



**ROBIT**  
ROBOT SPORT GAME TEAM

# C Language 기초 - 4일차

## 1. 구조체



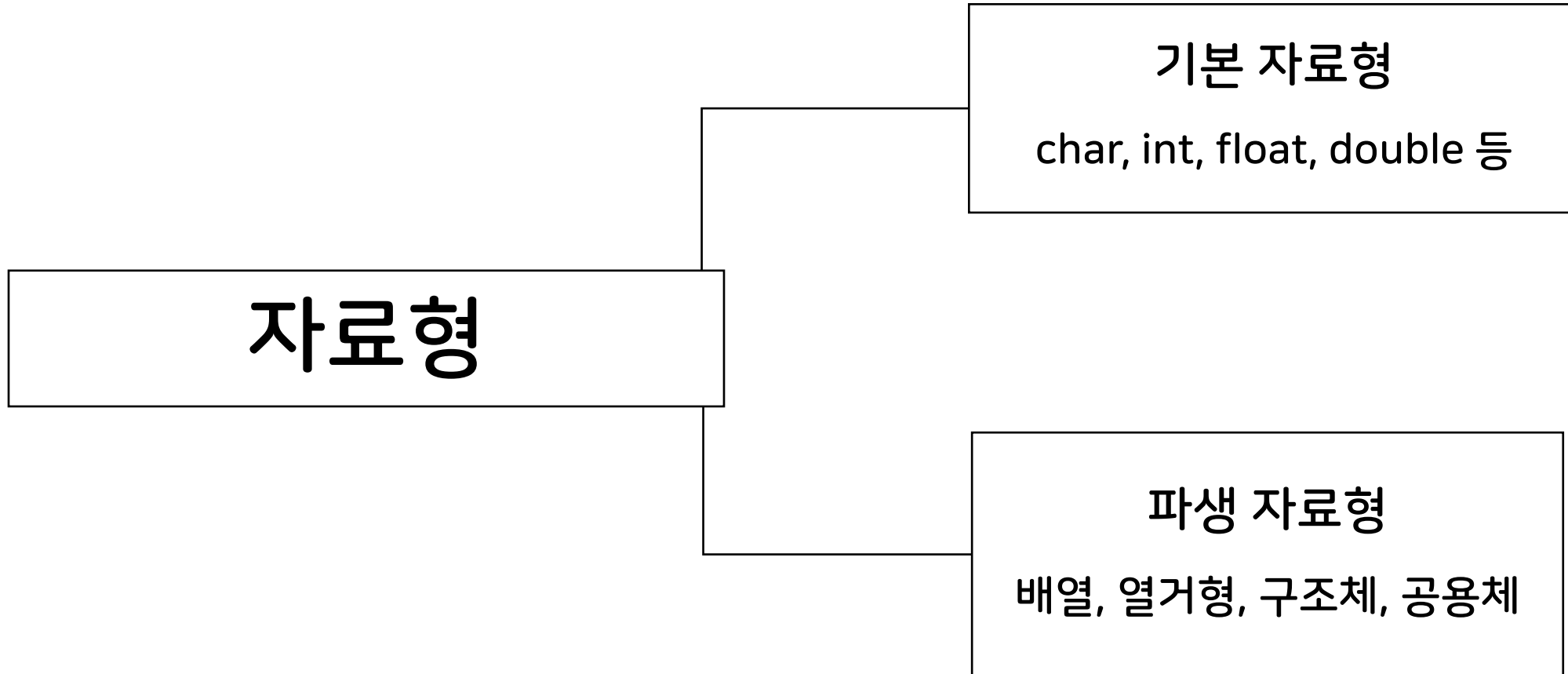
**ROBIT**  
ROBOT SPORT GAME TEAM

# 구조체(Structure)

# 자료형의 분류



**ROBIT**  
ROBOT SPORT GAME TEAM



# 구조체의 필요성



**ROBIT**  
ROBOT SPORT GAME TEAM

학생에 대한 데이터

```
int      number      = 0;
char     name[10]     = {0, };
int      age          = 0;
double   grade        = 0.0;
```



# 구조체(Structure)

- 하나 이상의 변수를 그룹을 묶어서 새로운 자료형을 정의하는 것

```
struct student {  
    int number;  
    char name[10];  
    double grade;  
};
```

