



C 프로그램의 기본

목차

- C 프로그램의 구성 요소
 - 주석
 - main 함수
 - 출력
- C 언어의 입력과 출력
 - 변수
 - printf 함수
 - scanf 함수

예제 2-1 : 첫 번째 C 프로그램

```
01 // ex02_01.c
02 #include <stdio.h> // 입출력 라이브러리를 사용하기 위한 준비
03                                     주석
04 int main(void) // 진입점 함수
05 {
06     printf("첫 번째 C 프로그램입니다.\n"); // 콘솔 출력
07
08     return 0; // 프로그램의 종료 코드 리턴
09 }
```

실행결과

첫 번째 C 프로그램입니다.

주석

- 주석을 다는 방법
 - /*과 */을 이용한 여러 줄 주석
 - //을 이용한 한 줄 주석
- 주석의 용도
 - 프로그램에 대한 설명

```
#include <stdio.h> // 입출력 라이브러리를 사용하기 위한 준비
```

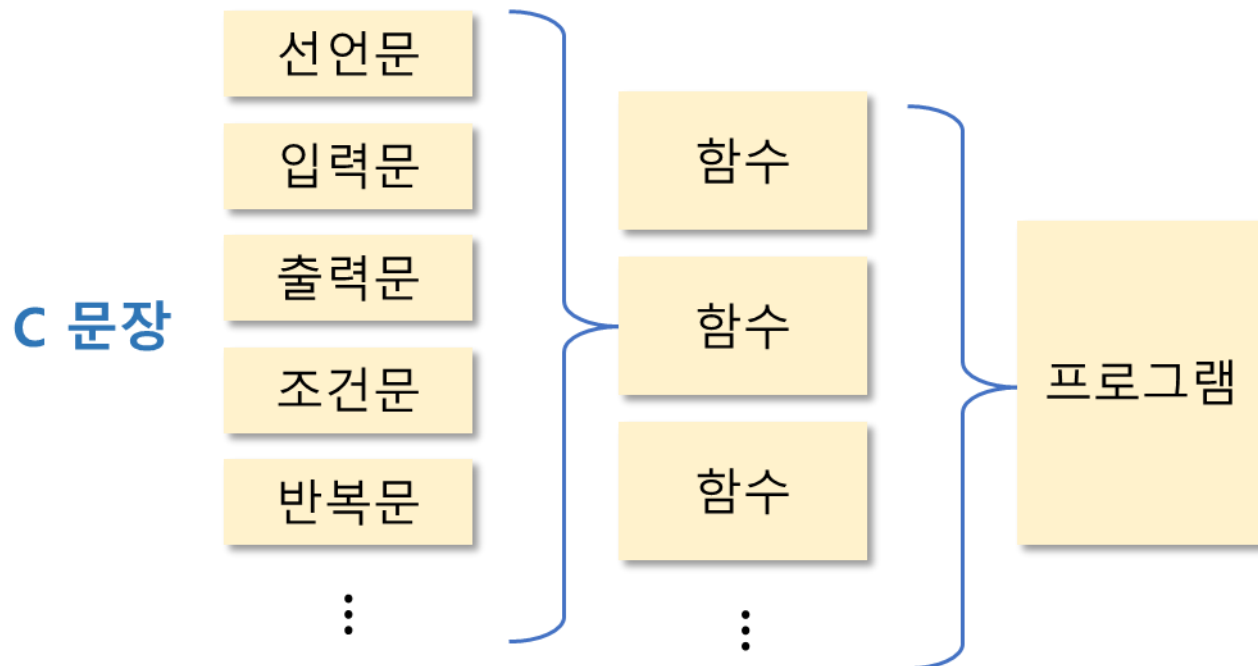
- 프로그램 전체에 대한 대략적인 정보를 제공
- 주석 처리(comment out)

```
printf("첫 번째 C 프로그램입니다.\n");  
//printf("주석 처리된 문장입니다.\n");
```

```
/*  
    프로그램명: ex02_01  
    설명: 간단한 출력을 수행하는 C 프로그램  
    작성 일시: 2019/1/1  
    작성자: 천정아  
*/
```

문장(statement)

