


컴퓨터프로그래밍의기초|

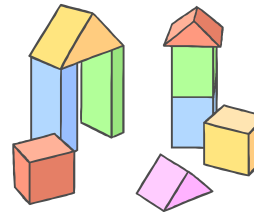
CLASS 2.

제 3장 C프로그램 구성요소



이번 장에서 학습할 내용

- 
- * 주석
 - * 전처리기
 - * 함수
 - * 변수
 - * 출력 함수 printf()
 - * 입력 함수 scanf()
 - * 연산자 - 대입, 산술



이번 장에서는
C프로그램을
이루는
구성요소들을
살펴봅니다.



일반적인 프로그램의 형태

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



데이터 입력



데이터 처리



결과출력

더샘 프로그램

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

주석

- 주석의 종류
 - // 한줄 주석
 - /* 시작과 끝 표기가 있는 주석 – 여러 줄 가능 */
- 주석의 중요성
 - 메모
 - 다른 사람이 프로그램을 보았을 때, 주석이 있다면 훨씬 쉽게 프로그램의 내용을 알 수 있다.
 - 많은 시간이 흘렀다면, 만든 사람이라고 하더라도 내용을 잘 기억할 수 없다.
 - 좋은 주석은 코드를 작성한 의도/의미, 목적을 명확히 나타내는 것.

들여쓰기

- 들여쓰기(indentation)

```
#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...

가독성이 떨어진다.

또한 들여쓰기가 안
되어 있어서 같은 수
준에 있는 문장들을
구분하기 힘듭니다.



전처리기

- **stdio.h**는 표준 입출력에 대한 라이브러리 함수의 정의가 들어 있다. → 링커

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

- 외부 파일을 포함시키라는 의미의 전처리기
- **#**기호로 시작

저처리기

```
/* 첫번째 프로그램 */
```

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

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

hello.c

