

字元&字串介紹

Char & String Type for C language

字元

資料型態	全稱	佔空間大小	輸出符號
Char 字元	Character	ASCII(8bits)	%c

1、Char宣告及使用程式範例：

```
1  #include <stdio.h>
2
3  int main() {
4
5      // 宣告一個名為c的字元型態, 並且初始化保存了a.
6      char c = 'a';
7      printf("show: %c\n", c); //output-> show: a
8
9      // 我們也可以通過指定%d輸出其整數型態
10     printf("show: %d\n", c); //output-> show: 97
11
12     return 0;
13 }
```

*必須要注意：賦值給**char**時，要使用英文單引號。

*注：想要顯示字元，必須使用**%c**來指定輸出格式，**%d**會顯示字元背後原本所保存的整數數值。

2、字元 a 在ASCII中所保存的數值位 97

```
5      // 宣告一個名 (int)97 型態, 並且初始化保存了a.
6      char c = 'a';
7      printf("show: %c\n", c); //output-> show: a
```

3、透過對字元的運算來實現移動效果 a 往後移動5位等於 f

```
12 //以知char背後保存的是ASCII數字,透過直接運算來進行移位
13 c = c + 5; // a 加 5 相當於往後移動5位
14 printf("show: %c, %d\n", c, c); //output-> show: f, 102
15
16 //14行 %d輸出 102 相當於 (97+5)
```

字串

資料型態	全稱	佔空間大小	輸出符號
String 字串	String	宣告時指定	%s

說明：

C語言中並沒有特別為字串建立一套String型態作為保留字，而是通過一組字元並在末端加上\0來表示字串。因此字串相當於一組字元的集合，並沒有一種型態叫做String（此處僅介紹C語言情況，其他程式語言有可能已經建立了String型態供直接宣告使用）

1、String宣告程式範例：

```
1 #include <stdio.h>
2
3 int main() {
4
5     //宣告一組長度為10的字元陣列(即字串)
6     //長度為10意味著最多只能保存 9個字元,第10個字元會保存 \0 以表示此字元陣列是字串
7     char str[10] = "Hello"; //雖然宣告了10這麼大的空間,但是 Hello只使用了 5個字元加上 \0那麼多空間(即佔 6空間大小)
8
9     //如果希望直接輸出字元陣列中所有字元,只需要呼叫名稱即可。
10    printf("show: %s\n", str); //output-> show: Hello
11
12    return 0;
13 }
```

***必須要注意：賦值給char陣列時，要使用英文雙引號。**

***注：想要顯示字元，必須使用%s來指定輸出格式，且不能有索引值。**

2、指定輸出字串中的某一位置字元：

```
12 //指定輸出字串中的某一個字元(需指定索引值)
13 //中括號為陣列組,由於陣列索引由0開始, 因此:
14 //索引值 0號 = H
15 //索引值 1號 = e
16 //索引值 2號 = l
17 //索引值 3號 = l
18 //索引值 4號 = o
19 //索引值 5號 = \0 (由於字串保存時最後必須用\0標記, 因此索引值為5號時 為 \0. *注: \0也佔用一空間)
20 printf("show: %c", str[1]); //output-> show: e
```

***必須要注意：**中括號中的索引值為 負數 或 超出了宣告時的大小, 則程式碼會報錯(錯誤訊息-> **Error: array(str) Out of range!**)