

變數

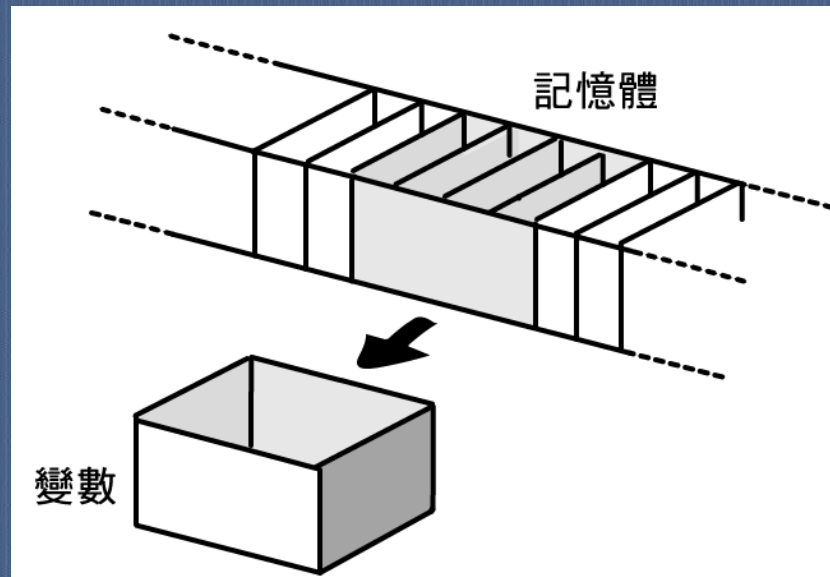
第3章



3-1 變數

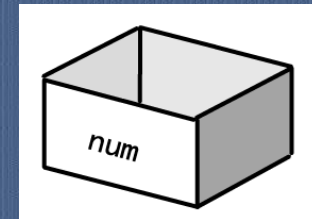
○ 關於變數

電腦在進行處理的過程中，也必須「記住」某些特定的資料，才能進行後續處理（例如進一步將結果顯示在螢幕上），這種讓電腦記住特定資料的功能就稱為變數（variable）。



3-2 識別字

- 在C++程式中，如果要使用某變數，就必須先針對該變數進行下列2項設定工作：
 - 指定變數的「名稱」
 - 指定變數的「型態」
- 識別字專門用來做為變數的「名稱」。
- 建立新的識別字時仍然要遵守下列的規則：
 - 一般而言，我們會使用英文字母、阿拉伯數字、底線（_）、「\$」等符號來作為識別字名稱。無法包含特殊記號。
 - 視環境而定，有些環境最長不得超過31個字元。
 - C++已保留起來的關鍵字（keyword）不能做為識別字名稱。常見的關鍵字如return。
 - 識別字名稱不能用阿拉伯數字做為開頭。
 - 英文字母的大小寫有區別。



3-3 變數的資料型態

- 變數可以用來「存放」特定的文字或數值，這些文字或數值一般稱為資料型態（data type）。
- C++的變數可以使用的基本資料型態：
 - 邏輯型--boolean
 - 文字型--char、unsigned char
 - 整數型--short int、int、
unsigned int、long int
 - 浮點數型--float、double、
long double

