

개정3판

Visual  
Studio  
2017

쉽게 풀어쓴

# C언어 EXPRESS



천인국 지음

## 제3장 C프로그램 구성요소

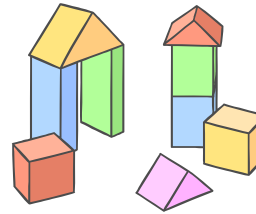




# 이번 장에서 학습할 내용



- \* 주석
- \* 변수
- \* 함수
- \* 문장
- \* 출력 함수 printf()
- \* 입력 함수 scanf()
- \* 산술 연산
- \* 대입 연산





# 일반적인 프로그램의 형태

- 데이터를 받아서(입력 단계), 데이터를 처리한 후에(처리 단계), 결과를 화면에 출력(출력 단계)한다.



데이터 입력



데이터 처리



결과출력



# 덧셈 프로그램 #1

add1.c

```
1  /* 두 개의 숫자의 합을 계산하는 프로그램 */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int x;    // 첫 번째 정수를 저장할 변수
7      int y;    // 두 번째 정수를 저장할 변수
8      int sum;  // 두 정수의 합을 저장하는 변수
9
10     x = 100;
11     y = 200;
12
13     sum = x + y;
14     printf("두수의 합: %d", sum);
15
16     return 0;
17 }
```

주석

전처리기

변수 선언

함수

연산



# 프로그램의 실행 화면





# 주석(comment)

```
/* 두개의 숫자의 합을 계산하는 프로그램 */
```

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

```
int main(void)
```

```
{
```

```
...
```

```
...
```

```
...
```

```
}
```

주석은 코드를  
설명하는 글입니다.





## 3가지 방법의 주석

□ /\* 한 줄로 된 주석 \*/

□ /\* -----

저자: 홍길동

날짜: 2013.3.4

여러 줄로 이루어진 주석

----- \*/

□ // 여기서부터 줄의 끝까지 주석

❖ 주석의 내용: 작성자, 작성목적, 작성일자, 코드 설명 등



# 주석의 예

- 주석

```
/* This program accepts an array of N elements and a key. *  
* Then it searches for the desired element. If the search *  
* is successful, it displays "SUCCESSFUL SEARCH". *  
* Otherwise, a message "UNSUCCESSFUL SEARCH" is displayed. */
```

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





# 들여쓰기

- **들여쓰기(indentation):** 같은 수준에 있는 문장들을 왼쪽 끝에서 몇 자 안으로 들여쓰는 것

```
#include <stdio.h>
int main(void)
{
    int x;
    int y;
    int sum;
    ...
    return 0;
}
```

빈줄을 넣어서 의미별로 구별을 한다.

프로그램의 의도를 주석으로 설명한다.

// 첫 번째 정수를 저장할 변수  
// 두 번째 정수를 저장할 변수  
// 두 정수의 합을 저장하는 변수

같은 내용의 처리이면 들여쓰기를 한다.



# 주석과 들여 쓰기가 없다면..

```
#include <stdio.h>
int main(void) { int x; int y; int sum;
x = 100; y = 200; sum = x + y;
printf("두수의 합: %d", sum); return 0; }
```

실행은 되지만 무슨 처리를 하고 있는 프로그램인지 알기가 힘들고 또한 들여쓰기가 안 되어 있어서 같은 수준에 있는 문장들을 구분하기 힘듭니다.





# 중간 점검

- 주석은 /\* /\* ..... \*/ \*/와 같이 중첩할 수 있을까?
- 주석은 한 줄 이상이 될 수 있는가?
- 주석에는 어떤 내용을 쓰면 좋은가?
- 주석은 프로그램의 동작에 어떤 영향을 끼치는가?





# 전처리기

- `stdio.h`는 표준 입출력에 대한 라이브러리 함수의 정의가 들어 있다.

`#include <stdio.h>`

- 외부 파일을 포함시키라는 의미의 전처리기
- `#`기호로 시작



# 전처리기

```
/* 첫번째 프로그램 */
```

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

```
int main(void)
{
    printf("Hello World!");
    return 0;
}
```

hello.c

```
// stdio.h
```

```
...
```

```
int printf(char *,...);
```

```
...
```

stdio.h



## 중간 점검

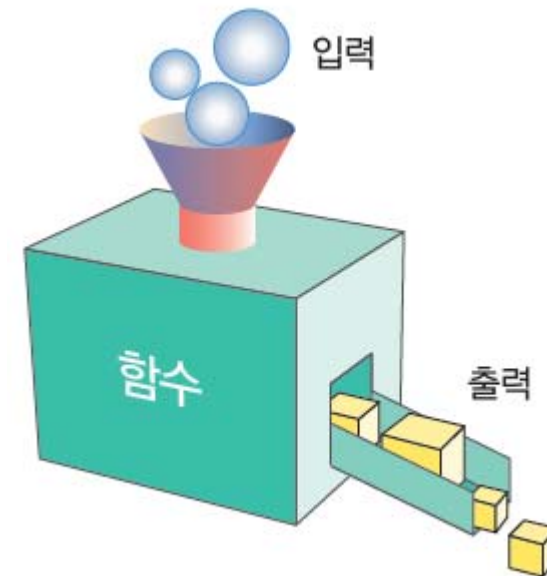
- printf()를 사용하기 위하여 포함시켜야 하는 헤더 파일은 무엇인가?
- 전처리기 #include의 의미는 무엇인가?





