

ACPS C++ 教學講義 08

C++ 字串處理

C++ 提供了以下兩種類型的字串表示形式：

- C 風格字串
- C++ 引入的 `string` 類別類型

C++ 風格字串

C 風格字串起源於 C 語言，並在 C++ 中繼續得到支持。

字串實際上是使用 `null` 字串 `'\0'` 終止的一維字串陣列。因此，一個以 `null` 結尾的字元陣列，可視為字串。

下面的宣告和初始化創建了一個 "Hello" 字串。由於在陣列的末尾儲存了空字串，所以字串陣列的大小比單詞 "Hello" 的字串數多一個 `char`。

```
greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

依據陣列初始化規則，您可以把上面的語句寫成以下語句：

```
char greeting[] = "Hello";
```

以下是 C/C++ 中定義的字串的記憶體表示：

Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456

其實，您不需要把 `null` 字串放在字串常數的末尾。C++ 編譯器會在初始化陣列時，自動把 `'\0'` 放在字串的末尾。讓我們嘗試輸出上面的字串

```
#include <iostream>

using namespace std;

int main ()
{
    char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

    cout << "Greeting message: ";
    cout << greeting << endl;

    return 0;
}
```

當上面的程式碼被編譯和執行時，它會產生下列結果：

```
Greeting message: Hello
```

C++ 中有大量的函數用來操作以 **null** 結尾的字串：

函數次序	函數說明
1	<code>strcpy(s1, s2);</code> 複製字串 <code>s2</code> 到字串 <code>s1</code> 。
2	<code>strcat(s1, s2);</code> 連接字串 <code>s2</code> 到字串 <code>s1</code> 的末尾。
3	<code>strlen(s1);</code> 返回字串 <code>s1</code> 的長度。

4	<code>strcmp(s1, s2);</code> 如果 s1 和 s2 是相同的，則返回 0； 如果 s1<s2 則返回小於 0； 如果 s1>s2 則返回大於 0。
5	<code>strchr(s1, ch);</code> 返回一個指標，指向字串 s1 中字串 ch 的第一次出現的位置。
6	<code>strstr(s1, s2);</code> 返回一個指標，指向字串 s1 中字串 s2 的第一次出現的位置。

下面的實例使用了上述的一些函數：

```
#include <iostream>
#include <cstring>

using namespace std;

int main ()
{
    char str1[10] = "Hello";
    char str2[10] = "world";
    char str3[10];
    int len ;

    // 複製 str1 到 str3
    strcpy( str3, str1);
    cout << "strcpy( str3, str1) : " << str3 << endl;

    // 連接 str1 和 str2
    strcat( str1, str2);
    cout << "strcat( str1, str2): " << str1 << endl;
```

```
// 連接後，str1 的總長度
len = strlen(str1);
cout << "strlen(str1) : " << len << endl;

return 0;
}
```

當上面的程式碼被編譯和執行時，它會產生下列結果：

```
strcpy( str3, str1) : Hello
strcat( str1, str2): HelloWorld
strlen(str1) : 10
```

C++ 中的 String 類別

C++ 標準庫提供了 **string** 類別類型，支持上述所有的操作，另外還增加了其他更多的功能。我們將學習 C++ 標準庫中的這個類別。

現在讓我們先來看看下面這個實例：

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

int main ()
{
    string str1 = "Hello";
    string str2 = "world";
    string str3;
    int len ;

    // 複製 str1 到 str3
    str3 = str1;
```

```
cout << "str3 : " << str3 << endl;

// 連接 str1 和 str2
str3 = str1 + str2;
cout << "str1 + str2 : " << str3 << endl;

// 連接後，str3 的總長度
len = str3.size();
cout << "str3.size() : " << len << endl;

return 0;
}
```

當上面的程式碼被編譯和執行時，它會產生下列結果：

```
str3 : Hello
str1 + str2 : HelloWorld
str3.size() : 10
```