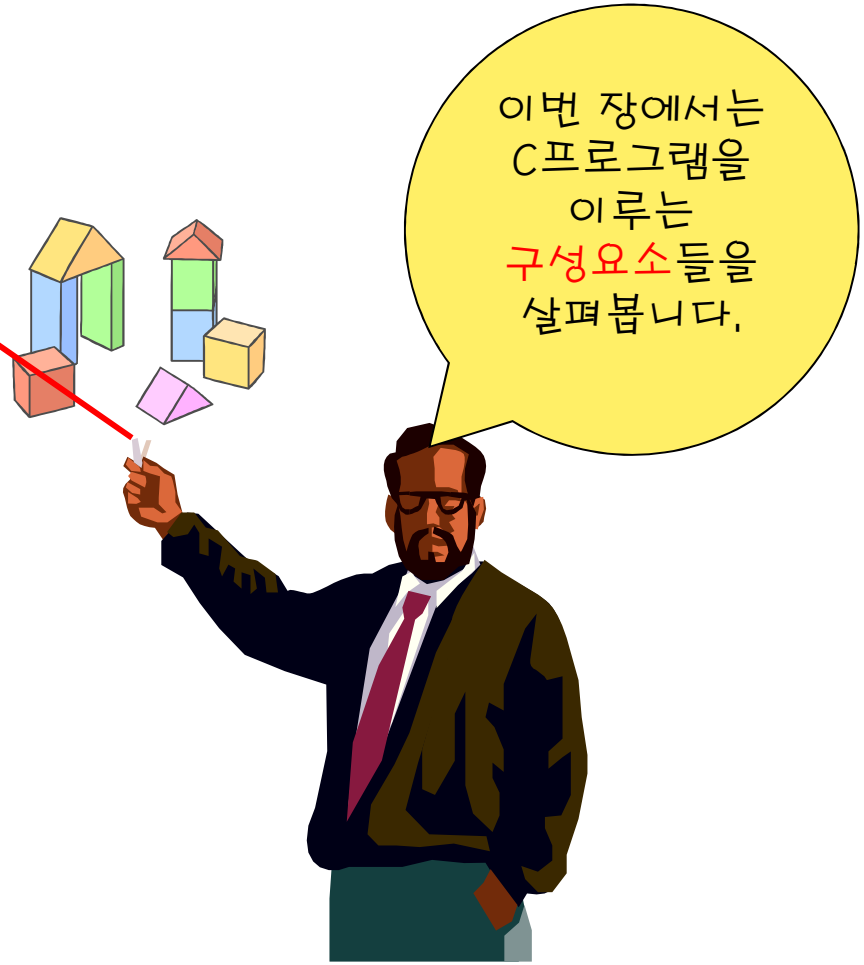
누구나 즐기는 C언어 콘서트

제2장 기초 사항

이번 장에서 학습할 내용

- 첫번째 프로그램 설명
- 화면 출력
- 연산이 있는 프로그램
- 입력이 있는 프로그램
- 오류 수정 및 디버깅
- 응용 프로그램



첫번째 프로그램

hello.c

```
/* 첫번째 프로그램 */
#include <stdio.h>

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

주석

헤더파일 포함

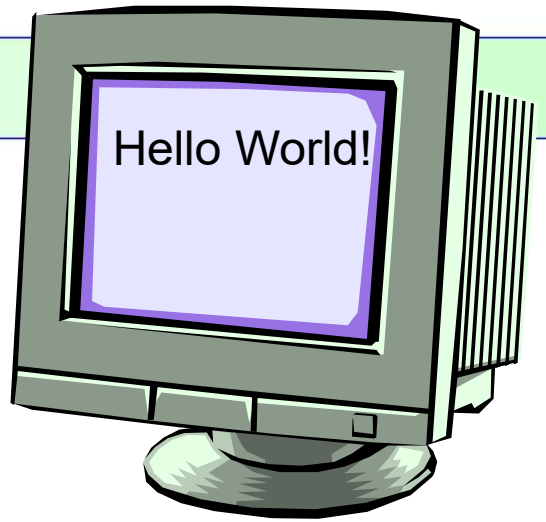
함수의 시작

실행되는 문장

함수의 종료

실행결과

