

# 구조체

## 1. 강의

구조체를 정의, 선언, 값 대입, 출력 해볼 줄 안다.

### ▼ 구조체

- 구조체 : 서로 다른 데이터형의 변수들을 하나로 묶어서 사용하는 기능
- (배열 : 같은 데이터형의 변수들을 하나로 묶어서 사용)
- 구조체가 필요한 예시

```
// 상품 하나의 정보를 c로 구현할 경우
char name[20]
int price, stock

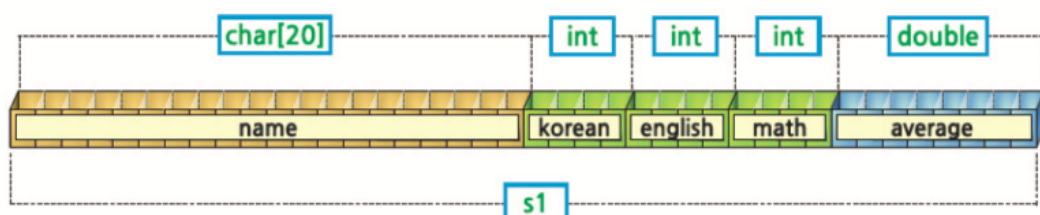
// 또 다른 상품 하나의 정보를 c에서 구현해야 한다면
char name2[20];
int price2, sotck2
```

이와 같이 해야 한다. 만약 상품이 10000 만개라면 어떻게 구현 할 것인가? ⇒ 구조체 사용

- 구조체 정의 형식

```
struct 태그명 {
    데이터형 멤버명;
    데이터형 멤버명;
    ...
};
//세미콜론을 반드시 붙인다.
(ex901.c)
```

```
struct student s1;
```



- 구조체 변수의 선언과 초기화

```
struct 태그명 변수명1, 변수명2, ...
```

예시>

```
struct product prd1;
struct point pt1, pt2;
```

(struct 키워드를 생략하면 에러가 난다.)

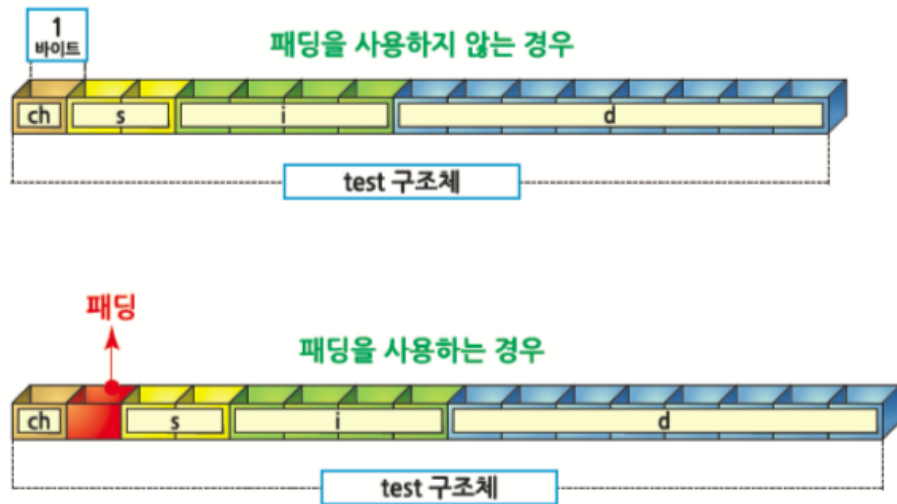
- 구조체 안의 멤버에 접근할 때는 멤버 접근 연산자 "." 을 사용한다.



- 예제 ex902.c 로 직접 확인해 보자.

