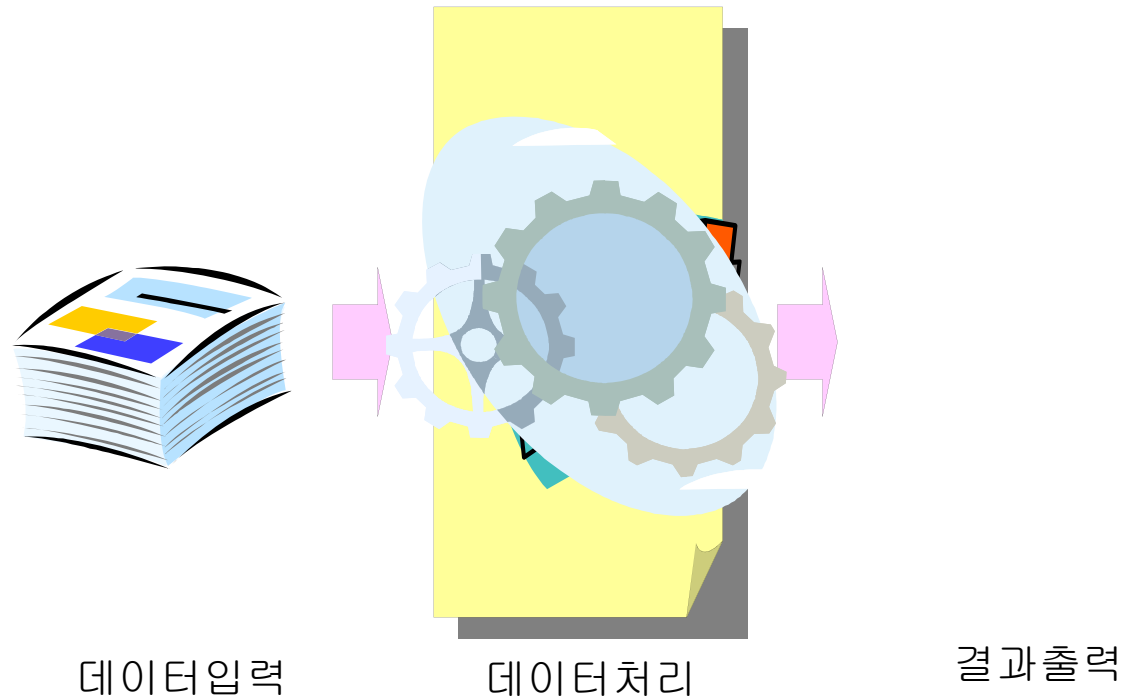


일반적인 프로그램의 기본 구성 형태

- ◆ 데이터를 받아서(입력단계), 데이터를 처리한 후에(처리단계), 결과를 화면에 출력(출력단계)한다.



덧셈 프로그램 #1

```
/* 두개의 숫자의 합을 계산하는 프로그램 */
#include <stdio.h>

int main(void)
{
    int x;           // 첫번째 정수를 저장할 변수
    int y;           // 두번째 정수를 저장할 변수
    int sum;         // 두 정수의 합을 저장하는 변수

    x = 100;
    y = 200;

    sum = x + y;
    printf("두수의합: %d", sum);

    return 0;
}
```

주석

전처리기 지시어

함수

변수선언

문장

연산

주석(comment)

```
/* 두개의 숫자의 합을 계산하는 프로그램 */  
#include <stdio.h>  
  
int main(void)  
{  
  ...  
  ...  
  ...  
}
```

주석은 코드를 설명하는 글입니다.



주석

3가지 방법의 주석

/* 한줄로 된 주석 */

**/* 여러
줄로
된 주석
*/**

// 여기서부터 줄의 끝까지 주석



프로그래밍언어 과목의 표준 주석 표기

◆ Standard comment at each source code file produced in this course

```
/**
 * File Name: "???.c" or "xxx.h", or "yyy.cpp"
 * Description:
 *   - This program is . . . . .
 *
 * Programmed by Gil-Dong Hong (November 15, 2009),
 * Last updated: Version 2.0, September 1, 2012 (by Young-Chul Kim).
 *
 * =====
 * Version Control (Explain updates in detail)
 * =====
 * Name      YYYY/MM/DD  Version  Remarks
 * Gil-Dong Hong  2009/11/15  v1.0    1:1 Chatting on UDP socket, 2 threads
 * John Doe      2010/05/01  v1.1    User interface has been updated with GUI
 * Young-Chul Kim 2012/09/01  v2.0    Major change in the program structure
 * =====
 */
```



들여쓰기 (indentation)

- ◆ 들여쓰기(indentation): 같은 수준에 있는 문장들을 왼쪽 끝에서 몇 자 안으로 들여 쓰는 것

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

빈줄을 넣어서 의미별로 구별을 한다.

같은 내용의 처리이면 들여쓰기를 한다.

// 첫번째 정수를 저장할 변수
// 두번째 정수를 저장할 변수
// 두 정수의 합을 저장하는 변수

프로그램의 의도를 주석으로 설명한다.



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 안되어 있어서 같은 수준에 있는 문장들을 구분하기 힘듭니다.



전처리기 (preprocessor)

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

- ◆ #기호로 시작
- ◆ 헤더 파일 **stdio.h**를 소스 코드 안에 포함
- ◆ **stdio.h**는 표준 입출력에 대한 라이브러리 함수의 정의가 들어 있다.



전처리기

```
/* 첫번째 프로그램 */  
  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!");  
    return 0;  
}
```

hello.c

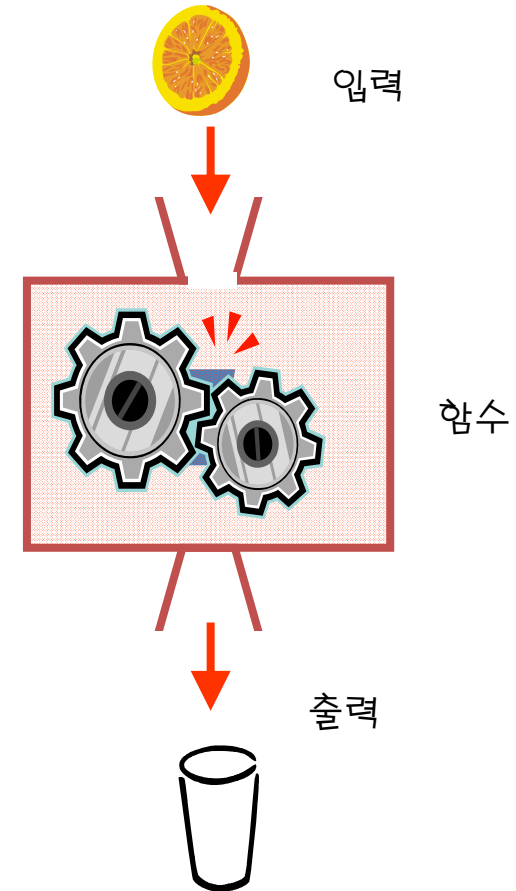
```
// stdio.h  
...  
int printf(char *,...);  
...
```

stdio.h

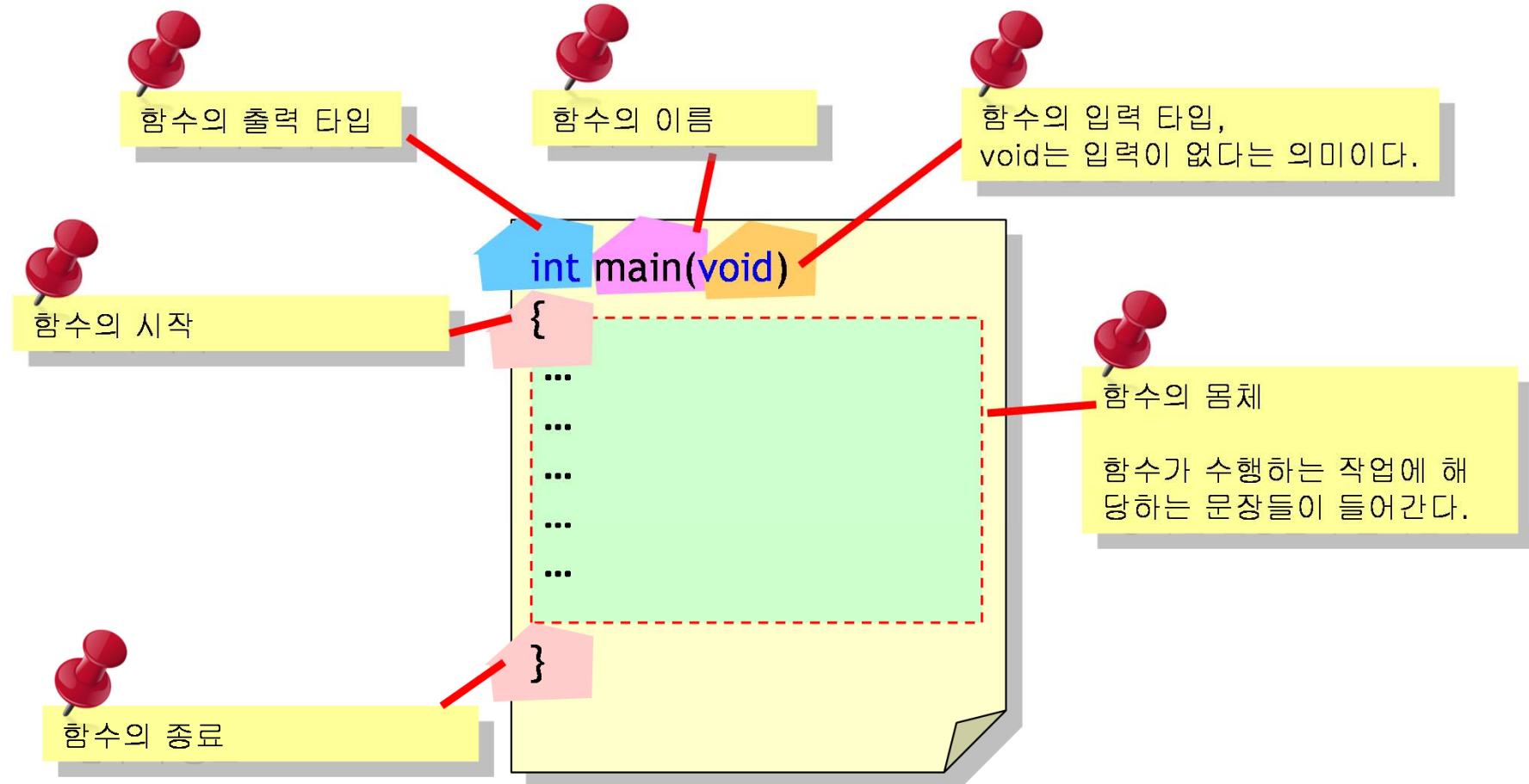


함수 (function)