```
Hello World!
```



주석

- 주석(comment): 프로그램에 대한 설명

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

hello.c

주석은 안내 도우미와 같다.

3가지 방법의 주석

- /* 한줄로 된 주석 */

- /* 여러
줄로
된 주석 */

- // 여기서부터 줄의 끝까지 주석

헤더 파일 포함

형식	#include <헤더파일>
예	#include <stdio.h>
	#include <memory.h>

- #include는 소스 코드 안에 특정 파일을 현재의 위치에 포함
- 헤더 파일(header file): 컴파일러가 필요로 하는 정보를 가지고 있는 파일
- stdio.h: standard input output header file
- 주의!: 전처리기 지시자 문장 끝에는 세미콜론을 붙이면 안 된다.

헤더 파일 포함

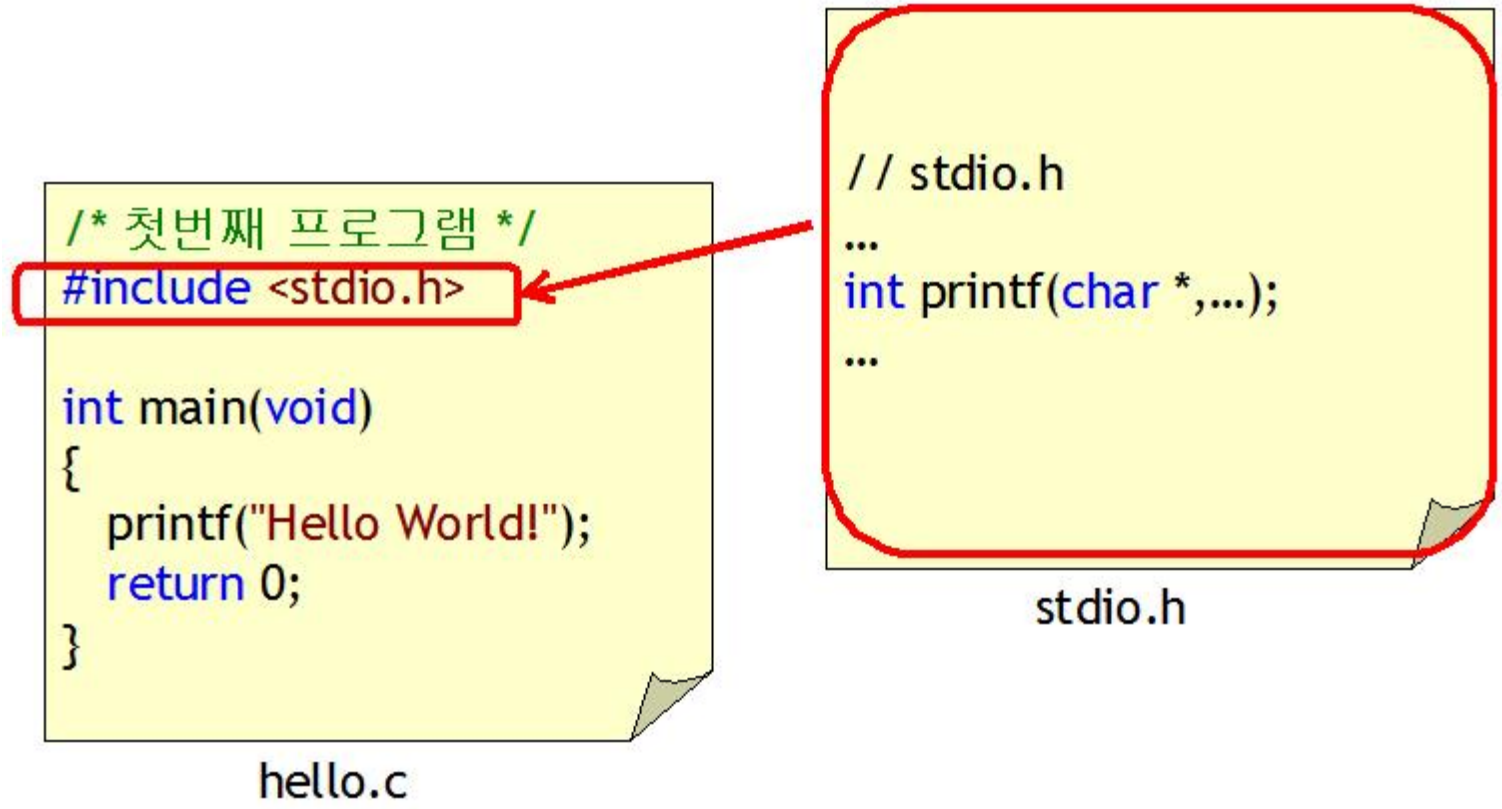
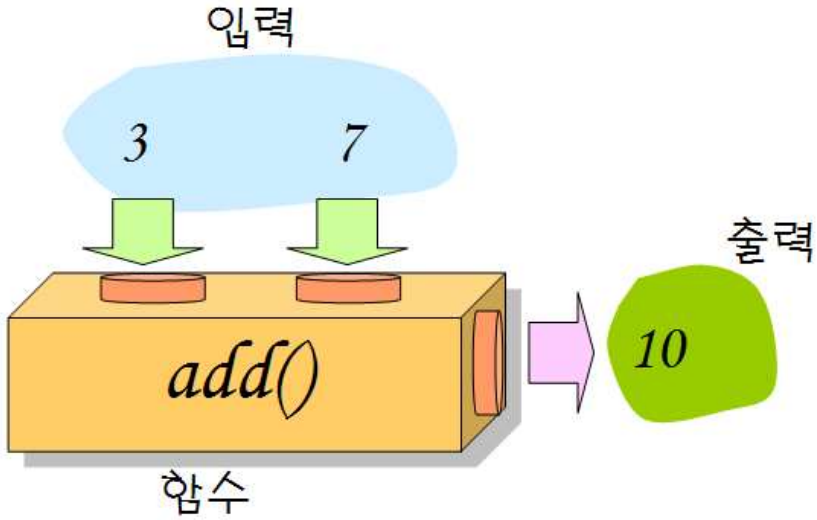


그림 2.2 헤더 파일이 #include 위치에 삽입된다.

줄바꿈 및 들여쓰기

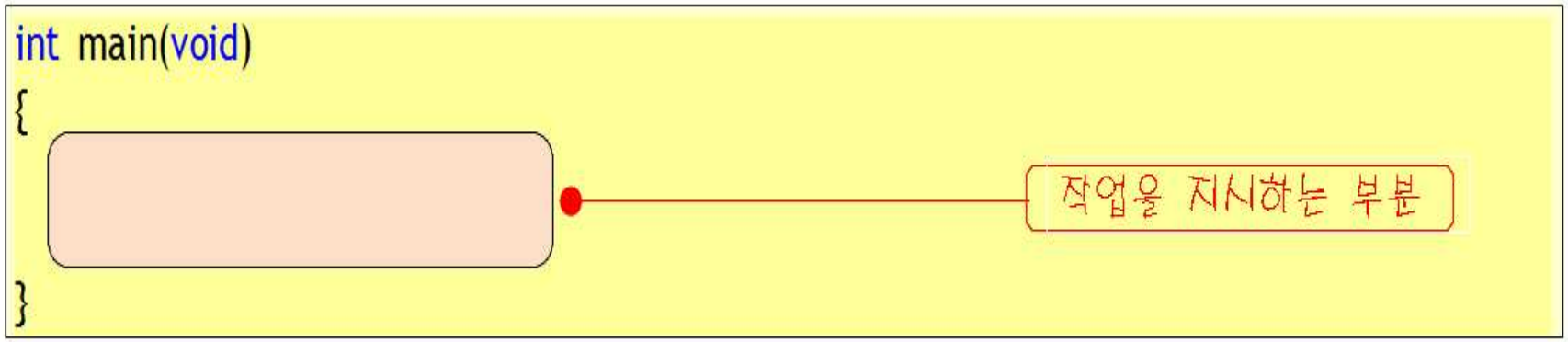
함수

- 함수(function): 특정한 작업을 수행하기 위하여 작성된 독립적인 코드



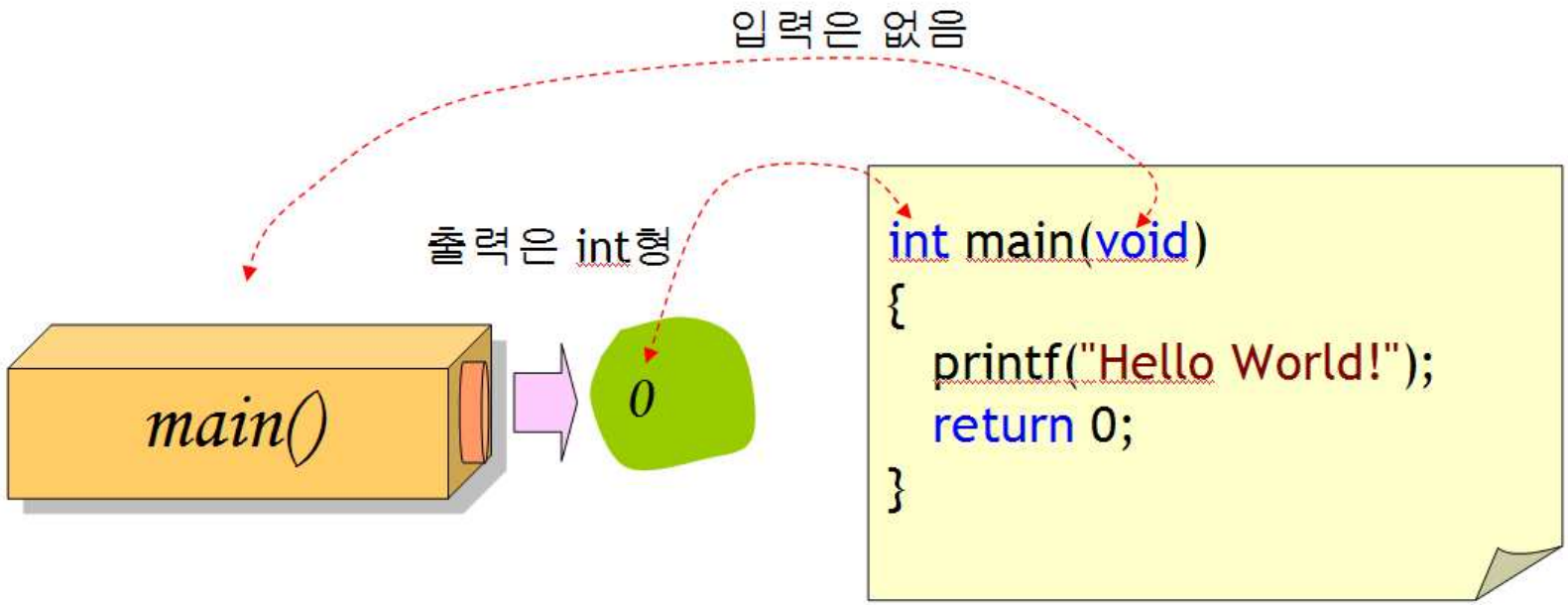
함수

- 작업을 수행하는 문장은 함수 안에 들어가야 함



main() 함수

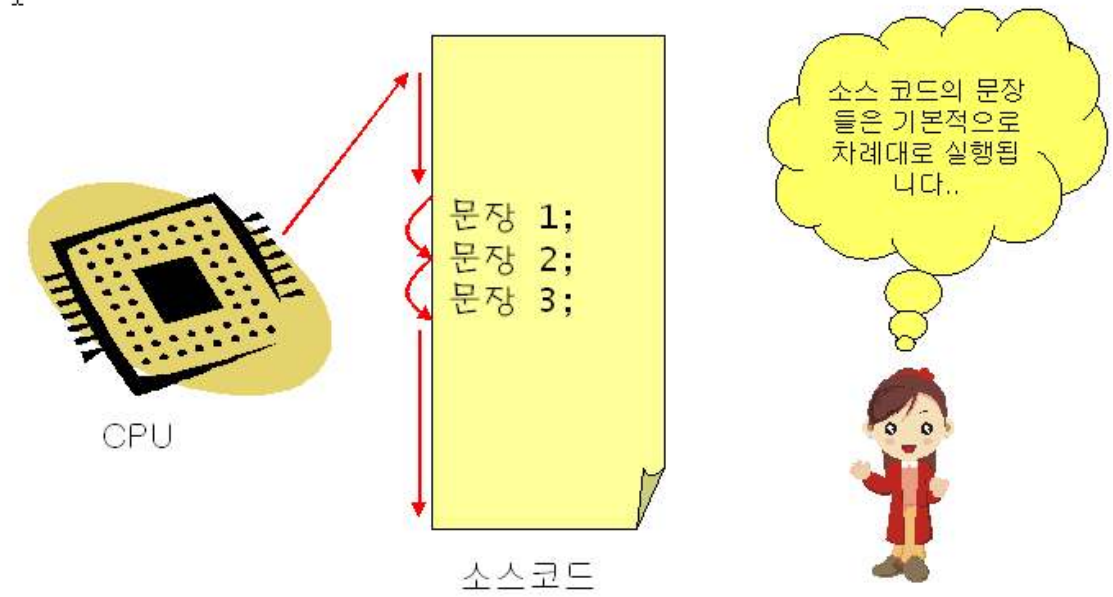
- main() 함수: C 프로그램에서 가장 먼저 실행되는 함수



함수의 구성요소

문장

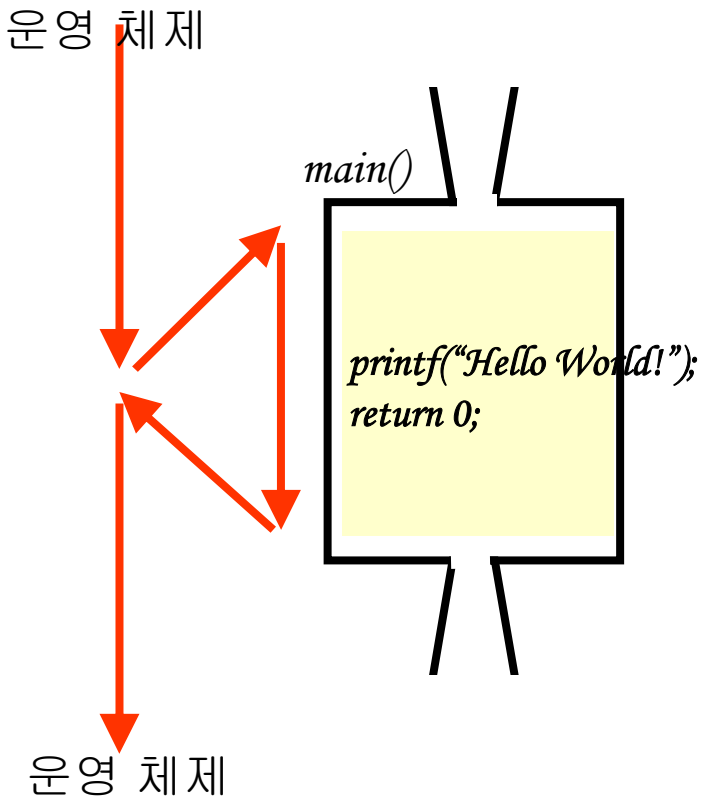
- 함수는 여러 개의 문장으로 이루어진다.
- 문장들은 순차적으로 실행된다.
- 문장은 ;(세미콜른)으로 끝나야 한다.



함수 반환문

