

struct

struct

구조체

- 1) 여러 개의 변수를 묶어 하나의 객체를 표현하고자 할 때 구조체를 사용할 수 있음
- 2) 캐릭터, 몬스터, 학생, 좌표 등 다양한 객체를 모두 프로그래밍 언어를 이용해 표현할 수 있음.

struct

구조체의 정의와 선언

```
struct 구조체명  
{ 자료형1 변수명1;  
  자료형2 변수명2;  
  ...  
};
```

struct

구조체의 정의와 선언

- 1) 한 명의 학생에 대한 정보를 담고 있는 구조체.

```
struct Student {  
    char studentId[10];  
    char name[10];  
    int grade;  
    int major;  
};
```

struct

구조체의 변수의 선언과 접근

1) 기본적으로 구조체의 변수에 접근할 때는 점(.)을 사용.

```
struct Student s; // 구조체 변수 선언
strcpy(s.studentId, "20153157"); // 구조체 변수에 접근
strcpy(s.name, "이순신");
s.grade = 4;
strcpy(s.major, "컴퓨터공학과");
```

struct

```
#include <stdio.h>

struct Student { // 학생 구조체 정의
    char studentId[10];
    char name[10];
    int grade;
    char major[51];
};

int main(void) {
    struct Student s; // 구조체 변수 선언
    strcpy(s.studentId, "20250215"); // 구조체 변수에 접근
    strcpy(s.name, "이순신");
    s.grade = 1;
    strcpy(s.major, "컴퓨터공학과");
    printf("학번: %s\n", s.studentId); // 구조체 내용 출력
    printf("이름: %s\n", s.name);
    printf("학년: %d\n", s.grade);
    printf("학과: %s\n", s.major);
    return 0;
}
```

struct

구조체의 정의와 선언

- 1) 하나의 구조체 변수만 사용하는 경우 정의와 동시에 선언을 할 수도 있음.
- 2) 이 경우 변수는 전역 변수로 사용됨.

```
struct Student { // 학생 구조체 정의 및 선언
    char studentId[10];
    char name[10];
    int grade;
    char major[51];
} s;
```

struct

구조체의 초기화

- 1) 구조체의 변수를 한 번에 초기화하기 위해서는 중괄호에 차례대로 변수의 값을 넣음.

```
#include <stdio.h>

struct Student { // 학생 구조체 정의
    char studentId[10];
    char name[10];
    int grade;
    char major[51];
};

int main(void) {
    struct Student s = { "20250215", "이순신", 4, "컴퓨터공학과" };
    printf("학번: %s\n", s.studentId); // 구조체 내용 출력
    printf("이름: %s\n", s.name);
    printf("학년: %d\n", s.grade);
    printf("학과: %s\n", s.major);

    return 0;
}
```


struct

구조체의 초기화

1) 그래서 이런 형태로 초기화하는 것도 가능.

```
#include <stdio.h>

struct Student { // 학생 구조체 정의, 선언 및 초기화
    char studentId[10];
    char name[10];
    int grade;
    char major[51];
} s = { "20250215", "이순신", 4, "컴퓨터공학과" };

int main(void) {
    printf("학번: %s\n", s.studentId); // 구조체 내용 출력
    printf("이름: %s\n", s.name);
    printf("학년: %d\n", s.grade);
    printf("학과: %s\n", s.major);
    return 0;
}
```

struct

더 짧게 구조체 정의하기

1) typedef 키워드를 이용하면 임의의 자료형을 만들 수 있으므로 선언이 더 짧아짐.

```
#include <stdio.h>

typedef struct Student { // 학생 구조체 정의
    char studentId[10];
    char name[10];
    int grade;
    char major[51];
} Student;

int main(void) {
    Student s = {"20250215", "이순신", 4, "컴퓨터공학과"};
    printf("학번: %s\n", s.studentId); // 구조체 내용 출력
    printf("이름: %s\n", s.name);
    printf("학년: %d\n", s.grade);
    printf("학과: %s\n", s.major);

    return 0;
}
```

struct

더 짧게 구조체 정의하기

1) 최근에는 익명 구조체의 개념이 등장하여, 구조체 이름 부분을 비워 놓아도 됨.

```
#include <stdio.h>

typedef struct { // 학생 구조체 정의
    char studentId[10];
    char name[10];
    int grade;
    char major[51];
} Student;

int main(void) {
    Student s = { "20250215", "이순신", 4, "컴퓨터공학과" };
    printf("학번: %s\n", s.studentId); // 구조체 내용 출력
    printf("이름: %s\n", s.name);
    printf("학년: %d\n", s.grade);
    printf("학과: %s\n", s.major);

    return 0;
}
```

struct

구조체 포인터 변수에 접근하기

- 1) 구조체가 포인터 변수로 사용되는 경우 내부 변수에 접근할 때 화살표(->)를 사용함.

```
typedef struct { // 학생 구조체 정의
    char studentId[10]; char
    name[10];
    int grade; char
    major[51];
} Student;

int main(void) {
    Student *s = malloc(sizeof(Student));
    strcpy(s->studentId, "20250215"); // 구조체 포인터 변수에 접근
    strcpy(s->name, "이순신");
    s->grade = 4;
    strcpy(s->major, "컴퓨터공학과");
    printf("학번: %s\n", s->studentId); // 구조체 내용 출력
    printf("이름: %s\n", s->name);
    printf("학년: %d\n", s->grade);
    printf("학과: %s\n", s->major);

    return 0;
}
```