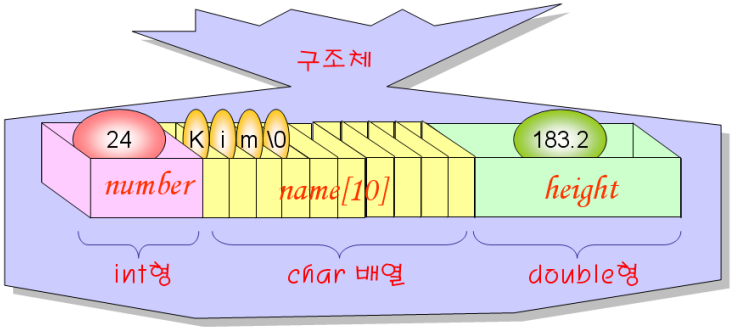
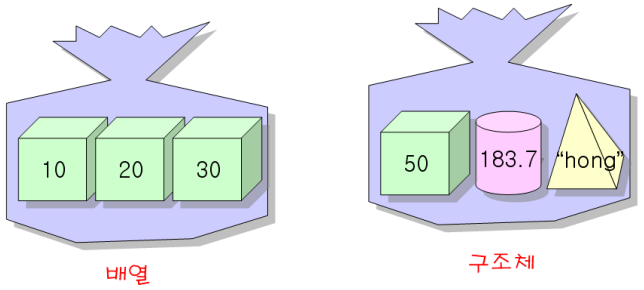
구조체 Part 1

struct(구조체)란?

- 논리적으로 관련된 데이터의 연속적 모임, 즉 집단 데이터를 표현할 수 있는 데이터 유형(type)
  - 서로 다른 유형의 데이터 연속적 모임이 가능



- 구조체 vs. 배열
  - 배열은 서로 동일한 유형의 데이터의 연속된 모임



그러면 WHY 구조체 ?

Q? IT 미디어과에서 여러분의 신상기록을 데이터로 만들고 싶다.
여러분의 신상기록은 (1) 이름, (2) 학번, (3) 생년월일, (4) 주소
(5) 출신고등학교, (6) 키, (7) 친구의 수 라고 한다. 그리고 학생 수는
50명이다. 어떤 형태의 자료 구조가 있으면 좋을까?

Q? 여러분의 주민등록증을 살펴보자. 어떤 정보들이 기록되어 있는가? 그러면 이러한 주민등록증은 몇 개가 존재하는가?

Q? 여러분의 학생부에는 어떤 정보들이 기록되어 있는가? 이러한 학생부는 몇 개가 존재하는가?

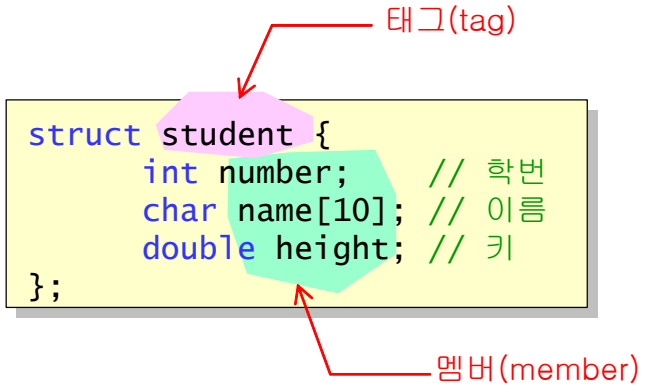
.
.
.

구조체의 선언

- 구조체 선언 형식

```
struct tag_name {
    data_type    member_name ;
    data_type    member_name ;
    …
} ;
```

- (예)학생에 대한 데이터



- 구조체 선언은 변수 선언이 아니며, 집단 데이터의 형태(template)만 정의
  - 메모리 영역을 차지하지 않는다
  - cf. java의 클래스 선언

구조체 선언의 예

```
// x값과 y값으로 이루어지는 화면의 좌표
struct point {
        int x;                  // x 좌표
        int y;                  // y 좌표
};
```

```
// 복소수
struct complex {
        double real;            // 실수부
        double imag;            // 허수부
};
```