# 구조체와 배열의 차이



**ROBIT**  
ROBOT SPORT GAME TEAM

## 배열

같은 자료형의 집합

값	153				21				5454			
이름	Name[0]				Name[1]				Name[2]			
주소	4	5	6	7	8	9	10	11	12	13	14	15

## 구조체

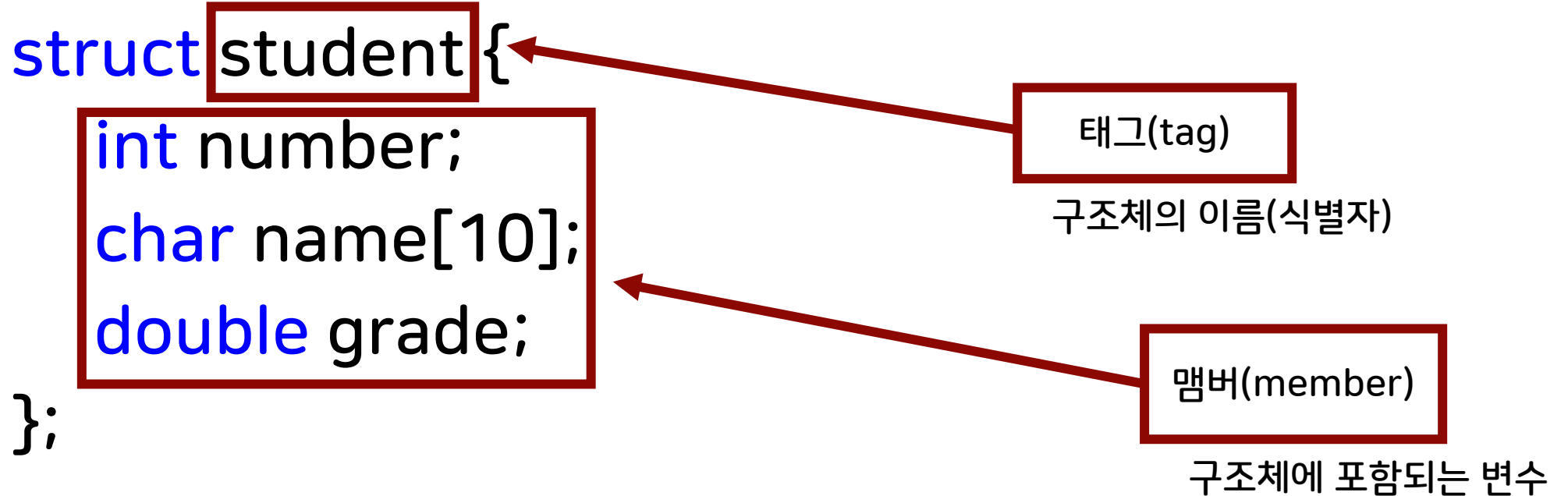
서로 다른 자료형의 집합

값	134				'a'	5.274							
이름	i				c	d							
주소	8	9	10	11	12	13	14	15	16	17	18	19	20

# 구조체 생성



**ROBIT**  
ROBOT SPORT GAME TEAM



구조체 선언은 변수 선언이 아니다.  
새로운 사용자 정의 자료형을 만든 것이다.



# 구조체 선언

- 구조체 정의와 구조체 변수 선언은 다르다.

```
struct student
{
    int number;
    char name[10];
    double grade;
};
```

구조체 정의

```
void main( )
{
    struct student s1;
}
```

구조체 변수 선언

“**struct** type\_name var\_name” 의 형태로 구조체 변수를 선언한다.





