
打程式、測試都可以在整個Code完成

2

問題 輸出 偵錯主控台 **終端機**

Microsoft Windows [版本 10.0.17763.1]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\TEST>cd desktop

C:\Users\TEST\Desktop>

該怎麼用終端機Compile?

1

(Global Scope) 第 1 行 · 第 1 欄 空格: 4 UTF-8 CRLF C++ Win32 ☺ 🔔

編譯寫好的程式碼 2

- 因為Code本身沒有編譯按鈕，所以需要以終端機輔助
- 基本指令：
 gcc [檔案名稱] -o [輸出檔案名稱]
 g++ [檔案名稱] -o [輸出檔案名稱]

gcc表示用C編譯，g++表示用C++編譯

gcc main.c

gcc main.c -o main.exe

g++ main.cpp

g++ main.cpp -o main.exe

如果沒有附加 -o 參數
預設產生的執行檔名字為 a.exe

執行程式 1

- 依照使用的終端機差異，執行程式的方法也有點不一樣。

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17763.1]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權
C:\Users\TEST\Desktop>g++ test.cpp -o test.exe
C:\Users\TEST\Desktop>test
Hello World!!
C:\Users\TEST\Desktop>test.exe
Hello World!!
C:\Users\TEST\Desktop>
```

使用cmd

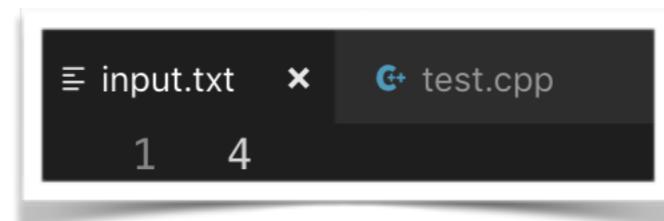
**在PowerShell要執行檔案
前端需要加上「./」**

```
Windows PowerShell
Windows PowerShell
著作權 (C) Microsoft Corporation. 著作權所有，並保留一切權利。
PS C:\Users\TEST\Desktop> g++ test.cpp -o test.exe
PS C:\Users\TEST\Desktop> ./test.exe
Hello World!!
PS C:\Users\TEST\Desktop> ./test
Hello World!!
PS C:\Users\TEST\Desktop> test
test : 無法辨識 'test' 詞彙是否為 Cmdlet、函數、指令檔或可執行程式
確認路徑是否正確，然後再試一次。
位於 線路:1 字元:1
```

使用PowerShell

執行程式 2 檔案輸入/輸出

```
1 #include<stdio.h>
2 int main()
3 {
4     int input;
5     scanf("%d",&input);
6     printf("%d\n",input);
7 }
```



這是一段具有鍵盤輸入、螢幕輸出的程式碼
假設這個編譯完成的檔案是test.exe

input.txt是這個程式的輸入

test.exe < input.txt

← test.exe接受來自input.txt之輸入
結果輸出在畫面上

test.exe < input.txt > output.txt

← test.exe接受來自input.txt之輸入
結果輸出到output.txt中

test.exe > output.txt

← test.exe接受來自鍵盤的輸入
結果輸出到output.txt中

執行程式 3

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17763.1]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\TEST\Desktop>test.exe <input.txt
Your Input: 100

C:\Users\TEST\Desktop>test.exe <input.txt> output.txt

C:\Users\TEST\Desktop>test.exe > output.txt
100

C:\Users\TEST\Desktop>
```

不管是要單獨開cmd或
在VSCode整合的終端機操作都可以

```
問題 輸出 偵錯主控台 終端機 1: cmd

Microsoft Windows [版本 10.0.17763.316]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\TEST>cd Desktop

C:\Users\TEST\Desktop>test.exe <input.txt
Your Input: 100

C:\Users\TEST\Desktop>test.exe <input.txt> output.txt

C:\Users\TEST\Desktop>test.exe > output.txt
100

C:\Users\TEST\Desktop>
```

The screenshot shows the VS Code interface with three tabs open:

- test.cpp**: The code is as follows:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int input;
6     scanf("%d",&input);
7     printf("Your Input: %d\n",input);
8 }
```
- input.txt**: Contains the text "1 100".
- output.txt**: Contains the text "1 Your Input: 100".

執行結果

執行程式 4 關於EOF(End of file)

- 在OnlineJudge平台中，大多是用輸入檔案來做測試，所以自然會在讀完檔案之後收到EOF。

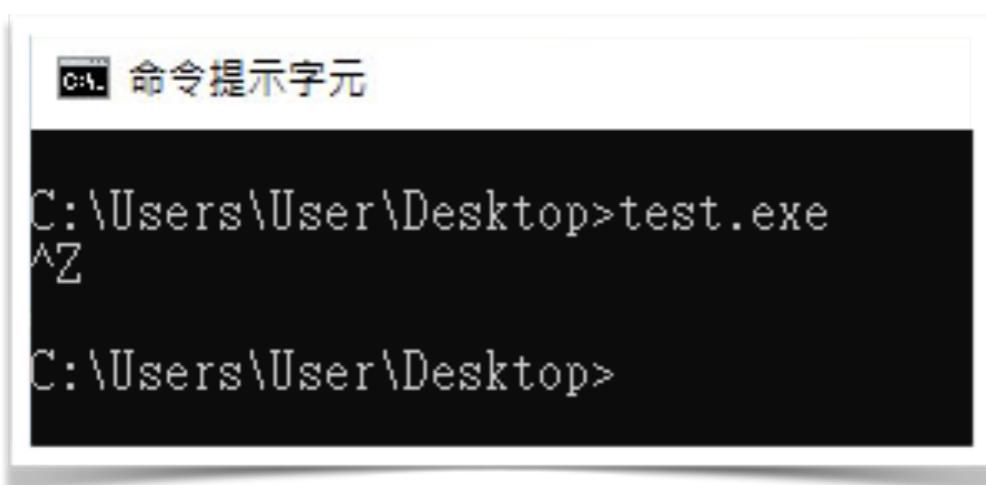
Like this : out.exe <input1.txt> output1.txt

之後再把output1.txt拿來比對是否一致，對了就Accept(AC)

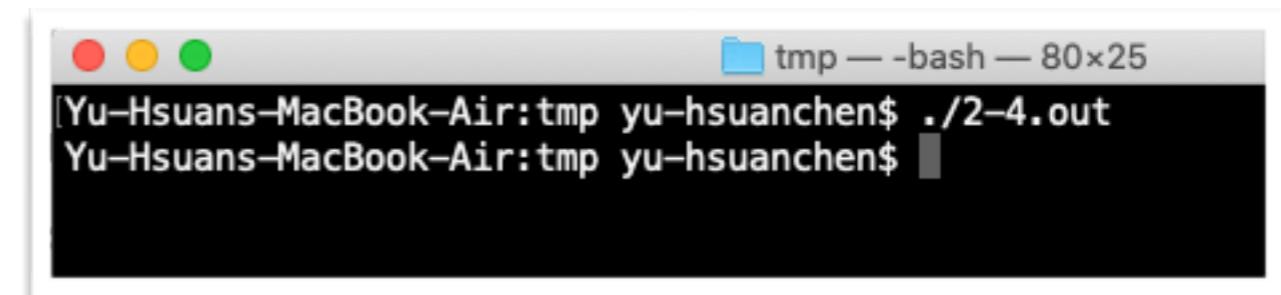
- 用鍵盤輸入，傳送EOF訊號：

Windows的使用者：按下**Ctrl + Z**（畫面上會出現^Z），再按Enter

macOS/Linux的使用者：按下**Ctrl + D**（畫面上不會顯示任何東西）