- C 프로그램을 이루는 기본 단위
- 각 문장은 세미콜론(;)으로 끝난다.
- ~문: 선언문, 입력문, 출력문, 조건문, 반복문 등



콘솔 프로그램과 main 함수

- 콘솔 프로그램에는 반드시 main 함수가 필요하다.

main 함수로만 구성된 프로그램

```
int main(void)
{
    return 0;
}
```

} main 함수

여러 개의 함수로 구성된 프로그램

```
void print_line(void)
{
    printf("-----\n");
}

int add(int a, int b)
{
    return a + b;
}

int main(void)
{
    return 0;
}
```

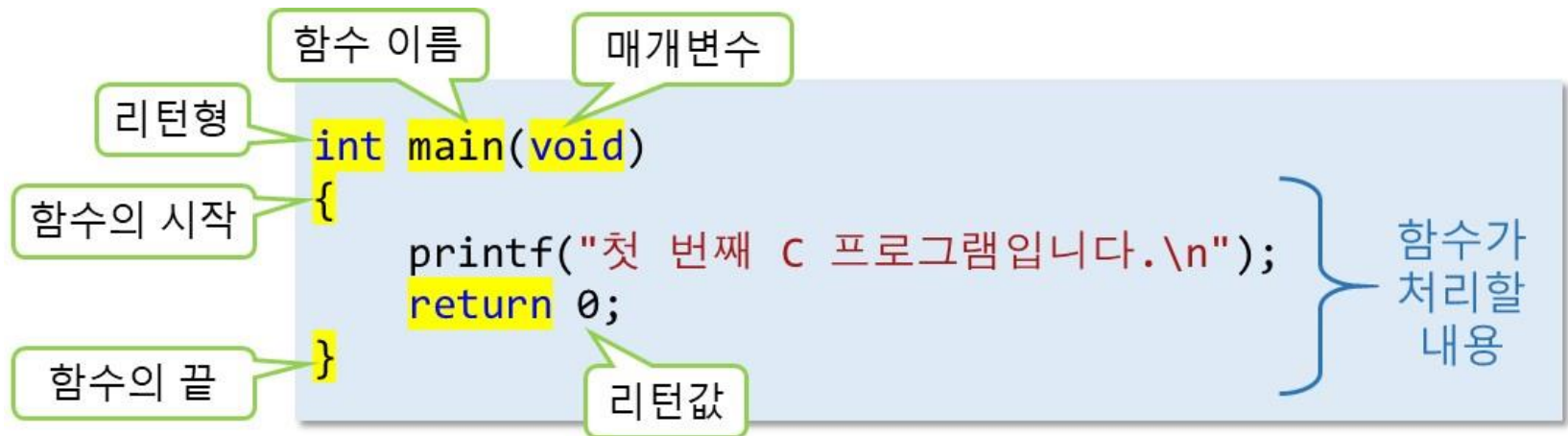
} 함수

} 함수

} main 함수

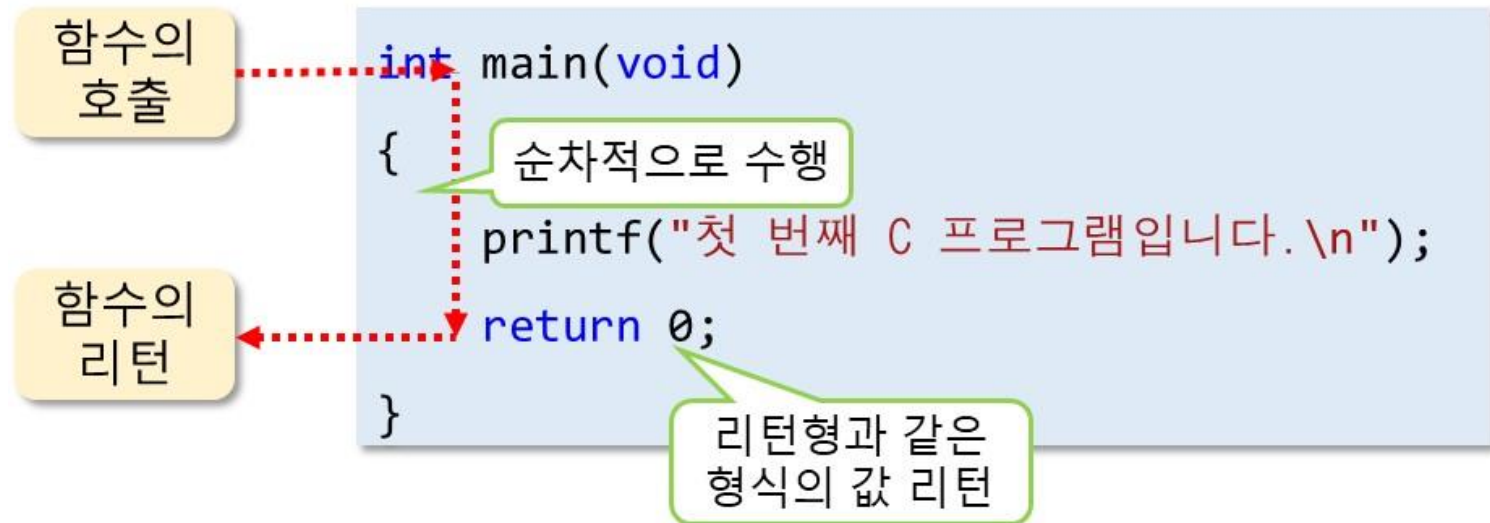
함수의 구성 요소

- 함수를 만들 때는 리턴형, 함수 이름, 매개변수가 필요하다.



진입점 함수

- C 프로그램이 처음 시작될 때 호출되는 함수
- main 함수는 콘솔 프로그램의 진입점 함수이다.



- C 프로그램에는 main 함수가 반드시 필요하다.

main 함수의 리턴 값

- 프로그램의 종료 코드(exit code)를 리턴한다.
 - main 함수의 리턴 값은 운영체제로 전달
 - 0이면, 정상 종료이고 0이 아니면 비정상 종료

```
int main()
{
    ...

    return 0;
}
```

정상 종료

```
int main()
{
    if (/*메모리 할당 실패*/)
        return 1;
    ...
}
```

비정상 종료

- main 함수의 return문은 생략할 수 있다.
 - 프로그램 종료 시 0 리턴

들여쓰기

- 알아보기 쉽도록 한 줄에 한 문장씩 작성한다.
- **블록(block)** : { }로 묶인 문장들

```
int main()
{
    printf("Good\n");
    return 0;
}
```

들여 쓰기를 하면
알아보기 쉽다.

```
int main() {
