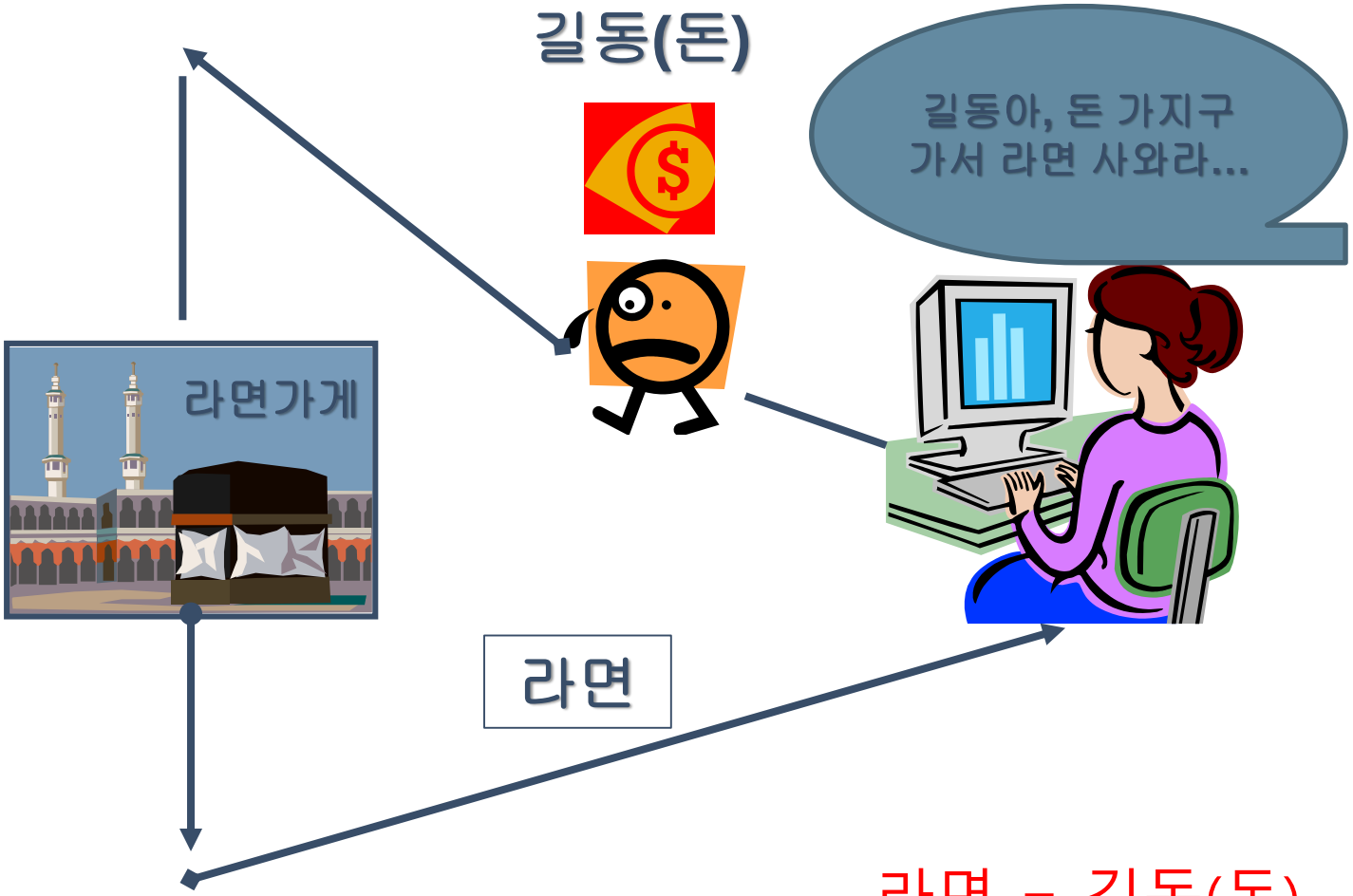
제3장 C프로그램 구성

C Express

C 프로그램

- **함수들의 집합** (a set of functions)
  - 그러나 반드시 "main" 이라는 아름을 가진 하나의 함수가 존재하여야 한다. --- 모든 C 프로그램 시작을 위한 진입점(entry point)
  - 그러므로 모든 C 프로그램은 최소한 1개 이상의 함수를 가지게 된다
- **블록**구조 언어(Block Structured Language)
  - 중괄호({, })로 둘러싸인 프로그램 영역을 block이라고 한다
  - 블록 내부에 또 다른 블록이 존재할 수 있으며 블록의 설정은 프로그래머의 의도에 따라 다르게 이루어질 수 있다.
  - 모든 함수는 하나의 블록이다
- **함수** (Function)
  - 어떤 특정 작업을 수행하는 프로그램으로 입력 매개변수를 가질 수 있으며, 작업의 결과를 호출한 함수로 반환할 수 있다.
  - Input -→ Process -→ Output
  - 라이브러리 함수(library function)
  - 사용자 함수(user function)

함 수(Function)



라면 = 길동(돈)

알았다

아빠(돈)

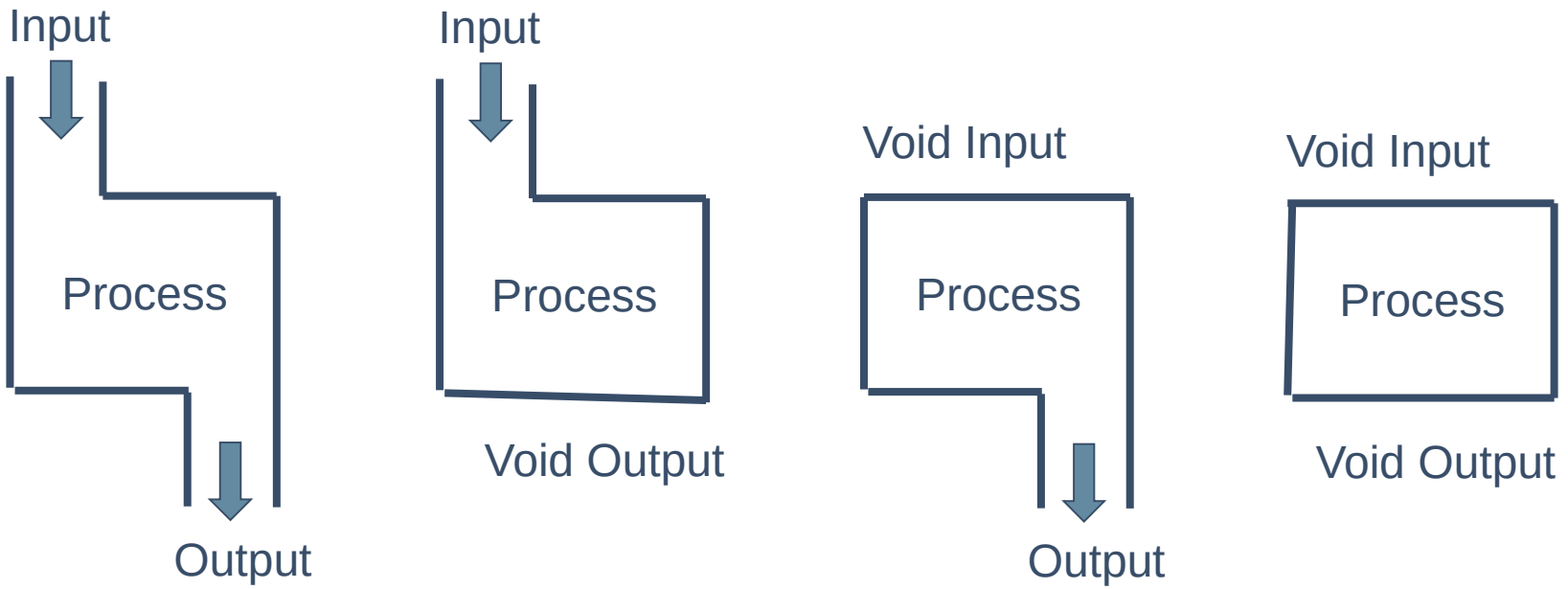
아빠 오시는 길에
라면 사오세요...

길동(돈)

길동아, 돈 가지구
가서 라면 사와라...