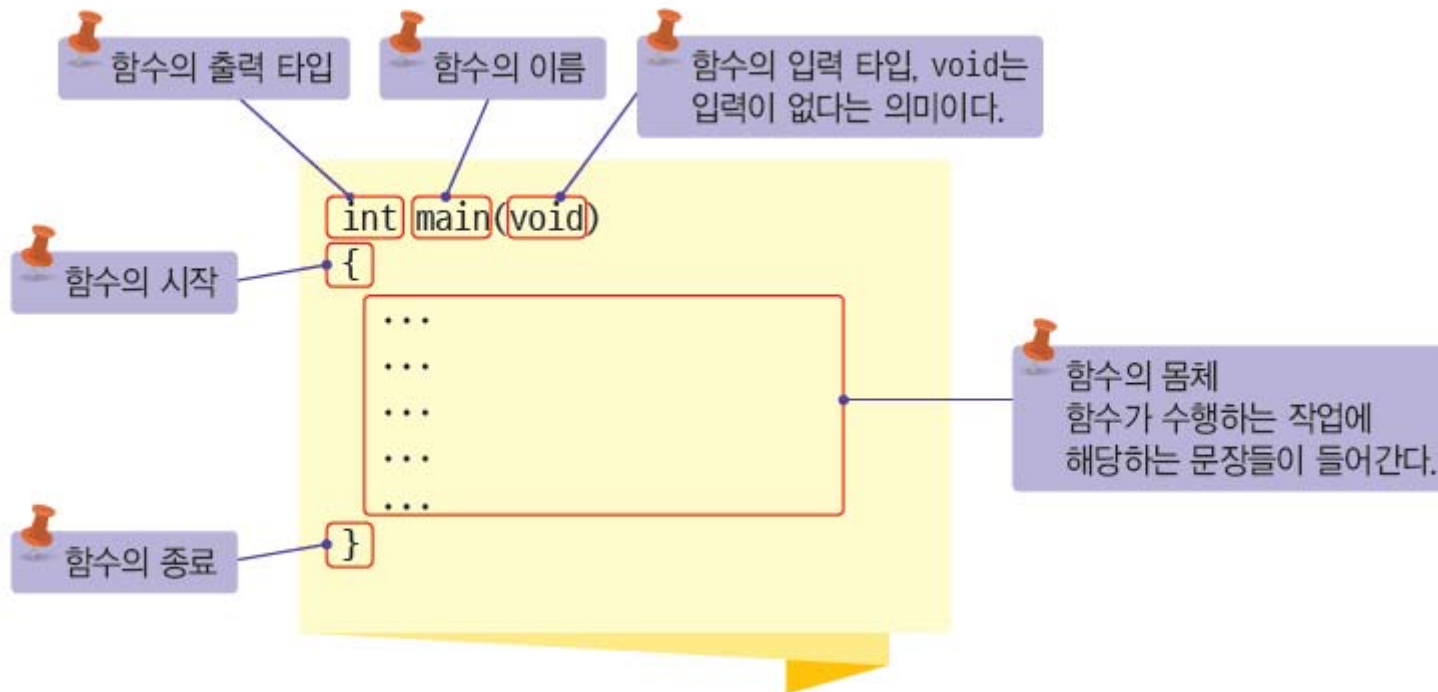
# 함수

- 함수(function): 특정 기능을 수행하는 처리 단계 (프로그램 문장)들을 괄호로 묶어서 이름을 붙인 것
- 함수는 프로그램을 구성하는 기본적인 단위(부품)





# 함수의 구조



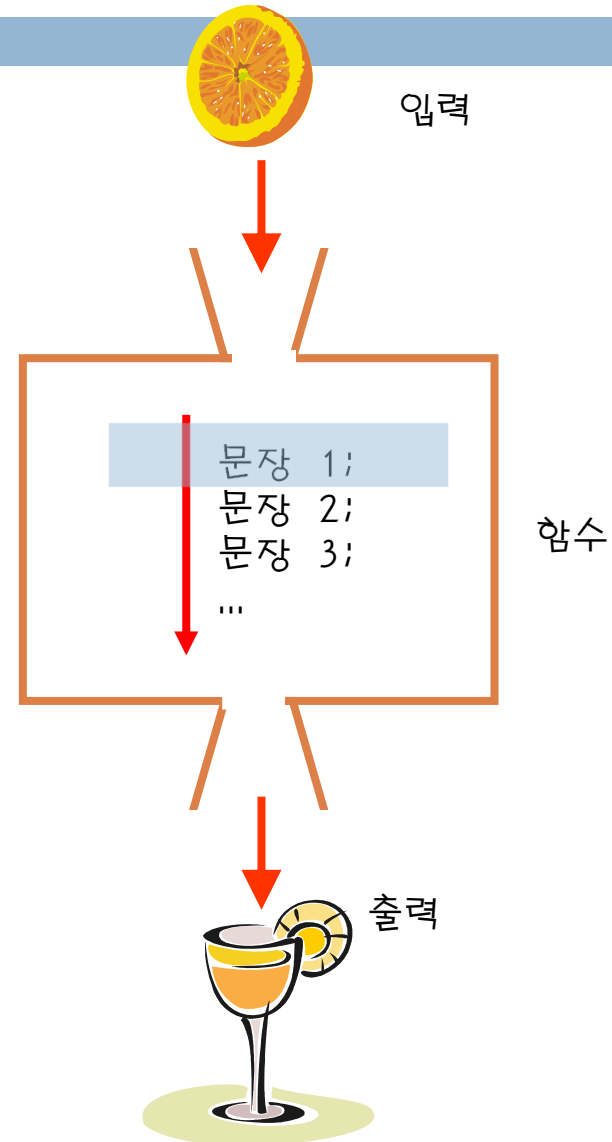




# 함수 안에 들어 있는 것

Q) 그렇다면 함수 안에 들어 있는 것은 무엇인가?

A) 함수 안에는 함수가 처리하는 처리 단계(문장, **statement**)들이 중괄호 안에 나열

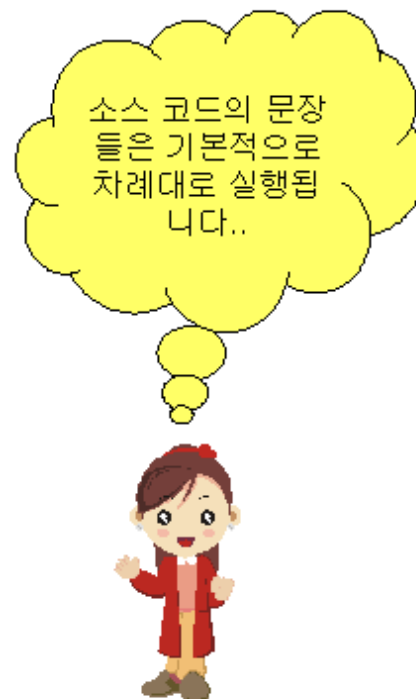
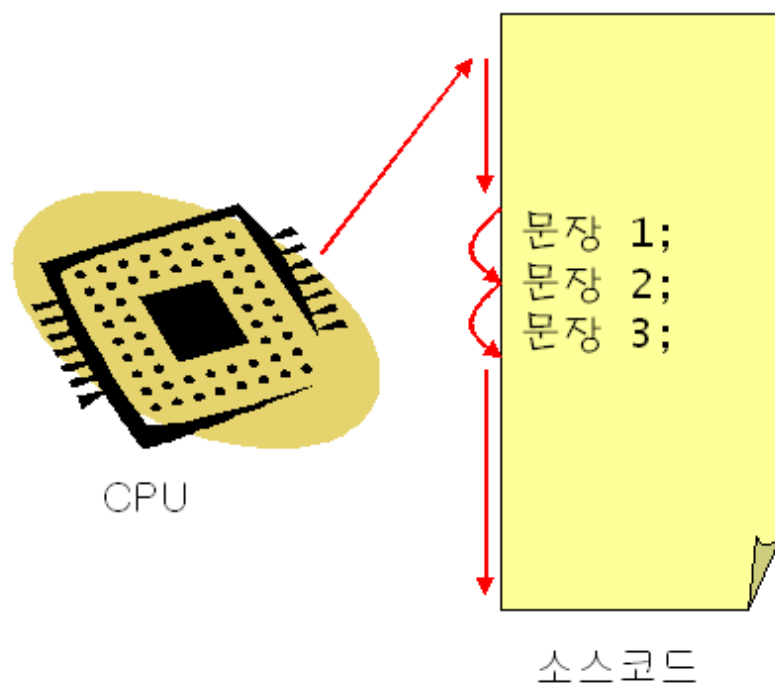




