

Chapter 0 – Start from 0

0是一切的起始，學程是語言也是要從零開始。

C語言簡介

特色

- 移植性高
- 兼具高低階語言的特性
- 可模組化
- 由上而下、由左而右執行
- 所有程式從main()開始
- 編譯過程分兩階段，程式碼->組語->機械碼。

環境配置

工欲善其事，必先利其器。

整合開發環境

- 通常要撰寫C語言程式我們必須要有 Editor + Compiler + Debugger
 - Ex: Vim + gcc + gdb
- IDE = Editor + Compiler + Debugger
- 推薦使用：
 - Orwell Dev-C++
 - Code::Blocks

下載Dev-C++

- Reduced overdraw in the class browser when switching between or saving files.
- Added the ability to syntax check the currently visible file only, even if it belongs to a project.
- Added icons for Syntax Check, Syntax Check Current File and Clean.
- Fixed an out of bounds problem in the parser.
- Dev-C++ now restores windows to their Aero Snap position.

安裝版

Download

- The **setup** which includes MinGW32 4.8.1 can be downloaded [here](#) (63MB).
- The **setup** which includes TDM-GCC x64 4.8.1 can be downloaded [here](#) (47MB).
- The **setup** which does not include a compiler can be downloaded [here](#) (2MB).

- The **portable** version which includes MinGW32 4.8.1 can be downloaded [here](#) (36MB).
- The **portable** version which includes TDM-GCC x64 4.8.1 can be downloaded [here](#) (34MB).
- The **portable** version which does not include a compiler can be downloaded [here](#) (2MB).

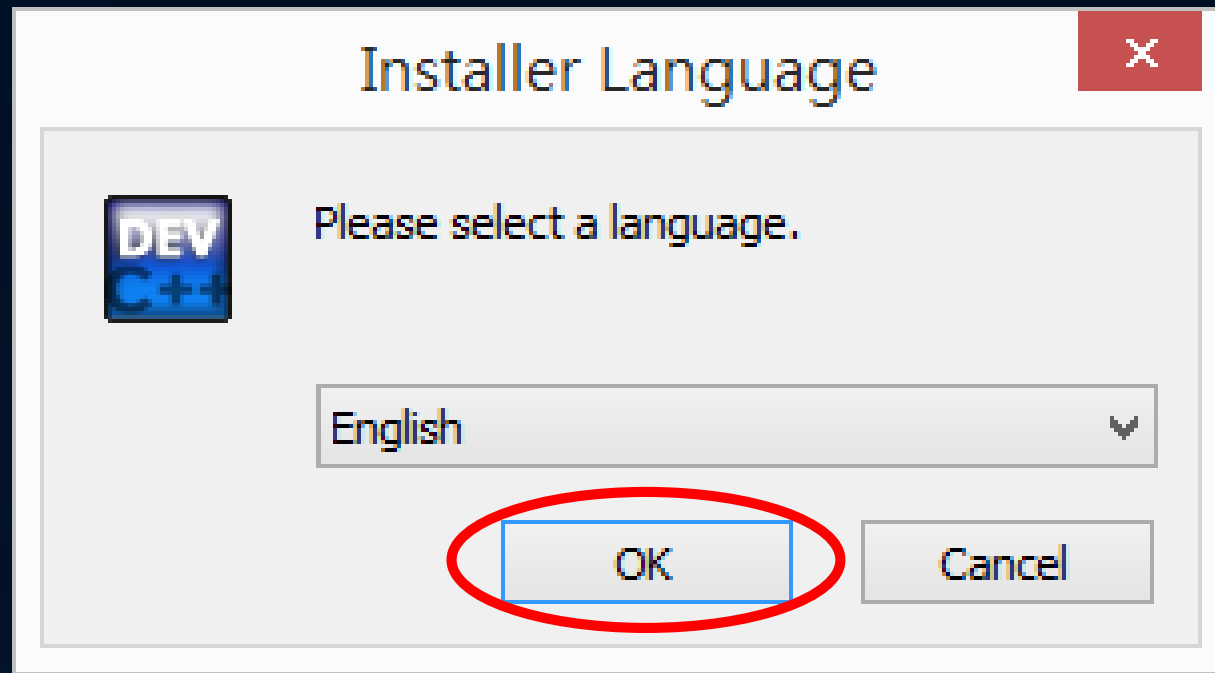
- The latest tested compilers can be downloaded [here](#).
- Lastly, the source code can be found [here](#) (1MB).