```
// stdio.h
```

```
...
```

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

```
...
```

stdio.h

장간 점검

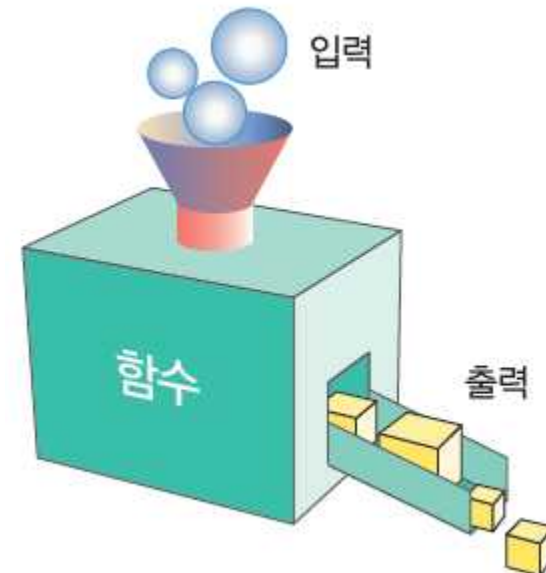
1. `printf()`를 사용하기 위하여 포함시켜야 하는 헤더 파일은 무엇인가?
2. 전처리기 `#include`의 의미는 무엇인가?



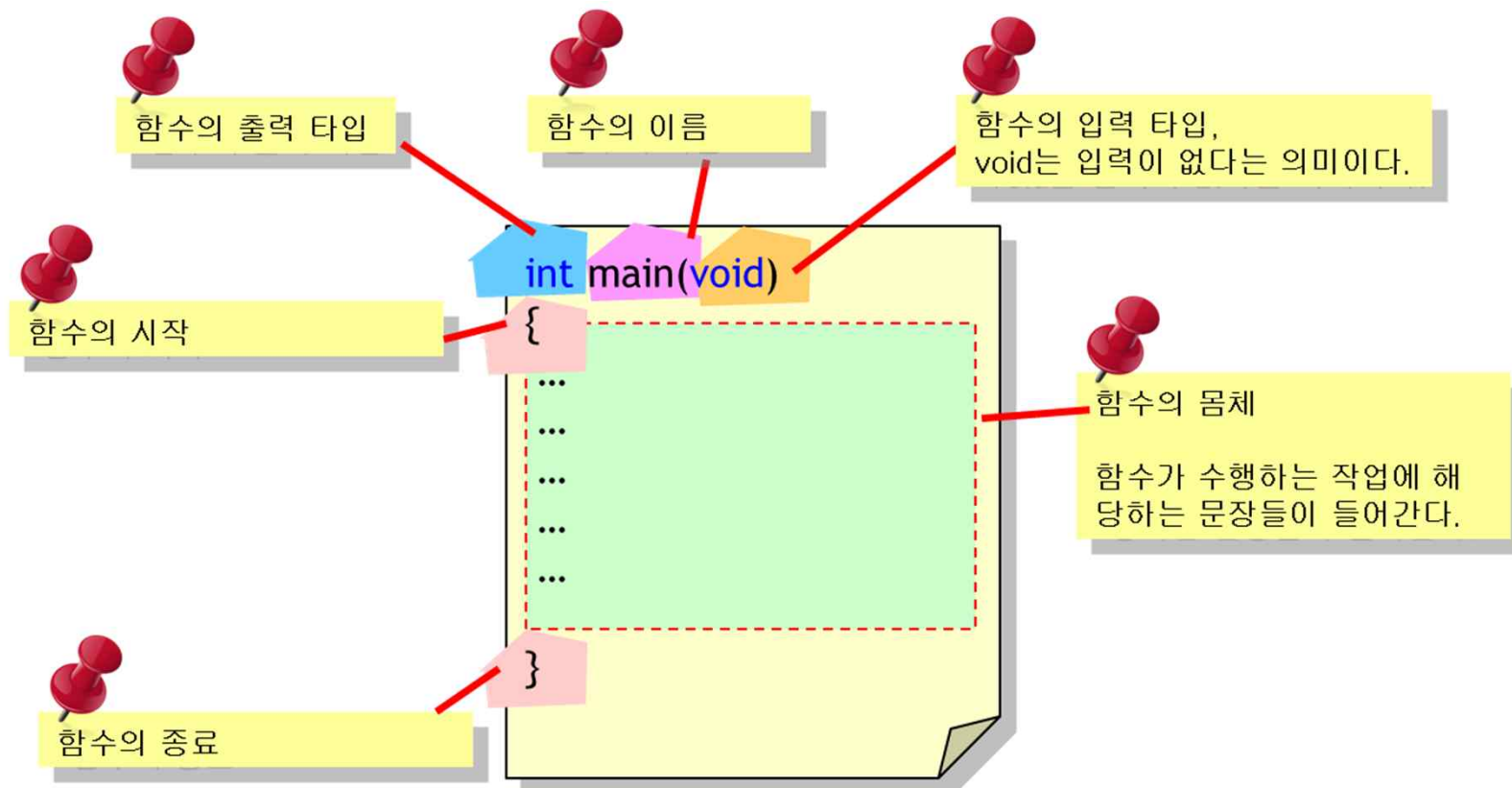
함수

- 함수(function)
 - 특정 기능을 수행하는 처리 단계들을 괄호로 묶어서 이름을 붙인 것
- 함수는 프로그램을 구성하는 기본적인 단위(부품)

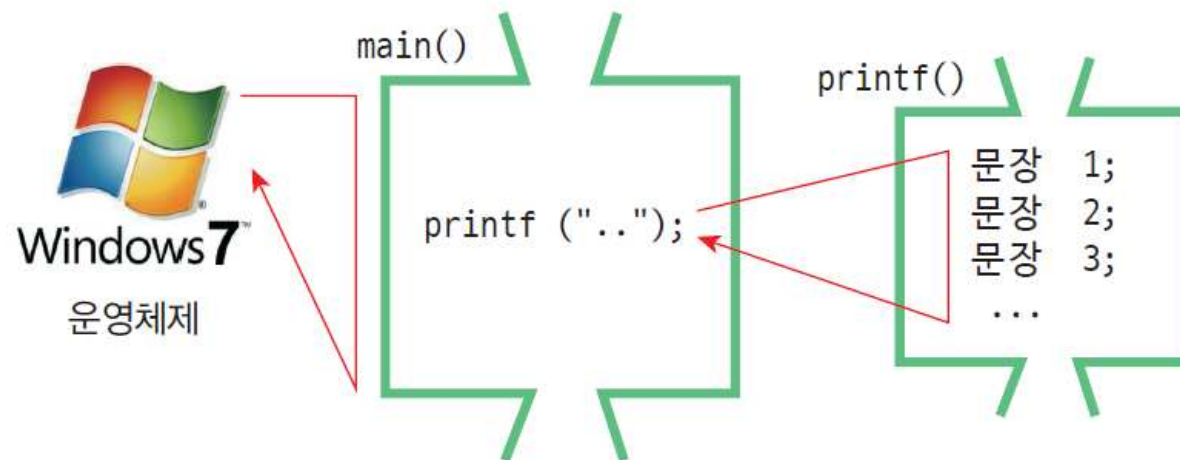
```
int main(void)
{
    ...
    ...
}
```



함수의 구조



main()은 누가 호출할까?



함수를 실행하려면 함수를
호출하면 됩니다.



장간 점검

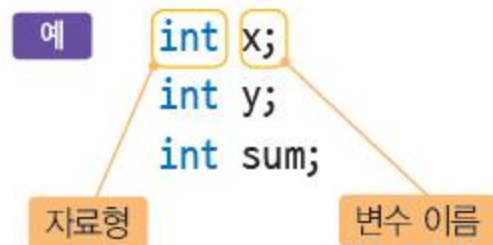
1. 모든 C 프로그램에 반드시 있어야 되는 함수는 무엇인가?
2. 함수의 시작과 끝을 나타내는 기호는 무엇인가?
3. 모든 문장은 어떤 기호로 끝나는가?



변수, variable ***

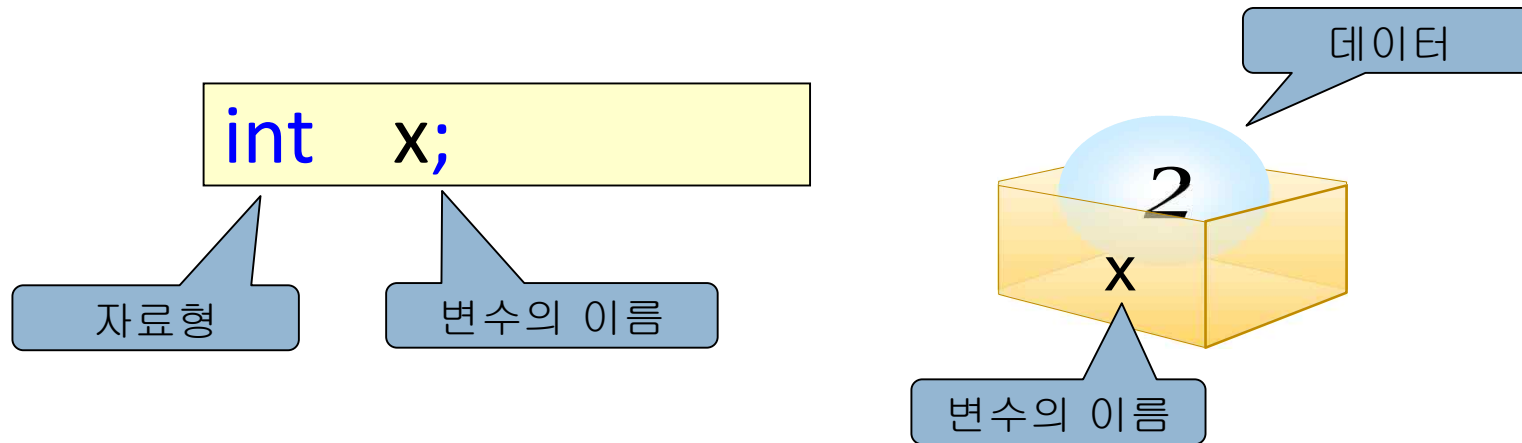
- Variable: 변수 / value: 값 + able(~가능한, ~할 수 있는)
