

計算機程式設計

Computer Programming

C Basics

Instructor: 林英嘉

2024/09/09

Outline

- C語言程式骨架
- 變數
 - 變數(資料)型態
 - 宣告變數
- printf 與 scanf
- 常見程式錯誤

C語言程式骨架: 首先你要有main

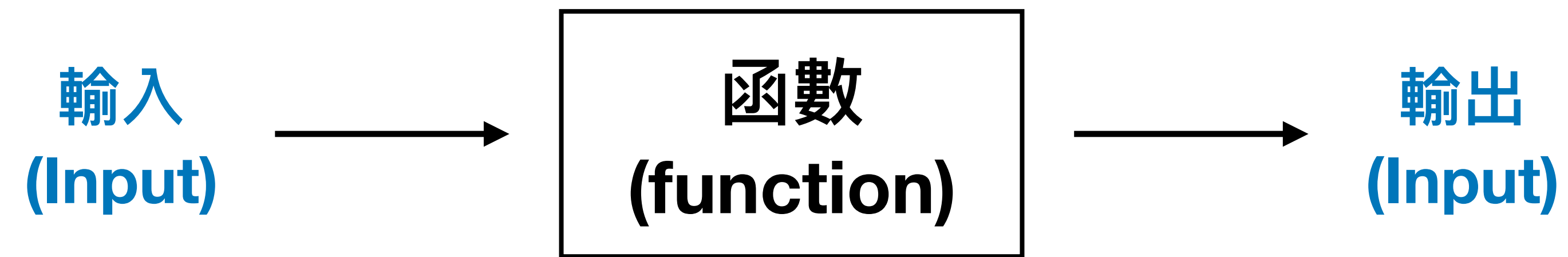
```
main() {  
  
}  

```

← 函數內容 (body): 程式碼寫在這裡面

- main 代表主程式，是C語言程式不可或缺的部分
- main 有小括弧，代表函數
- 小括弧目前是空的，代表此函數不需要放入參數
- 大括弧代表主程式從左大括弧開始，到右大括弧結束

函數



C語言程式骨架

代表函數回傳值為整數型態

函數的輸入

1
2
3

```
int main(){  
    // body  
    return 0;  
}
```

函數的輸出

- return 0; 代表main函數執行完畢，回傳整數0
 - 多數時候，main函數中沒有寫這行不會有問題
- 程式是一行一行執行，也就是body執行完後才會執行return 0;那行
- 注意分號

開始撰寫C語言 (使用線上編譯器)

- <https://ideone.com>
- AdGuard 廣告封鎖器
 - <https://addons.mozilla.org/zh-TW/firefox/addon/adguard-adblocker/>
 - <https://chromewebstore.google.com/detail/bgnkhhnnamicmpeenaelnjfhikgbkllg?hl=zh-TW>

簡單範例

```
#include <stdio.h>
int main() {
    int a = 2024;
    printf("%d", a);
}
```

簡單範例

```
#include <stdio.h>
int main() {
    int a = 2024;
    printf("%d", a);
}
```

標頭檔，可用來引入函式庫
代表可以使用 stdio.h 中的函式

簡單範例

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

```
int main() {
```

```
    int a = 2024;
```

```
    printf("%d", a);
```

```
}
```

標頭檔，可用來引入函式庫
代表可以使用 stdio.h 中的函式

引入 stdio.h 就可以使用 printf 函數

簡單範例

```
#include <stdio.h>
int main() {
    int a = 2024;
    printf("%d", a);
}
```

注意! 大括號中的每一行以分號做結尾
標頭檔不用加分號

stdio.h (standard input / output) 有什麼？

(僅列出部分函式)

Formatted input/output:

<u>fprintf</u>	Write formatted data to stream (function)
<u>fscanf</u>	Read formatted data from stream (function)
<u>printf</u>	Print formatted data to stdout (function)
<u>scanf</u>	Read formatted data from stdin (function)
<u>snprintf</u>	Write formatted output to sized buffer (function)
<u>sprintf</u>	Write formatted data to string (function)
<u>sscanf</u>	Read formatted data from string (function)
<u>vfprintf</u>	Write formatted data from variable argument list to stream (function)
<u>vfscanf</u>	Read formatted data from stream into variable argument list (function)
<u>vprintf</u>	Print formatted data from variable argument list to stdout (function)
<u>vscanf</u>	Read formatted data into variable argument list (function)
<u>vsprintf</u>	Write formatted data from variable argument list to sized buffer (function)
<u>vsprintf</u>	Write formatted data from variable argument list to string (function)
<u>vsscanf</u>	Read formatted data from string into variable argument list (function)

Figure source: <https://cplusplus.com/reference/cstdio/>

如果你要使用printf，但忘記引入stdio.h

```
comments.c: In function 'main':
comments.c:2:5: warning: implicit declaration of function
'printf' [-Wimplicit-function-declaration]
    2 |     printf("Hello, World!\n");
      |     ^~~~~~
comments.c:2:5: warning: incompatible implicit
declaration of built-in function 'printf'
comments.c:1:1: note: include '<stdio.h>' or provide a
declaration of 'printf'
+++ |+#include <stdio.h>
    1 | int main(){
```

