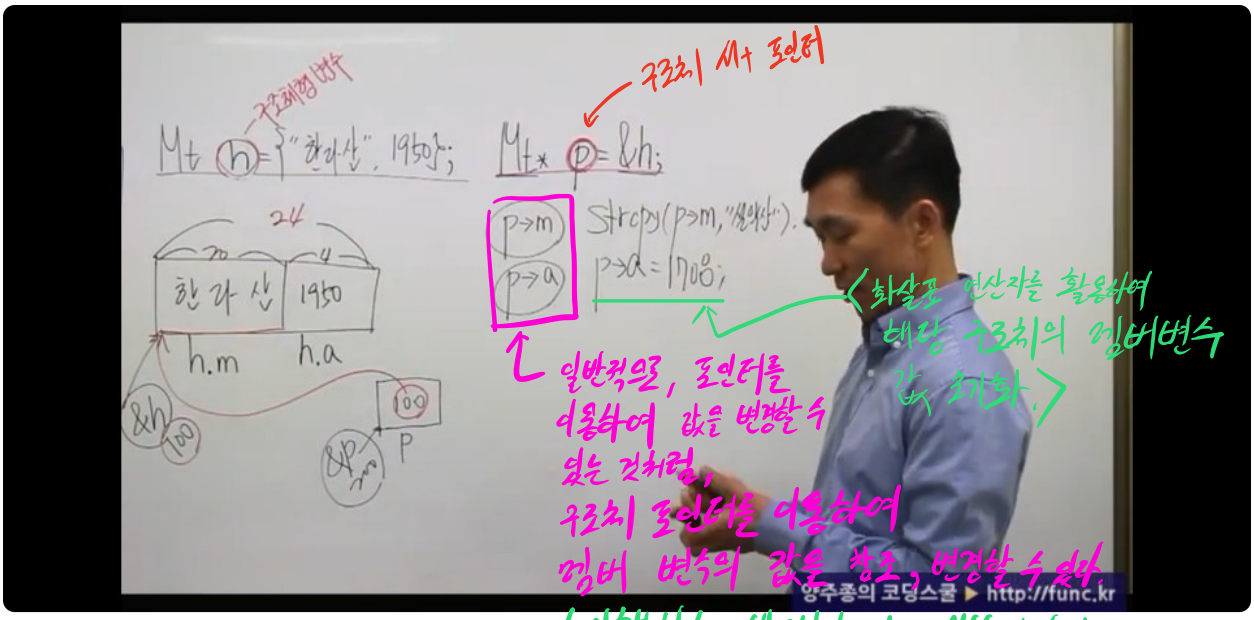
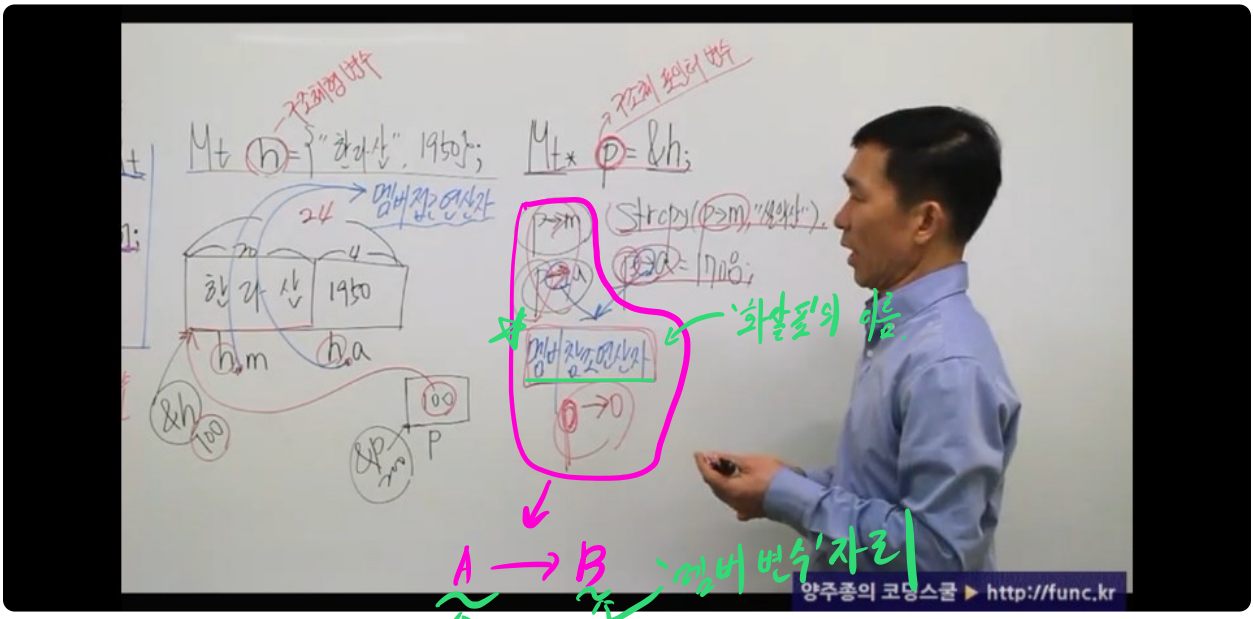


