

C언어 구조체

양유진

Contents

01 구조체 정의 및 기본구조

02 구조체 포인터

03 구조체 배열

04 자기 참조 구조체 연결리스트



구조체란?

여러 개의 자료형을 묶어서 새로운 자료형을 만들 수 있는 방법

배열

VS

구조체

자료형이 **같은** 것들을
하나로 묶는 것

자료형이 서로 **다른** 것들을
하나로 묶는 것

구조체 기본구조_구조체 선언

```
typedef struct node
```

```
{  
    int data;  
    struct node *next;
```

```
} NODE;
```

구조체의 이름

구조체 멤버

구조체의 별칭

struct node *node1; → NODE *node1;

구조체 기본구조_구조체 변수 선언 및 초기화

```
typedef struct
{
    char name[20] ;
    int number;
    double grade;
}STUDENT;
```

```
int main()
{
```

```
    STUDENT student1;
```

```
    strcpy(student1.name, "양유진");
```

```
    student1.number = 17;
    student1.grade = 4.43;
```

```
}
```

구조체 변수 생성

구조체 변수(student1)
초기화

구조체 멤버 변수는 “.” 연산자를
이용하여 접근할 수 있습니다.

문자열 복사 함수


strcpy(a, “문자열”)

※ 꼭 include <string.h> 해줘야 합니다.

구조체 포인터

```
typedef struct
{
    char name[20] ;
    int number;
    double grade;
}STUDENT;
```

구조체 포인터의
멤버 변수 접근방법



- student->name
- (*student).name

구조체 포인터_예시


```
typedef struct
{
    char name[20] ;
    int number;
    double grade;
}STUDENT;
```

```
int main()
{
    STUDENT* student=malloc(sizeof(STUDENT));
    strcpy(student->name, "양유진");

    student->number = 17;
    student->grade = 4.43;

    printf("이름: %s\n학번: %d\n학점: %.2f\n", student->name, (*student).number, student->grade);

    free(student);
}
```




```
이름: 양유진
학번: 17
학점: 4.43
```

구조체 배열

```
#define TOTAL 2      STUDENT student[TOTAL];  
  
typedef struct  
{  
    char name[20] ;  
    int number;  
    double grade;  
} STUDENT;
```

구조체 배열의
멤버 변수 접근방법



- student[i].name
- (student+i)->name
- *(student+i).name

구조체 배열_예제

```
#define TOTAL 2

typedef struct
{
    char name[20] ;
    int number;
    double grade;
}STUDENT;

int main()
{
    STUDENT student[TOTAL];

    for(int i=0; i<TOTAL; i++)
    {
        printf("이름을 입력하세요: ");
        scanf("%s", &student[i].name);

        printf("학번을 입력하세요: ");
        scanf("%d", &student[i].number);

        printf("학점을 입력하세요: ");
        scanf("%lf", &student[i].grade);

        printf("\n");
    }

    for(int i=0; i<TOTAL; i++)
    {
        printf("이름: %s\n학번: %d\n학점: %.2f\n\n", (student+i)->name, (*(student+i)).number, student[i].grade);
    }
}
```

```
이름을 입력하세요: 양유진
학번을 입력하세요: 17
학점을 입력하세요: 4.43
```

```
이름을 입력하세요: 김민지
학번을 입력하세요: 19
학점을 입력하세요: 3.43
```

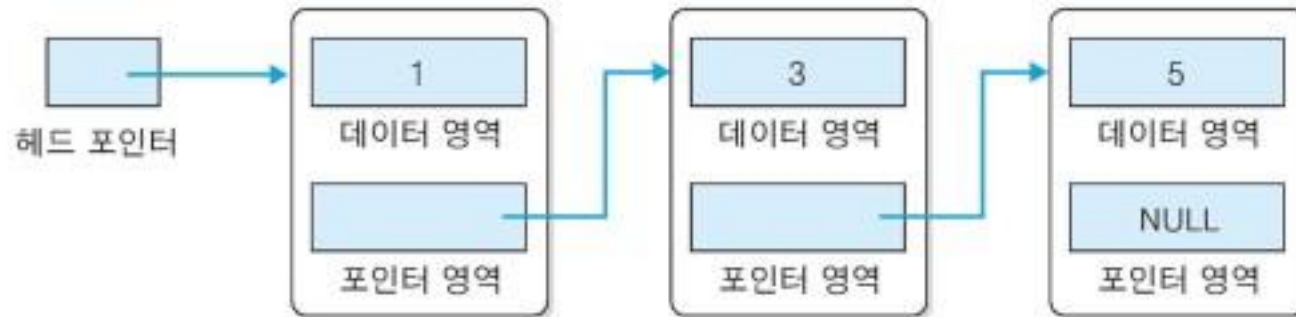
```
이름: 양유진
학번: 17
학점: 4.43
```

```
이름: 김민지
학번: 19
학점: 3.43
```

자기 참조 구조체 연결리스트

연결리스트란?


각각의 데이터를 포인터로 연결해서 관리하는 자료구조.



