



C++

變數與指定運算子



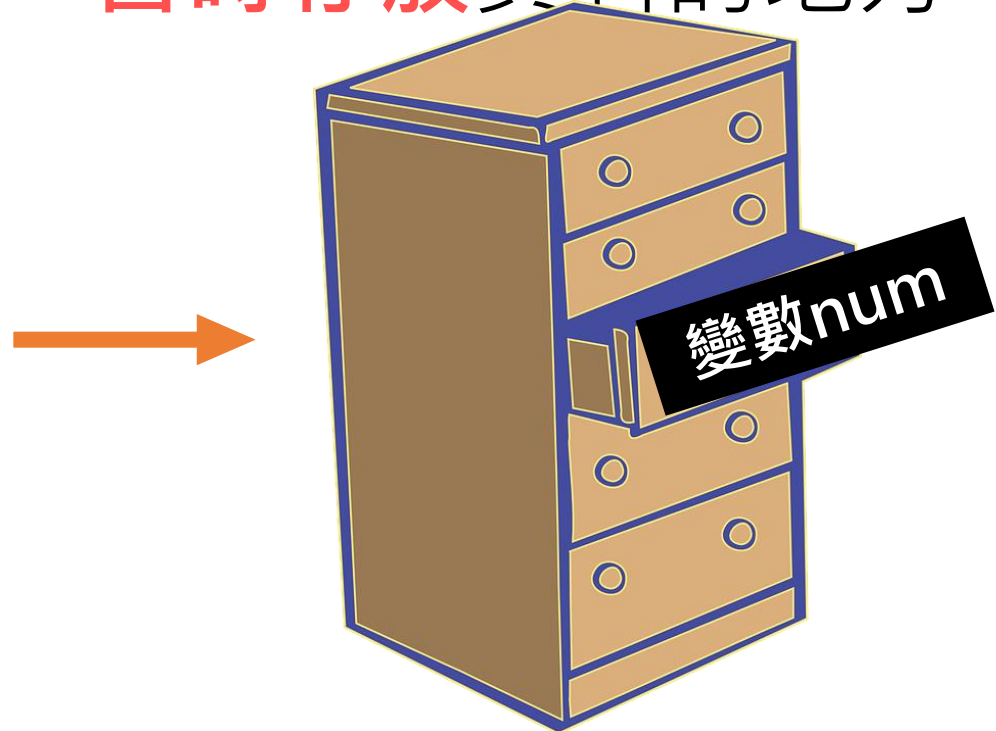
練習

寫一個程式，在**記憶體**放置一個**整數**，然後印出來。
例如：在記憶體放置3。



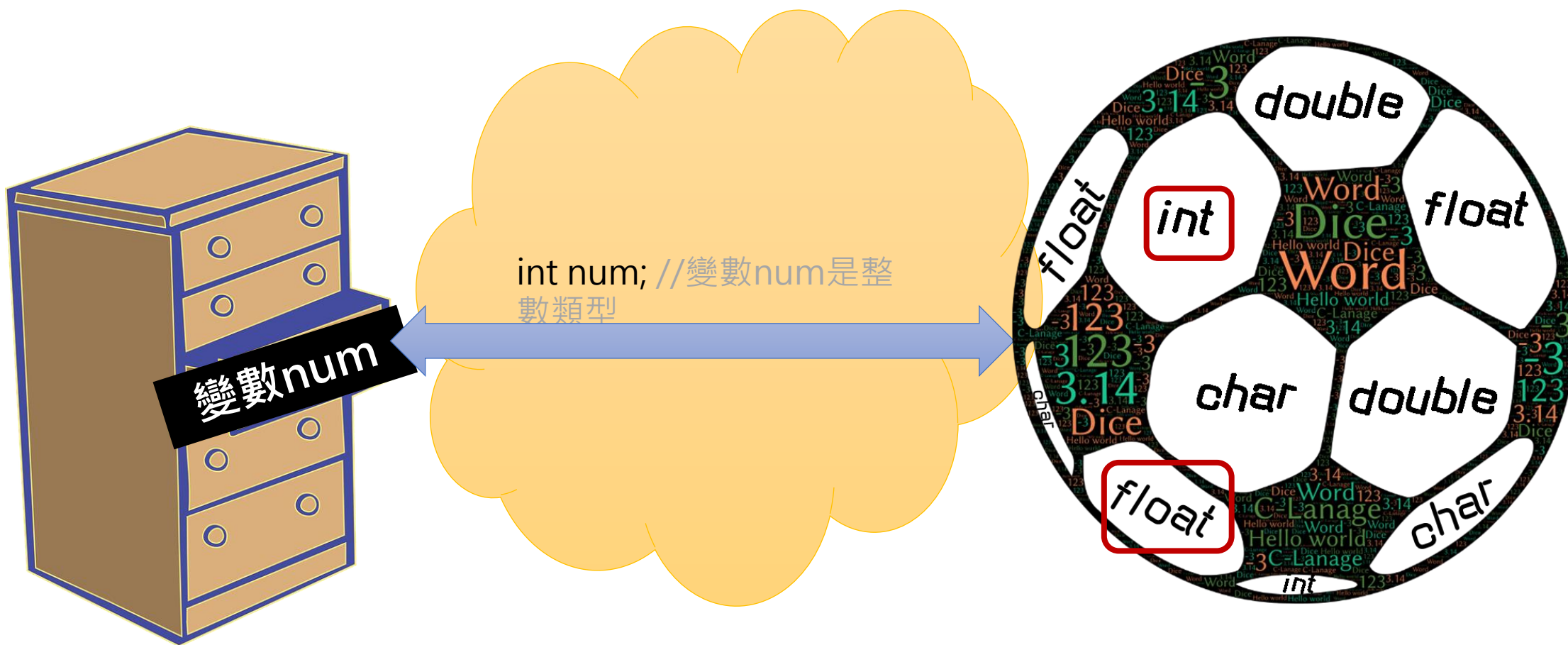
為什麼放置記憶體？

記憶體是電腦在運作過程中，暫時存放資料的地方。



- 記憶體猶如有編號的櫃子。
- 使用記憶體前，須先告知。

變數是屬於甚麼類型，必須事先聲明



先在記憶體放置一個整數，然後印出來

使用記憶體前，須先告知：**變數宣告**

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

int main(){
    int num;
    num = 3;
    cout<<num<<endl