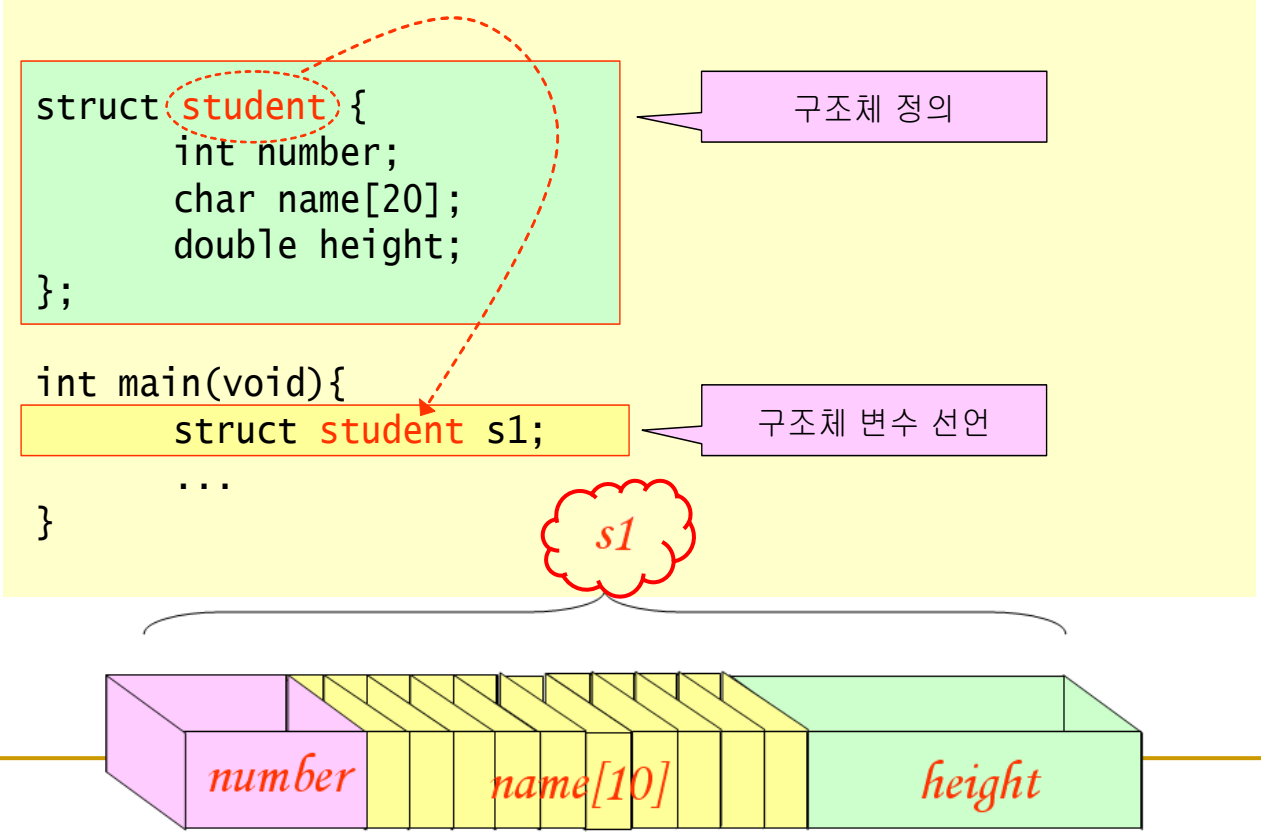
```
// 날짜
struct date {
        int month;
        int day;
        int year;
};
```

```
// 사각형
struct rect {
        int x;
        int y;
        int width;
        int height;
};
```

```
// 직원
struct employee {
        char name[20];          // 이름
        int age;                // 나이
        int gender;             // 성별
        int salary;             // 월급
};
```

구조체 변수 선언

- 구조체 **형태** 선언과 구조체 **변수** 선언은 다르다.
  - 변수 선언이 되어야 메모리 상에 영역을 점유
  - **struct tag_name** var_name



구조체 선언의 다른 방법들

-
```
struct tag_name {
.....
} ;
```

var_list를 생략하는 경우
- 변수 선언을 따로 하여야 한다
- 언제든지 필요시 tag_name을 이용하여 변수 선언 가능

-
```
struct {
.....
} var_list ;
```

