

# C程式語言

變數

格式化輸入輸出

陳佳宜

# 低階語言 V.S. 高階語言





# 2019 程式語言排名

May 2019	May 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.005%	-0.38%
2	2		C	7.830%	+2.24%
3	3		C++	7.830%	+2.43%
4	4		Python	5.193%	+2.64%
5	6	▲	Visual Basic .NET	3.984%	+1.07%
6	5	▼	C#	2.690%	-0.42%
7	8	▲	JavaScript	2.555%	+0.57%
8	9	▲	SQL	2.489%	-0.83%
9	7	▼	PHP	1.816%	+0.82%
10	13	▲	Assembly language		



- 執行快速!
- 程式設計檢定/比賽常用程式語言
- 臺灣多數大學資訊類科指定程式語言

# 程式如何運作？



# 你還想要哪些額外功能?

---

- 一鍵啟動 (編譯+執行)
- 美美的排版工具，看起來整齊舒適，心情好!
- 告訴我錯哪裡
- 給我一個程式範本看看先~
- 能夠自動寫程式更好



# 整合開發環境

## Integrated Development Environment

### Text Editor

Syntax Coloring

Autocomplete

Indexer

Doc Viewer

Code Templates



### Builder

Compiler

Lexer

Parser

Assembler

Linker



### Debugger GUI

Debugger



## Text Editor

Syntax Coloring

Autocomplete

Indexer

Doc Viewer

Code Templates



# 文字編輯器：語法突顯

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

int main()
{
    printf("Hello world!\n");
    return 0;
}
```

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

int main()
{
    printf("Hello world!\n");
    return 0;
}
```





## Text Editor

Syntax Coloring

Autocomplete

Indexer

Doc Viewer

Code Templates



# 文字編輯器：自動完成

```
*main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      pri
7      printf(): int
8      priority_queue
9      private
```

int printf(const char\*, ...) (function)  
[Open declaration](#)  
[Close Top](#)



## Text Editor

Syntax Coloring

Autocomplete

Indexer

Doc Viewer

Code Templates



# 文字編輯器

- 索引
- 文件閱覽器
- 程式碼範本

## Builder

### Compiler

Lexer

Parser

Assembler

Linker



# 編譯器：詞法分析

- 範例：sum = 3 + 2;

語素	標記類型
sum	標識符
=	賦值操作符
3	數字
+	加法操作符
2	數字
;	語句結束



## Builder

### Compiler

Lexer

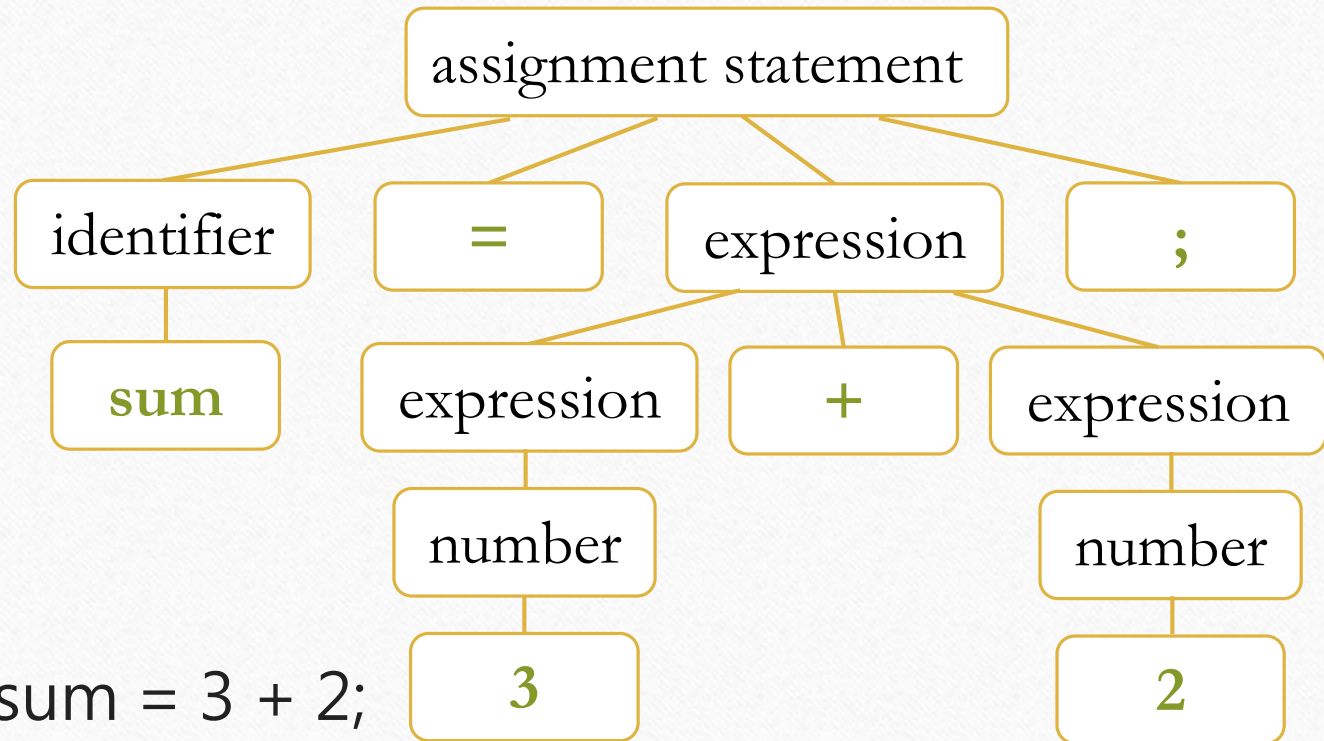
**Parser**

Assembler

Linker



# 編譯器：語法分析



- 範例：sum = 3 + 2;

## Builder

### Compiler

Lexer

Parser

**Assembler**

Linker



## 編譯器：組合語言

AL 

1	1	1	1	0	1	1	0
---	---	---	---	---	---	---	---

(a)