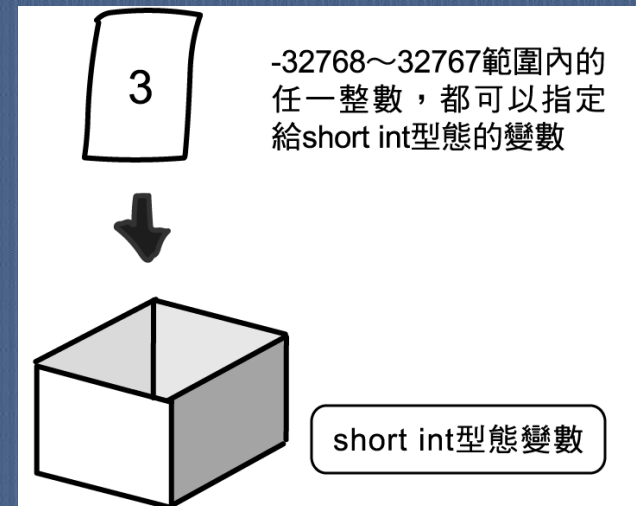


表 3-1：C++ 的變數可以使用的基本資料型態

種類	名稱	檔案大小	資料的範圍
邏輯型	boolean	1 byte	true 或 false
文字型	char	1 byte	1 個英數字 -128 ~ 127
	unsigned char	1 byte	1 個英數字（無符號） 0 ~ 255
整數型	short int	2 byte	整數 -32768 ~ 32767
	unsigned short int	2 byte	整數（無符號） 0 ~ 65535
	int	4 byte	整數 -2147483648 ~ 2147483647
	unsigned int	4 byte	整數（無符號） 0 ~ 4294967295
	long int	4 byte	長整數 -2147483648 ~ 2147483647
	unsigned long int	4 byte	長整數（無符號） 0 ~ 4294967295
浮點數型	float	4 byte	單精度浮點數 3.4E-38 ~ 3.4E+38
	double	8 byte	雙倍精度浮點數 1.7E-308 ~ 1.7E+308
	long double	8 byte	四倍精度浮點數 1.7E-308 ~ 1.7E+308



bit與byte

- 電腦內部的資料是由「0」、「1」的二進制數值所組成，每一個二進制數值就是一個bit(位元)，而八個bit就可以組成一個byte(位元組)。
- 位元組中的資料如下所示：

00101110

而這八個位元的排列組合可以有 $2^8=256$ 種，因此，一個位元組可以表示256個值，兩個位元組將能表示 $2^{16}=65536$ 個值。

- 每個二進制數值的最左邊的位數(第一個bit)，將影響該數值的正負。
- 各種資料型態的值的範圍，就是取決於它的位元大小和第一個位元。



3-4 宣告變數

- 宣告變數的語法：
變數的資料型態 識別字；

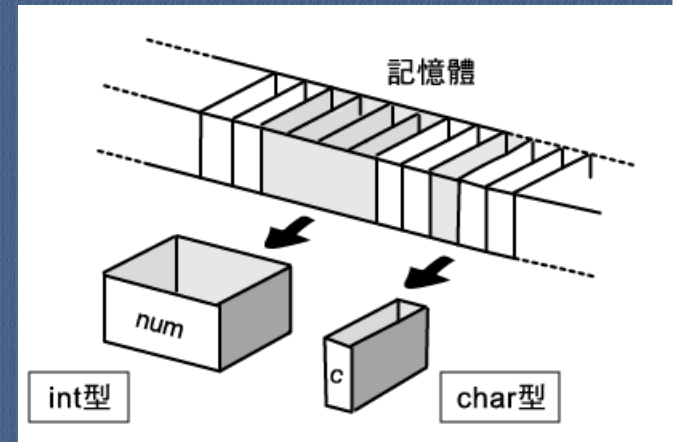
- 範例：

`int num;` ← **int**型態的變數 (變數名稱是**num**)

`char c;` ← **char**型態的變數 (變數名稱是**c**)

`double db, dd;`

← **double**型態的變數
(變數名稱是**db**、**dd**)



3-5 開始使用變數

- 指定「數值」或「其他資料」給某變數的語法為：
變數名稱 = 特定資料;

例如：`num = 3;`

- 等號(=)的意思並非是「等於」，而是「指定給」。

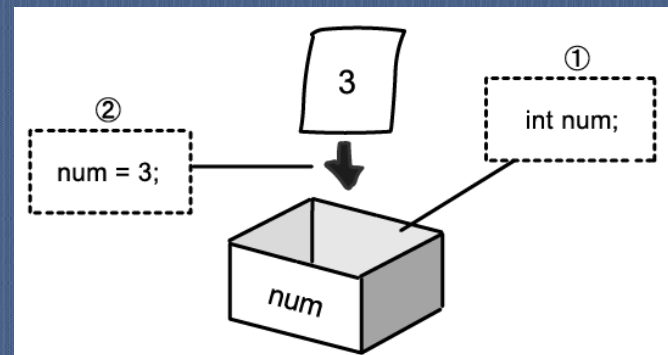
- 範例：

```
#include <iostream>
using namespace std;
int main()
{
    int num;
    num = 3;
    cout << "變數num的值是" << num << "。 \n";
    return 0;
}
```

