포인터란?

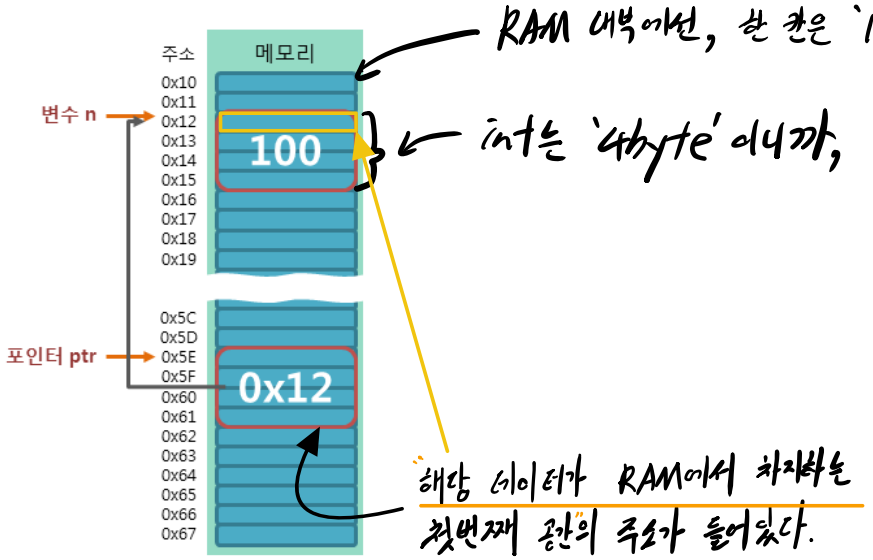
C언어에서 포인터(pointer)란 메모리의 주소값을 저장하는 변수이며, 포인터 변수라고도 부릅니다.
char형 변수가 문자를 저장하고, int형 변수가 정수를 저장하는 것처럼 포인터는 주소값을 저장합니다.

예제

```
int n = 100;   // 변수의 선언
int *ptr = &n; // 포인터의 선언
```

• '*ptr'로 변수를 호출하면, 100 이 호출됨.

다음 그림은 위의 예제에서 사용된 변수와 포인터가 메모리에서 어떻게 저장되는지를 보여주는 예제입니다.



RAM 내부에선, 한 칸은 '1byte'이다.

int는 '4byte'이니까, 네 칸을 차지한다.

해당 데이터가 RAM에서 차지하는 첫번째 공간의 주소가 들어있다.

• 포인터를 사용하는 이유 : ① 매개변수에 '얕은 복사'를 실시하기 위해서 (= call by reference)

② 여러 자료구조를 구현할 때. (ex) Linked List ...)

※ 함수의 매개변수에 '기본자료형'을 전달하면,

★ 해당 데이터(변수)가 복사된 후, 매개변수에 저장된다.

→ 데이터를 복사했기 때문에, 추가적인 메모리 공간이 할당됨.

① 포인터 변수를 선언할 때는 자료형 뒤에 * (Asterisk, 애스터리스크)를 붙입니다. *의 위치에 따른 차이는 없으며 모두 같은 뜻입니다.

```
int* numPtr;     // 자료형 쪽에 *을 붙임
int * numPtr;    // 자료형과 변수 가운데 *를 넣음
int *numPtr;     // 변수 쪽에 *을 붙임
```

② 포인터 변수를 선언했으면 다음과 같이 &로 변수의 주소를 구해서 포인터 변수에 저장합니다.

```
numPtr = &num1;   // num1의 메모리 주소를 포인터 변수에 저장
```

