

개정3판

Visual
Studio
2017

쉽게 풀어쓴



언어

EXPRESS

천인국 지음

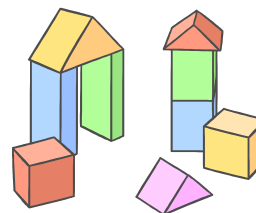
제3장 C프로그램 구성요소



이번 장에서 학습할 내용

2

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



이번 장에서는
C프로그램을
이루는
구성요소들을
살펴봅니다.



일반적인 프로그램의 형태

3

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



데이터 입력



데이터 처리



결과출력



덧셈 프로그램 #1

4

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

5





주석(comment)

6

```
/* 두개의 숫자의 합을 계산하는 프로그램 */
```

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

```
int main(void)
```

```
{
```

```
...
```

```
...
```

```
...
```

```
}
```

주석은 코드를
설명하는 글입니다.



주석



3가지 방법의 주석

7

- /* 한 줄로 된 주석 */
- /* -----
저자: 홍길동
날짜: 2013.3.4
여러 줄로 이루어진 주석
----- */
- // 여기서부터 줄의 끝까지 주석



주석의 예

8

- 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

9

- **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..

10

```
#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 알기
가 힘들고 또한 들
여쓰기가 안 되어
있어서 같은 수준에
있는 문장들을 구분
하기 힘듭니다.





중간 점검

11

- 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

12

`#include <stdio.h>`

- 외부 파일을 포함시키라는 의미의 전처리기
- `#`기호로 시작

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



전처리기

13

```
/* 첫번째 프로그램 */
```

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

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

hello.c