宣告變數num

將3指定給變數num

輸出變數num



- 輸出變數所代表的值：

```
Cout << "變數num的值是" << num << "。 \n";
```

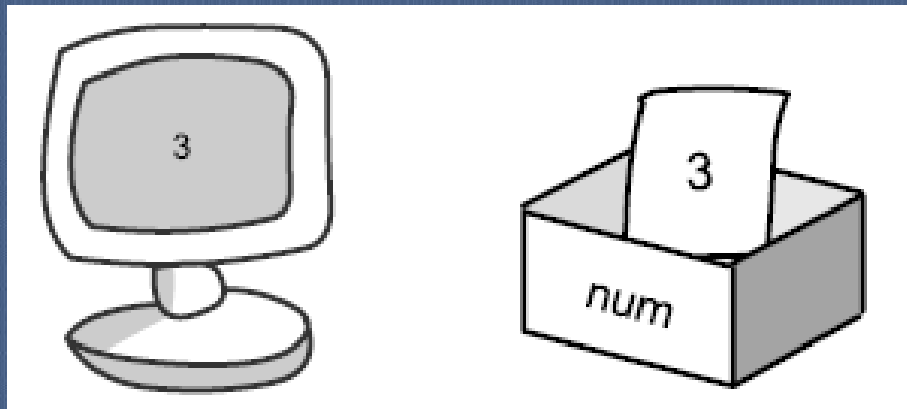
輸出結果：變數num的值是3。

- 變數的初始化語法為：

資料型態 識別字 = 特定資料;

例如：`int num = 3;`

- 初始化(initialization)就是在宣告變數的同時指定值。



○ 改變變數的值

...

```
num = 3;
```

```
cout << "變數num的值是" << num << "。 \n";
```

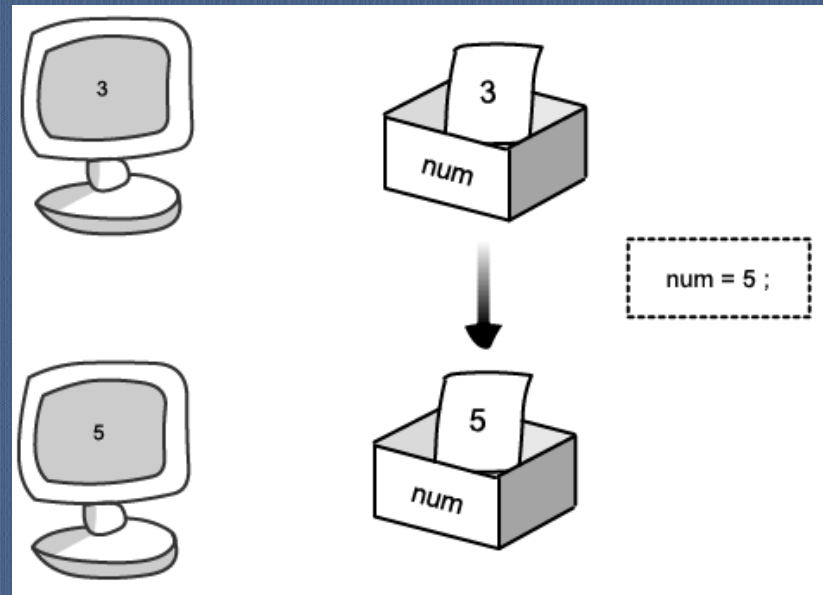
```
num = 5;
```

```
cout << "變數num更新後的值是" << num << "。 \n";
```

○ 執行結果：

變數num的值是3。

變數num更新後的值是5。



Sample2.cpp ▶ 改變變數的值

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

int main()
{
    int num = 3;

    cout << "變數 num 的值是 " << num << " 。 \n";

    num = 5;

    cout << "更新變數 num 的值。 \n";
    cout << "變數 num 更新後的值是 " << num << " 。 \n";

