

| 구조체와 포인터

|

구조체

구조체란 하나 이상의 변수를 묶어 그룹화하는 사용자 정의 자료형입니다. 사용자가 C언어의 기본 타입을 가지고 새롭게 정의할 수 있는 사용자 정의 타입이며 기본 타입만으로는 나타낼 수 없는 복잡한 데이터를 표현할 수 있습니다. 구조체는 똑같은 변수를 여러번 사용해야 하는 경우에 효과적으로 사용할 수 있습니다.

구조체

학생들의 성적을 저장해야함

학생 1의 국어 수학 영어 점수는 80 100 20

학생 2의 국어 수학 영어 점수는 10 10 10

학생 3의 국어 수학 영어 점수는 90 90 90

.

.

.

한명당 저장해야 할 정보가 많음

구조체

학생들의 성적을 저장해야함

학생 1의 국어 수학 영어 점수는 80 100 20

학생 2의 국어 수학 영어 점수는 10 10 10

학생 3의 국어 수학 영어 점수는 90 90 90

.

.

.

한명당 저장해야 할 정보가 많음

구조체

학생들의 성적을 저장해야함

학생 1의 국어 수학 영어 점수는 80 100 20

학생 2의 국어 수학 영어 점수는 10 10 10

학생 3의 국어 수학 영어 점수는 90 90 90

.

.

.

한명당 저장해야 할 정보

```
struct Score{  
    char name[10]; //이름  
    int kor; //국어 성적  
    int mat; //수학 성적  
    int eng; //영어 성적  
};
```

구조체

```
struct 구조체이름{  
    //data member  
    자료형 변수명;  
  
    //function member;  
    자료형 함수명(파라미터) {함수내용};  
}
```

구조체

```
struct 구조체이름{  
    //data member  
    자료형 변수명;  
  
    //function member;  
    자료형 함수명(파라미터) {함수내용};  
}
```

```
struct Score {  
    string name; // 이름  
    int kor;      // 국어성적  
    int mat;      // 수학성적  
    int eng;      // 영어성적  
  
    void print_average() { cout << (kor + mat + eng) / 3.0; }  
};
```

구조체 초기화

```
struct Score {  
    string name; // 이름  
    int kor;      // 국어성적  
    int mat;      // 수학성적  
    int eng;      // 영어성적  
};
```

```
int main() {  
    //중괄호를 이용하여 바로 초기화  
    Score s1 = {"김석희", 60, 100, 50};  
  
    // .을 활용하여 초기화  
    Score s2;  
    s2.name = "김석희";  
    s2.kor = 60;  
    s2.mat = 100;  
    s2.eng = 50;  
  
    return 0;  
}
```

구조체이름 변수명;

구조체 함수

```
struct Score {  
    string name; // 이름  
    int kor;      // 국어성적  
    int mat;      // 수학성적  
    int eng;      // 영어성적  
  
    void print_average() { cout << (kor + mat + eng) / 3.0; }  
};
```

구조체 안에 함수를 만들수도 있고,

```
s2.print_average();
```

객체명과 .을 붙여 그 객체로 함수를 실행할 수 있음

구조체 배열

```
#include <iostream>
#include <string>
using namespace std;

struct Score {
    string name; // 이름
    int kor;     // 국어성적
    int mat;     // 수학성적
    int eng;     // 영어성적
};

int main() {

    Score s[10] = {};

    for (int i = 0; i < 10; i++) {
        cin >> s[i].name;
        cin >> s[i].kor;
        cin >> s[i].mat;
        cin >> s[i].eng;
    }

    return 0;
}
```

똑같은 데이터 타입이라
배열로 만들 수 있음

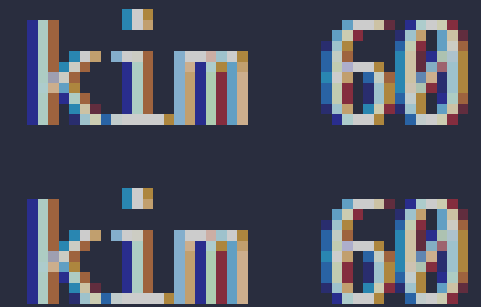
구조체 포인터

```
struct Score {  
    string *name; // 이름  
    int *kor;     // 국어성적  
    int *mat;     // 수학성적  
    int *eng;     // 영어성적  
};
```