SHR AL, 1

CF

AL 

0	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---

 → 

0
---

(b)

SAR AL, 1

CF

AL 

1	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---

 → 

0
---

(c)

CF ← SHL/SAL AL, 1

1
---

 ← 

1	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---

 AL



## Builder

### Compiler

Lexer

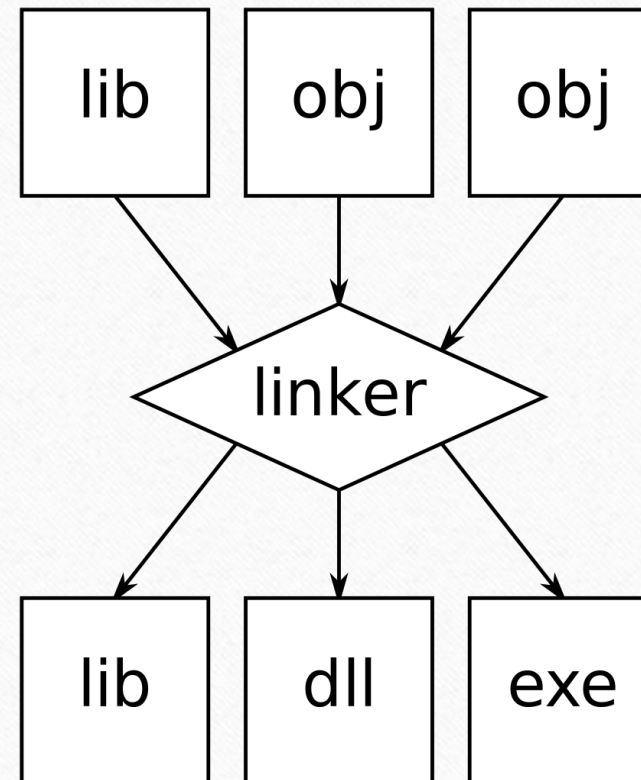
Parser

Assembler

Linker



## 編譯器：連結器



## Debugger GUI

Debugger



## 圖形化除錯器

```
main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      printf "Hello world!\n");
7      return 0;
8  }
9
```

Logs & others

- Code::Blocks
- Search results
- Cccc
- Build log
- Build messages
- CppCheck

File	Line	Message
=== Build: Debug in SyntaxColoring (compiler: GNU GCC Compiler) ===		
In function 'main':		
C:\Users\tp6...		warning: statement with no effect [-Wunused-value]
C:\Users\tp6... 6		error: expected ';' before string constant
C:\Users\tp6... 6		error: expected statement before ')' token
=== Build failed: 2 error(s), 1 warning(s) (0 minute(s), 0 second(s)) ===		

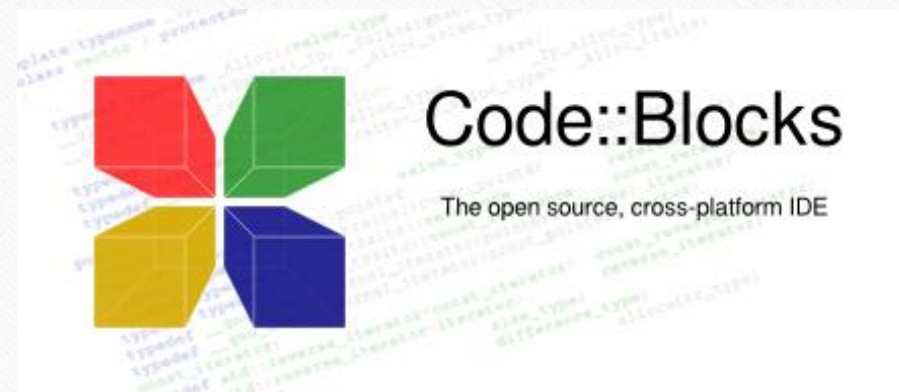
補充說明：Graphical User Interface (圖形化使用者介面)