Windows XP、Windows Vista 32bit、Windows 7 32bit、Windows 8 32bit、Windows 8.1 32bit

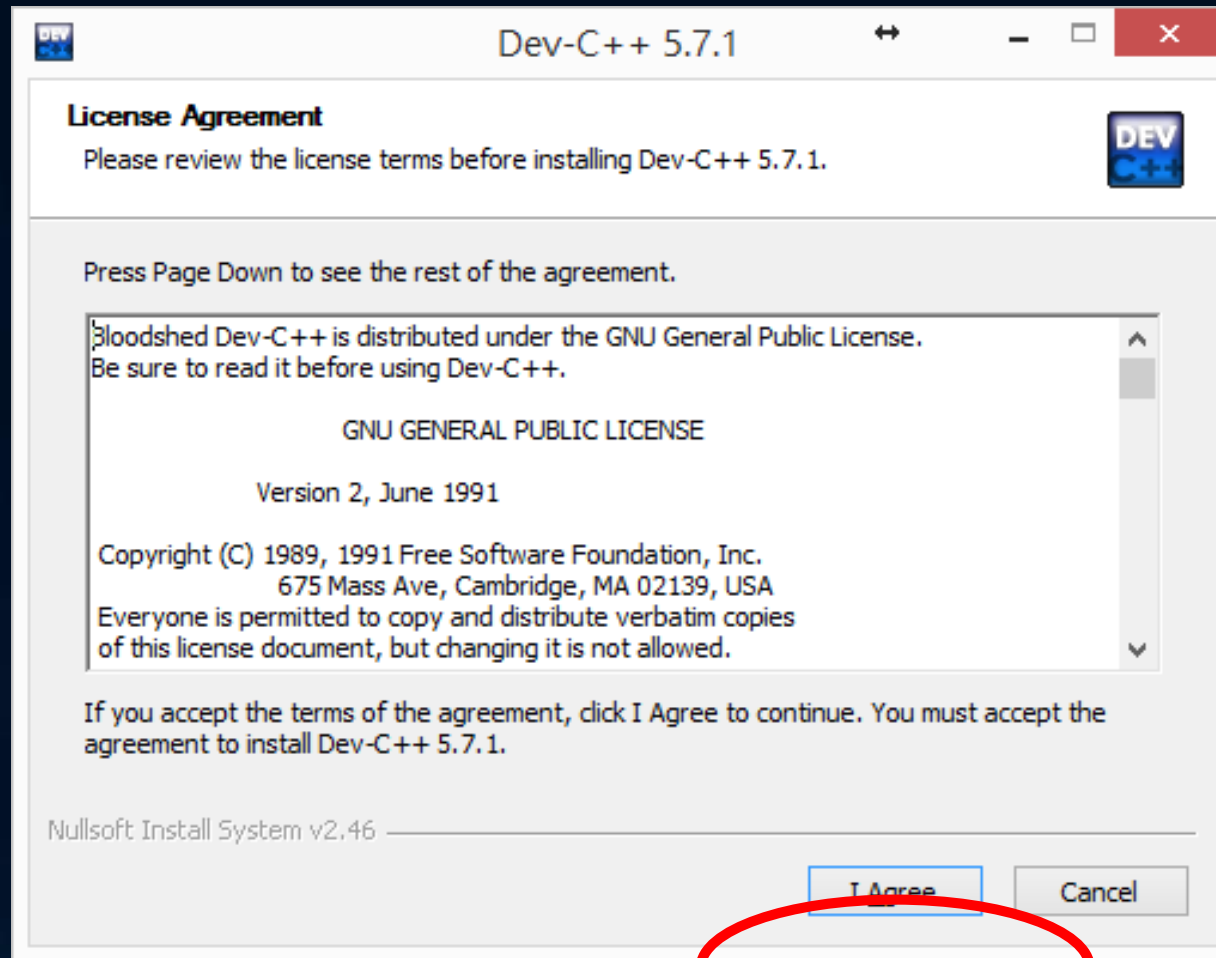
Windows Vista 64bit、Windows 7 64bit、Windows 8 64bit、Windows 8.1 64bit

