



EDDI

Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv1 과정

제 3기

2022. 02. 01

김원석

CONTENTS

- 포인터란?
- 포인터와 배열
- 포인터 배열과 배열 포인터
- 배열 포인터 예시

포인터란?

- 포인터 == 주소
- 포인터 변수는 특정 데이터 타입의 주소를 저장할 수 있는 메모리 공간을 말한다.

포인터 변수 선언 예) `int *a;`

포인터 변수와 *

- 데이터 타입에 *까지 붙인 것이 포인터의 데이터 타입이다.
- 데이터 타입이 없이 *을 붙인 것은 해당 주소에 접근하라는 명령이다.

ex) `int * a;` -> int 타입 변수를 가리키는 포인터 변수 선언

`*a = 4;` -> a에 저장된 주소에 접근해 4 저장

```
int num_arr[] = { 1, 2, 3, 4, 5, 6, 7 };  
int *p_arr = NULL;
```

- 배열의 이름은 배열의 시작 주소와 같다.
- 포인터 변수에 배열의 시작 주소를 저장하면

```
p_arr = num_arr;
```

- p_arr[0], p_arr[1], ... 과 같이 포인터를 활용하여 배열을 제어할 수 있다.

포인터 배열과 배열 포인터

포인터 배열

- 포인터 배열은 말 그대로 포인터들의 배열이다.
- 특정 데이터 타입의 변수들의 포인터들을 저장할 수 있다.

배열 포인터

- 배열 포인터는 특정 데이터 타입의 배열의 포인터를 저장하는 변수를 말한다.

포인터 배열과 배열 포인터

```
int *p2[3];  
int (*p)[3];
```

→ 포인터 배열
→ 배열 포인터

- 포인터 배열의 선언은 일반 배열의 선언과 방식이 같다. 하지만 배열 포인터는 선언 방식이 특이하다. 배열 포인터의 선언 방식에 대해 알아보자.

배열 포인터의 선언

- `int (*p)[3]`은 실제로 `(int[3] *) p`와 같은 의미이다.
- 즉, `int` 타입 변수 3개로 이루어진 배열을 하나의 데이터 타입으로 하는 변수의 포인터를 저장하는 변수가 `p`라는 뜻이다.
- `(int[3] *) p`와 같은 형태가 의미상 자연스럽지만, 언어 특성상 `[3]`과 `*`의 순서를 바꾸지 못한다.
- 그래서 괄호를 이용해 포인터 배열의 선언 방식과 구분한다.

배열 포인터 예시

```
#include <stdio.h>

int main(void)
{
    int i, j;
    int double_arr[4][2] = {
        { 1, 3 }, { 4, 5 },
        { 6, 7 }, { 8, 9 }
    };

    int *p2[3];          // 포인터 배열
    int (*p)[3];         // 배열 포인터
    // (int[3] *) p <---- 실제로 이와 같은 형태를 가짐

    p = double_arr;

    printf("p[0] = %d\n", *p[0]);
    printf("p[1] = %d\n", *p[1]);
    printf("p[2] = %d\n", *p[2]);

    return 0;
}
```

1 5 8 이 출력된다.