Structures

Summary

structures

- 어디에 쓰지?
- 선언은 어떻게 하지?
- 초기화는 어떻게 하지?
- 어떻게 활용해?
- 구조체에 꼬리표를 달면 재사용이 쉽다고?
- 새로운 형을 내 맘대로 만든다고?
- 구조체를 함수의 인자 또는 리턴 값으로 받기
- 구조체 안에 구조체 담기
- 구조체의 배열

구조체 어디에 쓰지?

- **서로 다른 형**의 데이터를 **하나의 이름**으로 관리 해야 할 때

- 예:
  - 파일의 변경 시간과 날짜, 파일 크기, 접근 권한 등의 정보를 관리해야 할 때
  - 음악에 대한 정보(작곡가, 작사가, 부른 사람/팀, 길이, 가사 등)를 관리해야 할 때
  - 연락처(이름, 생일, 전화번호1, 전화번호 2, 주소, 메모 등)를 관리해야 할 때

구조체의 선언은?

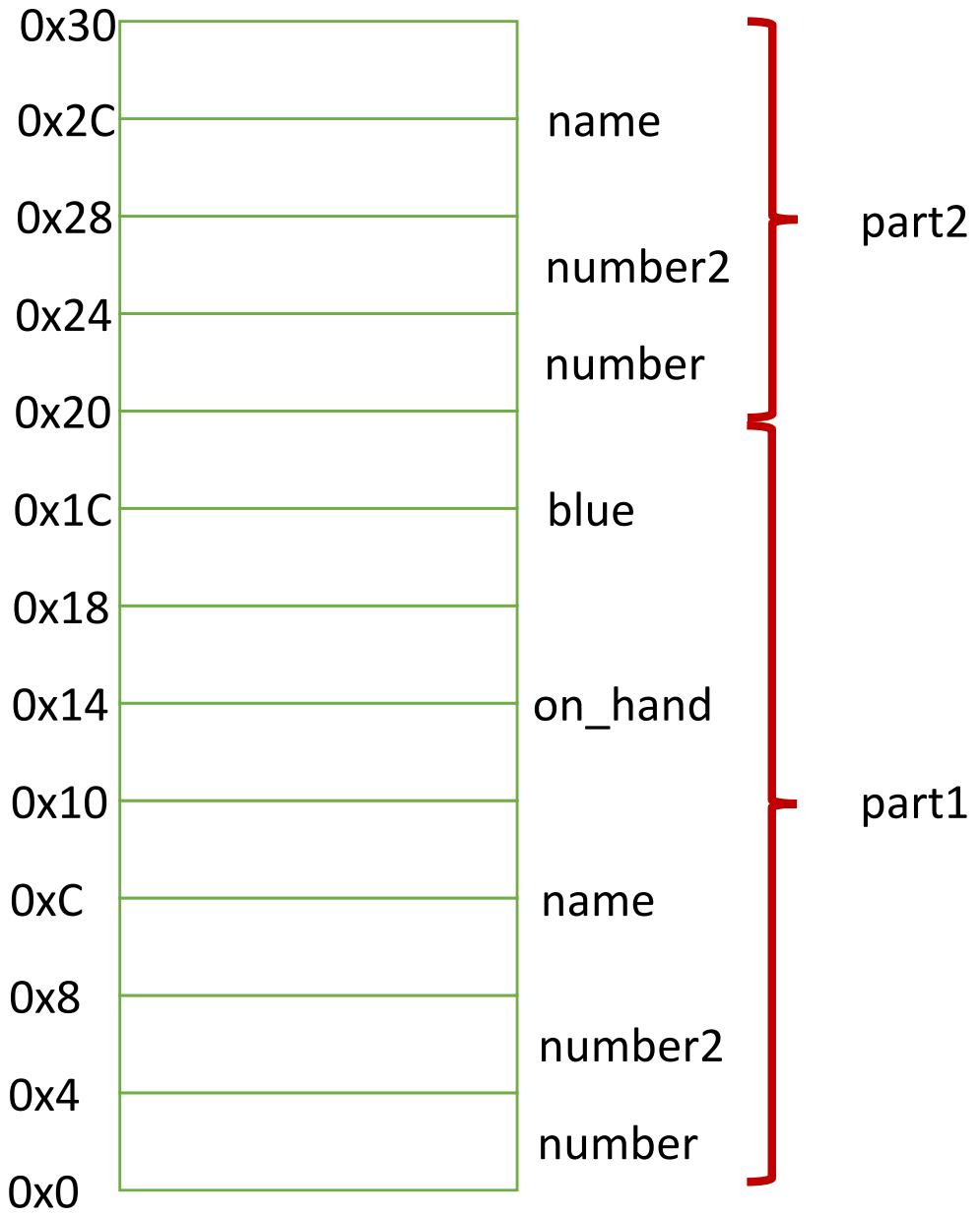
- 기본 형 (struct 키워드, 구조체로 쓸 변수 명, 이 구조체에서 쓸 멤버 변수들)
  - 서로 다른 구조체의 멤버 변수의 이름은 같아도 상관 없음

```
struct {
    int    number;
    float  number2;
    char   name[8];
    double on_hand;
    char   blue[8];
} part1, part2;
```

