



2017

程式設計
加強班

程式設計與實習(一)



BY 孫茂勛

Email:JOHN85051232@GMAIL.COM



String

字符串宣告：

- ❶ `char *str = "HelloWorld" ;`
- ❷ `char str[] = "HelloWorld" ;`
- ❸ `Char str[20] = "HelloWorld" ;`





String

🎄 Q：字串是一維的字元陣列，那字串陣列是什麼？

練習：

🎄 使用字串陣列name來存放5位同學的英文名字，每個英文名字最長限制為15個字元(含 '\0')。執行時輸入每位同學的名字，當五位同學都輸入完後，將每位同學的名字逐行顯示出來。





Character-handling Library

字元判斷

🎄 #include <ctype.h>





字元判斷

- ❁ isalnum(int c) : 是否為字母或數字
- ❁ isalpha(int c) : 是否為字母
- ❁ isdigit(int c) : 是否為數字
- ❁ islower(int c) : 是否為小寫字母
- ❁ ispunct(int c) : 是否為標點符號
- ❁ isspace(int c) : 是否為空白
- ❁ isupper(int c) : 是否為大寫字母
- ❁ isxdigit(int c) : 是否為16進位數字
- ❁ toupper(int c):小寫轉大寫
- ❁ tolower(int c):大寫轉小寫





字元判斷

```
char a = 's';  
char b = 'B';  
printf("%c %c\n", a, toupper(a));  
printf("%c %c\n", b, tolower(b));
```





String-handling Library

字串處理

🎄 #include <string.h>





- ❁ `strlen(char *s1)` : 取得s1字串的長度(不包含 `'\0'`)。
- ❁ `strcpy(char *s1,char *s2)` : 將s2字串複製到s1，並將s1的內容完全覆蓋。
- ❁ `strncpy(char *s1,char *s2,int n)` : 將s2字串複製到s1，最多複製n個字元。
- ❁ `strcat(char *s1,char *s2)` : 將s2的字串整個串接到s1後面。
- ❁ `strncat(char *s1,char *s2,int n)` :
- ❁ `strcmp(char *s1,char *s2)` : 回傳兩字串是否相同，相同傳回0，不同傳回非0值。
- ❁ `strncmp(char *s1,char *s2,int n)` :





字串處理

`strcpy(str1,str2)`

🌿 會將str2的字串複製到str1，並將str1的內容完全覆蓋。

`strncpy(char *s1,char *s2,int n)`

🌿 將s2字串複製到s1，最多複製n個字元。





字串處理

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <string.h>
4
5 int main() {
6
7     char str1[] = "helloworld";
8     char str2[] = "csie";
9     printf("%s\n", str1);
10    printf("%s\n", str2);
11
12    printf("\n\nstr2 部份複製到 str1 \n\n");
13    strncpy(str1, str2, 2);
14    printf("%s\n", str1);
15    printf("%s\n", str2);
16
17    system("pause");
18    return 0;
19 }
20
```

如果使用strncpy要注意
str1的陣列長度夠放str2的所有字元嗎?





字串處理

`strcat(str1,str2)`

🎄 會將str2的字串整個串接到str1後面

`strncat(char *s1,char *s2,int n)`

🎄 將s2字串串接到s1，最多串接n個字元。





字串處理

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <string.h>
4
5 int main() {
6
7     char str1[] = "book";
8     char str2[] = "csie";
9     printf("%s\n", str1);
10    printf("%s\n", str2);
11
12    printf("\n\nstr2 整個串接到 str1 後面\n\n");
13    strcat(str1, str2);
14    printf("%s\n", str1);
15    printf("%s\n", str2);
16
17    system("pause");
18    return 0;
19 }
20
```

Str1的陣列長度夠放str1+str2的所有字元嗎?





練習

設計一個登入程式，自訂一組英數字夾雜的密碼，開啟後讓使用者輸入密碼，如果密碼正確則印出“登入成功”否則一直讓使用者重新輸入密碼。





字串處理

strlen(str1)

🎄 傳回str1字串的長度(不包含 '\0')





字串處理

`strcmp(str1,str2)`

🎄 比對兩個字串的內容是否相同，相同就傳回0，不同會傳回非0值。





字串處理

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <string.h>
4
5 int main() {
6
7     char str1[] = "book";
8     char str2[] = "csie";
9     printf("%s\n", str1);
10    printf("%s\n", str2);
11
12    if(strcmp(str1, str2) == 0) {
13        printf("兩個字串相同\n");
14    }
15    else {
16        printf("兩個字串不同\n");
17    }
18
19    system("pause");
20    return 0;
21 }
22
```





字串處理

`strchr(str1, 'c')`

- 🎄 找尋str1中有沒有字元c的存在
- 🎄 Str1是目標字串，c是要尋找的字元
- 🎄 回傳第一次找到c字元的指標，找不到則回傳NULL

`strrchr`

- 🎄 與strchr相同，只是strchr是由前往後找，而strrchr是由後往前找





字串處理

