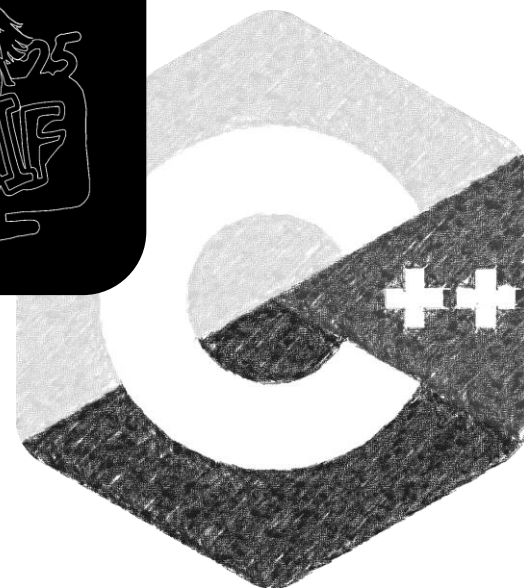




C++初探

0923資訊社





什麼是程式語言？

程式語言（programming language），是用來定義電腦程式的形式語言。它是一種被標準化的交流技巧，用來向電腦發出指令，一種能夠讓程式設計師準確地定義電腦所需要使用資料的電腦語言，並精確地定義在不同情況下所應當採取的行動。

Source: Wikipedia – 程式語言



什麼是程式語言？

簡單來說，程式語言是：

- 一種溝通的媒介
- 與電腦溝通的橋樑
- 人與電腦溝通的語言



程式語言分類

人類社會的語言會因為所在的地區、生長背景的不同而有所差異。

程式語言也是如此，會因為處理器的差異而有所不同。

主要分為高階語言與低階語言。



程式語言分類

低階語言 (Low-Level Language)

- 機器語言 (Machine Language)
使用二進位來表示，電腦可直接執行

```
10001000 110000 01100100
10001000 110001 01100100
10001001 110000 110001
```

- 組合語言 (Assembly Language)
使用接近人類語言的字組，如：ADD、SUB，需經過組譯器 (Assembler) 才可執行

```
MOV AX,100
MOV BX,10
ADD AX,BX
```



程式語言分類

高階語言 (High-Level Language)

- 可讀性較好
- 需經過直譯器或編譯器翻譯成機器語言才可執行
- E.g. C, C++, Java, Python

```
// C++範例
#include <iostream>

int main(){
    std::cout << "Hello, World!\n";
    return 0;
}
```



為什麼要學C++？

C++比較難，而且比較老，其語法經常被其他語言所參考，而相較語語法相較簡單的Python之後要自學比較容易，在其他人的幫助下學C++會比較好，之後也可以進一步去學其他語言。



