



## c언어 - HW1

임베디드스쿨1기

Lv1과정

2020. 04. 23

이충재

---

\* 배열 0으로 초기화하기: 자료형 배열명[크기] = { 0, };

ex) int arr[10] = { 0, }; → arr[0] ~ arr[9] 전부 0이 된다.

\* 배열을 포인터에 넣기

배열은 첫번째 요소의 주소를 가진다.

배열도 주소 값이기 때문에 포인터에 넣을 수 있다.

ex) int arr[3];

int\* pointer = arr;

→ pointer 포인터에 arr 배열을 넣었다.

---

## 포인터에 이중배열 넣기

```
Int array[3][3];  
Int** ppointer = array;
```

이중배열을 이중포인터에 넣을 수 있을것 같지만 실제로 프로그램을 실행해보면 안된다.

```
Int array[2][3];  
Int (*pointer)[3] = array;
```

포인터에 이중배열을 넣기 위해서는 다음과 같은 특별한 방법으로 포인터를 만들어야한다.

왼쪽 예시는 두번째 인덱스의 크기가 3인 배열을 가리키는 포인터를 나타낸다.

---

\* 포인터에 동적 메모리 할당하기

malloc함수를 사용하면 메모리를 할당할수 있다.

ex) `int* pointer = malloc(sizeof(int) * 4);`

위 예시는 pointer포인터에 int자료형의 크기의 4배만큼 메모리를 할당하였다.

포인터를 배열처럼 사용하기.

포인터에 동적메모리를 할당하면 배열처럼 사용 할 수 있다.

ex) `int* p = malloc(sizeof(int) * 4);`  
`P[0] = 1;`  
`p[1] = 2;`

왼쪽 예시는 포인터p에 int크기의 공간 4개를 만들고  
첫 번째 공간에 1, 두 번째 공간에 2를 넣은 것과 같다.

---

\* 포인터에 문자열 저장하기

Char포인터에 문자열을 큰따옴표에 넣어 할당 할 수 있다.

ex) `char* s1 = "Hello";` → 포인터 s1에 "Hello"문자열이 저장된 주소를 저장

포인터에 직접 문자열을 저장하는 것이 아닌 문자열이 저장된 주소를 저장한다.

문자열 포인터에서 인덱스로 접근 가능하다.

```
Char* s1 = "Hello";  
Printf("%c", s1[0]);
```

주의할점: 문자열 포인터는 인덱스로 문자 할당을 하지 못한다.

~~ex) `char* s1 = "Hello";`  
`s1[0] = A;`~~

---

\* 배열 형태로 문자열 저장하기

ex) `char s1[10] = "Hello";`

크기가 10인 배열에 문자열 "Hello"넣었다.  
이때 메모리 공간 하나에 문자 한 개씩 저장된다.  
저장하고 남은 공간에는 NULL문자로 채워진다.

\*주의점: 문자열 끝에는 항상 NULL문자가 붙는다.  
따라서 배열에 문자열을 저장 할 때 배열 크기가  
문자열 크기보다 최소 +1 더 커야한다.