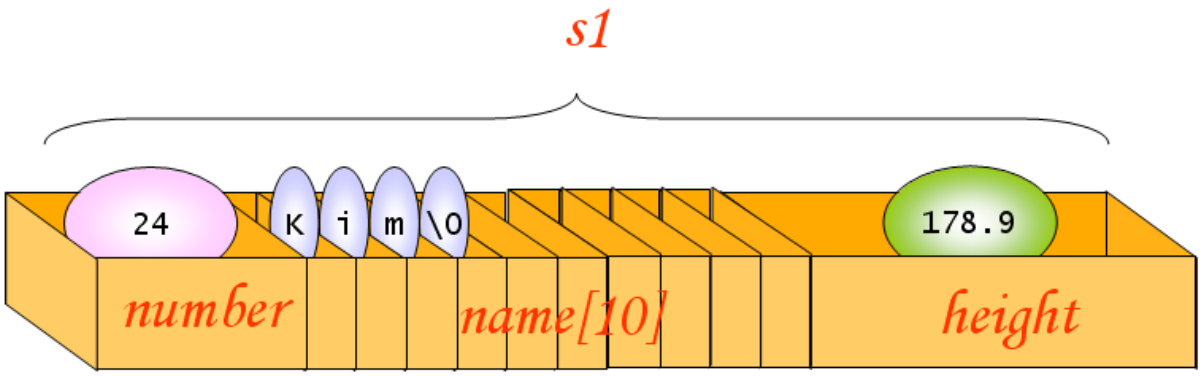
tag_name을 생략하는 경우
- 변수 선언을 하지 않아도 된다
- 구조체 선언과 동시에 var_list로 사용할 변수를 이미 선언
- 추후에 필요 시 구조체 변수를 더 이상 선언할 수 없다

-
```
struct tag_name {
.....
} var_list ;
```

구조체의 초기화

- 중괄호를 이용하여 초기값을 나열한다.

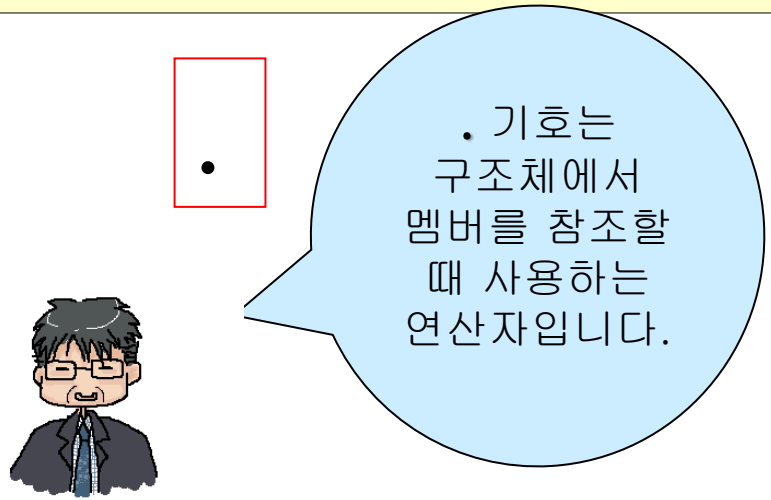
```
struct student {
                int number;
                char name[10];
                double height;
};
struct student s1 = { 24, "Kim", 178.9 };
```



구조체 멤버(혹은 field) 참조

- 구조체 특정 멤버를 참조하려면 다음과 같이 . 연산자를 사용한다.

```
s1.number = 26;          // 정수 멤버
strcpy(s1.name, "Kim");  // 문자열 멤버
s1.height = 183.2;                  // 실수 멤버
```



Nested Structure

- 구조체를 멤버로 가지는 구조체

```
struct date {                       // 구조체 선언
      int year;
      int month;
      int day;
};
```

```
struct student {                    // 구조체 선언
            int number;
            char name[10];
            struct date dob;     // 구조체 안에 구조체 포함
            double height;
};
struct      student    s1;          // 구조체 변수 선언
```

```
s1.dob.year = 1983;                 // 멤버 참조
s1.dob.month = 03;
s1.dob.day = 29;
```

예제 #1

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

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

int main(void)
{
        struct student s;

        s.number = 20070001;
        strcpy(s.name,"홍길동");
        s.height = 180.2;

        printf("학번: %d\n", s.number);
        printf("이름: %s\n", s.name);
        printf("신장: %f\n", s.height);