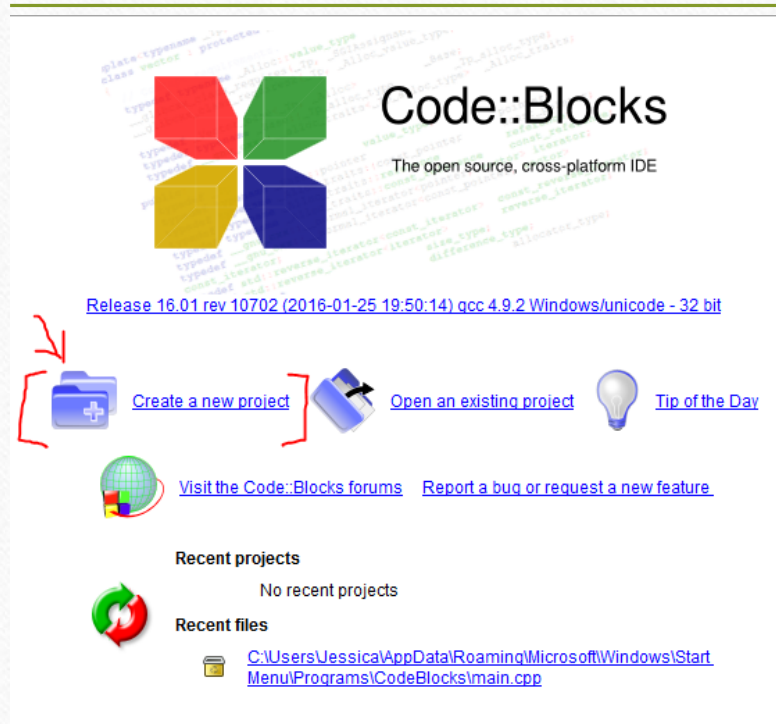
# Code::Blocks

---

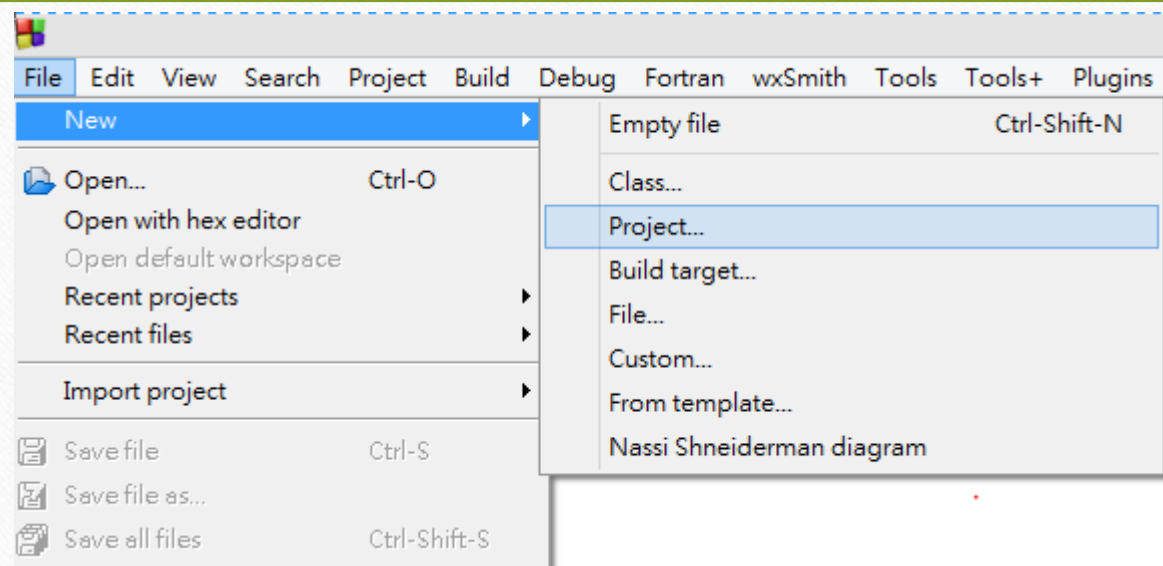
IDE 介紹



# Step 1. 開啟新專案(以下兩種方法皆可)

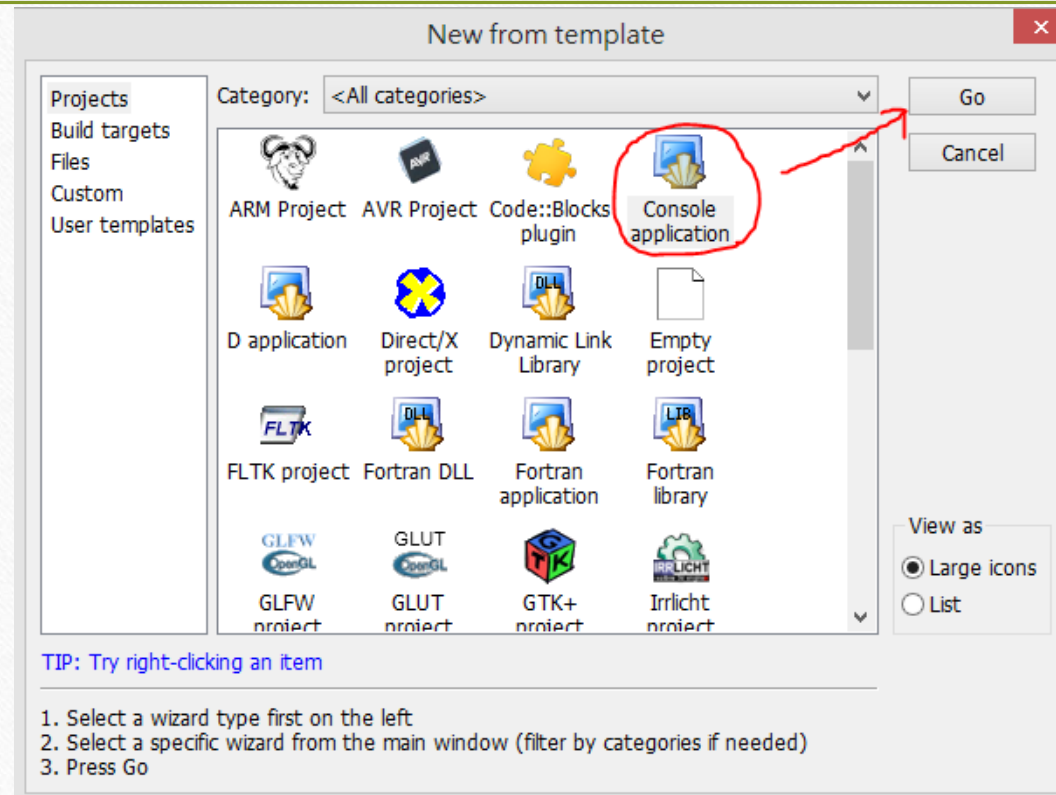


© 2004 - 2015, The Code::Blocks Team.

