Chapter 0： 基本資料型態與變數

本章將介紹 C/C++程式語言之基本資料型態，其主要用來宣告產生適當型態的變數，以儲存資料。基本資料型態的種類，有整數、浮點數、字元、布林值等。

0.1 整數(Integer)

說明：

整數就是不含小數部分的數值，如 1、50000、-99。在 C/C++程式語言中，整數型態依據使用記憶體空間大小的不同，分為短整數與長整數兩種。

關鍵字	記憶體空間	數值範圍
short 或 short int	2 bytes	-32768 ~ 32768
int， long 或 long int	4 bytes	-2147483648~2147483647

程式範例：

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

void main()
{
    short age;
    int population;

    age = 30;
    population = 21000000;

    cout<<sizeof(short)<<endl;
    cout<<sizeof(age)<<endl;
    cout<<sizeof(int)<<endl;
    cout<<sizeof(population)<<endl;
    cout<<sizeof(long)<<endl;
}
```

注意要點：

1. C/C++中的任何變數宣告與程式命令，均需用分號(;)符號做結尾。
2. 使用 sizeof( ) 可得任何資料型態或變數所佔記憶體空間的大小，單位為 byte。
3. 程式中之 age 與 population 為變數，可用來儲存符合其所宣告型態與數值範圍的任何資料值。
4. 變數的命名有其規則，將於稍後介紹。
5. 輸入變數之值如與變數宣告的型態不符，編譯器將自動做型別轉換，轉換的過程可能失去資料。

0.2 浮點數(Floating Point)

說明:

程式語言中，具有小數的數字稱為浮點數，如 3.14159、-66.66、20.0。在 C/C++程式語言中，浮點數型態依據使用記憶體空間大小的不同，分為 float 與 double 兩種。

關鍵字	記憶體空間	數值範圍
float	4 bytes	(+/-) 3.4*10^-38 ~ 3.4*10^38
double	8 bytes	(+/-) 1.7*10^-308 ~ 1.7*10^308

程式範例：

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

void main()
{
    float pi1;
    double pi2;

    pi1 = 3.14159;
    pi2 = 314159e-5;
}
```

注意要點：

1. 浮點數包含了範圍內的所有整數，而整數只能表達整數。
2. 設定浮點數數值時，可使用一般十進位表示法，或科學指數表示法。

0.3 字元(Character)

說明:

字元資料型態是為了存放字母及數字，如 a，A，1，2 等。對於字母與文字，系統一般是使用整數數字代碼的方式來儲存，故其可看做為另一種整數型態，惟一變數宣告為字元，其輸出將為代碼所代表的字元，而非記憶體所存的代碼。

代碼使用 ASCII 字元集，制定 128 個字元，做為構成單字或句子的單位。輸入字元至變數中，需用單引號[‘]將字元包起來，以供系統辨識資料為字元型態。

關鍵字	記憶體空間	數值範圍
char	1 bytes	-127 ~ 127

程式範例：

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

void main()
{
    char let1 = '1';
    char let2 = 50;

    cout<<let1<<endl;
    cout<<let2<<endl;
    cout<<(int)let1<<endl;
    cout<<(int)let2<<endl;
}
```

注意要點：

1. 設定字元變數的值，亦可直接使用代碼，代碼不需置於’‘內。
2. 使用(int)命令，可將字元變數強制轉換為整數型態，而得知某字元的 ASCII 代碼。

0.4 布林(Boolean)

說明:

布林為邏輯變數，其值只有真(true)與假(false)兩種。

關鍵字	記憶體空間	數值範圍
bool	1 bytes	0 與 1

程式範例：

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

void main()
{
    bool test1 = true;
    bool test2 = false;
    bool test3 = 10;

    cout<<test1<<endl;
    cout<<test2<<endl;
    cout<<test3<<endl;
}
```

注意要點：

1. MS Visual C++編譯環境中，true 與 false 為關鍵字，true 代表值為 1，false 代表值為 0。
2. 於 C/C++程式語言中，在邏輯判斷的情況下，0 代表 false，其他所有整數均代表 true。對於布林變數，將依此規則自動型別轉換。

0.5 unsigned 資料型態

說明:

以上的資料型態，均包含有正數與負數，且正負數各佔了一半的記憶體空間，在某些情況下，一變數的值不會有負數，如年齡、人口數，這時我們可用 unsigned 關鍵字，將負數的空間，全部轉為正數所使用，以增加最大值的上限。

關鍵字	記憶體空間	數值範圍
unsigned char	1 bytes	0 ~ 255
unsigned int	4 bytes	0 ~ 4294976295
unsigned short	2 bytes	0 ~ 65535

程式範例：

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

void main()
{
    unsigned int population;
    population = 21000000;
    cout<<population;
}
```

注意要點：

1. MS Visual C++編譯環境中，如輸入的 unsigned 變數值為負數，並不會自動轉換為 0，而是不明的數值，需注意。
2. 浮點數並沒有支援 unsigned 之宣告。

0.6 變數(Variable)

說明:

變數為程式中儲存資料的單位，其可以隨時被宣告產生，並可隨程式的進行而變更其所包含的值，不論是透過程式的運算，或是直接重新指定其值。

格式：

資料型態　變數名稱 [= 初始值];

註： [　] 中的可省略。

程式範例：

```
int population;                 //變數宣告
population = 21000000;          //變數設定初值

int age = 30;                   //變數宣告 + 設定初值
int area = 5*5*3.14159;
```

變數名稱規則：

變數的名稱大小寫有別，可任意自訂其名稱，惟必須遵守下列規則：

1. 開頭字，可為以下三種字元之一：
   - 大寫字母
   - 小寫字母
   - 底線(_)

2.其他字元，可為以下四種字元之一：
   - 大寫字母
   - 小寫字母
   - 底線(_)
   - 數字

此外，變數的長度不可超過 255 個字元，亦不能使用保留之關鍵字做為變數名稱。

注意要點：

1. 已宣告過的變數名稱，不能再重複宣告。
2. 變數的大小有別(case sensitive)，例： X 不等於 x。
3. 變數要先宣告才能使用。

0.7 常數(Constant)

說明:

常數與變數不同之處，在於常數一經宣告設定，其值即為固定，不能也不會在程式執行的過程中改變其值。

格式：

宣告設定常數的做法有二，其一(C++ style)，使用 const 關鍵字：

```
const 資料型態   常數名稱 = 初始值;
```

其二(C style)，使用#define 指令(前處理，通常宣告於程式前幾行)：

```
#define 常數名稱 初始值
```

程式範例：

```
#define   PI   3.14159

void main()
{
      const double pi = 3.14159;
}
```

注意要點：

1. 宣告常數時，同時必須設定其值。
2. 常數之命名規則，與變數相同。
3. 程式中，嘗試改變常數的值為不合法，編譯程式時會產生錯誤。