

K G 아 이 티 뱅 크

C언어

V I S U A L   S T U D I O

# C프로그래밍의 구성

# C프로그래밍의 기본구조

〈파일이름 : 01.기본구조.c〉

```
#include<stdio.h>
```

```
int main(void) ← main 함수임을 알림
```

```
{ ← main 함수의 몸체 시작
```

```
    printf("사과는 ");  
    printf("맛있어 ");  
    printf("맛있는 건 ");  
    printf("바나나");
```

```
return 0; ← main 함수의 종료
```

```
} ← main 함수의 몸체 종료
```

## C프로그래밍의 기본구조

---

- `#include` : ~을 포함한다. (전처리 문)
- `<stdio.h>` : 입출력을 위한 헤더 파일(헤더 파일 : 필요한 함수를 사용하기 위해 포함해 주는 파일)
- `main()` : 코드에서 한 개만 사용할 수 있는 기본함수, 가장 먼저 실행하는 함수
- `void` : 인수 없이 함수 사용, 입력 받는 값이 없다.
- `void main(void)` : 입력 받는 값 없고, 출력 하는 값도 없음(return 필요X)
- `return 0` : main 함수 종료
- 문자열 출력 : 큰 따옴표로 묶어서 표현
- 문장의 끝은 세미콜론(;) 으로 마무리

# 주석과 들여쓰기

<파일이름 : 02.주석과 들여쓰기.c>

```
/*
    파일이름 : 02.주석과 들여쓰기.c
    작성자 : 송진우
    작성일 : 2018년 3월 10일
*/
```

/\*\*/ 블록 단위 주석  
컴파일의 대상에서  
제외 되는 영역

```
#include<stdio.h>
int main(void)
```

```
{
    int a = 0;    //변수 선언 및 초기화
    a = 1 + 2;    //덧셈 연산 후에 대입연산
    printf("1 + 2 = %d ", a);
}
```

가독성을 위해  
존재하는  
들여쓰기(tab)

// 행 단위 주석  
한 행을 (가로줄)  
컴파일 제외

# ESCAPE문자

<파일이름 : 03.ESCAPE.c>

```
#include<stdio.h>

int main(void)

{
    printf("바나나는 %t 길어 %n");
    printf("긴건건 %b%b %n");
    printf("기차 %a %n");
    printf("나는 %r기차는 빨라 %n");
    printf("빠른건 %"비행기%" %n");
    printf("비행기는 12억%% %n");

    return 0;
}
```

## ESCAPE문자

- ESCAPE 문자 : 문자열 내에서 다른 의미로 해석 된다.

ESCAPE 문자	기      능	내부 코드 (ASCII) 값
'\n'	New line(새로운 줄로 이동)	10
'\r'	Carriage return(줄의 처음으로)	13
'\b'	Back space(한 문자 왼쪽으로)	8
'\t'	Tab(탭 크기만큼 이동)	9
'\a'	Alarm (벨 소리 냄)	7

## 연습 문제

<파일이름 : 04.예제1.c>, <파일이름 : 05.예제2.c>

예제1)

아래와 같이 출력되도록 코드를 작성하여 예제1.c로 저장한 후 실행해보세요.

[출력결과]

```
#####  
# C 프로그래밍의 세계에 발 담그다! #  
#####
```

예제2)

아래와 같이 출력되도록 코드를 작성하여 예제2.c로 저장한 후 실행해보세요.

[출력결과]

```
*  
***  
*****  
***  
*
```



## 서식문자 (제어문자)

- 서식 문자 : 문자열의 중간에 삽입되어 출력형식을 지정한다.

제어문자	출력결과
%d	10진(Decimal) 정수
%o	8진수(Octal) 정수
%x	16진(hexa_decimal) 정수
%p	포인터 정수(16진수)
%u	부호없는(Unsigned) 10진 정수
%f	10진형 소수점수
%c	단일 문자(Character)
%s	문자열(String)

## 서식문자 (제어문자)

● ★ ■

```
printf("정수 실수 문자", 10, 0.2, 'c');
```

The diagram illustrates the mapping of format specifiers to data types in a `printf` statement. Above the format specifiers, there are three colored shapes: a blue circle for '정수' (integer), a yellow star for '실수' (floating-point number), and a green square for '문자' (character). Below the format specifiers, there are three colored lines: a blue line connecting '정수' to '10', a yellow line connecting '실수' to '0.2', and a green line connecting '문자' to ''c''. This shows that the format specifiers determine the output format for the corresponding arguments.

여러 개의 출력형태를 결정하려면 ,(콤마)로 구분  
어떤 형태의 데이터를 출력할 것인가에 따라 다른  
서식문자가 필요

## 서식문자 (제어문자)

<파일이름 : 06.서식문자.c>

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
printf("[사과] 가격 : %d₩₩ ₩n", 1000);
```

```
printf("[사과] 가격 : %d₩₩ 무게 : %.2f kg ₩n", 1000, 1.2);
```

```
printf("[사과] 가격 : %d₩₩ 무게 : %.2f kg 제조 : %c ₩n", 1000, 1.2, 'c');
```

```
printf("[사과] 가격 : %d₩₩ 무게 : %.2f kg 제조 : %s ₩n", 1000, 1.2, "얼음골");
```

```
return 0;
```

```
}
```

## 진법

### ● 진법(진수) : 수를 표기하는 방법

진법	범위	표현식	사용 예
2진수	0, 1		0100 0001(C언어로 표기 불가)
8진수	0 ~ 7	0(숫자)	0101
10진수	0 ~ 9		65
16진수	0 ~ 9, A ~ F	0x	0x41

### ● 2진수를 10진수로 변환

예 > 1111

$$\begin{aligned} &= 2^3 * 1 + 2^2 * 1 + 2^1 * 1 + 2^0 * 1 \\ &= 8 + 4 + 2 + 1 \\ &= 15 \end{aligned}$$

### ● 10진수 <-> 2진수

① 1101 → 10진수

② 68 → 2진수

## ● 2진수와 16진수 사이의 변환

❖ 16진수 한자리는 2진수 네자리로 표현 가능

예 > 16진수 → 2진수

F E 5 → 1111 1110 0101

예 > 2진수 → 16진수

1011 0110 0101 → B 6 5

예 제> 16진수 ↔ 2진수

① 12DE → 2진수

② 110000101001 → 16진수

10진수	2진수	16진수
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

### ● 2진수와 8진수 사이의 변환

❖ 8진수 한자리는 2진수 세자리로 표현 가능

예 > 8진수  $\rightarrow$  2진수

7 3 2  $\rightarrow$  111 011 010

예 > 2진수  $\rightarrow$  8진수

001 100 101 001  $\rightarrow$  1 4 5 1

예 제> 8진수  $\leftrightarrow$  2진수

① 530  $\rightarrow$  2진수

② 10010011  $\rightarrow$  8진수

# 진법

<파일이름 : 07.진법.c>

```
int main(void)
{
    //10진수(Decimal number) : %d
    printf("Decimal : %d\n", 173);
    printf("Decimal : %d\n", 0255);
    printf("Decimal : %d\n", 0xAD);

    //8진수(Octal number) : %o
    printf("Octal : %o\n", 173);
    printf("Octal : %o\n", 0255);
    printf("Octal : %o\n", 0xAD);

    //16진수(Hexadecimal) : %x
    printf("Hexadecimal : %x\n", 173);
    printf("Hexadecimal: %x\n", 0255);
    printf("Hexadecimal : %x\n", 0xAD);

    //2진수(Binary number)는 c언어에서 표시안됨

    return 0;
}
```