화살표 연산자를 활용하여 해당 구조체의 멤버변수 값 참조.

일반적으로, 포인터를 이용하여 값을 변경할 수 있는 것처럼, 구조체 포인터를 이용하여 멤버 변수의 값을 참조, 변경할 수 있다. (사용하는 방법이 다를 뿐이다.)



A → B : 멤버 변수 자리

A : 구조체 포인터 자리

pointer_to_struct.c

```c
#include <stdio.h>

struct Person {    // 구조체 정의
    char name[20];        // 구조체 멤버 1
    int age;              // 구조체 멤버 2
    char address[100];    // 구조체 멤버 3
};

int main()
{
    struct Person p1;      // 구조체 변수 선언
    struct Person *ptr;    // 구조체 포인터 선언

    ptr = &p1;    // p1의 메모리 주소를 구하여 ptr에 할당

    // 화살표 연산자로 구조체 멤버에 접근하여 값 할당
    ptr->age = 30;

    printf("나이: %d\n", p1.age);      // 나이: 30: 구조체 변수의 멤버 값 출력
    printf("나이: %d\n", ptr->age);    // 나이: 30: 구조체 포인터의 멤버 값 출력

    return 0;
}
```

'Person' 구조체를 통해 생성된 'p1'이라는 인스턴스.

실행 결과

```
나이: 30
나이: 30
```

먼저 `struct Person p1;`과 같이 구조체 변수를 선언하고, `struct Person *ptr;`과 같이 구조체 포인터를 선언했습니다. 아직까지는 `p1`과 `ptr`은 관계가 없습니다.

구조체 변수는 주소 연산자 `&`를 사용하여 메모리 주소를 구할 수 있으며 이렇게 구한 메모리 주소는 구조체 포인터에 할당할 수 있습니다.

```c
ptr = &p1;    // p1의 메모리 주소를 구하여 ptr에 할당
```

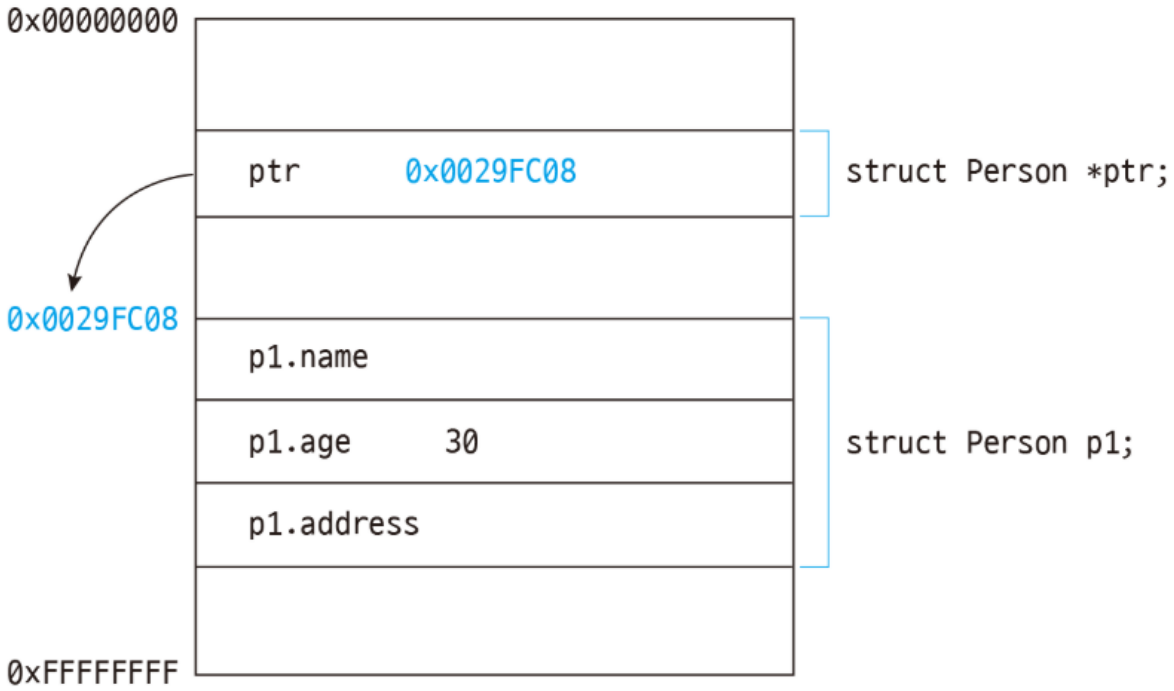
핵심!!!