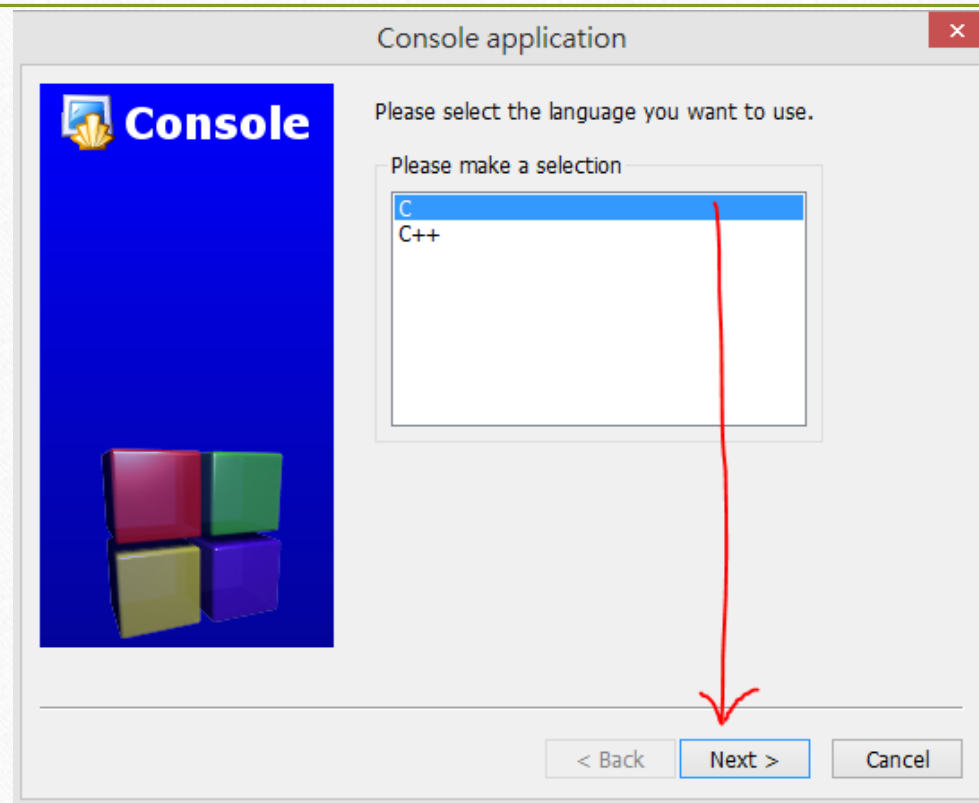




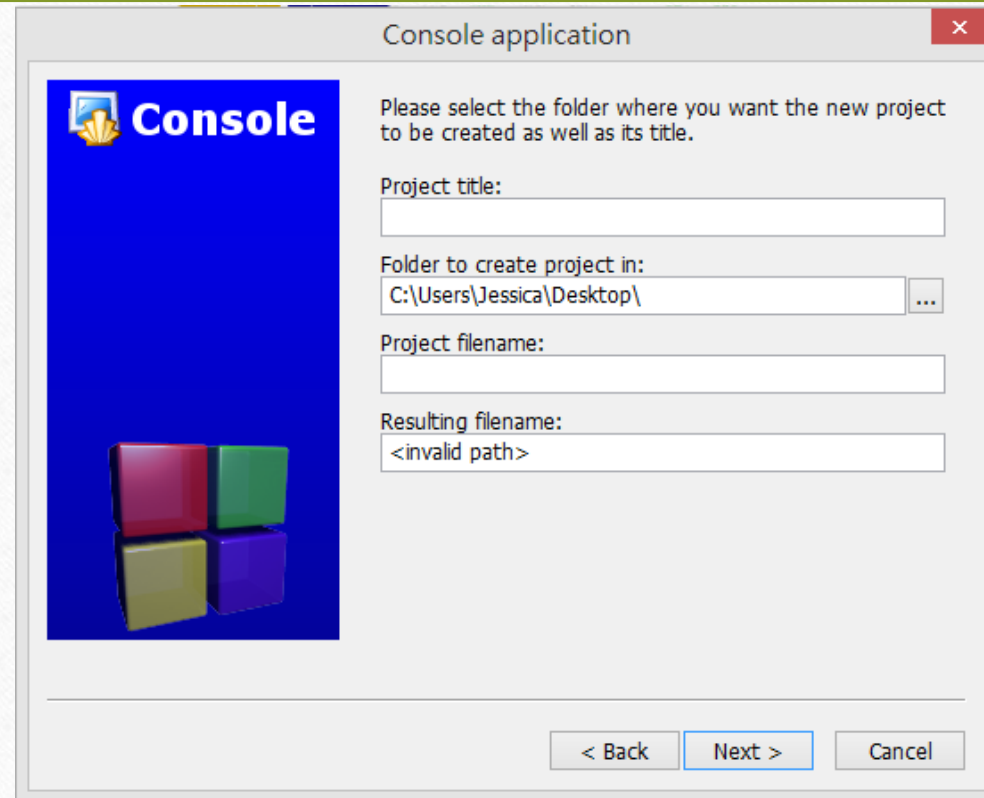
## Step 2. 選擇 Console application



## Step 3. 選擇 C



## Step 4. 填寫 Project title (Project filename 預設同 Project title)



Console application

**Console**

Please select the folder where you want the new project to be created as well as its title.

Project title:

Folder to create project in:  
 ...

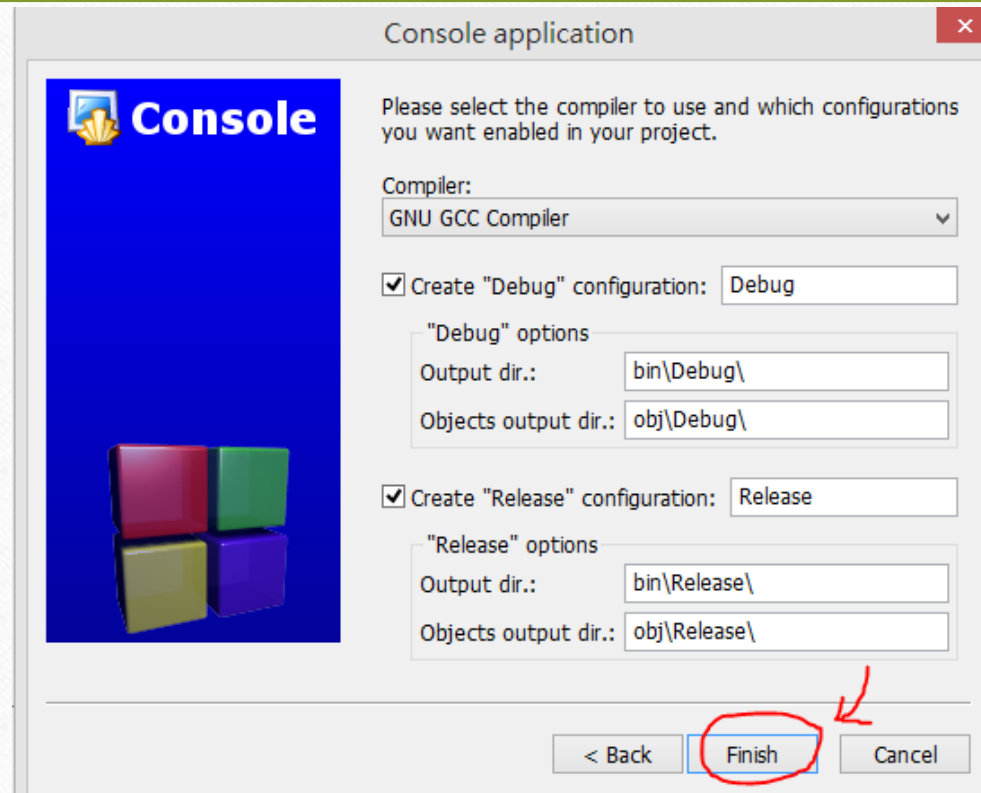
Project filename:

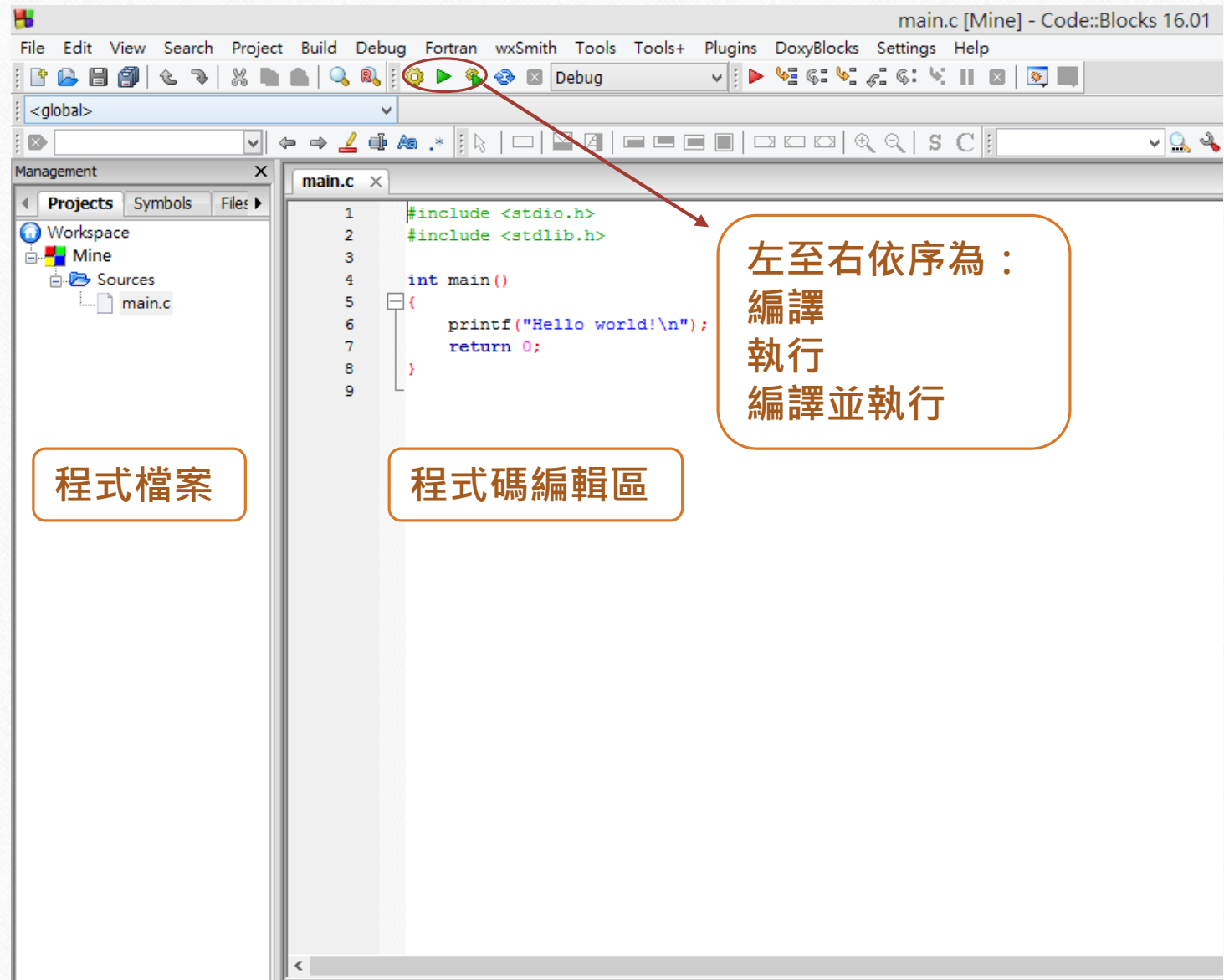
Resulting filename:

< Back   Next >   Cancel



## Step 5. 選擇編譯器 (Compiler)





程式檔案

程式碼編輯區

左至右依序為：  
編譯  
執行  
編譯並執行

# 如何輸出指定的內容？

---

格式化輸出：`printf`



# 觀察以下的範例程式

```
main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      printf("Hello world!\n");
7      return 0;
8  }
9
```

- 第1~2行
  - 將某 程式檔或標頭檔包括在目前的程式內,使目前的程式可引用該檔內的資料或程式
- 第4行
  - 主函式, 程式開始執行的地方

# 觀察以下的範例程式

```
main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      printf("Hello world!\n");
7      return 0;
8  }
9
```

- 第6行
  - 在螢幕上印出""內的字
  - \n 為換行
- 第7行
  - 主函式的回傳值為0

# 課堂練習

---

請嘗試使用 `printf` 輸出一段自我介紹!