C++簡介

20世紀80年代

丹麥Bjarne Stroustrup博士

貝爾實驗室

意旨開發更好用的C

- 在保留傳統C的優點時，加上了物件導向的設計理念



Julia Kryuchkova - Own work CC BY-SA 2.5



編譯器 & 直譯器

用簡單的比喻來形容：

直譯器就像同步口譯員一句句翻譯，而編譯器就像翻譯員直接將整篇文本翻譯

直譯器

直譯器

就像

同步口譯員

一句句

翻譯

例如：Python、JavaScript

編譯器

而編譯器就像翻譯員直接將整篇文本翻譯

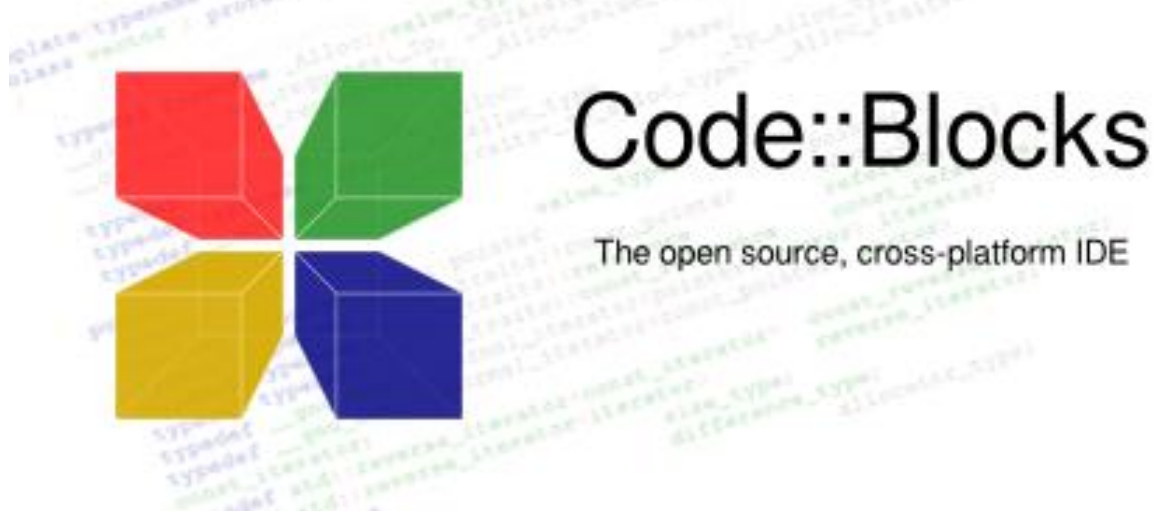
例如：C、C++



Code::Blocks 介紹

Code::Blocks是一個免費、開源、跨平台的整合式開發環境，使用C++開發，並且使用wxWidgets做為GUI函式庫。

Code::Blocks使用了外掛程式架構，其功能可以使用外掛程式自由地擴充。目前，Code::Blocks主要針對開發C／C++程式而設計。



Source: Wikipedia – Code::Blocks



Code::Blocks 安裝 以Windows為例



codeblocks



全部

圖片

影片

新聞

書籍

更多

工具

約有 2,430,000 項結果 (搜尋時間：0.39 秒)

<https://www.codeblocks.org> ▾ [翻譯這個網頁](#)

Code::Blocks

Code::Blocks is a free C/C++ and Fortran IDE built to meet the most demanding needs of its users. It is designed to be very extensible and fully configurable.

Downloads

[Binary releases](#) - [Source code](#) - [Nightly builds](#) - [Features](#) - ...

搜尋Code::Blocks



Code::Blocks 安裝 以Windows為例



選擇 Downloads

Code::Blocks / Downloads

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, Code::Blocks will be installed, ready for you to work with it. Can't get any ea

- Download a nightly build

選擇 Download the binary release



Code::Blocks 安裝 以Windows為例



Microsoft Windows

File	Download from
codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	FossHUB or Sourceforge.net
<u>codeblocks-20.03mingw-setup.exe</u>	<u>FossHUB</u> or <u>Sourceforge.net</u>
codeblocks-20.03mingw-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-nosetup.zip	FossHUB or Sourceforge.net

可以選這二個其中一個

選擇有 mingw 的，能夠
順便安裝編譯器


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).




Code::Blocks 安裝 以Windows為例

Home / Browse / Development / Integrated Development Environments (IDE) / Code::Blocks



Code::Blocks



A free C, C++ and Fortran IDE
Brought to you by: [killerbot](#), [mandrav](#), [mortenmacfly](#), [thomas-denk](#)




★★★★★ 144 Reviews

Downloads: 100,581 This Week

Last Update: 1 day ago

 [Download](#)  [Get Updates](#) [Share This](#)

