

# C Programming

## Ch.1-2 프로그램의 기본 구성

# Contents

---

1. Hello world! 들여다보기
2. 주석이 들어가야 완성된 프로그램
3. printf 함수의 기본적인 이해

# 첫 번째 프로그램

- ▶ 다음 프로그램을 작성하고 실행시켜 보라 (Hello.c)

```
#include <stdio.h>

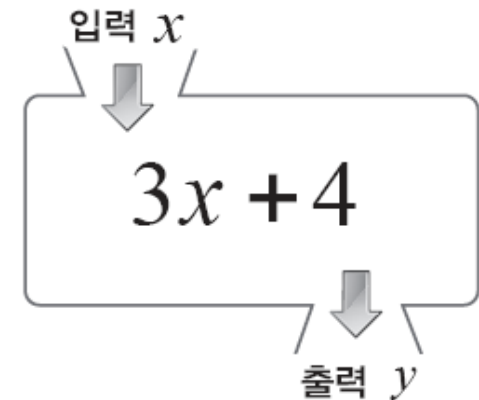
int main(void)
{
    printf("Hello, world!\n");
    return 0;
}
```

Hello, world!

실행 결과

# C 언어의 기본 단위인 '함수'의 이해

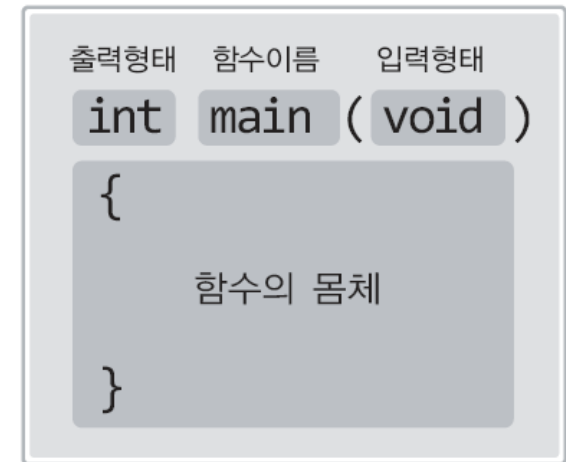
- ▶ C 언어의 기본단위 : 함수
  - 함수를 만들고, 만들어진 함수의 실행순서를 결정하는 것이 C 언어로 프로그램을 작성하는 것!
- ▶ 함수의 기본특성
  - 수학적으로 함수에는 입력과 출력이 존재
- ▶ C 언어의 함수
  - C언어의 함수에도 입력과 출력이 존재
- ▶ 용어 정리
  - 함수의 정의 : 만들어진 함수, 실행이 가능한 함수 자체
  - 함수의 호출 : 함수의 실행을 명령하는 행위
  - 인자의 전달 : 함수의 실행을 명령할 때 전달하는 입력 값



C 언어는 함수로 시작해서 함수로 끝이 난다.

# Hello.c에서의 함수는 어디에?

- ▶ 프로그램의 시작
  - 첫 번째 함수가 호출되면서 프로그램 시작
- ▶ 제일 먼저 호출되는 함수는?
  - main이라는 이름의 함수!
  - 모든 프로그램은 반드시 main 함수 정의
  - 즉, main이라는 이름의 함수가 자동으로 호출되면서 프로그램이 실행됨
- ▶ 함수의 기능(내용)
  - 함수의 기능은 중괄호 안에 표현됨
  - 중괄호 안에 표현된 함수의 기능을 함수의 몸체라 함
- ▶ 함수에 표시되는 세 가지
  - 함수 이름 : 함수를 호출할 때 사용하게 되는 이름
  - 출력 타입 : 실행 결과 반환되는 값의 종류! 일반적으로 반환형(return type)이라 한다.
  - 입력 데이터(들) : 함수를 호출할 때 전달되는 입력 값(들)의 종류와 변수



```
int main(void)
{
    printf("Hello world! \n");
    return 0;
}
```