免安裝版

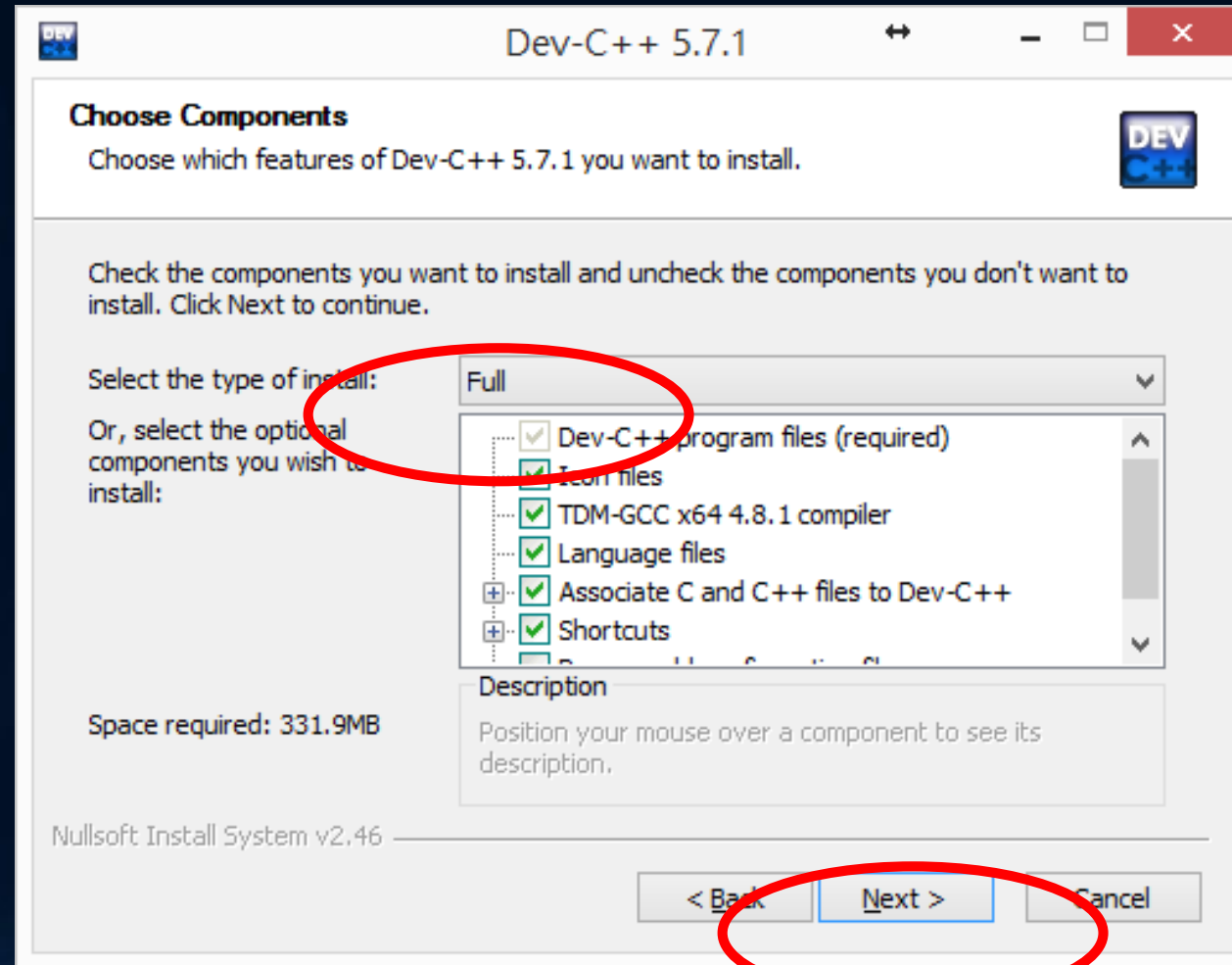
安裝Dev-C++



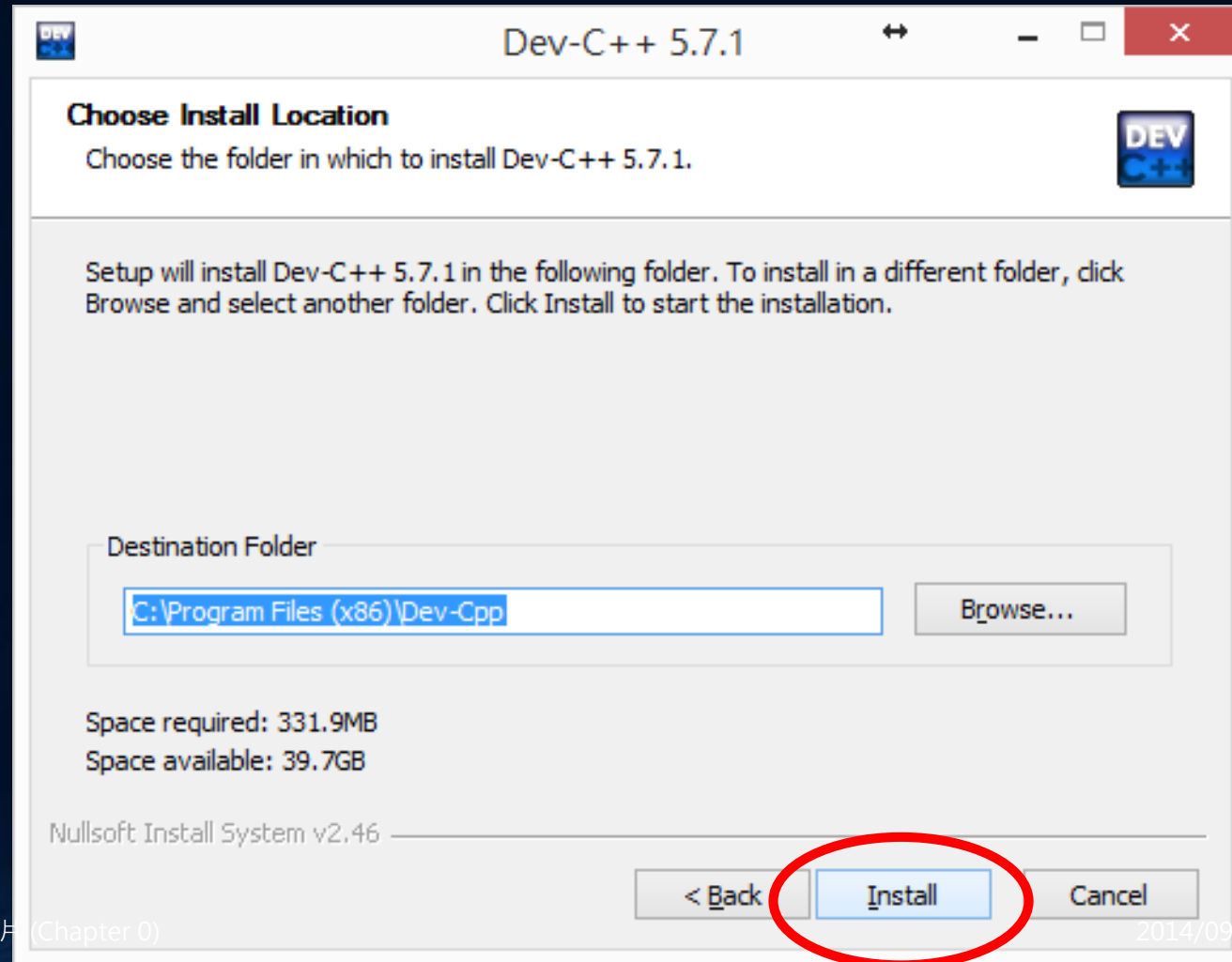
安裝Dev-C++



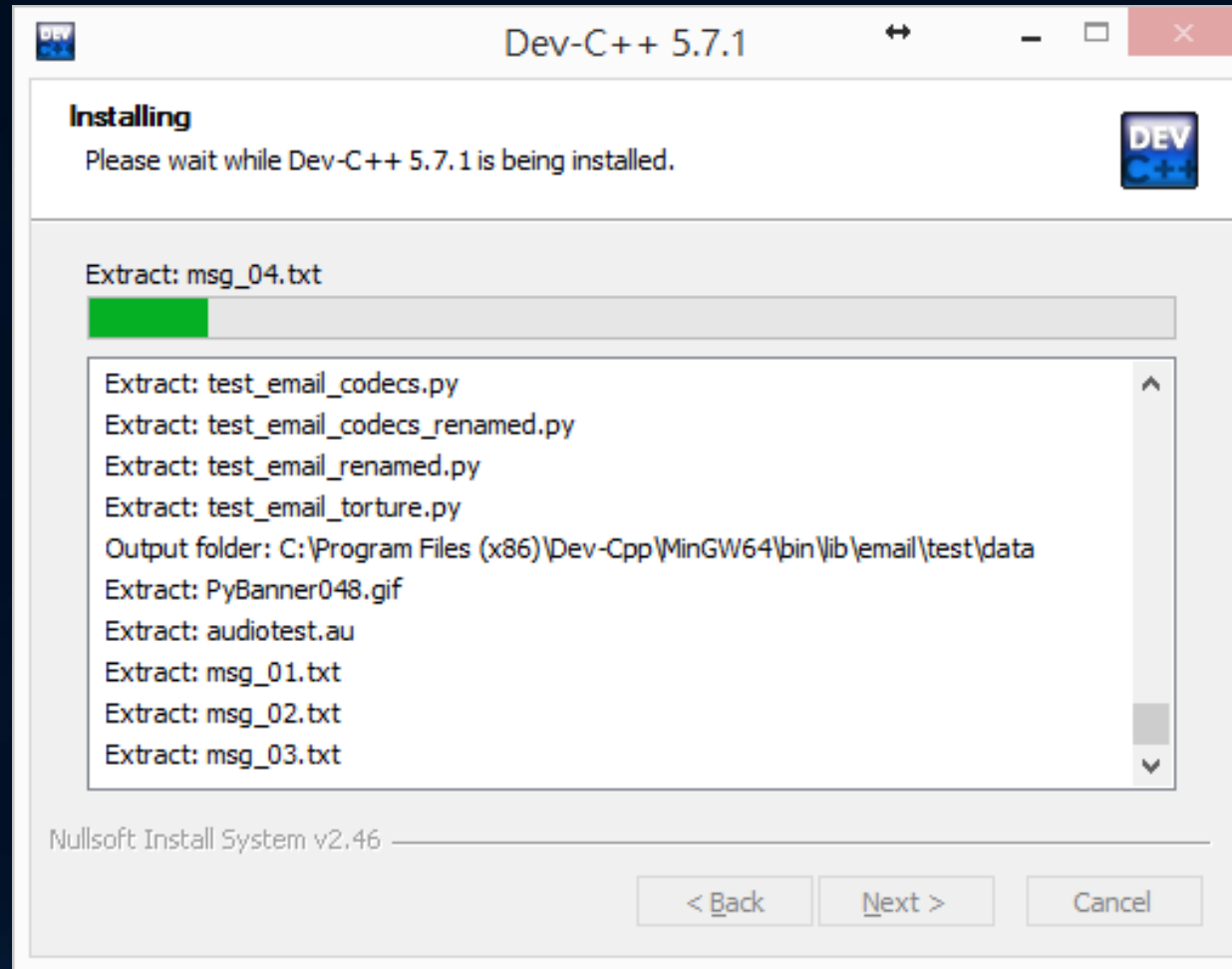
安裝Dev-C++



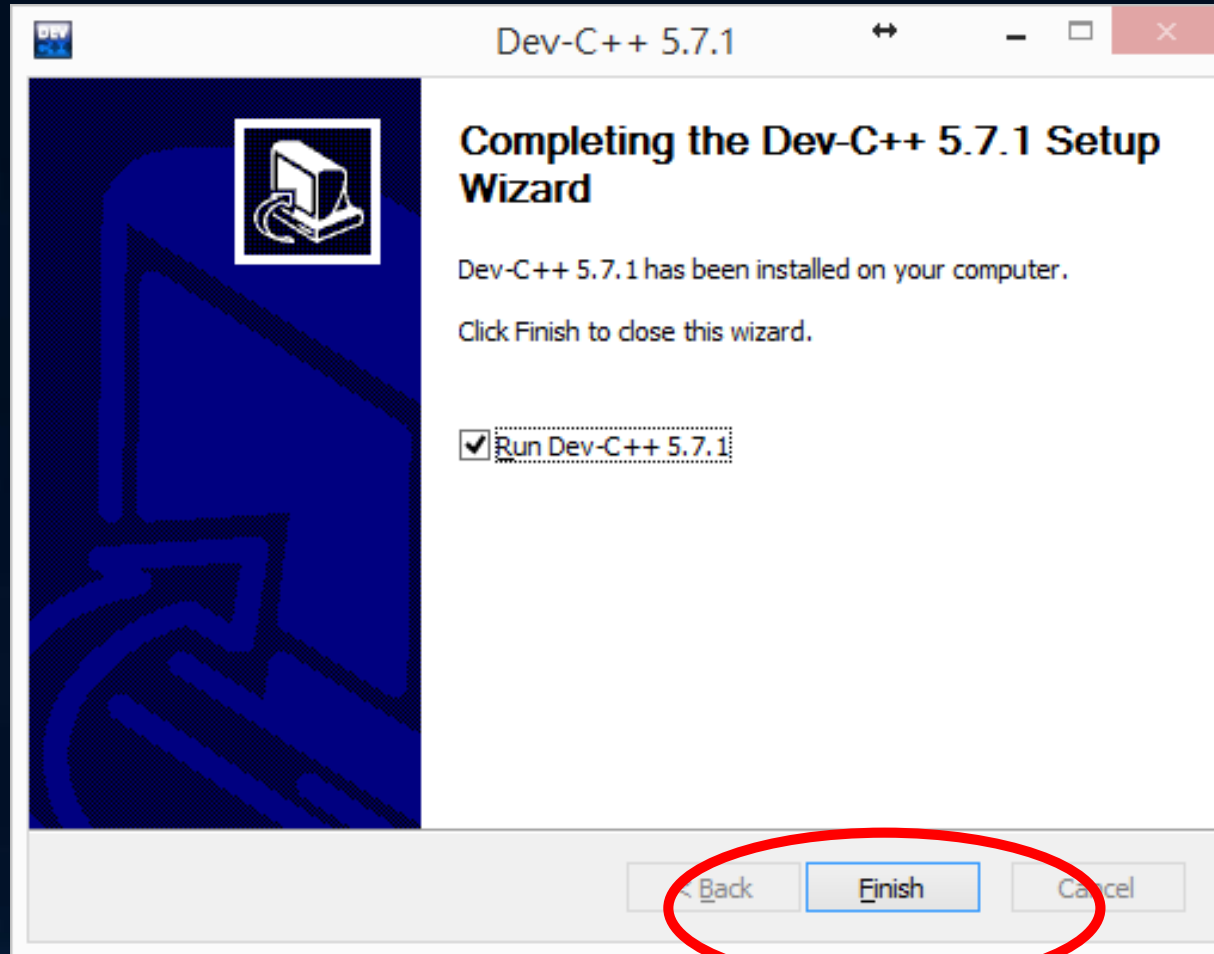
安裝Dev-C++



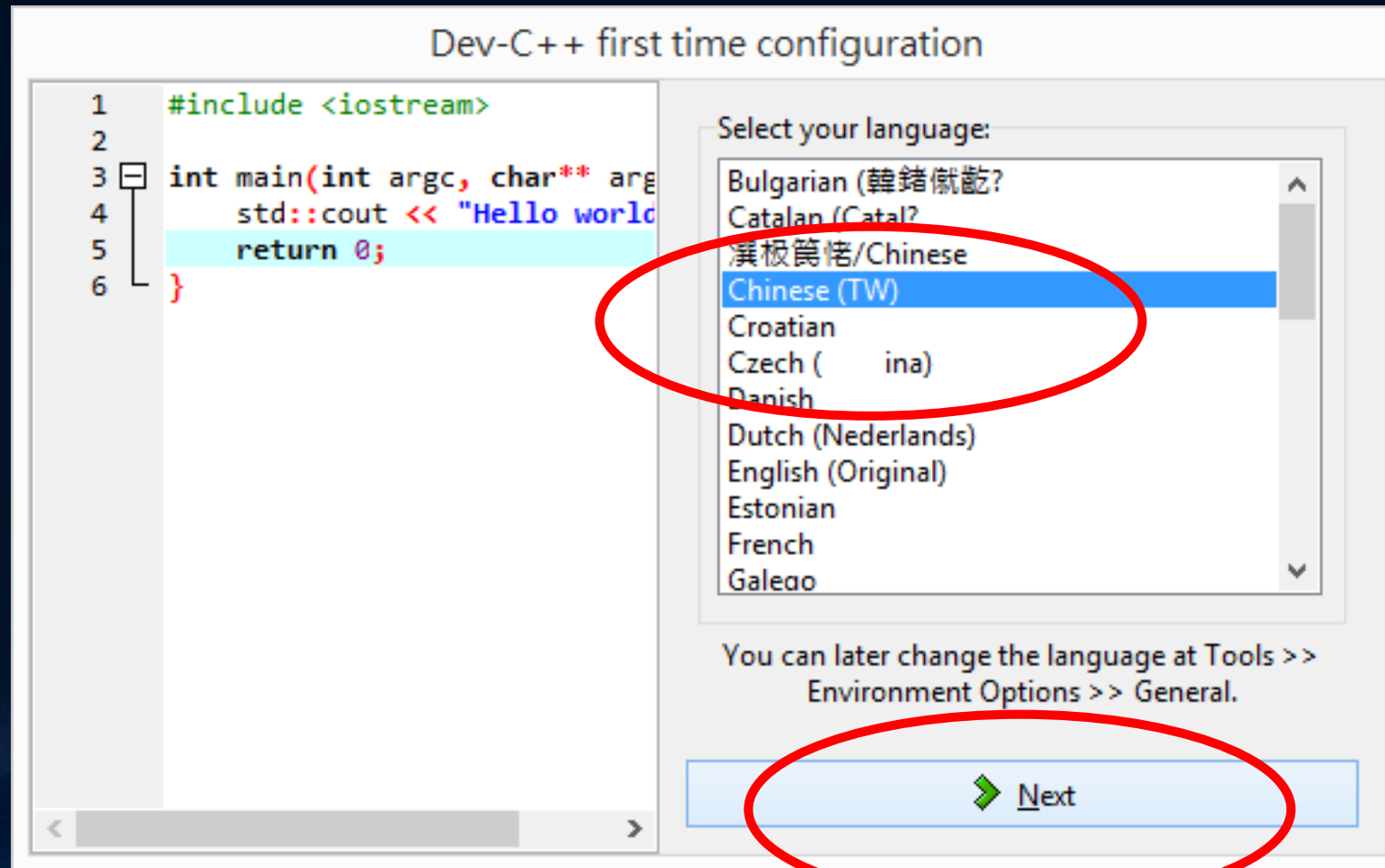
安裝Dev-C++



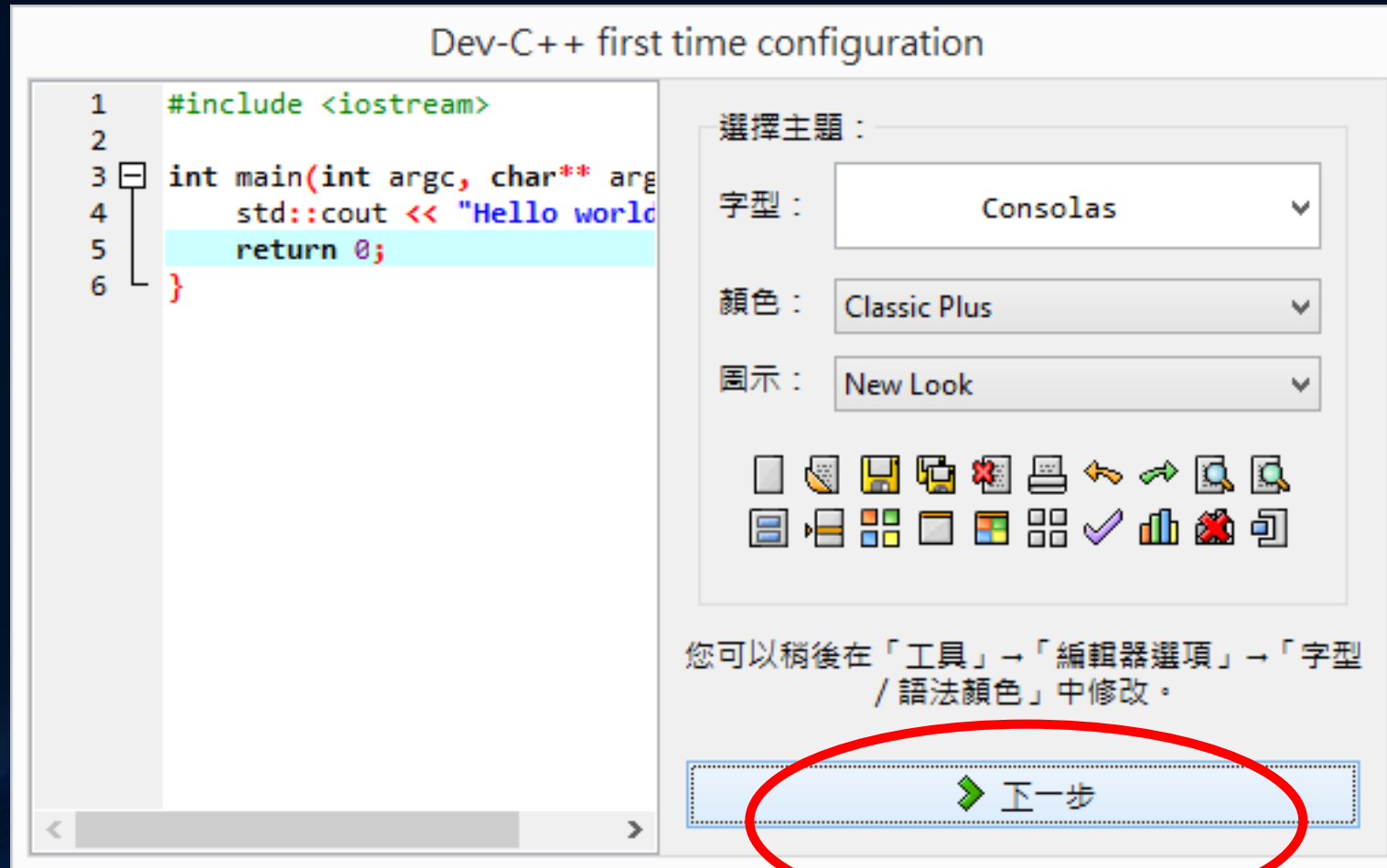
安裝Dev-C++



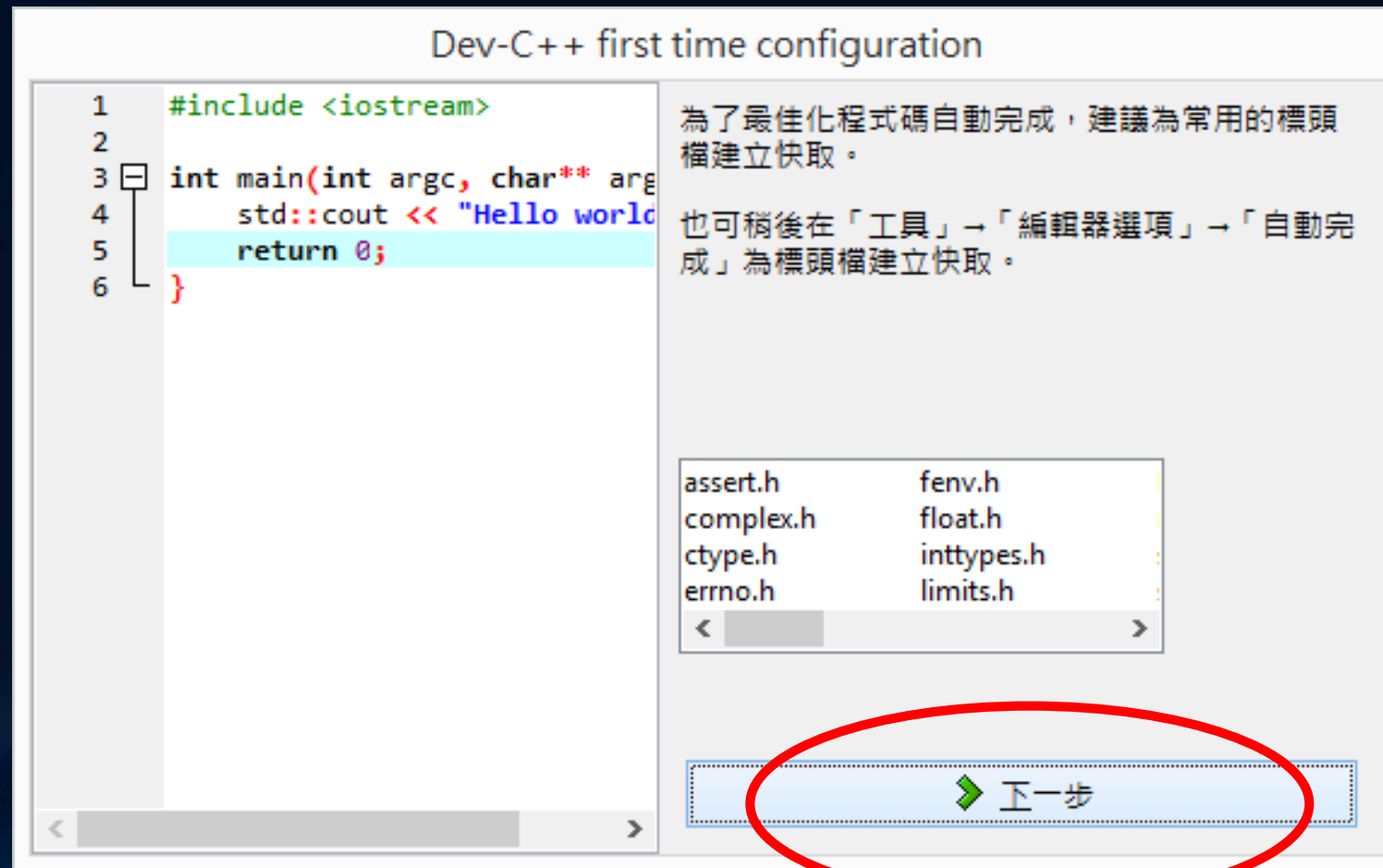
安裝Dev-C++