#### ▼ 구조체의 크기

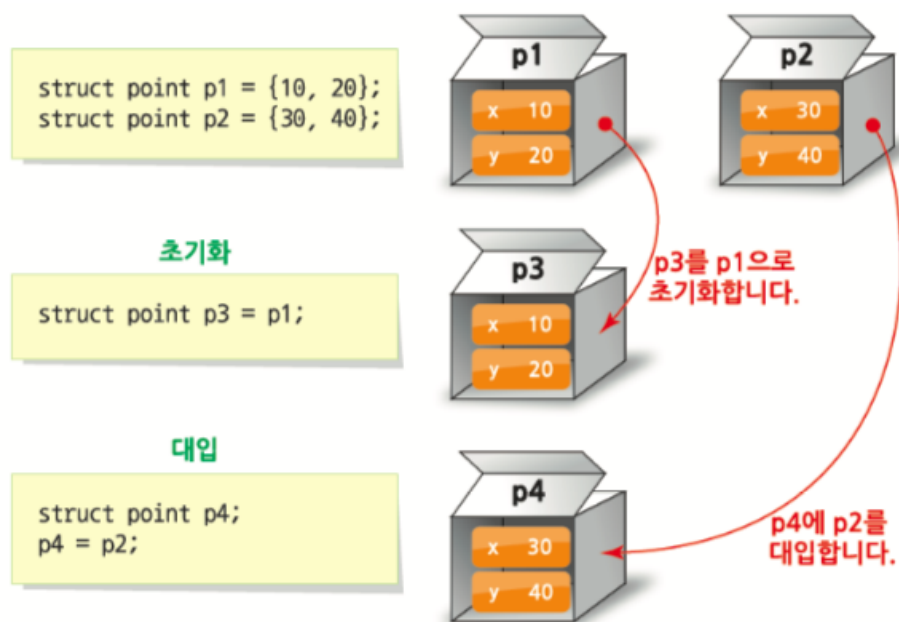
- 구조체의 크기는 모든 멤버들의 크기의 합보다 크거나 같다.
- c에서는 int 4 byte, double 8byte 처럼 자료형 단위로 메모리를 정렬하기 위해 구조체에서 정의하지 않은 바이트를 메모리에 삽입하는데 이를 패딩이라 한다.



- 만일 구조체 멤버의 크기가 15바이트 인경우에, 8바이트 정렬 방식으로 패딩을 삽입하면 16바이트가 된다.
- ex901.c 참조 (총 15 byte 지만, 1 byte 추가하여 16 byte를 할당)

#### ▼ 구조체 간의 초기화 및 대입

- 같은 구조체형의 변수들끼리는 서로 초기화나 대입이 가능



- 구조체 정의와 변수 선언을 동시에 할 수 있다. (ex904.c)

```
struct 구조체이름 {
    자료형 멤버이름;
```

```
} 변수;
```

```
// 위 방식으로 선언된 변수는 구조체의 각 멤버별로 초기값을 대입해줘야 한다.  
// 그렇지 않으면 에러가 발생할 수 있다.
```

- 구조체에 관계연산자를 사용할 경우에는 멤버대 멤버를 비교해야 한다.

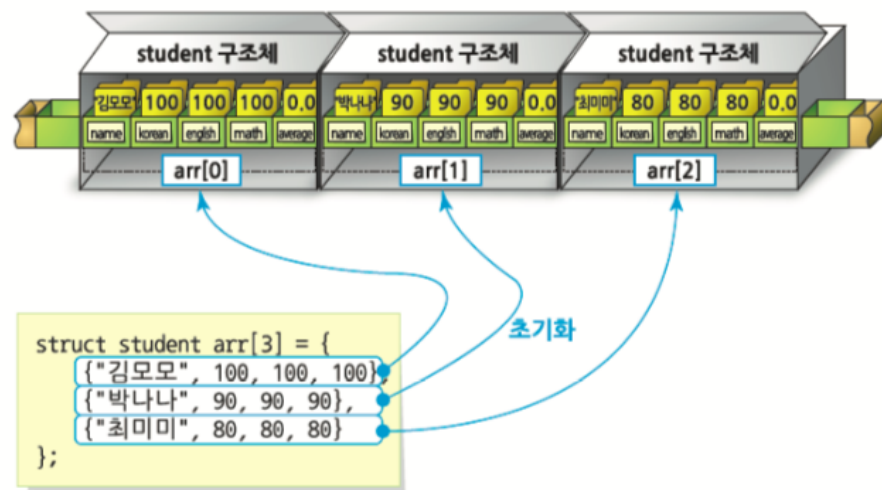
```
if (pt1 == pt2) // error
```

```
if (pt1.x == pt2.x)
```

```
(ex903.c)
```