```
// stdio.h
```

```
...
```

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

```
...
```

stdio.h



중간 점검

14

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

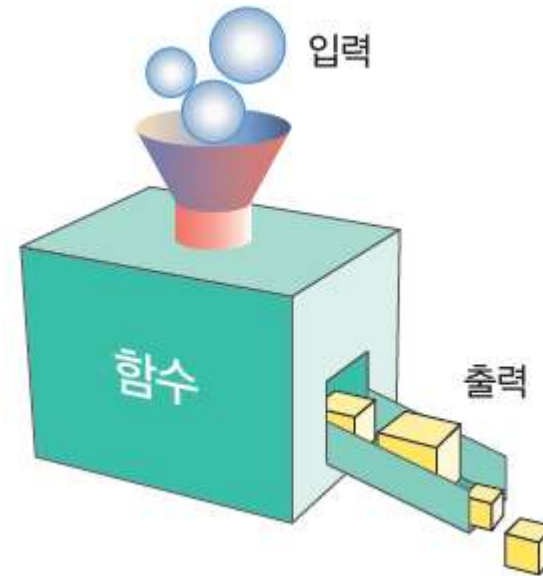




함수

15

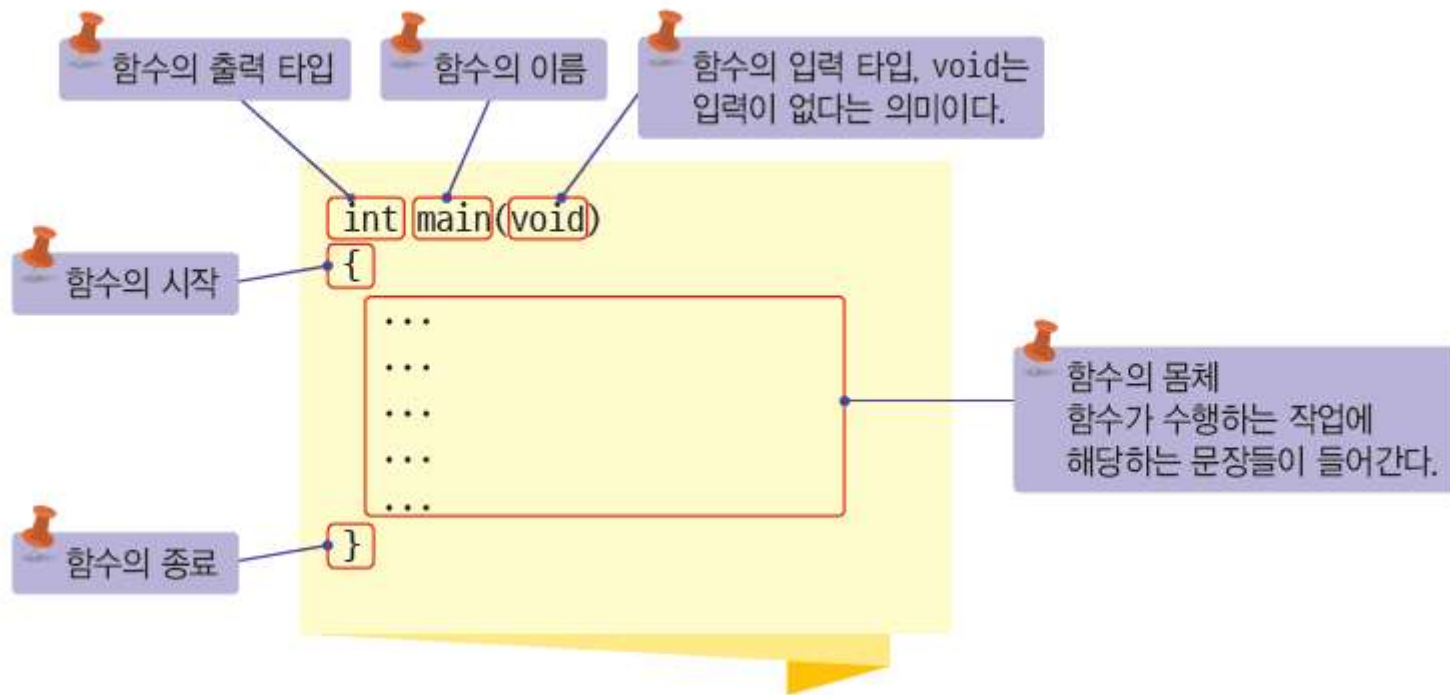
- 함수(function): 특정 기능을 수행하는 처리 단계들을 괄호로 묶어서 이름을 붙인 것
- 함수는 프로그램을 구성하는 기본적인 단위(부품)





함수의 구조

16