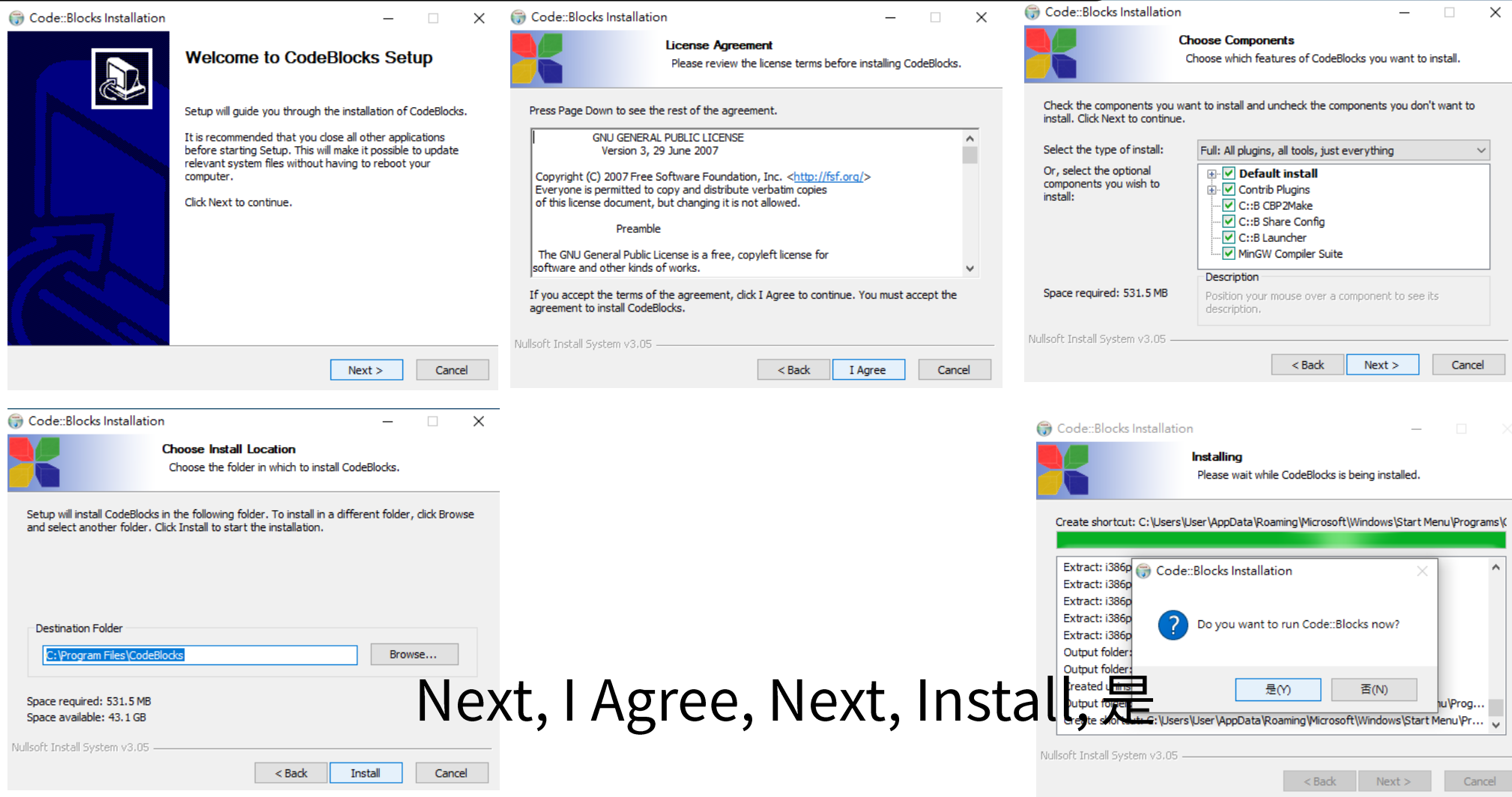
Summary	Files	Reviews	Support	SVN Repo	Git ▾	Tickets	Wiki	Donate 
-------------------------	-----------------------	-------------------------	-------------------------	--------------------------	-----------------------	-------------------------	----------------------	---

Code::Blocks is a free, open-source, cross-platform C, C++ and Fortran IDE built to meet the most demanding needs of its users. It is designed to be very extensible and fully configurable.