    printf("Not Good\n");
    return 0; }
```

들여 쓰기를 안 하면
알아보기 어렵다.

- Visual Studio의 들여쓰기 단축키
 - Ctrl+K, Ctrl+D 또는 블록 선택 후 Ctrl+K, Ctrl+F

입출력

- 콘솔 프로그램
 - 콘솔(명령 프롬프트)에서 실행되는 프로그램
 - **키보드**로부터 입력을 받아서 처리 결과를 콘솔에 **텍스트**로 출력

콘솔 프로그램



표준 입력

표준 출력

입출력 라이브러리를 사용하기 위한 준비

- <stdio.h>를 포함한다.
- 헤더 파일(.h)
 - 라이브러리 함수의 이름, 리턴형, 매개변수에 대한 정보를 담고 있는 파일

전처리기
문장

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

```
int main(void)
{
    printf("첫 번째 C 프로그램입니다.\n");
    return 0;
}
```

stdio.h의 내용을
이 위치에 복사한다.

```
// stdio.h
#pragma once
#ifndef _INC_STDIO
#define _INC_STDIO

#include <corecrt.h>
#include <corecrt_wstdio.h>
_CRT_BEGIN_C_HEADER
...
```

입출력 라이브러리
헤더 파일

콘솔 출력

- 콘솔에 텍스트를 출력하려면 printf 함수를 이용한다.
 - 출력할 내용을 " "로 묶어서 printf 함수의 () 안에 써준다.

```
printf("첫 번째 C 프로그램입니다.\n"); // 출력하고 커서를 다음 줄로 이동한다.
```

```
int main()  
{  
    printf("Different\n");  
    printf("Line\n");  
}
```

줄 바꿈 문자를 사용하는 경우

실행

```
C:\Windows\system32\cmd.exe  
c:\work\chap02\Ex02_01\Debug>Ex02_01.exe  
Different  
Line  
c:\work\chap02\Ex02_01\Debug>_
```

다른 라인에
출력한다.