 `printf ( "欲輸出的字串" );`

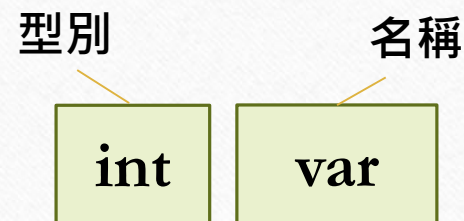


# 複習：變數

---

# 變數的宣告

- 宣告變數必須具備**型別**和**名稱**
- 變數像是杯子
  - 小杯 → 中杯 → 大杯
  - short → int → long
- 每個杯子都要裝載數值
  - 告訴店員 我要「一杯大的紅茶拿鐵」
  - 告訴編譯器 我要「一個 int 變數保存數字90」 (小差異：變數要命名，但杯子不用)



# 常用的變數型別

資料型別		用途
數字	int	整數
	float	浮點數 (小數)
字元	char	字元



# 整數

---

- int
- 格式指定字 ( format specifier ) : %d
- 宣告一個整數變數 → `int num;`

# 浮點數

---

- float
- 小數
- 格式指定字（ format specifier ）： %f
- 宣告一個浮點數變數 → float num;

# 字元

---

- char
- 使用 ASCII 碼表
- 以下寫法具有相同的效果

```
char ch = 'A';  
char ch = 65;
```


- 格式指定字 ( format specifier ) : %c




# 指定運算子 =

- 符號為 =
  - 將等號右方的值(變數或運算式)指定給等號左方的變數


```
int a;  
a = 10;
```




```
int a;  
a = b;
```



```
int a,b;  
a = b + 5;
```



```
int a,b,c;  
a = 3;  
b = 6;  
a + b = c;
```



如何印出變數的內容？

---

# printf ( "欲輸出的字串格式", 引<sub>用的變</sub>數 )

---

- printf ("今天溫度24度");
  - 氣象局顯示的溫度每天不一樣，只能寫出所有的狀況嗎?
- printf ("今天溫度%d度", temperature );
  - 變數的好處 → 使用同一段敘述句即可解決多種狀況
  - 使用變數(例：temperature)前，須先**宣告**(指定型別)並**賦值**(給予初始值)
  - 引數需配合與變數相符的**格式指定字**才可正確輸出



# 常見的格式指定字

符號	代表意義
%d	整數
%f	浮點數 (帶小數的值)
%c	字元
%s	字串

# 如何偵測使用者輸入的資料？

---

格式化輸入：scanf

# scanf( "控制字串", 引數 )


---

- control string 控制字串
  - 一次可輸入多個變數
    - 以空白或Enter鍵區隔
- arguments 引數
  - **&**變數名稱
- 範例：
  - scanf("%d %d",&a,&b);
  - 將鍵盤輸入的兩個整數，依序存至a和b



# 作業：AI Robert

---

 請寫出一個簡易問答機器人，能夠問妳問題，並根據妳的答案進行簡易交談。所有標點符號皆使用**全型**!

# 輸入輸出範例 I

---

## 範例輸出

你好，新來的朋友，請用一個字母代表你自己：

X，請問你今年幾歲？

我也是18歲耶！好巧！

(換行)我覺得圓周率是世界上最美的數字，你會背圓周率嗎？

不錯嘛！竟然可以背出3.141590！

(換行)我得下線了，下次聊：D

## 範例輸入

- X
- 18
- 3.14159

## 輸入輸出範例 II

```
C01_AIRobert\bin\Debug\C01_AIRobert.exe"
你好，新來的朋友，請用一個字母代表你自己：A
A，請問你今年幾歲？12
我也是12歲耶！好巧！
我覺得圓周率是世界上最美的數字，你會背圓周率嗎？3.1
不錯嘛！竟然可以背出3.100000！
我得下線了，下次聊：D
Process returned 0 (0x0)   execution time : 16.765 s
Press any key to continue.
```