↓  
순차적으로 실행

# 문장의 끝 - 세미콜론 (;)

- ▶ 문장의 끝을 표현하기 위한 문자 : 세미콜론 ;
  - 함수 내에 존재하는 문장의 끝에는 세미콜론 문자(;)를 붙임
- ▶ 열 줄에 표현된 코드는 열 개의 문장인가?
  - 하나의 문장이 둘 이상의 줄에 표시될 수도 있고, 한 줄에 둘 이상의 문장이 표시될 수도 있음
  - 즉, 줄 바꿈은 문장의 바꿈을 뜻하는 것이 아님!
- ▶ 한 줄에 하나의 문장을 표시하는 것이 가장 일반적이고 또 보기도 좋다
  - 다음 세 main 함수는 모두 동일한 프로그램이다. 줄 바꿈의 차이가 프로그램의 이해도 차이로 이어지지 않음

```
int main(void)
{
    printf("Hello world! \n"); return 0;
}
```

```
int main(void)
{
    printf("Hello world! \n");
    return 0;
}
```

```
int main(void) { printf("Hello world! \n"); return 0; }
```

# 소스 코드의 세부 분석 (1)

표준 = standard

- ▶ 표준 함수
  - 미리 만들어져 기본적으로 제공되는 함수
  - printf 함수 등
- ▶ 표준 라이브러리
  - 표준 함수들의 집합
  - 즉, printf 함수는 표준 라이브러리의 일부
- ▶ `#include <stdio.h>`
  - `stdio.h` 파일의 내용을 이 위치에 가져다 놓으라는 뜻
  - printf 함수의 호출을 위해서 선언해야 하는 문장
  - `stdio.h` 파일에는 printf 함수 호출에 필요한 정보 존재

```
#include <stdio.h>
int main(void)
{
    printf("Hello world! \n");
    return 0;
}
```

헤더파일 선언문

처음 보는 함수의 호출문

# 소스 코드의 세부 분석 (2)

- ▶ `printf("Hello world!\n");`
  - `printf`라는 이름의 함수를 호출하는 문장
  - 인자는 문자열 `"Hello world!\n"`
  - 인자는 소괄호를 통해서 해당 함수에 전달이 된다.
- ▶ `return 0;`
  - 함수를 호출한 영역으로 값을 전달(반환)
  - 현재 실행 중인 함수의 종료
- ▶ `return` 문의 특징
  - 반환 타입이 `void`라면 존재하지 않을 수도 있음
    - `return` 문이 없는 경우
      - 함수 끝에서 자동 `return`

```
#include <stdio.h>  헤더파일 선언문
int main(void)
{
    처음 보는 함수의 호출문
    printf("Hello world! \n");
    return 0;
}
```



# 주석 (설명글)

## ▶ 주석이란?

- 프로그래머에게 메모(memo)의 기능을 부여
- 컴파일러는 주석을 없는 것으로 간주
- 주석을 삽입함으로써 프로그램의 가독성 증가 → 유지 보수 용이
- 선택이 아닌 필수!

## ▶ 주석의 두 가지 형태

- 여러 줄 또는 한 줄에 걸친 주석 처리
  - `/*       */`
- 단일 행 주석 처리
  - `//`
- 이후의 줄 끝까지는 주석이 됨

```
/* 한 줄 짜리 주석 */
/*
    여러 줄에
    걸친 주석
*/

// 주석 하나
// 주석 둘
// 주석 셋
```

# 주석 처리의 예

```
/*
제 목: Hello world 출력하기
기 능: 문자열의 출력
파일이름: HelloComment.c
수정날짜: 2014. 07. 15
작성자: 윤성우
*/

#include <stdio.h>    // 헤더파일 선언

int main(void)    // main 함수의 시작
{
    /*
    이 함수 내에서는 하나의 문자열을 출력한다.
    문자열은 모니터로 출력된다.
    */
    printf("Hello world! \n");    // 문자열의 출력
    return 0;    // 0의 반환
}    // main 함수의 끝
```

과도하게 처리된 주석(주석도 과하면 좋지 않다)!  
주석을 다는 방법을 소개하기 위한 예제일 뿐이다.