- **변수 = 메모리 공간. 메모리 할당.**
 - 프로그램이 사용하는 데이터를,
 - 일시적으로 저장할 목적으로 사용하는 메모리 공간
- 데이터(data): 수, 값 value

Syntax: 변수 선언



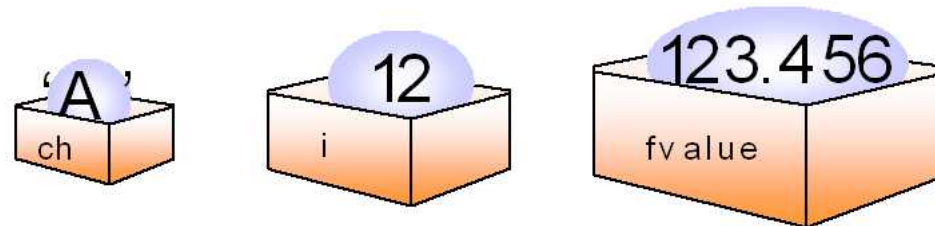
변수는 왜 필요한가?

- 변수는 데이터, 값을 일시적으로 저장하는 역할을 한다.



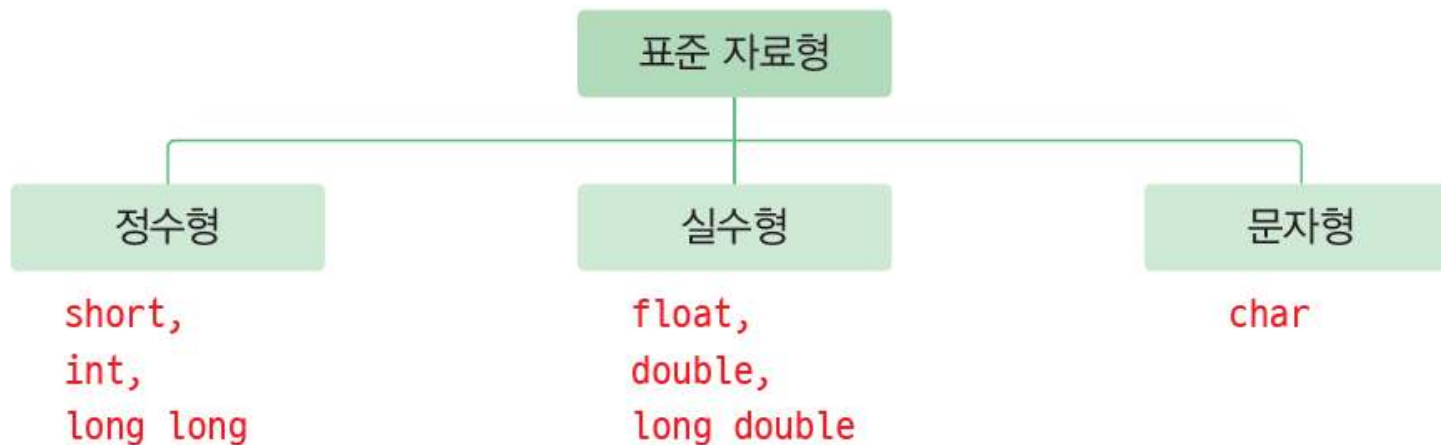
변수의 종류

- 변수에는 데이터의 종류에 따라 여러 가지 타입이 존재한다.
 - → 자료형 **datatype**
 - **size**



자료형, data type ***

- **datatype, 자료형 = data (자료 == variable 변수) + type (형)**
 - 변수가 저장할 데이터가
 - 정수인지 실수인지, 아니면 또 다른 어떤 데이터인지를 지정하는 것이다.
- 자료형 type에는
 - 정수형, 부동소수점형(실수형), 문자형이 있다.

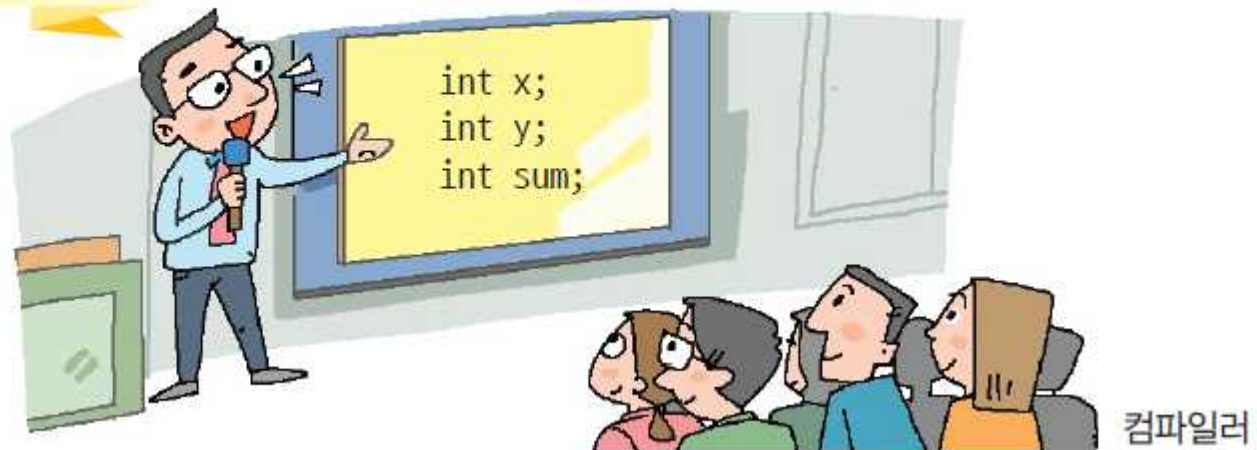


변수 선언

- 변수 선언

- 변수 생성, 변수 만드는것. To create a variable.
- 컴파일러에게 어떤 타입의 변수가 사용되는지를 미리 알리는 것

지금부터 이 프로그램에서 사용될
변수들을 소개하겠습니다.



변수 선언

자료형은 정수형

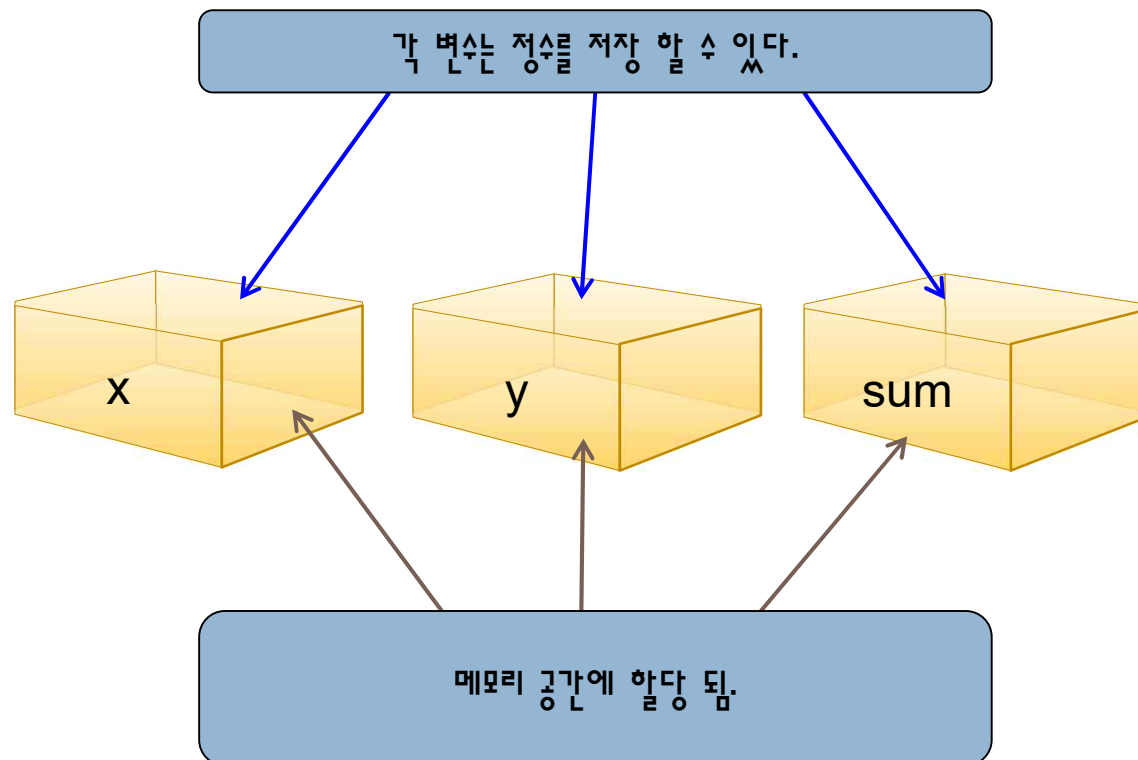
변수 이름