- 許多線上編譯器允許沒有引入stdio.h也能使用printf

為你的程式碼加上註解

```
#include <stdio.h>
int main() {
    // This is a comment for a single line
    /* This
    is a comment
    for multiple lines.
    */
}
```

單行註解：在任意一行加上 //

多行註解：

/*開始

*/結束

編譯器不會對註解的內容進行編譯
可以想成在程式內部做筆記

可以不用 return 0;

- 如果函數沒有要回傳值的話可以不用return 0;
- 但有些compilers會因為主程式中沒有return而產生Warnings

變數

變數 (Variable)

- 在程式語言中，變數扮演者存放資料或是綁定「值」的角色
- 變數的值可以是未知的 (未賦值)，且可能可以在程式執行過程中「改變」
 - 變數代表「可以改變的數」
- (重點!!) 在C語言中，變數需要先宣告才能使用

變數初始化 (以int為例)

```
#include <stdio.h>
int main() {
    int a; 宣告變數
    printf("%d", a);
}
```

宣告變數時

```
int a = 100;
```

- 編譯器會配置記憶體空間給變數a，以整數為例，變數a會佔4個位元組 (byte)

"0x7fffffffddde0"
100

4個位元組

變數命名規定

- 變數名稱可以由英文字母、數字以及底線 ('_') 所構成
- 變數名稱大小寫不同時，視為不同的變數 (int **a**, **A**;))
- 第一個字元不能是數字
- 變數名稱中不能有空白

使用等號運算子進行賦值

```
int a = 2024;  
int b = a;
```

- = 為等號運算子，代表將=右邊的值賦予給左邊的變數
 - 設定a為2024;
 - 設定b為a;
- 注意!! =並非指邏輯上的「相等」，C語言中==才是指邏輯上的「相等」

宣告時設值以及不設值

```
#include <stdio.h>
int main() {
    int a = 2024;
    printf("%d", a);
}
```

輸出為2024

```
#include <stdio.h>
int main() {
    int a;
    printf("%d", a);
}
```

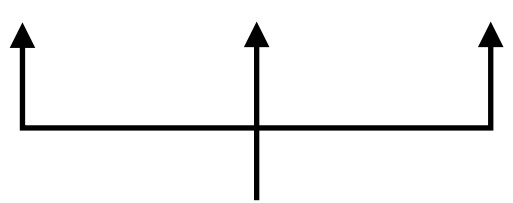
輸出為0

宣告變數時不設值

- 預設為0
- 區域變數（local variables）的數值可能為上一個值
 - 需特別注意此行為

一行宣告多個變數

```
int a, b, c;
```



都是int

優點：整體程式行數較少，可增加程式可讀性

Outline

- C語言程式骨架
- 變數
 - 變數(資料)型態
 - 宣告變數
- `printf` 與 `scanf`
- 常見程式錯誤

printf: 用來進行格式化輸出的函數

```
printf(format, ...);
```

- `printf`: Print **formatted** data.
- `format`: 一段可包含 **format specifiers** 的字串 (string)
- `...`: 值，依序取代 `format` 中 **format specifiers** 位置
 - 不一定要有，數量跟 **format specifiers** 一致

使用 `printf` 進行格式化輸出 (1)

宣告一個變數，並且把它印出來

```
#include <stdio.h>
int main() {
    int a = 2024;
    printf("今年是%d年", a);
}
```


於%d位置處印出

%d 代表以十進位 (decimal) 的形式印出變數 (此程式的a) 值

使用 printf 進行格式化輸出 (2)

宣告三個變數，並且把它們印出來

```
#include <stdio.h>
int main() {
    int year = 2024;
    int month = 2;
    int days = 29;
    printf("今年是%d年，%d月有%d天", year, month, days);
}
```



The diagram consists of red arrows connecting the format specifiers in the printf statement to the corresponding variable values. An arrow points from the first %d to the value 2024. Another arrow points from the second %d to the value 2. A third arrow points from the third %d to the value 29. A fourth arrow points from the %d in the string '今年是%d年' to the value 2024.

於%d位置處印出

How to do this?