# 주석 처리 시 주의 사항

- ▶ 주석을 나타내는 기호는 중복될 수 없음

```
/* 주석의 시작, 여러 행에 걸쳐서  
    /* 단일 행 주석 처리 */  
    이어서 주석  
*/
```

에러 발생

- ▶ 단, 단일 행 주석은 중복 가능

```
/* 주석의 시작, 여러 행에 걸쳐서  
    // 단일 행 주석 처리  
    이어서 주석  
*/
```

# printf 함수의 사용 예

```
/* printf1.c */  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello Everybody \n");  
    printf("%d \n", 1234);  
    printf("%d %d \n", 10, 20);  
    return 0;  
}
```

%d ?  
\n ?

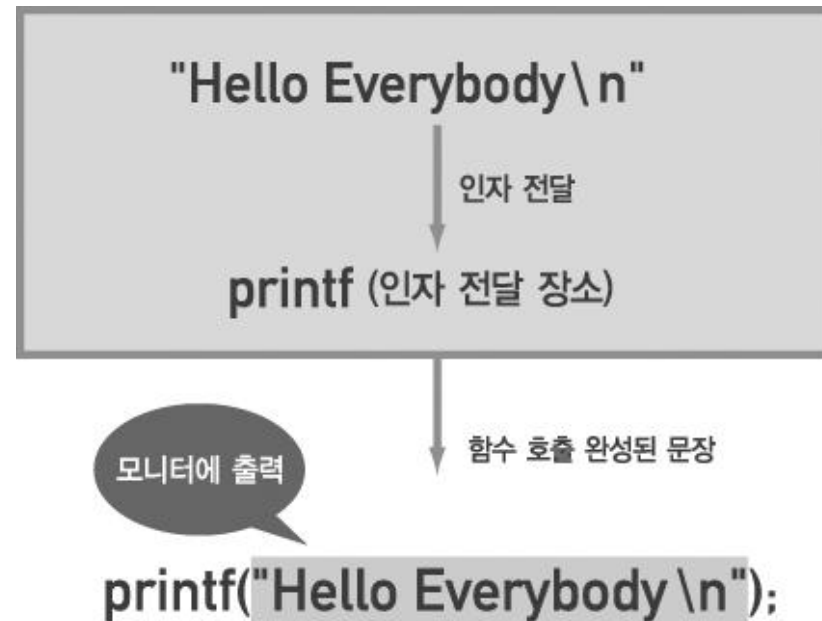
```
Hello Everybody  
1234  
10 20
```

# printf 함수의 호출 이해 (1)

## ▶ printf 함수의 출력 대상

```
printf("Hello Everybody \n");
```

- 큰 따옴표로 표시되는 문자열



# printf 함수의 호출 이해 (2)

## ▶ 서식 문자 : 출력 대상의 출력 형태 지정

```
printf("%d \n", 1234);  
printf("%d %d \n", 10, 20);
```

- 문자열에 삽입된 %d를 가리켜 ‘서식 문자’라 함
- %d 자리에 10진수 정수의 값이 출력됨을 의미
- 출력의 대상은?
  - 문자열 뒤에 이어서 표시
  - 콤마로 구분하여 여러 개의 서식 문자에 대응하는 데이터들 표기 가능
  - 서식 문자가 2개면 출력의 대상도 반드시 2개여야 함