```
命令提示字元  
C:\Users\User\Desktop>test.exe  
^Z  
C:\Users\User\Desktop>
```



```
tmp — -bash — 80x25  
[Yu-Hsuans-MacBook-Air:tmp yu-hsuanchen$ ./2-4.out  
Yu-Hsuans-MacBook-Air:tmp yu-hsuanchen$ ]
```

可是我在macOS的終端機按Ctrl + Z好像也有反應耶？

```
[Yu-Hsuans-MacBook-Air:tmp yu-hsuanchen$ ./2-4.out  
6-320-13394-7  
6-320-13394-5  
[Yu-Hsuans-MacBook-Air:tmp yu-hsuanchen$ ./2-4.out  
^Z  
[1]+ Stopped ./2-4.out  
Yu-Hsuans-MacBook-Air:tmp yu-hsuanchen$ ]
```

？？？？？

```
[Yu-Hsuans-MacBook-Air:tmp yu-hsuanchen$ ./2-4.out  
6-320-13394-7  
6-320-13394-5  
[Yu-Hsuans-MacBook-Air:tmp yu-hsuanchen$ ./2-4.out  
^Z  
[1]+ Stopped ./2-4.out  
[Yu-Hsuans-MacBook-Air:tmp yu-hsuanchen$ ps  
 PID TTY TIME CMD  
 4455 ttys000 0:00.05 -bash  
 4463 ttys000 0:00.01 ./2-4.out  
[Yu-Hsuans-MacBook-Air:tmp yu-hsuanchen$ kill 4463  
[1]+ Terminated: 15 ./2-4.out
```