Member
Record
Field

선언된 구조체의 메모리 표현?

```
struct {
        int     number;
        float   number2;
        char    name[8];
        double on_hand;
        char    blue[8];
} part1, part2;
```



주의

```
struct {
        int     number;
        float   number2;
        char    name[8];
        double  on_hand;
        char    blue[8];
} part1, part2;
```

part1, part2 는 같은 구조체

partA, partB 는 다른 구조체

Different

```
struct {
        int     number;
        float   number2;
        char    name[8];
        double  on_hand;
        char    blue[8];
} partA;
```

```
struct {
        int     number;
        float   number2;
        char    name[8];
        double  on_hand;
        char    blue[8];
} partB;
```

주의: 예제

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5       struct {
6           int    number;
7           char   name[8];
8           double on_hand;
9       } part1 = {128, "Helen", 3.14}, part2 ;
10
11      struct {
12          int    number;
13          char   name[8];
14          double on_hand;
15      } part3 ;
16
17      part2 = part1;
18      part3 = part1;
19
20      int result = part1.number + part2.number + part3.number;
21      printf("%d\n", result);
22      return 0;
23 }
```

```
t.c:19:11: error: assigning to 'struct (anonymous struct at t.c:11:5)' from
      incompatible type 'struct (anonymous struct at t.c:5:5)'
    part3 = part1;
          ^ ~~~~~
1 error generated.
```

구조체 초기화 방법 2 가지

1 선언과 함께 초기화 (예: int foo = 500;)

```
struct {
        int     number;
        float   number2;
        char    name[8];
        double  on_hand;
        char    blue[8];
} part1 = {528, 3.14, "Helen", 6.28, "Keller"}, part2 ;
```

순서 중요!

```
struct {
        int     number;
        float   number2;
        char    name[8];
        double  on_hand;
        char    blue[8];
} part1 = {.name= "Helen", .number =7}, part2 ;
```

C99의 경우

일부만 초기화 가능

구조체 초기화 방법 2 가지

2 선언 후에 초기화 (예: int foo; foo = 500;)

```
part2.number = 123 ;
part2.number2 = 8.01;
part2.name = “Will”;
part2.on_hand = 909.002;
part2.bule = “smith”;
```

구조체이름 ● 멤버변수이름 = 값;

구조체를 어떻게 활용해?

- 선언과 할당이 완료되면 변수처럼 쓰면 됨
  - 단, 이름이 길 뿐