라면

함수의 동작 유형



입출력장치

표준입력장치 : Standard Input (stdin) → 키보드 (keyboard)
표준출력장치 : Standard Output (stdout) → 화면 (monitor)

C 프로그램 맛보기-1

```c
/* Hello World를 화면에 출력하는 프로그램*/
#include <stdio.h>

int main(void)
{
        printf("Hello World"); // "" 내부의 내용을 그대로 출력

        return 0;
}
```

```c
/* Hello World를 화면에 출력하는 프로그램*/
#include <stdio.h>

int main(void)
{
        printf("Hello World\n"); // \n 추가

        return 0;
}
```

printf() 함수 개념

C 프로그램 맛보기-2

```c
/* 정해진 두 정수를 더하여 그 결과를 출력하는 프로그램*/
#include <stdio.h>

int main(void)
{
        int x = 100;      // 첫번째 정수를 저장할 저장소
        int y = 200;      // 두번째 정수를 저장할 저장소
        int sum = 0;      // 두 정수의 합을 저장할 저장소

        // x = 100;
        // y = 200;

        sum = x + y;
        printf("두 정수의합: %d", sum);
        // printf("%d + %d = %d 입니다" , x, y, sum);
        // printf("100 + 200 = %d 입니다" , sum);

        return 0;
}
```

맛보기-1 + % 개념 추가 + 저장소 개념 추가

C 프로그램 맛보기-3

```c
/* 키보드로부터 두개의 정수를 받아서,                */
/* 그 두 정수를 더하여 그 결과를 출력하는 프로그램*/

#include <stdio.h>

int main(void)
{
        int x = 0;          // 첫번째 정수를 저장할 저장소
        int y = 0;          // 두번째 정수를 저장할 저장소
        int sum = 0;        // 두 정수의 합을 저장할 저장소

        printf("첫번째 수를 입력하세요: ");
        scanf("%d", &x);
        printf("두번째 수를 입력하세요: ");
        scanf("%d", &y);

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

        return 0;
}
```

맛보기-2 + & 개념 추가 + scanf() 함수 개념 추가

C 프로그램 맛보기-4

```c
/* 키보드로부터 두개의 정수를 받아서,                */
/* 그 두 정수를 더한 결과와 뺀 결과를 출력하는 프로그램*/
#include <stdio.h>
int main(void)
{
        int x, y = 0;              // 두 정수를 저장할 저장소
        int sum, diff = 0;    // 합과 차이를 저장할 저장소

        printf("첫번째 수를 입력하세요: ");
        scanf("%d", &x);
        printf("두번째 수를 입력하세요: ");
        scanf("%d", &y);

        sum = add(x, y);
        diff = sub(x, y);
        printf("두 정수의 합: %d\n", sum);
        printf("두 정수의 차: %d\n", diff);
        return 0;
}
int add(int num1, int num2) {

        return(num1+num2);
}
int sub(int num1, int num2) {

        return(num1-num2);
}
```

맛보기-3 + 함수 개념 + 매개변수 개념 추가

주석(Comment)

Q) 주석(comment)이란 무엇인가?

A) 프로그램이 하는 작업을 설명하는 글

Q) 주석은 반드시 있어야 하는가?

A) 컴파일러는 주석을 무시한다

Q) 주석은 누구를 위한 것인가?

A) 주석은 프로그램을 읽는 사람을 위한 것이다

Q) 주석의 역할은 무엇인가?

A) 주석은 프로그램의 가독성을 높인다.

Q) 좋은 주석은 어떤 것인가?

A) 코드를 반복하거나 코드를 설명하기 보다는 코드를 작성한 의도를 나타내는 것이 좋다.

주석을 붙이는 방법

```
/* 한줄로 된 주석*/

int x;   /* 줄의 일부분인 주석*/

/* 여러
   줄로
   된 주석*/
```

```
// 이 줄은 전체가 주석이다.
int x; // 변수 x 선언
```

```
/*
 * 파일 이름: add.c
 * 설명     : 두수를 더하는 프로그램
 * 작성자   : In-Gook Chun
 */
/**************************************
 * 파일 이름: add.c
 * 설명     : 두수를 더하는 프로그램
 * 작성자   : In-Gook Chun
 **************************************/
```

들여쓰기(Indentation)

빈줄을 넣어서 의미별로 구분을 한다.

프로그램의 시작부분에는 파일 이름이나 작성자, 작성일자, 프로그램의 내용등을 적는다

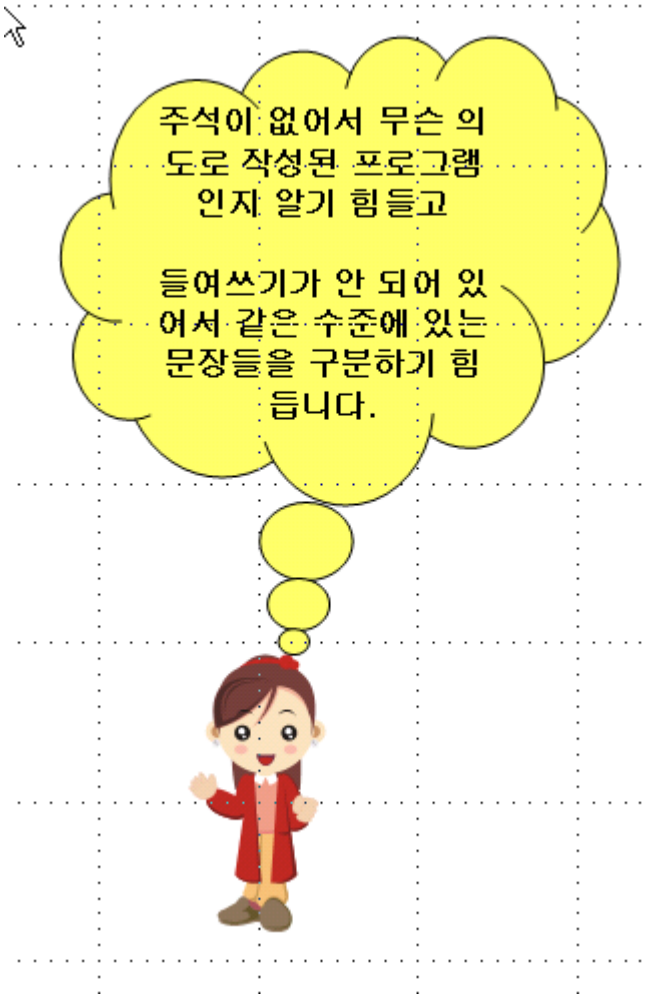
문장들의 의미(의도)를 주석으로 설명한다.

같은 내용의 처리이면 탭이나 공백을 넣어 들여쓰기를 한다.

```
/* 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;
}
```

주석과 들여 쓰기가 없다면..

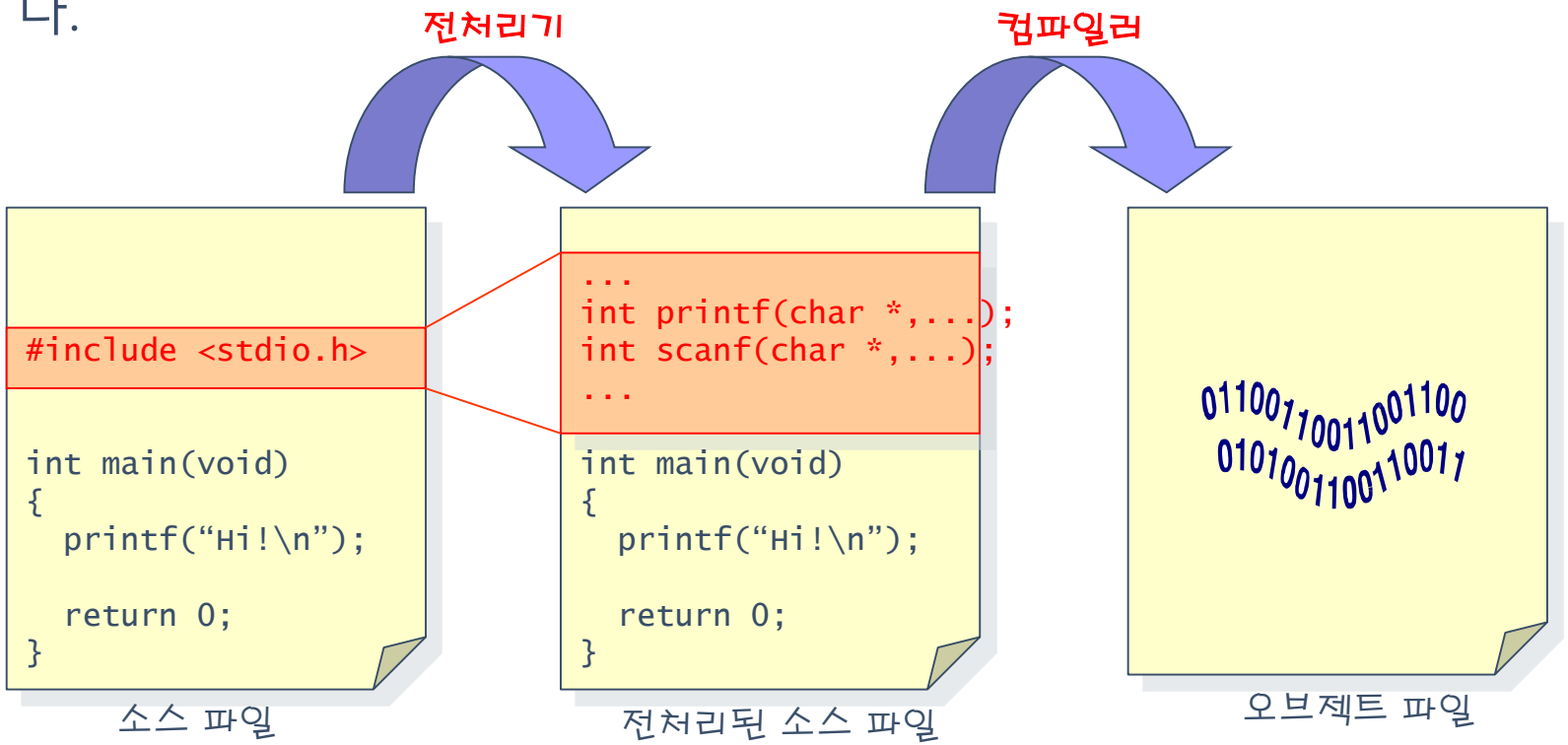
```
#include <stdio.h>
int main(void)
{
int x;
int y;
int sum;
x = 100;
y = 200;
sum = x + y;
printf("두수의 합: %d", sum);
return 0;
}
```



전처리기(Preprocessor)

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

- #기호로 시작
- 헤더 파일 stdio.h를 소스 코드 안에 포함
- stdio.h는 표준 입출력에 대한 라이브러리 함수의 정의가 들어 있다.



