

# C++基礎語法 Unit-2

---

- 基本輸入與輸出
- 變數
- 算數運算
- 資料型態

# 建置開發環境 (IDE)

- IDE (Integrated Development Environment)
  - [Code::Blocks](#) (高中生參加的檢定、比賽中常見的編輯除錯工具)
  - Xcode (for MacOS)
- 轉寄程式碼給別人的時候 (有syntax highlighter才容易看懂)
  - [Ubuntu Pastebin](#)
- 線上程式開發工具
  - <https://replit.com/>
- 自修參考書：
  - [C++程式設計入門\(第二版\)](#)

# 程式編譯與執行

- 寫程式的過程，大致上分成幾個步驟，分別為
  - 編輯 (edit)
  - 編譯 (compilation or **build**)
  - 執行 (execution or **run**)
  - 除錯 (debug)
- Build and Run
  - 快捷鍵 **F9** (Code::Blocks)

# Online Judge 線上評測系統

- 線上題庫，練習與測試程式代碼的方便工具
  - [ZeroJudge 高中生程式解題系統](#)
  - [AtCoder](#)
  - [CSES Problem Set](#)
- 參考網站：
  - [高中資訊科技概論教師黃建庭的教學網站](#)
  - [YUI HUANG 演算法學習筆記](#)

# 起手式

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      // 單行註解
6
7      /*
8          多行註解
9      */
10
11     return 0;
12 }
```

- #include <iostream>
- using namespace std;
- //
- /\* ... \*/
- main() { ... }
- ;
- return 0;
- 注意：半形符號

# 風格

There are two types of people.

```
if (Condition)
{
    Statements
    /*
     *
     */
}
```

```
if (Condition) {
    Statements
    /*
     *
     */
}
```

Programmers will know.

一天我路過一座橋，碰巧看見一個人想跳河自殺。我跑過去對他大喊道：「別跳，別死啊。」

「為什麼不讓我跳？」他說。

「因為還有很多東西值得我們活下去啊。」

「有嗎？比如說？」

「呃.....你做什麼工作？」

「程序員。」

我說：「我也是！瞧，有共同點了吧。你是軟體還是硬體？」

「軟體。」

「我也是！PC 還是 Web？」

「PC。」

「我也是！Windows 還是 Linux 平台？」

「Linux。」

「我也是！那你使用 C 還是 C++？」

「C++。」

「我也是。1998 年的 C++ 98 還是 2011 年的 C++ 11？」

「2011 年的 C++ 11。」

「我也是。大括弧寫在後面還是寫在下一行？」

「下一行。」

「去死吧！你這個異教徒人渣！」我一把將他推下橋去。

知乎專欄——代碼戰爭 <http://zhuanlan.zhihu.com/codewar/19700946>

# 輸入與輸出

- `cout`
  - 透過螢幕輸出程式執行/計算的結果
- `cin`
  - 從鍵盤取得輸入的資料
  - 資料會以「空白」或「換行符號」分割成一筆一筆

# 從螢幕輸出 cout

Example 2-1

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      cout << "Hello World!\n";
6
7      cout << "Hello World!" << endl;
8      return 0;
9  }
10
```

用一對雙引號標註一個字串

← “\n” (a new line)

← endl (end of a line)



```
Hello World!
Hello World!
Program ended with exit code: 0
```

<https://pastebin.ubuntu.com/p/JTqz7RG6V6/>



# 從鍵盤輸入 cin

Example 2-2

【練習】ZeroJudge [a001: 哈囉](#)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      string s;
6      cin >> s;
7      cout << "hello, " << s << "\n";
8      return 0;
9  }
```

10