# 문장

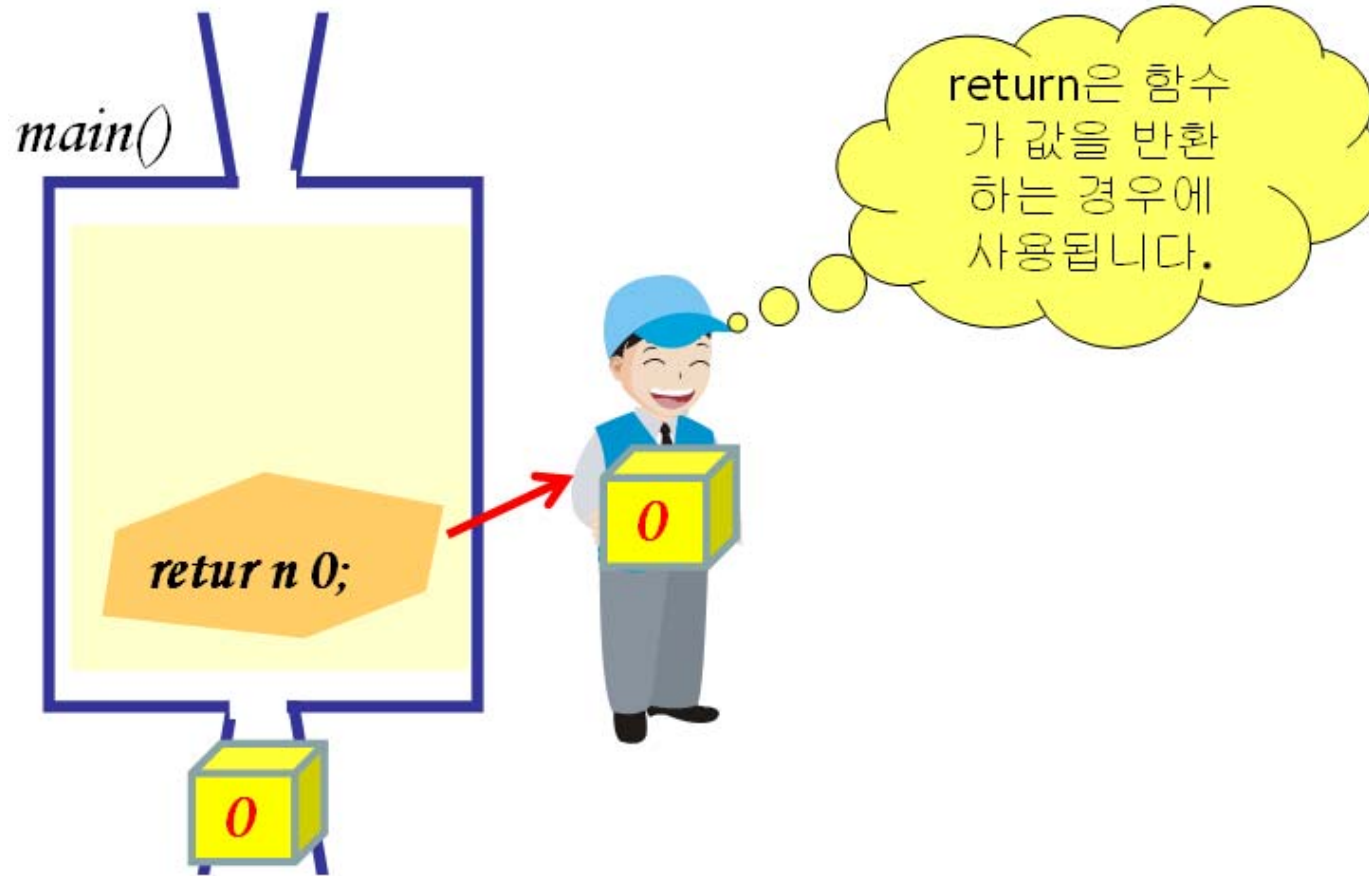
- 함수는 여러 개의 문장으로 이루어진다.
- 문장들은 순차적으로 실행된다.
- 문장은 ;(세미콜론)으로 끝나야 한다.

