

url : https://www.youtube.com/watch?v=T_JhyBV3gTI&list=PLz—ENLG_8TMdMJlwyqDIpcEOysvNoonf&index=15

● 구조체 배열

1. 간단한 실습

구조체 배열과 동적할당에 대해 간단한 실습을 하였다.

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

#define MAX 100
int cur_idx ;

typedef struct human
{
    int num;
    char name[20];
}human;

void add_man(human * IDs)
{
    for(;cur_idx < MAX;)
    {
        IDs[cur_idx].num = (cur_idx+1)*2;
        sprintf(IDs[cur_idx++].name, "Man%d", cur_idx+1);
        // 버퍼의 주소에 포맷형식대로 복사시킴
    }
}

void print(human * IDs)
{
    int i = 0;

    for(;i<MAX;i++)
    {
        printf("\nid : %d, name : %s\n", IDs[i].num, IDs[i].name);
    }
}

void main()
{
    human *IDs;

    IDs = (human*)malloc(sizeof(human)*MAX);

    add_man(IDs);
    print(IDs);

    free(IDs);
}
```

2. sprintf 함수

이 함수의 원형은 매뉴얼에 선언되어있다.

```
SYNOPSIS
#include <stdio.h>

int printf(const char *format, ...);
int fprintf(FILE *stream, const char *format, ...);
int dprintf(int fd, const char *format, ...);
int sprintf(char *str, const char *format, ...);
```

이 함수는 문자열을 버퍼로 하여, printf와 유사한 방식으로 버퍼에 해당 포맷을 문자열로 치환하여 복사시킨다.

● 배열의 장단점

장점	단점
<div><ul style="list-style-type: none">* 메모리번지로 접근하기 때문에 속도가 빠름* 사용하기 간편</div>	<div><ul style="list-style-type: none">* 런타임 시, 이미 선언한 배열을 늘리기가 어렵다* 배열 사이에 원하는 값을 끼워넣으려면 리소스가 너무 소모된다.* 런타임 시, 주소연산을 잘못하여, 잘못된 메모리 침범시 OS의 철퇴를 맞는다</div> <div><pre>*** stack smashing detected ***: terminated Aborted (core dumped)</pre><div>(리눅스에서 철퇴를 맞고서)</div></div>

이러한 이유로, 다양한 자료구조 특히 연결리스트를 사용하기도 한다.