# 구조체 선언

- 구조체 정의와 구조체 변수 선언은 다르다.

```
struct student
{
    int number;
    char name[10];
    double grade;
};
```

```
void main( )
{
    struct student s1;
}
```

	s1		
이름	number	name[10]	grade
주소	4	8	18

# 구조체 초기화



**ROBIT**  
ROBOT SPORT GAME TEAM

```
struct student
{
    int number;
    char name[10];
    double grade;
};

void main( )
{
    struct student s1 = {22, "Kim Jihun", 4.5};
}
```

선언과 동시에 초기화 가능  
배열의 초기화와 동일  
멤버의 순서대로 초기화할 대상 나열



# 구조체 멤버 참조

멤버 연산자(.)를 이용하여 멤버 참조

```
s1.number = 22;  
s1.name[0] = 'K';  
s1.name[1] = 'i';  
s1.name[2] = 'm';  
s1.grade = 4.5;
```

구조체 변수

구조체 멤버

```
#include <stdio.h>  
  
struct student  
{  
    int number;  
    char name[10];  
    double grade;  
};  
  
void main()  
{  
    struct student s1 = {22, "Kim Jihun", 4.5};  
  
    s1.  
}
```

- grade
- name
- number

public : double student::grade  
파일: main.c

# 구조체 실습1



**ROBIT**  
ROBOT SPORT GAME TEAM

```
struct student
{
    int number;
    char name[10];
    double grade;
};

void main( )
{
    struct student s1 = {22, "Kim Jihun", 4.5};

    printf("number : %d\n", s1.number);
    printf("name : %s\n", s1.name);
    printf("grade : %0.1lf\n", s1.grade);
}
```

```
number : 22
name : Kim Jihun
grade : 4.5
계속하려면 아무 키나 누르십시오 . . .
```

# 구조체 실습2



**ROBIT**  
ROBOT SPORT GAME TEAM

```
struct student
{
    int number;
    char name[10];
    double grade;
};

void main()
{
    struct student s;

    printf("input number : ");
    scanf("%d",&s.number);

    printf("input name : ");
    scanf("%s",s.name);

    printf("input grade : ");
    scanf("%lf",&s.grade);

    printf("\n\n");

    printf("number : %d\n",s.number);
    printf("name : %s\n",s.name);
    printf("grade : %0.1lf\n",s.grade);
}
```

```
input number : 44
input name : kim
input grade : 4.5
```

```
number : 44
name : kim
grade : 4.5
계속하려면 아무 키나 누르십시오 . . .
```

# 구조체 실습 3



**RO:BIT**  
ROBOT SPORT GAME TEAM

```
#include <stdio.h>
#include <math.h>

struct point
{
    int x;
    int y;
};

void main()
{
    struct point p1,p2;

    int xdiff = 0;
    int ydiff = 0;
    double distance = 0.0;

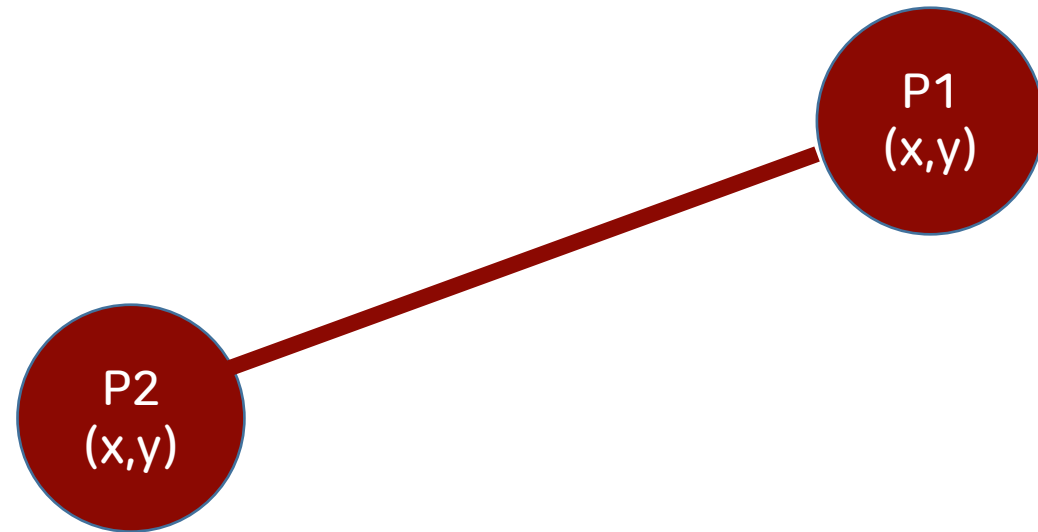
    printf("점의 좌표를 입력하시오(x,y): ");
    scanf("%d,%d", &p1.x, &p1.y);

    printf("점의 좌표를 입력하시오(x,y): ");
    scanf("%d,%d", &p2.x, &p2.y);

    xdiff = p1.x - p2.x;
    ydiff = p1.y - p2.y;

    distance = sqrt((double)(xdiff * xdiff + ydiff * ydiff));
    printf("두 점사이의 거리는 %f입니다.\n", distance);
}
```

점의 좌표를 입력하시오(x,y): 8,6  
점의 좌표를 입력하시오(x,y): 2,-5  
두 점사이의 거리는 12.529964입니다.  
계속하려면 아무 키나 누르십시오 . . .



