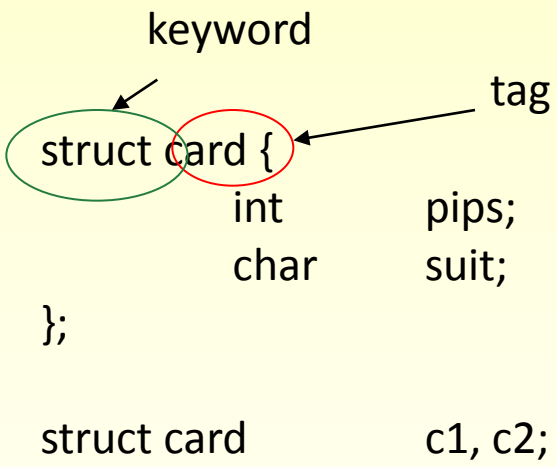
C 프로그래밍 4

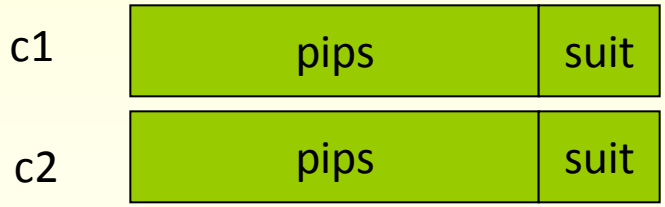
구조체

목포해양대 해양전자통신공학부

구조체와 구조체 선언

- 서로 다른 형의 변수들을 하나로 묶는 방법
- 구조체 선언 예





구조체 멤버 접근

- 멤버 접근 연산자 .
- 접근 형식

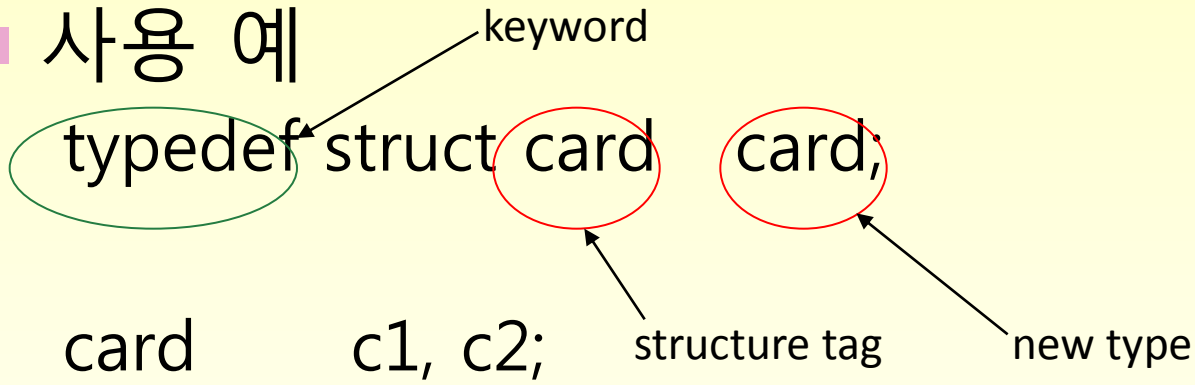
  structure_variable.member_name
- 사용 예

```
c1.pips = 10;
c1.suit = ‘s’;

c2 = c1;
```

typedef를 이용한 새로운 형 선언

- typedef를 이용하여 구조체를 새로운 형으로 선언
- 사용 예



구조체 선언과 형 선언을 동시에

- 형식

```
typedef struct struct_tag {
        int         member1;
        double      member2;
} type_name;

type_name       v1, v2;
```

- 예

```
typedef struct card {
        int         pips;
        char        suit;
} card;

card c1, c2;
```

- *struct_tag*는 생략 가능

일반적으로 구조체 tag를 생략하는 경우

- 그 구조를 갖는 변수 선언이 같이 이루어져서, 더 이상 그 구조를 갖는 변수가 존재하지 않을 때

```
struct {
    int         pips;
    char        suit;
} c1, c2;
```

- typedef과 같이 써서 새로운 형 이름이 부여될 때

```
typedef struct {
    int         pips;
    char        suit;
} card;
```

구조체 멤버 접근

- 멤버접근연산자 .

```
card            c1, c2;

c1.pips = 1;
c1. suit = 'S';
c2 = c1;
```

- 포인터 변수일 경우 멤버접근연산자 -> 이용이 더 편리

```
card            *c3;

c3 = &c1;
c3->pips = 3;           // (*c3).pips = 3과 동일
(*c3).suit = 'H';       // c3 -> suit = 'H'와 동일
```

함수에서의 구조체 사용

- 함수의 매개변수로 구조체를 이용하고, 구조체를 반환하는 방법

```
employee_data update(employ_data e)
{
    employee_data tmp;
    …
    return tmp;
}
```

- 구조체의 주소를 넘겨받아 직접 값을 수정하는 방법

```
void update(employ_data *p)
{
    p->employee_id = 200123;
    …
}
```

- 첫번째 방법의 경우 구조체 전체를 위한 **공간들**이 준비되어야 하고, 값이 **복사**되어 전달
- 일반적으로 두번째 방법 선호

구조체의 초기화

- 배열의 초기화 방법과 유사
- 예

```
card c = {13, 'H'};
complex a[3][3] = {
   {{1.0, -0.1}, {2., 2.}. {3., 3.}},
   {{4., -0.4}, {5., .5}, {6., .6}},
};
struct fruit frt = {"plum", 150};
```