```
Amy
hello, Amy
Program ended with exit code: 0
```

<https://pastebin.ubuntu.com/p/HpjgkVrWsX/>

# 資料型態 (data type)

資料種類	資料型態	佔用記憶體空間	數值範圍	範例
字元	<b>char</b>	1 Byte	0 ~ 255	'a', 'A', 'b', 'B', '1', '2', '3'
整數	short	2 Bytes	-32768 ~ 32767	
	unsigned short	2 Bytes	0 ~ $2^{16} - 1$	0 ~ 65535
	<b>int</b>	4 Bytes	- $2^{31}$ ~ <b><math>2^{31} - 1</math></b>	-2,147,483,648 ~ 2,147,483,647 <b><math>\sim 2 \times 10^9</math></b>
	unsigned int	4 Bytes	0 ~ $2^{32} - 1$	0 ~ 4,294,967,295
	<b>long long</b>	8 Bytes	- $2^{63}$ ~ <b><math>2^{63} - 1</math></b>	約-9.223372e+18 ~ 9.223372e+18
浮點數	float	4 Bytes	- 3.4e38 ~ 3.4e38	有效位數：7位 (整數+小數)
	<b>double</b>	8 Bytes	- 1.7e308 ~ 1.7e308	有效位數：16位 (整數+小數)
布林值	<b>bool</b>	1 Byte	true 或 false	
字串	<b>string</b>			"I am a student."

# 股神的波克夏A股價位太高「弄壞」那斯達克系統

「股神」巴菲特執掌的波克夏集團，今年來股價持續上漲，導致那斯達克交易所的電腦自4日起就無法顯示股價，連官網也無法呈現相關資訊，因為如今波克夏A股已超出系統的顯示範圍。

Entrepreneur報導，波克夏A股4日收盤價報42萬1,420美元，而且還呈走高態勢，雖然這對巴菲特和投資人而言是個好消息，但再漲下去卻讓那斯達克交易所驚覺將面對股市版的「千禧蟲危機」。

那斯達克電腦系統一開始設定成，能顯示的最高數字為4,294,967,296，原因是該系統的演算法，將股價設定成用32位元（bit）來儲存，以節省記憶體空間，並使軟體運作得更有效率。

另據科技網站Gizmondo分析，在這種緊密的儲存格式下，最高能顯示的數字應是2的32次方再減1，也就是4,294,967,295，但由於那斯達克交易所將股價記錄到小數點後四位，因此能顯示的最高價最後是42萬9,496.7295美元（算法為 $(2^{32} - 1) / 10000$ ）。

➡ 這是什麼資料型別？

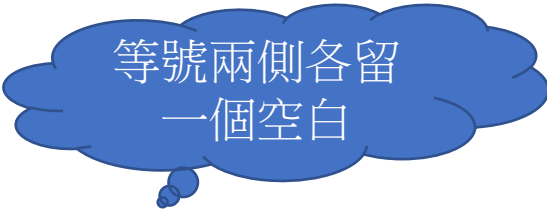
這在過去不成問題，直到波克夏A股近來正式突破這個數字。波克夏A股7日早盤漲抵43萬6,110美元。那斯達克交易所4日就對券商和金融網站，暫停波克夏A股的報價。

至於波克夏掛牌的紐交所表示，就算波克夏或可能漲到這麼高的其他檔股票繼續上漲，該交易所將不受影響。

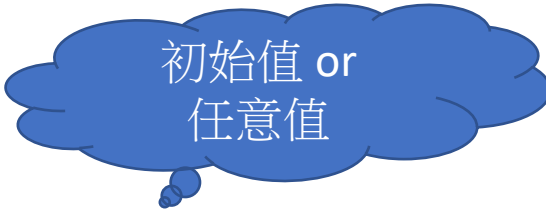
儘管那斯達克交易所如今出問題，巴菲特仍不願將波克夏A股拆股。這也是為何他創造出波克夏B股，而波克夏B股7日盤中報288美元。

# 變數宣告 (variable declaration)

- 變數必須經過宣告才能使用
  - 類別 (type)
  - 名字 (name)
  - 值 (value)
  - 位址 (address)
- 【範例】
  - `int x;`
  - `int x = 5;`
  - `int x, y, z;`
  - `int x = 0, y = 0, z = 0;`
- 變數命名
  - 英文字母、數字、底線 (`_`)



等號兩側各留  
一個空白



初始值 or  
任意值

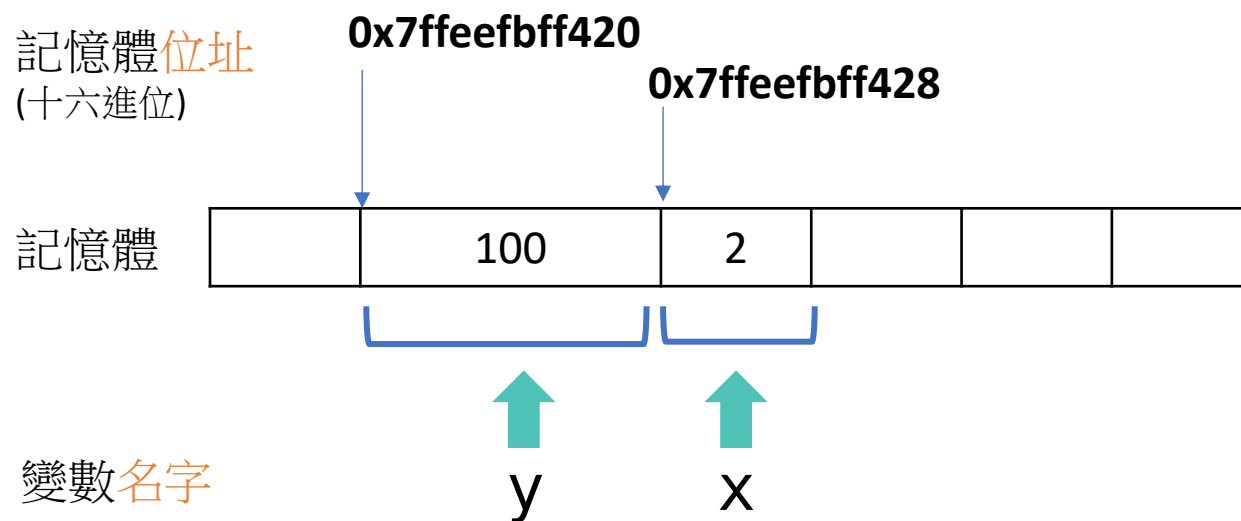
= (指派運算, assignment operator)

不是數學的  
「等號」

宣告變數