# 구조체 실습 4



**RO:BIT**  
ROBOT SPORT GAME TEAM

```
#include <stdio.h>

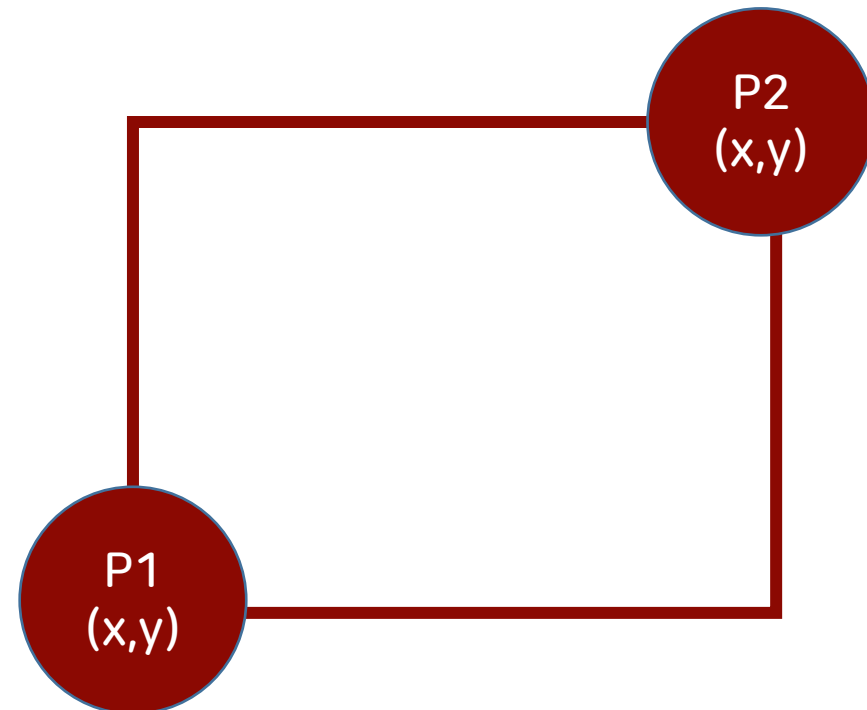
struct point
{
    int x;
    int y;
};

struct rect
{
    struct point p1;
    struct point p2;
};

void main()
{
    struct rect r;
    int width = 0;
    int height = 0;
    int area = 0;

    printf("왼쪽 하단의 좌표를 입력하시오(x,y): ");
    scanf("%d,%d", &r.p1.x, &r.p1.y);
    printf("오른쪽 상단의 좌표를 입력하시오(x,y): ");
    scanf("%d,%d", &r.p2.x, &r.p2.y);
    width = r.p2.x - r.p1.x;
    height = r.p2.y - r.p1.y;
    area = width * height;
    printf("면적은 %d입니다.\n",area);
}
```

왼쪽 하단의 좌표를 입력하시오(x,y): 1,5  
오른쪽 상단의 좌표를 입력하시오(x,y): 6,42  
면적은 185입니다.  
계속하려면 아무 키나 누르십시오 . . .



# 구조체 실습 5



**ROBIT**  
ROBOT SPORT GAME TEAM

```
#include <stdio.h>
```

```
struct point  
{  
    int x;  
    int y;  
};
```

```
void main()  
{  
    struct point p1 = {10,20};  
    struct point p2 = {15,5};  
  
    printf("Before\n");  
    printf("p1 (%d,%d) p2 (%d,%d)\n\n", p1.x, p1.y, p2.x, p2.y);  
  
    p2 = p1;  
  
    printf("After\n");  
    printf("p1 (%d,%d) p2 (%d,%d)\n\n", p1.x, p1.y, p2.x, p2.y);  
}
```

Before  
p1(10,20) p2(15,5)

After  
p1(10,20) p2(10,20)

계속하려면 아무 키나 누르십시오 . . .