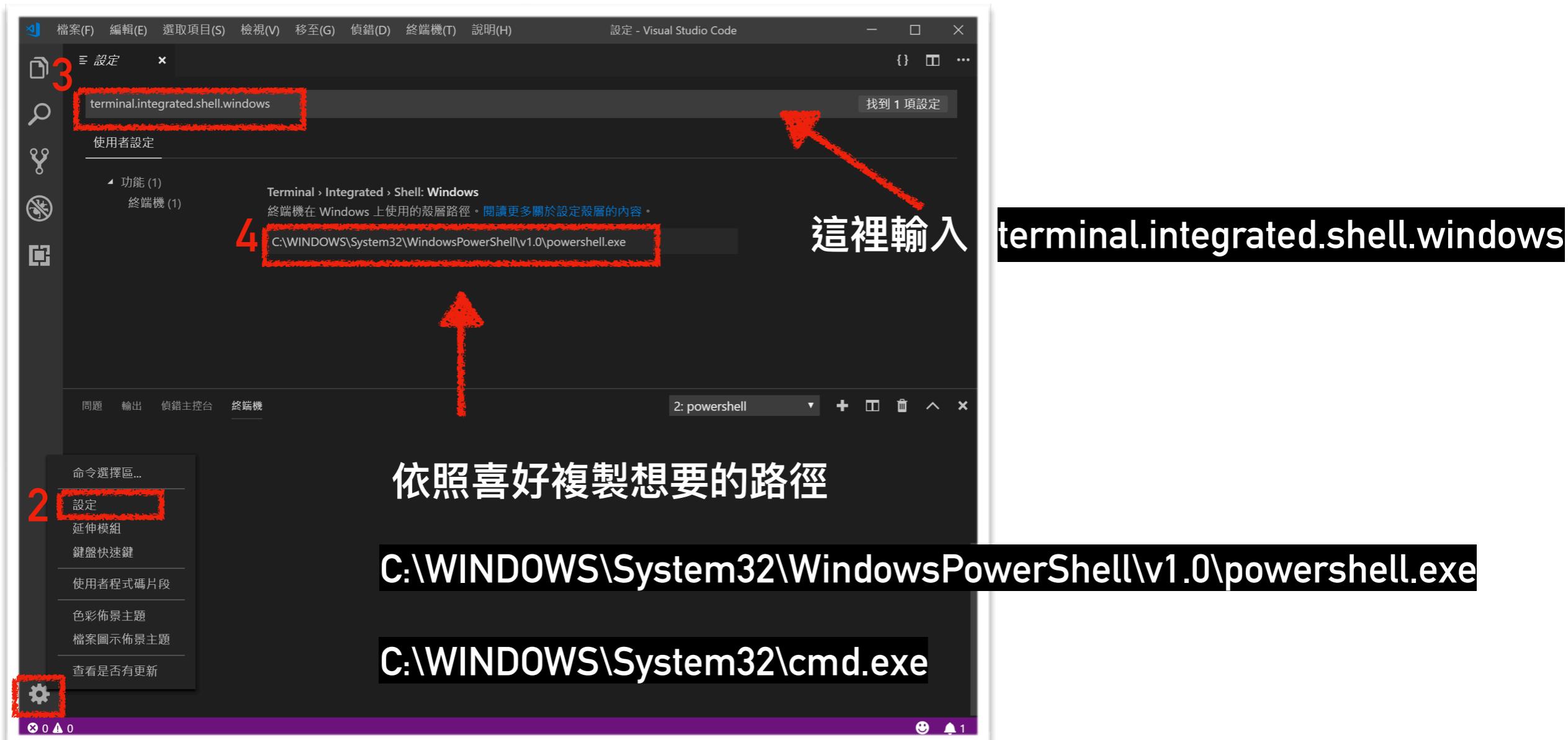
在macOS/Linux終端機上，按下Ctrl + Z是
凍結(暫停)當前的程式(SIGTSTP)

可以用**ps**指令(Process Status)查看剛剛暫
停的程式PID，並使用**kill**把它終止

順帶一提，按Ctrl + C則是**強制終止**

切換Code裡綁定的終端機

- 因為Windows 10已經將Powershell做為預設，如果想改成舊版的命令提示字元(CMD)請這樣做

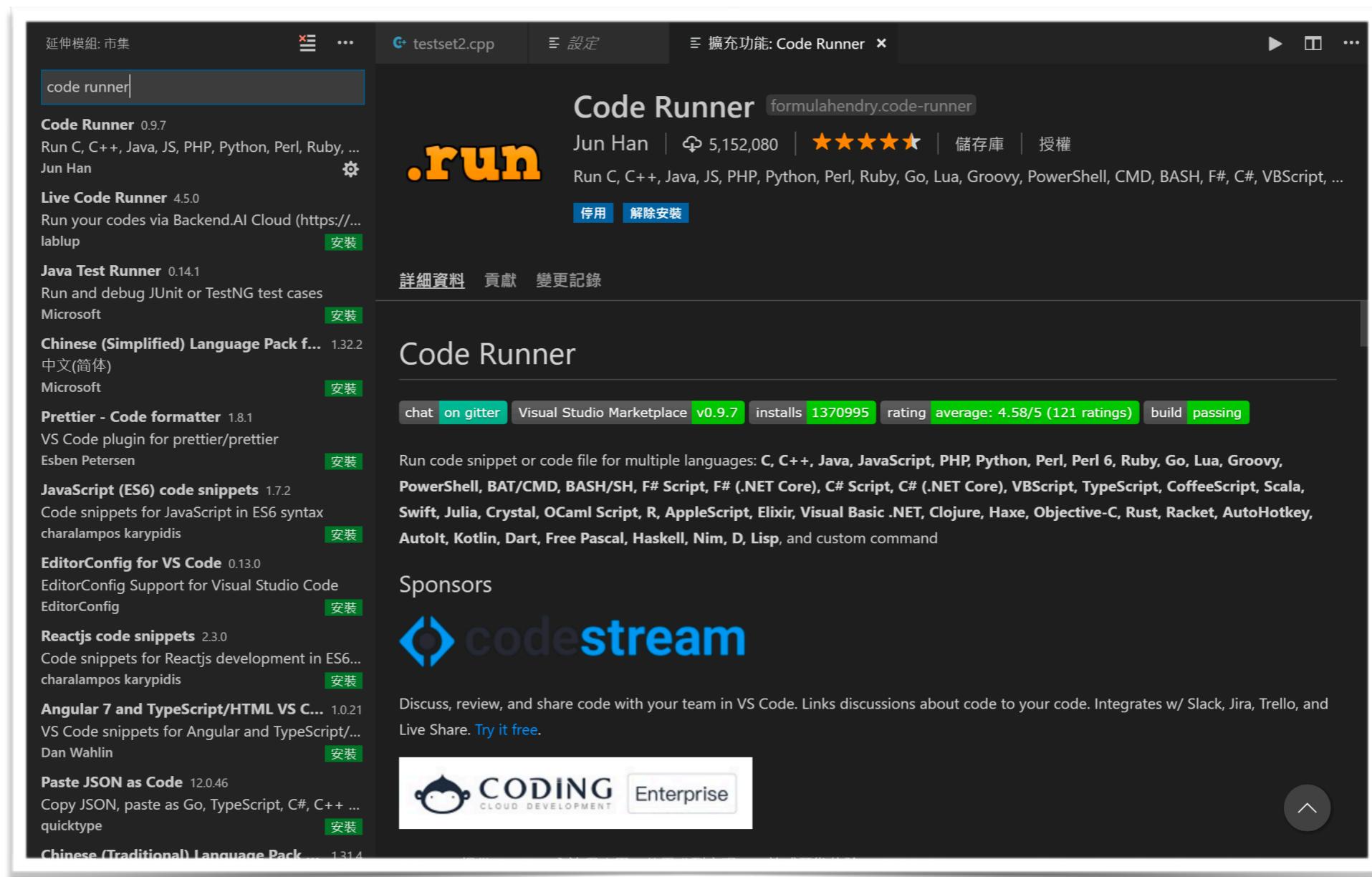


終端機常用指令

- `cd [資料夾名稱]` : 表示前往該資料夾
- `cd ..` : 表示回到上一層目錄
- `cd \` : 表示回到根目錄
- `dir` : 顯示該目錄底下的資料夾與檔案內容
- 在輸入資料夾或檔案名稱時可以先打開頭幾個字，再利用 Tab 鍵呼叫自動完成(補完檔名)

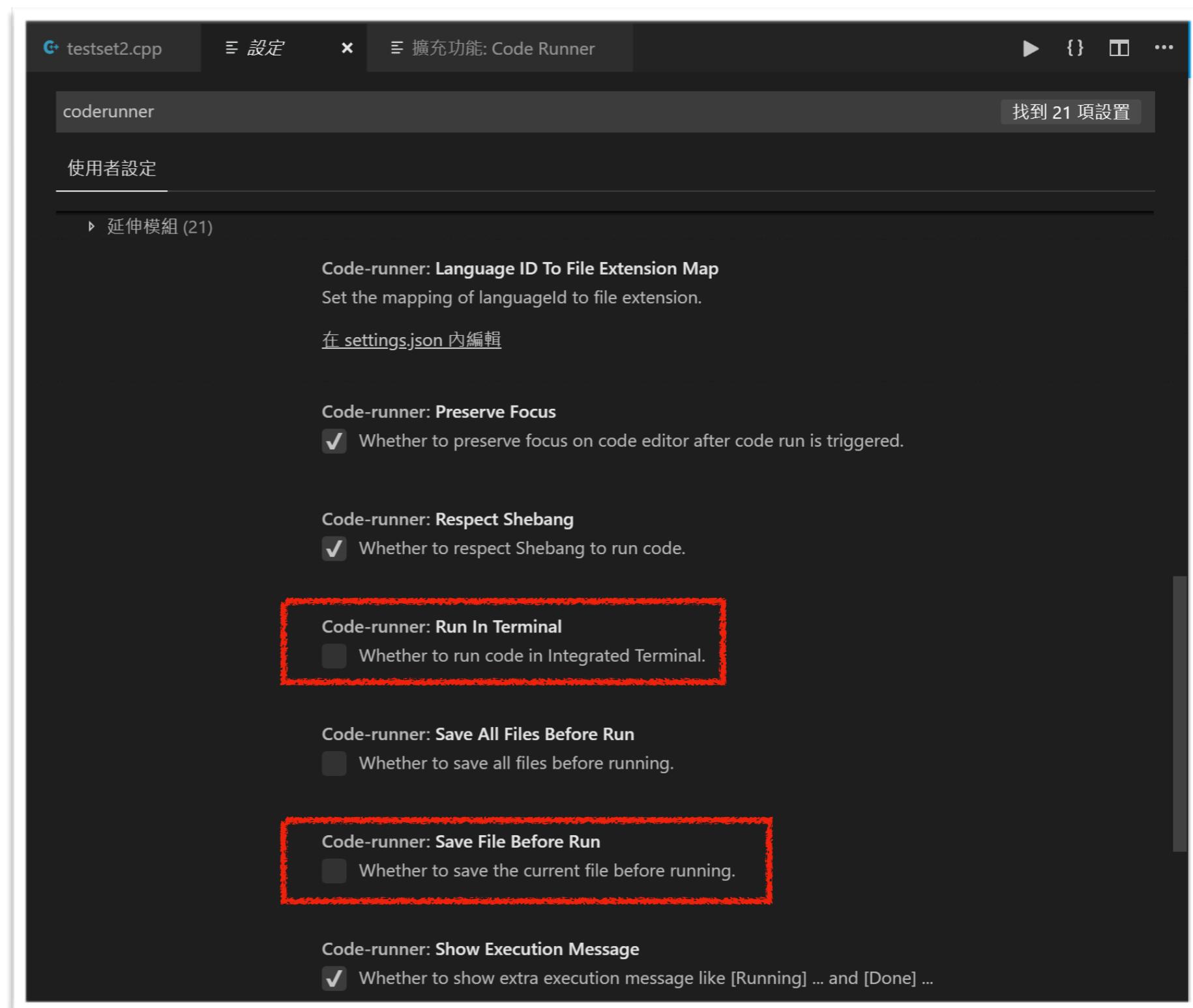
不想使用終端機的替代方案

● Code Runner (formulahendry.code-runner)



Code Runner 相關設定

將框起來兩個選項打勾



使用Code Runner

The screenshot shows a dark-themed Code Runner interface. In the top-left, there's a code editor window titled "test.cpp" containing the following C++ code:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int input;
6     scanf("%d",&input);
7     printf("Your Input: %d\n",input);
8 }
```