I



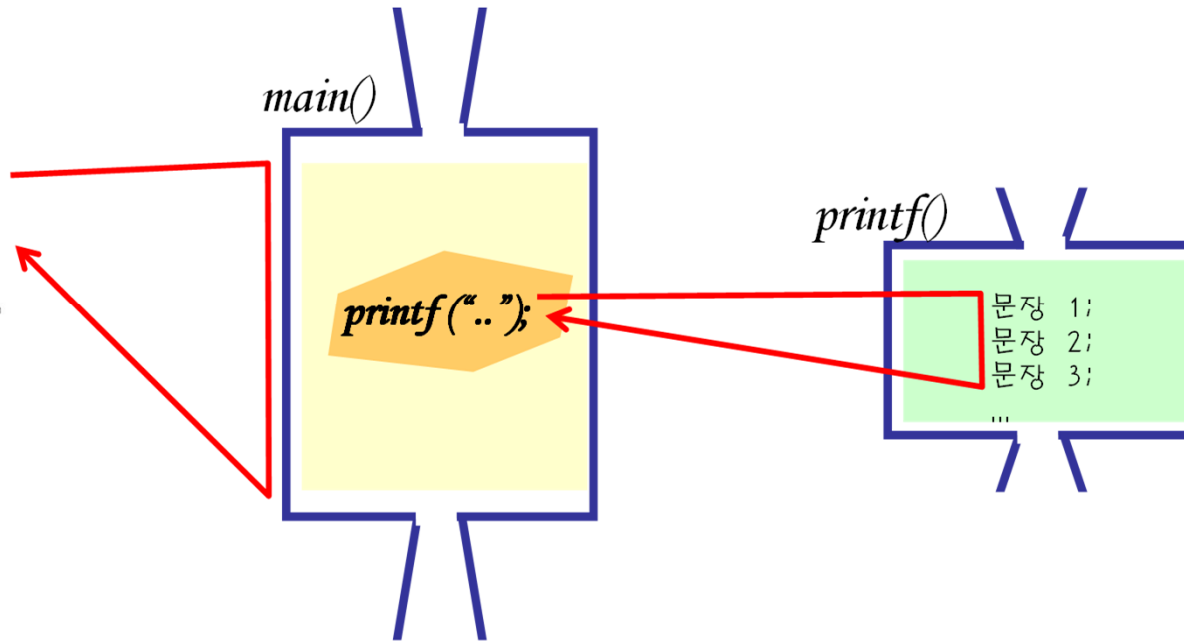


# return 문장





# main()은 누가 호출할까?



함수를 실행하  
려면 함수를  
호출하면 됩니  
다.





# 중간 점검

- 모든 C 프로그램에 반드시 있어야 되는 함수는 무엇인가?
- 함수의 시작과 끝을 나타내는 기호는 무엇인가?
- 모든 문장은 어떤 기호로 끝나는가?





# 변수

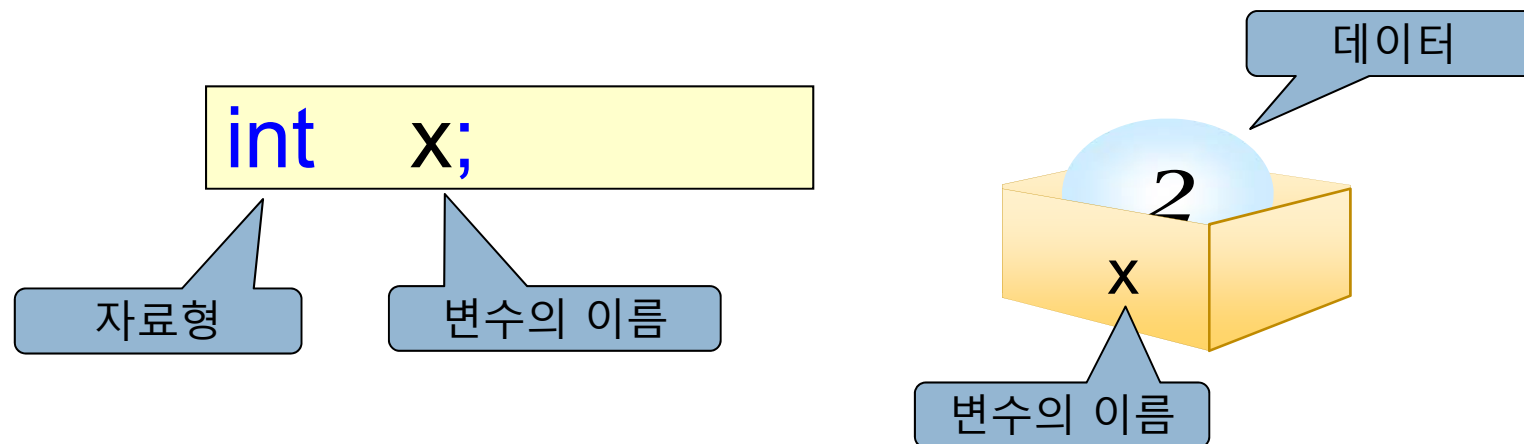
- 프로그램이 사용하는 데이터를 일시적으로 저장할 목적으로 사용하는 메모리 공간





# 변수의 종류

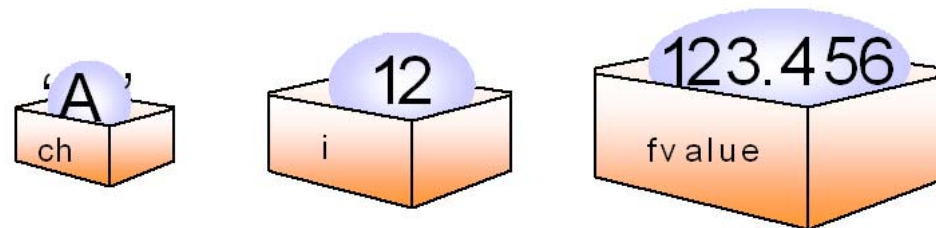
- 변수는 데이터를 담는 상자로 생각할 수 있다.





# 변수의 종류

- 변수에는 데이터의 종류에 따라 여러 가지 타입이 존재한다.