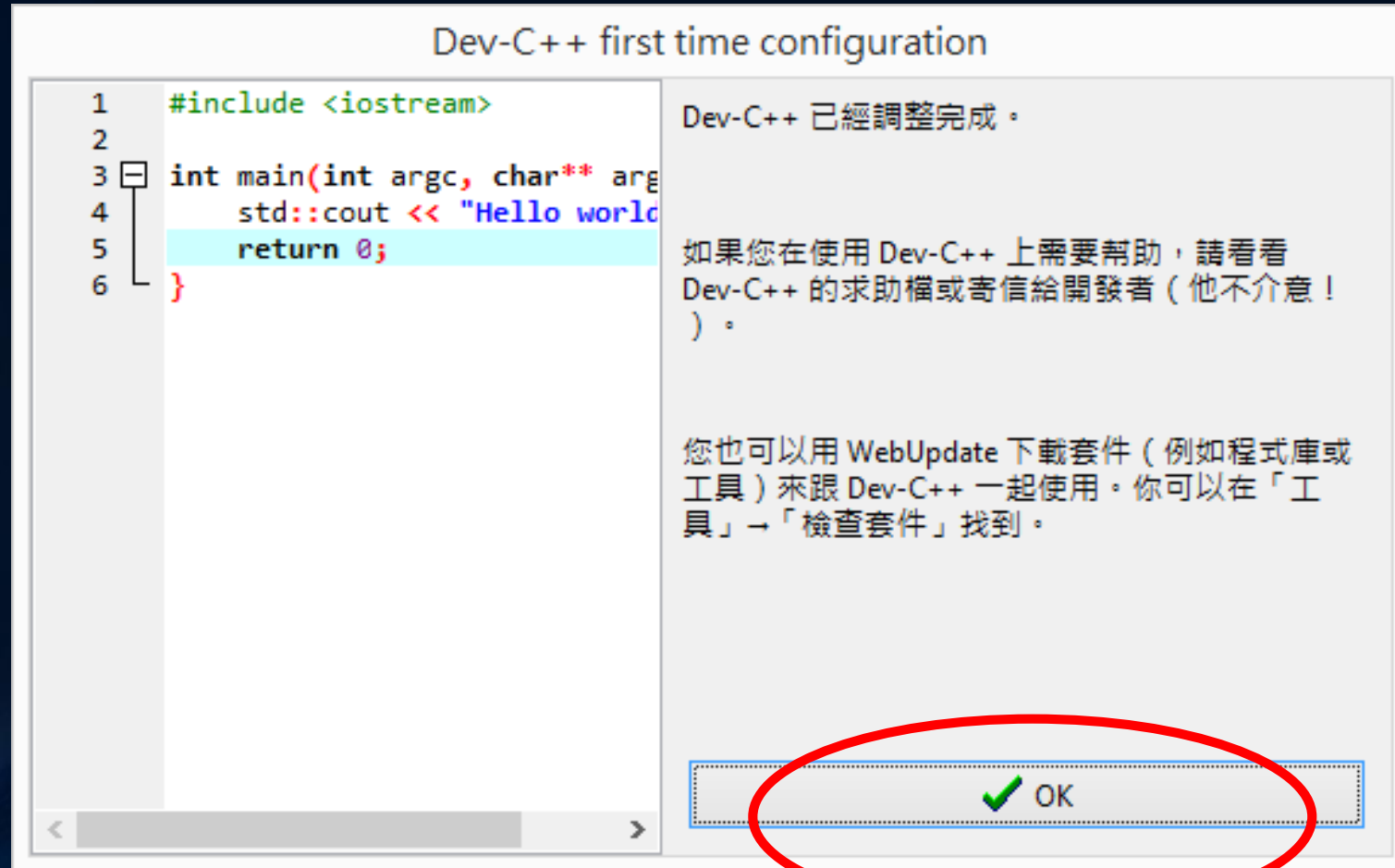
安裝Dev-C++



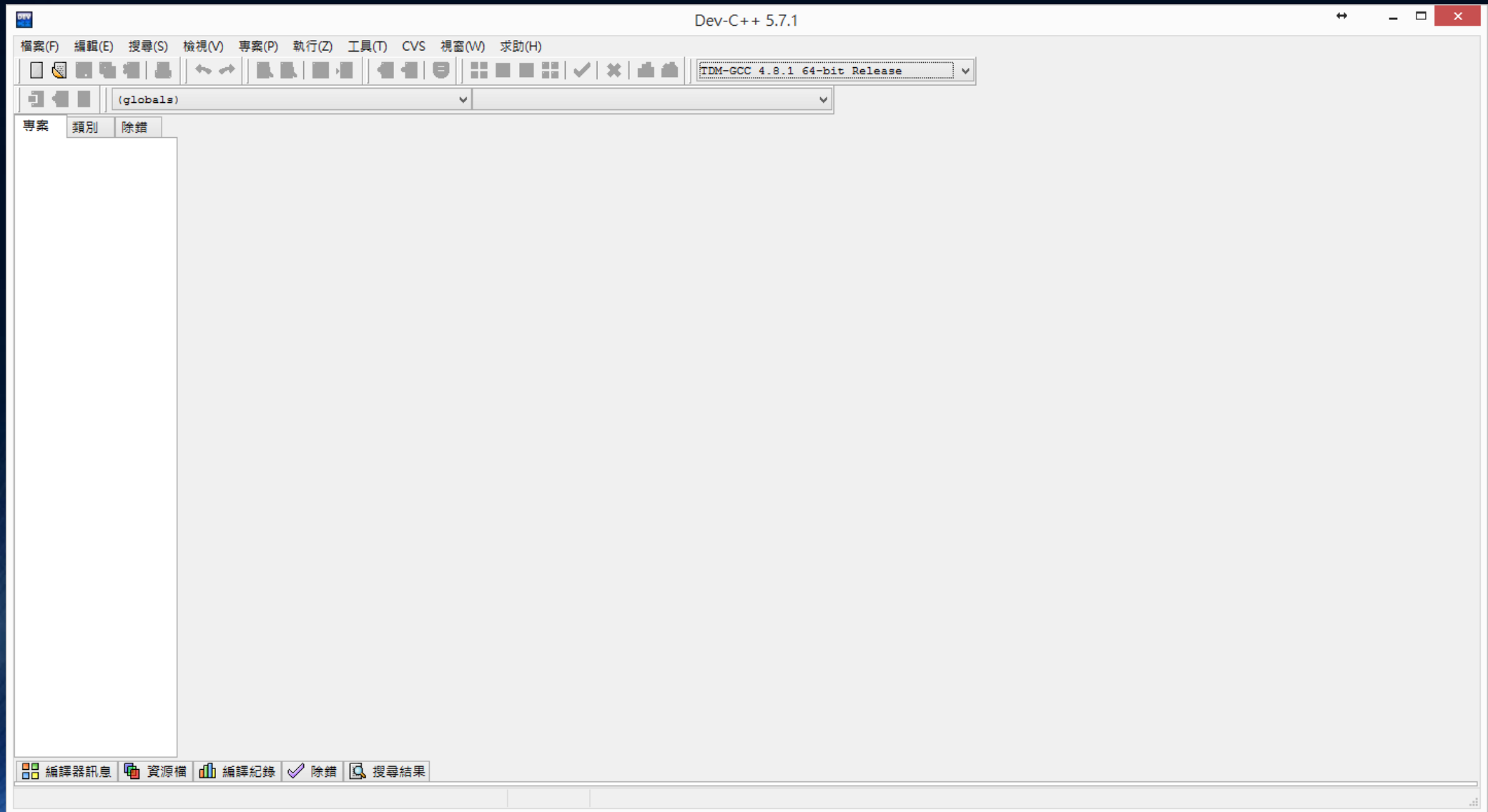
安裝Dev-C++



安裝Dev-C++



Dev-C++ 安裝完成



開始寫C

HELLO WORLD!

架構

- C語言基本上可以分兩部分來看，前置區與函式區。

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
```

前置區

```
int main(void){
    int A, B, A_bin = 0, B_bin = 0, ans = 0, i;
    printf("Please enter two integer: ");
    do{
        scanf("%d %d", &A, &B);