- ◆ **함수(function):** 특정 기능을 수행하는 처리 단계들을 종괄호로 묶어서 이름을 붙인 것
- ◆ 함수는 프로그램을 구성하는 기본적인 단위(부품)
- ◆ **main()** 도 함수 중 하나



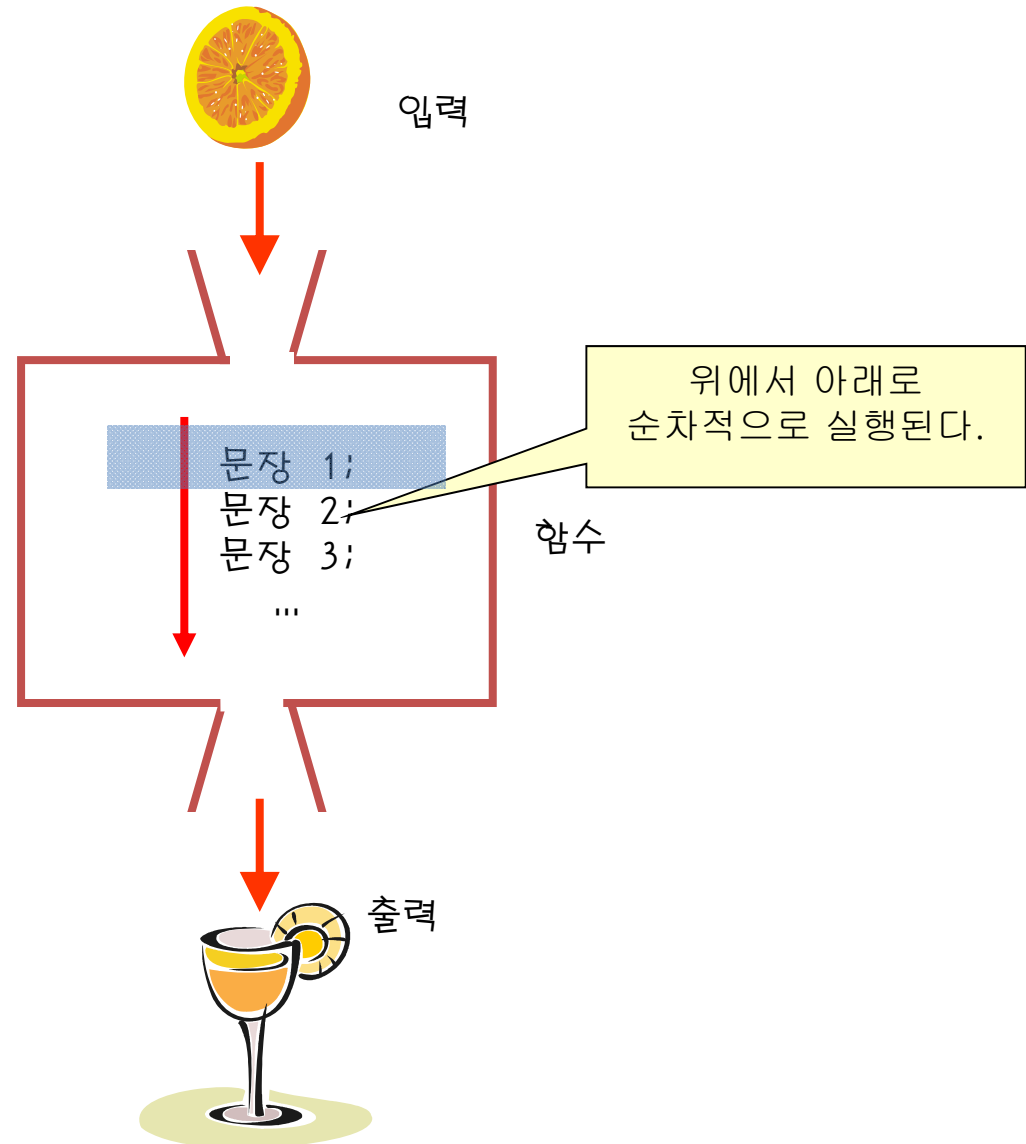
함수의 구조



함수안에 들어 있는 것

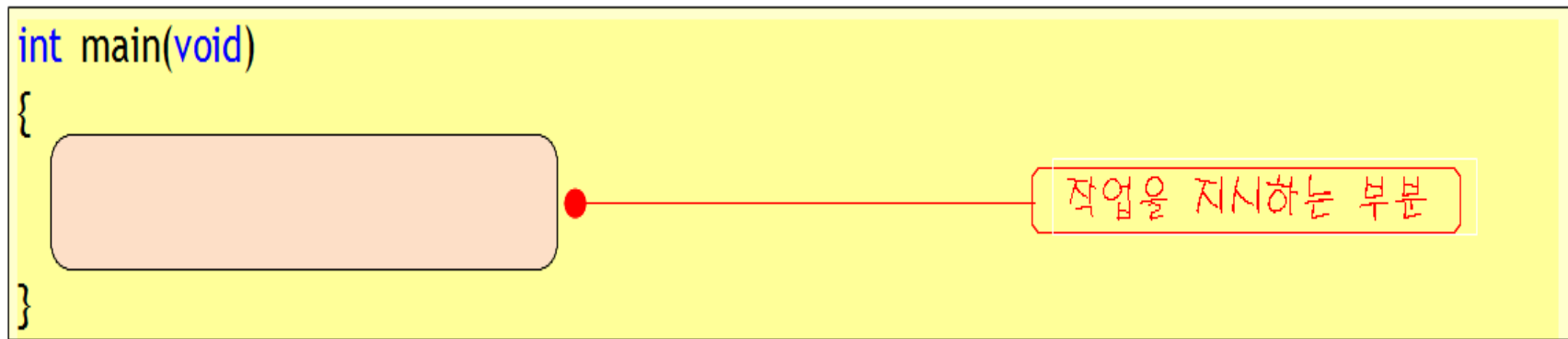
Q) 그렇다면 함수 안에 들어 있는 것은 무엇인가?

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



함수

- ◆ 작업을 수행하는 문장 (statement)들은 함수 안에 들어 가야 함



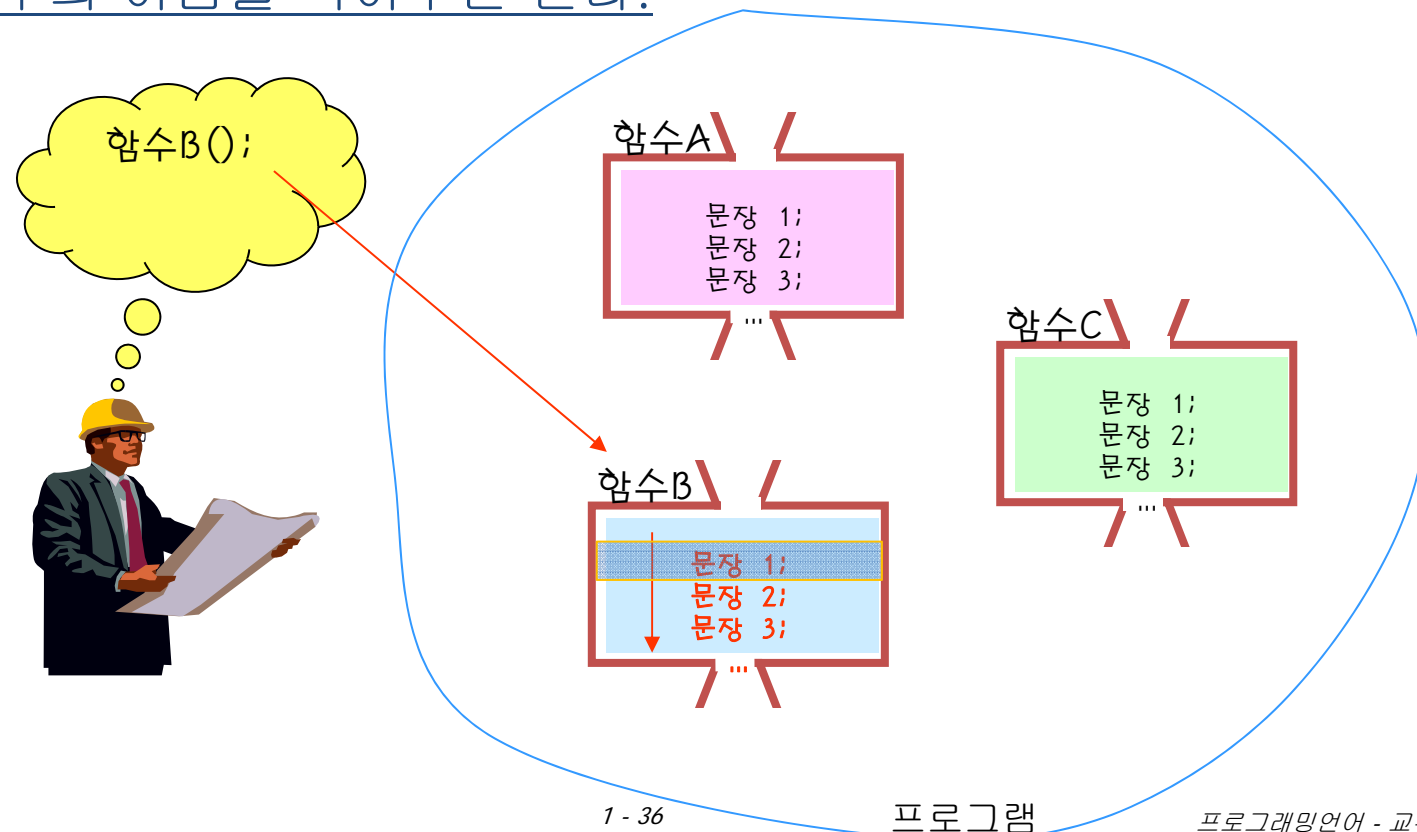
함수 호출 (function call)

Q) 함수 안에 있는 문장들은 언제 실행되는가?