```
int x; // 첫 번째 정수를 저장하는 변수  
int y; // 두 번째 정수를 저장하는 변수  
int sum; // 두 정수의 합을 저장하는 변수
```

변수 선언

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

```
int x, y, sum;  //(같은 자료형끼리는) 한줄에 여러 개의 변수선언 가능
```



장갑점검 중심점검

1. `double`형 변수 `f`를 선언하는 문장을 작성하여 보자.
2. 변수와 자료형을 영어로 하면 무엇인가?
3. 자료형의 `type`에는 무엇이 있는가?
4. 변수 선언은 함수의 어떤 위치에서 하여야 하는가?



변수의 이름

- 변수이름(식별자) 만드는 규칙
 - 식별자는 영문자와 숫자, 밑줄 문자 _로 이루어진다.
 - 중간에 공백이 들어가면 안 된다.
 - 첫 글자는 반드시 영문자 또는 밑줄 기호 _이어야 한다.
 - 숫자로 시작할 수 없다.
 - 대문자와 소문자는 구별된다.
 - 따라서 변수 index와 Index, INDEX은 모두 서로 다른 변수이다.
 - C언어의 키워드(예약어)는 똑같이 쓸 수 없다.
 - int ; char ; ...

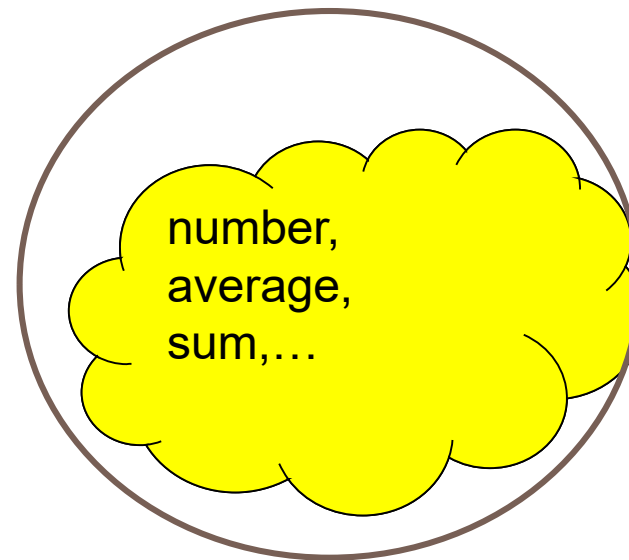
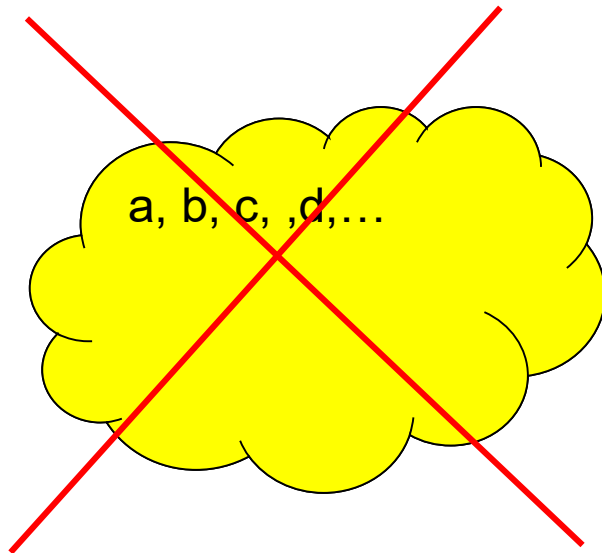
변수의 이름

- `sum` // 영문 알파벳 문자로 시작
- `_count` // 밑줄 문자로 시작할 수 있다.
- `number_of_pictures` // 중간에 밑줄 문자를 넣을 수 있다.
- `King3` // 맨 처음이 아니라면 숫자도 넣을 수 있다.

- `2nd_base(X)` // 숫자로 시작할 수 없다.
- `money#` // #과 같은 기호는 사용할 수 없다.
- `double` // `double`은 C 언어의 키워드이다.

좋은 변수 이름

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



키워드

- 키워드(keyword)
 - C언어에서 고유한 의미를 가지고 있는 특별한 단어
- 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

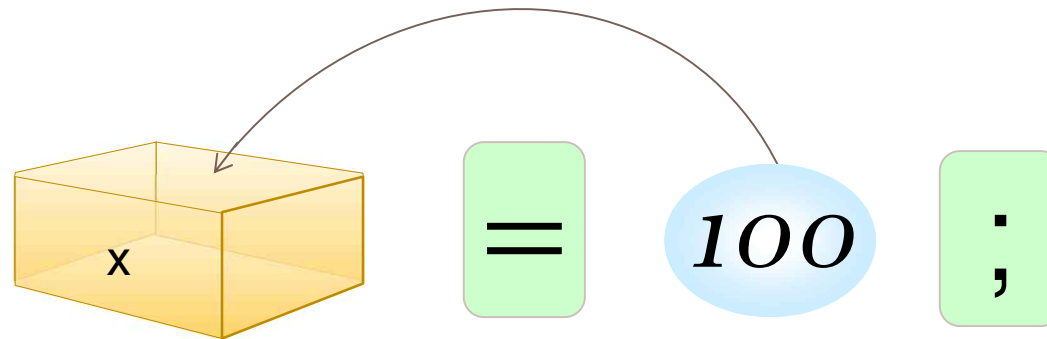
변수의 초기화 (초기값)

- 변수에 초기값을 줄 수 있다.
 - 변수의 초기화란 곧, 변수에 변수 선언시에 초기값을 주는 것.
 - `int x = 10;`
 - `int y = 20;`
 - `int sum = 0;`
- 동일한 타입의 변수인 경우, 같은 줄에서 선언과 동시에 변수들을 초기화할 수 있다.
 - `int width = 100, height = 200;`
- 다음과 같이 초기화하는 것은 문법적으로는 오류가 아니지만 피하는 것이 좋다.
 - `int width, height = 200;`

변수에 값 저장하기

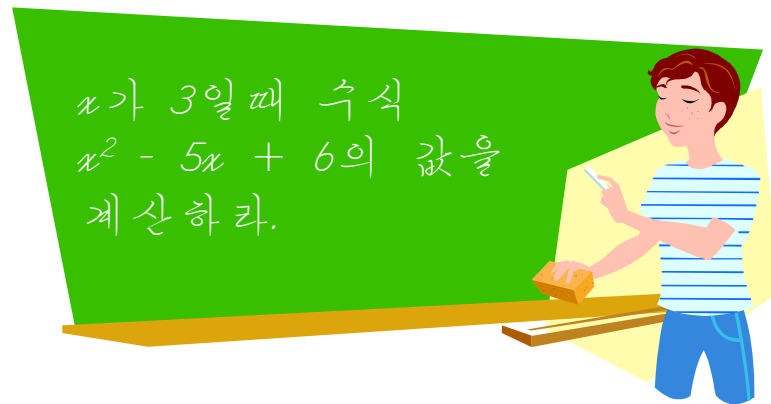
```
x = 100;
```

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



수식

- 수식(expression): 피연산자와 연산자로 구성된 식
 - Ex) 방정식, 논리식
- 수식은 결과값을 가진다.



x 가 3일 때 수식
 $x^2 - 5x + 6$ 의 값을
계산하라.



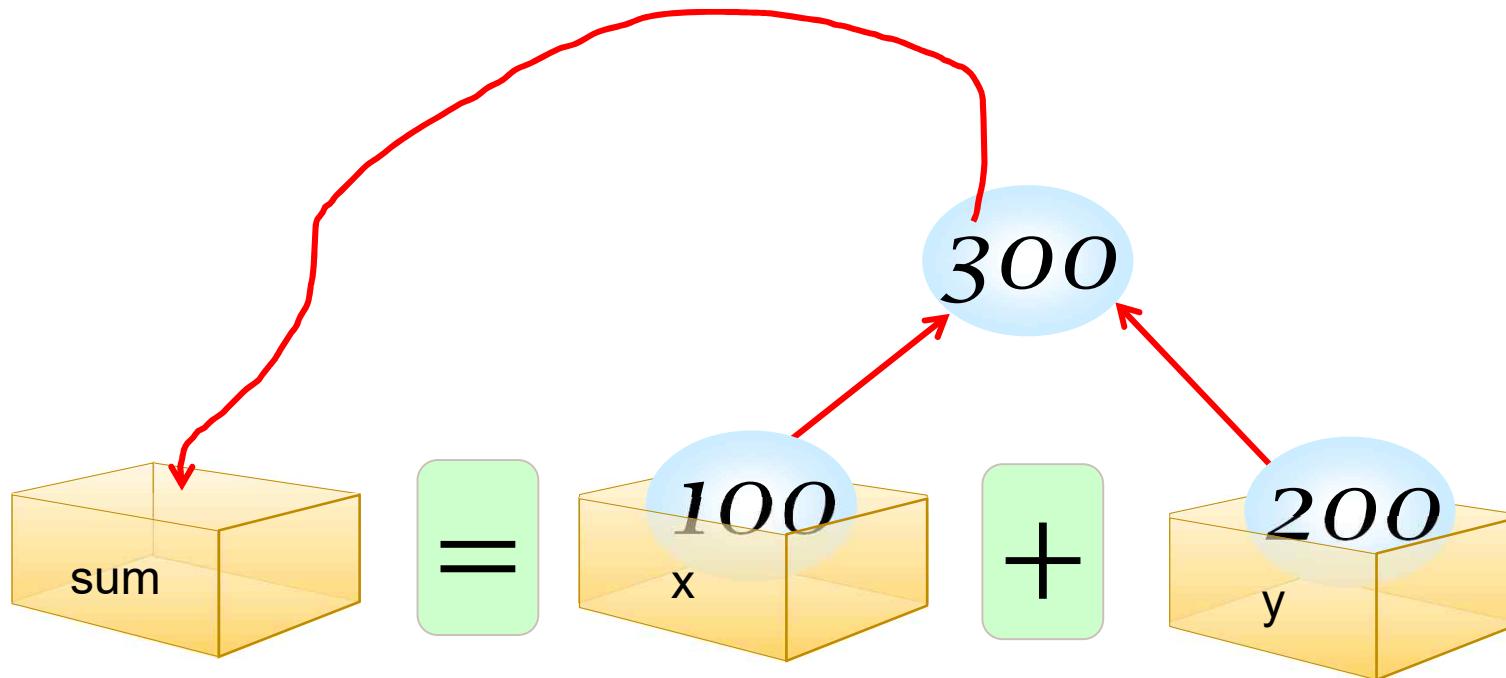
```
int x, y;  
  