```
return 0;
```

- return은 함수의 결과값을 외부로 반환합니다.

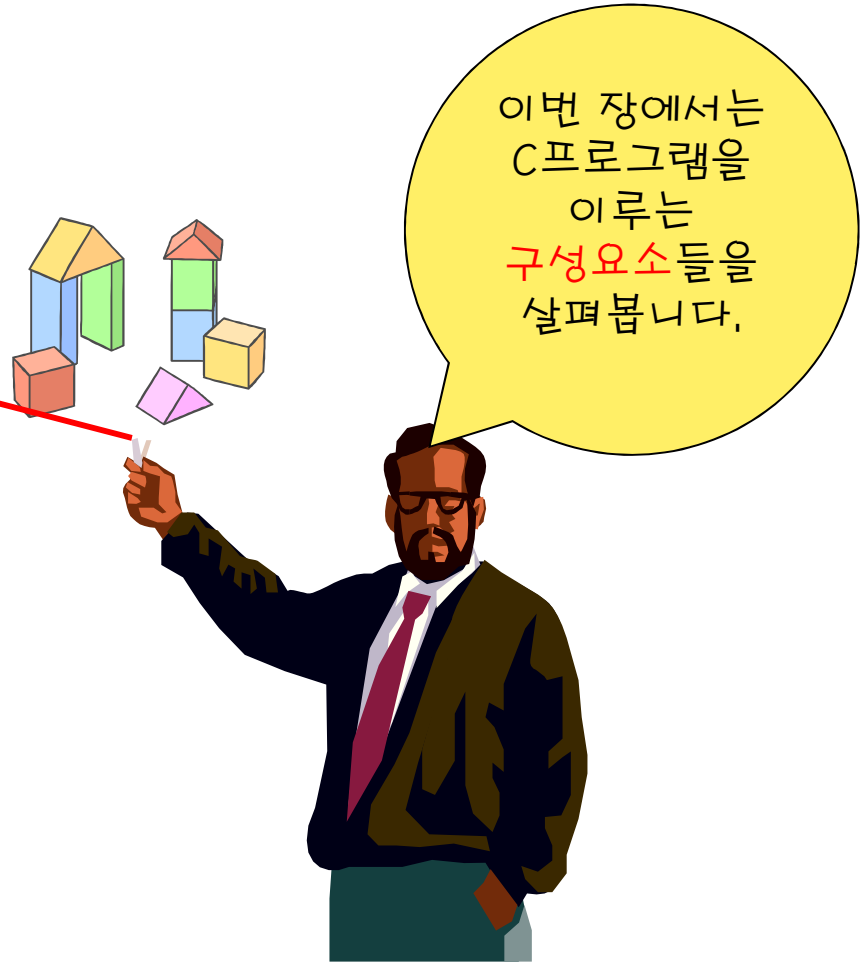


중간 점검

1. 주석이 하는 역할은 무엇인가?
2. 주석을 여러 줄로 하려면 어떤 스타일을 사용하여야 하는가?
3. #include 문은 어떤 동작을 하는가?
4. 모든 문장의 끝에 있어야 하는 기호는?

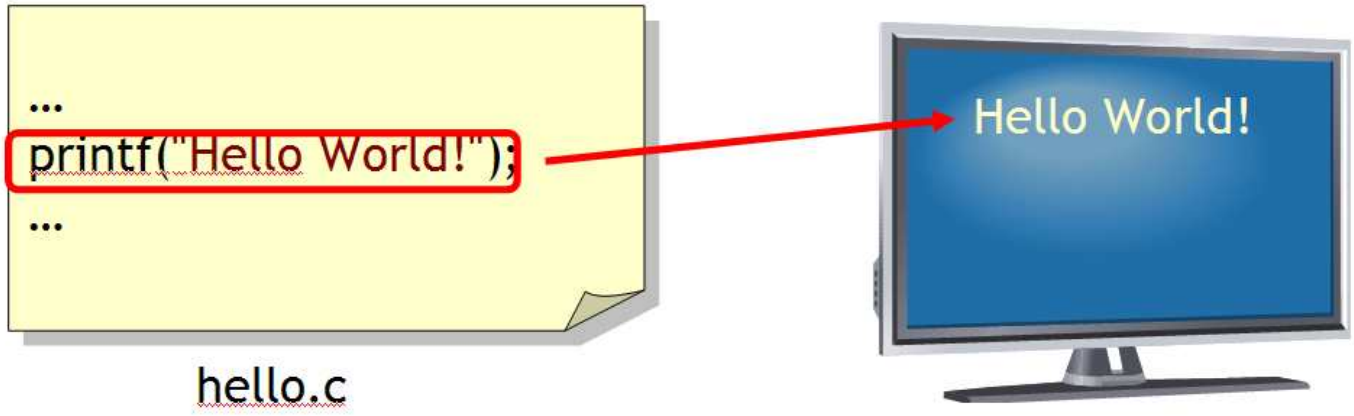
이번에 학습할 내용

- 첫번째 프로그램 설명
- 화면 출력
- 연산이 있는 프로그램
- 입력이 있는 프로그램
- 오류 수정 및 디버깅
- 응용 프로그램



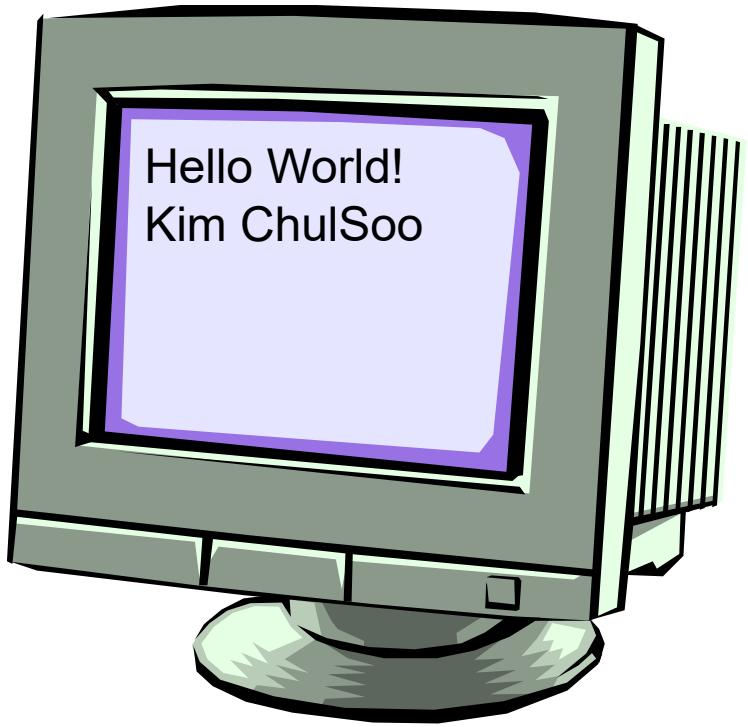
출력 함수 printf()

- printf()는 컴파일러가 제공하는 함수로서 출력을 담당합니다.
- printf()는 큰따옴표 안의 문자열을 화면에 출력합니다.



응용 프로그램 #1

- 다음과 같은 출력을 가지는 프로그램을 제작하여 보자.



첫번째 버전

- 문장들은 순차적으로 실행된다는 사실 이용

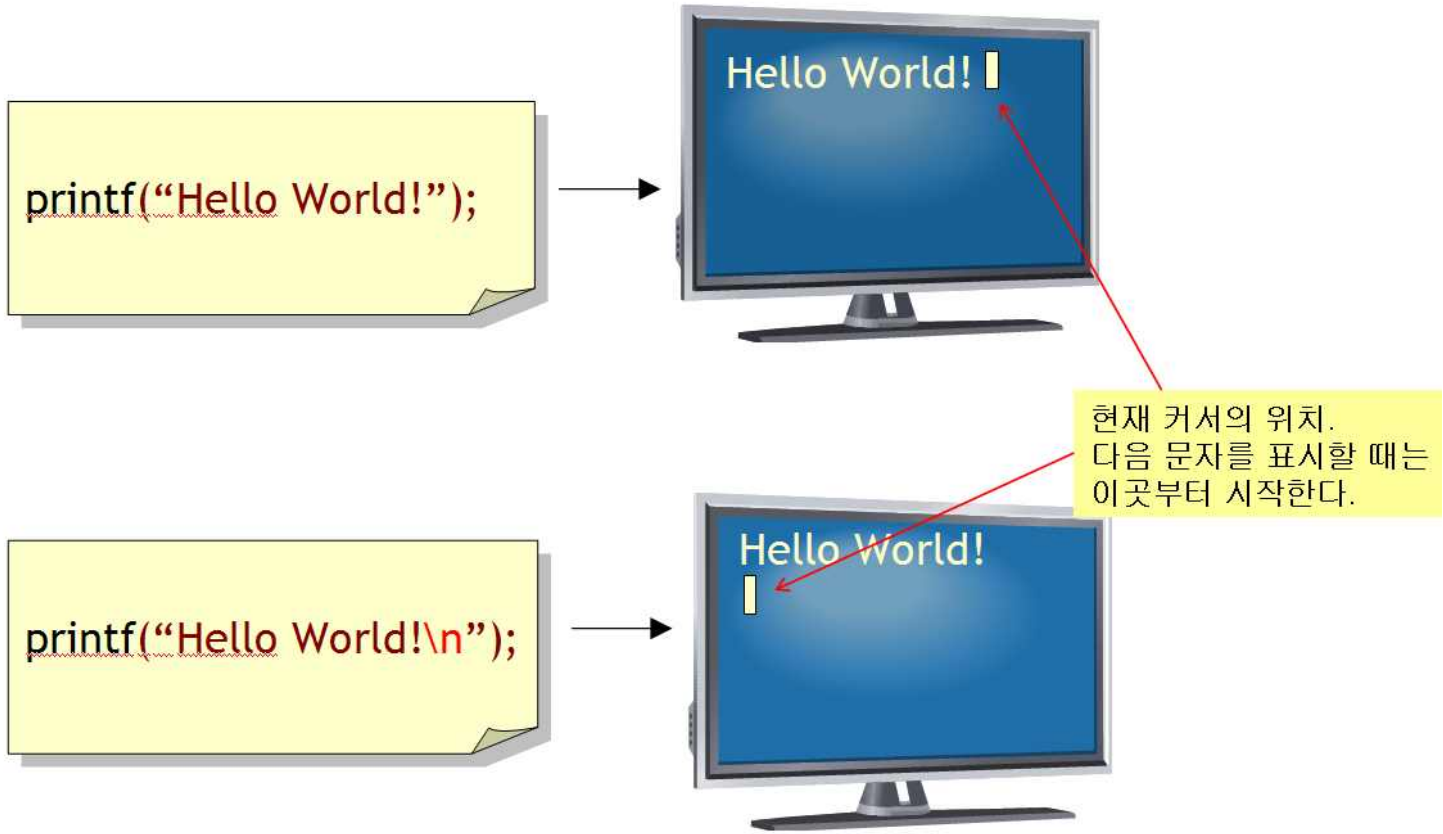
```
/* 첫번째 프로그램의 응용 */
#include <stdio.h>
int main(void)
{
        printf("Hello World!");
        printf("Kim ChulSoo");
        return 0;
}
```

```
Hello World!Kim ChulSoo
```

우리가
원하는
결과가 아님!

줄바꿈 문자 \n

- 줄바꿈 문자인 **\n**은 화면에서 커서를 다음줄로 이동시킨다.



변경된 프로그램

- 줄바꿈 문자를 포함하면 우리가 원하던 결과가 된다.

```
/* 첫번째 프로그램의 응용 */
#include <stdio.h>
int main(void)
{
        printf("Hello World!\n");
        printf("Kim ChulSoo\n");
        return 0;
}
```

```
Hello World!
Kim ChulSoo
```

구구단 출력 프로그램

- 구구단의 일부를 출력

```
#include <stdio.h>
int main(void)
{
        printf("3 X 1 = 3\n");
        printf("3 X 2 = 6\n");
        printf("3 X 3 = 9\n");
        return 0;
}
```

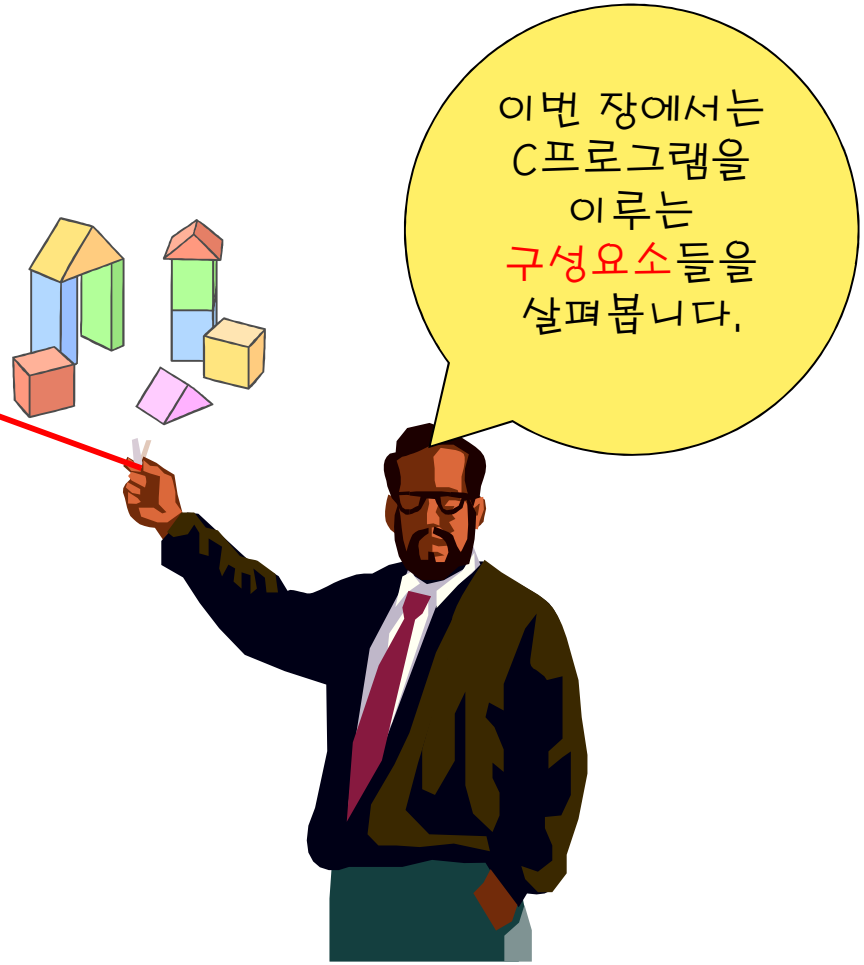
```
3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
```

중간 점검

1. 줄바꿈 문자인 "\n"이 하는 역할은 무엇인가?
2. main() 함수 안의 문장들은 어떤 순서대로 실행되는가?

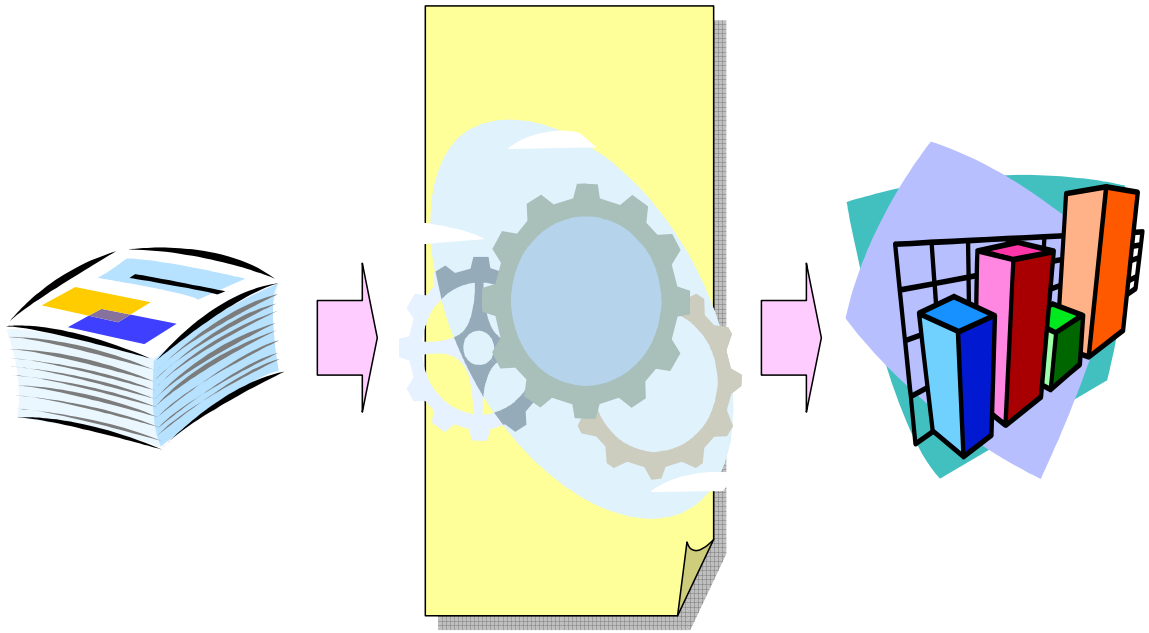
이번에 학습할 내용

- 첫번째 프로그램 설명
- 화면 출력
- 연산이 있는 프로그램
- 입력이 있는 프로그램
- 오류 수정 및 디버깅
- 응용 프로그램



일반적인 프로그램의 형태

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



첫번째 덧셈 프로그램

add1.c

