

7 함수

기능을 수행하는 코드 단위

→ main 함수, printf, scanf

표준라이브러리함수

· 함수제작에 중요한 요소

- 함수 정의 : 코드로 기능 구현

- 함수 호출 : 호출해야 함수 사용 가능

- 함수 선언 : 프로그램의 상단에서 어떤 함수를 사용할 것이라고 컴파일러에 정보 주는 역할

① 함수정의

(함수명, 매개변수, 반환형 필요)

함수 원형 function prototype

```
반환형 함수명(매개변수 1, 매개변수 2)
{
    // 함수가 수행하는 명령. 결과값을 돌려보냄
}
```

② 함수 호출, 반환 (인수)

· 인수 (argument) 사용하여 호출

상수, 변수, 수식 가능

→ 호출된 함수의 매개변수로 사용

· 값을 반환할 때 return 사용

③ 함수 선언 (형태 알람)

`int sum(int x, int y);`

↑ 함수명 ↓ 매개변수
반환형

· 함수 선언이 필요한 이유

-1. 함수 선언에서 반환값의 형태 확인 (OR 함수 호출 이전에 함수 정의)

-2. 함수의 호출 형식에 문제가 없는지 검사.

↳ 매개변수의 개수와 형태에 대한 정보가 있으므로 호출할 때 정확한 값 주는지 검사 가능

여러가지 함수 유형

· 매개변수가 없는 함수 선언 `int get-num(void);`
`int get-num();`

호출하는 함수 → ① 호출 → 매개변수가 없는 함수 → ② 입력 키보드
← ③ 양수 반환

+ 함수의 매개변수 자리에 void 사용

· 반환 형이 없는 함수 선언 `void print-char(char ch, int count);`

→ 선언과 정의의 반환형에 void 사용

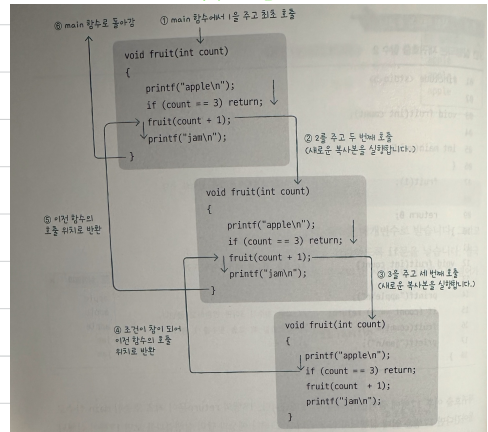
return 값 생략 가능

· 매개변수, 반환값 모두 없는 함수 선언 `void print-title(void);`

↳ void 사용

· 재귀 호출 함수 선언 `void fruit() { ... fruit(); ... }`

★ 재귀호출 멈추는 조건 !!



· 재귀호출 함수는 이전에 호출했던 곳으로 돌아감

(복잡한 반복문을 간단히 표현할 수 있으나 코드 읽기가 쉽지 않고 반복 호출되며 메모리 사용)