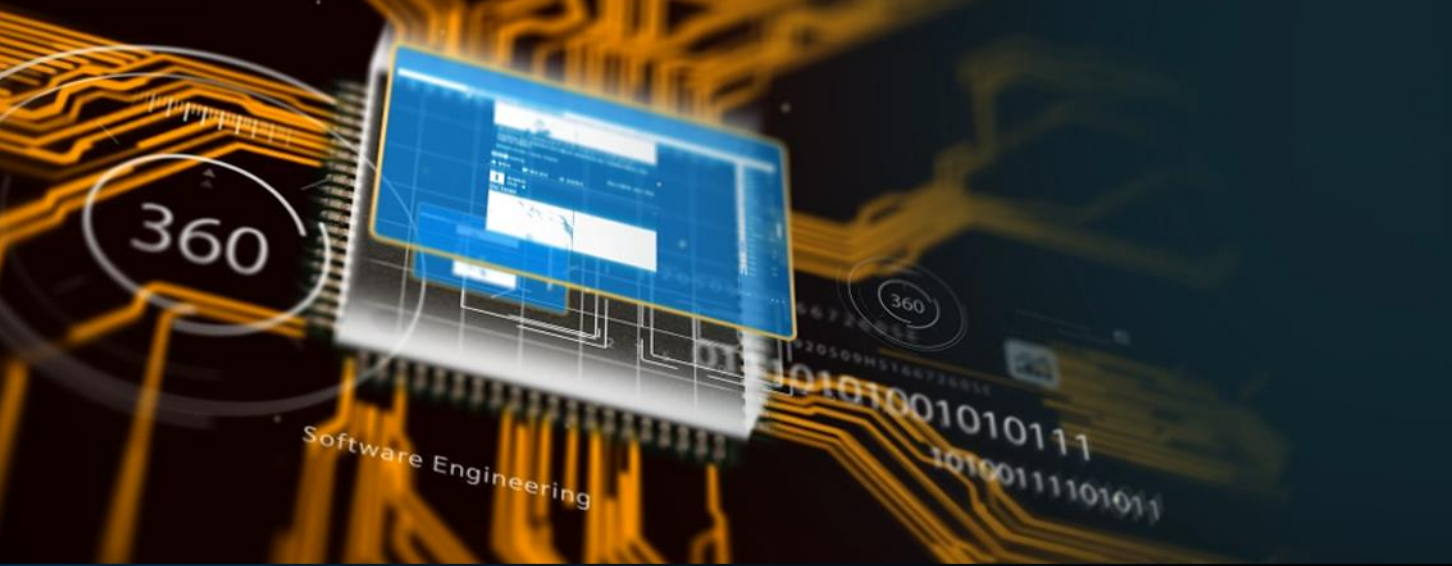


프로그래밍 언어 활용

1011101010001010101

part 1



구조체



한국기술교육대학교
온라인평생교육원

학습내용

- 구조체 이해
- 구조체 처리

학습목표

- 구조체가 무엇인지 알고 용도를 설명할 수 있다.
- 구조체를 용도에 맞게 정의하고 선언할 수 있다.

구조체 이해

1 구조체란?

