

# Norm

## Conventions de denomination - 명명 규칙

- **s\_** 구조체 시작
- **t\_** typedef 된 자료형 시작
- **u\_** 공용체 시작
- **e\_** enum 시작 (enumeration - 열거)
- **g\_** 전역변수의 시작

- 함수

모든 함수는 이름이 정의된 **매개변수**를 최대 **네 개**까지 가질 수 있습니다.

인자를 받지 않는 함수는 선언 시 매개변수 자리에 **"void"**를 명시적으로 표기해야 합니다.

함수 선언 시 **매개변수**의 **이름**이 명시되어야 합니다.

모든 함수는 하나의 빈 줄로 구분되어있어야 합니다.

한 구문 당 최대 **5개**의 **변수**만 선언할 수 있습니다.

함수의 반환 값은 한 쌍의 괄호 안에 표기되어야 합니다.

- **전역변수**를 사용할 때에는 확실한 이유가 있어야 합니다.
- 모든 함수는 중괄호를 제외하고 **25줄** 이하여야 합니다.

한 줄에 **80자** 이하로 작성해야 합니다.

한 줄에 **단 하나의 명령문**만 사용해야 합니다.

- 중괄호{ } 다음 / 제어구조(if, while 등)가 끝나는 경우 다음 줄로 넘어가야 합니다.
- 줄의 마지막이 아닌 이상, 모든 **coma( , )**와 **세미콜론( ; )** 뒤에는 **공백**이 따라와야 합니다.
- **이진 연산자** / **삼진 연산자**를 사용할 때, 모든 연산자와 **피연산자** 사이에는 **공백**이 있어야 합니다.
- 모든 **C 키워드** 다음에는 **공백**이 따라와야 합니다. <자료형(int, char, float 등)과 sizeof 는 제외>
- 한 줄에 **단 하나의 변수** 선언. **전역변수**와 **정적변수**를 제외하고는 **선언**과 **초기화**를 한 줄에서 할 수 없습니다.
- 선언은 함수의 시작부에서 이루어져야 하고 **한 줄**을 띄워놓음으로써 구분해야 합니다.
- **다중대입(multiple assignment ex. a = b = 0)**는 금지됩니다.
- 명령어나 제어구조의 뒤에는 줄 바꿈이 나타나야 합니다. 괄호나 대입 연산자가 있을 경우에는 **들여쓰기(tab)**를 해주어야 합니다.
- 구조체나 공용체, enum(서로 연관된 상수들의 집합)을 선언할 때에는 **탭**을 삽입하시기 바랍니다.
- **struct, enum, union** 타입의 변수를 선언할 때에는 **자료형**에 **스페이스**가 하나 존재해야 합니다.

- `typedef`를 사용할 때에는 두 `인자` 사이에 `탭`을 넣어주시기 바랍니다.
- `struct`, `union`, `enum`을 `typedef`로 선언할 때에는 위의 모든 규칙이 적용됩니다.
- `typedef`의 이름은 `struct/union/enum`의 이름에 `정렬`시켜야 합니다.
- `.c` 파일에 구조체를 선언하는 것은 금지됩니다.
- 헤더파일

헤더파일에는 다음의 작업들만 허용됩니다 : `헤더파일 인클루드`, `선언`, `정의`, `프로토타입`, `매크로`

모든 인클루드(`.c/.h`)는 파일의 `시작부`에 위치해야 합니다.

중복 인클루드를 방지해야 합니다. 예를 들어, 파일 이름이 `ft_foo.h`일 경우, 인클루드 가드는 `FT_FOO_H`가 됩니다.

함수의 `프로토타입`은 `.h`파일에 명시적으로 표기되어야 합니다.

- 매크로

`매크로 상수`(`#define`)는 문자 그대로의 문자열이나 상수에만 사용 가능합니다.

`norm`을 우회하거나 코드를 알아보기 어렵게 만드는 모든 `#define`은 금지됩니다.

<이 점은 사람에게 의해 검사되어야 합니다.>

`표준라이브러리`의 매크로는 프로젝트에서 사용이 허가되었을 경우에만 사용 가능합니다.

여러 줄에 걸친 매크로는 금지됩니다.

매크로의 이름은 `대문자`로만 써야 합니다.

- `#if`, `#ifdef`, `#ifndef` 다음에 나오는 구문들은 `들여쓰기`(`tab`)가 되어야 합니다.
- 다음 구문은 사용이 금지됩니다 : `for`, `do...while`, `switch`, `case`, `goto`, 중첩된 `삼진 연산자`. (예 : `?').
- VLA - 가변 길이 배열.
- 주석

소스 파일 내에서 주석을 사용할 수 있습니다.

함수 내부에 주석을 사용하는 것은 금지됩니다.

주석의 시작과 끝은 별개의 줄로 표시되어야 합니다.

중간의 모든 줄은 `정렬`이 맞춰져야 하고 `**`로 시작해야 합니다.

`//`는 사용이 금지됩니다.

- `.c` 파일 하나당 `최대 5개`까지의 `함수 정의`를 포함할 수 있습니다.
- 다음 룰(rule)들은 필수적으로 만들어야 합니다 : `$(NAME)`, `clean`, `fclean`, `re`, `all`
- `makefile`이 `리링크`(`relink`)를 할 경우, 프로젝트는 정상적으로 작동하지 않는 것으로 처리됩니다.

- 실행 파일이 여러 개일 경우, 각각의 실행 파일을 컴파일하는 것처럼 모든 실행 파일을 컴파일하는 룰(rule)을 첫 번째 룰로 작성해야 합니다.
- 지금까지 본 규칙 외에도 바이너리를 컴파일하는 규칙과 컴파일된 각 바이너리에 대한 특정 규칙이 있어야 합니다.
- 라이브러리(예: `libft`)를 사용하는 프로젝트의 경우, `makefile`이 해당 라이브러리를 자동으로 컴파일하도록 만들어야 합니다.
- 프로젝트를 컴파일하기 위하여 필요한 모든 소스파일은 `makefile`에 명시적으로 기입되어야 합니다.