출력 후
커서 위치

변수

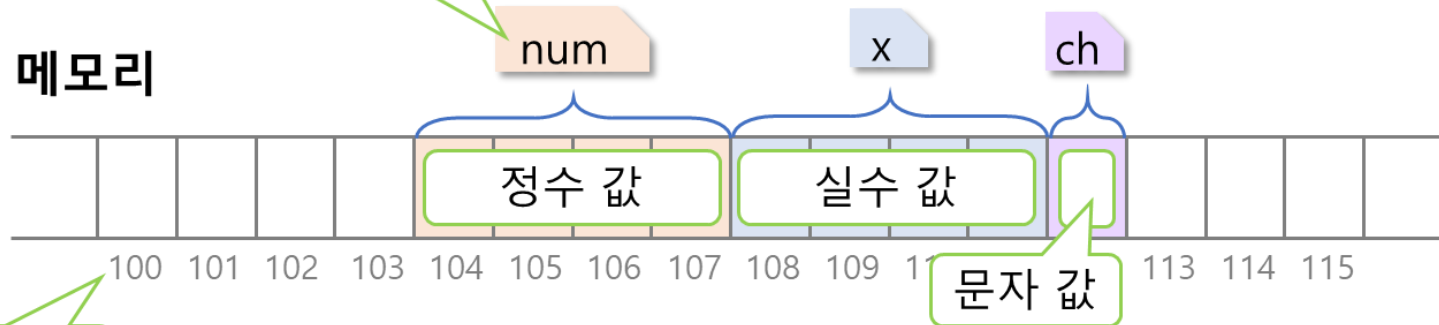
- 어떤 값을 저장하기 위한 공간
- 변수를 사용하려면 **변수명**과 **데이터형(data type)**이 필요하다.

```
int num;    // 정수형 변수 선언  
float x;    // 실수형 변수 선언  
char ch;    // 문자형 변수 선언
```

주소 대신
이름으로 접근

변수는 선언 후
사용해야 한다.

메모리



메모리 주소

변수의 선언

- 변수의 선언
 - 변수명은 영문자와 숫자, 밑줄 기호(_)를 사용해서 만든다.
 - 첫 글자로는 반드시 영문자나 밑줄 기호가 와야 한다.

형식

데이터형 변수명;

사용예

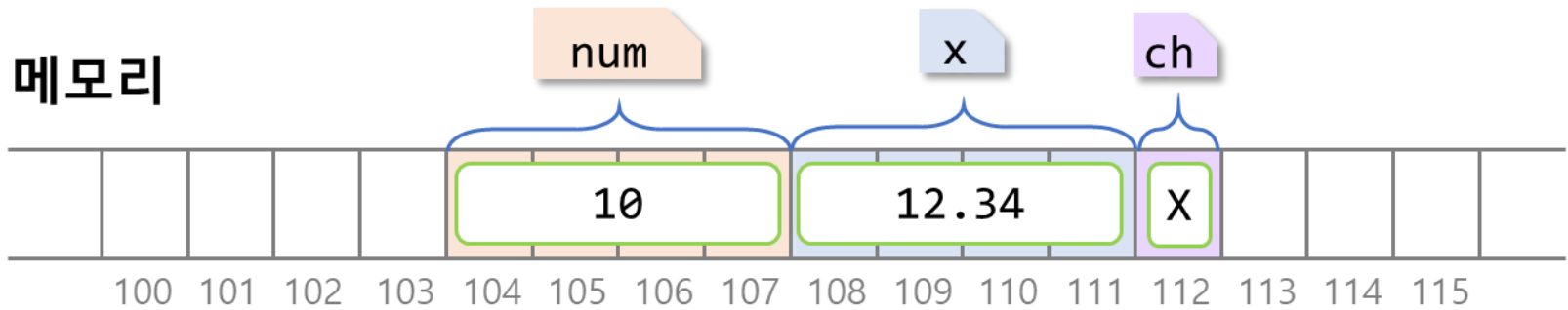
```
int num;  
float x;  
char ch;
```

변수의 사용

- 변수의 사용
 - 변수에 값을 대입하려면 =을 이용한다.

```
num = 10;  
x = 12.34;  
ch = 'X';
```

변수와 같은 데이터
형의 값을 대입한다.



printf 함수의 형식 문자열 (1/3)