01

사용자 정의 데이터 타입

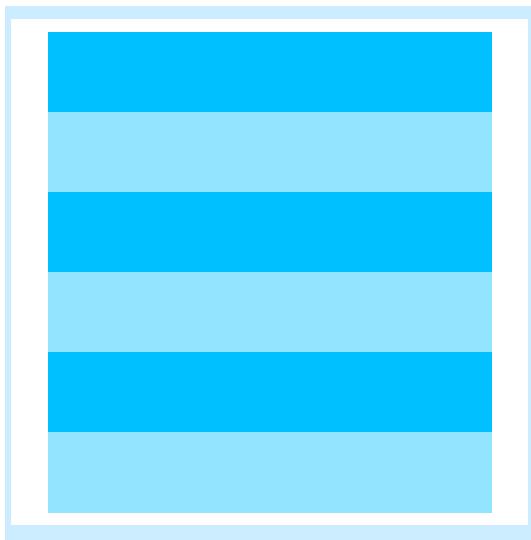
02

관련 있는 데이터를 묶어서 처리할 수 있는 자료구조

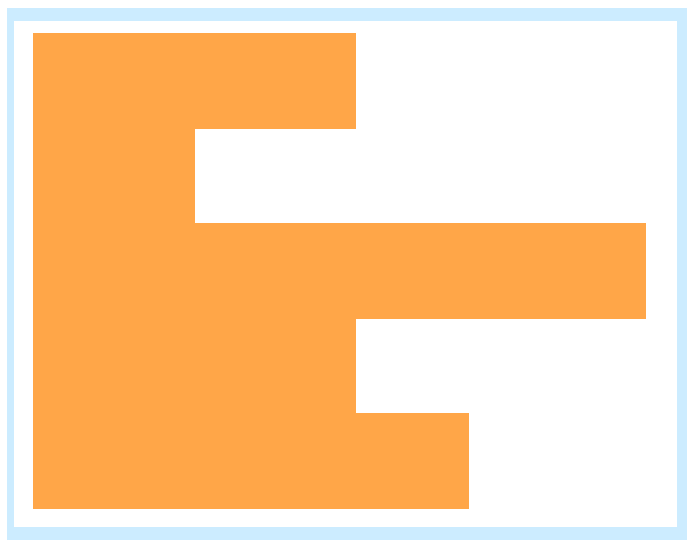
03

서로 다른 데이터 타입의 묶음

배열(Array)



구조체(Structure)



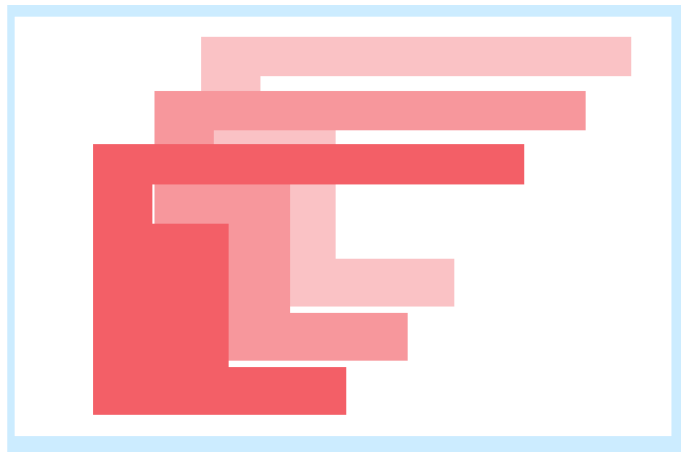
구조체 이해

1 구조체란?

04

서로 다른 데이터 타입을 묶어서 처리할 수 있도록 사용자가 정의하는 데이터 타입

이름	
번호	
국어	
영어	
수학	
평균	



2 구조체 정의방법

1 구조체의 멤버로 일반 변수뿐만 아니라 **배열이나 포인터 선언** 가능

2 구조체를 정의하면 새로운 데이터형이 만들어짐

3 구조체 정의 → 메모리 할당의 의미는 아님

4 구조체형 변수 선언 → 메모리에 할당

구조체 이해

2 구조체 정의방법

형식

```
struct 구조체명 {
    데이터형 멤버명 ;
    데이터형 멤버명 ;
    ...
};
```

예제

```
struct student {
    char name[20];
    int num, kor, eng, mat;
    double average;
};
```



구조체의멤버

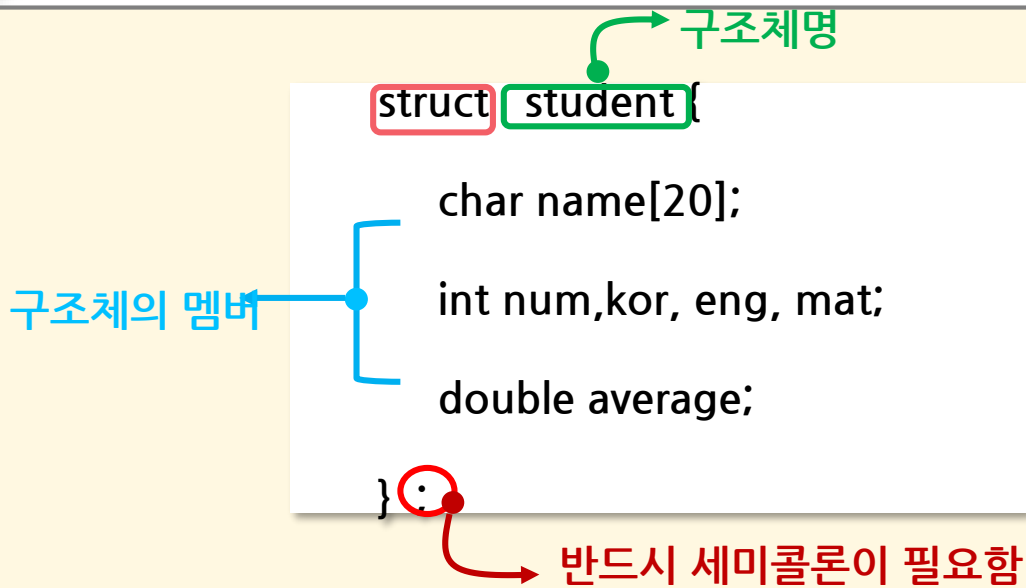
```
struct student {
    char name[20];
    int num, kor, eng, mat;
    double average;
};
```