## ▶ 이스케이프 시퀀스(escape sequence)

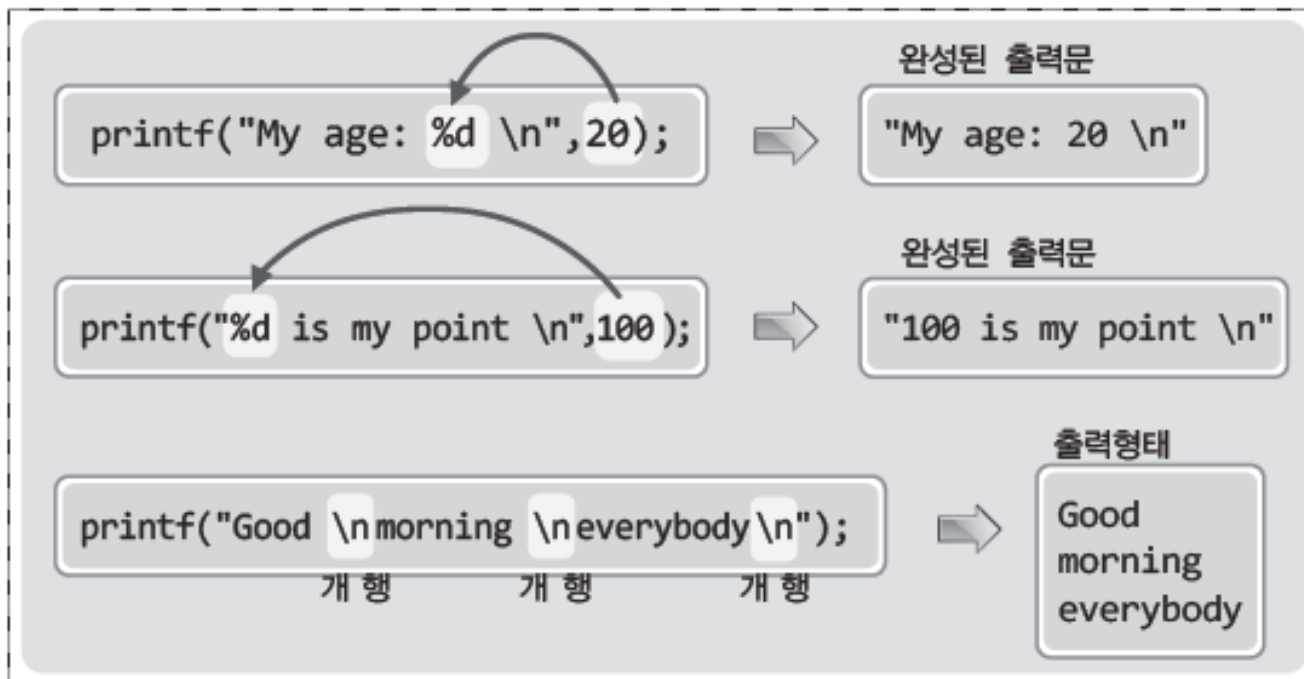
- \n : 다음 줄을 의미하는 특수 문자

# printf 함수의 사용 예

```
int main(void)
{
    printf("My age: %d \n", 20);
    printf("%d is my point \n", 100);
    printf("Good \nmorning \neverybody\n");
    return 0;
}
```

실행결과

```
My age: 20
100 is my point
Good
morning
everybody
```



이후 다양한  
서식 문자와  
이스케이프  
문자에 대해  
공부함

# 이번 장에서 배운 것

- C 프로그램은 main 함수를 시작으로 프로그램이 실행된다.
- 함수는 이름, 전달인자, 반환 타입, 내용으로 구성된다.
- 함수를 실행하기 위해서는 함수이름(전달인자)와 같이 호출한다.
- 주석은 실제 실행되는 내용은 아니지만 프로그래머에게 코드에 대한 정보를 주기 위해 활용된다.
- printf 함수는 표준 함수이다.
- printf 함수는 출력할 문자열을 전달할 수 있으며, 문자열 내에 서식 문자와 이스케이프 문자를 사용할 수 있다.