

# *Do it!* C언어 입문

20164389 멀티미디어공학과 박병주

김상엽 지음 [이지스 퍼블리싱]



# ☞ 목차

15장 : 배열과 포인터

16장 : 메모리 할당



# 배열과 포인터



# 배열과 포인터

배열 표기법과 포인터 표기법

```
char data[5];  
data[1]=5;  
*(data + 1)=5;
```

```
char data;  
char *p = &data  
*p = 3;  
p[0] = 3;
```

```
data[0] -> *(data + 0)  
data[1] -> *(data + 1)  
data[2] -> *(data + 2)  
data[3] -> *(data + 3)  
data[4] -> *(data + 4)
```



# 배열과 포인터

---

배열 시작 주소

```
char data[4];  
char *p = &data[0]; //배열의 첫 번째 항목의 주소가  
                    배열 전체의 시작 주소와 같음
```

```
char *p=&data[0];  
char *p=&*(data+0);  
char *p=&*data;  
char *p=data;
```



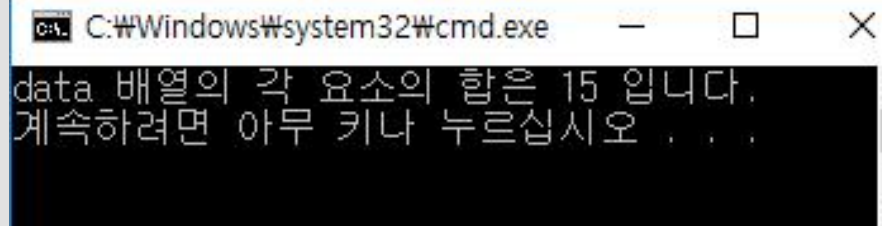
# 배열과 포인터

배열 시작 주소

```
#include <stdio.h>

void main()
{
    char data[5] = { 1,2,3,4,5 };
    int result = 0, i;
    char *p = data;

    for (i = 0; i < 5; i++) {
        result = result + *p;
        p++;
    }
    printf("data 배열의 각 요소의 합은 %d 입니다.", result);
}
```



C:\Windows\system32\cmd.exe

data 배열의 각 요소의 합은 15 입니다.  
계속하려면 아무 키나 누르십시오 . . .



# 배열과 포인터

배열과 포인터 합체

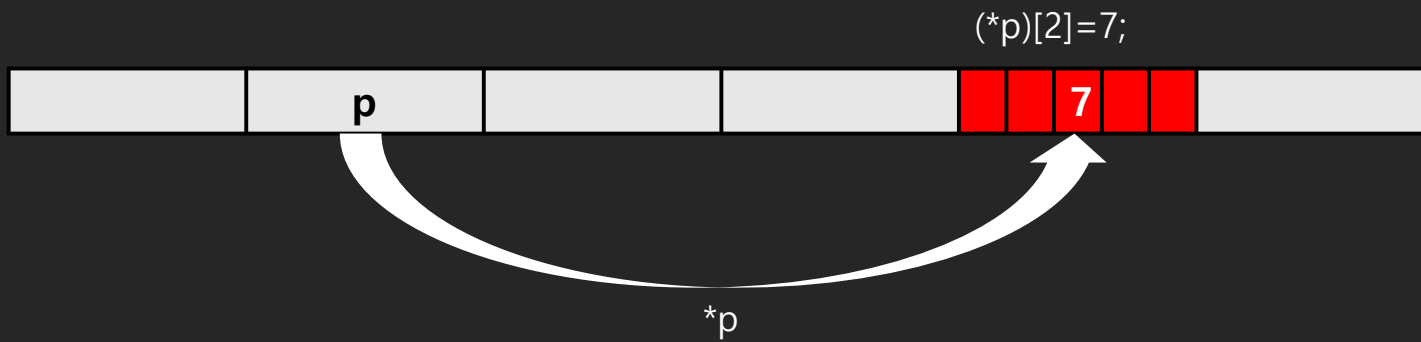
`char *p[5];`

`p[0], p[1], p[2], p[3], p[4]`

`*p[0], *p[1], *p[2], *p[3], *p[4]`

`char (*p)[5];`

char형 5개인 5바이트  
크기 메모리를 가르킴



# 메모리 할당





# 배열과 포인터

메모리 할당이란?