In the top-right, there's a toolbar with a play button icon (highlighted with a red circle), a refresh icon, a close icon, and a more options icon. Below the toolbar, the text "編譯並執行" (Compile and Run) is displayed, followed by "快速鍵：Ctrl + Alt + N".

At the bottom of the interface, there are tabs for "問題" (Issues), "輸出" (Output), "偵錯主控台" (Debugger Console), and "終端機" (Terminal). The "終端機" tab is active, showing the following terminal output:

```
Microsoft Windows [版本 10.0.17763.1]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

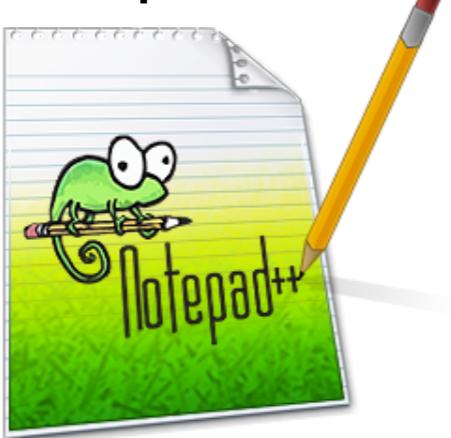
C:\Users\TEST>cd "c:\Users\TEST\Desktop\" && g++ test.cpp -o test && "c:\Users\TEST\Desktop\"test
4
Your Input: 4

c:\Users\TEST\Desktop>
```

Below the terminal output, the text "停止正在執行的程式：對著終端機按下Ctrl + C" (Stop running program: Press Ctrl + C on the terminal) is displayed.

其他編輯器的選擇

Notepad++



Sublime Text



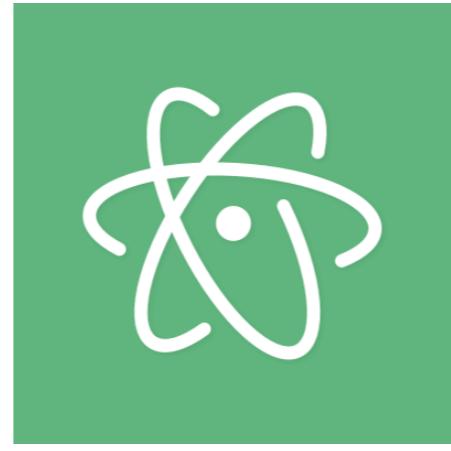
Brackets



Komodo Edit



Atom Editor



Notepad



鍵盤的浪漫



給我一把鍵盤，我給你整個世界。



Code::Blocks

安裝與中文化

APRIL 2020, Bonus Pack
想不到過了一年還有DLC可以看吧



打臉一年前的我

Code::Blocks是有社群維護的中文翻譯的！

Code::Blocks 手把手安裝

- 官方載點：<http://www.codeblocks.org/downloads>

The screenshot shows the official website for Code::Blocks. At the top, there is a logo consisting of four 3D cubes in red, green, yellow, and purple. To the right of the logo, the text "Code::Blocks" is displayed in a large, bold, black font. Below the logo, the tagline "Code::Blocks - The IDE with all the features" is visible. A navigation bar at the top includes links for "Home", "Features", "Downloads", "Forums", and "Wiki". On the left side, there is a sidebar with a "Main" heading and a list of links: Home, Features, Screenshots, Downloads (with sub-links for Binaries, Source, and SVN), Plugins, User manual, Licensing, and Donations. Below the sidebar, the word "Downloads" is highlighted in a large, bold, dark blue font. Underneath, a text block states: "There are different ways to download and install Code::Blocks on your computer:" followed by a bulleted list: "Download the binary release" (which is highlighted with a red border), "Download a nightly build" (with a note about other distributions following), and "Download the source code". At the bottom, there is a note about building from source and contributing patches.

Code::Blocks - The IDE with all the features

Home Features Downloads Forums Wiki

Main

- Home
- Features
- Screenshots
- Downloads
 - Binaries
 - Source
 - SVN
- Plugins
- User manual
- Licensing
- Donations

Downloads

There are different ways to download and install Code::Blocks on your computer:

- **Download the binary release**

This is the easy way for installing Code::Blocks. Download the setup file, run it on your computer

- **Download a nightly build:** There are also more recent so-called *nightly builds* available in the forums.
- Other distributions usually follow provided by the **community** (big "Thank you!" for that!).

- **Download the source code**

If you feel comfortable building applications from source, then this is the recommend way to download versions or, even better, create patches for bugs you may find and contributing them back to the community.

Quick links

手把手安裝 2



File	Date	Download from
codeblocks-20.03-setup.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03mingw-setup.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net

NOTE: The codeblocks-20.03-setup.exe file includes Code::Blocks with all plugins. The codeblocks-20.03-setup-nonadmin.exe file is provided for convenience to users that do not have administrator rights on their machine(s).

NOTE: The codeblocks-20.03mingw-setup.exe file includes *additionally* the GCC/G++/GFortran compiler and GDB debugger from **MinGW-W64 project** (version 8.1.0, 32/64 bit, SEH).