```c
/* 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("두수의 합: %d", sum);

        return 0;
}
```

변수 선언

변수에 값을 할당

덧셈 연산

변수의 값을 출력

실행결과

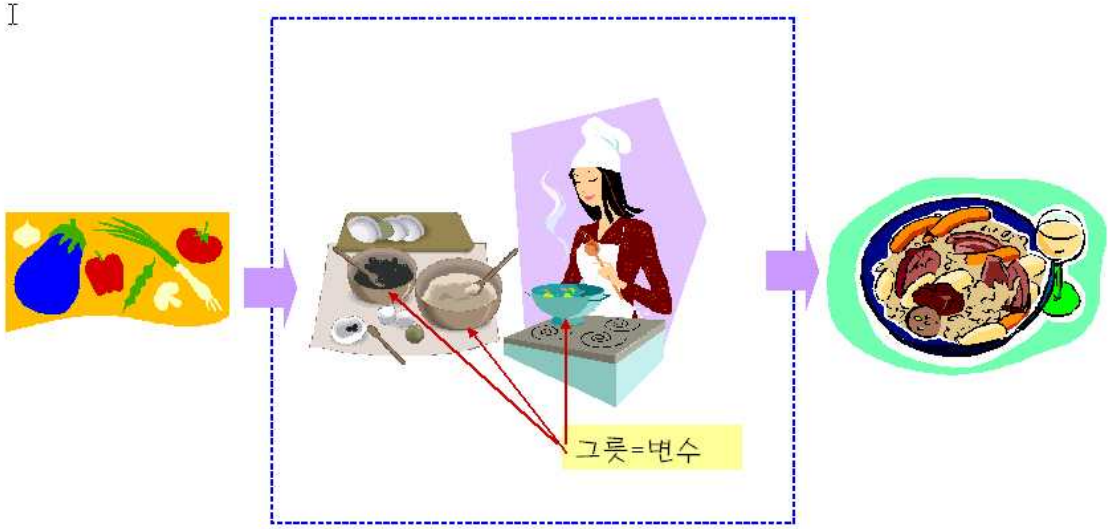
```
두수의 합: 300
```

변수

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

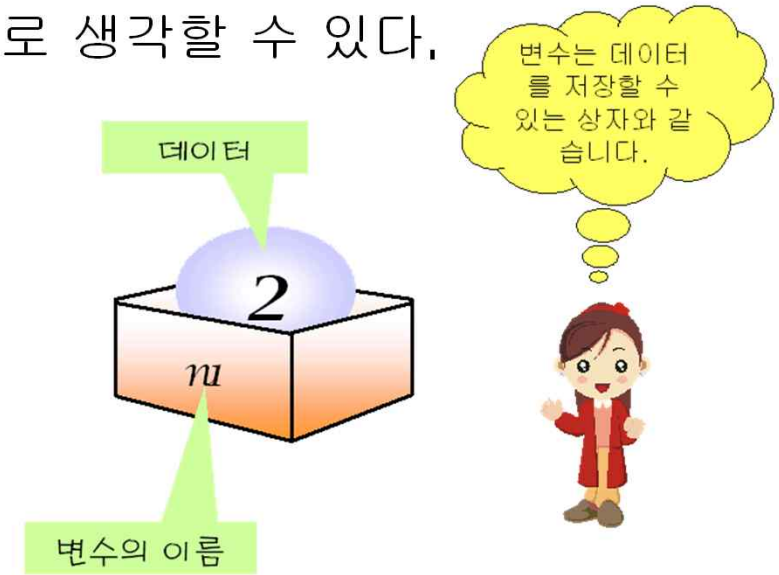
Q) 변수란 무엇인가?

A) 프로그램이 사용하는 데이터를 일시적으로 저장할 목적으로 사용하는 메모리 공간

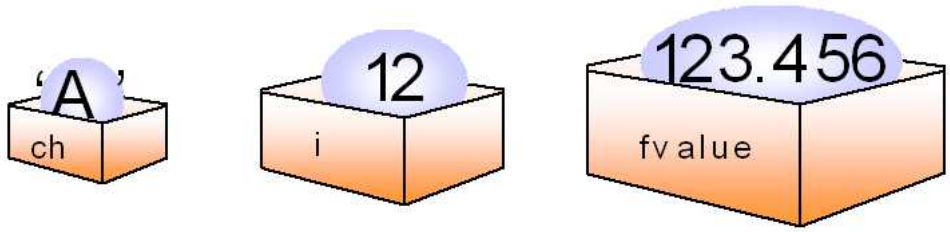


변수의 종류

- 변수는 데이터를 담는 상자로 생각할 수 있다.



- 변수에는 데이터의 종류에 따라 여러 가지 타입이 존재한다.



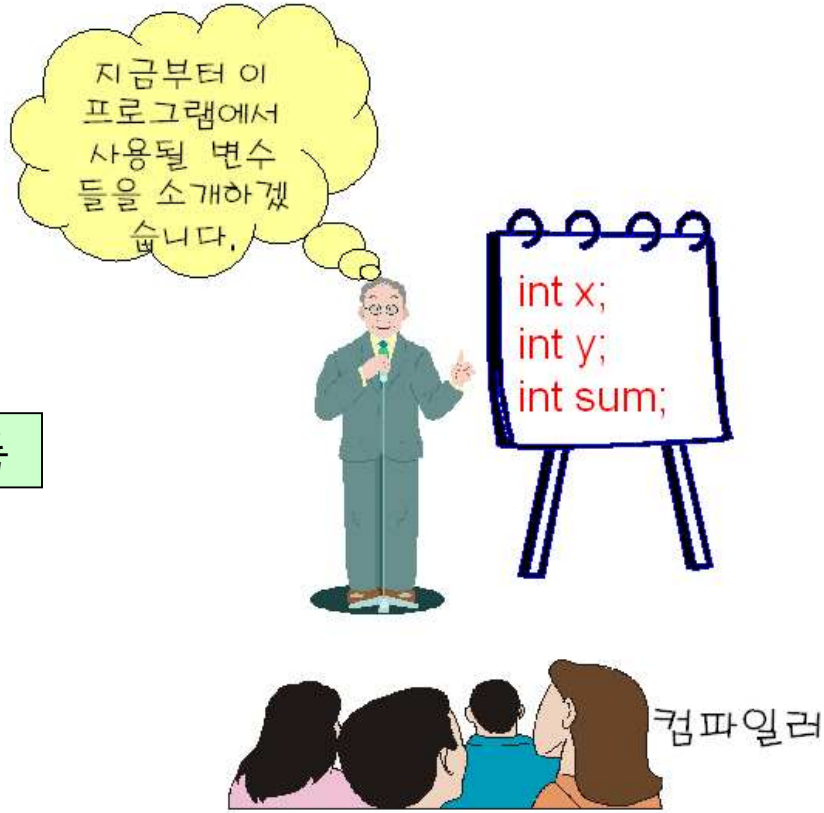
변수 선언

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