```
int x = 3;
```

```
long long y = 100;
```



口訣：把「右邊的運算結果」指派給「左邊的變數」

= (指派運算, assignment operator)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x = 3;
6      long long y = 100;
7      cout << &x << " | " << sizeof(x) << "\n";
8      cout << &y << " | " << sizeof(y) << "\n";
9      return 0;
10 }
11
```

& 取址運算子  
sizeof()

```
0x7ffeebfff428 | 4
0x7ffeebfff420 | 8
Program ended with exit code: 0
```

# 算數運算：+ - \* /

- 加 (+) 減 (-) 乘 (\*) 除 (/)
- 優先序：先乘除、後加減
- 可以利用 括號 ( ) 改變運算順序

# 算數運算

(Quiz)

整數除法沒有小數  
(直接捨去)

```
int x = 13, y = 5;  
cout << x / y << "\n";
```

2

2.6



```
double x = 13, y = 5;  
cout << x / y << "\n";
```

2

2.6

```
int x = -13, y = 5;  
cout << x / y << "\n";
```

-2

-3

## 【練習】 a002: 簡易加法

Example 2-4

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a, b;
6      cin >> a >> b;
7      cout << a + b << "\n";
8      return 0;
9  }
```

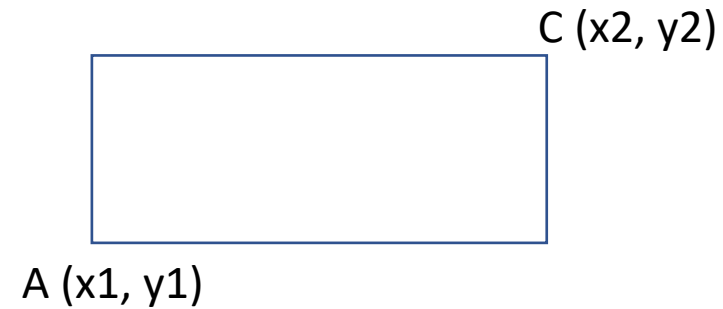
<https://pastebin.ubuntu.com/p/WWBPJTMZQb/>

# 【練習】求十位數及個位數

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x; //二位數
6      cin >> x;
7
8      cout << "十位數 = " << x / 10 << "\n";
9      cout << "個位數 = " << x % 10 << "\n";
10     return 0;
11 }
```

<https://pastebin.ubuntu.com/p/J64rC7WNFb/>

# 【練習】長方形



```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x1, y1;
6      int x2, y2;
7      cin >> x1 >> y1;
8      cin >> x2 >> y2;
9      cout << "長方形面積 = " << (x2 - x1) * (y2 - y1) << "\n";
10     cout << "長方形邊長 = " << 2 * ((x2 - x1) + (y2 - y1)) << "\n";
11     return 0;
12 }
```