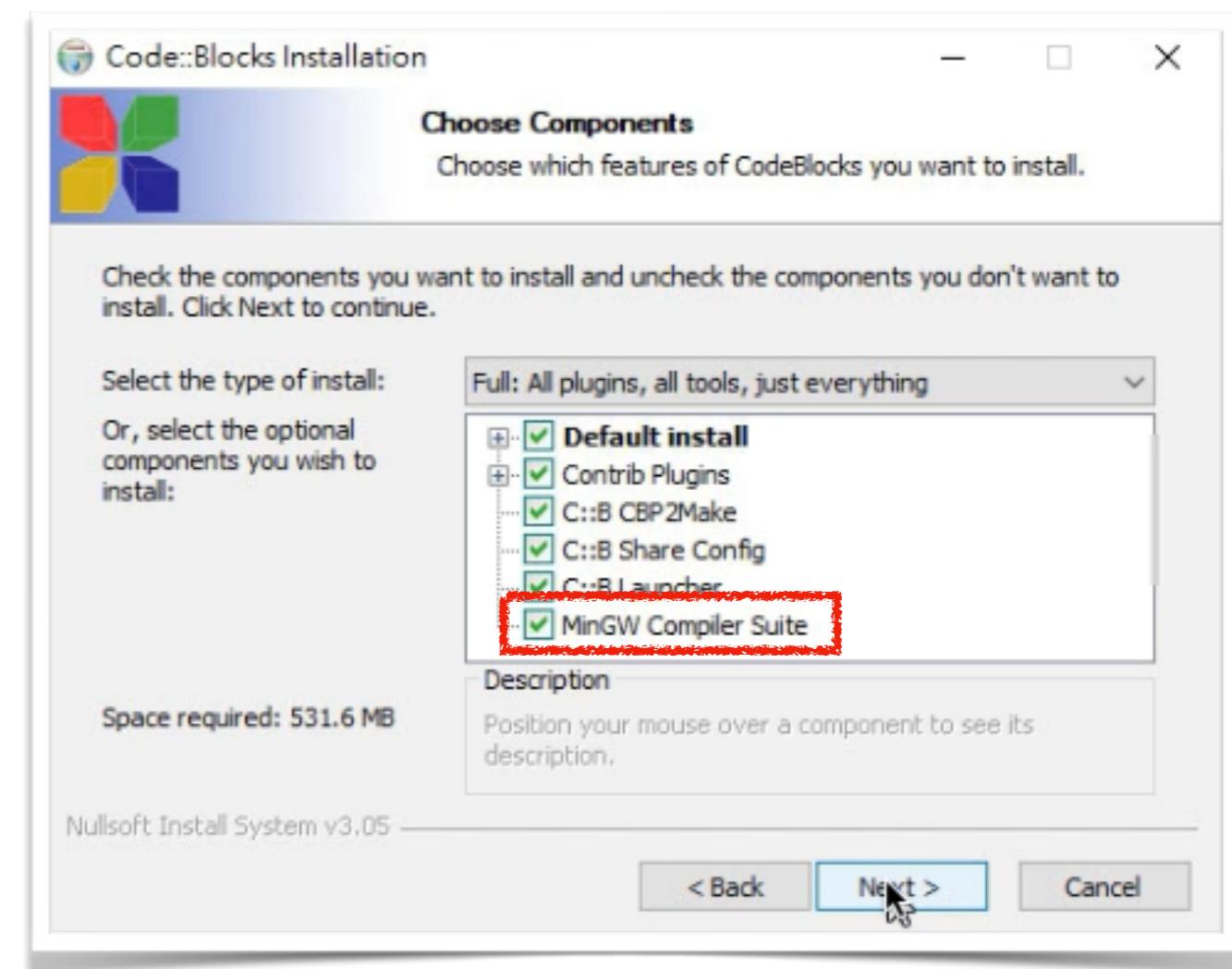
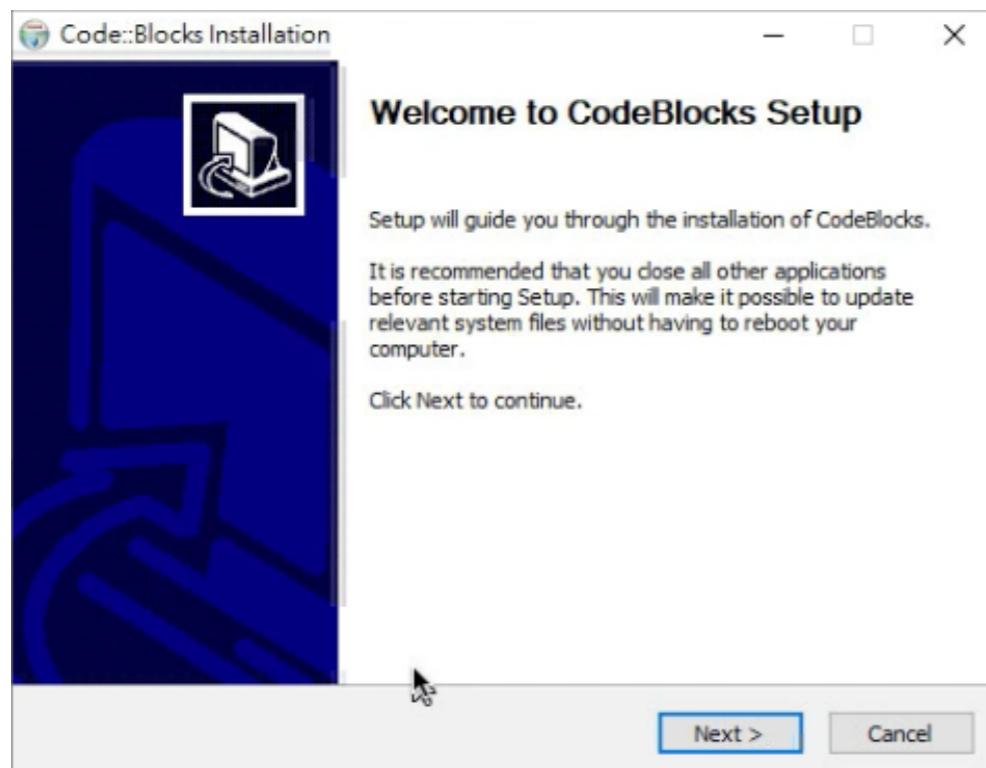
NOTE: The codeblocks-20.03(mingw)-nosetup.zip files are provided for convenience to users that are allergic against installers. However, it will not allow to select plugins / features to install (it includes everything) and not create any menu shortcuts. For the "installation" you are on your own.

If unsure, please use codeblocks-20.03mingw-setup.exe!

- 如果電腦已經有安裝Dev-C++且想要沿用隨附的編譯器
或是，你已經裝了MinGW
那麼請選**code-blocks-20.23-setup.exe**
- 反之選擇**code-blocks-20.23mingw-setup.exe**
讓Code::Blocks在安裝時一起安裝編譯器（內附MinGW-W64 v.8.1.0）