x = 3;  
y = x * x - 5 * x + 6;  
printf("%d\n", y);
```

산술 연산

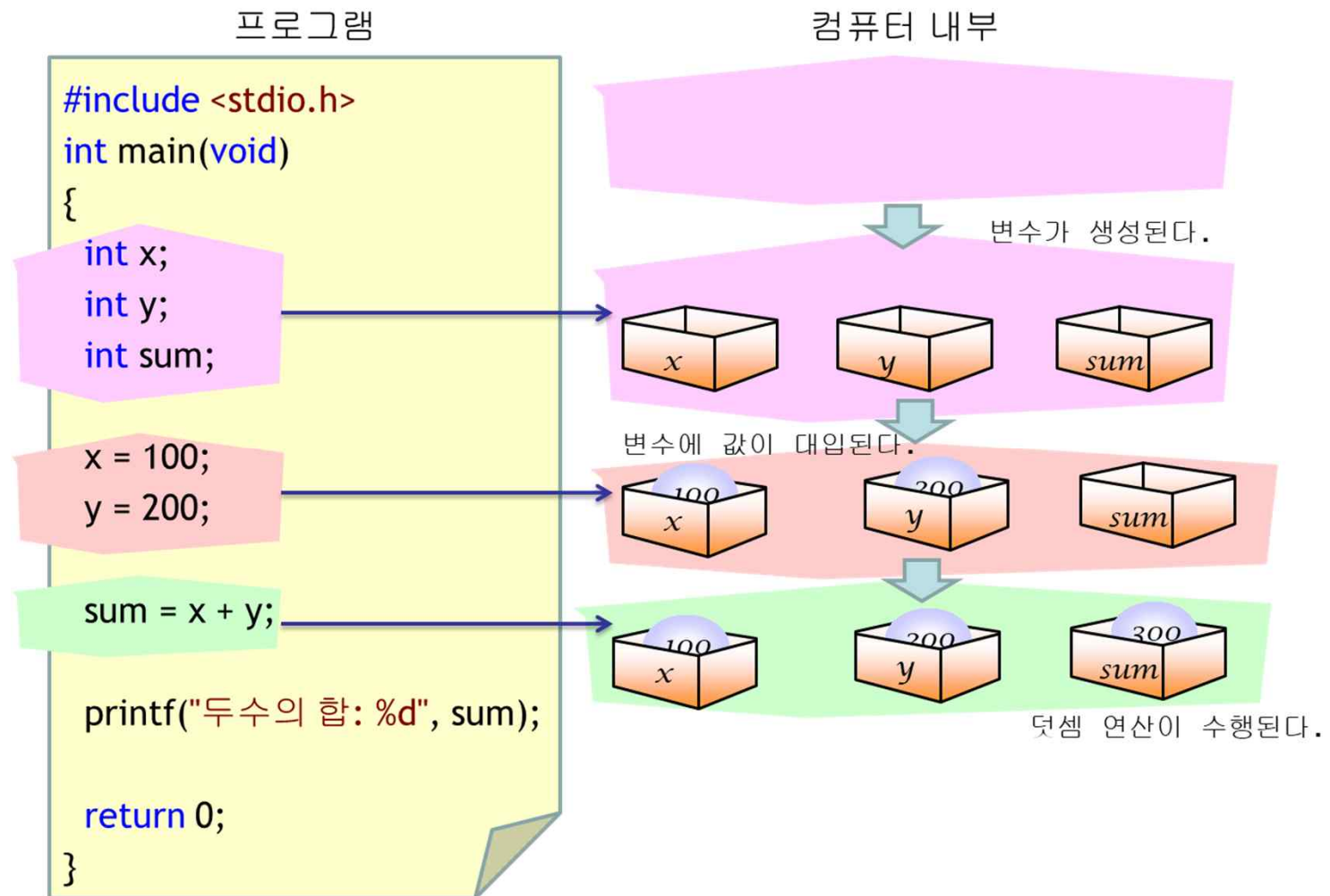
연산	연산자	C 수식	수학에서의 기호
덧셈	+	$x + y$	$x + y$
뺄셈	-	$x - y$	$x - y$
곱셈	*	$x * y$	xy
나눗셈	/	x / y	x/y 또는 $\frac{x}{y}$ 또는 $x \div y$
나머지	%	$x \% y$	$x \bmod y$

산술 연산

sum = x + y;

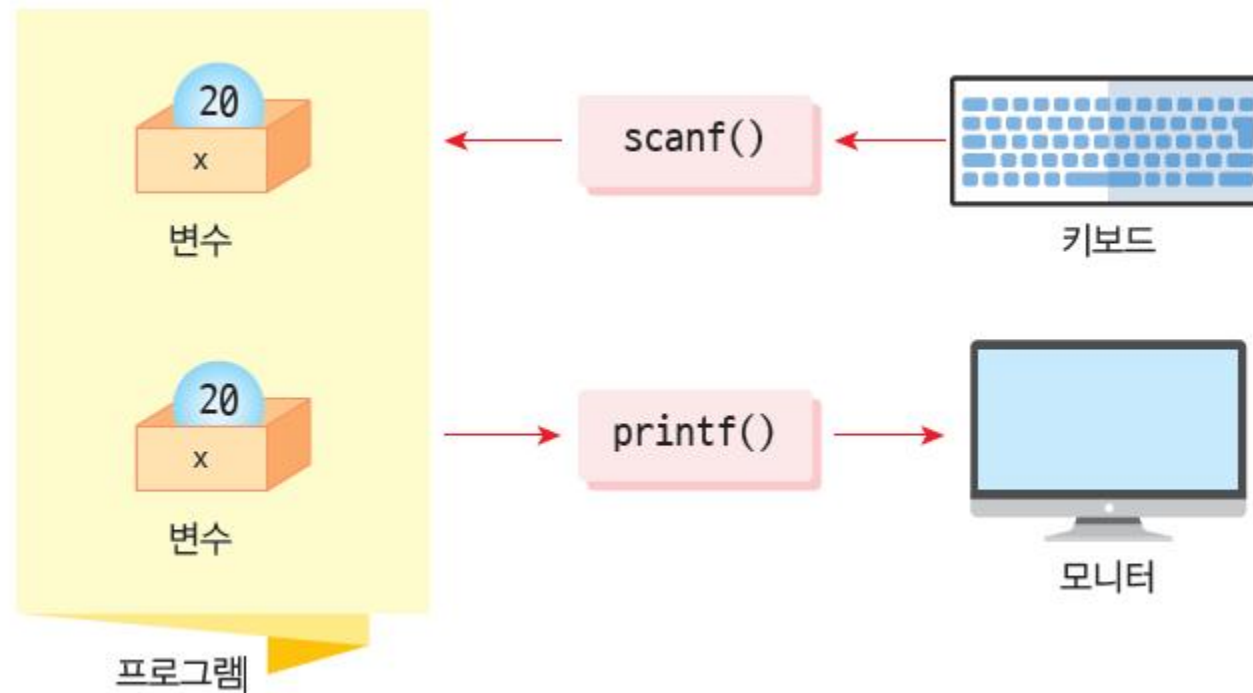


정리



printf()

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



문자열 출력

- 문자열(string): “Hello World!\n”와 같이 문자(char)를 여러 개 나열한 것.

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



변수가 출력