    return 0;
}
```

①輸出變數的值

②重新指定變數的值

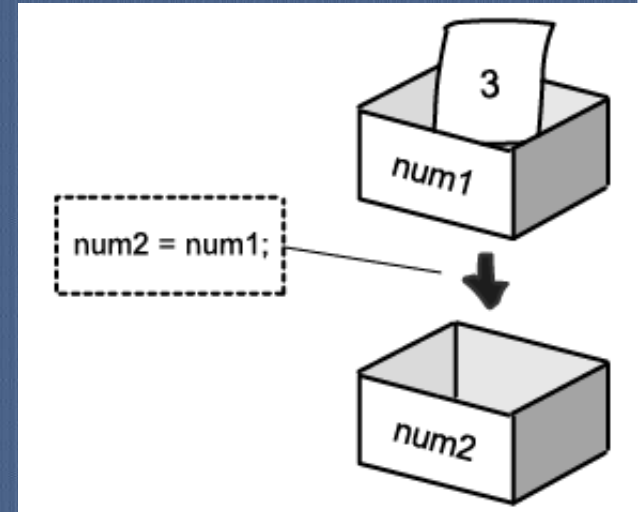
③輸出新的變數值

○ 變數值再指定

- 指定各種資料給變數時，等號（=）右邊並不是只能使用3或5這種數字型的資料，也可以使用變數。
- 例如：

```
...  
int num1, num2;  
num1 = 3;  
num2 = num1;  
cout << "變數num2的值是" << num2 << "。 \n";
```

↑
結果num2的值會是3，
這表示既有的變數值也
可以指定給新的變數。



Sample3.cpp ▶ 變數值再指定

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

int main()
{
    int num1, num2;

    num1 = 3;

    cout << " 變數 num1 的值是 " << num1 << " 。 \n";

    num2 = num1; ●———— 變數 num1 的值再指定給變數 num2

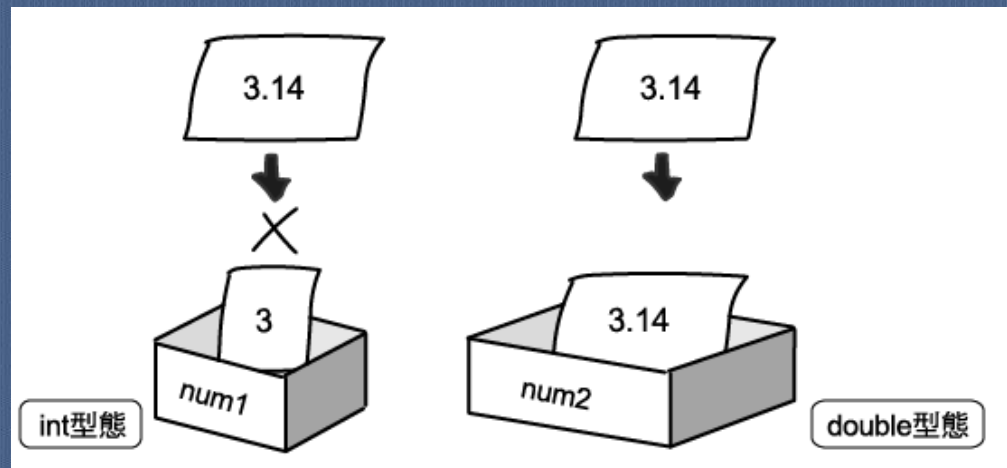
    cout << " 把變數 num1 指定到變數 num2 之中。 \n";
    cout << " 變數 num2 的值是 " << num2 << " 。 \n";

    return 0;
}
```

○ 指定變數值時的注意事項

- 變數會根據宣告的型態來決定可以記憶的數值種類。如果將帶有小數的數值指定給整數值的變數，變數的資料型態就會自動被轉換。
- 範例：

```
int num1;  
double num2;  
num1 = 3.14;  
num2 = 3.14;  
cout << "變數num1的值是" << num1 << "。 \n";  
cout << "變數num2的值是" << num2 << "。 \n";
```



Sample4.cpp ▶ 指定變數的值

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

```
int main()
{
```

```
    int num1;
```

```
    double num2;
```

```
    num1 = 3.14;
```

```
    num2 = 3.14;
```

指定給 int 型態的變數

指定給 double 型態的變數

```
    cout << " 變數 num1 的值是 " << num1 << " 。 \n";
```

```
    cout << " 變數 num2 的值是 " << num2 << " 。 \n";
```

```
    return 0;
```

```
}
```

- 留意變數的宣告位置：

原則上，變數都是在main()函數的程式區塊（block）內宣告，如下所示：

```
int main()  
{  
    ...  
}
```

- C++可以在函數內的任何地方宣告變數，而C語言則不行。



3-6 從鍵盤輸入變數值

- 若要讓使用者從鍵盤輸入變數值，只要使用`cin >>`這樣的語法便可讀入從鍵盤輸入的值，如下所示：

```
#include <iostream>
using namespace std;
int main()
{
    宣告變數;
    cin >> 變數;
    ...
}
```