A) 함수가 호출되면 실행된다.

Q) 함수 호출은 어떻게 하는가?

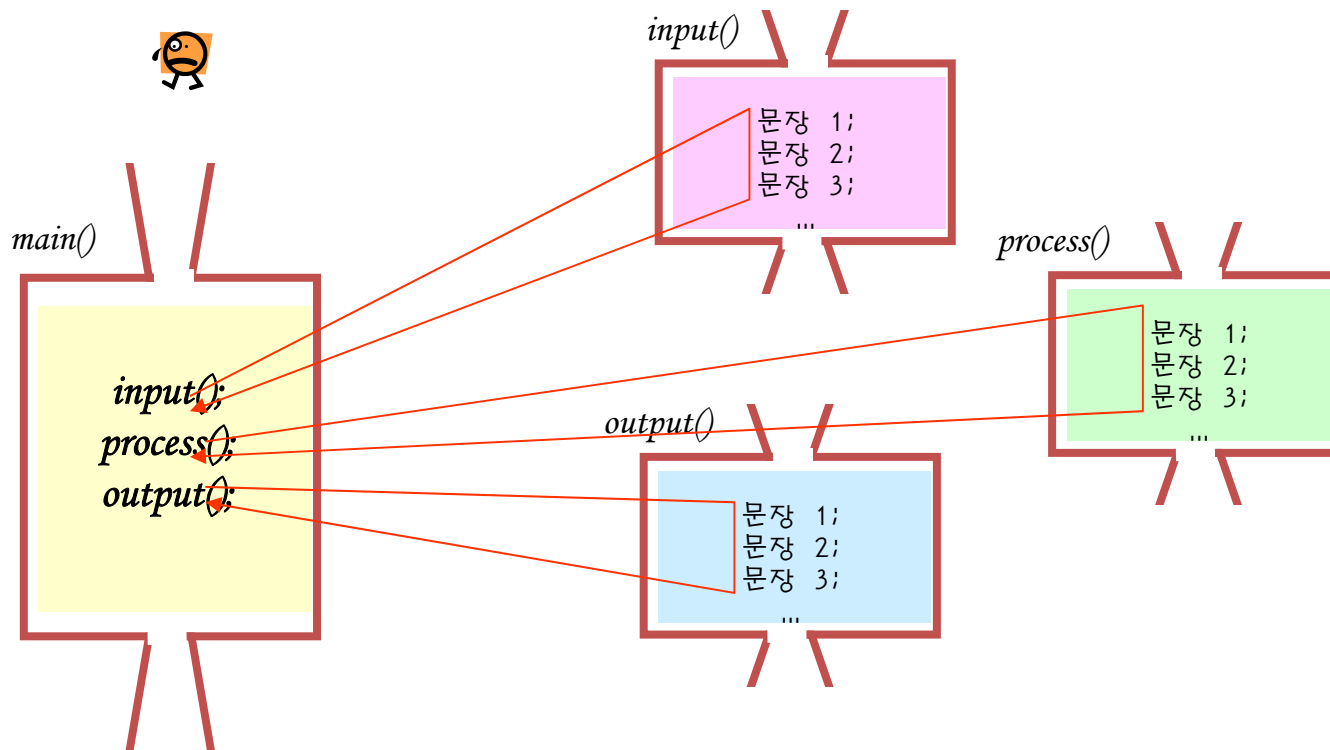
A) 함수의 이름을 적어주면 된다.



함수 호출

Q) C 프로그램의 많은 함수 중에서 가장 먼저 실행되는 것은?

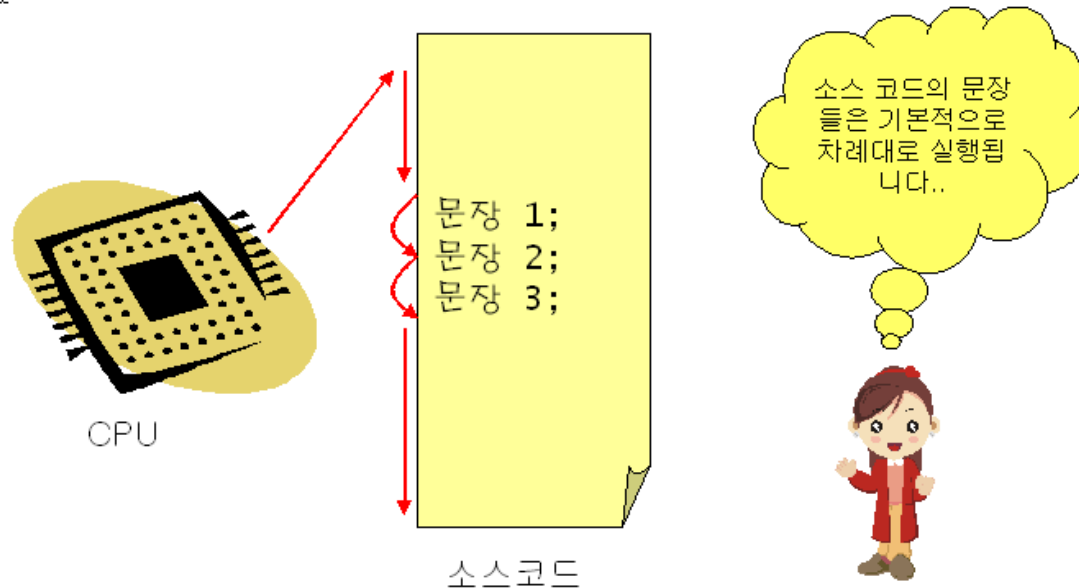
A) `main()` 함수이다. 다른 함수들은 `main()`으로부터 직간접적으로 호출된다.



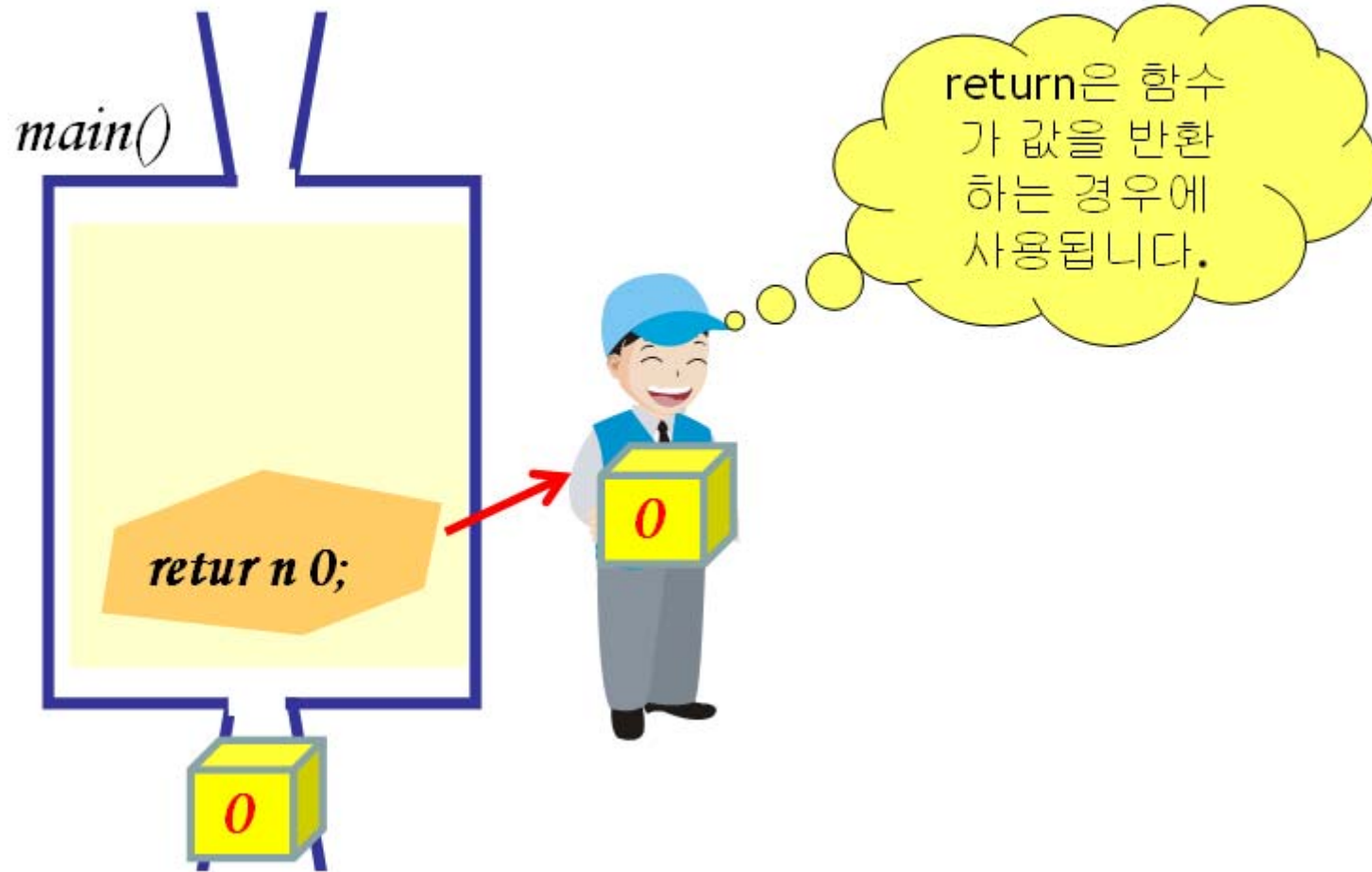
문장 (statement)