```
int x;
```

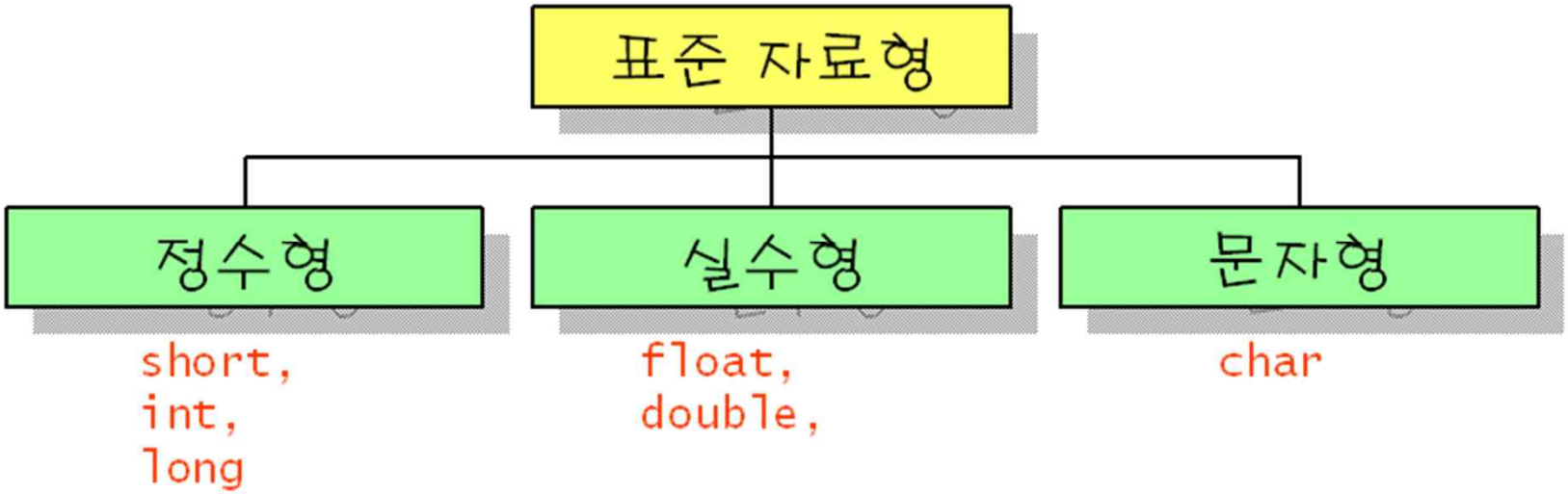
변수의 이름

자료형



자료형

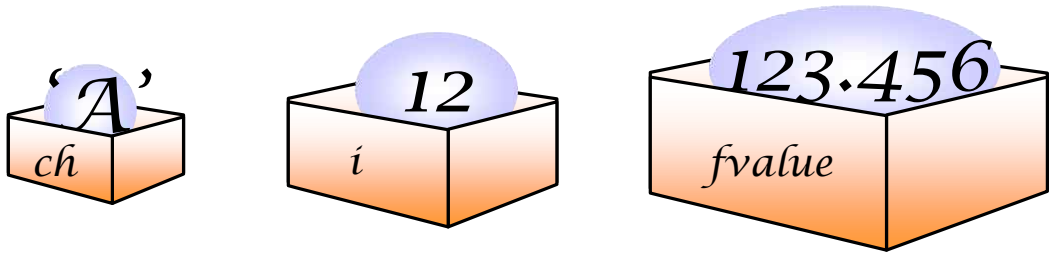
- 자료형(data type): 변수가 저장할 데이터가 정수인지 실수인지, 아니면 또 다른 어떤 데이터인지를 지정하는 것



변수의 이름

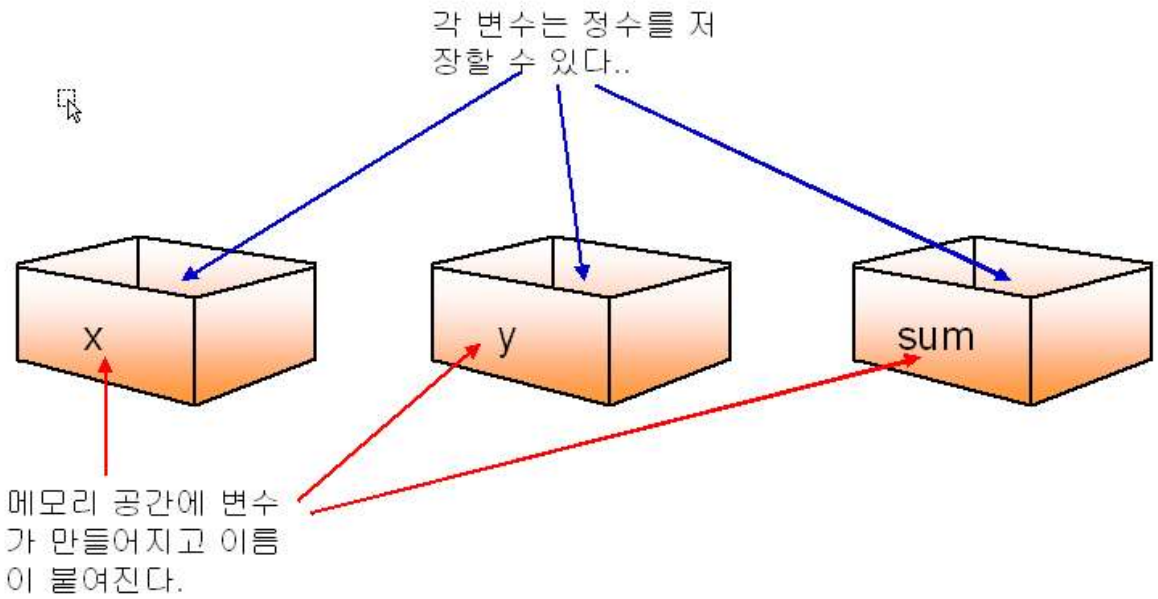
- 식별자(identifier): 변수나 함수의 이름
- 식별자를 만드는 규칙
  - 식별자는 영어의 대소문자, 숫자, 밑줄 문자 _로 이루어진다.
  - 식별자는 숫자로 시작할 수 없다.
  - 대문자와 소문자를 구별하며 C 언어의 키워드와 똑같은 이름은 허용되지 않는다.

- 식별자의 예:
  - s, s1, student_number: 올바른 식별자
  - s#, 2nd_student , int: 잘못된 식별자



변수 선언

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

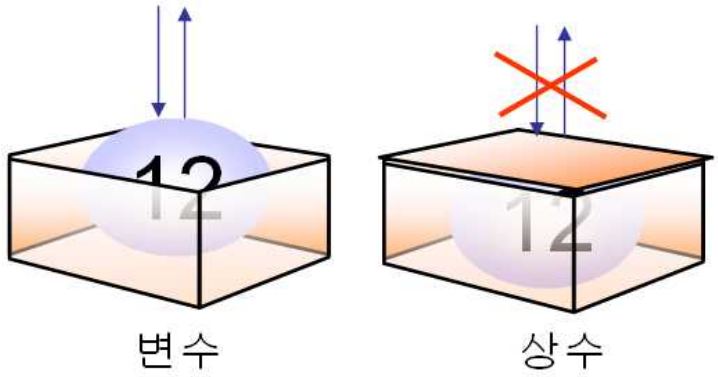


상수

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

상수

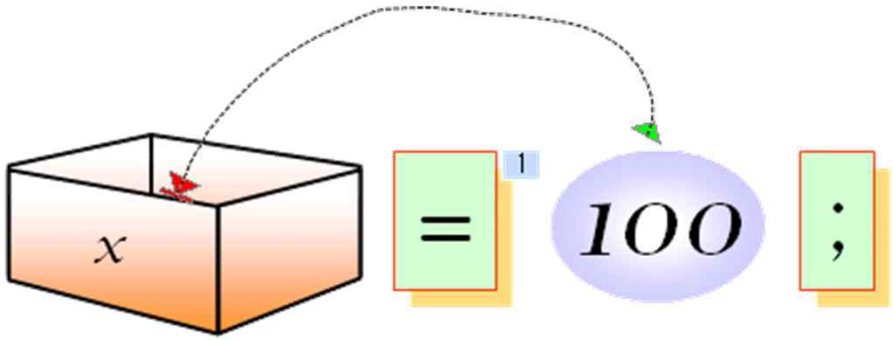
- 상수(constant): 그 값이 프로그램이 실행하는 동안 변하지 않는 수



대입 연산

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

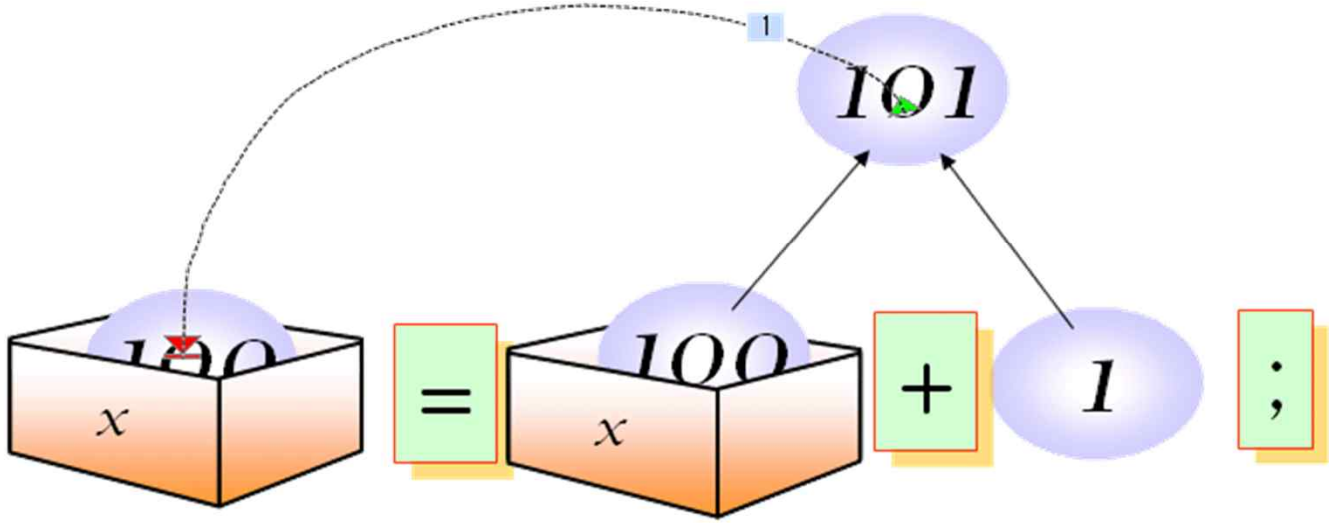
```
x = 100;
y = 200;
```



대입 연산(cont.)

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

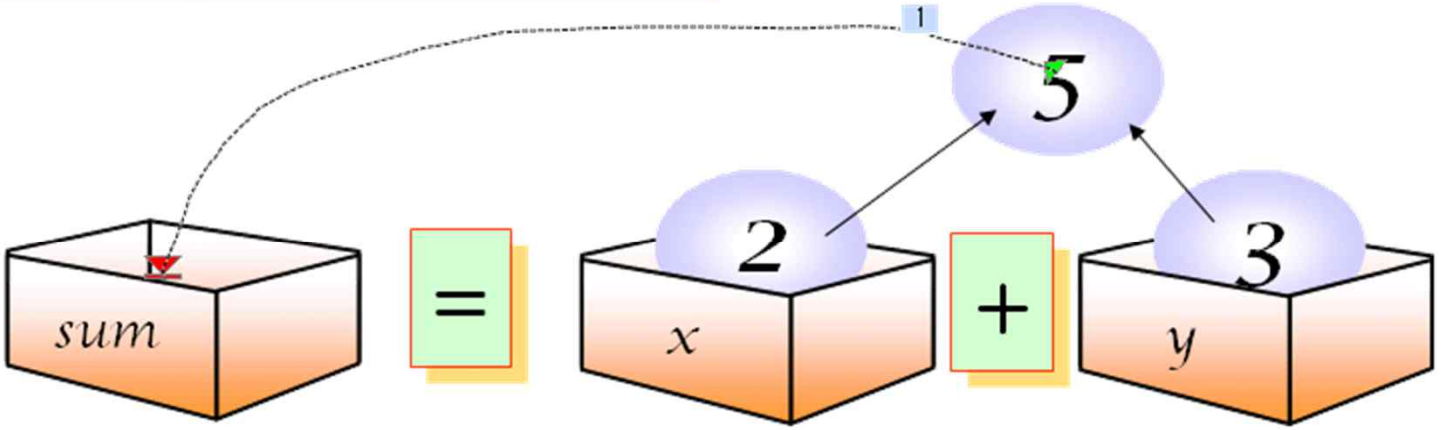
```
x = x + 1;
```



산술 연산

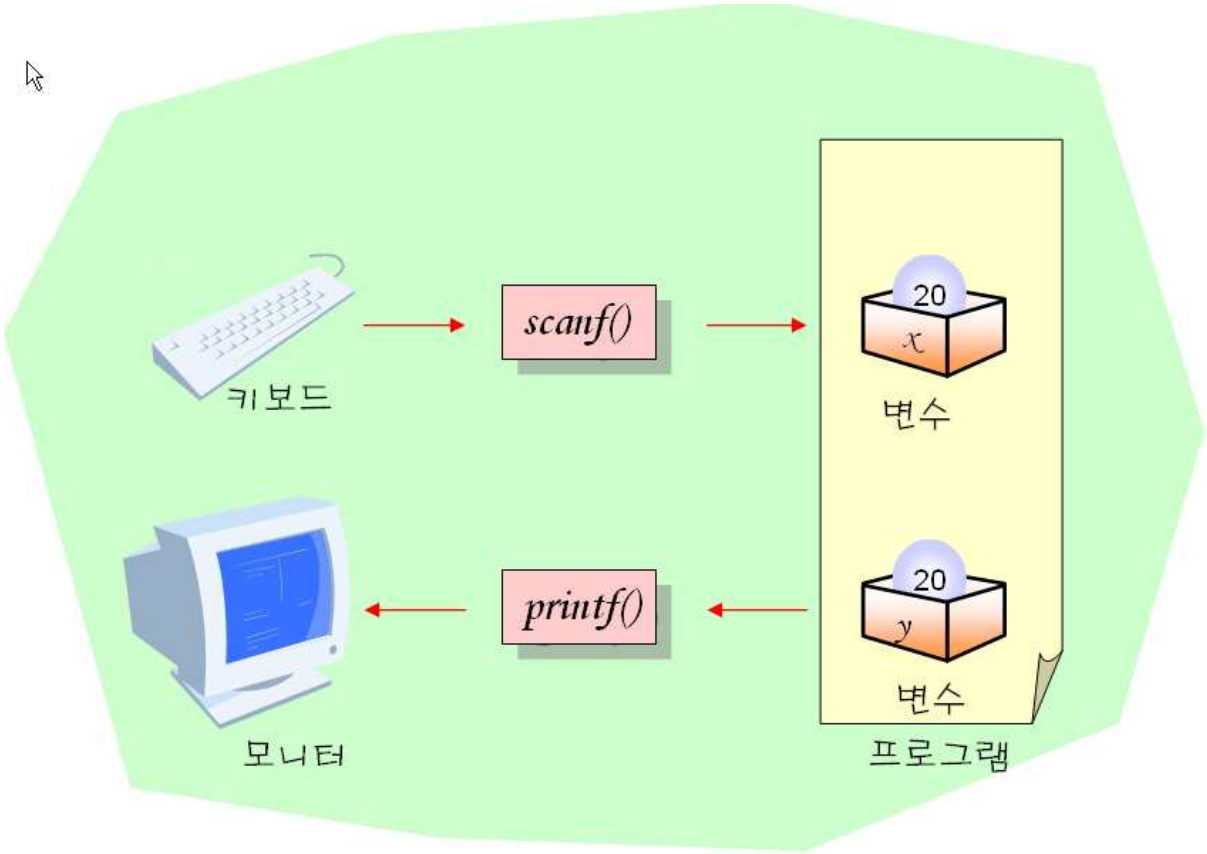
연산	연산자	C 수식	수학에서의 기호
덧셈	+	x + y	x + y
뺄셈	-	x - y	x - y
곱셈	*	x * y	xy
나눗셈	/	x / y	x/y 또는 또는 x ÷ y
나머지	%	x % y	x mod y

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



printf()

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



printf()의 형식

형식	printf(형식제어문자열, 변수);
예	printf("%d", sum);

표 2.2 형식 지정자의 종류

형식 제어 문자열	의미	형태
%d	정수 형태로 출력	100
%f	실수 형태로 출력	3.141592
%c	문자 형태로 출력	A
%s	문자열 형태로 출력	Hello

printf()의 출력 과정

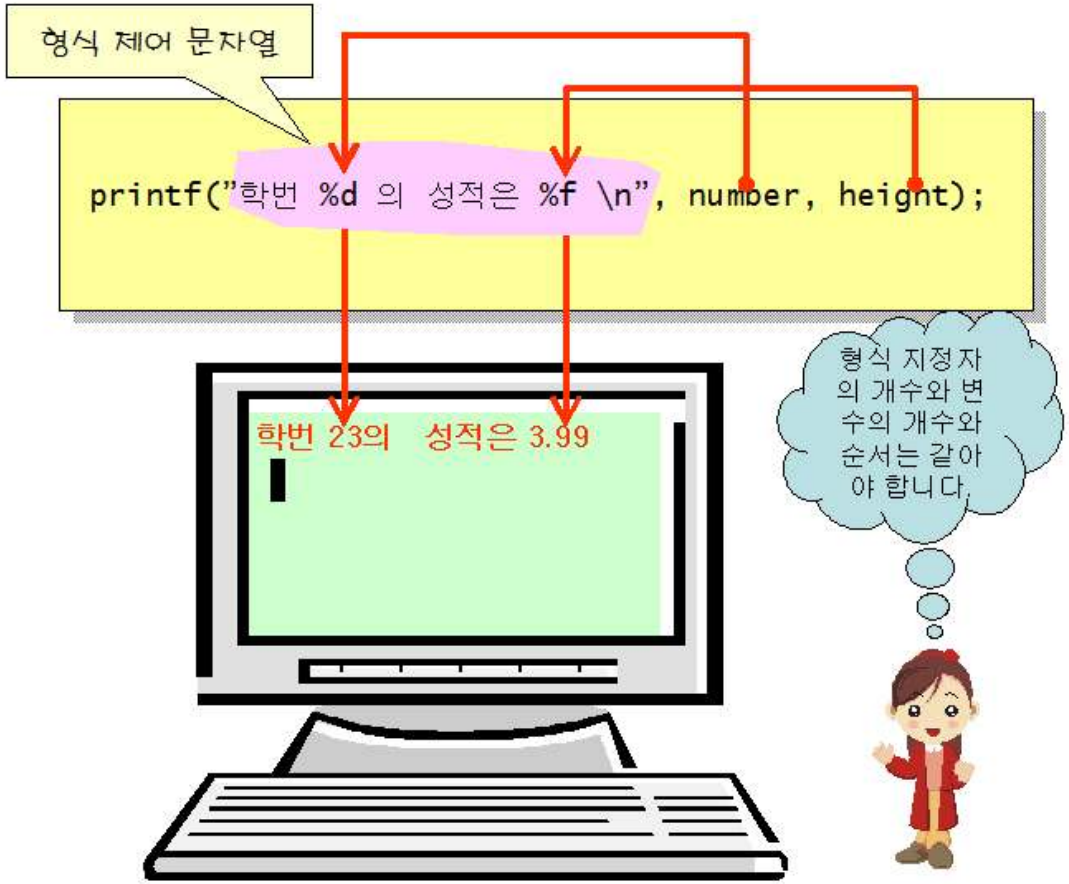
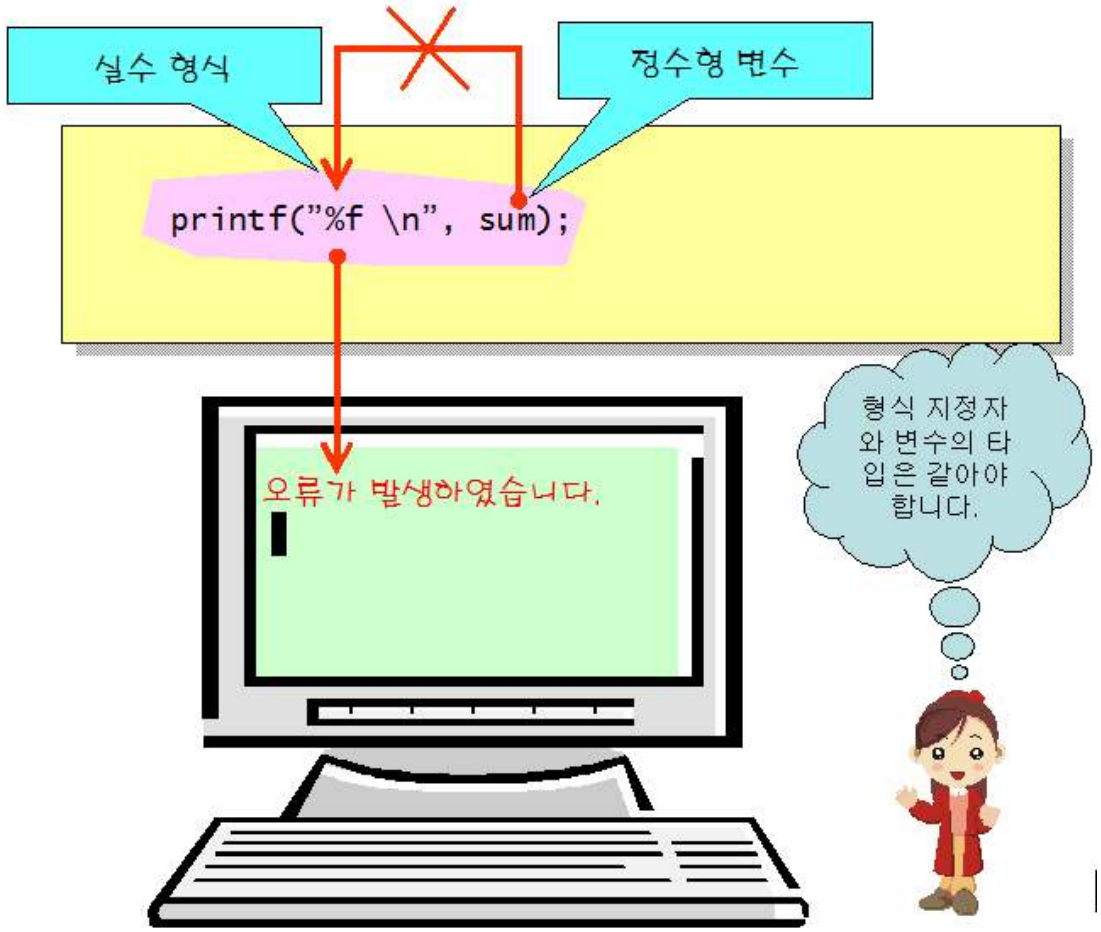


그림 2.22 printf()에서의 형식 제어 문자열

형식 지정자와 변수의 타입은 일치하여야 함



복습

add1.c