ptr은 구조체 포인터이므로 ->로 멤버에 접근하여 값을 할당하였습니다. 그리고 printf 함수로 p1의 멤버와 ptr의 멤버를 출력해보면 같은 값이 나오는 것을 알 수 있습니다.

```
// 화살표 연산자로 구조체 멤버에 접근하여 값 할당
ptr->age = 30;

printf("나이: %d\n", p1.age);      // 나이: 30: 구조체 변수의 멤버 값 출력
printf("나이: %d\n", ptr->age);    // 나이: 30: 구조체 포인터의 멤버 값 출력
```

즉, ptr에 p1의 메모리 주소를 할당했으므로 ptr의 멤버를 수정하면 결국 p1의 멤버도 바뀝니다. 접근하는 방식만 차이가 있을 뿐 결국 같은 곳의 내용을 수정하게 됩니다(메모리 주소는 컴퓨터마다, 실행할 때마다 달라집니다).

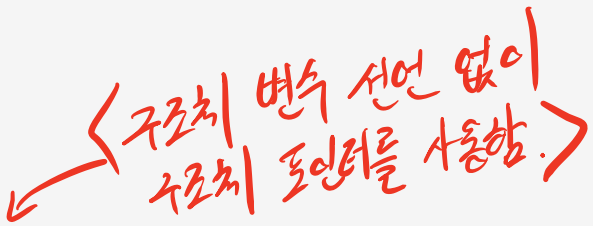
▼ 그림 49-3 구조체 변수의 주소와 구조체 포인터



지금까지 구조체 포인터의 사용 방법을 배웠습니다. 여기서는 구조체 포인터에 malloc으로 메모리를 할당하고 free로 해제한다는 점과 멤버에 접근할 때 ->를 사용한다는 점만 기억하면 됩니다.

struct_alloc_memory.c

```c
#define _CRT_SECURE_NO_WARNINGS    // strcpy 보안 경고로 인한 컴파일 에러 방지
#include <stdio.h>
#include <string.h>    // strcpy 함수가 선언된 헤더 파일
#include <stdlib.h>    // malloc, free 함수가 선언된 헤더 파일

struct Person {    // 구조체 정의
    char name[20];        // 구조체 멤버 1
    int age;              // 구조체 멤버 2
    char address[100];    // 구조체 멤버 3
};

int main()
{
    struct Person *p1 = malloc(sizeof(struct Person));    // 구조체 포인터 선언, 메모리 할당

    // 화살표 연산자로 구조체 멤버에 접근하여 값 할당
    strcpy(p1->name, "홍길동");
    p1->age = 30;
    strcpy(p1->address, "서울시 용산구 한남동");

    // 화살표 연산자로 구조체 멤버에 접근하여 값 출력
    printf("이름: %s\n", p1->name);       // 홍길동
    printf("나이: %d\n", p1->age);        // 30
    printf("주소: %s\n", p1->address);    // 서울시 용산구 한남동

    free(p1);    // 동적 메모리 해제

    return 0;
}
```



실행 결과

```
이름: 홍길동
나이: 30
주소: 서울시 용산구 한남동
```