### ▼ 구조체 배열

- 구조체 배열도 사용이 가능하며, 멤버의 원소 접근은 배열 접근 방법과 같다.  
(ex905.c)



### ▼ 구조체 포인터

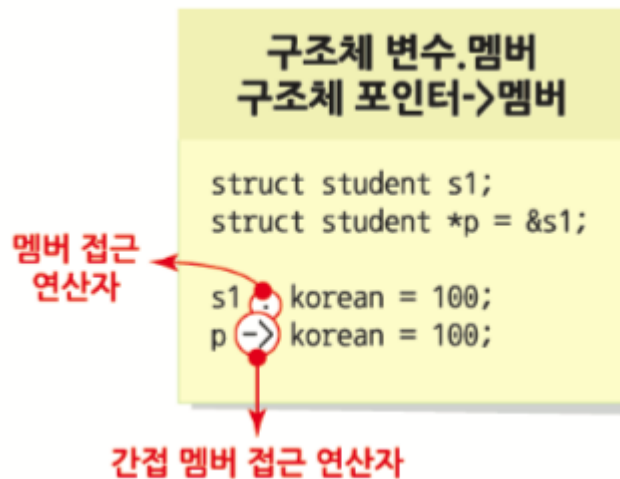
- 구조체 포인터는 구조체 변수의 주소를 저장하는 포인터이다.
- 포인터 수식어 \* 를 구조체 변수명의 앞에 적어 구조체 포인터를 만든다.
- 구조체 변수의 주소도 & 연산자로 구하며, 구한 값을 형이 일치하는 구조체 포인터 변수에 저장할 수 있다.

```
struct product prd1 = {\"coeffee\", 1000, 3};  
struct product *p = &prd1;
```

- 구조체 변수의 멤버 값에 접근하기 위해, 간접 참조 연산자 \* 를 사용하며, (\*p).name 과 같이 소괄호로 감싸고, 멤버 이름을 적는다.

```
printf("%s \n", (*p).name);
(*p).price = 1000;
```

- 간접 멤버 접근 연산자 → 를 사용해 p → name 도 (\*p).name 과 동일한 결과를 얻는다. (ex907.c)



```
printf("%s \n", p->name);
p->price = 1000;
```

## ▼ typedef

- 데이터형에 이름을 부여해 사용할 수 있게 해준다.

```
typedef 기존데이터형 새이름;

typedef unsigned int UINT;
typedef unsigned char BYTE;
typedef struct point POINT;
//ex908.c
```

- 프로그램의 가독성과 이식성을 위해 사용

## ▼ 공용체

```
typedef union ubox{
    int mem1;    //4 byte
    int mem2;    //4 byte
    double mem3; //8 byte
}UBOX;

int main(){
    UBOX ubx;
    ubx.mem1 = 20;
```

## 2. 실습

### ▼ ex903.c

1. 구조체 태그명 : point, 멤버 : 정수형 x, y 선언
2. 구조체 변수 : pt1, pt2
3. pt1, pt2 의 x, y 값을 scanf 로 입력 받으시오.
4. pt1, pt2 의 x, y를 각각 출력해 보시오.
5. pt1 과 pt2 의 x, y 가 서로 같은지 검사하여 결과를 출력 하시오.

### ▼ ex906.c

1. ex905.c 의 프로그램을 변경하여, stock <20 이하인 경우, "재고 부족"이라는 경고가 출력되게 하시오.
2. %-20s : 왼쪽부터 문자 출력하는데 전체 칸수는 20칸이 되게 출력

### ▼ ex909.c

1. 구조체를 사용 : 태그명 = person, 멤버 : name[20], age, address[100]
2. 홍길동, 20, 대한민국 서울 자료를 입력
3. 위의 내용을 출력 하는 프로그램을 작성하시오.

### ▼ ex910.c

ex909.c 프로그램을 typedef 를 사용하여 변경해 보시오.

### ▼ 포인터 실습