```
#include <stdio.h>

int main(void)
{
    struct {
        int    number;
        char   name[8];
    } part1 = {528, "Helen"}, part2 = {.number = 0} ;

    int result = part1.number + part2.number;
    printf("%d\n", result);
    return 0;
}
```

꼬리표(tag)를 달면 쓰기 쉽다고?

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5       struct one {
6           int    number;
7           char   name[8];
8           double on_hand;
9       } part1 = {128, "Helen", 3.14}, part2 ;
10
11      struct one part3 ;
12
13      part2 = part1;
14      part3 = part1;
15
16      int result = part1.number + part2.number + part3.number;
17      printf("%d\n", result);
18      return 0;
19 }
```

Tag 정보

같은 구조체란 정보를 알려줌

같은 구조체인 경우 할당을 통해 복사 가능

멤버인 배열의 복사도 간단히 해결할 수 있음
단, 구조체를 복사하는 경우만 가능

구조체를 새로운 형처럼 쓴다고?

- typedef 라는 키워드를 쓰면 새로운 형을 만들 수 있음


```
typedef int age;

int foo = 5;
age bar = 5; // 기존의 형처럼 사용

if(foo == bar)
    printf("true\n");
```

구조체를 새로운 형처럼 쓴다고?

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5    typedef int OTZ;
6    typedef struct {       // 또는 “typedef struct TAG이름 {“ 도 가능
7        OTZ  age;
8        char name[10];
9    } alpha;
10
11    alpha foo = {.age = 10, .name = "helen"};
12    alpha bar;
13    bar = foo;
14    printf("%d\n", bar.age);
15    return 0;
16 }
```

함수의 인자와 리턴 값은 어떻게 전달해?

Type I

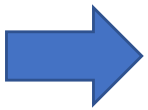
```c
#include <stdio.h>

typedef int OTZ;
struct go {
    OTZ  age;
    char name[10];
} ;

struct go sum(struct go one, struct go two)
{
    struct go res;
    res.age = one.age + two.age;
    return res;
}

int main(void)
{
    struct go foo = {.age = 10};
    struct go bar = {.age = 20};
    struct go result;
    result = sum(foo, bar);
    printf("%d\n", result.age);
    return 0;
}
```

Type II

```c
#include <stdio.h>

typedef int OTZ;
typedef struct {
    OTZ  age;
    char name[10];
} alpha;

alpha sum(alpha one, alpha two)
{
    alpha res;
    res.age = one.age + two.age;
    return res;
}

int main(void)
{
    alpha foo = {.age = 10};
    alpha bar = {.age = 20};
    alpha result;
    result = sum(foo, bar);
    printf("%d\n", result.age);
    return 0;
}
```

구조체 안에 구조체를 넣는다고?

```
typedef int OTZ;
typedef struct {
    OTZ  age;
    char name[10];
} sinfo;
```



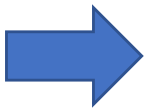
```
typedef int OTZ;

struct Name {
    char first[10];
    char middle[10];
    char last[10];
};

typedef struct student{
    OTZ  id, age;
    struct Name sname;
} sinfo;

sinfo students1;
```

// 문자열의 복사 안전한 방법 (string.h 헤더 필요)

```
strcpy(student1.sname.last, "Gildong");
```

구조체로 배열을 만들 수 있어?

```
typedef int OTZ;

struct Name {
    char first[10];
    char middle[10];
    char last[10];
};

typedef struct student{
    OTZ  id, age;
    struct Name sname;
} sinfo;

sinfo students1, students2, …;
```



```
typedef int OTZ;

struct Name {
    char first[10];
    char middle[10];
    char last[10];
};

typedef struct student{
    OTZ  id, age;
    struct Name sname;
} sinfo;

sinfo students[100];
```

보통의 배열 선언처럼 활용

```
strcpy(students[60].sname.last, ”Giltong”);

students[59].sname.last[3] = “d”;
```

60번째 students 구조체에
last 멤버의 4번째 문자를 d로 변경