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

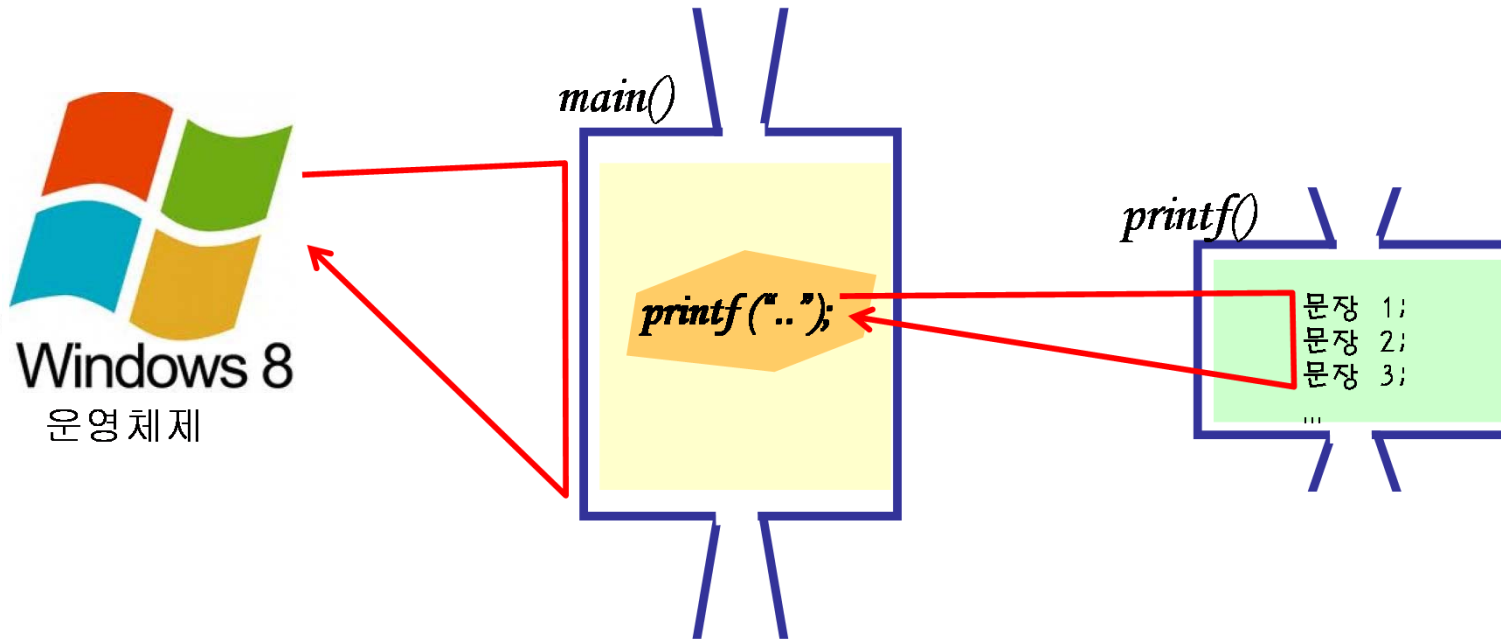
I



return 문장



main()은 누가 호출할까?



함수를 실행하려면 함수를 **호출**하면 됩니다.

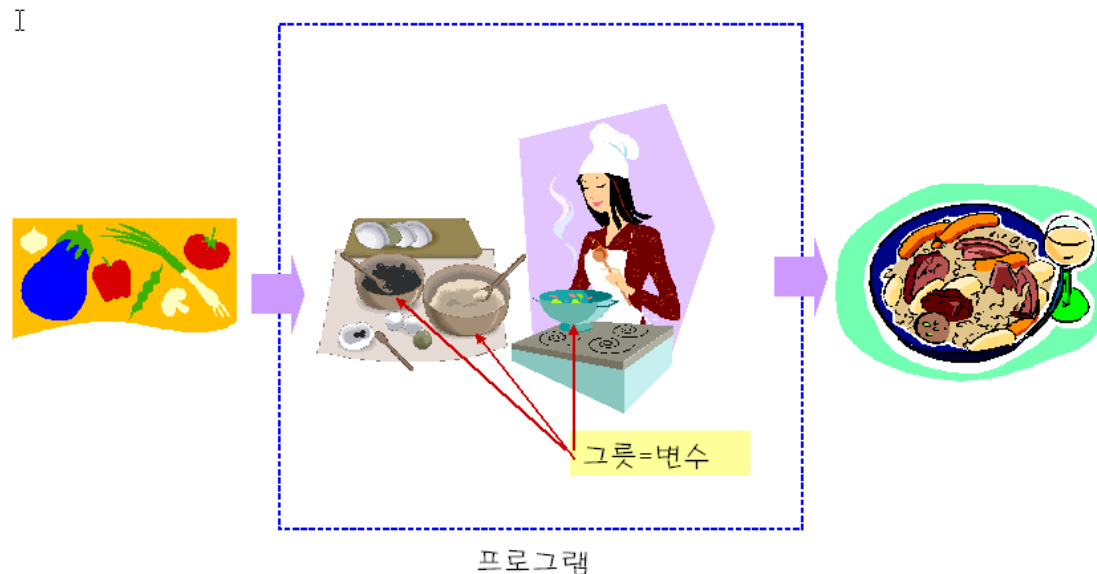


변수 (variable)

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

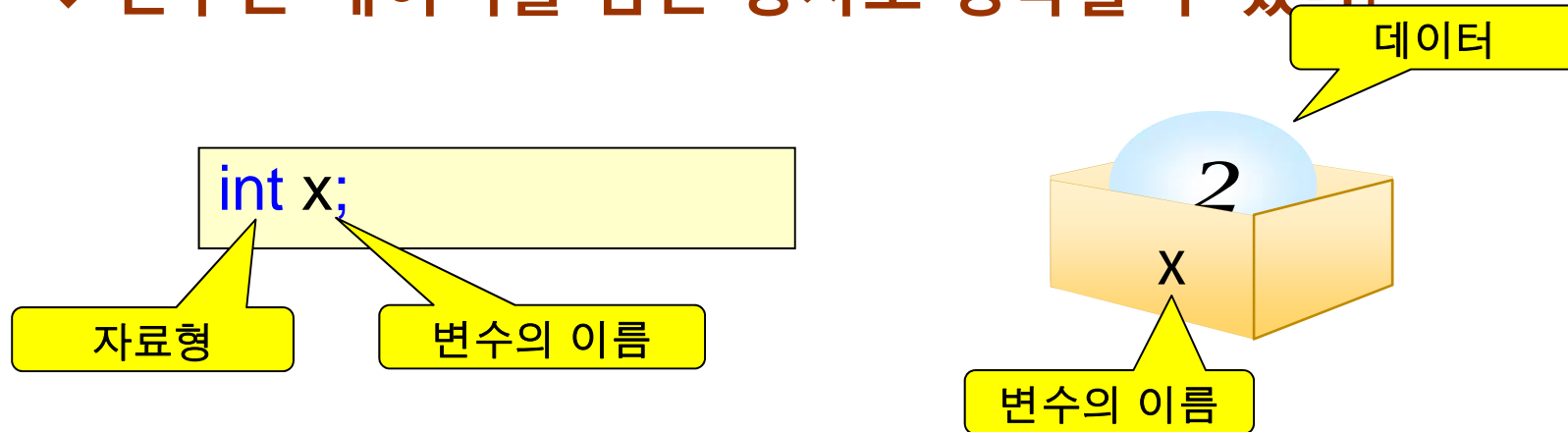
Q) 변수란 무엇인가?

- A) 프로그램이 사용하는 데이터를 일시적으로 저장할 목적으로 사용하는 메모리 공간

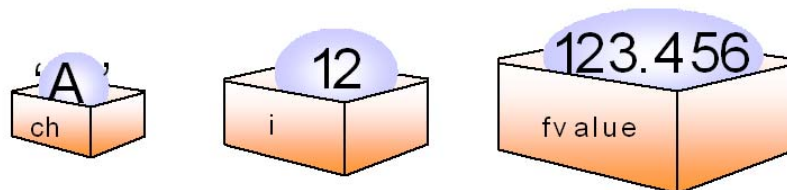


변수의 종류

◆ 변수는 데이터를 담는 상자로 생각할 수 있다.



- 변수에는 데이터의 종류에 따라 여러 가지 타입이 존재한다:
 - char (문자)
 - int (정수)
 - double (실수, 부동소숫점, double precision floating point)



변수의 이름

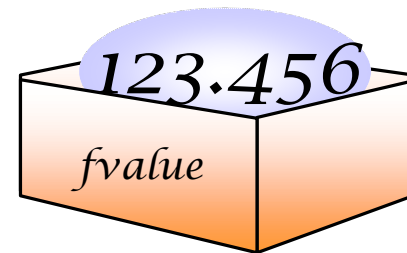
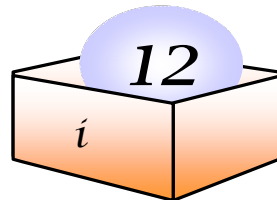
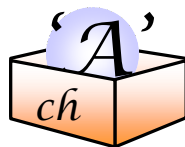
◆ 식별자(identifier): 변수나 함수의 이름

◆ 식별자를 만드는 규칙

- 식별자는 영어의 대소문자, 숫자, 밑줄 문자 _로 이루어진다.
- 식별자는 숫자로 시작할 수 없다.
- 대문자와 소문자를 구별하며 C 언어의 키워드와 똑같은 이름은 허용되지 않는다.