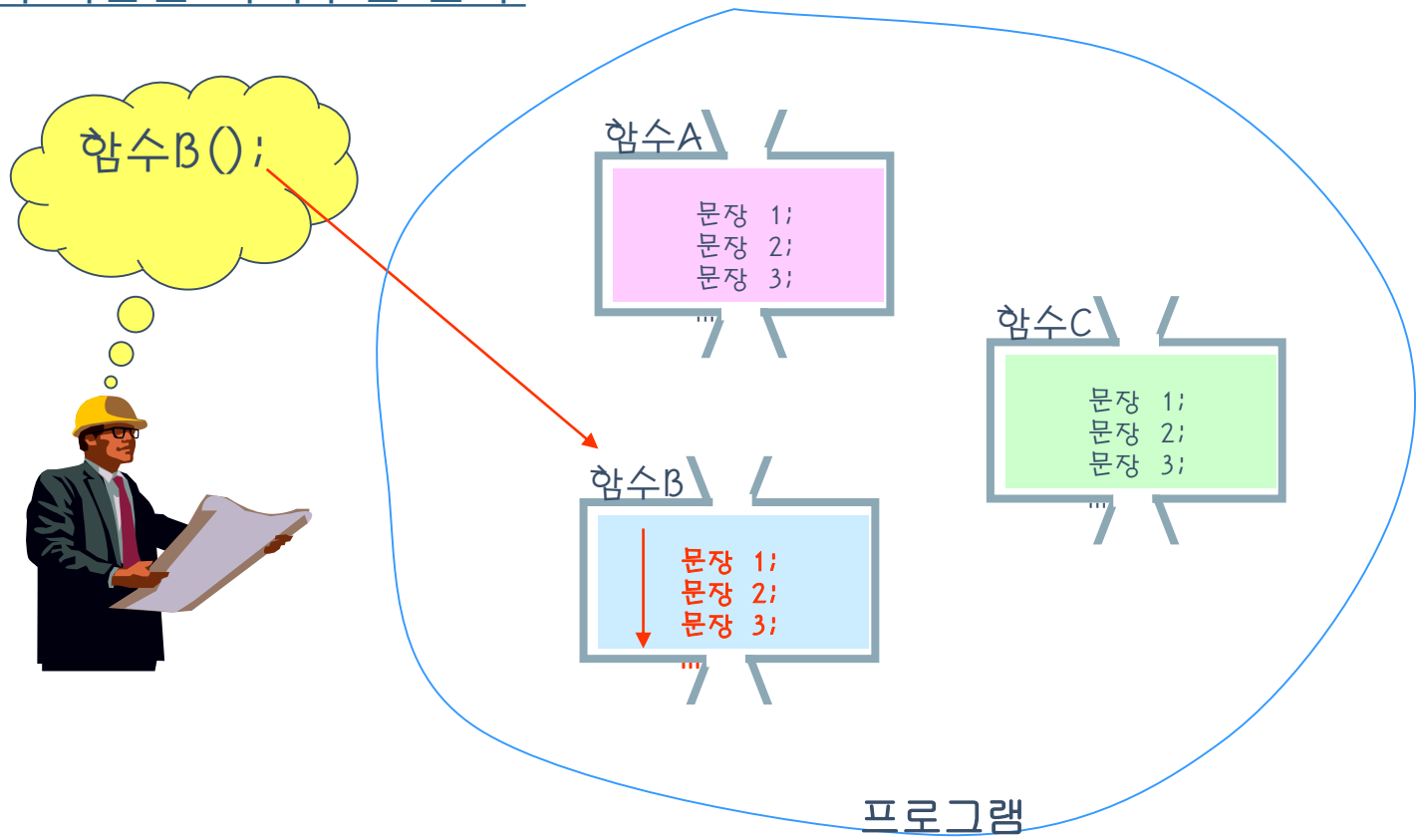
함수 호출(call)

Q) 함수 안에 있는 문장들은 언제 실행되는가?

A) 함수가 호출되면 실행된다.

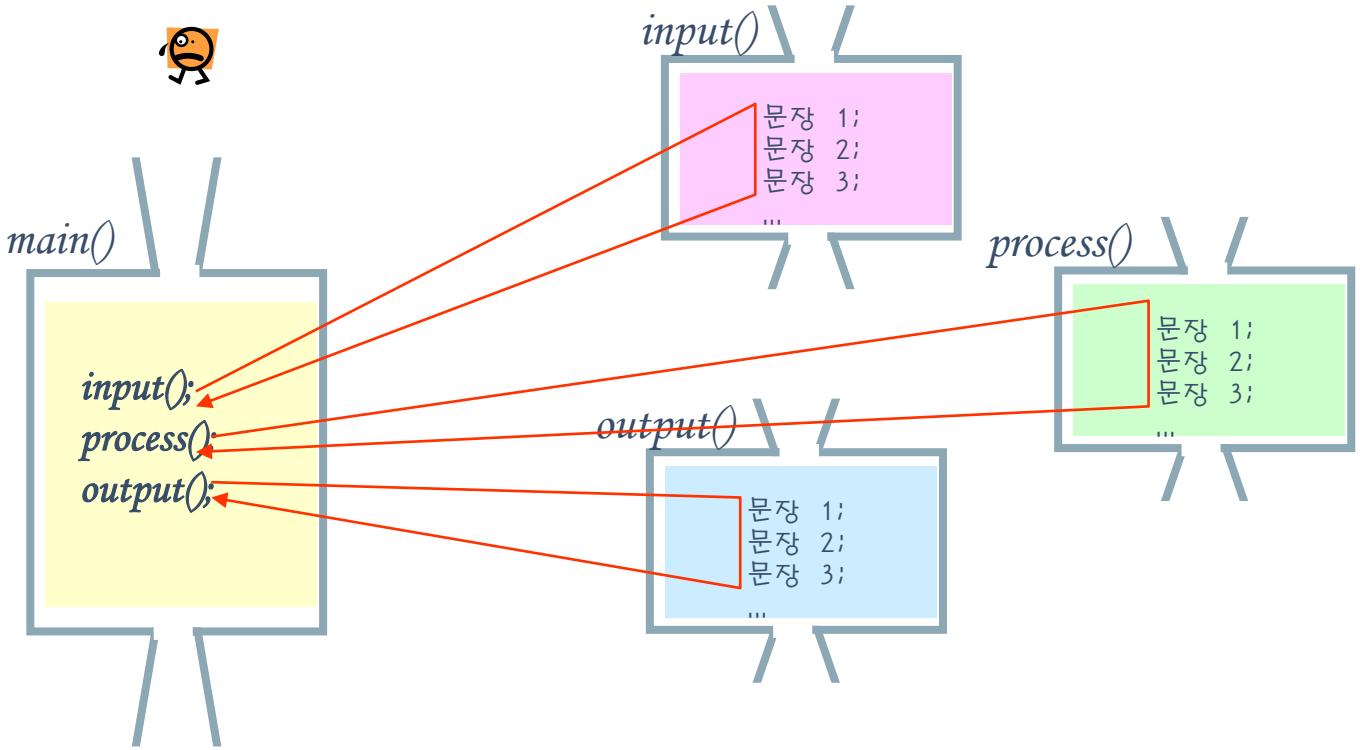
Q) 함수 호출은 어떻게 하는가?

A) 함수의 이름을 적어주면 된다.



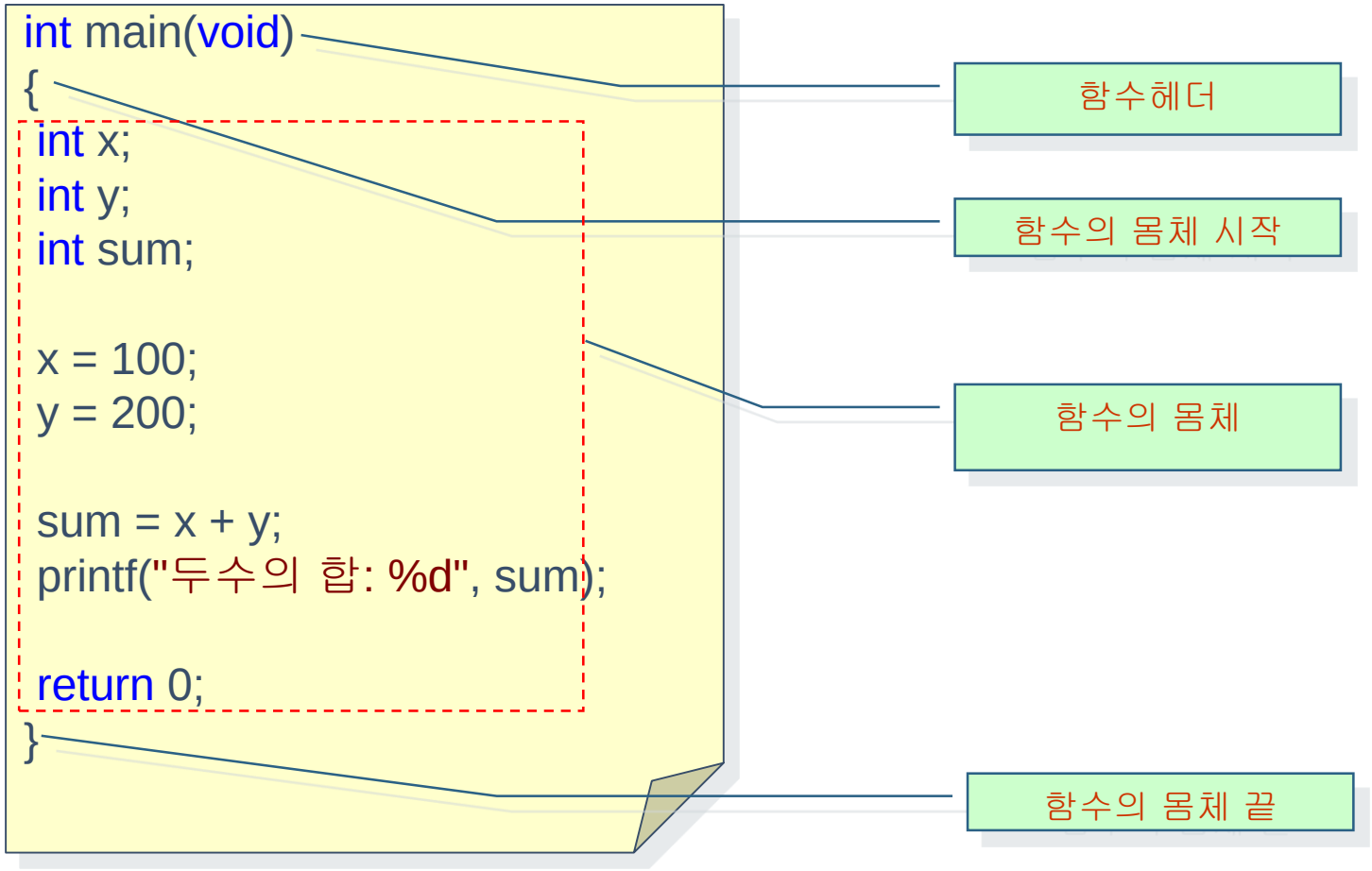
Q) 많은 함수 중에서 가장 먼저 실행되는 것은?