    return 0;
}
```

使用記憶體前，須先告知

指定值給整數
指定運算子 =



C++

變數與指定運算子

變數定義的屬性(以int num說明)

資料型態：

指的是**數**或者**字**。數先簡單區分整數與小數(浮點數)。

int 代表的是整數 (integer) ;
float 代表的是**浮點數** (float) 。

變數名稱：

一個變數必須有名字，例如: num 。

1. 由英文字母(大小寫被視為不同)、數字以及底線組成。
2. 不能用數字開頭。
3. 組成: 限一個字詞。

值:

被賦予的值，可以使用**指定運算子**或者是**輸入函式**改變其值。

地址:

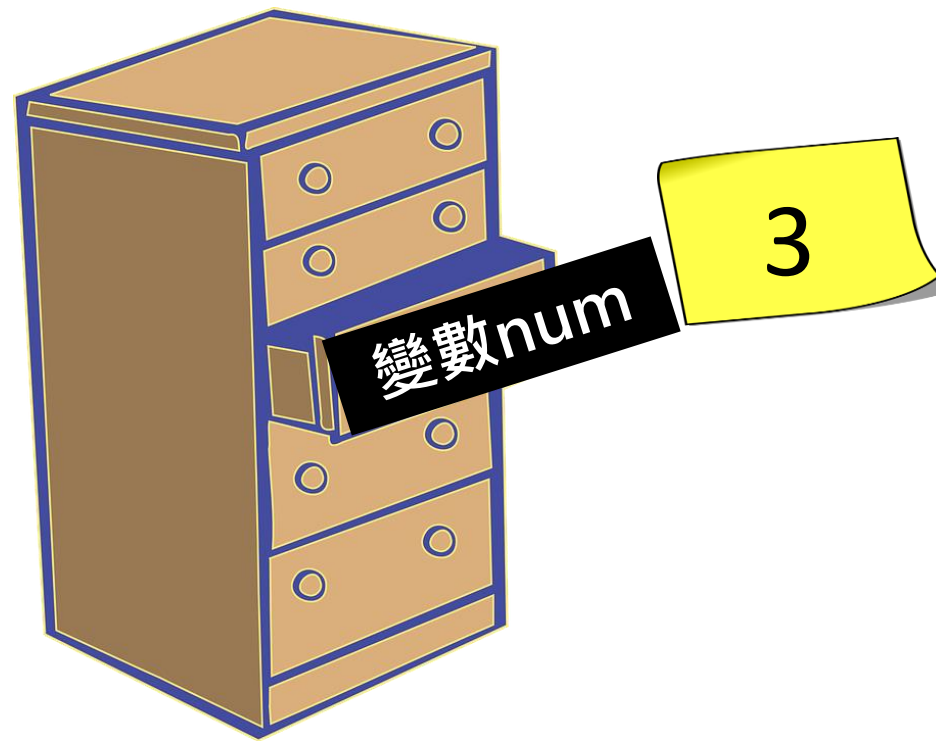
一個變數相對應於電腦記憶體中的位址。

指定運算子 = : 指定一個值或字元給變數