下載下來，並執行安裝檔

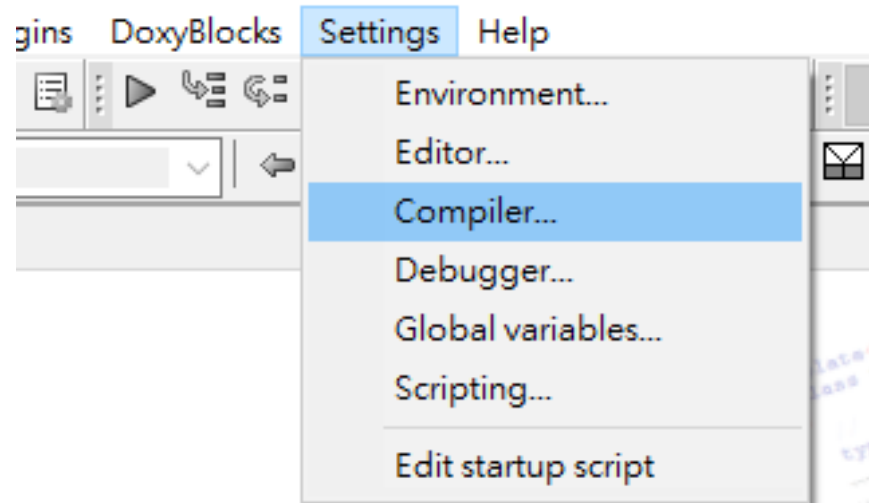


Code::Blocks 安裝 以Windows為例

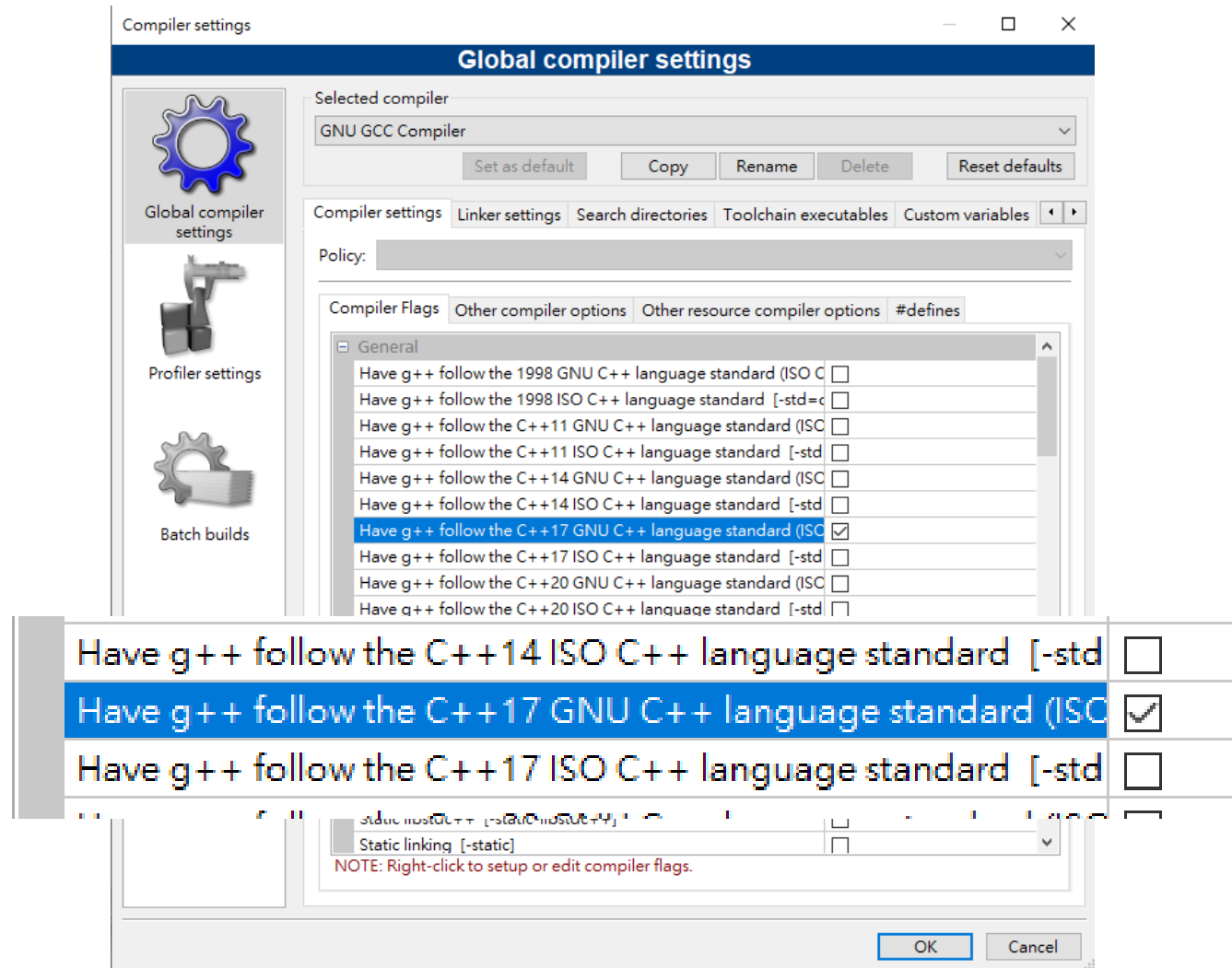




選擇編譯器



Settings -> Compiler...



選一個較新的，例如上面這個



C++起手式

```
1  #include <iostream>           //輸入輸出的函式庫
2  using namespace std;          //使用命名空間 std
3  /* 一些註解，給人看。        }
4     並不是給電腦看！ */       }
5  int main () {                 }
6      cout << "Hello, World!";
7      return 0;
8  }
```

單行註解

區塊註解

程式起點

程式敘述



來輸出一些東西吧！

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      cout << "Hello, World!";
6      return 0;
7  }
```

A screenshot of a Windows command prompt window titled '1.exe'. The window has standard Windows window controls (minimize, maximize, close) in the title bar. The command prompt shows the output of the program: 'Hello, World!'. Below this, it displays 'Process returned 0 (0x0)' and 'execution time : 0.023 s'. At the bottom, it says 'Press any key to continue.'.

```
1.exe
Hello, World!
Process returned 0 (0x0)    execution time : 0.023 s
Press any key to continue.
```



如果想要輸出多行呢？這樣可以嗎？

```
1    #include <iostream>
2    using namespace std;
3
4    int main () {
