



[9주차 1강] 구조체(5)



학습 내용

📍 11.6 중첩 구조체 및 자기참조 구조체 (2)

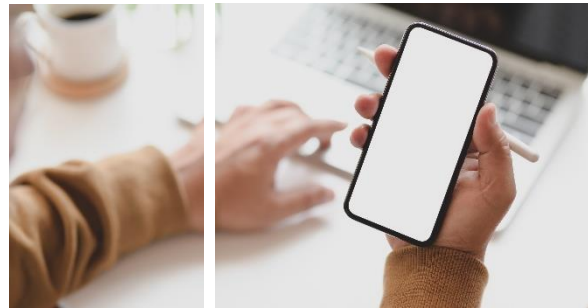
📍 11.7 typedef 사용자 형정의





학습 목표

- 📍 11.6 자기참조 구조체가 무엇인지 이해한다.
- 📍 11.7 typedef를 이용한 사용자 자료형 정의 방법을 익힌다.





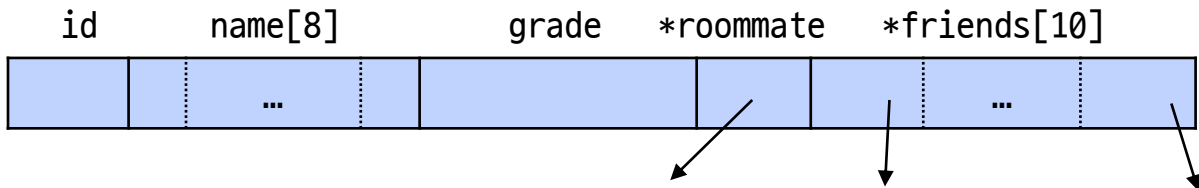
1. 구조체 개요
2. 구조체의 정의, 선언, 사용
3. 구조체 배열
4. 구조체 포인터
5. 구조체와 함수
6. 중첩 구조체 및 자기참조 구조체 (2)
7. typedef 사용자 형정의

6. 중첩 구조체 및 자기참조 구조체

자기참조 구조체

- 자신과 동일한 구조체의 "포인터"는 멤버로 가능
 - ✓ 아래에서 roommate는 포인터 변수로 주소 값을 저장하므로 struct student의 정의에 포함 가능 (메모리 그림 그릴 수 있음)

```
struct student7{ // 구조체 student 정의
    int id;   char name[8];   double grade;
    struct student7 *roommate; // 문제 없음
    struct student7 *friends[10]; // 문제 없음
};
```



6. 중첩 구조체 및 자기참조 구조체

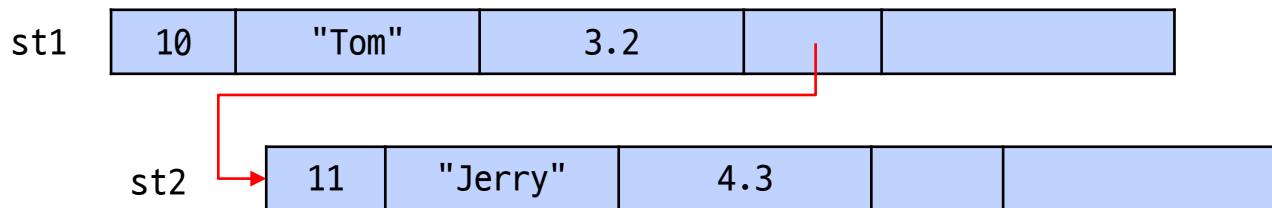


자기참조 구조체 사용 예

```
struct student7 st1 = {10, "Tom", 3.2}; // roommate와 friends
struct student7 st2 = {11, "Jerry", 4.3}; // NULL로 초기화

st1.roommate = &st2; // st1의 룸메이트는 st2

printf("id: %d\n", st1.id);
printf("name: %s\n", st1.name);
printf("grade: %.2f\n", st1.grade);
printf("roommate's name: %s\n", st1.roommate->name);
```



※ 실습하기



[예제 11.8] struct student7을 사용하여 3명의 학생을 나타내는 변수 3개를 선언하고, 한 학생의 친구로 다른 두 명의 학생을 등록하라.

- 구조체 멤버는 적절히 초기화 or 값 대입
- friends 멤버를 이용하여 친구의 정보 출력

실행 결과(예시)

000과 000는 000의 친구입니다.