서식 지정자	의미	사용 예	실행 결과
%d	정수를 10진수로 출력	<pre>int num = 123; printf("%d", &num);</pre>	123
%x	정수를 16진수로 출력 (0~9, a~f 이용)	<pre>int num = 123; printf("%x", &num);</pre>	7b
%X	정수를 16진수로 출력 (0~9, A~F 이용)	<pre>int num = 123; printf("%X", &num);</pre>	7B
%f, %F	실수를 부동소수점 표기 방식으로 출력	<pre>float x = 1.23; printf("%f", &x);</pre>	1.230000
%e, %E	실수를 지수 표기 방식으로 출력	<pre>float x = 1.23; printf("%e", &x);</pre>	1.230000e+00
%c	문자 출력	<pre>char ch = 'A'; printf("%c", &ch);</pre>	A
%s	문자열 출력	<pre>char name[20] = "abc"; printf("%s", name);</pre>	abc

printf 함수의 형식 문자열 (2/3)

- 서식 지정자의 개수와 출력할 값의 개수가 일치해야 한다.

서식 지정자와 출력할 값이 순서대로 대응된다.

```
printf("%d %x\n", num, num);
```

123 7b

```
printf("%f %e\n", x, x);
```

1.230000 1.230000e+00

서식 지정자와 출력할 값의 개수가 일치해야 한다.

- 16진수 정수 출력

```
printf("%x\n", num);      // 7b 출력
printf("%X\n", num);      // 7B 출력
printf("%#x\n", num);     // 0x7b 출력
printf("%#X\n", num);     // 0X7B 출력
```

예제 2-2 : 형식 문자열을 이용해서 출력하기 [1/2]

```
01  #include <stdio.h>
02
03  int main()
04  {
05      int num;        // 정수형 변수 선언
06      float x;        // 실수형 변수 선언
07      char ch;        // 문자형 변수 선언
08
09      num = 123;       // 정수형 변수에 값 대입
10      x = 1.23;        // 실수형 변수에 값 대입
11      ch = 'A';        // 문자형 변수에 값 대입
12
13      printf("%d %x\n", num, num);    // 10진수, 16진수 출력
14      printf("%f %e\n", x, x);        // 부동소수점, 지수 출력
15      printf("%c\n", ch);              // 문자 출력
```

형식 문자열

형식 문자열 안에 서식 지정자를 여러 개 지정할 수 있다.

예제 2-2 : 형식 문자열을 이용해서 출력하기 [2/2]

```
17 printf("%x\n", num);    // 7b 출력
18 printf("%X\n", num);    // 7B 출력
19 printf("%#x\n", num);   // 0x7b 출력
20 printf("%#X\n", num);   // 0X7B 출력
21
22 return 0;
23 }
```

#을 지정하면 0x나 0X를 함께 출력

실행결과

```
123 7b
1.230000 1.230000e+00
A
7b
7B
0x7b
0X7B
```

10진수 123은 16진수 7b에 해당

printf 함수의 형식 문자열 (3/3)

- 문자 폭 지정

```
printf("%08d\n", num);    // 빈칸 대신 문자 폭의 남은 부분에 0 출력
```

- 정밀도 지정

- 실수의 정밀도 : 소수점 이하 자릿수

```
printf("%f\n", x);        // 폭과 정밀도를 지정하지 않는 경우  
printf("%.2f\n", x);|    // 정밀도를 지정하는 경우 (소수점 이하 2자리)  
printf("%8.2f\n", x);    // 폭과 정밀도를 지정하는 경우
```

- 정수의 정밀도 : 출력할 숫자의 자릿수

```
printf("%8.4d\n", 123);    // 0000123 출력 (0은 빈칸)
```

예제 2-3 : 문자 폭 지정하기

```
03 int main()
04 {
05     int num = 12345;
06
07     printf("%d\n", num);
08     printf("%d\n", num * 10);
09     printf("%d\n", num * 100);
10     printf("%d\n", num * 1000);
```

%d는 폭을 지정하지 않았으므로
왼쪽에서부터 출력

```
11
12     printf("%8d\n", num); // 8문자 폭에 맞춰서 출력
13     printf("%8d\n", num * 10);
14     printf("%8d\n", num * 100);
15     printf("%8d\n", num * 1000);
16     printf("%8d\n", num * 100000); // 문자 폭보다 큰 수 출력
```

```
17
18     printf("%08d\n", num); // 빈칸 대신 문자 폭의 남은 부분에 0 출력
```