구조체를 정의함

구조체 이해

2 구조체 정의방법



5 구조체의 크기는 모든 멤버들의 크기의 합보다 크거나 같음

6 구조체 멤버 중 가장 큰 멤버의 크기를 기준으로 멤버 할당(Padding)

7 구조체의 크기를 구하려면 **sizeof 연산자**를 이용함

```
struct sungjuk {
    char a;
    int kor ;
};
```



구조체 이해

3 구조체 선언

형식

struct 구조체명 변수명1, 변수명2, ...;

예제

```
struct student
struct student
```

구조체명

```
s1;
s2, s3;
```

구조체 변수명

구조체 변수를 선언하면 구조체 변수가 가진 멤버들이
메모리에 선언된 순서대로 할당됨

```
struct student
s1;
```

char[20]

int

int

int

double

name

korean

english

math

average

s1

구조체 이해

3 구조체 선언

구조체를 정의하면서 구조체 변수를 함께 선언할 수 있음

```
struct point {
    int x;
    int y;
} p1, p2; ●———— point 구조체를 정의하면서 변수를 함께 선언함
```

구조체를 정의하면서 변수를 함께 선언할 때는 태그명을 생략할 수 있음

```
struct { ●———— 구조체명을 생략함
    char sex;
    int age;
} m1, m2; ●———— 이름 없는 구조체의 변수를 선언함
```


구조체 처리

1 초기화

```
int main()
{
    struct sungjuk m2_1 ;
    m2_1.name = "hong kildong";
    m2_1.kor = 95;
    m2_1.eng = 87 ;
    m2_1.avg = (double)(m2_1.kor + m2_1.eng)/2.0
    printf("이름 : %s 국어 : %d 영어 : %d 평균 : %f " , m2_1.name,
    m2_1.kor, m2_1.eng, m2_1.avg);

    return 0;
}
```

05

같은 구조체형의 변수들끼리는 서로 초기화나 대입이 가능함

06

구조체 간의 초기화 : 멤버 대 멤버 초기화



동일한 멤버 간에 1:1로 복사해서 초기화

07

구조체 간의 대입 : 멤버 대 멤버 대입

구조체 처리

1 초기화

```
struct point {
    int x;
    int y;
};
```

```
struct point p1 = {10, 20};
```

```
struct point p2 = {30, 40};
```

```
struct point p3 = p1;      ————— 구조체 간의 초기화
```

```
struct point p4;
```

```
p4 = p2;      ————— 구조체 간의 대입
```

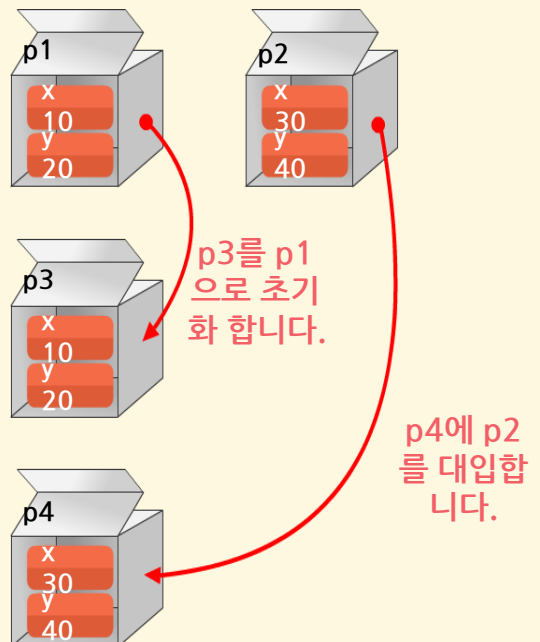
```
struct point p1 = {10, 20};
struct point p2 = {30, 40};
```

