

# 구조체

# 연관된 여러 값을 하나로 묶기

## ▶ 비디오 게임 예제

- ▶ 플레이어의 이름뿐만 아니라 화면 좌표(x와 y 값)도 저장해야 함
- ▶ 세 값은 서로 짝을 이루고 있음 → **구조체**에 모으자.

```
int x_coordinate;  
int y_coordinate;  
string name;
```

- ▶ SpaceShip 이라는 새로운 데이터타입이 만들어짐

```
SpaceShip my_ship;
```

- ▶ x\_coordinate, y\_coordinate, name은 **필드**라고 함
  - ▶ .을 이용하여 필드에 접근하기

```
// 변수를 사용한다.
```

```
my_ship.x_coordinate = 40;
```

```
my_ship.y_coordinate = 40;
```

```
my_ship.name = "USS Enterprise (NCC-1701-D)";
```

## ▶ 예제 1

```
#include <iostream>

using namespace std;

struct PlayerInfo
{
    int skill_level;
    string name;
};
```

```
int main ()
{
    // 일반적인 변수 타입처럼 구조체도 배열로 만들 수 있다.
    PlayerInfo players[ 5 ];
    for ( int i = 0; i < 5; i++ )
    {
        cout << "Please enter the name for player : "
              << i << '\n';
        // 우선 일반적인 배열 문법으로 배열의 요소에 접근한 다음,
        // '.' 문법으로 구조체의 필드에 접근한다.
        cin >> players[ i ].name;
        cout << "Please enter the skill level for "
              << players[ i ].name << '\n';
        cin >> players[ i ].skill_level;
    }
    for ( int i = 0; i < 5; ++i )
    {
        cout << players[ i ].name
              << " is at skill level "
              << players[ i ].skill_level << '\n';
    }
}
```

## ▶ 예제2

//구조체 선언

```
struct EnemySpaceShip
{
    int x_coordinate;
    int y_coordinate;
    int weapon_power;
};
```

//초기화 함수

//초기화된 구조체를 return

```
EnemySpaceShip getNewEnemy ()
{
    EnemySpaceShip ship;
    ship.x_coordinate = 0;
    ship.y_coordinate = 0;
    ship.weapon_power = 20;
    return ship;
}
```

### 지역변수

Return 안되면 외부에서는  
필드 바뀐 것을 알 수 없음

//업그레이드 함수

//필드가 변경된 구조체를 return

```
EnemySpaceShip upgradeWeapons (EnemySpaceShip ship)
{
    ship.weapon_power += 10;
    return ship;
}
```

### 지역변수

Return 안되면 외부에서는  
필드 바뀐 것을 알 수 없음

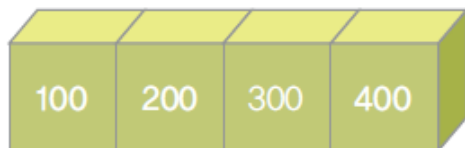
int main () //사용해보기

```
{
    EnemySpaceShip enemy = getNewEnemy();
    enemy = upgradeWeapons( enemy );
}
```

# “문자열” 특집

– array 사용하기

### 정수형 배열

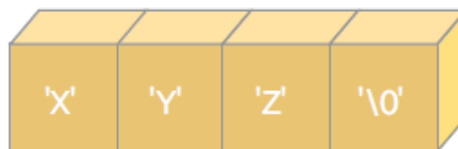


aa[0] aa[1] aa[2] aa[3]

정수형(int) : 각 4바이트

```
int aa[4] = {100, 200, 300, 400};
```

### 문자형 배열



ss[0] ss[0] ss[1] ss[2]

문자형(char) : 각 1바이트

```
char ss[4] = {'X', 'Y', 'Z', '\0'};
```

또는

```
char ss[4];
```

```
ss[0] = 'X';
```

```
ss[1] = 'Y';
```

```
ss[2] = 'Z';
```

```
ss[3] = '\0';
```