# 구조체 실습 6



**ROBIT**  
ROBOT SPORT GAME TEAM

```
#include <stdio.h>

struct point
{
    int x;
    int y;
};

void main()
{
    struct point p1 = {10,20};
    struct point p2 = {15,5};

    printf("p1값을 입력하시오(x,y): ");
    scanf("%d,%d", &p1.x, &p1.y);
    printf("p2값을 입력하시오(x,y): ");
    scanf("%d,%d", &p2.x, &p2.y);

    if(p1==p2)
    {
        printf("같은ㄴn");
    }
    else
    {
        printf("다른ㄴn");
    }
}
```

컴파일 오류

# 구조체 실습 6



**ROBIT**  
ROBOT SPORT GAME TEAM

```
#include <stdio.h>
```

```
struct point  
{  
    int x;  
    int y;  
};
```

```
void main()  
{
```

```
    struct point p1 = {10,20};  
    struct point p2 = {15,5};
```

```
    printf("p1값을 입력하시오(x,y): ");  
    scanf("%d,%d", &p1.x, &p1.y);  
    printf("p2값을 입력하시오(x,y): ");  
    scanf("%d,%d", &p2.x, &p2.y);
```

```
    if((p1.x==p2.x)&&(p1.y==p2.y))  
    {
```

```
        printf("같은\n");
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("다름\n");
```

```
    }
```

```
}
```

```
p1값을 입력하시오(x,y): 2,5  
p2값을 입력하시오(x,y): 2,5  
같은  
계속하려면 아무 키나 누르십시오 . . .
```

```
p1값을 입력하시오(x,y): 5,6  
p2값을 입력하시오(x,y): 1,9  
다름  
계속하려면 아무 키나 누르십시오 . . .
```



***ROBIT***  
ROBOT SPORT GAME TEAM

typedef



# typedef

- 기존 자료형에 새로운 자료형을 추가
- C의 기본 자료형을 확장 시키는 역할
  1. Byte
  2. U8
  3. U16
  4. INT32



# typedef

`typedef old_type new_type;`

```
#include <stdio.h>
```

```
typedef unsigned char byte;
```

자료형의 이름 unsigned char에  
byte라는 이름을 추가로 붙여준다.

```
void main()  
{  
    byte newType;  
  
    newType = 'A';  
  
    printf("%c\n", newType);  
}
```



# typedef

구조체로 새로운 타입을 정의할 수 있다.

```
]struct student  
{  
    int number;  
    char name[10];  
    double grade;  
};
```

```
typedef struct _Student  
{  
    int number;  
    char name[10];  
    double grade;  
}Student;
```



# typedef

구조체로 새로운 타입을 정의할 수 있다.

```
struct student s1 = {22, "Kim Jihun", 4.5};
```

```
void main()  
{  
    Student s;  
  
    printf("input number : ");  
    scanf("%d",&s.number);  
  
    printf("input name : ");  
    scanf("%s",s.name);  
  
    printf("input grade : ");  
    scanf("%lf",&s.grade);  
  
    printf("\n\n");  
  
    printf("number : %d\n",s.number);  
    printf("name : %s\n",s.name);  
    printf("grade : %0.1lf\n",s.grade);  
}
```