# 如何繳交作業

---

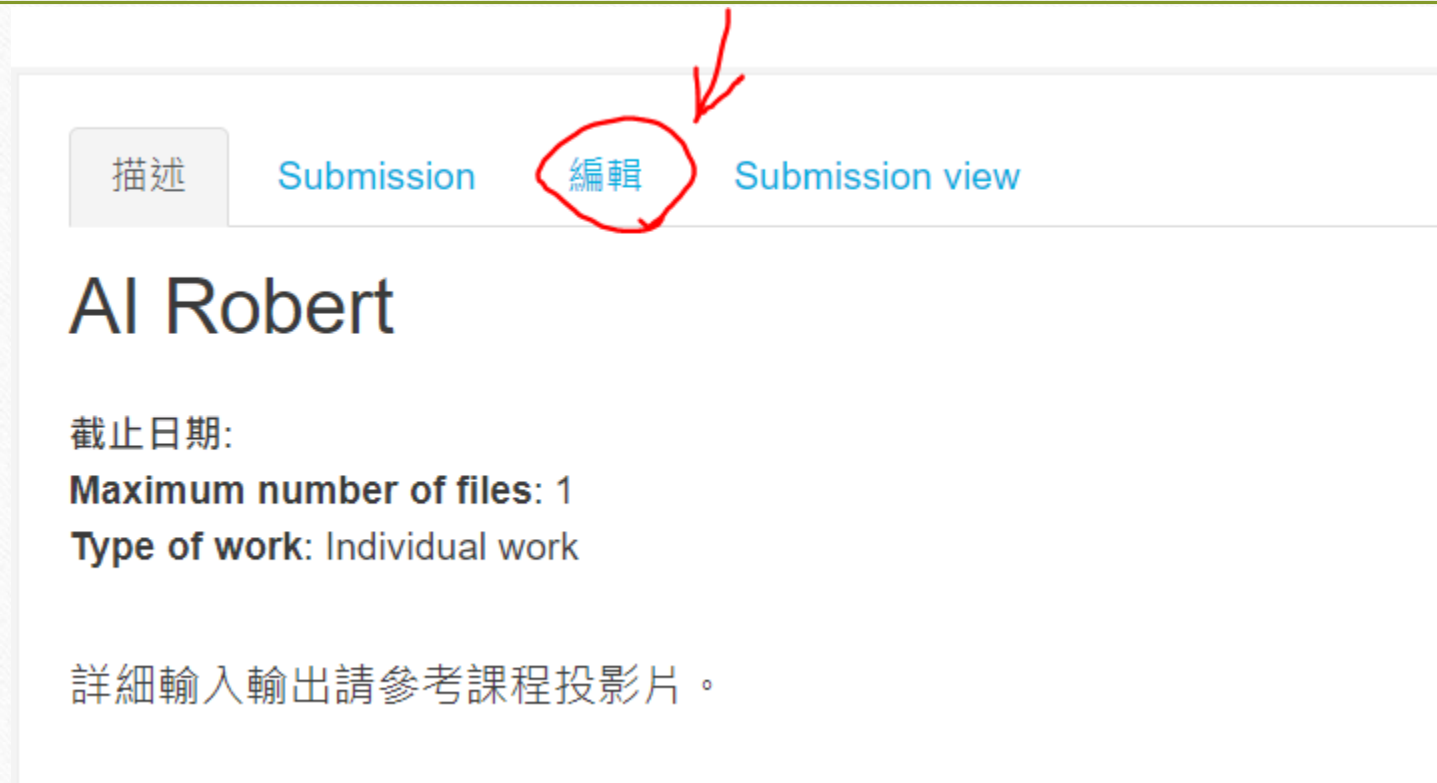
使用 Moodle 的 Online Judge

# Step 1. 點選



AI Robot

## Step 2. 點選「編輯」分頁



The screenshot shows a web interface for a submission page. At the top, there is a horizontal navigation bar with four tabs: '描述' (Description), 'Submission', '編輯' (Edit), and 'Submission view'. The '編輯' tab is circled in red, and a red arrow points to it from above. Below the navigation bar, the main content area displays the title 'AI Robert'. Underneath the title, there are three lines of text: '截止日期:' (Deadline:), 'Maximum number of files: 1', and 'Type of work: Individual work'. At the bottom of the content area, there is a line of text: '詳細輸入輸出請參考課程投影片。' (For detailed input/output, please refer to the course presentation slides.).