```
printf("두수의 합: %d", sum);
```

sum의 값이 정수형으로 출력된다.

두수의 합: 30

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 변수가 출력

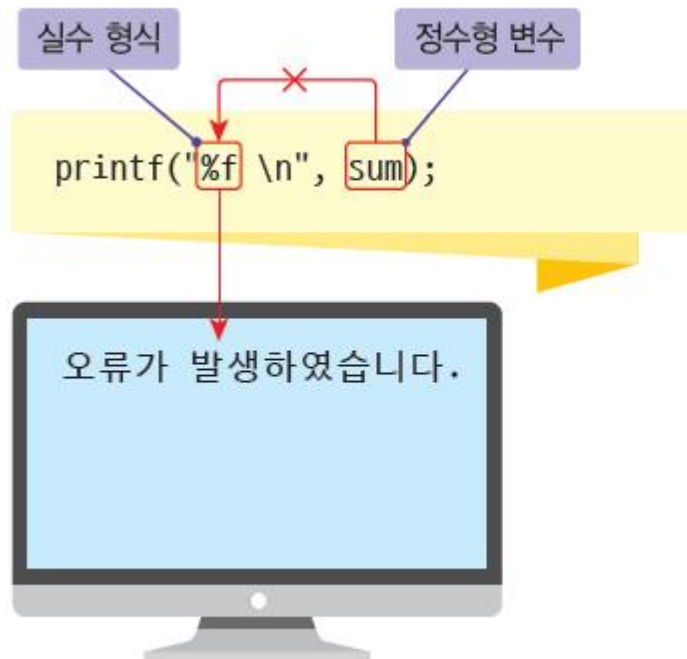
형식제어 문자열

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

23 3.99

형식 지정자의 개수와
변수의 개수와 순서는
같아야 합니다.

주의!



형식 지정자와 변수의
타입은 같아야 합니다.

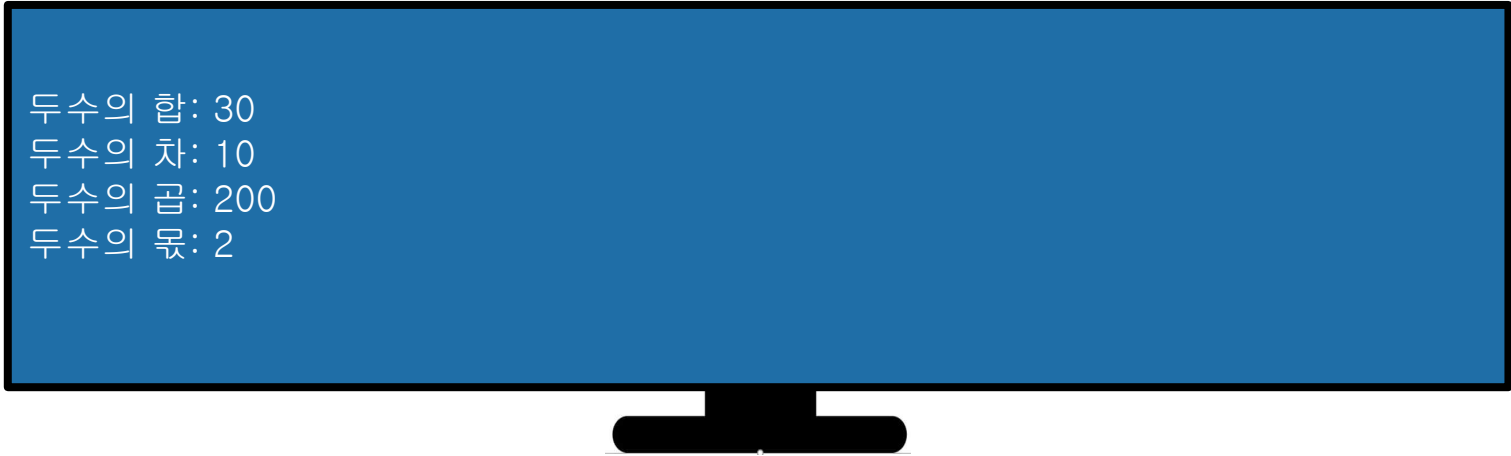
장간 점검

1. `printf()`에서 변수의 값을 실수 형태로 출력할 때 사용하는 형식 지정자는 무엇인가?
2. `printf()`를 사용하여 정수형 변수 `k`의 값을 출력하는 문장을 작성하여 보자.



Lab: 사칙 연산

- 변수 x 와 y 에 20과 10을 저장하고 $x+y$, $x-y$, $x*y$, x/y 을 계산하여서 변수에 저장하고 이들 변수를 화면에 출력하는 프로그램을 작성해보자.



두수의 합: 30
두수의 차: 10
두수의 곱: 200
두수의 몫: 2

Solution

```
// 정수 간의 가감승제를 계산하는 프로그램
#include <stdio.h>

int main(void)
{
    int x;           // 첫 번째 정수를 저장할 변수
    int y;           // 두 번째 정수를 저장할 변수
    int sum, diff, mul, div; // 두 정수 간의 연산의 결과를 저장하는 변수

    x = 20;          // 변수 x에 20을 저장
    y = 10;          // 변수 y에 10을 저장

    sum = x + y;      // 변수 sum에 (x+y)의 결과를 저장
    diff = x - y;     // 변수 diff에 (x-y)의 결과를 저장
    mul = x * y;      // 변수 mul에 (x*y)의 결과를 저장
    div = x / y;      // 변수 div에 (x/y)의 결과를 저장
```

Solution

```
printf("두수의 합: %d\n", sum);           // 변수 sum의 값을 화면에 출력
printf("두수의 차: %d\n", diff); // 변수 diff의 값을 화면에 출력
printf("두수의 곱: %d\n", mul);          // 변수 mul의 값을 화면에 출력
printf("두수의 몫: %d\n", div); // 변수 div의 값을 화면에 출력

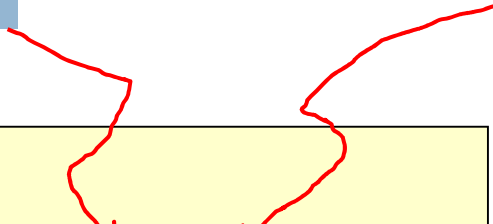
return 0;
}
```

scanf() ***

- 키보드로부터 값을 받아서 변수에 저장한다.
- 변수의 주소를 필요로 한다.
- **&int_a: 주소 라는 뜻!**
- ***int_a: 값이라는 뜻!**

• 형식 지정자

• 값을 저장할 변수의 주소

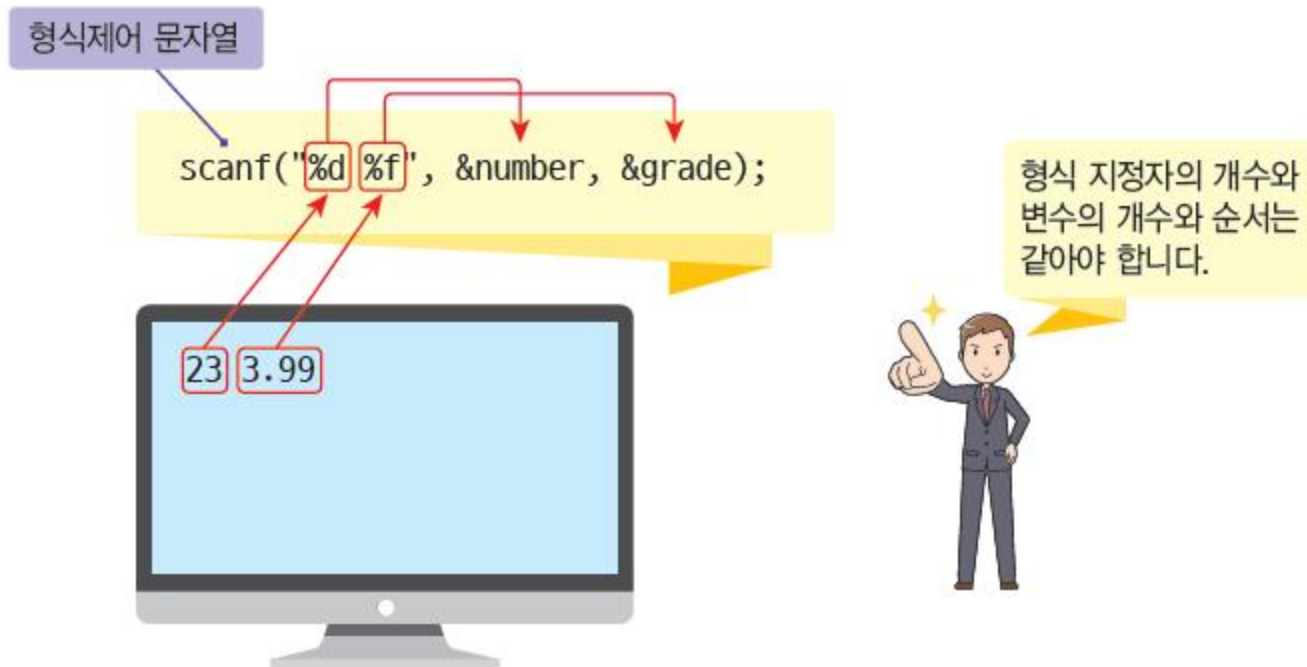


