字元陣列

C-String

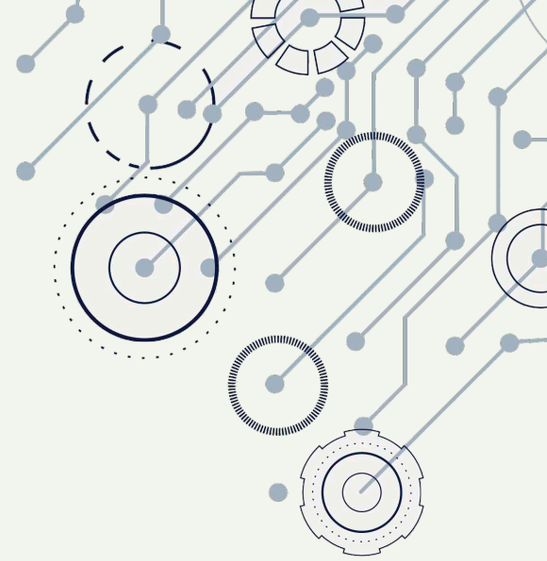
Ya-Chun Hsu @ Taipei Municipal Jianguo High School

字元陣列 C-string

[0]	[1]	[2]	[3]	[4]	[5]
'H'	'A'	'P'	'P'	'Y'	'\0'

NULL字元

NULL字元 NULL Character



'\0'

- 紀錄字串的結尾位置
- 字元陣列必須宣告「字串長度+1」才夠用

字元陣列 輸入與輸出

- 需事先宣告所要使用的陣列大小

```
4 int main(){
5     char name[105];
6     cin>>name;
7     cout<<"Hello "<<name<<endl;
8     return 0;
9 }
```

字元陣列 cin

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 int main(){
6     char str[105];
7     cout<<"Please input a name...>";
8     cin>>str;
9     cout<<"Hello "<<str<<"."<<endl;
10    return 0;
11 }
```

Input	Output
Tony Stark	Hello Tony.

字元陣列 cin.getline()

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

int main(){
    char str[105];
    cout<<"Please input a name...>";
    cin.getline(str,105);
    cout<<"Hello "<<str<<"."<<endl;
    return 0;
}
```

Input	Output
Tony Stark	Hello Tony Stark.

字元陣列 初始化

```
5       char non_init[100];       //未初始化
6       char empty[100] = "";     //空字串 (100格'\0')
7       char init[100] = "Happy";
8       char init2[100] = {'H', 'a', 'p', 'p', 'y'};
9       char six[] = "Happy";     //6格的char陣列
10      char five[] = {'H', 'a', 'p', 'p', 'y'}; //5格的char陣列
11      char *p = "Happy";  // 一個指向char的pointer
12      cout<<five<<endl;    // 未知的結果
```

字串長度 strlen()

```
size_t strlen ( const char * str );
```

```
5  int main(){
6      char name[15]={};
7      cout<<"Please input a name...>";
8      cin.getline(name,15,'\n');
9      cout<<name<<endl;
10     cout<<strlen(name)<<endl;
11     return 0;
12 }
```

Input	Output
Tony Stark	Tony Stark 10

複製字串 strcpy()

```
char * strcpy ( char * destination, const char * source );
```

```cpp
int main(){
    char source[]="Hello, World!";
    char destination[40];
    strcpy(destination,source);
    cout<<destination<<endl;
    return 0;
}
```

Output

```
Hello, World!
```

連接字串 strcat()

```
char * strcat ( char * destination, const char * source );
```

```
 5 int main(){
 6     char str1[105]={};
 7     char str2[105]={"World!"};
 8     strcat(str1,"Hello, ");
 9     strcat(str1,str2);
10     cout<<str1<<endl;
11     return 0;
12 }
```

Output

```
Hello, World!
```

比較字串 strcmp()

```
int strcmp ( const char * str1, const char * str2 );
```

```cpp
int main(){
    char str1[]={"APPLE"};
    char str2[]={"APPEAL"};
    int n=strcmp(str1,str2);
    if(n==0)
        cout<<str1<<" is the same as "<<str2<<endl;
    else if(n>0)
        cout<<str1<<" is greater than "<<str2<<endl;
    else
        cout<<str1<<" is less than "<<str2<<endl;
    return 0;
}
```

Output

```
APPLE is greater than APPEAL
```

字串

String

Ya-Chun Hsu @ Taipei Municipal Jianguo High School

字串 string

- string 是一個長度可變之字元序列
- 若要使用string型態，必須加入cstring的標頭檔

字串 宣告與初始化

```
 5      string s1;                  // s1 是空字串
 6      string s2 = s1;             // s2 是 s1 的複製
 7      string s2(s1);              // 與 string s2 = s1; 相同
 8      string s3 = "hiya";         // s3 是 "hiya"
 9      string s3("hiya");          // 與 string s3 = "hiya"; 相同
10      string s4(10, 'c');         // s4 是 "cccccccccc"
```

字串長度 length()

```
size_t length() const;
```

```
 5 int main(){
 6     string str;
 7     getline(cin,str);
 8     cout<<str.length()<<endl;
 9     for(int i=0;i<str.length();i++){
10         cout<<str[i]<<" ";
11     }
12     return 0;
13 }
```

Input	Output
Tony Stark	10 T o n y S t a r k

複製字串

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

int main(){
    string source="Hello, World!";
    string destination;
    destination=source;
    cout<<destination<<endl;
    return 0;
}
```

Output

```
Hello, World!
```

連接字串 +

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

int main(){
    string str1="Hello, ";
    string str2="World!";
    string str3="Tony";
    cout<<str1+str3<<endl;
    str1+=str2;
    cout<<str1<<endl;
    return 0;
}
```

Output

```
Hello, Tony
Hello, World!
```

比較字串 >, ==, <

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

int main(){
    string s1="APPLE",s2="APPEAL";
    if(s1==s2)
        cout<<s1<<" is the same as "<<s2<<endl;
    else if(s1>s2)
        cout<<s1<<" is greater than "<<s2<<endl;
    else
        cout<<s1<<" is less than "<<s2<<endl;
    return 0;
}
```

Output

```
APPLE is greater than APPEAL
```