또는

```
char ss[4] = "XYZ"
```

**ss = "XYZ"는 안됨!!**

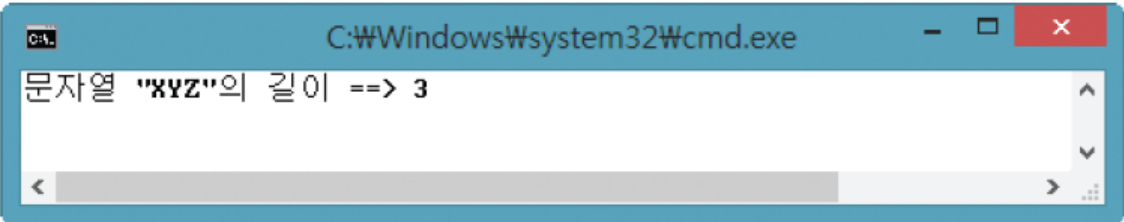
문자열을 배열에 저장하기 위해서는 문자열의 끝을 나타내는 '\0'을 고려하여 **'문자열 길이 +1'** 크기의 배열이 필요하다.

# #include <string>

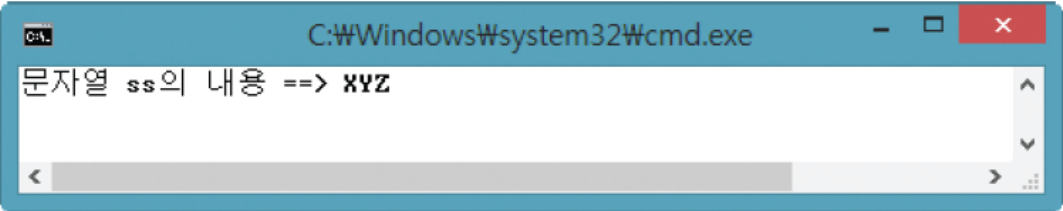
종류	기능
strlen()	문자열의 길이를 알려준다.
strcpy_s()	문자열을 복사한다.
strcat_s()	두 문자열을 이어준다.
strcmp()	두 문자열을 비교한다.
gets_s(), puts()	문자열을 키보드로 입력하거나 화면에 출력한다.

## 실행결과 ▼

```
char ss[] = "XYZ";
int len;
len = strlen(ss);
```



```
char hh[] = "XYZ";
char ss[4];
strcpy_s(ss, 4, hh);
```



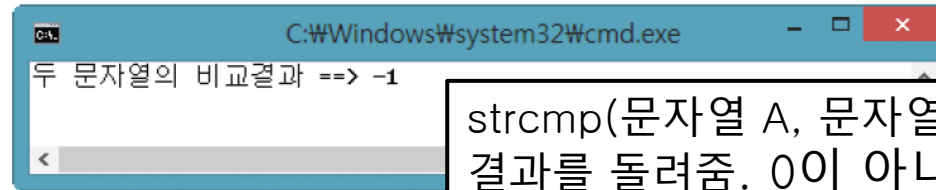
```
char ss[7] = "XYZ";
strcat_s(ss, 7, "ABC");
```

기존 문자열 뒤에 이어준다.

기존 '0' 자리부터 이어진다.

```
char ss[] = "XYZ";  
char tt[] = "xyz";  
int r;  
r = strcmp(ss, tt);
```

실행결과 ▼



strcmp(문자열 A, 문자열 B)는 'A-B'의 결과를 돌려줌. 0이 아니면 다르다는 뜻

```
char ss[10];  
gets_s(ss, 10);
```

최대 입력 문자는 9 문자를 고려해서 '배열크기 -1'까지 입력  
Enter 키를 입력할 때까지 ss에 문자열을 받아들임  
(공백도 문자로 인식할 수 있어요)

```
char ss[10] = "XYZ";  
puts(ss);
```

'\n'이 없어도 출력한 후 자동으로 줄을 넘김



# “문자열” 특집

- String 데이터 타입 사용하기

```

#include <iostream>
#include <string>

using namespace std;

int main ()
{
    string user_first_name;
    string user_last_name;

    cout << "Please enter your first name: ";
    cin >> user_first_name;
    cout << "Please enter your last name: ";
    cin >> user_last_name;
    string user_full_name =
        user_first_name + " " + user_last_name;

    cout << "Your name is: " << user_full_name << "\n";
}