```
/* 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("두수의 합: %d", sum);

        return 0;
}
```

변수 선언

변수에 값을 할당

덧셈 연산

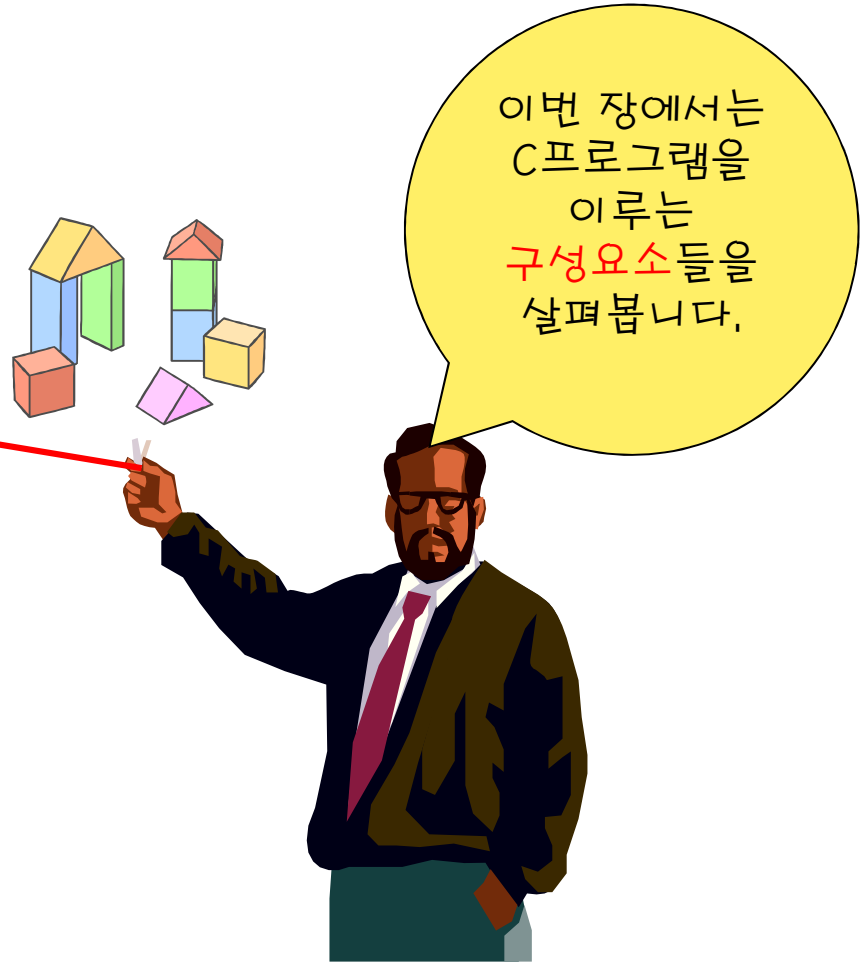
변수의 값을 출력

실행결과