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



함수의 구조

- **함수 = 함수 헤더 + 함수 몸체**



함수 헤더와 몸체

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

함수 헤더

- int: 함수가 반환하는 값의 형태
- main: 함수 이름
- (void):입력이 없다는 의미

함수 몸체

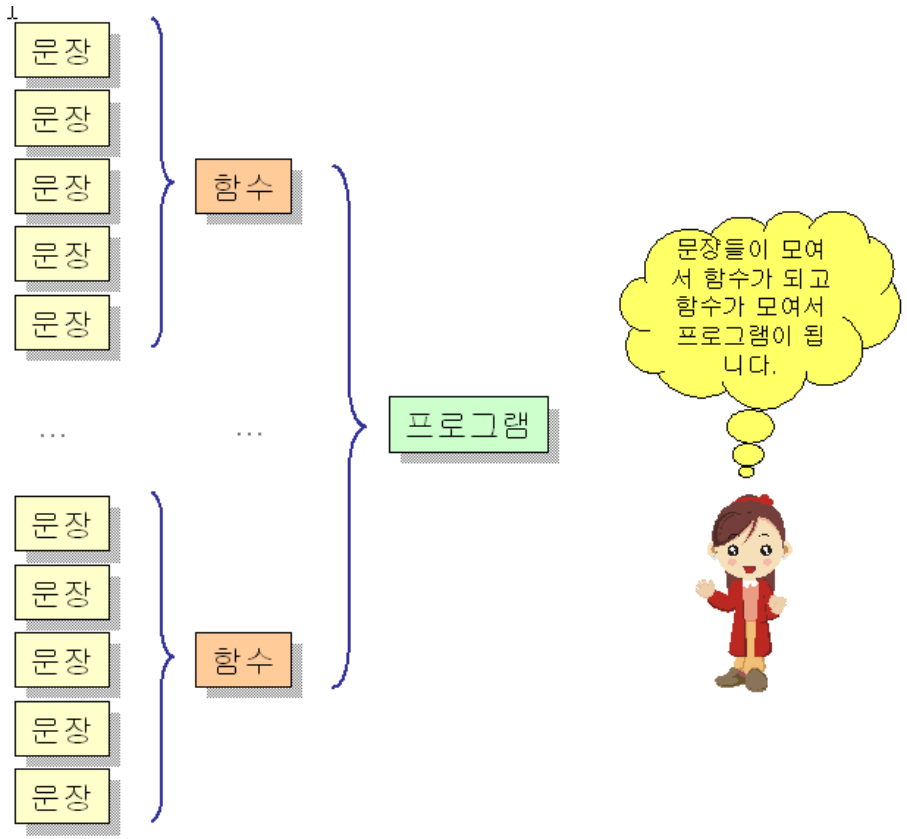
- 함수가 하는 작업에 해당
- 문장들로 구성된다.
- 문장이 하나도 없을 수도 있다.

return 문장

- 함수를 종료하면서 값을 반환
- 일반적으로 main 함수의 경우, 0값은 성공, 1은 실패를 의미

문장(statement)

- *문장(statement):* 컴퓨터에게 작업을 지시하는 단위
- 문장의 끝은 ;으로 끝난다.

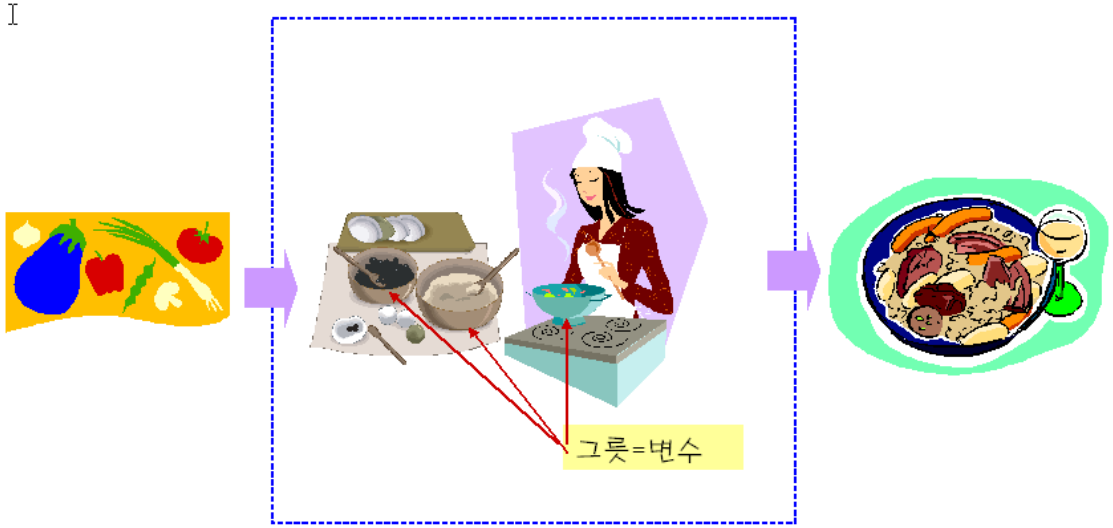


변수(variable)

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

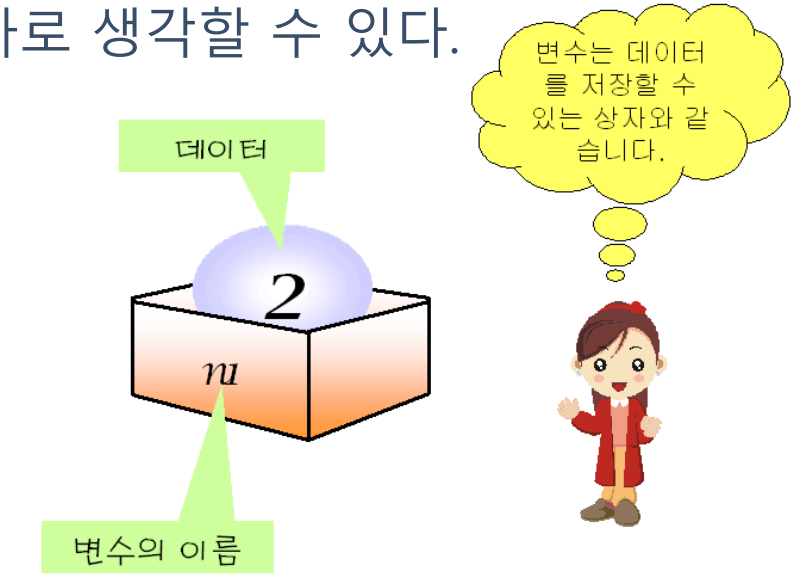
Q) 변수란 무엇인가?