```

//문자열 변수 2개 선언

//cin으로 입력 받아서 대입

//+ 기호를 문자열을 연결하는  
용도로 쓸 수 있음

```
string user_name = "어떤 값"; //초기화
```

```
getline( cin, user_first_name, '\n' );
```

→이 문자를 만나기 전까지의 문자열을  
user\_first\_name에 저장

→ 이후에 입력한 것은 버퍼에 저장되어 있다가  
다음 getline에서 읽어들이

종류	String에서는..
strlen( )	ss.length()
strcpy_s( )	ss1 = ss2
strcat_s( )	ss1 + ss2
strcmp( )	ss1 == ss2

<http://www.cplusplus.com/reference/string/string/>  
String의 그 밖의 다양한 기능들

## TEAM EX1 구조체로 명함정리 프로그램 만들기

```
01 #include <iostream>
02 using namespace std;
03 struct namecard{
04     char name[20];
05     char job[30];
06     char tel[20];
07     char email[40];
08 };
09 void main()
10 {
11     namecard x={ "Rhee", "Professor", "418-9876", "chae.rhee@inha.ac.kr"};
12     namecard y={ "Jang", "TA", "551-6986", "jang@inhau.ac.kr"};
13     namecard z={ "Kim", "Student", "318-3961", "ds@naver.com"};
14
15     cout<<" Name\Wt Job \Wt\Wt Tel \Wt email ";
16     cout<<"\Wn =====";
17     cout<<"\Wn "<<x.name <<"\Wt"<< x.job <<"\Wt"<< x.tel <<"\Wt"<< x.email;
18     cout<<"\Wn "<<y.name <<"\Wt"<< y.job <<"\Wt"<< y.tel <<"\Wt"<< y.email;
19     cout<<"\Wn "<<z.name <<"\Wt"<< z.job <<"\Wt"<< z.tel <<"\Wt"<< z.email;
20     cout<<"\Wn =====\Wn";
21 }
```

새로운 namecard k에  
namecard x의 내용을 그대로  
복사하고 싶으면 어떻게??

## TEAM EX2 입력된 문자열을 거꾸로 출력

```
01 #include <iostream>
02 #include <string>
03 using namespace std
04 int main( ){
05     char ss[100];
06     char tt[100];
07     int count, i;
08
09     cout<<"enter the sentence : ";
10     gets_s(ss, 100);
11
12     count = strlen(ss);
13
14     for(i=0 ; i<count ; i++)
15         tt[i] = ss[count-(i+1)];
16
17     tt[count] = '\0';
18     cout<<"reversed sentence : ";
19     puts(tt);
20 }
```

----문자형 배열 ss를 선언한다.

----문자형 배열 tt를 선언한다.

----문자를 입력받는다.

----입력받은 문자의 개수를 구한다.

----문자열의 개수만큼 반복해서 tt 배열에  
문자열을 거꾸로 저장한다.

----tt 배열의 마지막에 널 문자를 입력한다

## TEAM EX3 문자열 내 특정 문자의 변환

배열로 문자열을 입력 받고

그 문자열에서 변환될 문자와 변환할 문자를 입력 받아

해당하는 문자를 변환해주기

Enter the sentence: Microsoft Visual Studio

From: i

To: #

Replaced sentence:

M#crosoft V#sual Stud#o

아래 함수들을 활용하면 구현 가능함

gets\_s(str, 100);

strlen(str)

puts(str) 등..

## TEAM EX4 문자열을 위한 함수 사용 기초

```
#include <string.h> // to use strcpy(), strcmp(), strlen()
#include <stdio.h>   // to use printf(), scanf()

void main(){
    char x[50];
    strcpy(x, "hello");
    printf("x is %s\n", x);
    printf("the length of x is %d\n", strlen(x));

    char y[50];
    printf("Enter a string\n");
    scanf("%s", y);
    printf("y is %s\n", y);

    if (strcmp(x, y)==0){
        printf("x and y are same\n");
    }else{
        printf("x and y are not same\n");
    }
}
```

## TEAM EX5 문자열을 위한 함수 사용 연습

배열로 문자열을 입력

```
Enter a string: envelope
There are 3 'e's;
Enter a string: alleged
There are 2 'e's;
Enter a string: book
There is no e. bye~
```