**ROBIT**  
ROBOT SPORT GAME TEAM

# 구조체 배열





# 구조체 배열

## • 구조체 배열의 선언

```
#include <stdio.h>
#include <string.h>

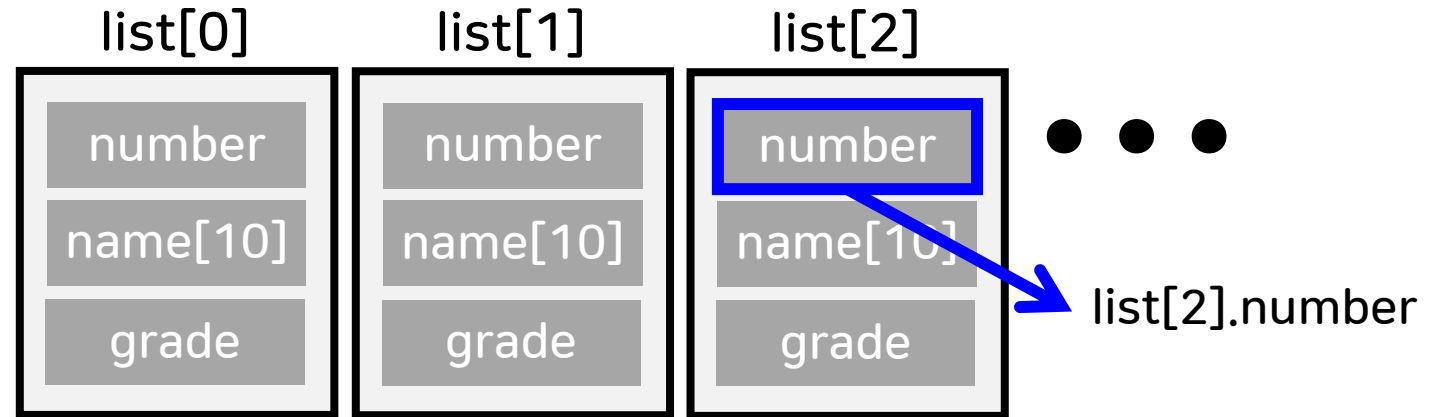
typedef struct _Student
{
    int number;
    char name[10];
    double grade;
}Student;

void main()
{
    Student list[100];

    list[5].number = 27;
    strcpy(list[5].name, "Kim");
    list[5].grade = 4.5;

    printf("number : %d\n", list[5].number);
    printf("name : %s\n", list[5].name);
    printf("grade : %0.1f\n", list[5].grade);
}
```

일반적인 배열의 선언방법과 동일



number : 27

name : Kim

grade : 4.5

계속하려면 아무 키나 누르십시오 . . .



# 구조체 배열

## • 구조체 배열의 초기화

```
#include <stdio.h>

typedef struct _Student
{
    int number;
    char name[10];
    double grade;
}Student;

void main( )
{
    Student list[2] = { {44,"Kim",4.5}, {38,"Lee",4.0} };
}
```

# 구조체 배열 실습 1



**ROBIT**  
ROBOT SPORT GAME TEAM

```
#include <stdio.h>

#define SIZE 2

typedef struct _Student
{
    int number;
    char name[10];
    double grade;
}Student;

void main(void)
{
    Student list[SIZE];
    int i = 0;

    for(i = 0; i < SIZE; i++)
    {
        printf("학번을 입력하시오: ");
        scanf("%d", &list[i].number);
        printf("이름을 입력하시오: ");
        scanf("%s", list[i].name);
        printf("학점을 입력하시오(실수): ");
        scanf("%lf", &list[i].grade);
    }

    for(i = 0; i < SIZE; i++)
    {
        printf("학번: %d, 이름: %s, 학점: %f\n", list[i].number, list[i].name, list[i].grade);
    }
}
```

```
학번을 입력하시오: 44
이름을 입력하시오: kim
학점을 입력하시오(실수): 4.5
학번을 입력하시오: 38
이름을 입력하시오: lee
학점을 입력하시오(실수): 4.0
학번: 44, 이름: kim, 학점: 4.500000
학번: 38, 이름: lee, 학점: 4.000000
계속하려면 아무 키나 누르십시오 . . .
```

# 구조체 함수



**ROBIT**  
ROBOT SPORT GAME TEAM

```
#include <stdio.h>

typedef struct _Student
{
    int number;
    char name[10];
    double grade;
}Student;

int isEqualGrade(Student s1, Student s2);

void main(void)
{
    Student s1 = {44, "kim", 4.5};
    Student s2 = {10, "qwe", 4.5};
    int result = 0;

    result = isEqualGrade(s1, s2);

    if(result == 1)
    {
        printf("같은\n");
    }
    else
    {
        printf("다름\n");
    }
}

int isEqualGrade(Student s1, Student s2)
{
    if(s1.grade == s2.grade)
        return 1;
    else
        return 0;
}
```