```
int num;  
num = 3;
```

數學符號等於 = ，在許多的程式語言中是指定運算值，可以讀成 **得到** 。

變數num的值一開始設為3。如果未設定初始值，則變數的初始值可能是任意數。



輸出 - 使用 cout

- 必須包含標頭檔#include <iostream>
(本教材輸出以cout為主。)

| 程式碼 | 輸出結果 |
|----------------------|-------------|
| cout<< "num" <<endl; | num |
| cout<<num<<endl; | 3(以num=3為例) |



C ++ +

延伸的概念

Extended concept

1：設定兩個變數並印出

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

int main(){
    int a;
    int b;
    a = 11;
    b = 12;
    cout<<a<<endl;
    cout<<b<<endl;
}
```

或 int a,b;

2 : 除了印常數外，也可以加入其他字詞

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

int main(){
    int cat;
    cat=3;
    cout<<"There are ";
    cout<<cat;
    cout<<" cats."<<endl;
}
```

或 `cout<<"There are "<<cat<<" cats."<<endl;`

3：浮點數如何處理？

(資料型態不只這些，後面補充說明)

| 資料型態 | | |
|------|-------|------|
| 數 | 表示方法 | 格式符號 |
| 整數 | int | %d |
| 浮點數 | float | %f |

浮點數的輸出

(資料型態不只這些，後面補充說明)

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     float a;
6     a=11.1100;
7     cout<<a;
8     return 0;
9 }
```

輸出： 11.11

4：如何顯示小數點後指定的位數？

(例如: 小數點後第5位) -

```
1 #include<iostream>
2 #include<iomanip>
3 using namespace std;
4 int main()
5 {
6     float a;
7     a=11.1100;
8     cout<<fixed<<setprecision(5)<<a;
9     return 0;
10 }
```

在 cout 中，沒有格式符號，但是我們可以使用 `setprecision()` 函式來實現這個需求。想要使用 `setprecision()` 函式，必須包含標頭檔 `#include <iomanip>`。

輸出：

11.11000

資料型態

- 在程式執行的過程中，為了能夠方便有效的執行程式的各種運算，需要的資料型態有：整數、浮點數與文字等
- 電腦內部使用0與1(位元)來表示資料，並且儲存於記憶單元
- 任何資料都可以轉換成一連串位元的組合

常見的資料型態

| 資料型態 | 名稱 | 大小(bytes) | 範圍 | 格式符號 |
|--------|---------------|-----------------------------|--|------|
| 整數 | int | 2bytes (16位元系統) 或 4bytes | -32768至32767或 -2147483648至2147483647 | %d |
| 長整數 | long int | 4bytes 或 8bytes | -2147483648至2147483647 或 -9223372036854775808至 9223372036854775807 | %ld |
| 長整數 | long long int | 8bytes | -9223372036854775808至 9223372036854775807 | %lld |
| 單精度浮點數 | float | 4bytes | 2.939×10^{-38} 至 $3.403 \times 10^{+38}$ (7 sf) | %f |
| 雙精度浮點數 | double | 8bytes | 5.563×10^{-309} 至 $1.798 \times 10^{+308}$ (15 sf) | %lf |
| 字元 | char | 1bytes | 通常為-128至127或0至255 | %c |