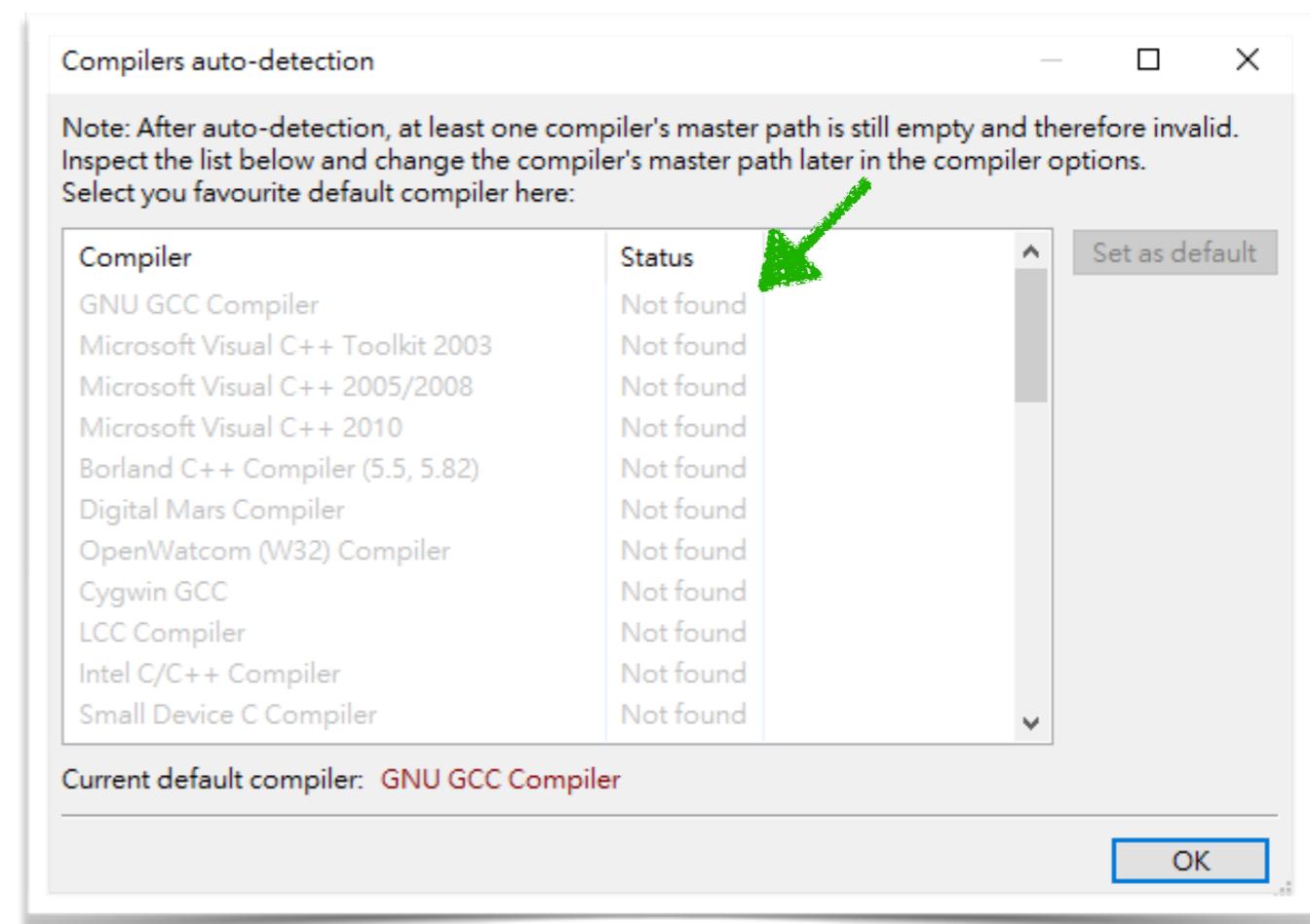
手把手安裝 3 開始安裝

- 不就是一路Next點到底嗎？(還真的是)
- 如果選擇有MinGW的版本，會多出一個MinGW Compiler Suite可以勾選，依自己喜好決定吧
- 決定完畢，選擇安裝路徑就開始安裝



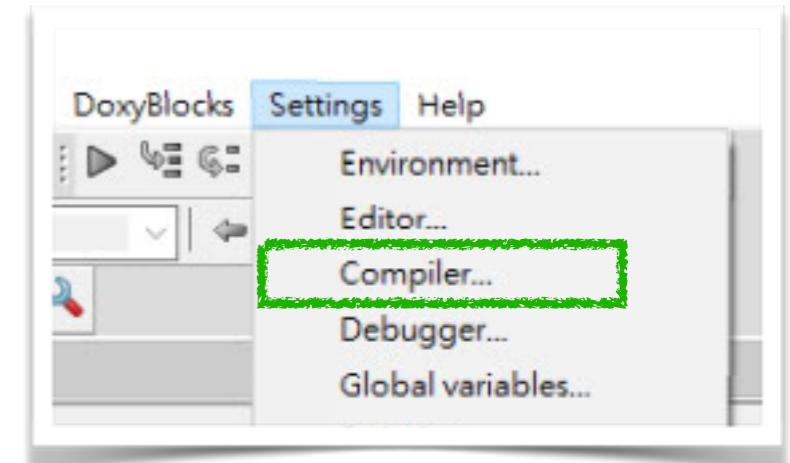
第一次啟動Code::Blocks

- 安裝後的第一次啟動，會主動偵測電腦安裝的編譯器，你可以在這個畫面確認Code::Blocks要使用哪一個編譯器做預設
- 正常來說應該會找到GNU GCC Compiler，但如果你不幸跟右圖一樣，請照著接下來的步驟完成編譯器設定。