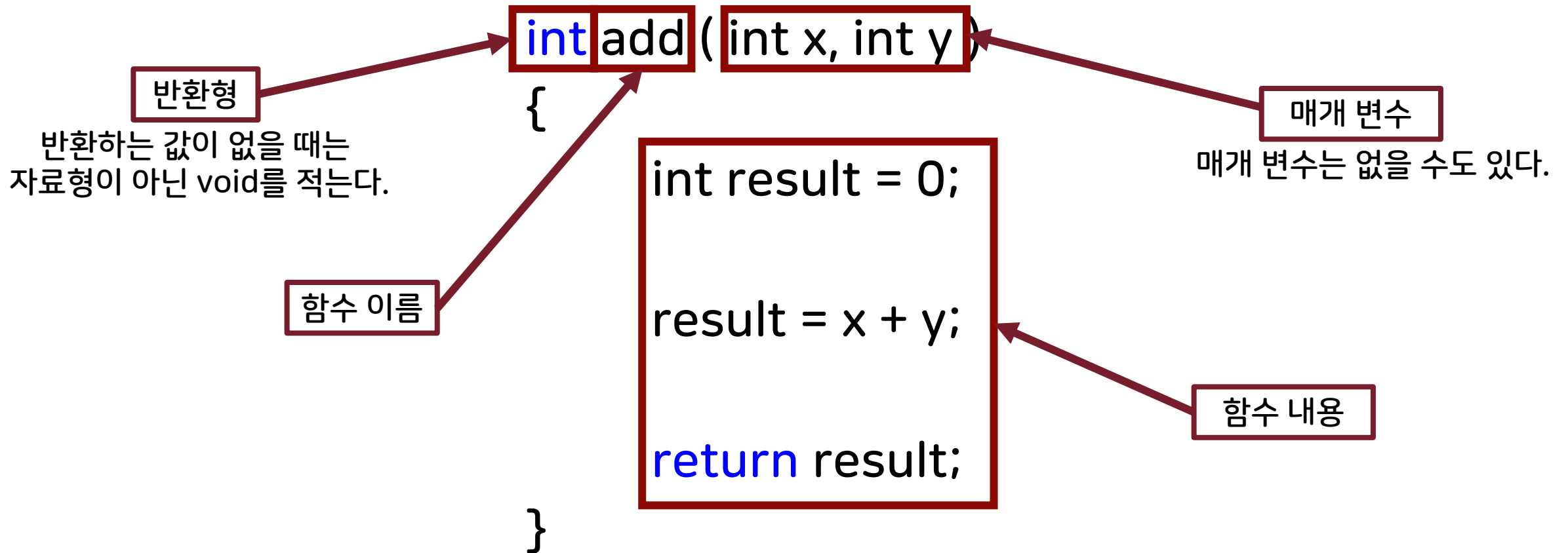
## 구조체를 함수의 인수로 전달하는 경우

1. 구조체의 복사본이 함수로 전달 된다.
2. 구조체의 크기가 크면 그만큼 시간과 메모리 소비

# 함수 정의



**ROBIT**  
ROBOT SPORT GAME TEAM



# 구조체 반환



**ROBIT**  
ROBOT SPORT GAME TEAM

```
#include <stdio.h>

#define SIZE 2

typedef struct _Student
{
    int number;
    char name[10];
    double grade;
}Student;

Student makeStudent();

void main(void)
{
    int i=0;
    Student student[SIZE];

    for(i=0; i<SIZE; i++)
    {
        printf("%d\n", i);
        student[i] = makeStudent();
        printf("\n");
    }
    printf("\n\n");

    for(i=0; i<SIZE; i++)
    {
        printf("학번 : %d, 이름 : %s, 학점 : %0.1f\n", student[i].number, student[i].name, student[i].grade);
    }
}
```

```
Student makeStudent()
{
    Student s;

    printf("number :");
    scanf("%d", &s.number);
    printf("name :");
    scanf("%s", s.name);
    printf("grade :");
    scanf("%lf", &s.grade);
    return s;
}
```

```
0
number :44
name :kim
grade :4.5
```

```
1
number :42
name :lee
grade :4.2
```

```
학번 : 44, 이름 : kim, 학점 : 4.5
학번 : 42, 이름 : lee, 학점 : 4.2
계속하려면 아무 키나 누르십시오 . . .
```