輸出

今年是2024年
2月有29天

\n 代表換行符號

輸出

今年是2024年
2月有29天

```
#include <stdio.h>
int main() {
    int year = 2024;
    int month = 2;
    int days = 29;
    printf("今年是%d年\n%d月有%d天", year, month, days);
}
```

\n 代表換行符號

輸出

今年是2024年
2月有29天

```
#include <stdio.h>
int main() {
    int year = 2024;
    int month = 2;
    int days = 29;
    printf("今年是%d年\n%d月有%d天", year, month, days);
}
```

那如果連續放兩個\n呢？

以修飾子來控制輸出欄位的寬度

```
#include <stdio.h>
int main() {
    int target = 95;
    printf("|%-6d|\n", target); // 總共寬度為6個字元
    printf("| %-5d|\n", target); // 前有1個空格
    printf("|  %-4d|\n", target); // 前有2個空格
    printf("|   %-3d|\n", target); // 前有3個空格
    printf("|    %-2d|\n", target); // 前有4個空格
    printf("|%6d|\n", target); // 總共寬度為6個字元
}
```

以修飾子來控制輸出欄位的寬度 (說明)

以寬度 = x 為例

- %**x**d 代表印出來的值總共有**x**個字元，且數值向**右**對齊
- %**-x**d 代表印出來的值總共有**x**個字元，且數值向**左**對齊

Outline

- C語言程式骨架
- 變數
 - 變數(資料)型態
 - 宣告變數
- printf 與 scanf
- 常見程式錯誤

scanf: 用來進行輸入的函數

```
scanf(format, &變數1, &變數2, ...);
```

- scanf: Read **formatted** data and store them according to the locations.
 - 可取得自鍵盤輸入的值
- format: 一段可包含 **format specifiers** 的字串 (string)
- &代表位置運算子，可以將數值存到變數的記憶體位置
- 同printf，第二個參數以後的數量與跟 **format specifiers** 一致

用scanf來取得鍵盤輸入的值

```
#include <stdio.h>
int main() {
    int a;
    scanf("%d", &a);
    printf("%d", a);
}
```

一次讀進多個值

- 以鍵盤輸入多個值時，須以空白作為分隔

```
#include <stdio.h>
int main() {
    int a, b;
    scanf("%d", &a);
    scanf("%d", &b);
    printf("%d\n", a);
    printf("%d\n", b);
}
```

```
#include <stdio.h>
int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", a);
    printf("%d\n", b);
}
```

scanf 會自動跳過空格，格式
字串中不一定要有空格分開

在用scanf讀取資料時輸入過多數值

```
#include <stdio.h>
int main() {
    int target;
    scanf("%d", &target);
    printf("輸入的數字是: %d", target);
}
```

- (範例) 輸入：95 96
- 輸出為95 (第一個值)

指定長度來使輸入時不加空格

```
#include <stdio.h>
int main() {
    int a, b;
    scanf("%1d%1d", &a, &b);
    printf("%d\n", a);
    printf("%d\n", b);
}
```

- %1d 告訴 scanf 每次只讀取 1 個整數

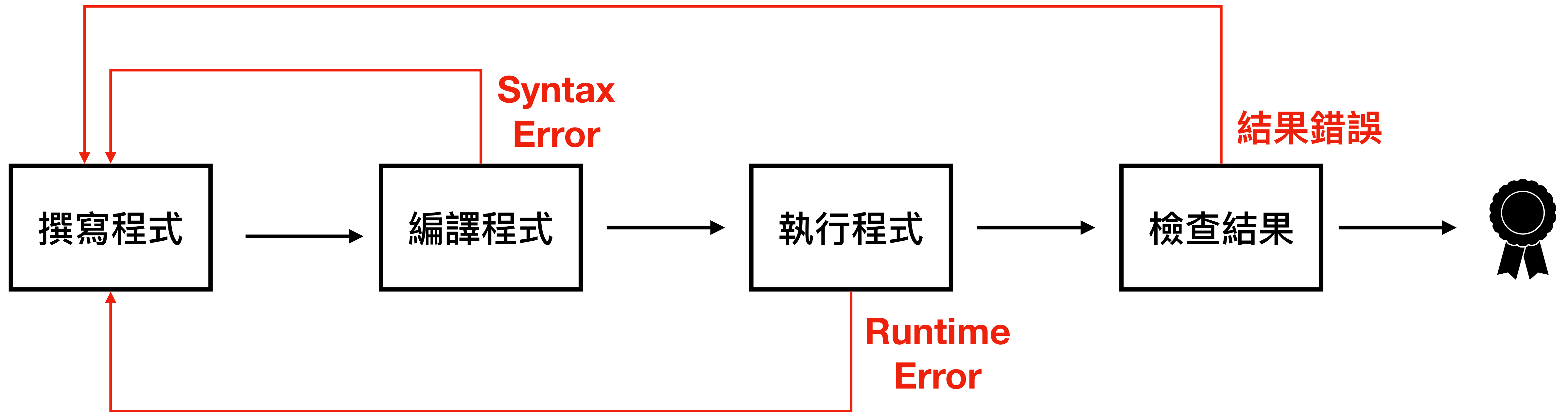
Syntax Error

- C語言有非常嚴謹的語法規定
- syntax 代表「語法」
- Syntax Error 代表程式碼不符合C的規定，導致編譯失敗 Compilation Error (CE)
 - 沒加分號
 - 括號沒關好
 - ...

Runtime Error

- Runtime Error (RE) 代表程式寫法符合C語言的規定
- 但是出於某種問題導致程式執行後出現錯誤
 - 無窮迴圈
 - 除法時0在分母
 - ...

程式撰寫流程



→ 紅色箭頭代表Debug