◆ 식별자의 예:

- s, s1, student_number: 올바른 식별자
- \$s, 2nd_student, int: 잘못된 식별자



변수 선언

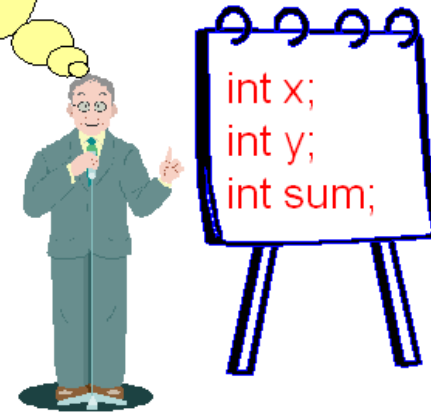
- ◆ 변수 선언: 컴파일러에게 어떤 타입의 변수가 사용되는지를 미리 알리는 것

```
int x;
```

자료형

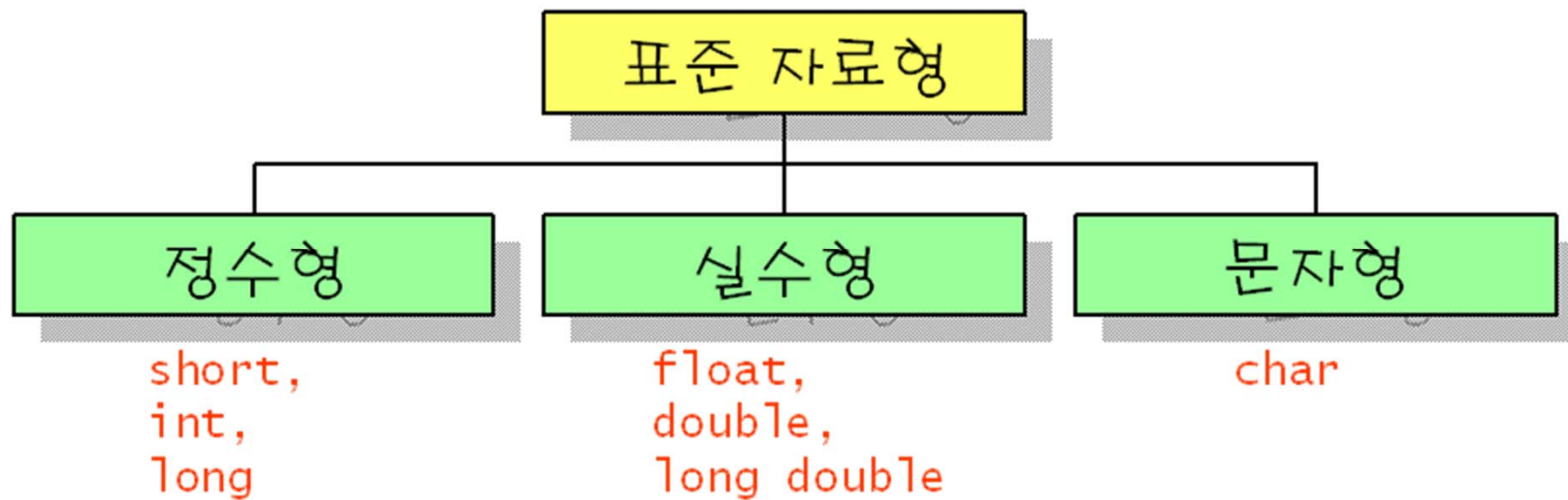
변수의 이름

지금부터 이 프로그램에서 사용될 변수들을 소개하겠습니다.



자료형 (data type)

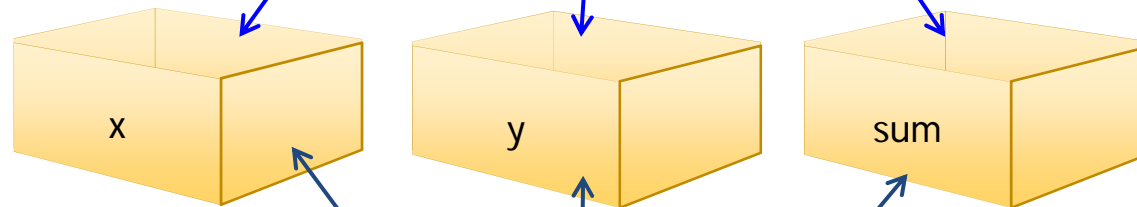
- ◆ 자료형(data type): 변수가 저장할 데이터가 정수인지 실수인지, 아니면 또 다른 어떤 데이터인지를 지정하는 것



변수 선언

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

각 변수는 정수를 저장 할 수 있다.



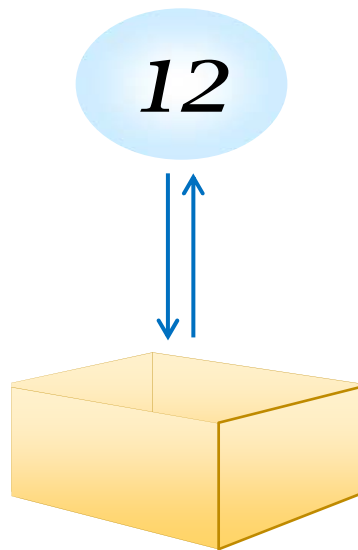
메모리 공간에 변수 가 만들어지고
이름이 붙여진다.

상수 (Constant)

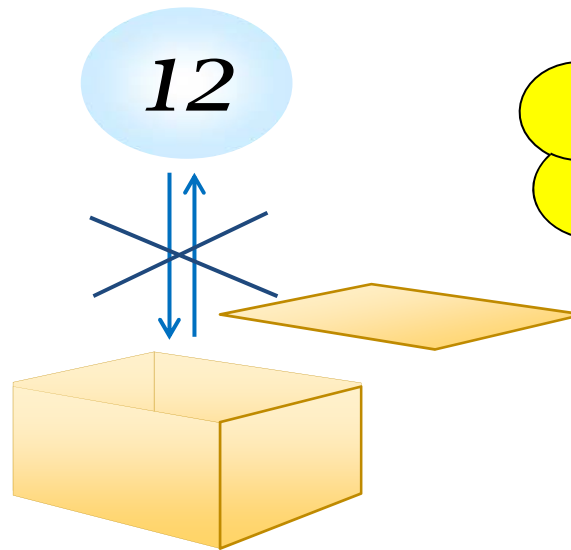
```
x = 100;  
y = 200;
```

상수

◆ 상수(constant): 그 값이 프로그램이 실행하는 동안 변하지 않는 수



변수



상수

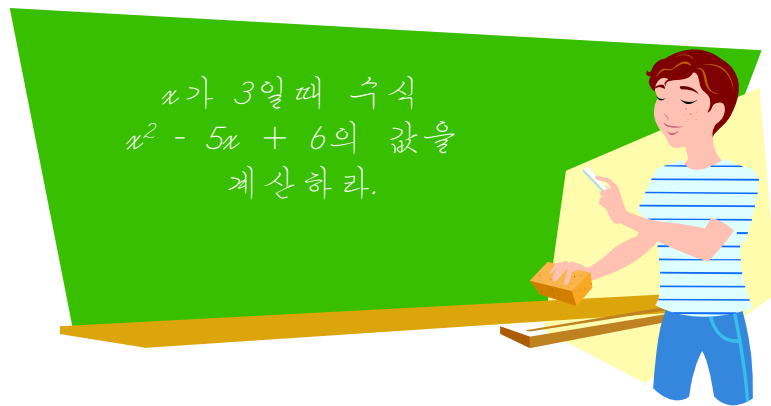
변수는 실행도중에 값을 변경할 수 있으나 상수는 한번 값이 정해지면 변경이 불가능합니다.



수식 (expression)

```
sum = x + y;
```

- ◆ 수식(expression): 피연산자 (operand) 와 연산자 (operator) 로 구성된 식
- ◆ 수식은 결과값을 가진다.



```
int x, y;  
  
x = 3;  
y = x * x - 5 * x + 6;  
printf("%d\n", y);
```

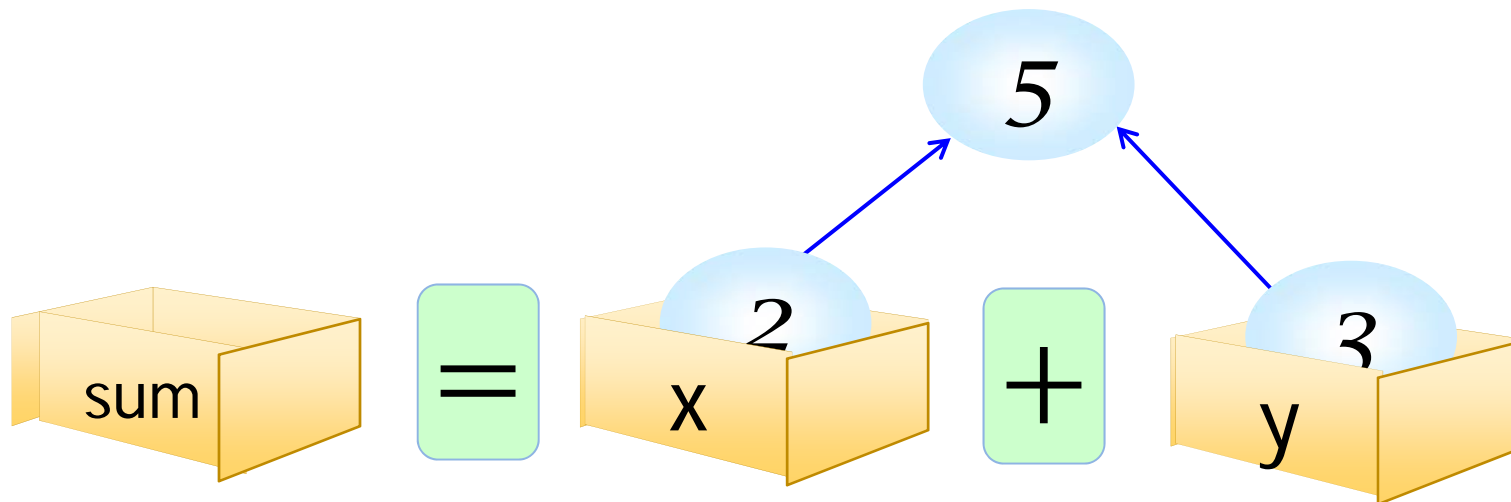
산술 연산

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
나머지	%	$x \% y$	$x \text{ mod } y$