5        cout << "Never gonna give you up";
6        cout << "Never gonna let you down";
7        cout << "Never gonna run around and desert you";
8        return 0;
9    }
```



句子們都連在一起了！

kulfyapp.com

A large yellow thinking face emoji is centered on a dark brown background. The emoji has a yellow face with a single raised eyebrow, a downward-curving mouth, and two hands resting on its chin. The hands are yellow with orange fingers. The text 'kulfyapp.com' is written in white in the top left corner.



跳脫字元 (Escape Character) 介紹

在編寫程式語言時，可能因為各種原因，無法直接在程式碼中寫出需要的字元，這時候跳脫字元就可以派上用場！

"\'"	# 單引號	"\r"	# 輸入鍵 (Enter)
"\""	# 雙引號	"\t"	# Tab
"\""	# 反斜線	"\v"	# 垂直 Tab
"\a"	# 蜂鳴器警報	"\888"	# 八進位 ASCII 字元
"\b"	# 退格	"\xFF"	# 十六進位 ASCII 字元
"\e"	# 跳脫	"\uFFFF"	# 十六進位萬國碼字元
"\f"	# 換頁	"\u{0}".." \u{10FFFF}"	# 十六進位萬國碼字元
"\n"	# 換行		



有了跳脫字元，我們可以這樣做

```
1  #include <iostream>
2  using namespace std;
3
4  int main () {
5      cout << "Never gonna give you up" << endl;
6      cout << "Never gonna let you down\n";
7      cout << "Never gonna run around and desert you\n";
8      return 0;
9  }
```

加上 << endl

句尾加上 \n



其實也可以這樣做

