

基本資料型別

C



C 基本資料型別

圖 C.1 列出了 C++ 的基本資料型別。C++ 標準文件 (The C++ Standard Document) 並沒有提供在記憶體中儲存具有這些型別的變數，所需要的精確位元組數量。不過，C++ 標準文件卻有指出各基本型別的記憶體空間需求之間，彼此的關係。將基本型別依據其記憶體空間的需求，由小而大加以排列，則有號整數型別分別是 `signed char`、`short int`、`int` 和 `long int`。這代表著，`short int` 的記憶體空間至少必須和 `signed char` 相同，`int` 的記憶體空間也至少必須和 `short int` 相同；同樣地，`long int` 的記憶體空間至少必須和 `int` 相同。每個有號整數型別都會有一個相對應的無號整數型別，而且兩者之間具有相同的記憶體空間需求。無號型別並無法表示負數，但是，與對應的有號型別相比較，它們所表示出的正數則是其兩倍。依據記憶體空間的需求，由小而大加以排列，浮點數型別有 `float`、`double` 和 `long double`。就像整數型別一樣，`double` 的記憶體空間至少必須和 `float` 相同，而且 `long double` 的記憶體空間至少必須和 `double` 相同。

整數型別	浮點數型別
<code>bool</code>	<code>float</code>
<code>char</code>	<code>double</code>
<code>signed char</code>	<code>long double</code>
<code>unsigned char</code>	
<code>short int</code>	
<code>unsigned short int</code>	
<code>int</code>	
<code>unsigned int</code>	
<code>long int</code>	
<code>unsigned long int</code>	
<code>wchar_t</code>	

圖 C.1 C++基本資料型別

基本型別的數值的精確大小和範圍，與系統的實作方式有關。標頭檔 `<climits>` (用於整數型別) 和 `<cmath>` (用於浮點數型別)，會指明在你的電腦系統上所支援的數值範圍。

一個型別所支援的數值範圍，會與該型別使用的位元組數量有關。舉例來說，讓我們考慮某個使用四個位元組 (32 位元) `int` 的系統。對於有號 `int` 型別而言，其非負數

的數值範圍是 0 到 2,147,483,647 ($2^{31}-1$)。其負數的數值範圍是 -1 到 -2,147,483,648 (-2^{31})。加起來一共有 2^{32} 個可能的數值。另一方面，在相同系統上，`unsigned int` 也是使用相同的位元組數量，來代表資料，但是它無法表示任何負數。結果造成其數值範圍是從 0 到 4,294,967,295 ($2^{32}-1$)。此外，在這個相同系統上，`short int` 表示資料時，不能超過 32 個位元，而 `long int` 則必須使用至少 32 個位元。

C++ 也有提供資料型別 `bool`，具有這種型別的變數只能儲存數值 `true` 和 `false`。

C-4 C++程式設計藝術(第六版)(國際版)