描述 Submission **編輯** Submission view

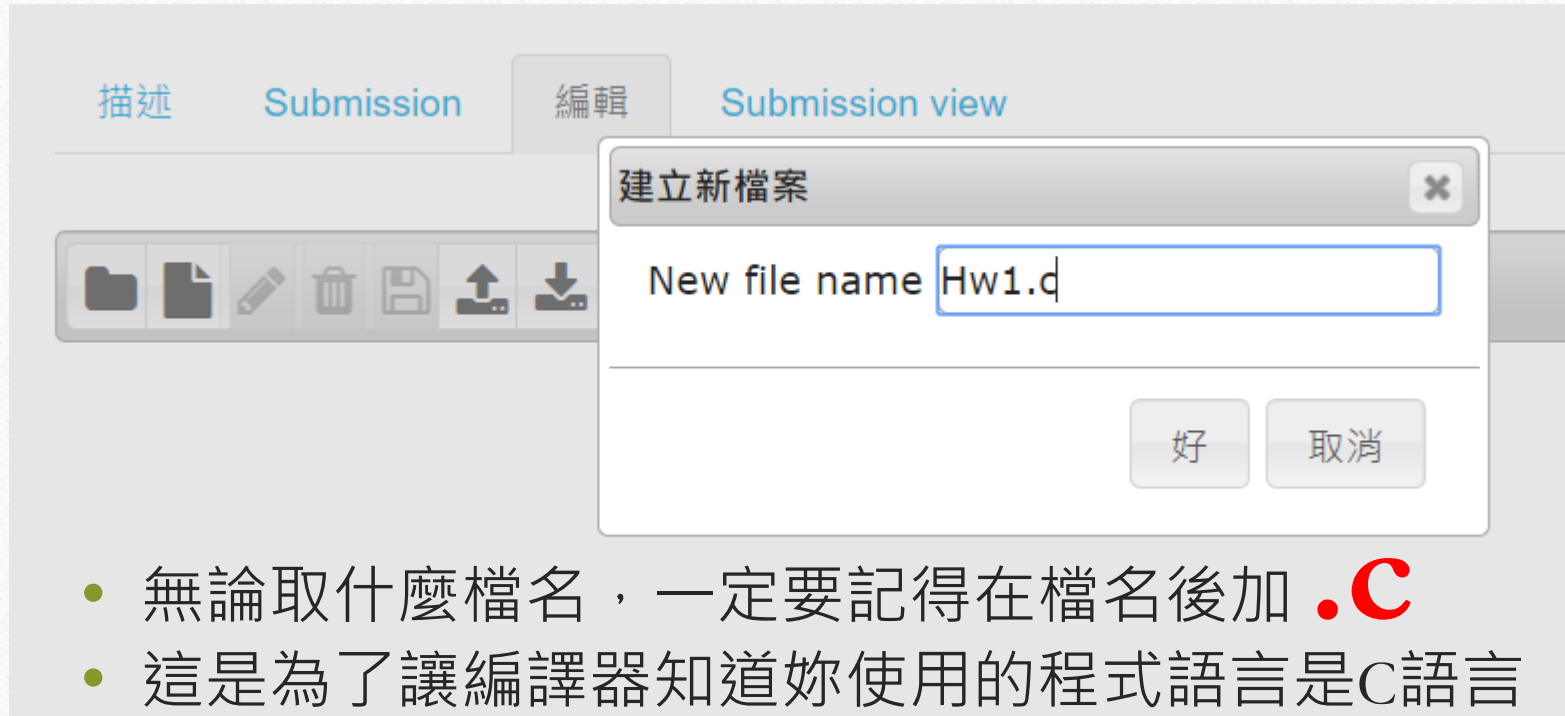
### AI Robert

截止日期:  
**Maximum number of files:** 1  
**Type of work:** Individual work

詳細輸入輸出請參考課程投影片。



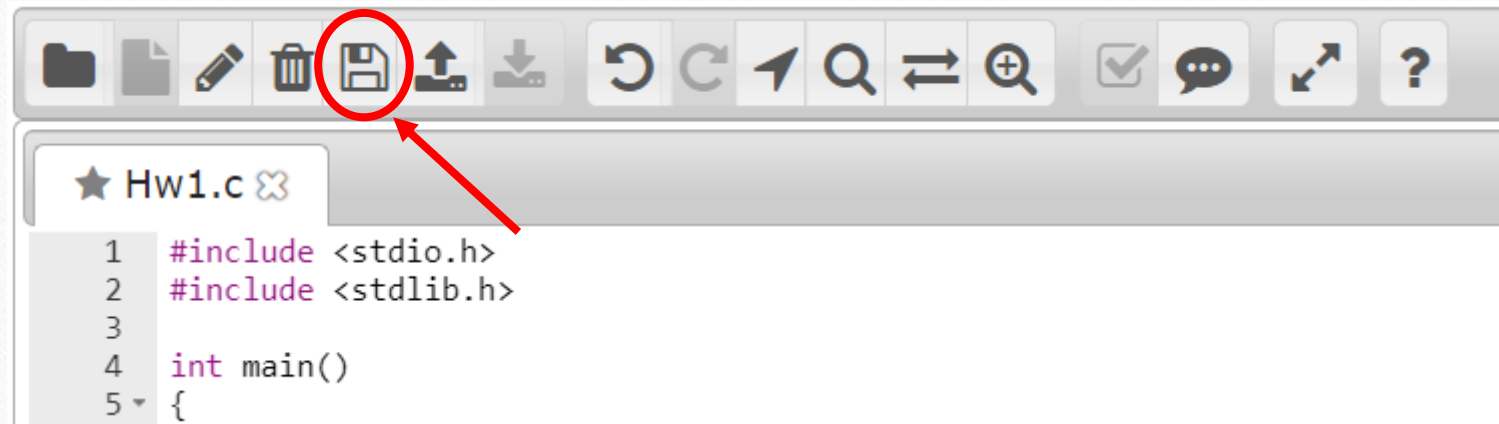
## Step 3. 輸入檔名，檔名可自取



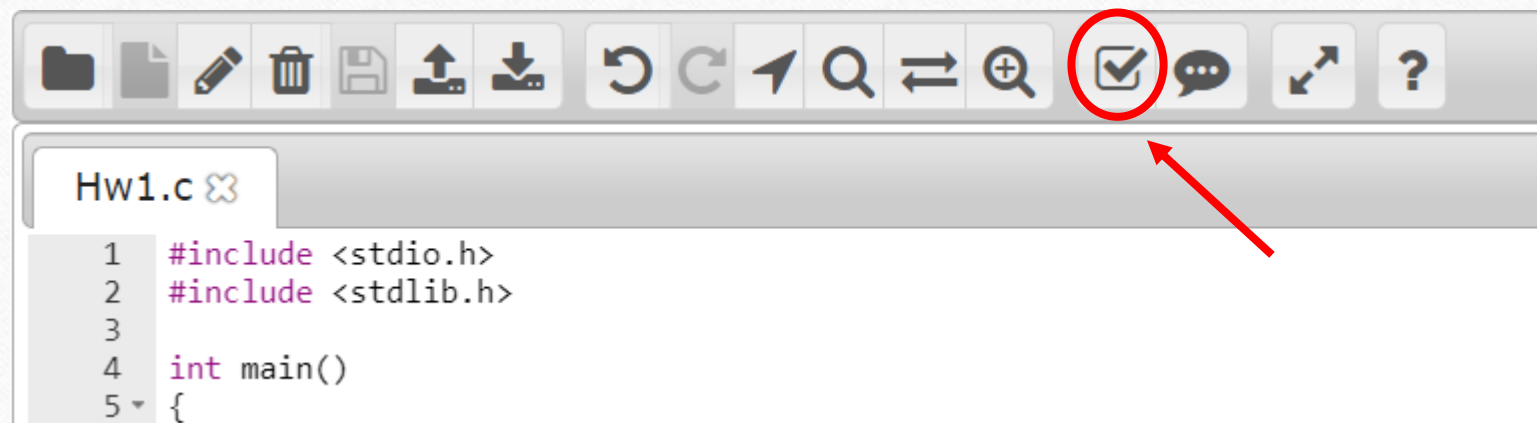
無論取什麼檔名，一定要記得在檔名後加 **.C**

這是為了讓編譯器知道妳使用的程式語言是C語言

Step 4. 貼上完整程式碼，並點選存檔



## Step 5. 點選評估，即能自動評分





# 評估結果顯示於右欄

- Proposed grade：所得分數/總分
  - 顯示目前程式所得分數
- --- Program output ---
  - 你撰寫的程式輸出之結果
- --- Expected output (exact text)---
  - 預期的輸出 (正確答案)
- Summary of tests
  - 測試筆數/通過筆數

The screenshot shows a programming evaluation interface. At the top, a red box highlights the text "Proposed grade: 80 / 100". Below this, a section titled "Test 1: Test" shows an "Incorrect program result". A red box highlights the "Program output" section, which contains the text: "你好，新來的朋友，請用一個字母代表你... 我覺得圓周率是世界上最美的數字，你會... 我得下線了，下次聊：D". Another red box highlights the "Expected output (exact text)" section, which contains the same text: "你好，新來的朋友，請用一個字母代表你... 我覺得圓周率是世界上最美的數字，你會... 我得下線了，下次聊：D". At the bottom, a red box highlights the "Summary of tests" section, which shows a table with the following content:

Summary of tests	
1	test run/ 0 tests passed

# The End

---

See you next week !

陳佳宜