메모리 할당이란?

데이터를 저장할 메모리 공간을  
적절하게 나누는 작업



# 메모리 할당

---

정적 메모리 할당

컴파일러가 코드를 기계어로 번역하는 시점에 변수를 저장할  
메모리 위치를 배정하는 것을 정적 메모리 할당 이라고함



# 메모리 할당

---

데이터 구조

1. 코드 영역 : 실행 할 프로그램의 코드가 저장되는 메모리 공간
2. 데이터 영역 : 전역변수와 static으로 선언되는 변수가 할당
3. 스택 영역 : 지역변수와 매개변수가 할당된다.
4. 힙 영역 : 운영체제가 운영하는 메모리 공간



# 메모리 할당

스택 구조

많은 양의 데이터를 효과적으로 관리

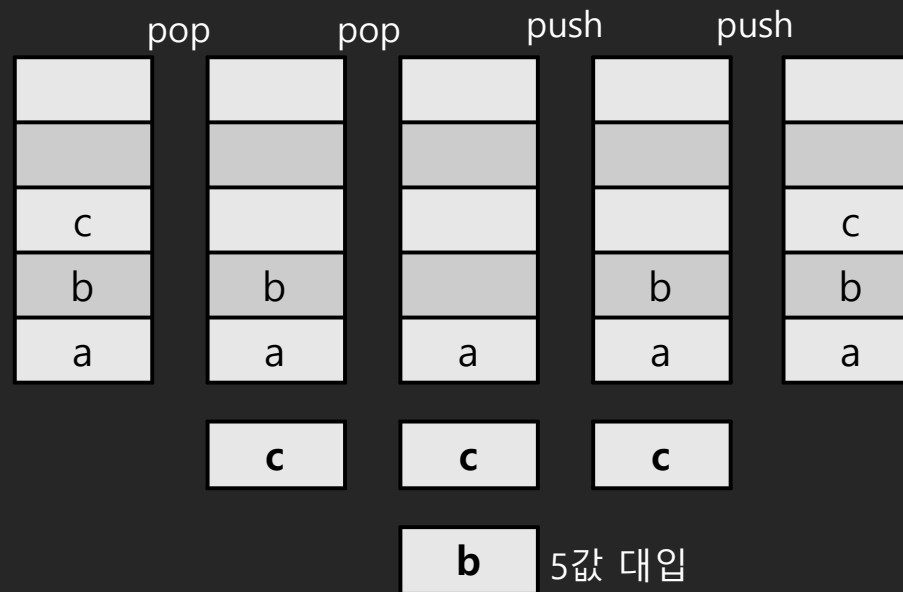
※데이터를 추가하는 작업을 PUSH라고 함  
※데이터를 제거하는 작업을 POP라고 함



# 메모리 할당

스택 구조

b의 값에 5를 대입하기



# 메모리 할당

---

동적 메모리 할당 및 해제

malloc 함수로 동적 메모리 할당하기

함수 원형 : `void *malloc(size_t size);`  
함수 사용 형식 : `void *p = malloc(100)`

`short *p=(short *)malloc(100);`  
[100바이트 – 2바이트 50개]  
`int *p=(int *)malloc(100);`  
[100바이트 – 4바이트 25개]