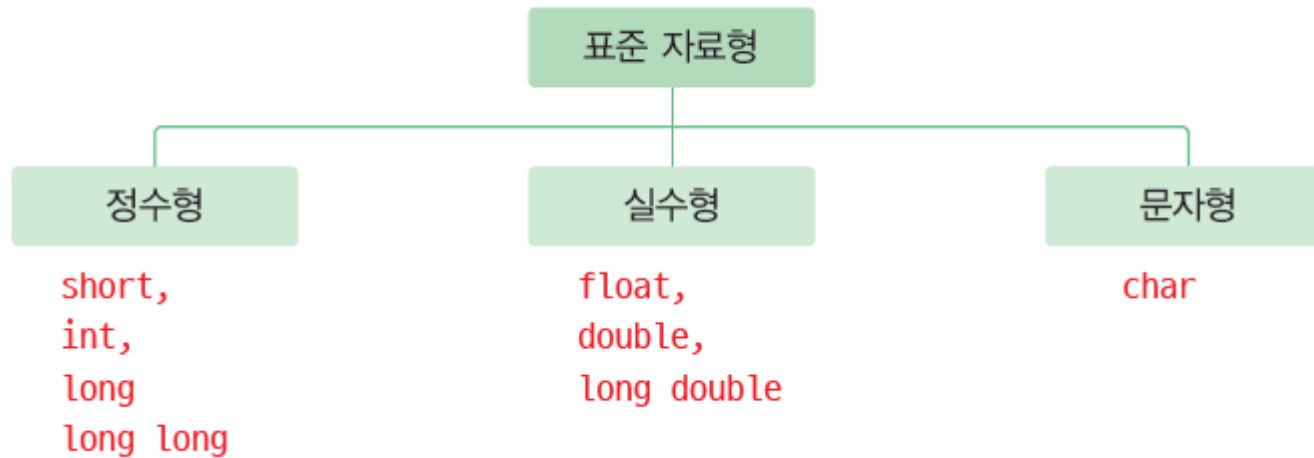






# 자료형

- 자료형(data type): 변수가 저장할 데이터가 정수인지 실수인지, 아니면 또 다른 어떤 데이터인지를 지정하는 것

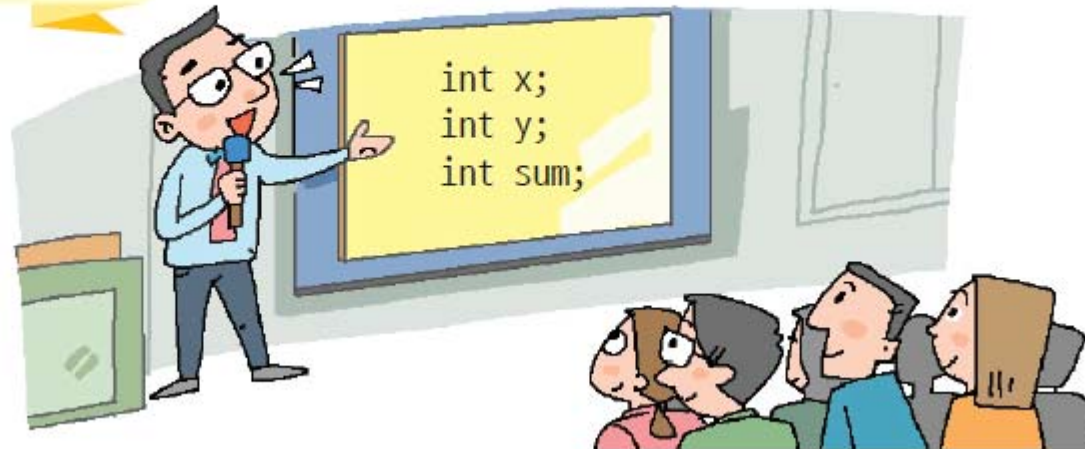




# 변수 선언

- 변수 선언: 컴파일러에게 어떤 타입의 변수가 사용되는지를 미리 알리는 것

지금부터 이 프로그램에서 사용될 변수들을 소개하겠습니다.

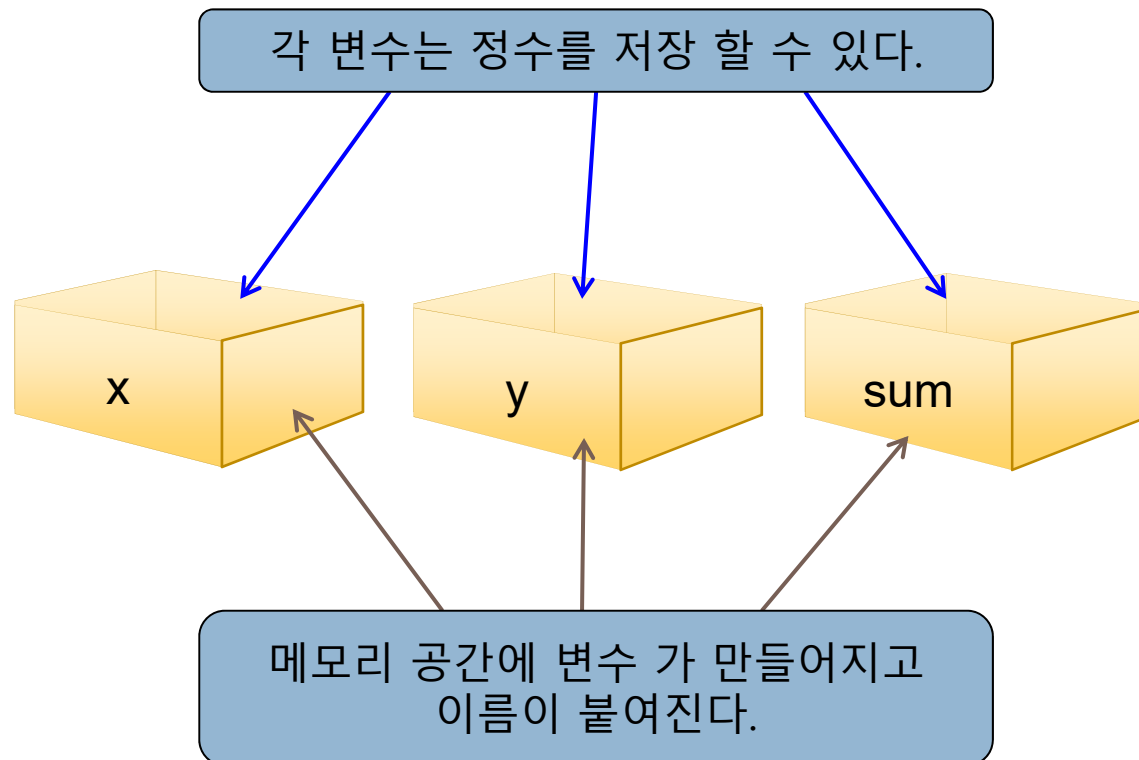


컴파일러



# 변수 선언

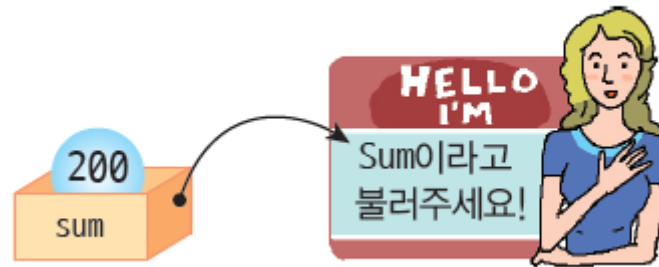
```
int x;    // 첫번째 정수를 저장하는 변수  
int y;    // 두번째 정수를 저장하는 변수  
int sum;  // 두 정수의 합을 저장하는 변수
```





# 변수의 이름

- 식별자(identifier): 변수나 함수의 이름





# 변수의 이름

## □ 식별자 만드는 규칙

- ▣ 식별자는 영어의 대소문자, 숫자, 밑줄 문자 \_로 이루어진다.
- ▣ 식별자는 숫자로 시작할 수 없고 첫 글자는 영문자 또는 밑줄 문자 \_이어야 한다.
- ▣ 식별자의 중간에 공백이 들어가면 안 된다.
- ▣ 대문자와 소문자를 구별한다. (sum, SUM은 다른 식별자)
- ▣ C 언어의 키워드와 똑같은 이름은 허용되지 않는다.