초기화

```
struct point p3 = p1;
```

대입

```
struct point p4;
p4 = p2;
```



구조체 처리

2 구조체 응용

1 구조체 멤버

구조체를 다른 구조체의 멤버로 사용 가능

```
struct student {  
    char *name;  
    int num;  
    int age;  
};
```

```
struct sungjuk {  
    struct student stud;  
    int kor, eng;  
    double avg;  
};  
  
struct sungjuk m2 = { {"hong kildong", 20200001, 27}, 95, 87};  
  
printf(" 학번 : %d", m2.stud.num);
```

구조체 처리

2 구조체 응용

2 구조체 배열

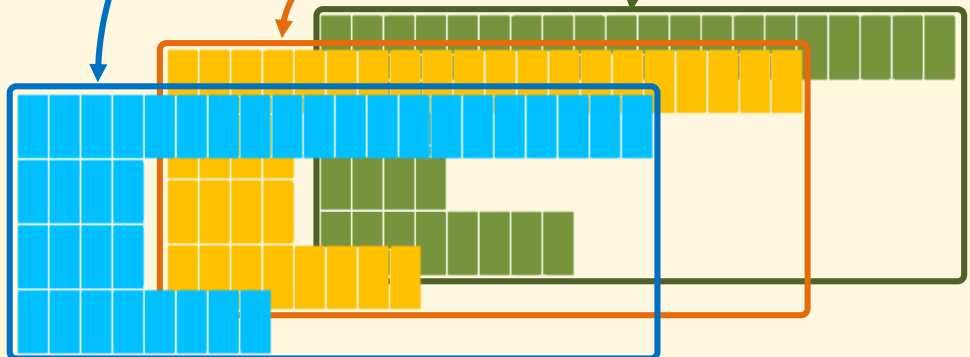
구조체를 배열로 선언하는 것이 가능

```
struct sungjuk {
    char name[20];
    int kor, eng;
    double avg;
};
```

```
struct sungjuk m[50]; // 50명 학생의 성적을 처리
```

구조체 배열을 초기화하려면 {} 안에 배열 원소의 초기값을 나열

```
struct sungjuk m[3] = {{ "hong", 95, 87 }, { "park", 85, 67 }, { "kim", 78, 92 }};
```



구조체 처리

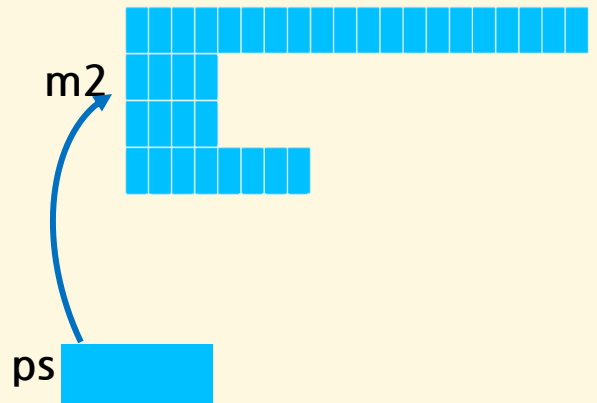
2 구조체 응용

3 구조체 포인터

구조체 포인터를 선언하는 것이 가능

```
struct sungjuk {
    char name[20];
    int kor, eng;
    double avg;
};

struct sungjuk m2={ "hong", 95, 87 };
struct sungjuk *ps = &m2;
```



구조체 포인터로 구조체의 멤버에 접근할 때는 “->” 간접 접근 연산자를 사용

```
struct sungjuk {
    char name[20];
    int kor, eng;
    double avg;
};
```

```
struct sungjuk m2={ "hong", 95,
87 };
struct sungjuk *ps = &m2;

m2.kor = 92;
ps->kor = 92;
```

학습정리

1. 구조체 이해



- 구조체는 서로 다른 데이터를 묶어서 처리하기 용이한 자료구조임
- 구조체는 먼저 정의하고 변수를 선언하여 사용함
- 구조체 선언 시 구조체명 앞에 struct라는 키워드를 사용해야 함
- 구조체 변수는 모든 데이터 타입을 사용할 수 있음

2. 구조체 처리



- 구조체는 선언 시 초기화하는 것이 가능함
- 선언 시 초기화에는 배열과 동일한 방법으로 초기화함
- 구조체도 필요에 따라 구조체 배열을 선언하는 것이 가능함
- 구조체 포인터로 구조체 멤버에 접근할 때는 간접 접근 연산자 ->를 사용함