```
1  #include <iostream>
2  using namespace std;
3
4  int main () {
5      cout << "Never gonna give you up" << endl <<
        "Never gonna let you down\n" << "Never gonna run
        around and desert you\n";
6      return 0;
7  }
```

直接使用 << 串接多個元素



如果我們想這樣呢？

```
C:\N 1.exe × + ∨ — □ ×

一      二      三      四      五
英文    地理    國文    國文    英文
生物    歷史    體育    物理    體育
資訊    英文    英文    選修    數學
資訊    公民    生物    選修    歷史

Process returned 0 (0x0)    execution time : 0.063 s
Press any key to continue.
|
```




可以這樣做

```
1    #include <iostream>
2    using namespace std;
3
4    int main () {
5        cout << "一\t二\t三\t四\t五\t\n"
6              << "英文\t地理\t國文\t國文\t英文\t\n"
7              << "生物\t歷史\t體育\t物理\t體育\t\n"
8              << "資訊\t英文\t英文\t選修\t數學\t\n"
9              << "資訊\t公民\t生物\t選修\t歷史\t\n";
10       return 0;
11    }
```

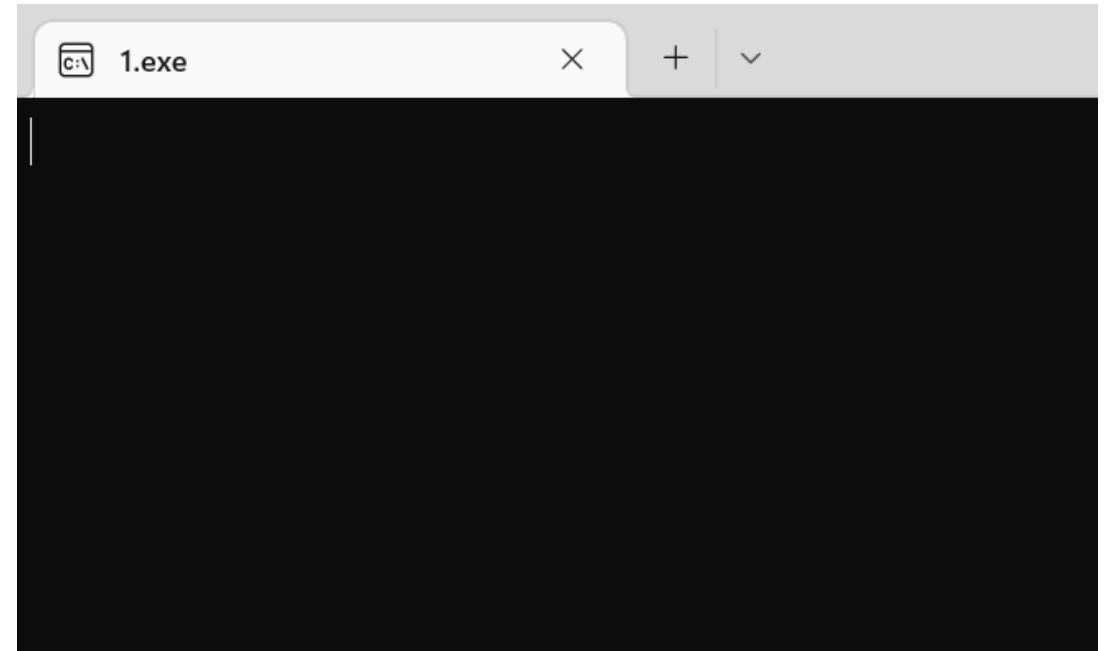
利用 \t 來產生空格

\n 換行



如果我想要輸入東西？

```
1    #include <iostream>
2    using namespace std;
3
4    int main () {
5        string name;
6
7        cin >> name;
8        cout << "Hello, " << name << endl;
9    }
```





如果我想要輸入東西？

```
1    #include <iostream>
2    using namespace std;
```

3

```
4    int main () {
```

```
5        string name;
```

要先宣告一個字串來存輸入的字串

6

```
7        cin >> name; 利用cin >>來輸入到字串name中
```

```
8        cout << "Hello, " << name << endl;
```

```
9    }
```

輸出"Hello, name"