변수들을 포인터로 만들수도 있고

구조체 포인터

```
int main() {  
    // 중괄호를 이용하여 바로 초기화  
    Score s1 = {"kim", 60, 100, 50};  
  
    Score *p = nullptr;  
    p = &s1;  
  
    cout << s1.name << " " << s1.kor << "\n";  
    cout << p->name << " " << p->kor << "\n";  
  
    return 0;  
}
```



kim 60
kim 60

구조체를 포인터로 만들어서
같은 구조체를 가리킬 수 있음.

구조체 포인터

```
int main() {  
    // 중괄호를 이용하여 바로 초기화  
    Score s1 = {"kim", 60, 100, 50};  
  
    Score *p = nullptr;  
    p = &s1;  
  
    cout << s1.name << " " << s1.kor << "\n";  
    cout << p->name << " " << p->kor << "\n";  
  
    return 0;  
}
```

S1.name
↑
p

구조체를 포인터로 만들어서
같은 구조체를 가리킬 수 있음.

주의할점은 포인터로 접근을 할 경우
-> 로 접근을 해야함

구조체 포인터

```
struct Score {  
    int total;  
    Score *next_man;  
};  
  
int main() {  
    // 중괄호를 이용하여 바로 초기화  
    Score s1 = {210, nullptr};  
  
    Score s2 = {100, nullptr};  
  
    s1.next_man = &s2;  
  
    cout << s1.total << " " << s1.next_man->total << "\n";  
  
    return 0;  
}
```

구조체 안에 자기와 동일한 데이터 타입을 가리키는 포인터를 넣을 수도 있음 (리스트를 만들기 위하여)

구조체 최종

```
#include <stdio.h>

struct Score {
    int kor; //국어성적
    int mat; //수학성적
    int eng; //영어성적
    Score *link1; //자기참조 구조체 선언
    sub_Score *link2; //외부참조 구조체 선언
};

struct sub_Score {
    int soc; //사회성적
    int sci; //과학성적
};
```

구조체 최종

```
int main()
{
    Score s1 = { 100,100,100,NULL,NULL};
    Score s2 = { 50,50,50,NULL,NULL };
    Score s3 = { 0,0,0,NULL,NULL };

    s1.link1 = &s2; //s1 -> s2 자기 참조
    s2.link1 = &s3; //s2 -> s3 자기 참조

    printf("국어 : %d 수학 : %d 영어 : %d \n", s1.kor, s1.mat, s1.eng);
    printf("국어 : %d 수학 : %d 영어 : %d \n", s1.link1->kor, s1.link1->mat, s1.link1->eng);
    printf("국어 : %d 수학 : %d 영어 : %d \n", s1.link1->link1->kor, s1.link1->link1->mat, s1.link1->link1->eng);

    sub_Score sub1 = { 100,100 };
    sub_Score sub2 = { 50,50 };
    sub_Score sub3 = { 0,0 };

    s1.link2 = &sub1; //s1 -> sub2 외부 참조
    s2.link2 = &sub2; //s2 -> sub3 외부 참조
    s3.link2 = &sub3; //s2 -> sub3 외부 참조

    printf("사회 : %d 과학 : %d \n", s1.link2->soc, s1.link2->sci);
    printf("사회 : %d 과학 : %d \n", s2.link2->soc, s2.link2->sci);
    printf("사회 : %d 과학 : %d \n", s3.link2->soc, s3.link2->sci);

    return 0;
}
```


문제

기초

28431 - 양말 짝 맞추기

10818 최소, 최대

23808 - 골뱅이찍기

2869 - 달팽이는 올라가고 싶다.

1476 - 날짜계산

11047 - 동전 0(숙제)

별찍기 시리즈

고수

10808 - 알파벳개수

2292 - 벌집

3052 - 나머지

18311-왕복(생각하는 법 기르기)

9613 - GCD 합(최대공약수들의 합 구하기)

유클리드 호제법 모르면 검색

11729 - 하노이의 탑 이동 순서(숙제)

1074 - Z

5525 - IOIOI

색종이의 합