# 키워드

- 키워드(keyword): C언어에서 고유한 의미를 가지고 있는 특별한 단어이며 예약어(reserved words) 라고도 한다.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while



# 변수의 이름



```
sum  
_count  
number_of_pictures  
King3
```

// 영문 알파벳 문자로 시작  
// 밑줄 기호로 시작할 수 있다.  
// 중간에 밑줄 문자를 넣을 수 있다.  
// 맨 처음이 아니라면 숫자도 넣을 수 있다.



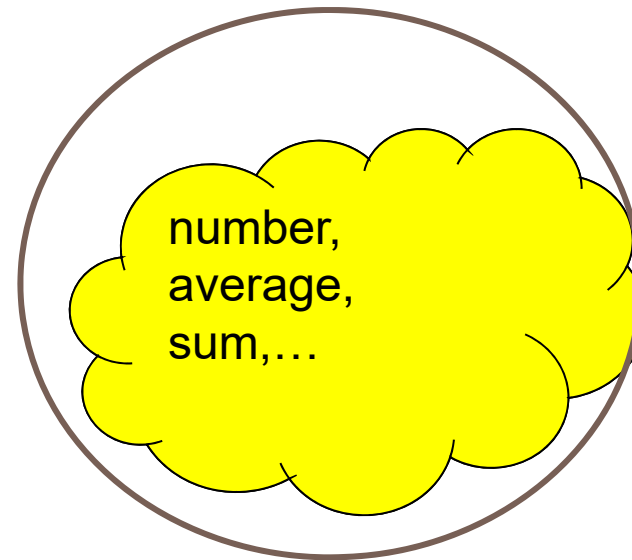
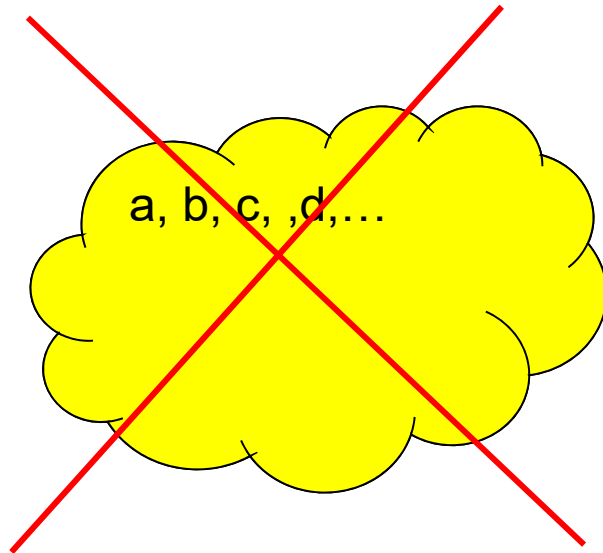
```
2nd_base (X)  
money# (X)  
double (X)
```

// 숫자로 시작할 수 없다.  
// #과 같은 기호는 사용할 수 없다.  
// double은 C언어의 키워드이다.



# 좋은 변수 이름

- 변수의 역할을 가장 잘 설명하는 이름
  - ▣ 밑줄 방식: `bank_account`
  - ▣ 단어의 첫번째 글자를 대문자: `BankAccount`



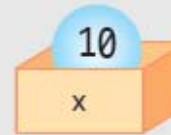




# 변수의 초기화

```
int x = 10;  
int y = 20;  
int sum = 0;
```

초기값





# 중간 점검

- 변수 이름을 만들 때 지켜야 하는 규칙은 무엇인가?
- 변수 이름의 첫 번째 글자로 허용되는 것은 무엇인가?
- C에서 고유한 의미를 가지고 있는 단어들을 무엇이라고 하는가?