```
두수의 합: 300
```

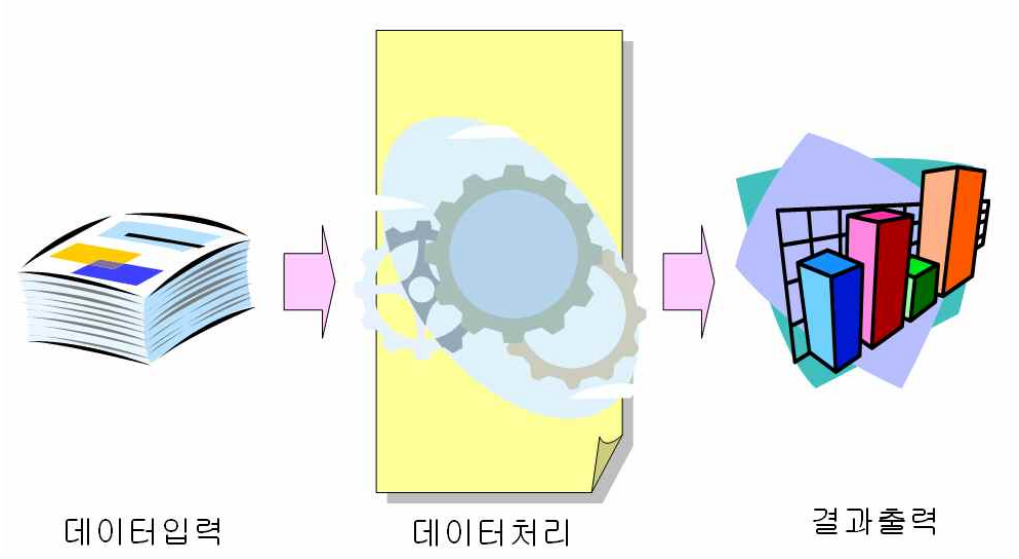
이번에 학습할 내용

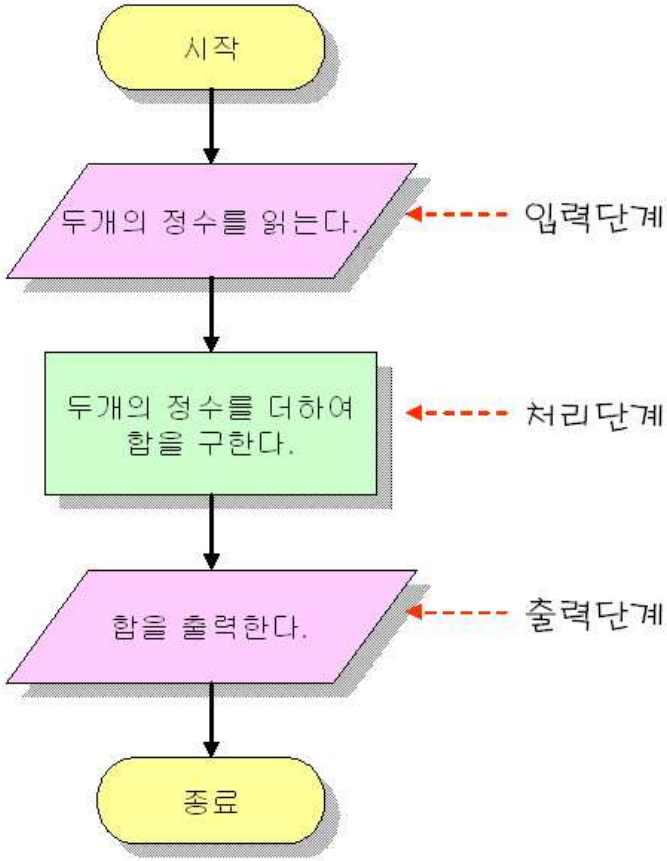
- 첫번째 프로그램 설명
- 화면 출력
- 연산이 있는 프로그램
- 입력이 있는 프로그램
- 오류 수정 및 디버깅
- 응용 프로그램



덧셈 프로그램 #2

- 사용자로부터 입력을 받아보자.





두번째 덧셈 프로그램

```c
// 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 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
}
```

```
첫번째 숫자를 입력하시오:10
두번째 숫자를 입력하시오:20
두수의 합: 30
```

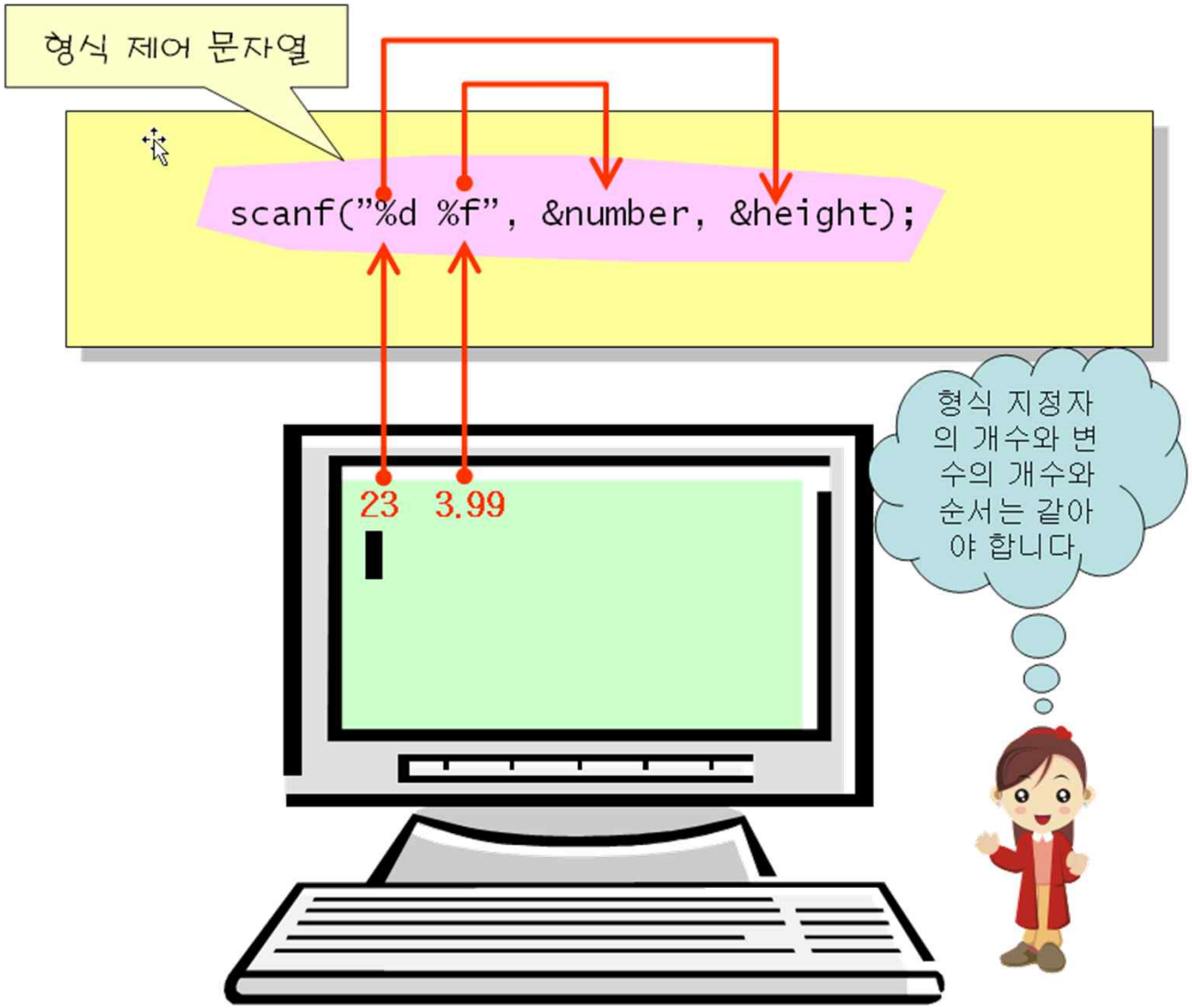
scanf()

- scanf(): 키보드로부터 입력을 하기 위한 라이브러리 함수