**ROBIT**  
ROBOT SPORT GAME TEAM

# 과제

# 과제 1



**ROBIT**  
ROBOT SPORT GAME TEAM

- 문자열 길이를 비교하는 프로그램

문자열 `str1`과 문자열 `str2`의 길이를 비교하여  
`str1`이 더 길면 -1을 반환, `str2`가 더 길면 1을 반환, 같으면 0을 반환하는 함수 작성

(조건 : 구조체 사용)

입력

```
문자열 str1을 입력하시오. abcdef
문자열 str2을 입력하시오. abcd
str1이 더 길니다.
```

출력

```
문자열 str1을 입력하시오. abcd
문자열 str2을 입력하시오. abcd
문자열의 길이가 같습니다.
```

```
문자열 str1을 입력하시오. abcd
문자열 str2을 입력하시오. abcdef
str2가 더 길니다.
```



# 과제 2



**ROBIT**  
ROBOT SPORT GAME TEAM

- 문자열을 복사하는 프로그램

문자열 str1의 내용을 문자열 out에 복사

복사를 할 수 없을 경우 -1을 반환, 복사를 했을 경우 1을 반환하는 함수를 통해 과제 수행

(조건 : 구조체 사용)

```
문자열을 입력하시오(str1). robit  
out 문자열을 입력하시오. good  
복사할 수 없습니다.
```

입력

출력

```
문자열을 입력하시오(str1). good  
out 문자열을 입력하시오. robit  
복사되었습니다.(good)
```

# 과제 3



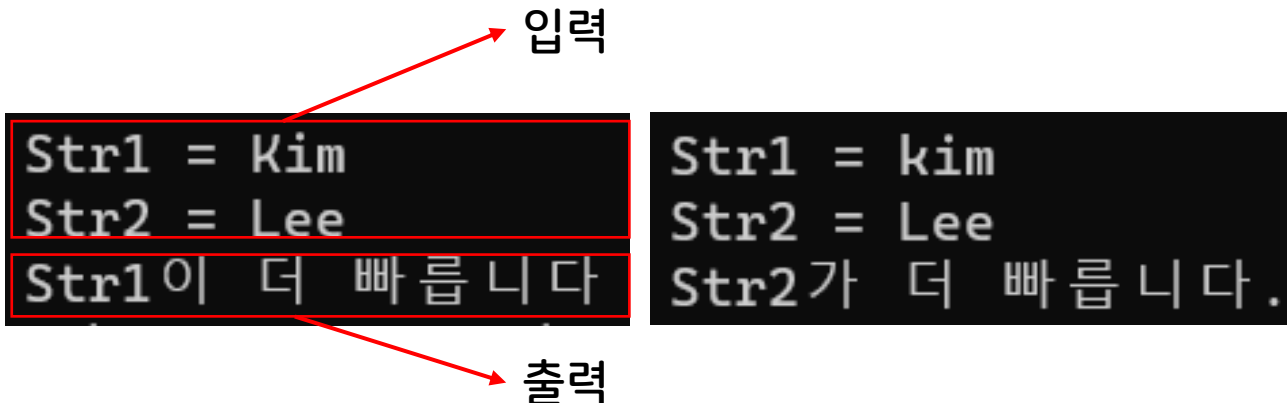
**ROBIT**  
ROBOT SPORT GAME TEAM

## • 문자열 순서를 비교하는 프로그램

문자열 str1과 문자열 str2의 이름 순서를 비교하는 함수 작성

str1의 순서가 더 빠르면 -1을 반환, str2의 순서가 더 빠르면 1을 반환 ,같으면 0을 반환

(조건 : 구조체 사용)



Ex)

Str1 = "Kim";  
Str2 = "Lee"; 를 함수 인자에 넣으면

알파벳 순서가  
str1은 K, str2는 L이므로 -1을 반환

영어 사전 순

# 과제 4



**ROBIT**  
ROBOT SPORT GAME TEAM

- 합집합 면적 구하기

2개의 직사각형이 존재한다. 직사각형의 좌표는 정수이다.

-서로 떨어져 있을 수도 있고 포함할 수도 있고 겹칠 수도 있다.

2개의 직사각형의 꼭짓점을 입력 받고 면적 구하기 (조건 : 구조체 사용)