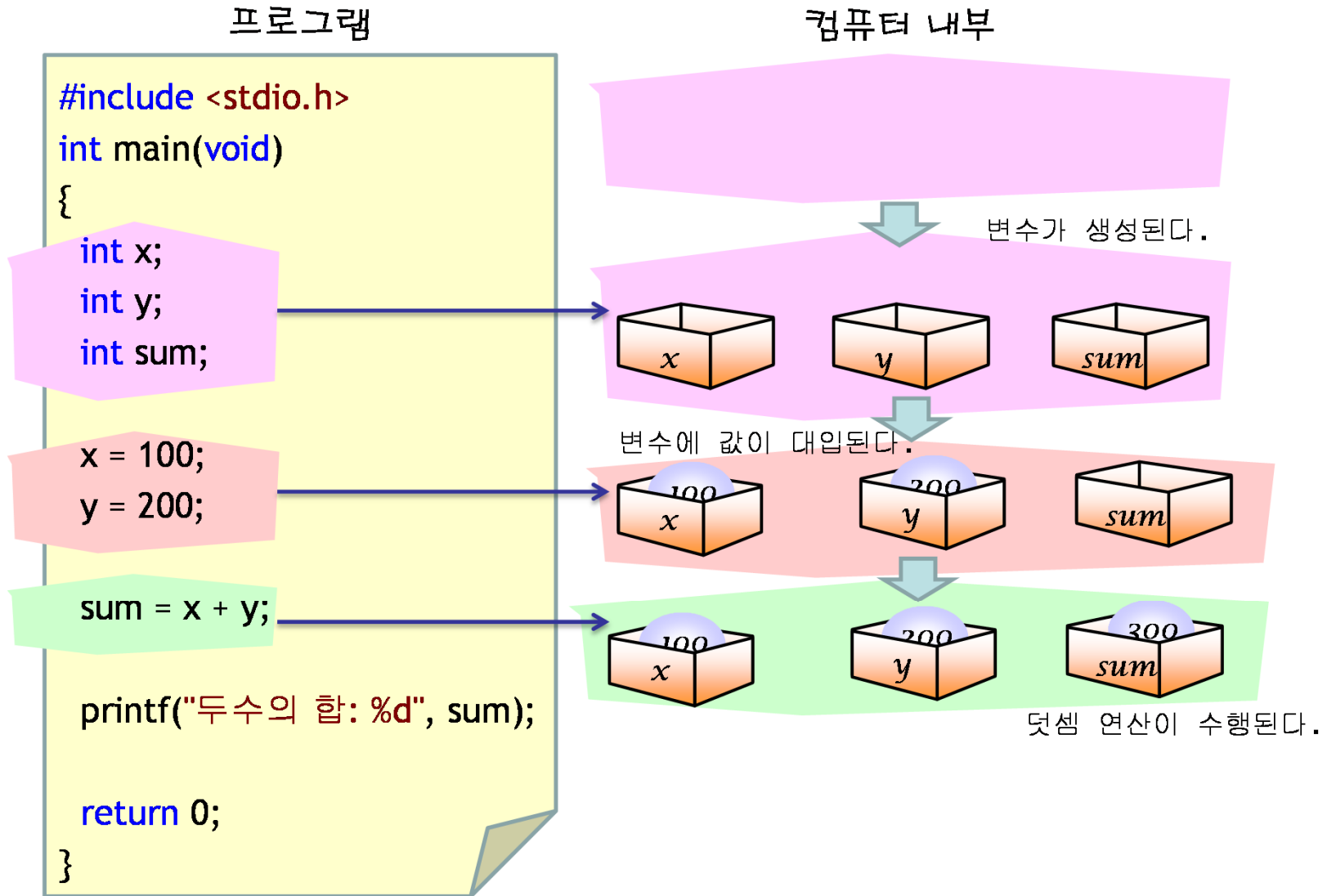


산술 연산

```
sum = x + y;
```

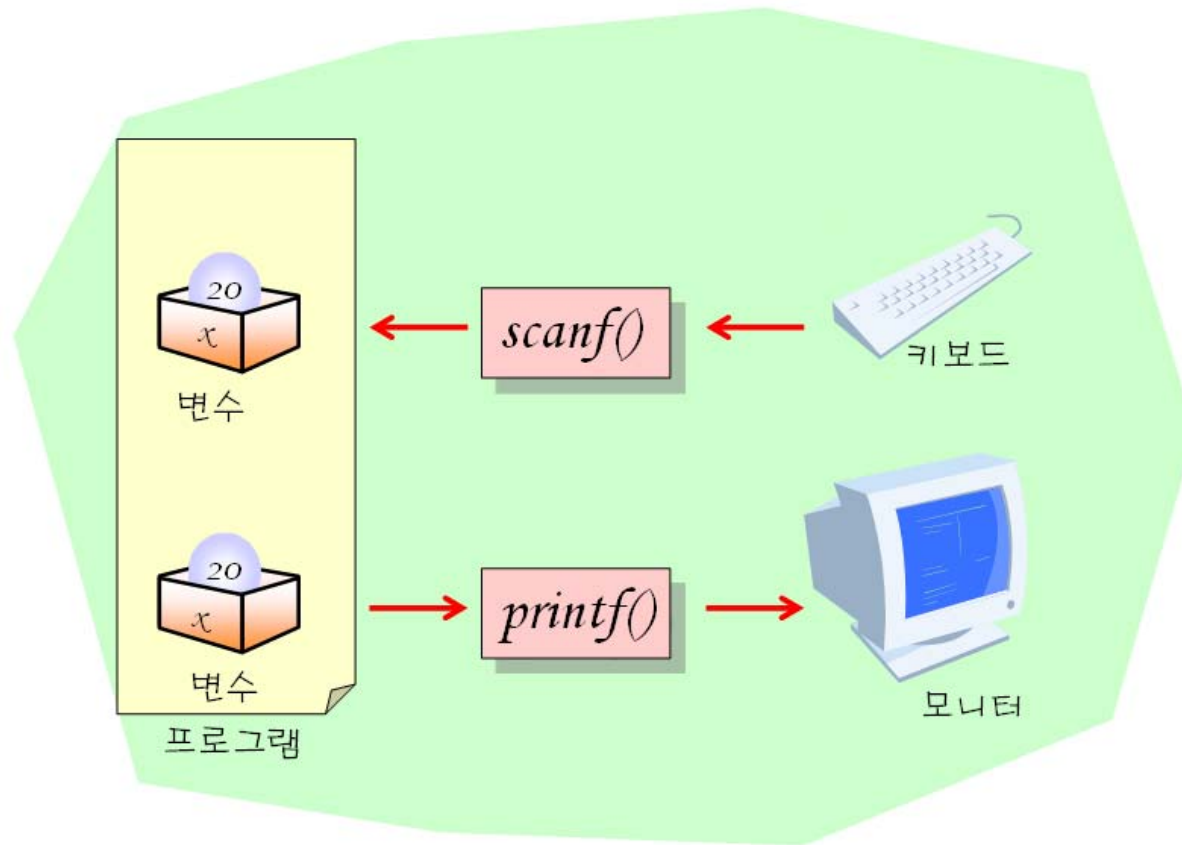


정리



printf()

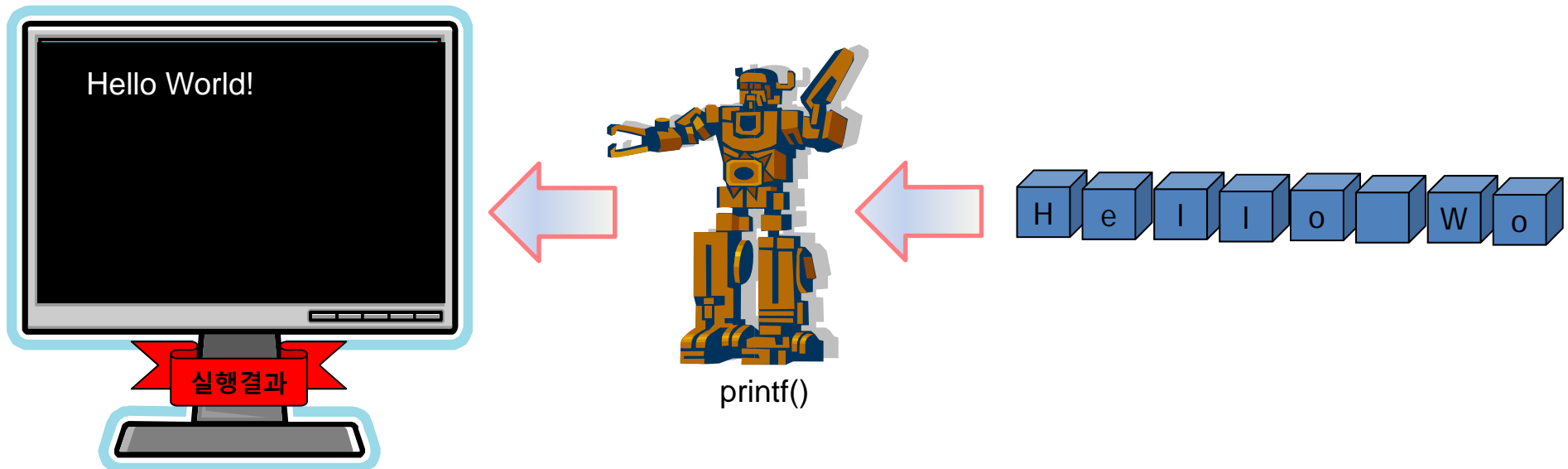
- ◆ printf(): 모니터에 출력을 하기 위한 표준 출력 라이브러리 함수