1. 구조체 개요
2. 구조체의 정의, 선언, 사용
3. 구조체 배열
4. 구조체 포인터
5. 구조체와 함수
6. 중첩 구조체 및 자기참조 구조체 (2)
7. typedef 사용자 형정의

7. typedef 사용자 형정의

사용자 형정의

- typedef문을 이용하여 자료형의 이름을 새로 정의할 수 있음
- 예 `typedef int INT; // INT 자료형 정의 (마지막에 세미콜론)`
 - ✓ int라는 자료형을 INT라는 이름으로 정의
 - ✓ 주의!! 위 정의에서 INT는 변수 이름이 아니고 **자료형 이름**
 - ✓ INT를 이용한 변수 선언

```
INT num;           // INT형 변수 num 선언
INT arr[5];        // INT형 배열 변수 arr 선언
INT *pi;           // INT형 포인터 변수 pi 선언
```

7. typedef 사용자 형정의



다양한 사용자 형정의의 예

```
typedef int INT;  
typedef int * INTPTR;  
typedef unsigned int AGE;  
typedef unsigned int ID;  
typedef unsigned char UCHAR;  
typedef unsigned char * UCHARPTR;  
  
AGE age1, age2;  
ID id1, id2;  
UCHARPTR p;           // unsigned char *p; 와 동일한 선언
```

- ✓ 위 예에서 AGE, ID 모두 unsigned int 이지만,
위와 같은 식으로 자료형에 의미 부여 가능
- ✓ 긴 이름을 간결하게 줄일 수 있음



7. typedef 사용자 형정의



구조체와 typedef

- 구조체의 경우 키워드 struct로 인해서 선언이 길어짐
- typedef문을 이용하여 자료형의 이름을 재정의하여 프로그래밍의 간결성 확보

```
struct student{  
    int id;  char name[8];  double grade;  
};  
  
typedef struct student STUDENT;           // 자료형 재정의  
  
int main() {  
    STUDENT st1 = {10, "Tom", 3.2};  
  
    printf("id: %d\n", st1.id);           // 구조체 멤버의 값  
    return 0;  
}
```



7. typedef 사용자 형정의



구조체 재정의의 다양한 형태

- typedef와 구조체 정의를 하나로 묶기 (이 형태를 많이 사용)

```
typedef struct student{  
    int id;   char name[8];   double grade;  
} STUDENT;    // 자료형 정의  
STUDENT st;    // STUDENT 형으로 변수 st 선언
```

```
typedef struct student{  
    int id;   char name[8];   double grade;  
} student;    // 구조체 이름과 사용자 자료형 이름이 같아도 됨  
student st;    // student 형으로 변수 st 선언
```

```
typedef struct student{    // 구조체 이름 생략 가능  
    int id;   char name[8];   double grade;  
} student;  
student st;    // student 형으로 변수 st 선언
```



7. typedef 사용자 형정의



주의!! typedef 사용자 정의와 구조체 정의 비교

- 모양이 유사하지만, typedef가 앞에 있느냐 없느냐에 따라 의미가 완전히 달라짐

```
typedef struct student{  
    int id;   char name[8];   double grade;  
} STUDENT;           //STUDENT는 자료형
```

```
struct student{  
    int id;   char name[8];   double grade;  
} st;         // st는 변수
```



※ 실습하기



[예제 11.9] 예제 11.1에서 정의한 구조체의 '배열'을 이용하여 2개의 런치 박스의 정보를 사용자로부터 입력 받고 출력하는 프로그램을 작성하시오. (예제 11.4와 동일한 문제)

- 단, "예제 11.1에서 정의한 구조체"를 **typedef**를 이용하여 사용자 자료형으로 정의할 것 (자료형 이름은 알아서 정하기)

입력

30
3
5
0
10
40
3
5
0
20

출력

30
3
5
0
10
40
3
5
0
20



- **자기참조 구조체**는 자신과 동일한 구조체에 대한 포인터를 멤버로 사용하는 구조체
- 사용자 자료형은 **typedef** 키워드를 사용하여 정의
- typedef 사용자 정의와 구조체 정의의 모양이 유사하지만, typedef의 유무에 따라 의미가 완전히 달라지므로 주의 필요