# 메모리 할당

---

동적 메모리 할당 및 해제

free 함수로 할당된 메모리 해제하기

`free(p);` //p가 가지고 있는 주소에 할당된 메모리를 해제함



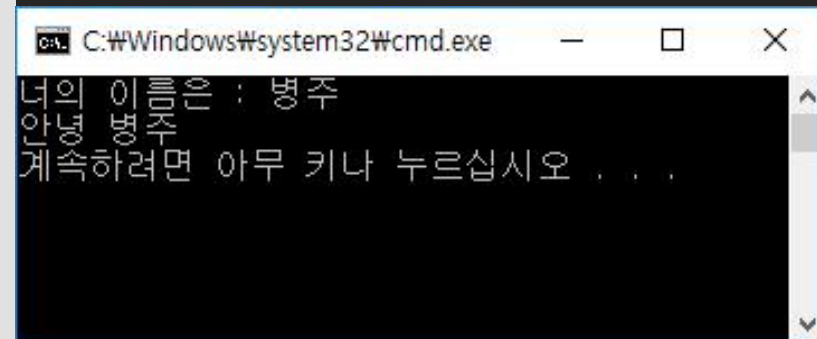
# 메모리 할당

동적 메모리 할당 및 해제

```
#include <stdio.h>
#include <malloc.h>

void main()
{
    char *p_name;
    p_name = (char*)malloc(32);
    if (p_name != NULL) {
        printf("너의 이름은 : ");
        gets(p_name);

        printf("안녕 %s\n", p_name);
        free(p_name);
    }
    else {
        printf("메모리 할당 에러!!");
    }
}
```



```
C:\Windows\system32\cmd.exe
너의 이름은 : 병주
안녕 병주
계속하려면 아무 키나 누르십시오 . . .
```





# 메모리 할당

---

동적 메모리 할당 및 해제

동적 메모리를 할당하는 또 다른 방법

```
int *p = (int *)malloc(sizeof(int)*3); //sizeof(int)*3 ==12
```

```
short *p = (short *)malloc(sizeof(short)*6); //sizeof(int)*6==12
```



# 메모리 할당

동적 메모리 할당 및 해제

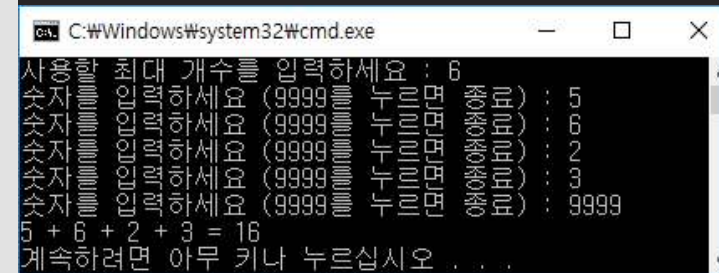
동적 메모리 할당을 사용하여 숫자를 입력 받아 합산하기

```
#include <stdio.h>
#include <malloc.h>

void main()
{
    int *p_num_list, count = 0, sum = 0, limit = 0, i;

    printf("사용할 최대 개수를 입력하세요 : ");
    scanf("%d", &limit);

    p_num_list = (int *)malloc(sizeof(int)*limit);
    while (count < limit) {
        printf("숫자를 입력하세요 (9999를 누르면 종료) : ");
        scanf("%d", p_num_list + count);
        if (*(p_num_list + count) == 9999) break;
        count++;
    }
    for (i = 0; i < count; i++) {
        if (i > 0) printf(" + ");
        printf("%d", *(p_num_list + i));
        sum = sum + *(p_num_list + i);
    }
    printf(" = %d\n", sum);
    free(p_num_list);
}
```



```
C:\Windows\system32\cmd.exe
사용할 최대 개수를 입력하세요 : 6
숫자를 입력하세요 (9999를 누르면 종료) : 5
숫자를 입력하세요 (9999를 누르면 종료) : 6
숫자를 입력하세요 (9999를 누르면 종료) : 2
숫자를 입력하세요 (9999를 누르면 종료) : 3
숫자를 입력하세요 (9999를 누르면 종료) : 9999
5 + 6 + 2 + 3 = 16
계속하려면 아무 키나 누르십시오 . . .
```





**감사합니다.**  
The End.