문자열 출력

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

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



변수값 출력

```
printf("두수의 합: %d", sum);
```



형식 지정자 의 개수와
변수의 개수와 순서는
같아야 한다.



printf() 함수의 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( "학번 %d 의 성적은 %f \n" , st_id , score );
```

학번 23의 성적은 3.99

형식 지정자 의 개수와
변수의 개수와 순서는
같아야 한다.



두번째 덧셈 프로그램

```
// 사용자로부터 입력받은 2개의 정수의 합을 계산하여 출력

#include <stdio.h>

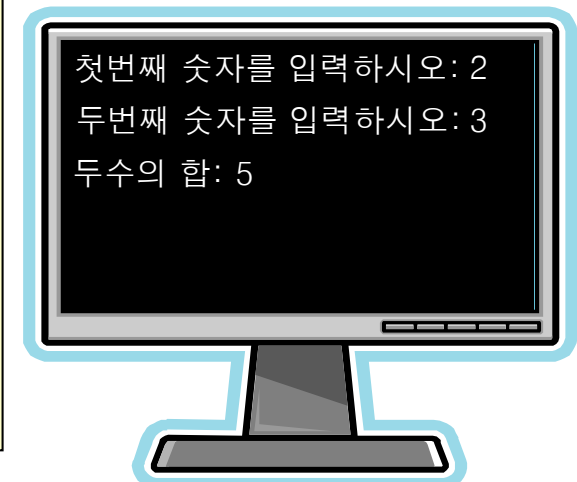
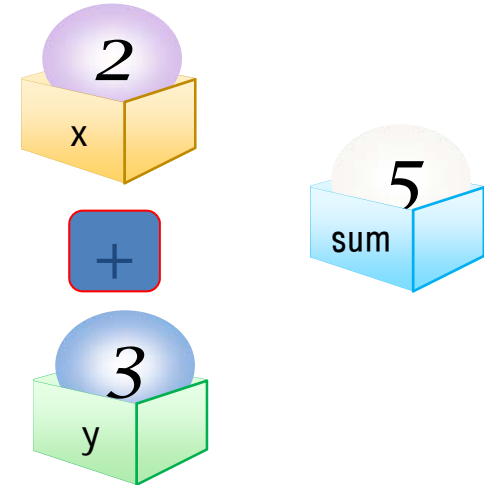
int main(void)
{
    int x;           // 첫번째 정수를 저장할 변수
    int y;           // 두번째 정수를 저장할 변수
    int sum;         // 2개의 정수의 합을 저장할 변수

    printf("첫번째 숫자를 입력하십시오:"); // 입력 안내 메시지 출력
    scanf("%d", &x);                       // 하나의 정수를 받아서 x에 저장

    printf("두번째 숫자를 입력하십시오:"); // 입력 안내 메시지 출력
    scanf("%d", &y);                       // 하나의 정수를 받아서 x에 저장

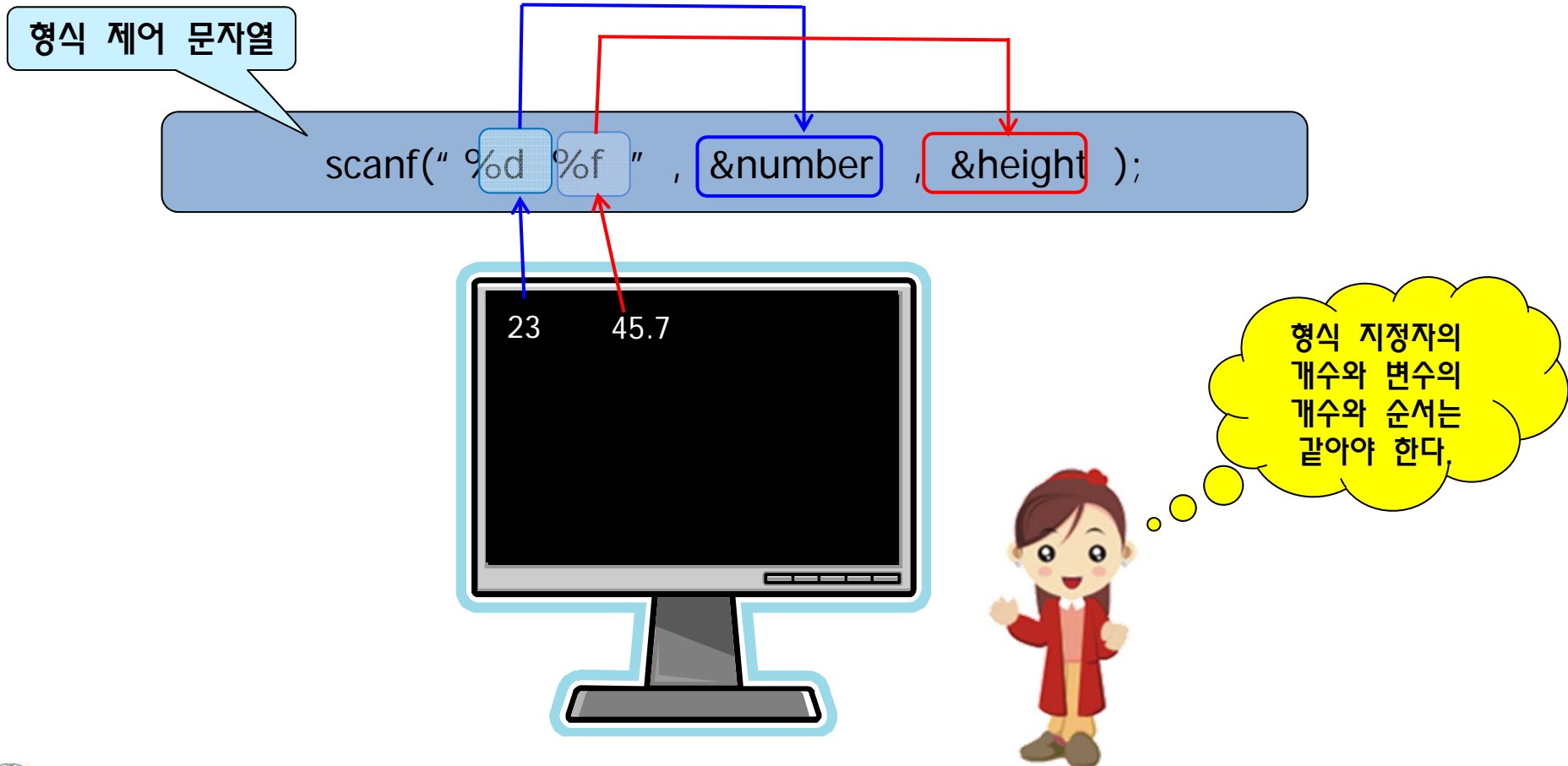
    sum = x + y;           // 변수 2개를 더한다.
    printf("두수의 합: %d", sum);           // sum의 값을 10진수 형태로
    출력