A) 프로그램이 사용하는 시간에 따라 변하는 데이터를 일시적으로 저장할 목적으로 사용하는 주메모리 상의 공간

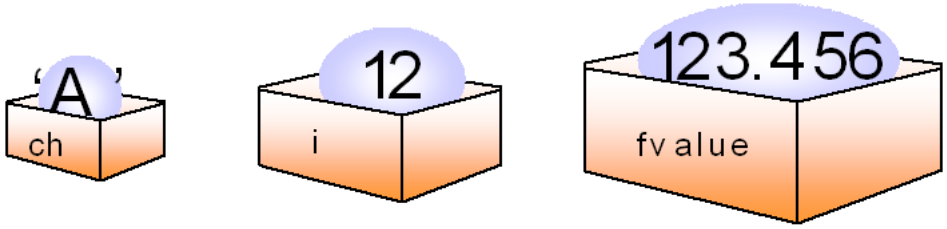


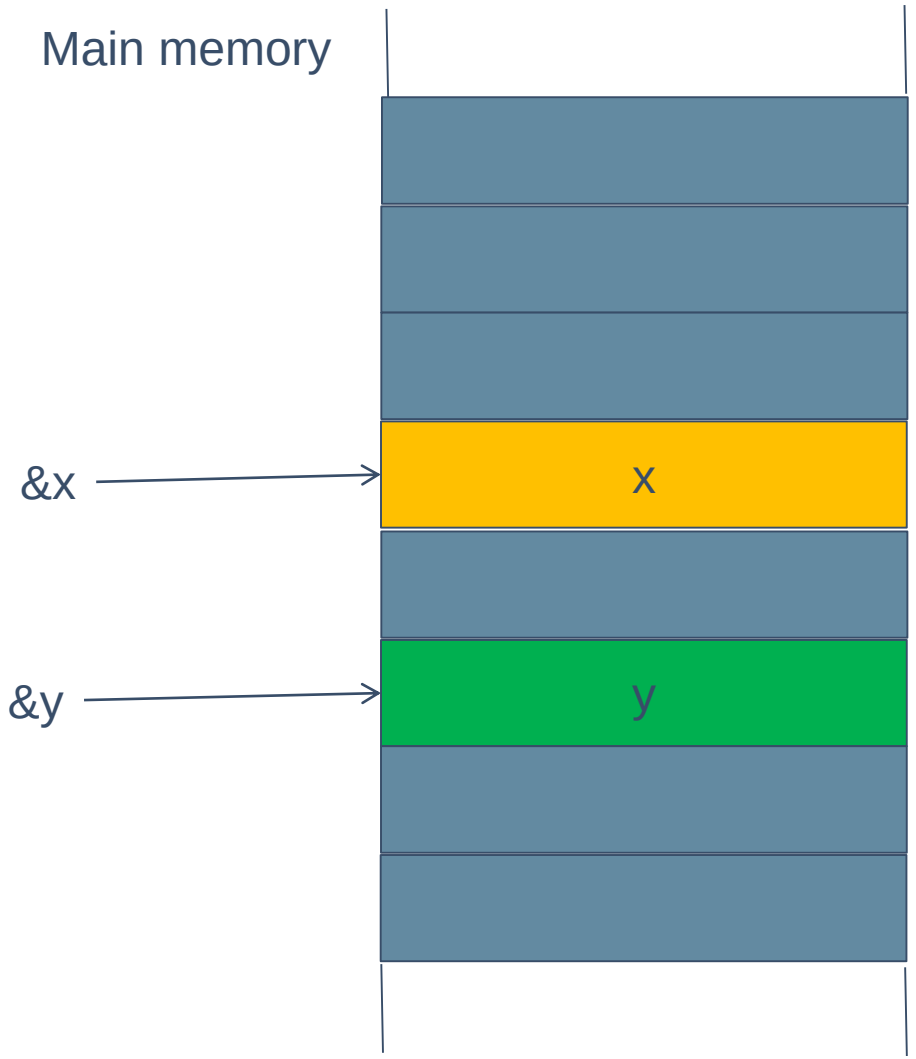
변수의 종류

- 변수는 데이터를 담는 상자로 생각할 수 있다.



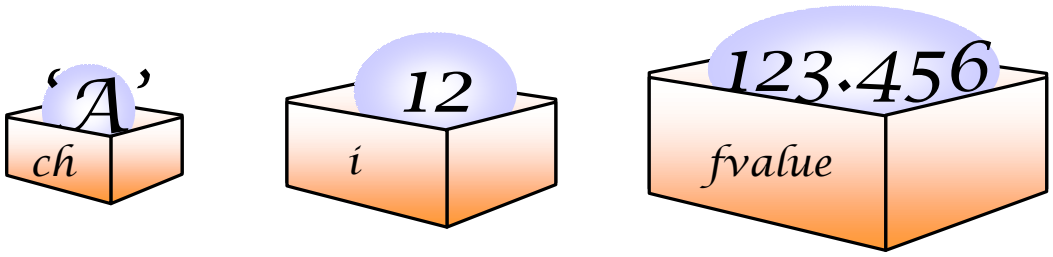
- 변수에는 데이터의 종류에 따라 여러 가지 타입이 존재한다.