구조체_ 연결리스트 예제(구조체 정의)

```
typedef struct Node{  
    char name[SIZE];  
    int age;  
    struct Node *link;  
} NODE;
```

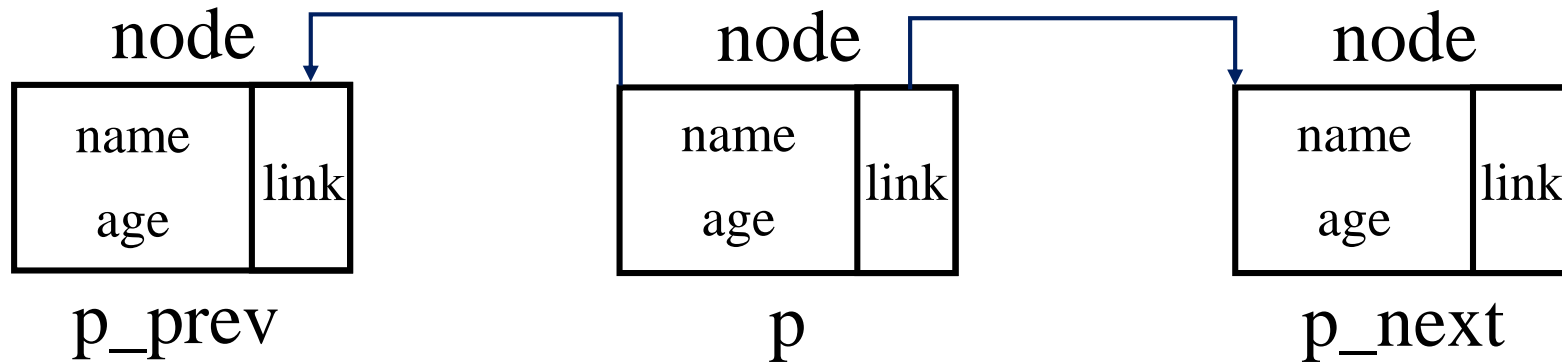
자기 참조 구조체
(연결리스트 주소 저장)



구조체_연결리스트 예제(구조체 변수 정의 및 초기화)

가장 첫번째 노드를 가리키는 헤드 포인터

```
NODE *list = NULL;  
NODE *p_prev, *p, *p_next = NULL;  
char buffer[SIZE];
```



구조체_ 연결리스트 예제(연결리스트 생성)

```
#define SIZE 20
```

```
typedef struct Node{  
    char name[SIZE];  
    int age;  
    struct Node *link;  
} NODE;
```

```
while (1)  
{  
    printf("\n 성명 입력([enter]치면 종료) : ");  
    gets(buffer);  
    if (buffer[0] == '\0')  
        break;
```

gets() vs scanf()

scanf(): 공백 포함 불가능

gets(): 공백 포함 가능

```
p = (NODE *)malloc(sizeof(NODE));  
strcpy(p->name, buffer);
```

— 동적 메모리 할당

```
printf(" 나이 입력 : ");  
gets(buffer);  
p->age = atoi(buffer);
```

— atoi(문자열): 문자열을 정수로 변환해주는 함수

p가 첫 번째 노드인 경우

```
if (list == NULL)  
    list = p;
```

```
else  
{  
    p_prev->link = p;  
    p->link = NULL;  
}  
p_prev = p;  
}
```

— p가 첫 번째 노드가 아닌 경우
이전노드(p_prev)와 새로운 노드(p)를 연결

구조체_연결리스트 예제(연결리스트 출력)

```
typedef struct Node{  
    char name[SIZE];  
    int age;  
    struct Node *link;  
} NODE;
```

```
printf("\n");  
p = list;  
while (p != NULL)  
{  
    printf(" [%s, %d]", p->name, p->age);  
    p = p->link;  
    if (p != NULL)  
        printf(" -> ");  
}
```

다음 노드로 이동

다음 노드가 존재할
때

성명 입력([enter]치면 종료) : aaa
나이 입력 : 1

성명 입력([enter]치면 종료) : bbb
나이 입력 : 2

성명 입력([enter]치면 종료) : ccc
나이 입력 : 3

성명 입력([enter]치면 종료) : ddd
나이 입력 : 4

성명 입력([enter]치면 종료) : eee
나이 입력 : 5

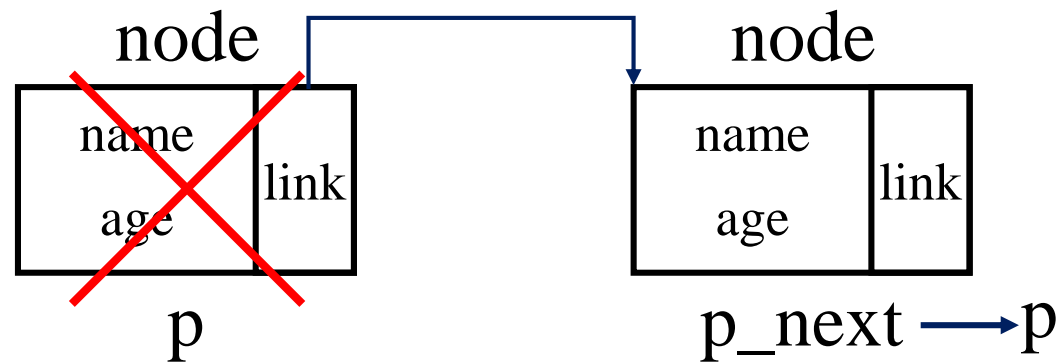
성명 입력([enter]치면 종료) :

[aaa, 1] -> [bbb, 2] -> [ccc, 3] -> [ddd, 4] -> [eee, 5]

구조체_연결리스트 예제(동적 메모리 반납)

```
typedef struct Node{  
    char name[SIZE];  
    int age;  
    struct Node *link;  
} NODE;
```

```
p = list;  
while (p != NULL)  
{  
    p_next = p->link;  
    free(p);  
    p = p_next;  
}
```



감사합니다