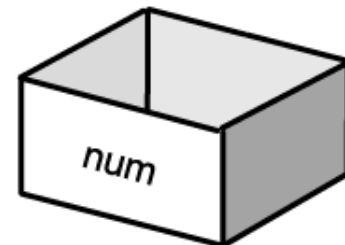
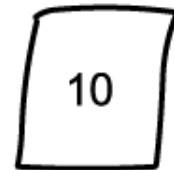
↑
使用者從鍵盤
輸入的變數

當程式處理到`cin>>...`，
電腦就會停下來等待使
用者從鍵盤輸入



`cin >> num;`

輸入完畢後按下Enter
鍵，變數就會被載入



Sample5.cpp ▶ 輸入數值

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

```
int main()
{
```

```
    int num = 0;
```

```
    cout << " 請輸入整數 。 \n"; ●
```

顯示請使用者利用鍵盤輸入的訊息

```
    cin >> num; ●
```

從鍵盤輸入的數值被讀入為變數 num

```
    cout << " 已輸入 " << num >> " 。 \n";
```

顯示已輸入的數值

```
    return 0;
```

```
}
```

○ 輸入2個以上的數值

- 可以利用>>符號緊接著再輸入下一個數字，來連續輸入2個以上的數值。
- 例如：

...

```
int main()
{
    cout << "請輸入兩個整數。\\n ";
    cin >> num1 >> num2;
    cout << "首先輸入的是" << num1 << "。\\n ";
    cout << "接著輸入的是" << num1 << "。\\n ";
    return 0;
}
```

○ 標準輸入與標準輸出

- 正如cout所代表的是「標準輸出」一樣；cin所指的就是標準輸入（standard input）這個概念。所謂的「標準輸入」通常就是電腦的鍵盤。



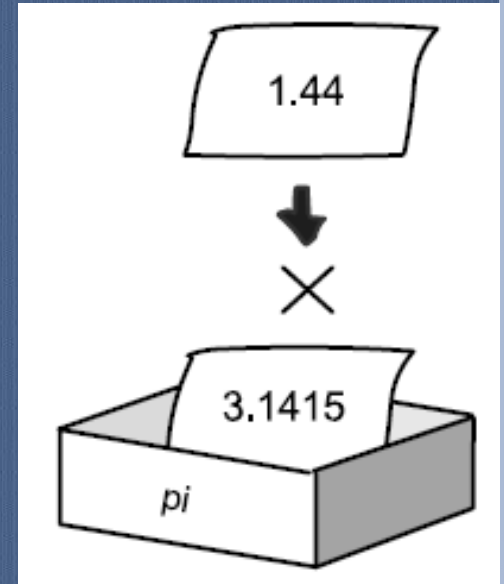
3-7 常數

○ 宣告和指定const

- 例如：`const double pi = 3.141592;`
- 使用**const**就可以宣告常數。
- 因為宣告為常數的pi將無法在之後才指定值，所以必須在初始化的時候指定其中的值。

○ 關於const的語法：

- `const` 資料型態 識別字 = 特定資料;



Sample7.cpp ▶ 利用常數

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

int main()
{
    const double pi = 3.1415; ● ①宣告和指定 const 並將 pi 初始化

    cout << "圓周率的值為 " << pi << " 。 \n";
    cout << "圓周率的值無法變更。 \n";

    // 無法以指定的方式作變更
    //pi = 1.44; ● ②無法變更 pi 的值

    return 0;
}
```

綜合整理

- 本章學習過的內容與重點：
 - 變數就像一個盒子一樣，可以用來容納特定的值。
 - 宣告變數的時候，必須清楚地指定變數的名稱和資料型態。
 - 使用識別字做為變數的「名稱」。
 - 將某個值「指定」給變數時，必須使用等號(=)。
 - 宣告變數的同時，可以順便完成變數初始化的工作，也就是直接把值指定給變數。
 - 指定新的值給變數時，變數舊有的值會被新的值所取代（您可以想像成變數這個盒子已經裝了新值）。
 - C++程式可以接受使用者從鍵盤輸入資料。
 - 已經宣告和指定為const的變數，就是「常數」，不可以變更其值。