Main memory
&x
x
&y
y

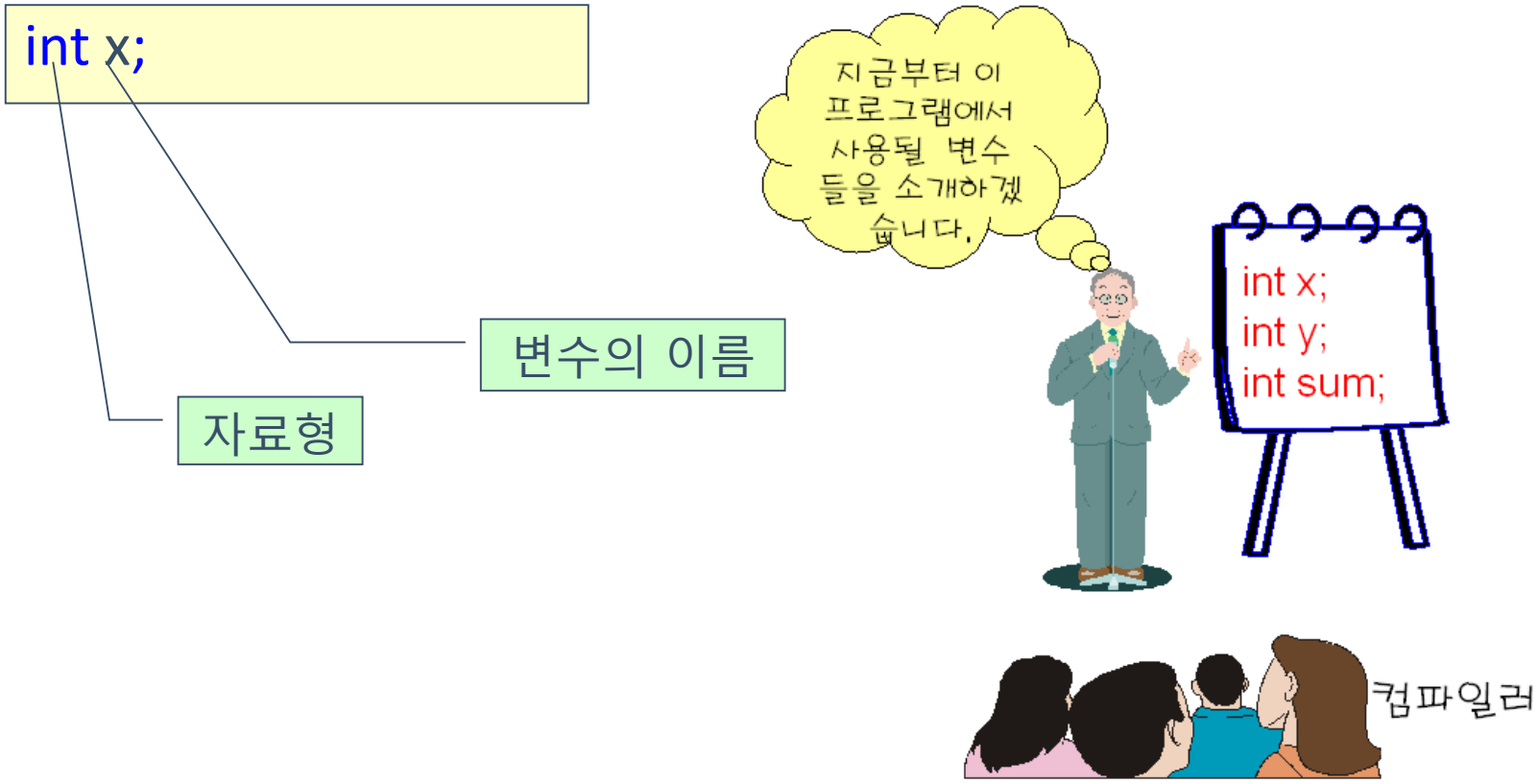
변수의 이름

- 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



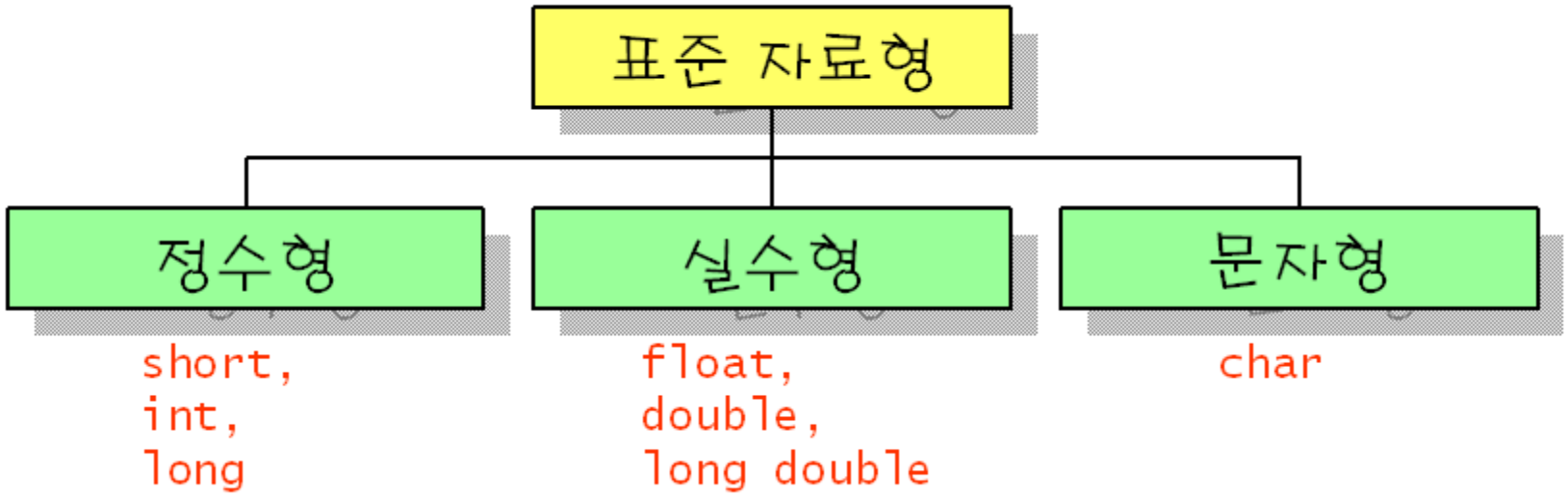
변수 선언

- 변수 선언: 컴파일러에게 어떤 타입의 변수가 사용되는지를 미리 알리며, 그 변수를 위한 저장공간 할당요청

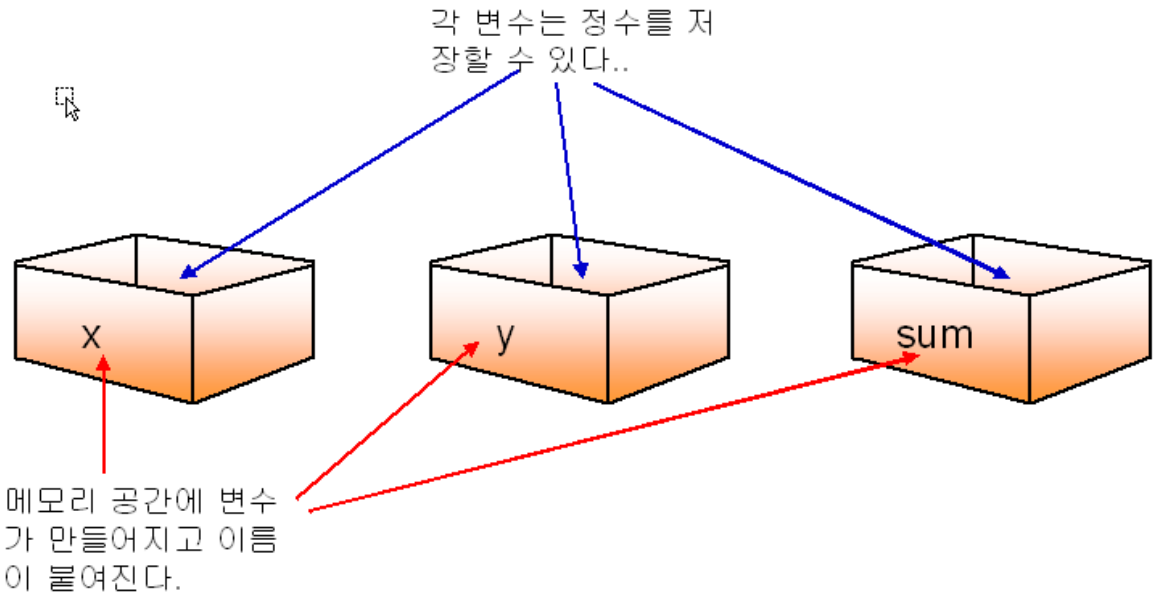


자료형(Data Type)

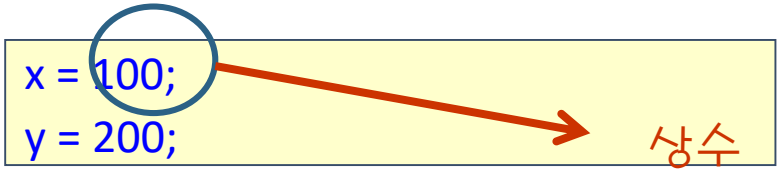
- 자료형(data type): 변수가 저장할 데이터가 정수인지 실수인지, 아니면 또 다른 어떤 데이터인지를 지정하는 것이며, 유형에 따라 그 길이가 다르다



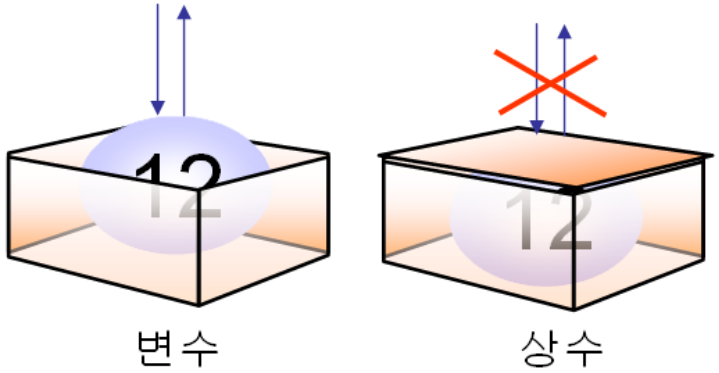
```
int x;      // 첫번째 정수를 저장하는 변수
int y;      // 두번째 정수를 저장하는 변수
int sum; // 두 정수의 합을 저장하는 변수
```

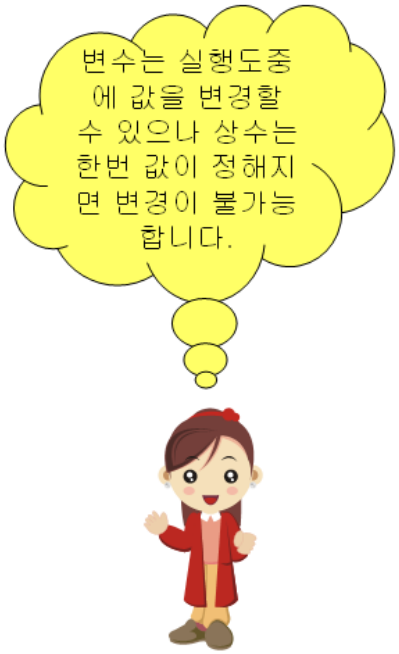


상수(constant)



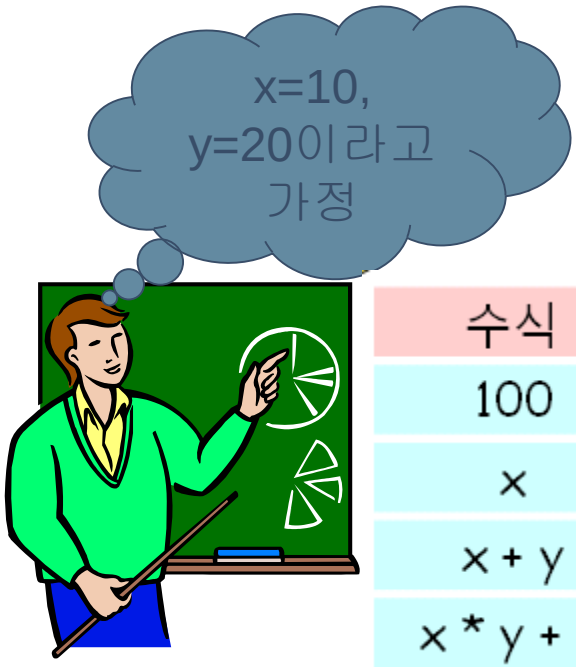
- 상수(constant): 그 값이 프로그램이 실행하는 동안 변하지 않는 데이터





수식(Expression)

- 수식(expression): 변수, 상수, 연산자 등으로 구성된 식
- 수식은 결과값을 가진다.

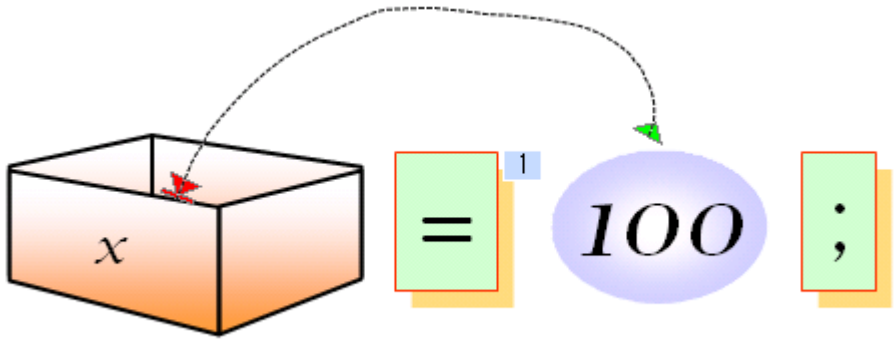


수식	결과값	설명
100	100	하나의 상수로 이루어진 수식
x	10	하나의 변수로 이루어진 수식
x + y	30	변수와 연산자로 이루어진 수식
x * y + 5	205	상수, 변수, 연산자로 이루어진 수식