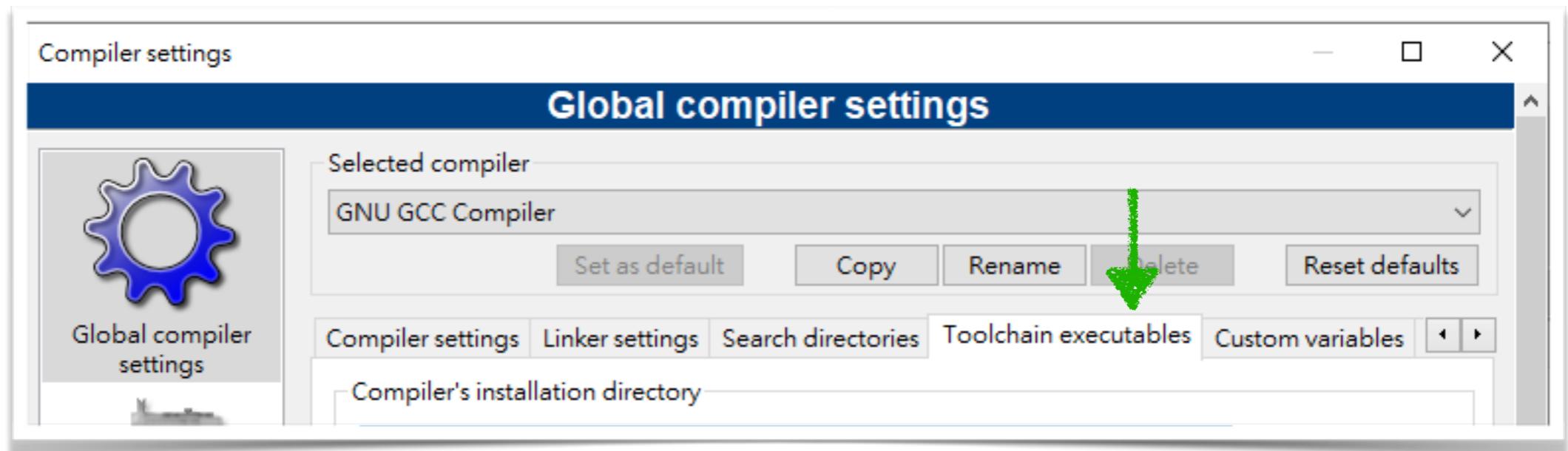


設定Code::Blocks使用的編譯器

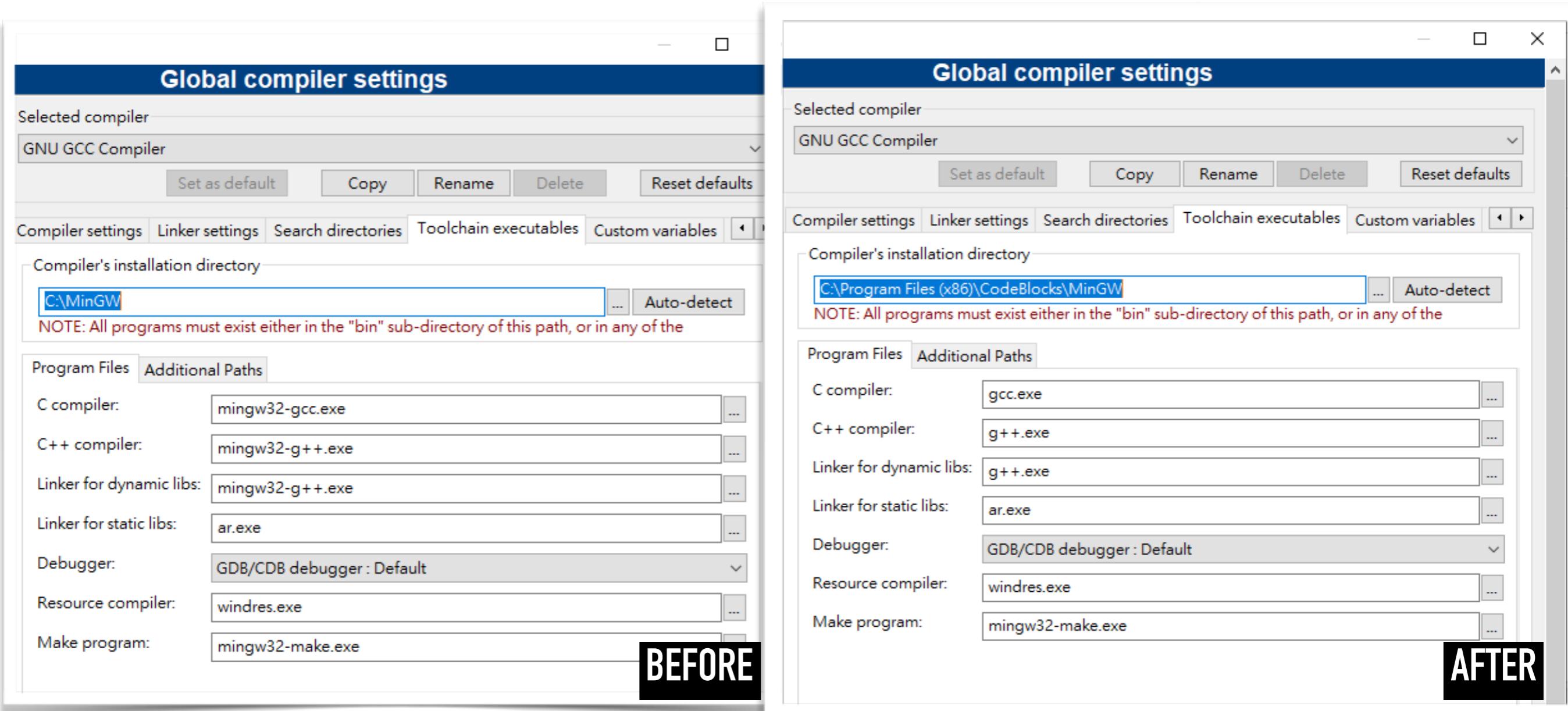
- 最上排工具列：Settings > Compiler



- 確認Selected compiler是選到「GNU GCC Compiler」
接著切換到"Toolchain executables"標籤



- Code::Blocks隨附的GNU Compiler，路徑會放在
C:\Program Files (x86)\CodeBlocks\MinGW
將Compiler Installation directory改成上面路徑
- 接著將底下的Program Files逐一指定為該路徑的對應檔案，完成後保存



完成這步之後，把C:\Program Files (x86)\CodeBlocks\MinGW\bin
也加入到系統環境變數吧

中文化

- https://translations.launchpad.net/codeblocks/trunk/+pots/codeblocks/zh_TW/+details

- 我真的錯惹，繁體中文可是有一群熱心的玩家在維護



Overview Code Bugs Blueprints **Translations** Answers

Details for Chinese (Traditional) translation

Series trunk » Template "codeblocks" » Chinese (Traditional) (zh_TW) » Details

Translation file details

Chinese (Traditional)

Creator:

sshanty

For: trunk

Translation policy: Open

Plural forms: 1

Plural expression:

0

Statistics

Messages: 2173

Translated: 2173 (100.0%)

Untranslated: 0 (0.0%)

Shared between Ubuntu and upstream: 2173 (100.0%)

Translated differently between Ubuntu and upstream: 0 (0.0%)

Only translated on this side: 0 (0.0%)

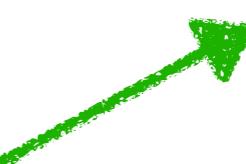
Latest contributor:

ycpu

Contributors to this translation

The following people have made some contribution to this specific translation:

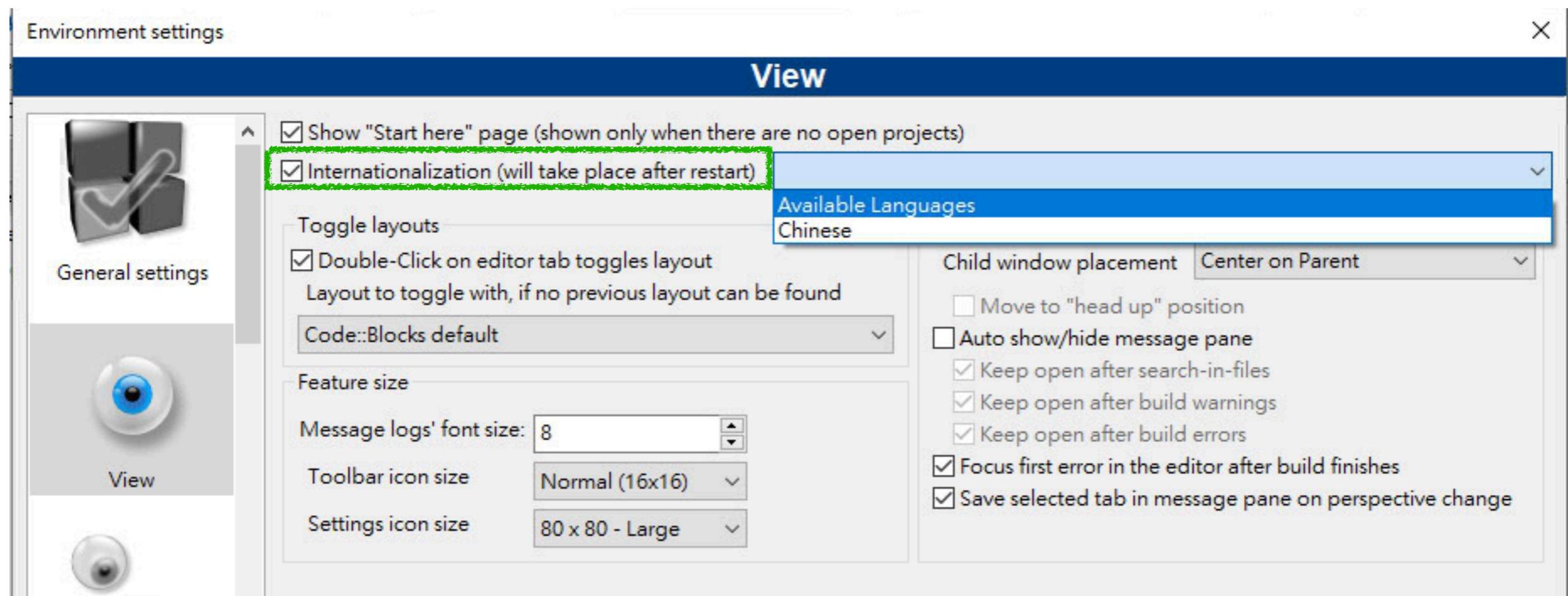
- Akihiro (filter)
- Alvin Tseng (filter)
- Anbang Li (filter)
- Bob (filter)
- CIN.GETLINE (filter)
- Chen Xuan-Ting (filter)
- Chiu Chuan-Hsien (filter)
- Dave Chang (filter)
- David day (filter)
- Firef0x (filter)
- HoneyMagic (filter)
- Hsiu-Ming Chang (filter)
- Junny (filter)
- Justin Lee (filter)
- LinJoy (filter)
- Maggie, Chan (filter)
- MienFong (filter)
- Ming Tsay (filter)
- PPC (filter)
- Patrick (filter)
- Reack Chen (filter)
- Timothy Lin (filter)
- Victor Lin (filter)
- aa516888 (filter)
- flucheng (filter)
- jacob wang (filter)
- kuwater (filter)
- newsoonly (filter)
- oiw (filter)
- qiyunluoxing (filter)
- seeing (filter)
- sshanty (filter)
- wade (filter)
- wei177 (filter)
- ycpu (filter)
- 李元航 (filter)
- 珍奇 (filter)
- 陳楷文 (filter)
- V字龍(Vdragon) (filter)

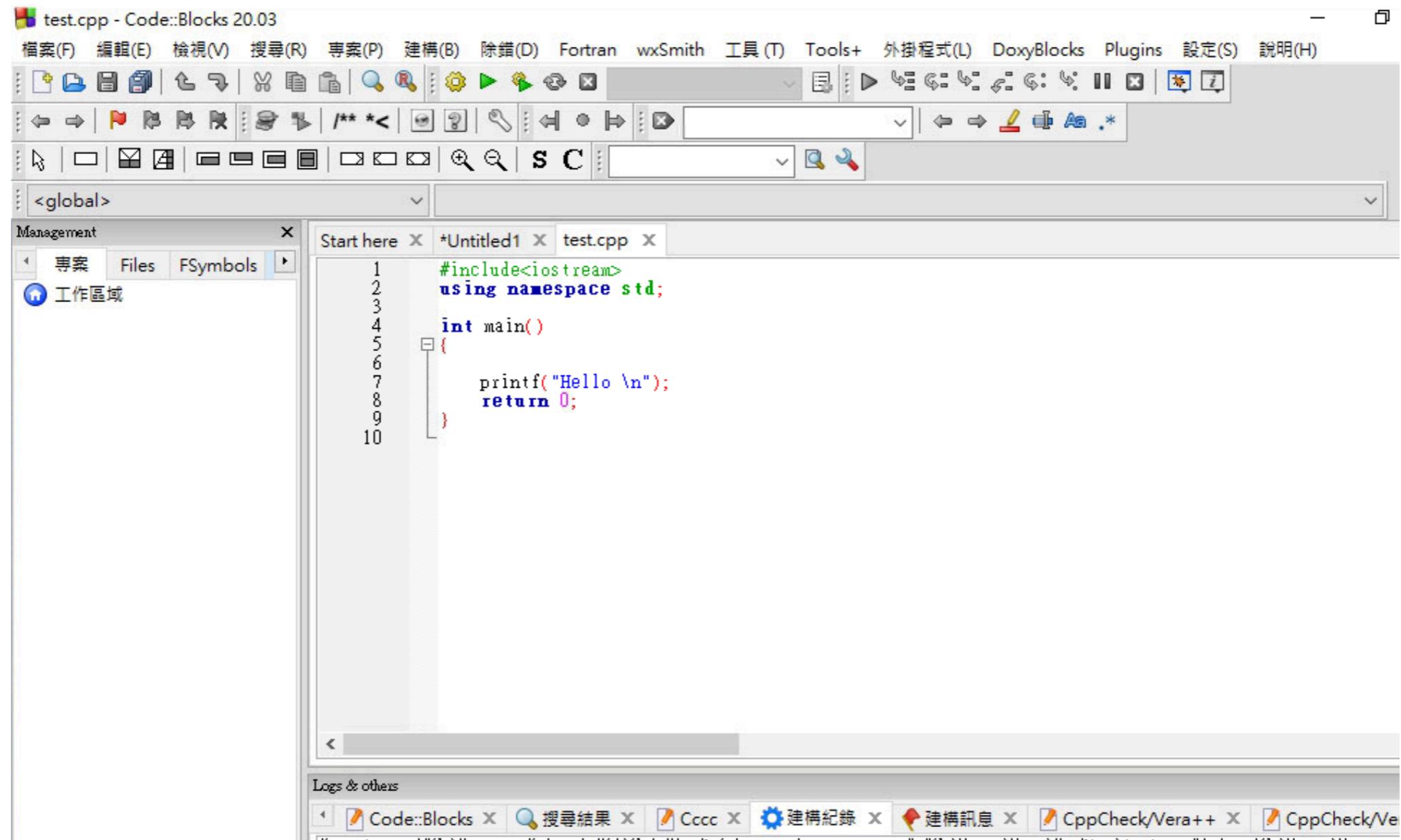


下載中文化檔案

- 我不懂為什麼想要下載一個語系檔還要特地註冊一個帳號...
(滴咕)
- 總之註冊帳號這件事交給助教我來就好
請到這個地方下載：<https://reurl.cc/5lGGbv>

- 把剛才下載的語系檔案放到
[C:\Program Files \(x86\)\CodeBlocks\share\CodeBlocks\locale\zh_TW](C:\Program Files (x86)\CodeBlocks\share\CodeBlocks\locale\zh_TW)
如果沒有locale資料夾就自己建立一個
- 工具列 Settings > Environment Settings
切換到View，勾選Internationalization，並選擇Chinese，確定關閉視窗，將Code::Blocks重開





Done!

將將～雖然有些部分沒有變成中文，但是也足夠了