```
형식   scanf(형식제어문자열, &변수1);
예     scanf("%d", &x);
```

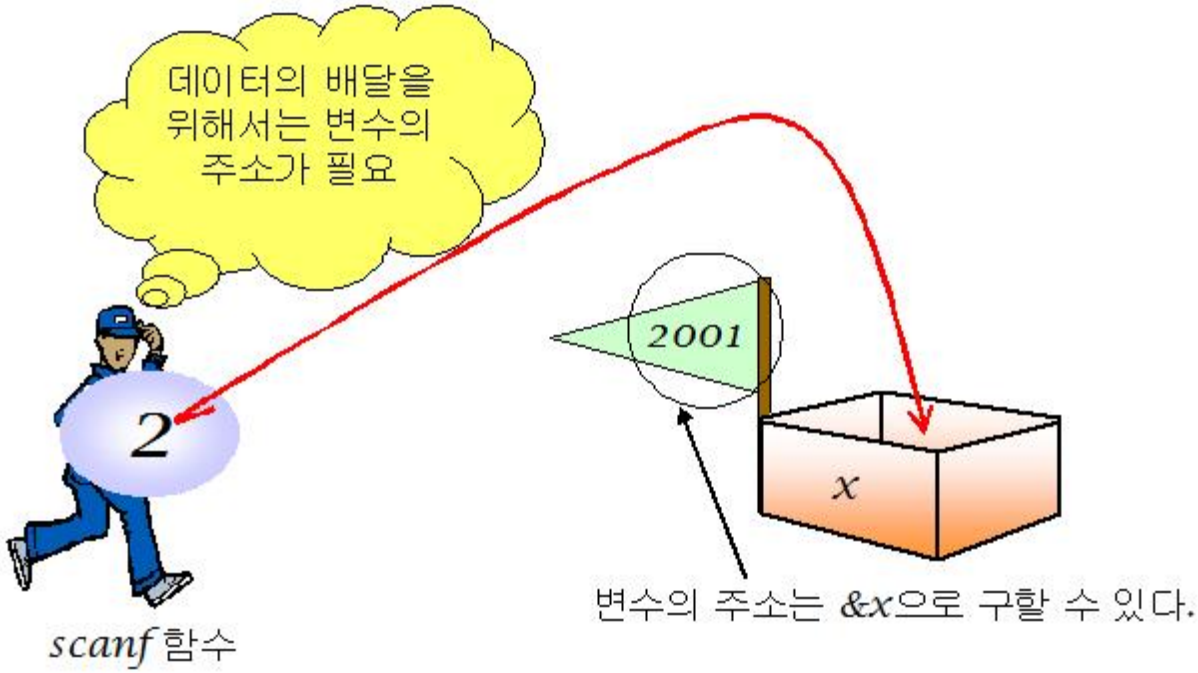
형식 지정자	의미	형태
%d	정수	100
%f	실수(float)	3.14
%lf	실수(double)	3.141592
%c	문자	A
%s	문자열	Hello World!

scanf()



&의 의미

- & 연산자: 변수의 주소를 계산하는 연산자
- 변수에 값을 저장하려면 변수의 주소가 필요



실수 입력

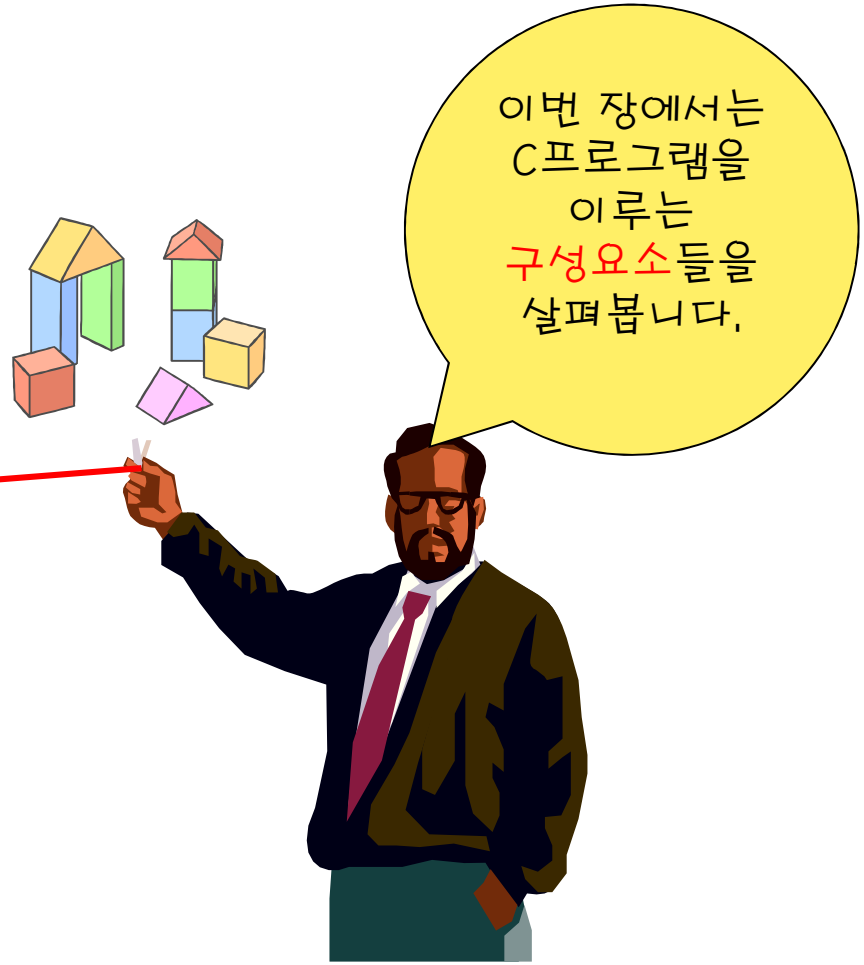
- float ratio = 0.0;
  scanf("%f", &ratio);

- double scale = 0.0;
  scanf("%lf", &scale);

주의!!!

이번에 학습할 내용

- 첫번째 프로그램 설명
- 화면 출력
- 연산이 있는 프로그램
- 입력이 있는 프로그램
- 오류 수정 및 디버깅
- 응용 프로그램



오류 수정 및 디버깅

- 컴파일이나 실행 시에 오류가 발생할 수 있다.
- 에러와 경고
  - 에러(error): 심각한 오류
  - 경고(warning): 경미한 오류

- 오류의 종류
  - 컴파일 시간 오류: 대부분 문법적인 오류
  - 실행 시간 오류: 실행 중에 0으로 나누는 연산 같은 오류
  - 논리 오류: 논리적으로 잘못되어서 결과가 의도했던 대로 나오지 않는 오류

오류가 발생하는 프로그램

error.c

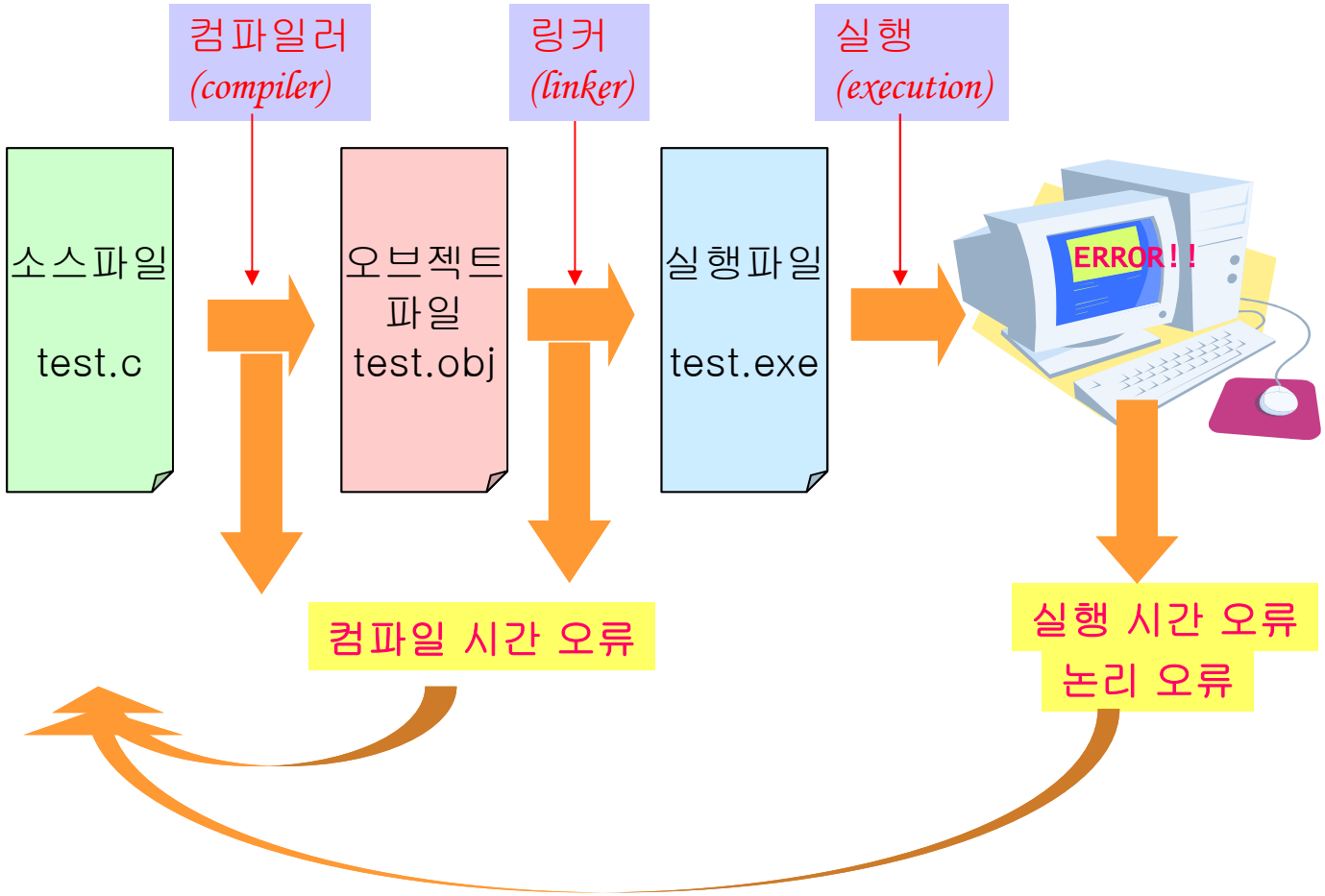
```
/* 에러가 발생하는 프로그램*/
#include <stdio.h>

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

문장의 끝에 세미콜론이 빠져 있다.

7번째 라인에서 오류

```
1>------ 빌드 시작: 프로젝트: test, 구성: Release Win32 ------
1>컴파일하고 있습니다...
1>error.cpp
1>.\error.cpp(7) : error C2143: 구문 오류 : ';'이(가) 'return' 앞에 없습니다.
1>빌드 로그가 "file://c:\source\error\error\Release\BuildLog.htm"에 저장되었습니다.
1>test - 오류: 1개, 경고: 0개
========== 빌드: 성공 0, 실패 1, 최신 0, 생략 0 ==========
```

오류 수정 과정



디버깅

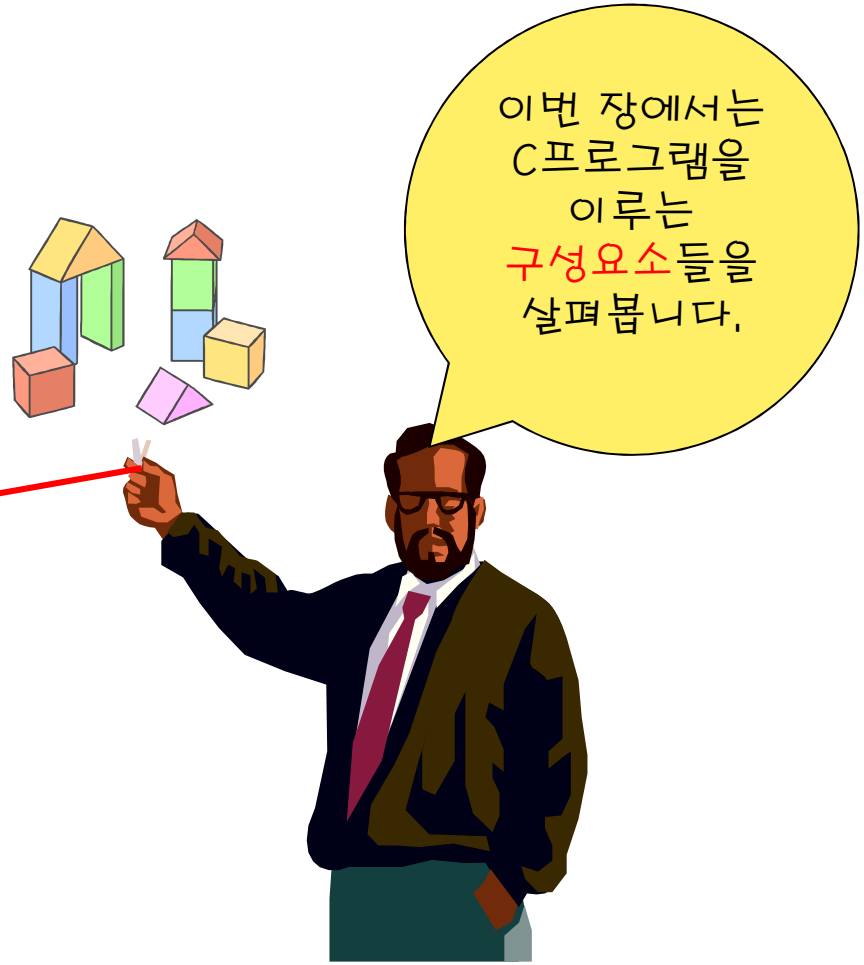
- 디버깅: 논리 오류를 찾는 과정

중간 점검

1. 오류를 심각성의 정도에 따라 분류하여 보자.
2. 작성된 프로그램이 C언어의 문법을 지키지 않았으면 어떤 오류에 속하는가?

이번에 학습할 내용

- 첫번째 프로그램 설명
- 화면 출력
- 연산이 있는 프로그램
- 입력이 있는 프로그램
- 오류 수정 및 디버깅
- 응용 프로그램



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
```

Q & A