대입 연산

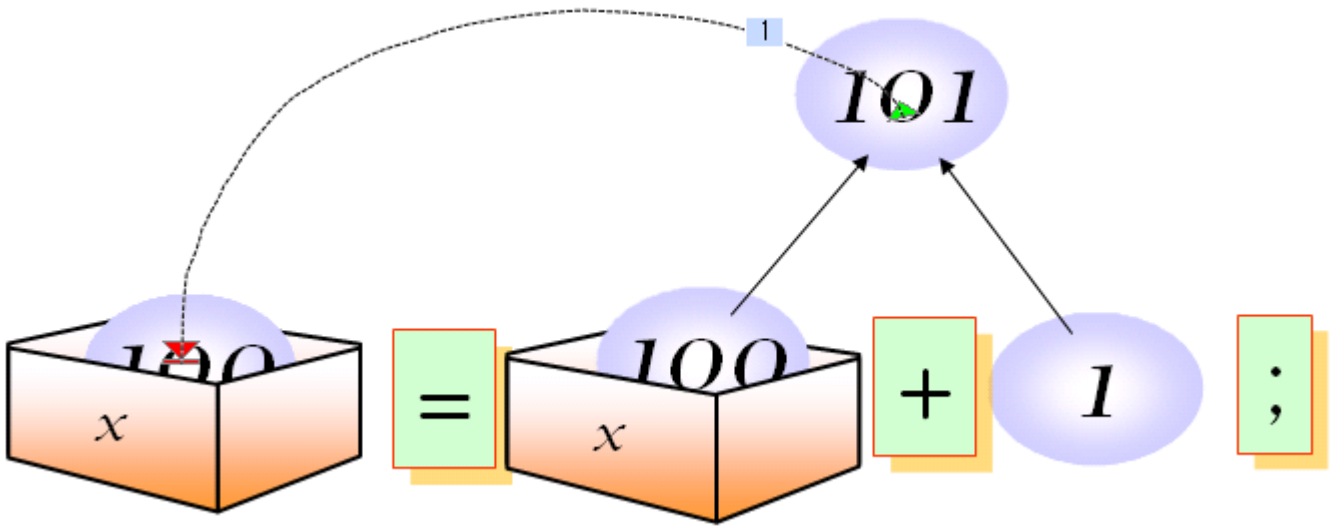
- 대입 연산(assignment operation): 변수에 값을 저장하는 연산
- 대입 연산 = 배정 연산 = 할당 연산

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



- 다음과 같은 연산은 변수 x의 값을 하나 증가시킨다.
- 수학적인 의미와는 다름

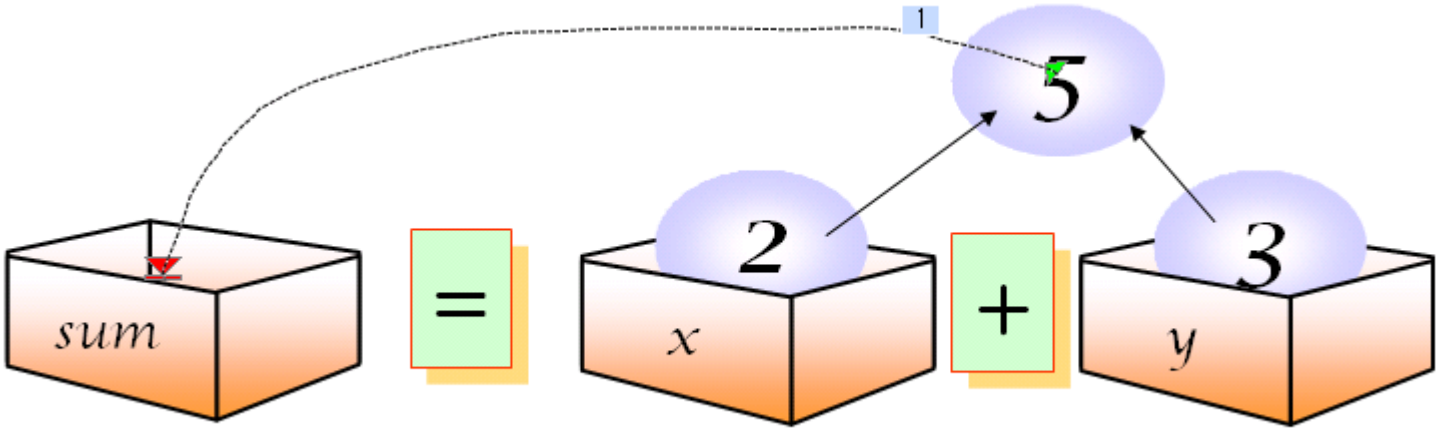
```
x = x + 1;
```



산술 연산

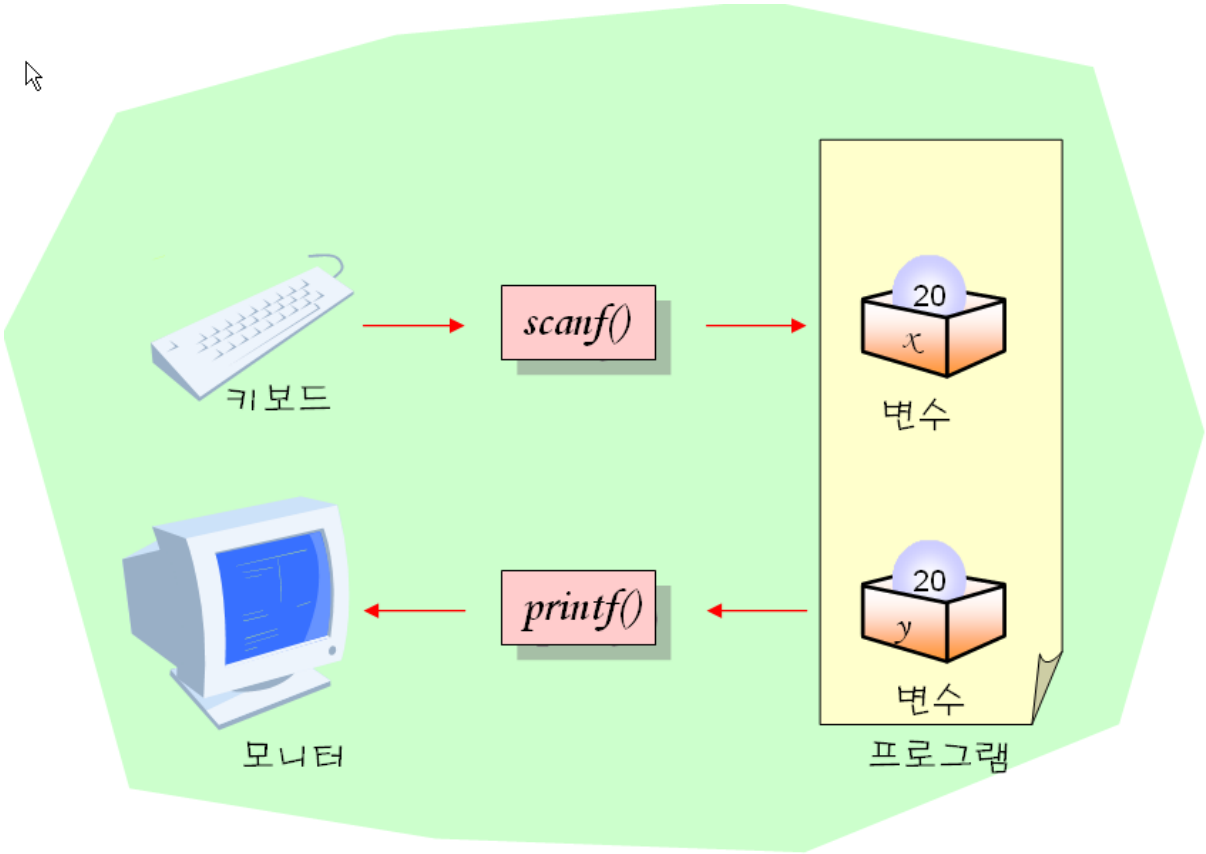
연산	연산자	C 수식	수학에서의 기호
덧셈	+	x + y	x + y
뺄셈	-	x - y	x - y
곱셈	*	x * y	xy
나눗셈	/	x / y	x/y 또는 또는 x ÷ y
나머지	%	x % y	x mod y

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



printf()

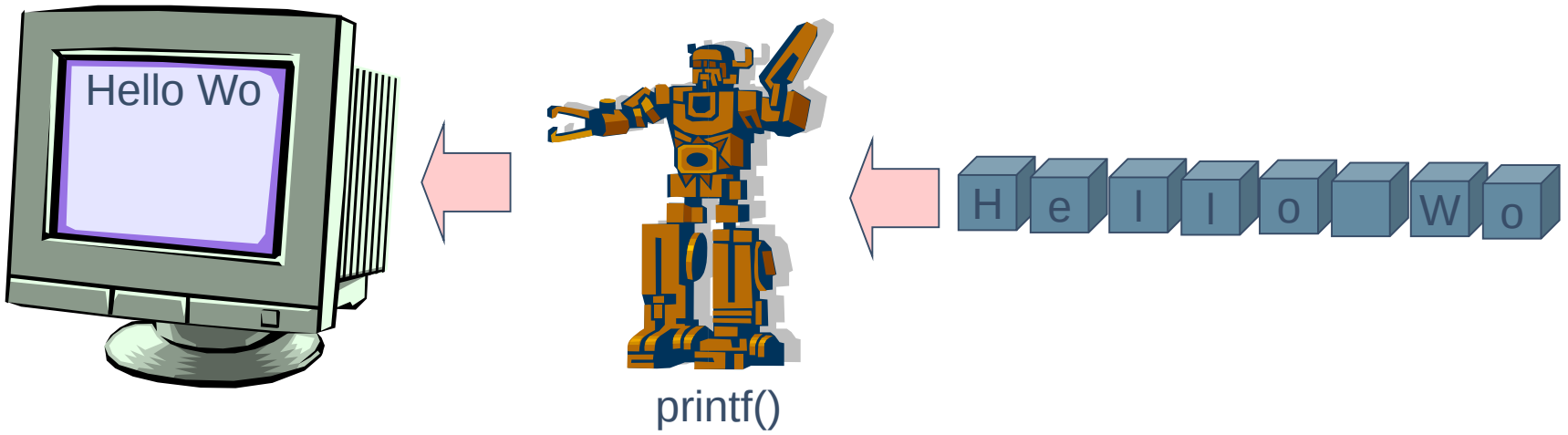
- printf(): 모니터에 출력을 하기 위한 표준 출력 라이브러리 함수