```
scanf("%d", &x);
```

형식 지정자

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

scanf()



정수를 받아들이는 프로그램

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

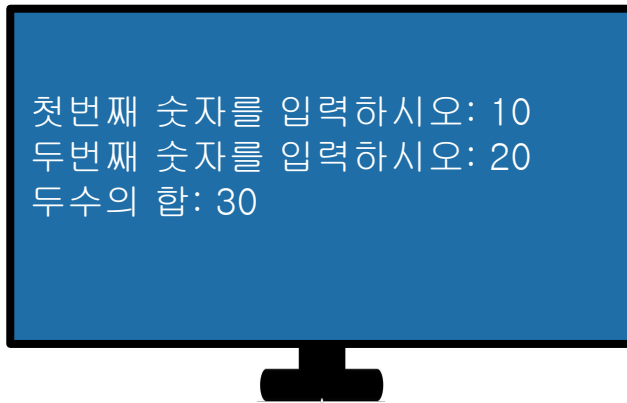
```
int main(void)  
{  
    int x;                // 정수를 저장할 변수  
    printf("정수를 입력하시오: ");  
    scanf("%d", &x);  
    printf("입력된 정수 = %d \n", x);  
    return 0;  
}
```

만약 scanf() 오류가 발생하면 소스 파일의
처음에서
_CRT_SECURE_NO_WARNINGS를
정의해준다.

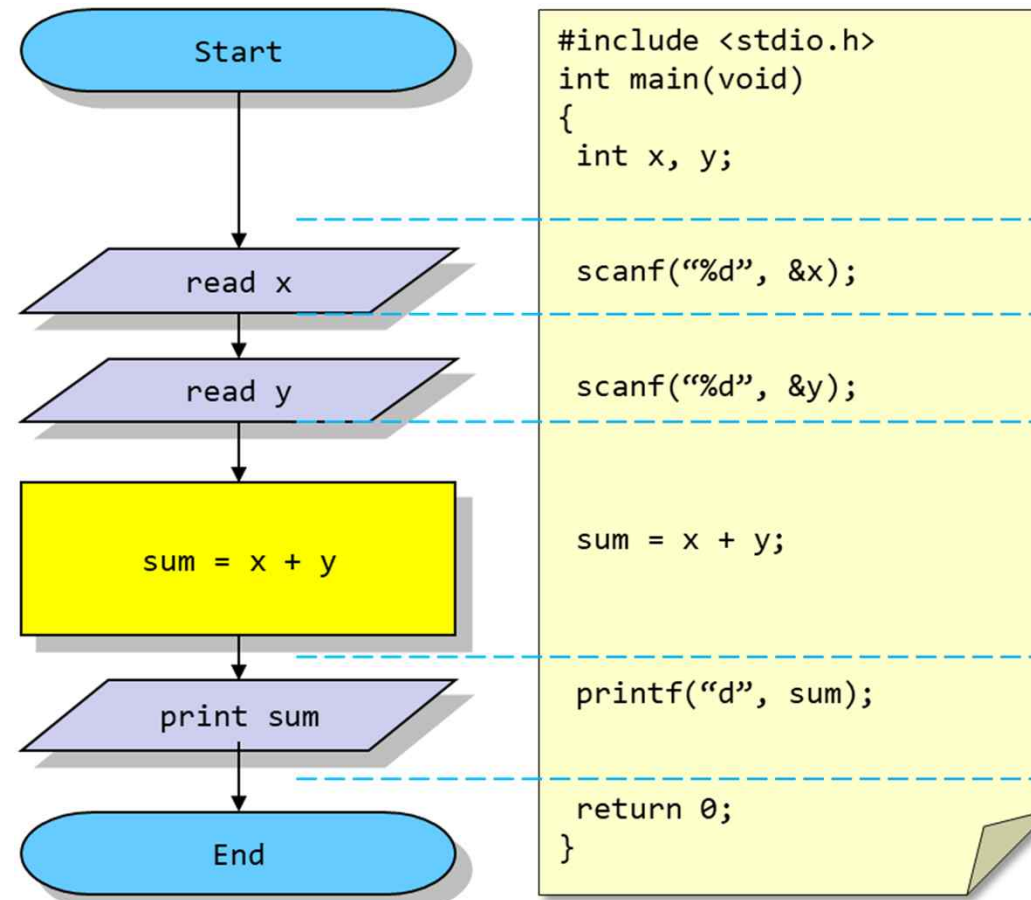
```
Microsoft Visual Studio 디버그 콘솔  
정수를 입력하시오: 20  
입력된 정수 = 20  
C:\Users\chun\source\repos\Project8\Debug\Project8.exe(28548 프로세스)이(가) 0 코드로 인해 종료되었습니다.  
이 창을 닫으려면 아무 키나 누르세요.
```

더샘 프로그램 #2

- 사용자로부터 입력을 받아보자.



알고리즘



두번째 더셈 프로그램

```
// 사용자로부터 입력받은 2개의 정수의 합을 계산하여 출력
#define _CRT_SECURE_NO_WARNINGS

#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 입력하시오:"); // 입력 안내 메시지 출력
    scanf("%d", &x);                // 하나의 정수를 받아서 x에 저장

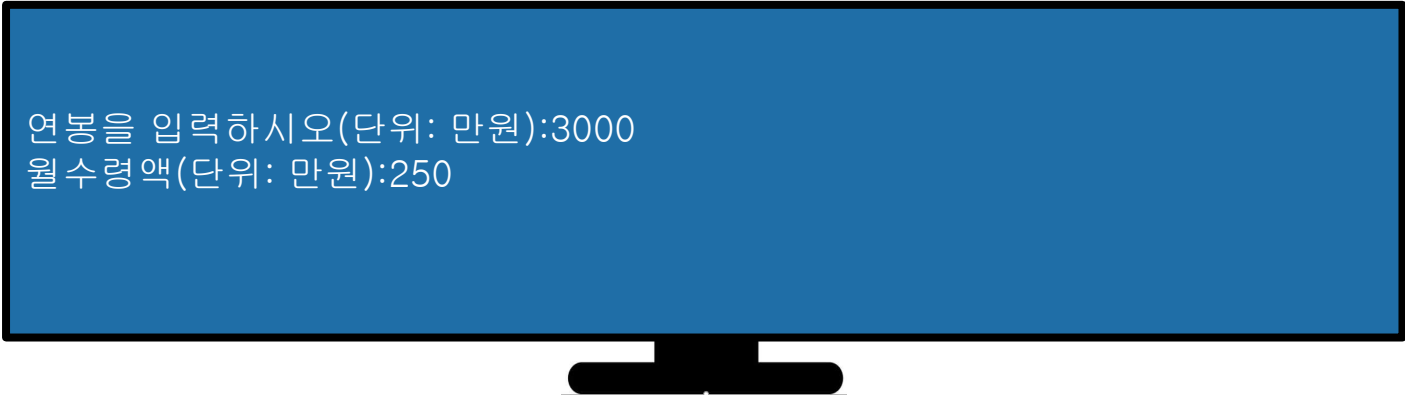
    printf("두번째 숫자를 입력하시오:"); // 입력 안내 메시지 출력
    scanf("%d", &y);                // 하나의 정수를 받아서 x에 저장

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

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

연봉 계산기 프로그램

- 사용자가 연봉을 입력하면 월 수령액을 계산하는 프로그램을 작성해 보자.