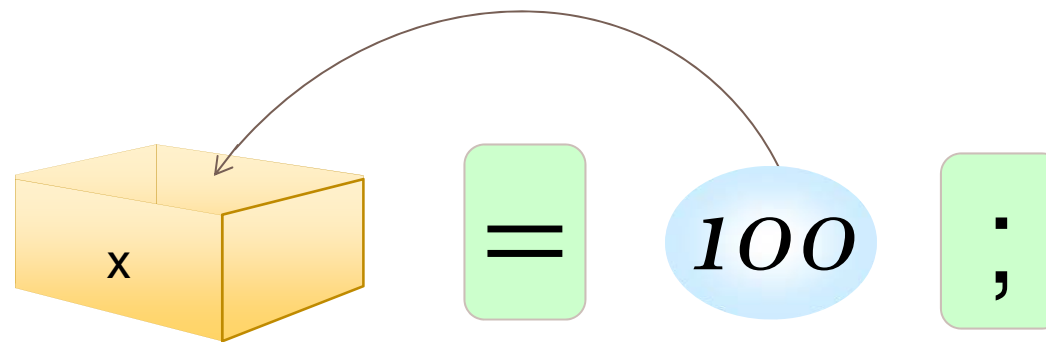




# 대입 연산

```
x = 100;
```

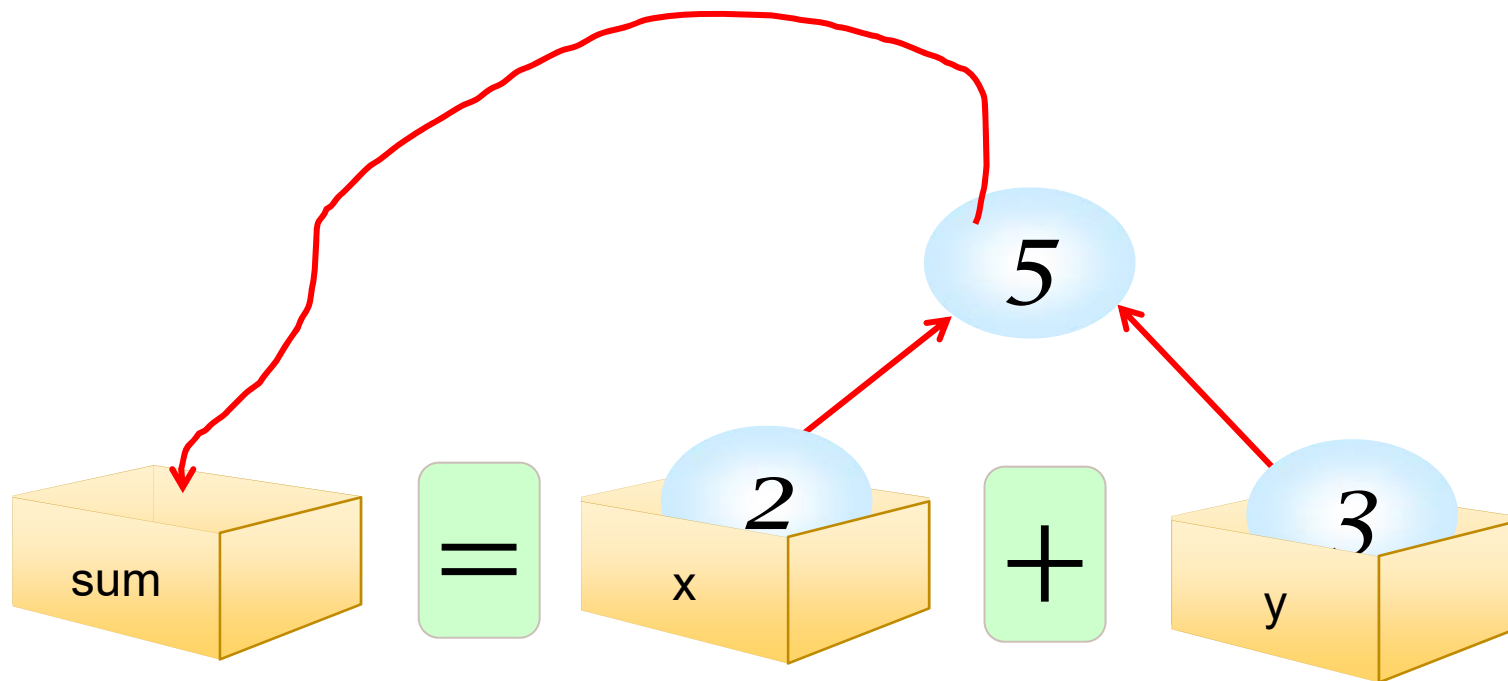
- 대입 연산(assignment operation): 변수에 값을 저장하는 연산
- 대입 연산 = 배정 연산 = 할당 연산





# 산술 연산

sum = x + y;





# 산술 연산

연산	연산자	C 수식	수학에서의 기호
덧셈	+	$x + y$	$x + y$
뺄셈	-	$x - y$	$x - y$
곱셈	*	$x * y$	$xy$
나눗셈	/	$x / y$	$x/y$ 또는 $\frac{x}{y}$ 또는 $x \div y$
나머지	%	$x \% y$	$x \bmod y$



# 정리

프로그램

```
#include <stdio.h>
int main(void)
{
    int x;
    int y;
    int sum;

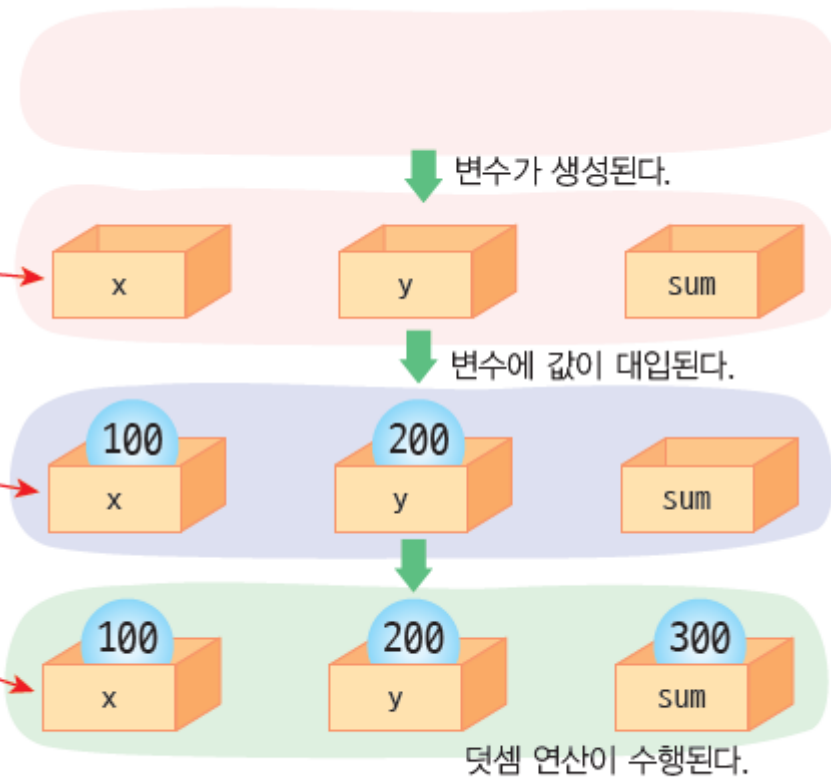
    x = 100;
    y = 200;

    sum = x + y;

    printf("두수의 합: %d", sum);

    return 0;
}
```

