



# [7주차 1강] 구조체(3)



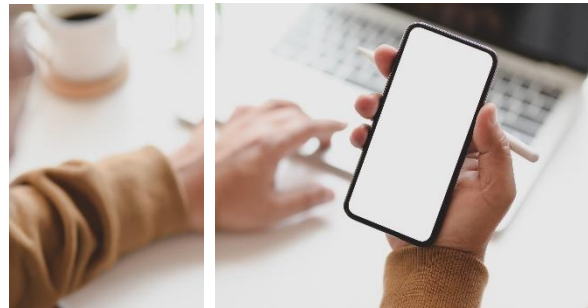
# 학습 내용

## 11.4 구조체 포인터



# 학습 목표

📍 11.4 구조체와 포인터의 관계를 이해한다.





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



## 4. 구조체 포인터



### 구조체 포인터

- 구조체 변수를 가리키는 포인터
- 즉, 구조체 변수의 시작 주소가 저장



### 기본적인 사용법은 일반적인 포인터와 동일

- 다만, 구조체 포인터에서만 사용하는 표현법 존재

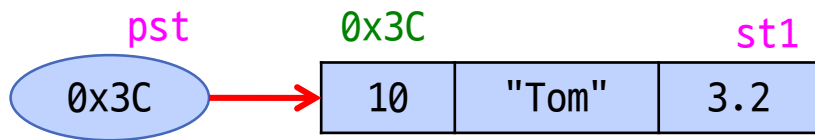


## 4. 구조체 포인터

### 구조체 포인터 변수 선언 및 연결

- 일반 포인터 선언과 동일: \* 사용
- 주소 연산자(&): 구조체 변수의 시작 주소

```
struct student st1 = { 10, "Tom", 3.2};  
struct student *pst; // 구조체 포인터 변수 선언  
pst = &st1;         // 연결
```



## 4. 구조체 포인터



### 간접 연산자(\*)를 이용한 구조체 변수 접근

```
struct student st1 = { 10, "Tom", 3.2}, st2;  
struct student *pst = &st1; // 선언 및 연결  
  
st2 = *pst; // st2에 pst가 가리키는 구조체 변수를 대입  
printf("%d, %s, %.2f\n", st1.id, st1.name, st1.grade);  
printf("%d, %s, %.2f\n", st2.id, st2.name, st2.grade);
```

실행 결과

10,Tom,3.20

10,Tom,3.20



## 4. 구조체 포인터



### 간접 참조를 이용한 구조체 변수의 **멤버** 접근 (방법 1)

- 포인터가 가리키는 변수에 접근하기 위해 간접연산자 \* 사용
- 구조체의 멤버에 접근하기 위해 멤버 연산자 . 사용

```
struct student st1, *pst = &st1;  
  
(*pst).id = 20; // pst가 가리키는 구조체의  
               // 멤버 id에 20 대입  
  
printf("id: %d\n", (*pst).id + 15 );
```

실행 결과

id: 35

- \* 보다 . 의 우선순위가 높기 때문에 괄호 반드시 필요





## 4. 구조체 포인터

### 간접 참조를 이용한 구조체 변수의 **멤버** 접근 (방법 2)

- 구조체 포인터에서만 사용하는 전용 연산자: **->**
- 주로 이 연산자를 사용함

```
(*pst).id = 20; // pst가 가리키는 구조체의  
                // 멤버 id에 20 대입
```

||

```
pst->id = 20;   // pst가 가리키는 구조체의  
                // 멤버 id에 20 대입
```



## 4. 구조체 포인터



### 프로그램 1의 포인터 버전

- 간접 참조로만 구조체에 접근하도록 수정한 코드

```
struct student st1 = {10, "Tom", 3.2}; // 변수 선언 및 초기화
struct student *pst = &st1;           // 포인터 선언 및 연결

pst->id += 20;      // pst가 가리키는 구조체의 멤버 id에 접근
strcpy(pst->name, "alice"); // 멤버 name에 접근
pst->name[0] = 'A'; // 멤버 name[0]에 접근

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



# ※ 실습하기



[예제 11.5] 예제 11.1을 구조체 포인터를 사용하여 작성하시오.

- 구조체 변수에 직접 접근하지 말고, 포인터를 이용하여 간접 접근할 것

```
struct lunchbox{  
    int maindish  
    int sidedish[3];  
    ...  
};  
  
int main() {  
    struct lunchbox box, *p = &box;
```

이후 포인터 p를 사용하여 코드 작성

```
    ...  
}
```



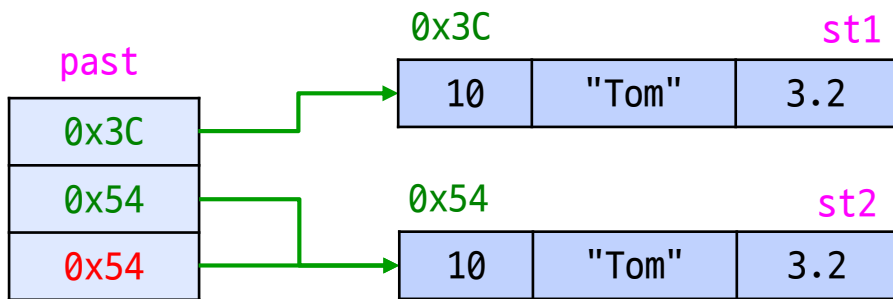
## 4. 구조체 포인터



### 구조체 포인터 배열

- 구조체 포인터가 원소인 배열

```
struct student st1 = { 10, "Tom", 3.2 }, st2;  
struct student *past[3] = { &st1, &st2 }; // 구조체 포인터 배열  
  
past[2] = past[1]; // 주소 값 대입: past[2]도 st2를 가리킴  
*past[2] = *past[0]; // past[2]가 가리키는 구조체에  
                        past[0]이 가리키는 구조체 대입
```





# 학습 정리

- **구조체 포인터**는 구조체를 가리키는 포인터
- 구조체 포인터를 사용하여 멤버에 간접 접근하기 위해서는 **구조체 포인터 전용 참조 연산자 ->**를 사용
- **구조체 포인터 배열**은 구조체 포인터가 원소인 배열