실행결과

12345

123450

1234500

12345000

12345

왼쪽에 빈칸 3개 출력

123450

왼쪽에 빈칸 2개 출력

1234500

왼쪽에 빈칸 1개 출력

12345000

1234500000

폭보다 큰 값은 폭을 넘어서 출력

00012345

왼쪽에 0을 3개 출력

예제 2-4 : 정밀도 지정하기

```
01  #include <stdio.h>
02
03  int main()
04  {
05      float x;
06
07      x = 12.34567;           // x에 값 저장
08
09      printf("%f\n", x);      // 폭과 정밀도를 지정하지 않는 경우
10      printf("%.2f\n", x);   // 정밀도를 지정하는 경우 (소수점 이하 2자리)
11      printf("%8.2f\n", x);  // 폭과 정밀도를 지정하는 경우
12
13      return 0;
14  }
```

실행결과

12.345670 디폴트로 소수점 이하 6자리 출력

12.35 소수점 이하 2자리 출력

12.35 전체 8문자 폭에 소수점 이하 2자리 출력

scanf 함수의 형식 문자열 (1/3)

서식 지정자	의미	사용 예
%d	정수를 10진수로 입력	<pre>int num; scanf("%d", &num);</pre>
%x	정수를 16진수로 입력	<pre>int num; scanf("%x", &num);</pre>
%i	정수를 10진수, 8진수, 16진수로 입력 (012는 8진수, 0x12는 16진수)	<pre>int num; scanf("%i", &num);</pre>
%f	float형 실수 입력	<pre>float x; scanf("%f", &x);</pre>
%lf	double형 실수 입력	<pre>double y; scanf("%lf", &y);</pre>
%c	문자 입력	<pre>char ch; scanf("%c", &ch);</pre>
%s	문자열 입력	<pre>char name[20]; scanf("%s", name);</pre>

예제 2-5 : 입력받은 10진수 정수를 16진수로 변환해서 출력하기

```
01  #include <stdio.h>
02
03  int main()
04  {
05      int num;           // 정수형 변수 선언
06
07      printf("정수? ");  // 정수를 입력을 하도록 사용자에게 알려주기 위한 출력문
08      scanf("%d", &num); // num에 10진수로 정수 입력
09
10      printf("10진수 %d는 16진수 %x입니다.\n", num, num);
11
12      return 0;
13  }
```

10진수 정수출력

16진수 정수출력

실행결과

정수? 10 — 10진수 정수 입력
10진수 10는 16진수 a입니다.

scanf 함수의 형식 문자열 (2/3)

- 문자 배열에 입력받을 때는 &를 지정하지 않는다

```
char name[20];    // 문자 20개를 연속으로 저장하는 변수
scanf("%s", name); // name에는 &가 필요 없다.
```

- 서식 지정자를 여러 개 사용할 수도 있다.
 - 서식 지정자와 입력받을 변수의 개수가 같아야 한다.

```
scanf("%s %d %c", name, &age, &gender);
```

3개

3개

- 실수형 변수 입력
 - %f : float 입력
 - %lf : double 입력

```
float x;
double y;
scanf("%f", &x);
scanf("%lf", &y);
```

예제 2-6 : 여러 개의 서식 지정자 사용하기

```
03  int main()
04  {
05      char name[20];
06      int age;
07      char gender;
08
09      printf("이름, 나이, 성별(M/F) 순으로 입력하세요.\n");
10      scanf("%s %d %c", name, &age, &gender); // 3개의 변수 입력
11      printf("이름: %s\n", name);
12      printf("나이: %d\n", age);
13      printf("성별: %c\n", gender);
14
15      return 0;
16  }
```

실행결과

이름, 나이, 성별(M/F) 순으로 입력하세요.

김석진 25 M

세 가지 항목을 빈칸으로 구분해서 입력

이름: 김석진

나이: 25

성별: M

name은 문자 배열이므로
&가 필요 없다.

scanf 함수의 형식 문자열 (3/3)

- 형식 문자열의 공백 문자
 - 이전 입력 이후의 공백을 모두 무시하고 다음 입력을 읽어 오게 한다.