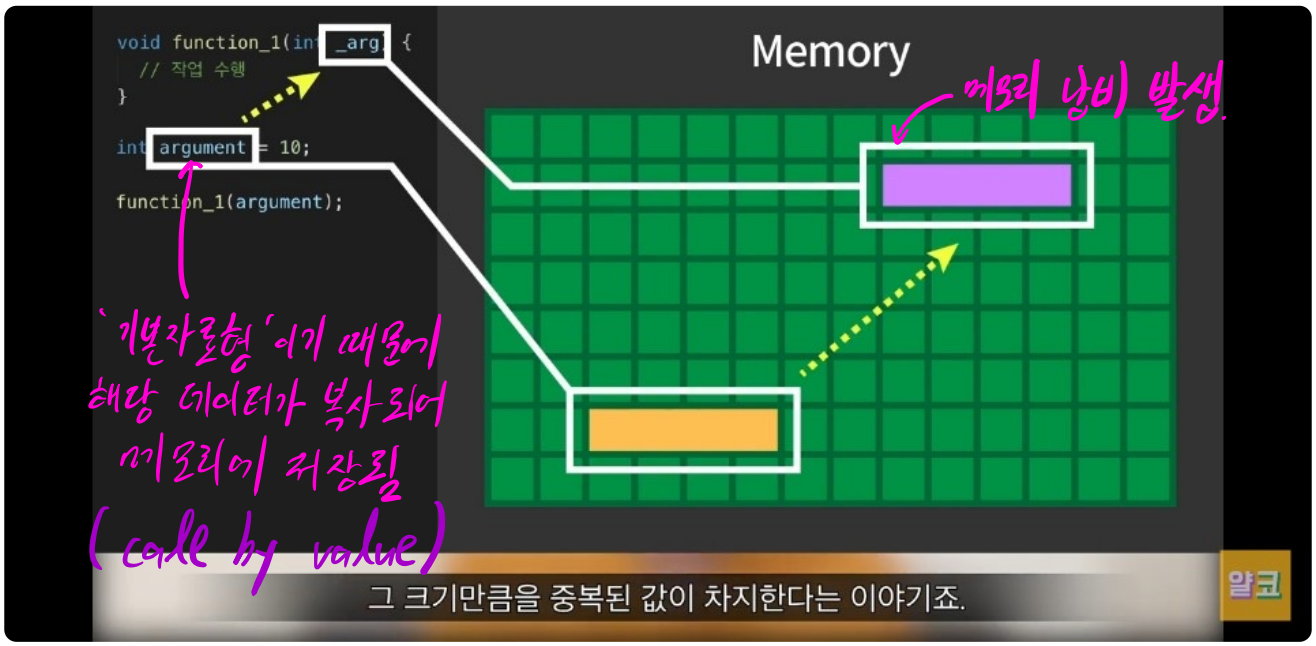
이제 printf로 포인터 numPtr의 값을 출력해보면 변수 num1의 메모리 주소가 나옵니다. 즉, 포인터와 메모리 주소는 같은 의미입니다.

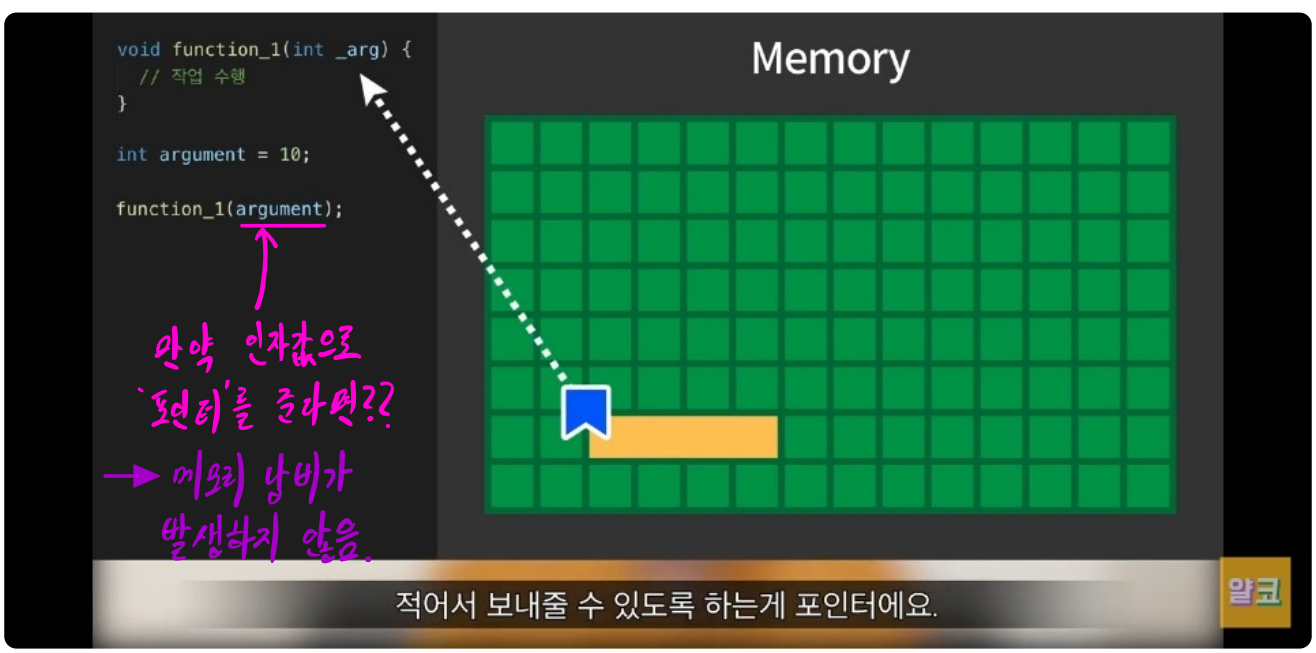
```
printf("%p\n", numPtr);    // 0055FC24: 포인터 변수 numPtr의 값 출력
                           // 컴퓨터마다, 실행할 때마다 달라짐
printf("%p\n", &num1);     // 0055FC24: 변수 num1의 메모리 주소 출력
                           // 컴퓨터마다, 실행할 때마다 달라짐
```