연봉을 입력하시오(단위: 만원):3000
월수령액(단위: 만원):250

두번째 더셈 프로그램

```
#define _CRT_SECURE_NO_WARNINGS
/* 월수령액을 계산하는 프로그램 */
#include <stdio.h>

int main(void)
{
    int ysalary; // 연봉
    int msalary; // 월수령액


    printf("연봉을 입력하시오(단위: 만원): ");
    scanf("%d", &ysalary);

    msalary = ysalary / 12;
    printf("월수령액(단위: 만원): %d\n", msalary);

    return 0;
}
```

원의 면적 계산 프로그램

- 사용자로부터 원의 반지름을 입력받고 이 원의 면적을 구한 다음, 화면에 출력한다.



반지름을 입력하시오: 10.0
원의 면적: 314.000000

원의 면적 계산 프로그램

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

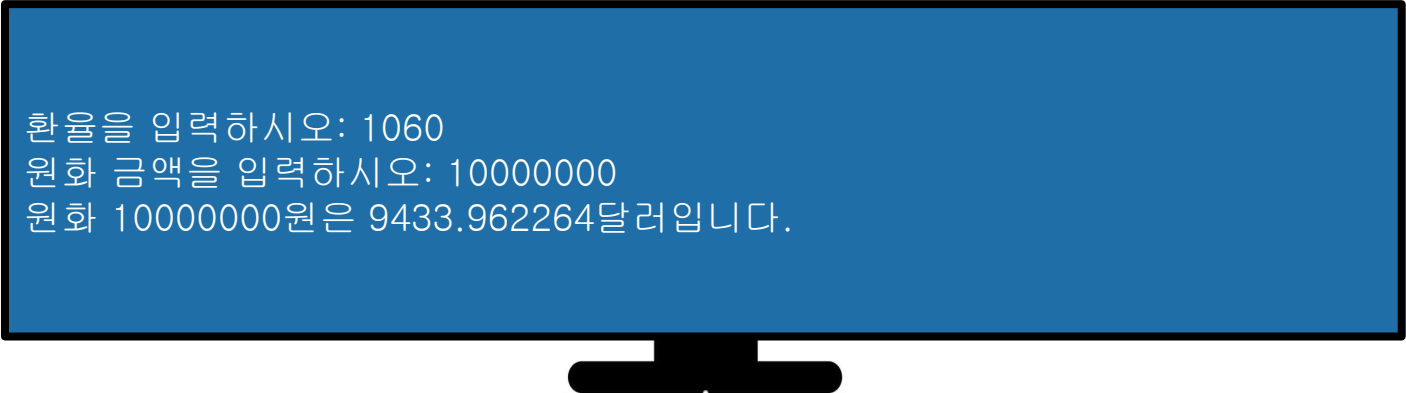
int main(void)
{
    float radius; // 원의 반지름
    float area; // 면적

    printf("반지름을 입력하시오 : ");
    scanf("%f", &radius);
    area = 3.14 * radius * radius;
    printf("원의 면적 : %f\n", area);

    return 0;
}
```

환율 계산 프로그램

- 사용자가 입력하는 원화를 달러화로 계산하여 출력하는 프로그램은 작성하여 보자.



환율을 입력하시오: 1060
원화 금액을 입력하시오: 10000000
원화 10000000원은 9433.962264달러입니다.

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

int main(void)
{
    double rate; // 원/달러 환율
    double usd; // 달러화
    int krw; // 원화는 정수형 변수로 선언

    printf("환율을 입력하시오: "); // 입력 안내 메시지
    scanf("%lf", &rate); // 사용자로부터 환율입력

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

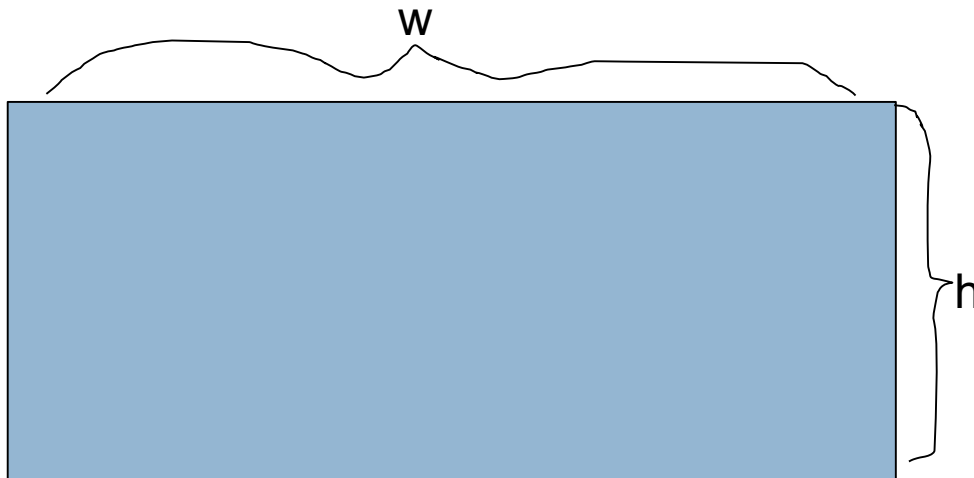
    usd = krw / rate; // 달러화로 환산

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

    return 0; // 함수 결과값 반환
}
```

Mini Project: 사각형의 둘레와 면적

- 가로(w) 10.0, 높이(h) 5.0 인 사각형의 둘레와 면적을 구하시오.
 - 필요한 변수는 w, h, area, perimeter라고 하자.
 - 변수의 자료형은 실수를 저장할 수 있는 double형으로 하자.
 - $area = w * h$;
 - $perimeter = 2 * (w + h)$;



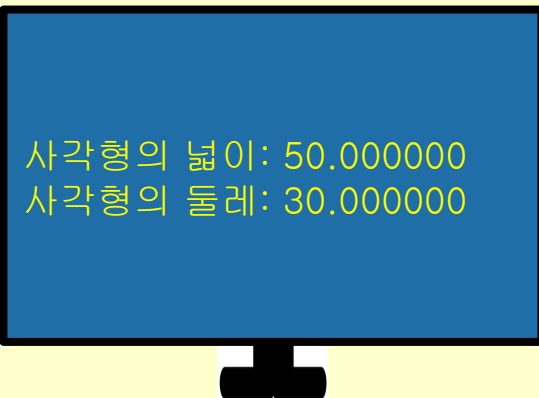

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
```

```
int main(void)
{
    double w;
    double h;
    double area;
    double perimeter;

    w = 10.0;
    h = 5.0;
    area = w * h;
    perimeter = 2 * (w + h);

    printf("사각형의 넓이: %lf\n", area);
    printf("사각형의 둘레: %lf\n", perimeter);
    return 0;
}
```

사각형의
사각형의



Q & A

- In conclusion,
 - 전처리기
 - 함수
 - 변수
 - printf(), scanf()
 - 연산자 – 대입, 산술