        return 0;
}
```

구조체 선언

구조체 변수 선언

구조체 멤버 참조

```
학번: 20070001
이름: 홍길동
신장: 180.200000
```

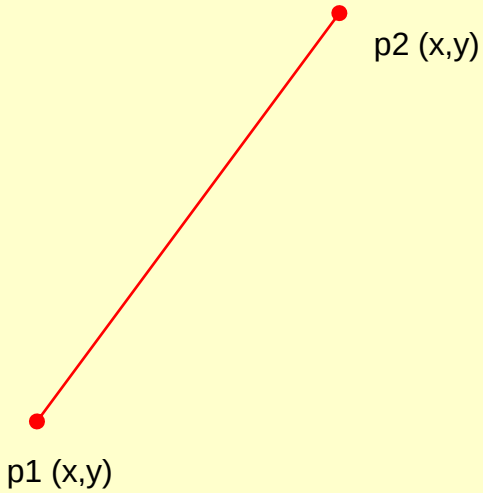
예제 #2

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

int main(void)
{
        struct student s;

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

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

        printf("신장을 입력하시오(실수): ");
        scanf("%lf", &s.height);

        printf("학번: %d\n", s.number);
        printf("이름: %s\n", s.name);
        printf("신장: %f\n", s.height);
        return 0;
}
```

```
학번을 입력하시오: 20070001
이름을 입력하시오: 홍길동
신장을 입력하시오(실수): 180.2
학번: 20070001
이름: 홍길동
신장: 180.200000
```

예제 #3

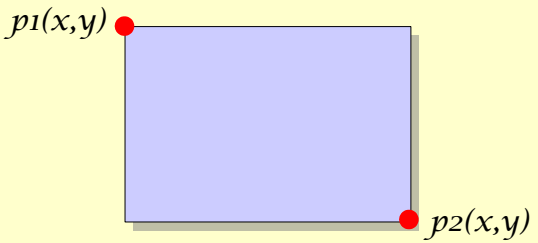
```c
#include <math.h>

struct point {
	int x;
	int y;
};

int main(void)
{
	struct point p1, p2;
	int xdiff, ydiff;
	double dist;

	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;

	dist = sqrt(xdiff * xdiff + ydiff * ydiff);

	printf("두 점사이의 거리는 %f입니다.\n", dist);

	return 0;
}
```




```
점의 좌표를 입력하시오(x y): 10 10
점의 좌표를 입력하시오(x y): 20 20
두 점사이의 거리는 14.142136입니다.
```

예제 #4

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

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

int main(void)
{
        struct rect r;
        int w, h, area, peri;

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

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

        w = r.p2.x - r.p1.x;
        h = r.p2.x - r.p1.x;

        area = w * h;
        peri = 2 * w + 2 * h;
        printf("면적은 %d이고 둘레는 %d입니다.\n", area, peri);

        return 0;
}
```




```
왼쪽 상단의 좌표를 입력하시오: 1 1
오른쪽 상단의 좌표를 입력하시오: 6 6
면적은 25이고 둘레는 20입니다.
```

구조체 변수의 대입과 비교

- 동일한 형태의 구조체 변수끼리 대입은 가능하지만 비교는 불가능하다

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

int main(void)
{
      struct point p1 = {10, 20};
      struct point p2 = {30, 40};

      p2 = p1;                                          // 대입 가능

      if( p1 == p2 )                                    // 비교 -> 컴파일 오류!!
            printf("p1와 p2이 같습니다.")

      if( (p1.x == p2.x) && (p1.y == p2.y) )            // 올바른 비교
            printf("p1와 p2이 같습니다.")
}
```

구조체 배열 (array of structures)

- 구조체 배열의 선언

```
struct student {
	int number;
	char name[20];
	double height;
};

int main(void)
{
	struct student list[100];	// 구조체의 배열 선언

	list[2].number = 27;
	strcpy(list[2].name, "홍길동");
	list[2].height = 178.0;
}
```

- 구조체 배열의 초기화

```
struct student list[3] = {
	{ 1, "Park", 172.8 },
	{ 2, "Kim", 179.2 },
	{ 3, "Lee", 180.3 }
};
```

구조체 배열 예제

```c
#define SIZE 3

struct student {
	int number;
	char name[20];
	double height;
};
int main(void)
{
	struct student list[SIZE];
	int i;

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

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

	return 0;
}
```

```
학번을 입력하시오: 20070001
이름을 입력하시오: 홍길동
신장을 입력하시오(실수): 180.2
학번을 입력하시오: 20070002
이름을 입력하시오: 김유신
신장을 입력하시오(실수): 178.3
학번을 입력하시오: 20070003
이름을 입력하시오: 이성계
신장을 입력하시오(실수): 176.3
학번: 20070001, 이름: 홍길동, 신장: 180.200000
학번: 20070002, 이름: 김유신, 신장: 178.300000
학번: 20070003, 이름: 이성계, 신장: 176.300000
```