    return 0;             // 0을 외부로 반환
}
```



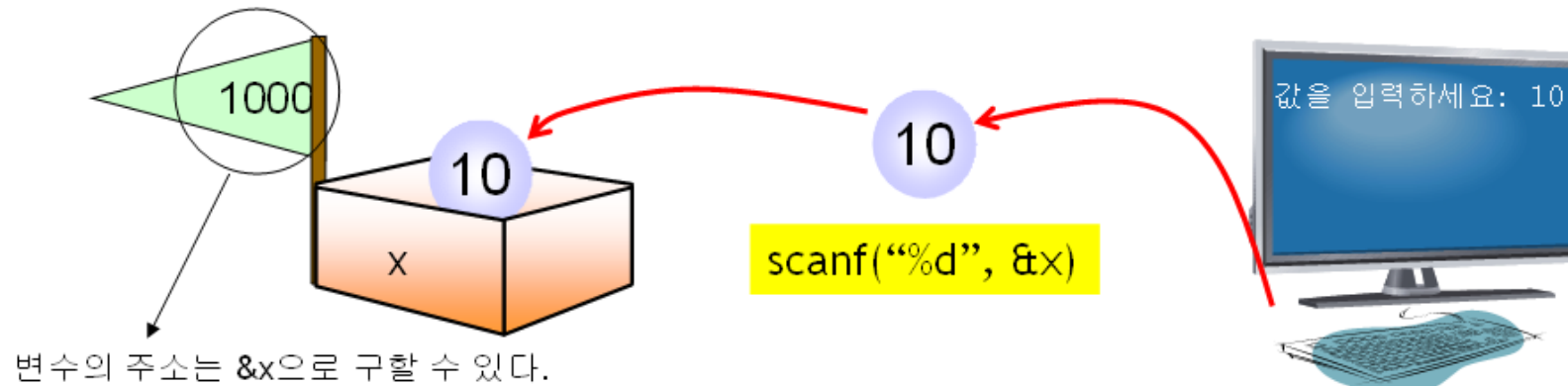
scanf()

- ◆ scanf(): 표준입력장치 (키보드)로부터 입력을 하기 위한 라이브러리 함수



scanf()의 동작

- ◆ 키보드로부터 값을 받아서 변수에 저장한다.
- ◆ 변수의 주소 (address)를 필요로 한다.



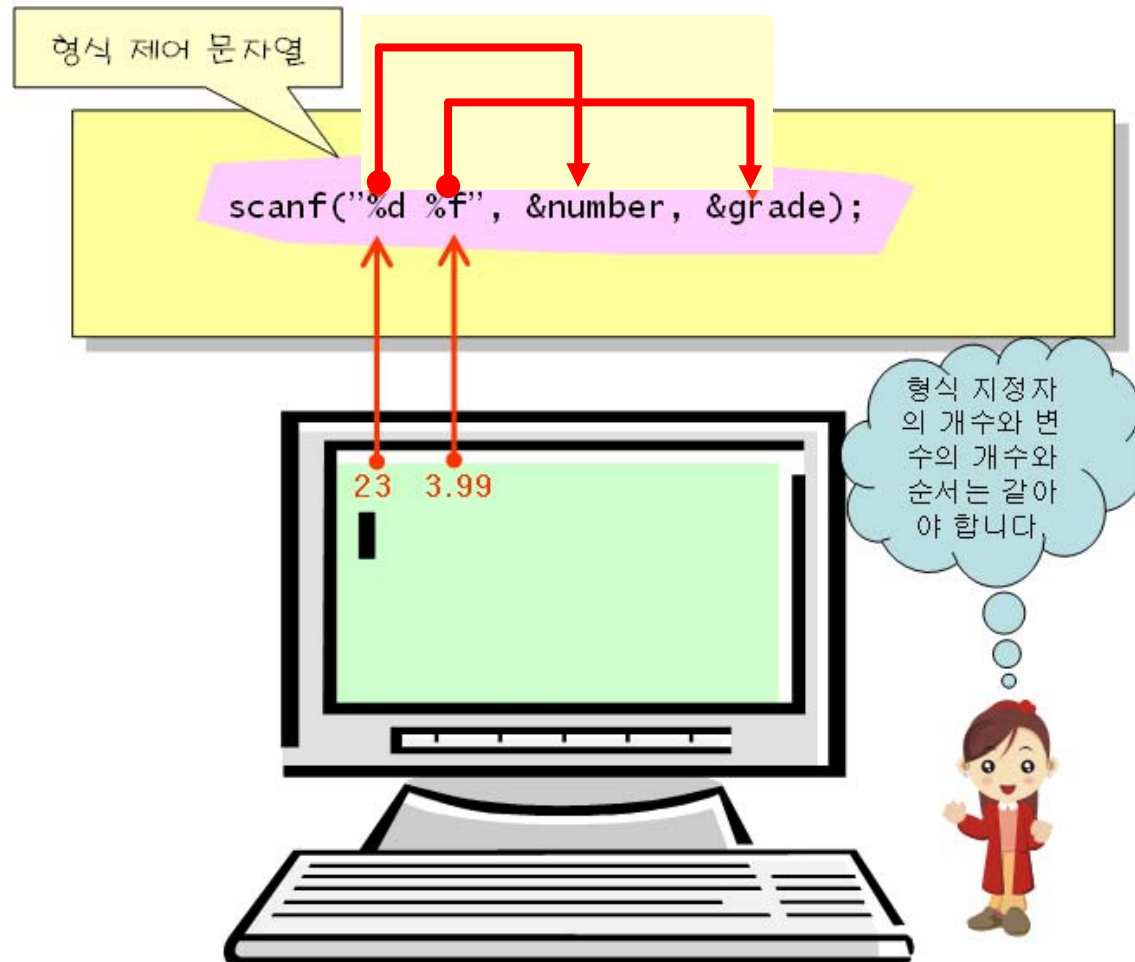
scanf()의 형식지정자

형식 지정자	의미	예
%d	정수를 10진수로 입력한다	scanf("%d", &i);
%f	float 형의 실수로 입력한다.	scanf("%f", &f);
%lf	double 형의 실수로 입력한다.	scanf("%lf", &d);
%c	문자 형태로 입력한다.	scanf("%c", &ch);
%s	문자열 형태로 입력한다.	char s[10]; scanf("%s", &s);



scanf()

◆ 형식 지정자와 변수의 자료형은 일치하여야 함



실수 입력

- ◆ `float ratio = 0.0;`
`scanf("%f", &ratio);`
- ◆ `double scale = 0.0;`
`scanf("%lf", &scale);`

주의!!!



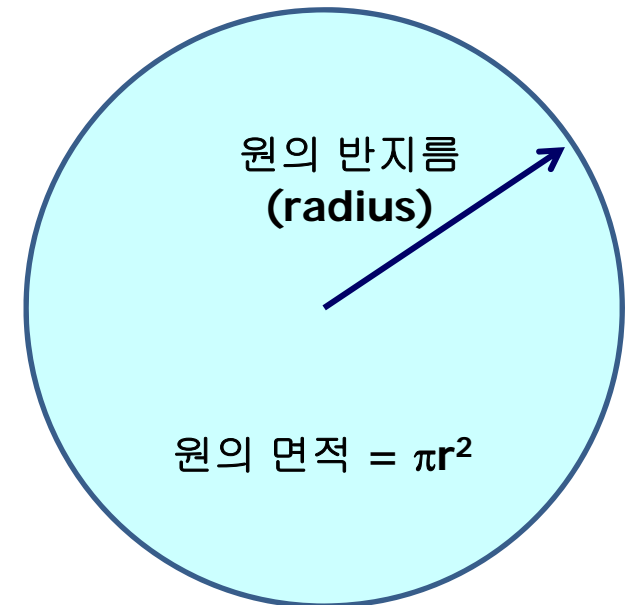
원의 면적 계산 프로그램

◆ 원의 반지름이 주어졌을 때, 면적 계산하기

- 원의 면적 = 3.14159 x 반지름 x 반지름

Algorithm AreaOfCircle()

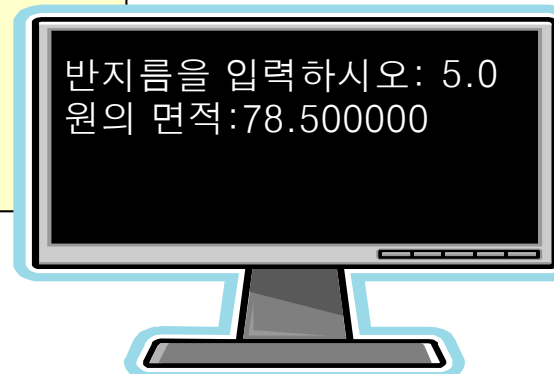
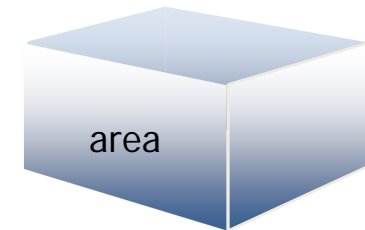
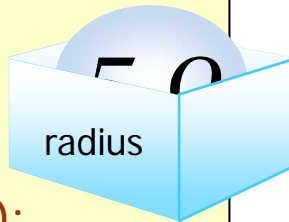
```
1: // input arguments: none
2: double radius; // local variable
3: double area; // local variable
4: input radius;
5: area = 3.14159 * radius * radius;
6: printout radius and area;
7: end
```



원의 면적 프로그램 소스코딩

```
/* 원의 면적을 계산하는 프로그램*/  
#include <stdio.h>  
  
int main(void)  
{  
    float radius; // 원의 반지름  
    float area; // 면적  
  
    printf("반지름을 입력하시오: ");  
    scanf("%f", &radius);  
  
    area = 3.14 * radius * radius;  
  
    printf("원의 면적: %f\n", area);  
  
    return 0;  
}
```

$$5.0 * 5.0 * 3.14 = 78.500000$$

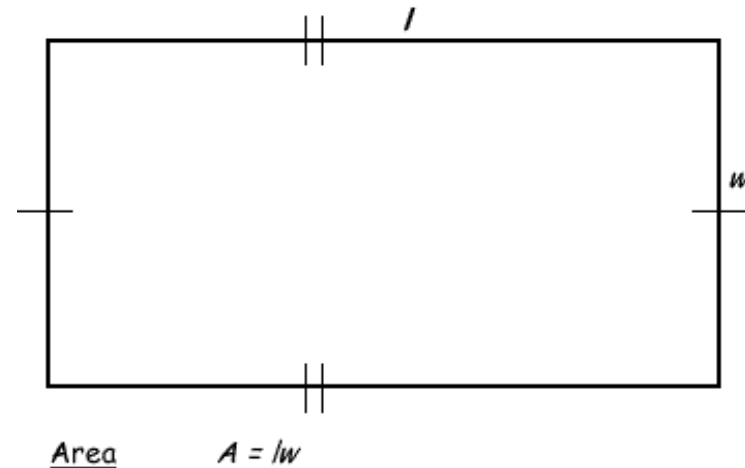


사각형의 둘레와 면적

- ◆ 필요한 변수는 w , h , $area$, $perimeter$ 라고 하자.
- ◆ 변수의 자료형은 실수를 저장할 수 있는 $double$ 형으로 하자.
- ◆ $area = w * h$;
- ◆ $perimeter = 2 * (w + h)$;

Rectangle

Perimeter $p = 2(l + w)$
or
 $p = 2l + 2w$



사각형의 둘레와 면적 계산 – Pseudo code

Algorithm PerimeterAndAreaOfRectangle()

```
1: // input arguments: none
2: double width; // local variable
3: double height; // local variable
4: double area;
5: double perimeter;
6: input width;
7: input height;
8: area = width * height;
9: perimeter = (width + height) * 2.0;
10: printout area and perimeter of rectangle
    with width and height;
11: end
```



사각형의 둘레와 면적 계산 – Source Coding

```
/** Perimeter and Area of Rectangle */
#include <stdio.h>

int main(void)
{
    double width;
    double height;
    double area;
    double perimeter;

    printf("사각형의 폭과 높이를 입력하시오: ");
    scanf("%lf", &width);
    scanf("%lf", &height);
    area = width*height;
    perimeter = 2*(width+height);
    printf("폭 (%lf), 높이(%lf)인 사각형: ");
    printf(" 넓이: %lf,", area);
    printf(" 둘레: %lf\n", perimeter);
    return 0;
}
```



Homework 1

- 1.1 1부터 10까지를 더하여 합을 구하고, 그 평균을 구하는 알고리즘의 pseudo code를 작성하라.
- 1.2 100명의 학생들의 성적을 차례로 입력하고, 가장 우수한 성적, 가장 낮은 성적, 평균 성적을 구하는 알고리즘의 pseudo code를 작성하라.
- 1.3 사용자로부터 원기둥의 반지름, 원기둥의 높이를 입력받고, 이 원기둥의 체적 (volume)과 표면적 (surface area)를 계산하는 프로그램의 pseudo code를 작성하라.
- 1.4 위1.3문제에서 작성한 pseudo code에 따라 C 프로그램으로 작성하고, 실행화면을 capture하여 source code와 함께 제출하라.