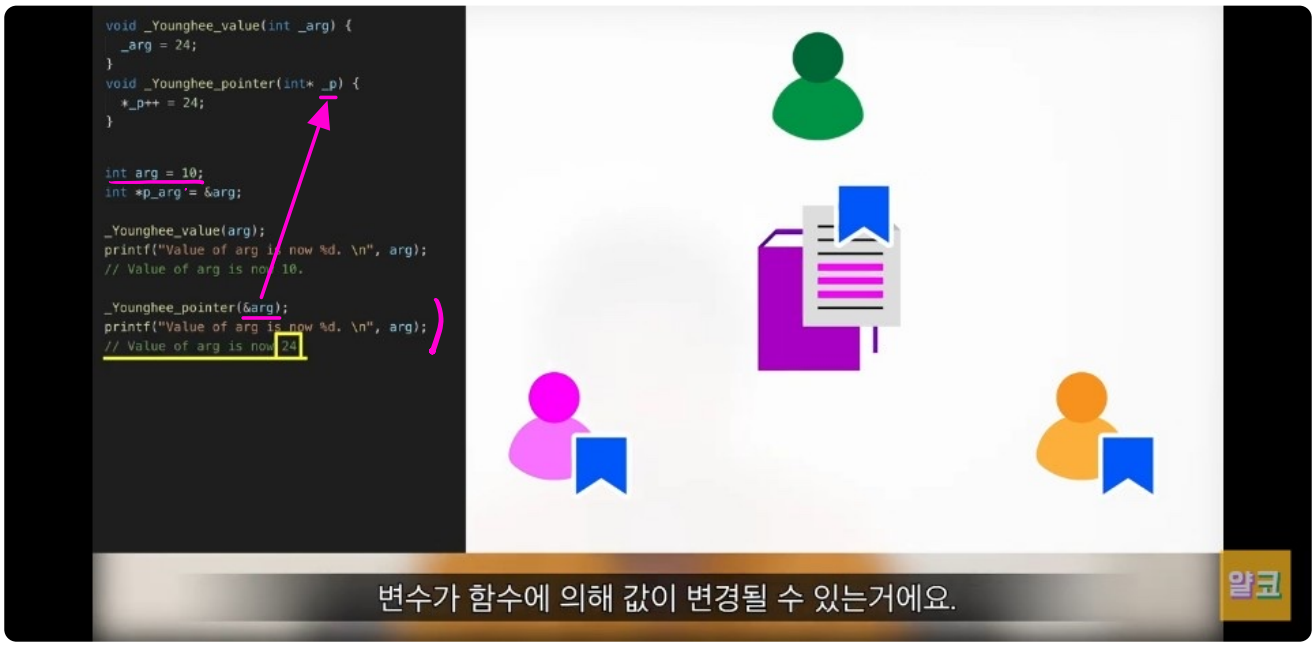
포인터 변수를 선언할 때는 자료형을 알려주고 *를 붙이는 방식을 사용합니다. 만약 변수가 int형이면 이 변수의 메모리 주소를 저장하는 포인터는 int *라야 합니다.

여기서 int *는 영어로 pointer to int라고 읽는데 int형 공간을 가리키는 포인터라는 뜻입니다(간단하게 int 포인터라고도 부릅니다).

※ 포인터는 자료형에 상관없이 무조건 '4byte'임.


void function_1(int _arg) {
// 작업 수행
}
int argument = 10;
function_1(argument);
Memory
메모리 낭비 발생.
`기본자료형`이기 때문에 해당 데이터가 복사되어 메모리에 저장됨 (call by value)
그 크기만큼을 중복된 값이 차지한다는 이야기죠.
알코


void function_1(int _arg) {
// 작업 수행
}
int argument = 10;
function_1(argument);
Memory
만약 인자값으로 `포인터`를 준다면??
→ 메모리 낭비가 발생하지 않음.
적어서 보내줄 수 있도록 하는게 포인터에요.
알코


void _Younghee_value(int _arg) {
_arg = 24;
}
void _Younghee_pointer(int* _p) {
*_p++ = 24;
}
int arg = 10;
int *p_arg = &arg;
_Younghee_value(arg);
printf("Value of arg is now %d. \n", arg);
// Value of arg is now 10.
_Younghee_pointer(&arg);
printf("Value of arg is now %d. \n", arg);
// Value of arg is now 24
변수가 함수에 의해 값이 변경될 수 있는거에요.
알코


이처럼, 특정 변수나 객체의 값을
알코

(=주소)

parameter_pointer.c

```c
#include <stdio.h>

void swapNumber(int *first, int *second)    // 반환값 없음, int 포인터 매개변수 두 개 지정
{
    int temp;    // 임시 보관 변수

    // 역참조로 값을 가져오고, 값을 저장함
    temp = *first;
    *first = *second;
    *second = temp;
}

int main()
{
    int num1 = 10;
    int num2 = 20;

    swapNumber(&num1, &num2);        // &를 사용하여 num1과 num2의 메모리 주소를 넣어줌

    printf("%d %d\n", num1, num2);   // 20 10: swapNumber에 의해서 num1과 num2의 값이 서로 바
꿈

    return 0;
}
```

실행 결과

```
20 10
```