    }while((A>100||A<1)||B>7||B<1));
    printf("\n=====\nOrigin Input(Dec): %d, %d\n\n", A, B);
    i = 0;
    while(A!=0){
        A_bin += A%2*pow(10.0, (double) i);
        A/=2;
        i++;
    }
    i = 0;
    while(B!=0){
        B_bin += B%2*pow(10.0, (double) i);
        B/=2;
        i++;
    }
}
```

函式區

架構

- 舉凡前面帶有 # 皆屬於前置區域。
 - `#include<stdio.h>`
 - `#define PI 3.14`
- 前置區域在編譯時會優先處理。
- 函式區的規則我們日後再講。

標頭檔

- 剛剛我們提到了前置區的 `#include<stdio.h>`，這就是標頭檔。
- 標頭檔是讓編譯器了解這個函式的**作用**。
 - 換言之，標頭檔內宣告了函式。
- `stdio.h`
 - `stdio` = Standard Input and Output Library
- `stdlib.h`
 - `stdlib` = Standard General Utilities Library

標準式

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

```
int main(void){
```

```
    return 0;
```

```
}
```

標準式

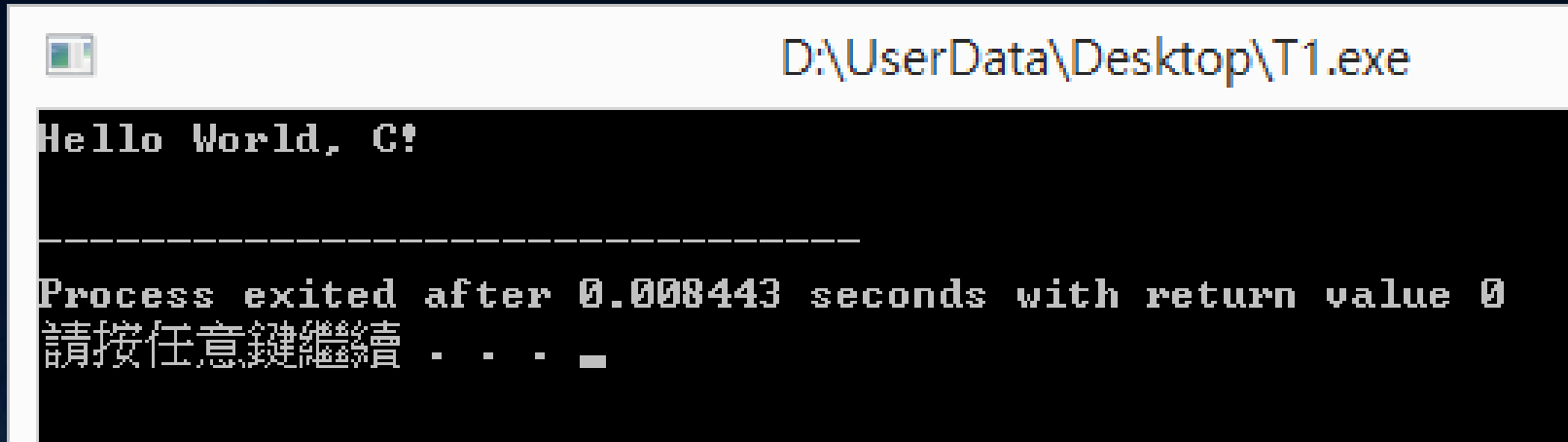
- `#include<stdio.h>` 代表引入 `stdio` 這個標頭檔。
 - `stdio` = Standard I/O
- `return 0;` 是告訴系統程式順利無錯誤的結束了。
- `main()`是C語言撰寫的程式中第一個被執行到的函式。
- 由於最後`return 0;`因此`main()`前面只可以是 `int`。

撰寫C語言程式的幾個建議

1. 背好標準式，所有的C語言程式一定會由那標準式去擴充。
2. 每一行指令最後都要分號。
3. 寫程式碼時要有縮排的習慣。
4. 除非你知道你在做甚麼，否則別簡寫。
5. 記得加上大括號！
6. - 10. **記得加上大括號！！！！**

小試身手

- 請寫出一個程式執行結果如下：



```
D:\UserData\Desktop\T1.exe
Hello World, C!
-----
Process exited after 0.008443 seconds with return value 0
請按任意鍵繼續 . . . _
```

- 提示：`printf("Hello World, C!\n");`
可以印出Hello World, C! 這句話。

解答

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

// 引入標頭檔 "stdio.h"

```
int main(){
```

// 讓程式知道從這一行開始執行

```
printf("Hello World, C!\n");
```

// 輸出 Hello World, C!

```
return 0;
```

// 告訴電腦完美無誤的結束了

```
}
```