```
5 int main() {  
6  
7     char str1[] = "helloworld";  
8     char str2[] = "book";  
9     char test = '\\0';  
10    printf("%s\\n" , str1);  
11    printf("%s\\n" , str2);  
12  
13  
14  
15    if(strchr(str1,'e')==NULL) {  
16        printf("找不到符合的字元\\n");  
17    }  
18    else{  
19        printf("有找到目標字元\\n");  
20    }  
21  
22    system("pause");  
23    return 0;  
24 }  
25
```





字串處理

Strstr(str1,str2)

- 🎄 找尋str1中有沒有str2的存在
- 🎄 Str1是目標字串，str2是要尋找的字串
- 🎄 回傳第一次找到c字元的指標，找不到則回傳NULL

strrstr

- 🎄 與strstr相同，只是strstr是由前往後找，而strrstr是由後往前找





字串處理

```
5 int main() {  
6  
7     char str1[] = "helloworld";  
8     char str2[] = "hello";  
9     char *strtest = NULL;  
10    printf("%s\n", str1);  
11    printf("%s\n", str2);  
12  
13    strtest = strstr(str1, str2);  
14  
15    if (strtest == NULL) {  
16        printf("找不到符合的子字串\n");  
17    }  
18    else {  
19        printf("找到符合的子字串\n");  
20    }  
21  
22  
23    system("pause");  
24    return 0;  
25 }
```





字串處理

`strspn(str1, str2);`

- ❁ 將str1和str2比對，如果str1中有字元是str2中沒有的，就會傳回那個字元在str1中的位置。
- ❁ 如果都符合(str1中的字元str2都有)，就會傳回str1的長度

`strcspn(str1, str2);`

- ❁ 將str1和str2比對，如果str1中有字元是str2中的，就會傳回那個字元在str1中的位置。
- ❁ 如果都符合(str1中的字元str2都沒有)，就會傳回str1的長度





字串處理

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <string.h>
4
5 int main() {
6
7     char str1[] = "helloworzd";
8     char str2[] = "helloworld";
9     size_t strtest ;
10    printf("%s\n" , str1);
11    printf("%s\n" , str2);
12
13    strtest = strstr(str1, str2);
14
15    printf("%d\n" , strtest);
16
17    system("pause");
18    return 0;
19 }
20
```

```
C:\Users\ethan\Desktop\新文件1.exe
helloworzd
helloworld
8
請按任意鍵繼續 . . .
```





字串處理

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <string.h>
4
5 int main() {
6
7     char str1[] = "helloworld";
8     char str2[] = "acfgto";
9     size_t strtest ;
10    printf("%s\n" , str1);
11    printf("%s\n" , str2);
12
13    strtest = strcspn(str1,str2);
14
15    printf("%d\n" , strtest);
16
17    system("pause");
18    return 0;
19 }
20
```

C:\Users\ethan\Desktop\新文件1.exe

helloworld

acfgto

4

請按任意鍵繼續 . . .





練習

宣告一個字串，內容為“hello world cise”，請用空格將此字串分開，並輸出像這樣子：

```
hello  
world  
cise
```





字串處理

`strtok(str1, " ")`

🎄 分割字串





字串處理

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <string.h>
4
5 int main() {
6
7     char str1[] = "hello world csie";
8     char* strttest ;
9
10    strttest = strtok(str1, " ");
11
12    while (strttest != NULL) {
13        printf("%s\n", strttest);
14        strttest = strtok(NULL, " ");
15    }
16
17    system("pause");
18    return 0;
19 }
```

```
C:\Users\ethan\Desktop\新文件1.exe
hello
world
csie
請按任意鍵繼續 . . .
```





字串處理

第一次用 `strtest = strtok(str1, " ");`

- ❶ `funcio` 內部有一個指標指向了 `str1`

- ❷ 執行後指標指向了 `hello` 後的 `world`

之后的迴圈

- ❸ `strtest = strtok(NULL, " ");`

- ❹ 可以看成如果“第一個參數為 `NULL`”，仍然使用上次的 `world` 的位置(存在指標中)開始往後找，直到找到下一個用來分隔的符號(這邊我設為空格)。

- ❺ 不斷用 `NULL` 當參數，直到找完所有的字串





字串處理

`memcpy(str1, str2, n)`

🎄 與 `strncpy` 同樣功能

🎄 程式從字元陣列 `str2` 拷貝 `n` 個字元替換字元陣列 `str1` 前端的 `n` 個字元





字串處理

`memmove(str1, str2, n)`

🎄 程式從字元陣列 `str2` 拷貝 `n` 個字元替換字元陣列 `str1` 前端的 `n` 個字元





字串處理

`memcmp(str1, str2, n)`

- 🎄 比較str1及str2前n個字元有無相同
- 🎄 相同就傳回0
- 🎄 不同就傳回非0的數字





參考資料

-  <http://www.cplusplus.com/reference/cctype/>
-  <http://www.cplusplus.com/reference/cstring/?kw=string.h>





THANK YOU