컴퓨터 내부





## 중간 점검

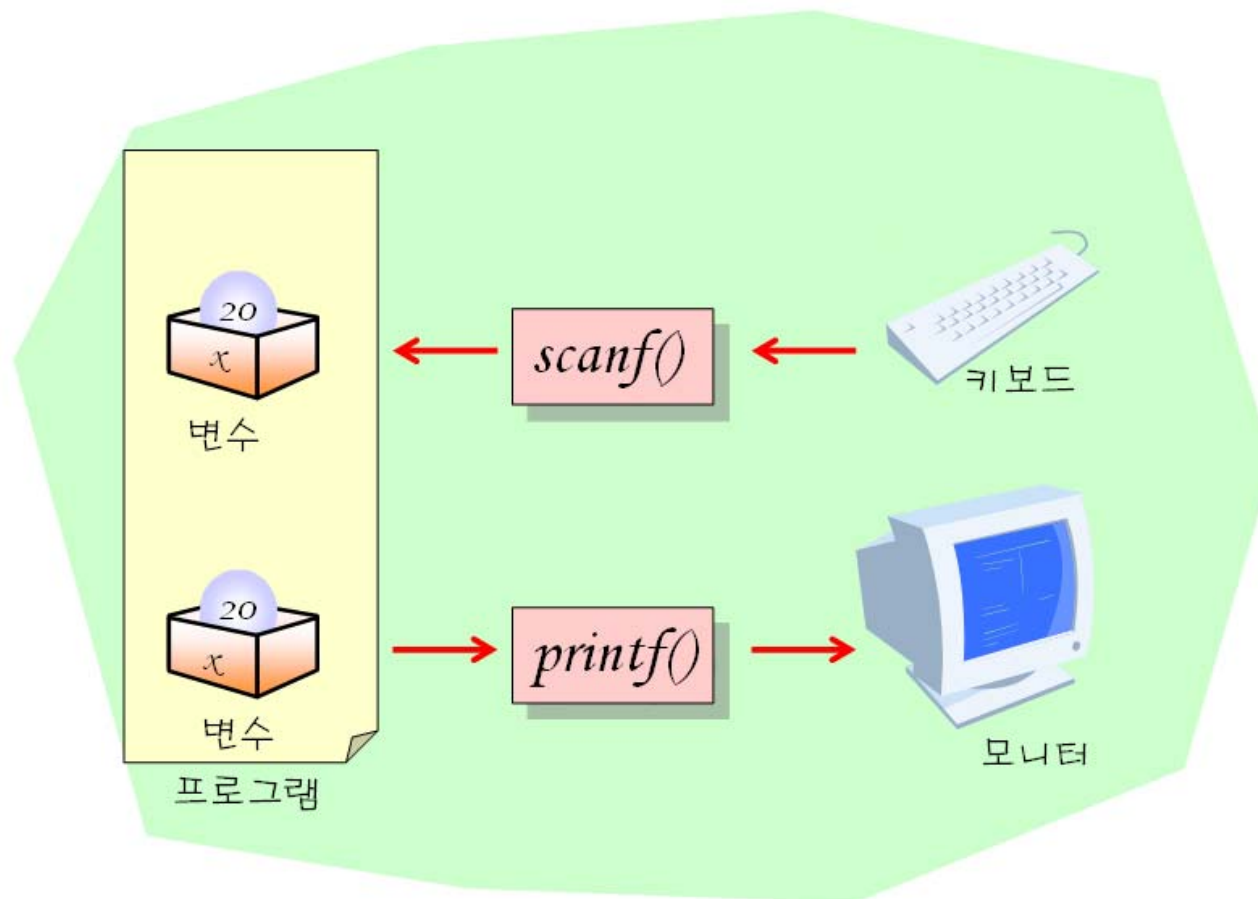
- 함수의 중간에서 변수를 선언할 수 있는가?
- int형 변수 x와 y를 한 줄에 선언하고 1과 0으로 각각 초기화하라.
- 변수a와 변수 b의 곱을 변수 product에 저장하는 문장을 작성하여 보자.
- 변수a를 변수 b로 나눈 값을 변수 quotient에 저장하는 문장을 작성하여 보자.





# printf()

- printf(): 모니터에 출력을 하기 위한 표준 출력 라이브러리 함수



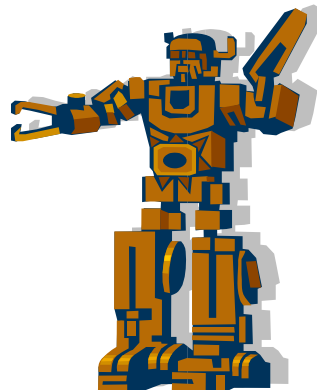
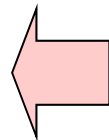




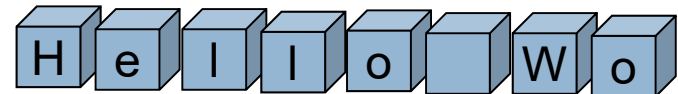
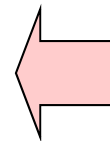
# 문자열 출력

```
printf("Hello World!\n");
```

- 문자열(string): "Hello World!\n"와 같이 문자들을 여러 개 나열한 것



printf()





# 변수값 출력

형식 제어 문자열

```
printf("두수의 합: %d", sum);
```

두수의 합: 30

형식 지정자의  
개수와 변수의  
개수와  
순서는 같아  
야 합니다.





# 형식 지정자

- 형식 지정자: printf()에서 값을 출력하는 형식을 지정한다.

형식 지정자	의미	예	실행 결과
%d	10진 정수로 출력	<code>printf("%d \n", 10);</code>	10
%f	실수로 출력	<code>printf("%f \n", 3.14);</code>	3.14
%c	문자로 출력	<code>printf("%c \n", 'a');</code>	a
%s	문자열로 출력	<code>printf("%s \n", "Hello");</code>	Hello



# 여러 개의 변수값 출력

형식 제어 문자열

```
printf("%d %f", number, grade);
```

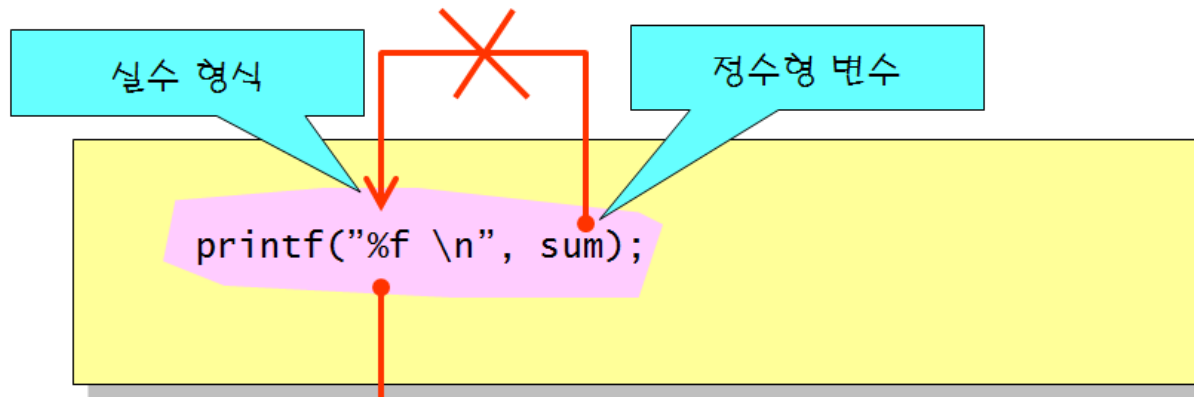
23 3.99

형식 지정자의 개수와 변수의 개수와 순서는 같아야 합니다.





# 주의!



형식 지정자  
와 변수의 타  
입은 같아야  
합니다.





## 중간 점검

- printf()에서 변수의 값을 실수 형태로 출력할 때 사용하는 형식 지정자는 무엇인가?
- printf()를 사용하여 정수형 변수 k의 값을 출력하는 문장을 작성하여 보자.