문자열 출력

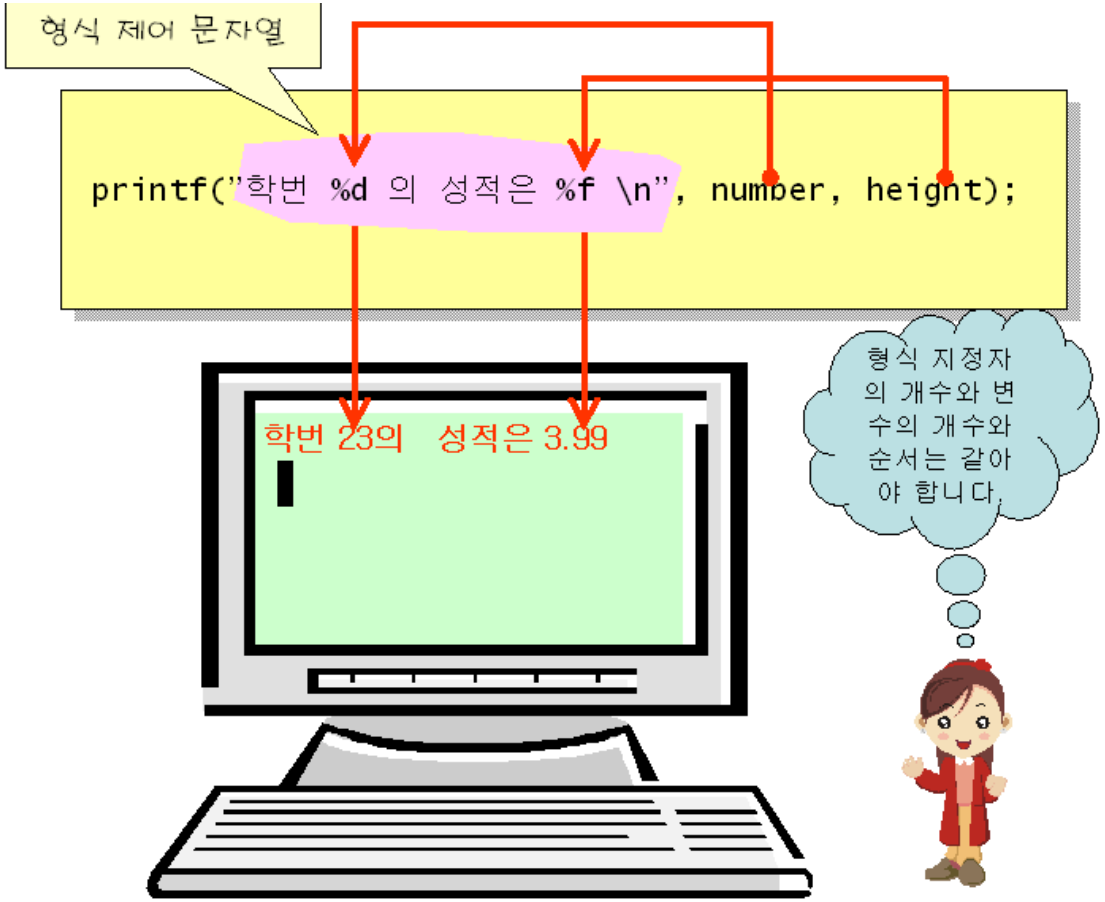
```
printf("Hello World!\n");
```

- 인수(argument): 함수에게 전달하는 데이터
- 문자열(string): 문자들을 여러 개 나열한 것



변수값 출력

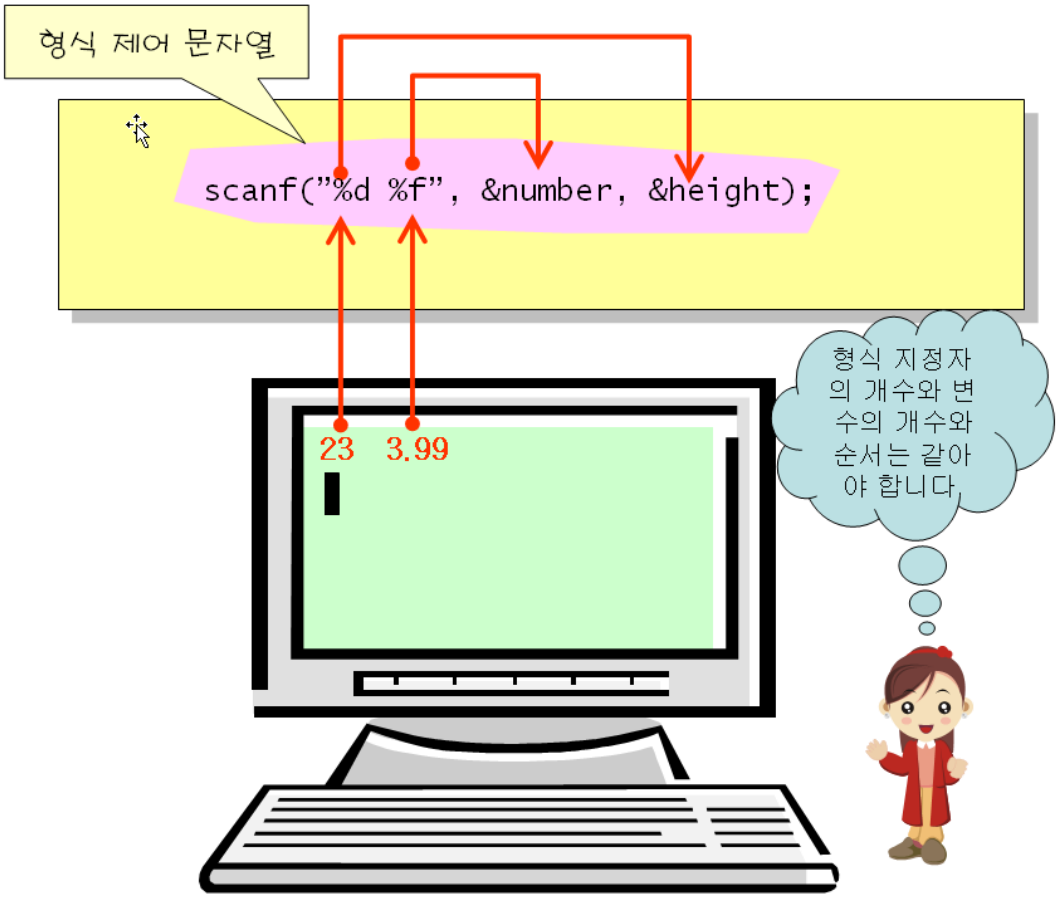
형식 지정자	의미	예
%d	정수를 10진수로 출력한다	1, -2, 10, 20, -100
%f	소수점이 있는 실수로 출력한다.	0.1, 10.1, 3.14
%c	문자 형태로 출력한다.	'a', 'A'
%s	문자열 형태로 출력한다.	"abc", "ABC"



scanf()

- scanf(): 키보드로부터 입력을 하기 위한 라이브러리 함수

```
scanf("%○ %○...", &변수1, &변수2, ...);
```



연봉 계산 프로그램

```c
/* 저축액을 계산하는 프로그램 */
#include <stdio.h>

int main(void)
{
        int salary;         // 월급
        int deposit;        // 저축액

        printf("월급을 입력하시오: ");
        scanf("%d", &salary);

        deposit = 10 * 12 * salary;

        printf("10년 동안의 저축액: %d\n", deposit);

        return 0;
}
```

사용자로부터 월급을 입력받는다.

월급에 10*12를 곱하여 10년동안의 저축액을 계산한다.

결과를 출력한다.

```
월급을 입력하시오: 200
10년 동안의 저축액: 24000
```

원의 면적 프로그램

```c
/* 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;
}
```

원의 면적 계산

```
반지름을 입력하시오: 5.0
원의 면적: 78.500000
```

환율 계산 프로그램

```c
/* 환율을 계산하는 프로그램*/
#include <stdio.h>

int main(void)
{
	float rate;                                          // 원/달러 환율
	float usd;                                           // 달러화
	int krw;                                             // 원화

	printf("달러에 대한 원화 환율을 입력하시오: ");  // 입력 안내 메시지
	scanf("%f", &rate);                                          // 사용자로부터 환율입력

	printf("원화 금액을 입력하시오: ");                // 입력 안내 메시지
	scanf("%d", &krw);                                           // 원화 금액 입력

	usd = krw / rate;                                            // 달러화로 환산

	printf("원화 %d원은 %f달러입니다.\n", krw, usd);  // 계산 결과 출력

	return 0;                                            // 함수 결과값 반환
}
```

```
달러에 대한 원화 환율을 입력하시오: 928.78
원화 금액을 입력하시오: 1000000
원화 1000000원은 1076.681204달러입니다.
```