<https://pastebin.ubuntu.com/p/H9wNYGQBKF/>

# 【範例】找零錢 (用最少的硬幣數)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x; //要找的零錢，二位數
6      cin >> x;
7      //硬幣面額：50，10，5，1
8      cout << "50元硬幣： " << x / 50 << "\n";
9      x = x - (x / 50) * 50;
10     cout << "10元硬幣： " << x / 10 << "\n";
11     x = x - (x / 10) * 10;
12     cout << "5元硬幣： " << x / 5 << "\n";
13     x = x - (x / 5) * 5;
14     cout << "1元硬幣： " << x << "\n";
15     return 0;
16 }
```

# 溢位 (Overflow)

Example 2-5

(Quiz)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x = 100000;
6      int y = x;
7
8      cout << x * y << "\n";
9
10     cout << x * y << "\n";
11     return 0;
12 }
```

$10^{10}$

$2 \times 10^9$

不知道

# 除以零 (Divide by Zero)

Example 2-6

(Quiz)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x = 10000;
6      int y = 0;
7
8      cout << x / y << "\n";
9      return 0;
10 }
```

≡ Thread 1: EXC\_ARITHMETIC (code=EXC\_I386\_DIV, subcode=0x0)

0

中斷  
程式

$\infty$



# 算數運算子：取餘 (%)

Example 2-7

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      cout << 7 / 3 << "\n";
6      cout << 7 % 3 << "\n";
7      return 0;
8  }
9
```



2

1

Program ended with exit code: 0

商數 (quotient)

餘數 (remainder)

$$a \div b = q * b + r$$

$$a \div b = (a / b) * b + (a \% b)$$

$a, b$  都是整數

# 算數運算子：取餘 (%)

Example 2-13

(Quiz)

- 求餘數 (%) (或稱模運算modulo/mod)
- 判斷奇偶數

```
int x = -13, y = 5;  
cout << x % y << "\n";
```

求負數 (-)

負號跟數字之間不要加空白

2

-3

# 強制資料型別轉換

Example 2-8

轉成浮點數運算

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      cout << 7 / 3 << "\n";
6      cout << (double) 7 / 3 << "\n";
7      return 0;
8  }
9  |
```



2

2.33333

Program ended with exit code: 0

# 控制小數點以下的位數

【範例】 [d051: 糟糕，我發燒了！](#)

```
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main() {
6      double f;
7      cin >> f;
8      cout << fixed << setprecision(3) << (f - 32) * 5 / 9 << "\n";
9      return 0;
10 }
11
```

104

40.000

Program ended with exit code: 0

# 容易犯的錯誤

```
1 // d051: 糟糕，我發燒了！
2 // https://zerojudge.tw/ShowProblem?problemid=d051
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6
7 int main () {
8     int f = 2147483647; //華氏溫度
9
10    //float: 有效位數：7位（整數+小數）
11    //double: 有效位數：16位（整數+小數）
12    float c1 = ((float)f - 32) * 5 / 9;
13    double c2 = ((double)f - 32) * 5 / 9;
14    double c3 = (f - 32) * 5 / 9; //overflow
15    double c4 = (f - 32) / 9 * 5;
16
17    cout << fixed << setprecision(7);
18    cout << c1 << "\n";
19    cout << c2 << "\n";
20    cout << c3 << "\n";
21    cout << c4 << "\n";
22
23    return 0;
24 }
```

```
1193046528.0000000
1193046452.7777777
238609275.0000000
1193046450.0000000
```

# 寫作業時可能遇到的問題

---

### d073. 分組報告 -- 板橋高中教學題

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

int main() {
    cin >> x;
    cout << (x/3) << "\n";
    return 0;
}
```

#0: 10% CE ()

/7806911/code\_7806911.cpp: In function 'int main()':  
/7806911/code\_7806911.cpp:5:12: error: 'x' was not declared in this scope

```
    cin >> x;
           ^
```

## d050. 妳那裡現在幾點了？ -- 板橋高中教學題

```
#include <iostream>
using namespace std;
int main(){
    int h;
    cin>>h;
    cout<<(h+9)%24<<"/n";
    return 0;
}
```

\n 才是換行符號

#0: 10% WA (line:1)

您的答案為： 9/n

正確答案為： 9

#1: 10% WA (line:1)

您的答案為： 10/n

正確答案為： 10



### d050. 妳那裡現在幾點了？ -- 板橋高中教學題

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

int main() {
    int h ;
    cin >> h ;
    cout << abs( h -15 );
    return 0;
}
```

假解  
解題方法不完整

#0: 10% WA (line:1)

您的答案為: 15  
正確答案為: 9

#1: 10% WA (line:1)

您的答案為: 14  
正確答案為: 10