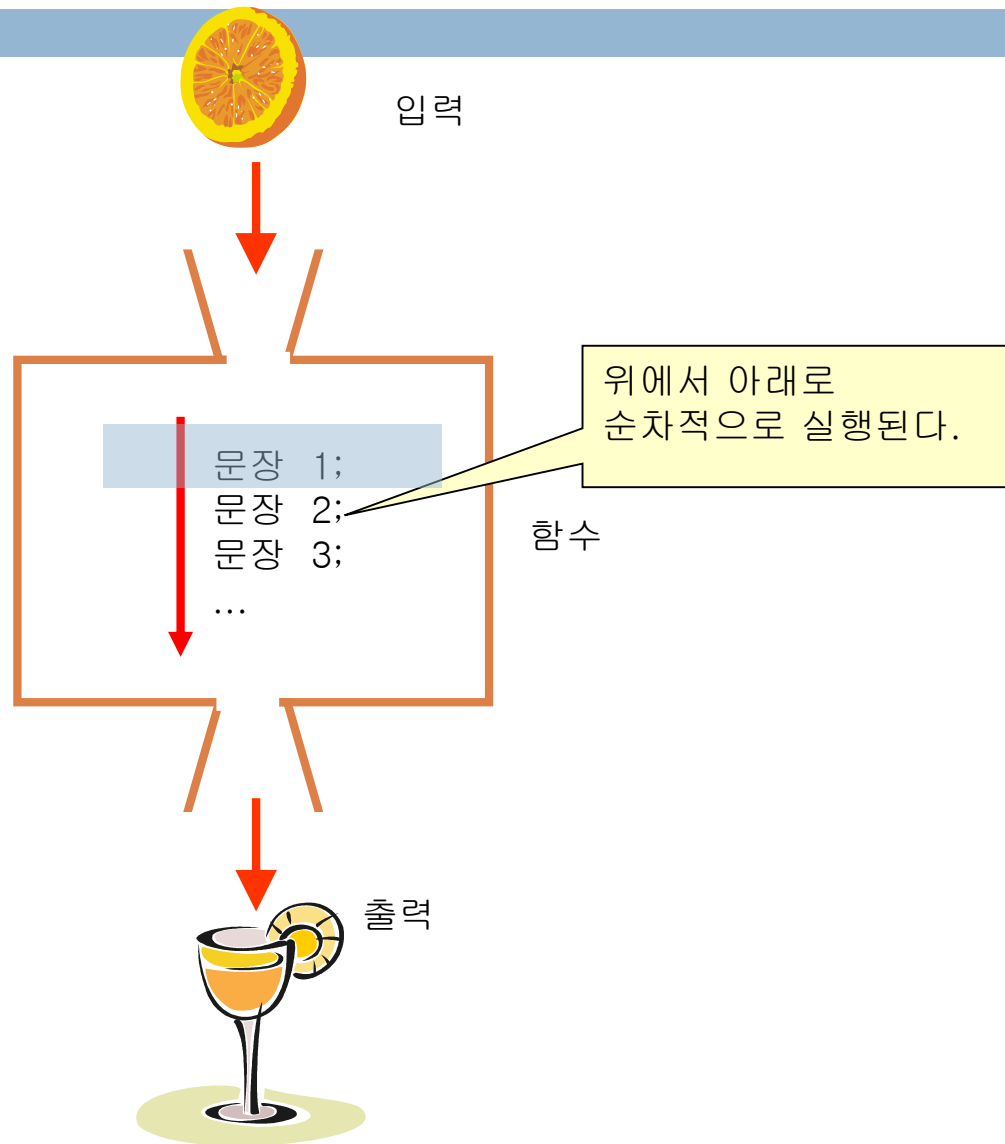


함수안에 들어 있는 것

17

Q) 그렇다면 함수 안에 들어 있는 것은 무엇인가?

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






함수

18

- 작업을 수행하는 문장은 함수 안에 들어가야 함

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

- 여기에 작업을 넣을 것.



함수 호출

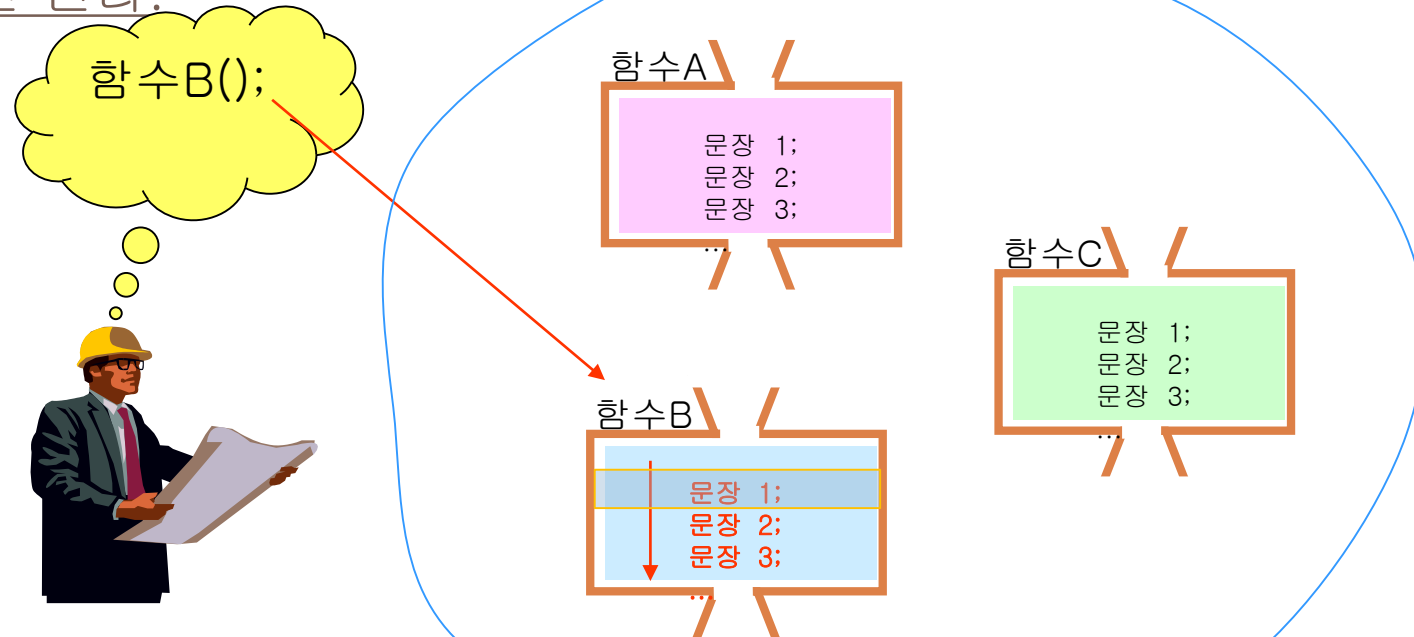
19

Q) 함수 안에 있는 문장들은 언제 실행되는가?

A) 함수가 호출되면 실행된다.

Q) 함수 호출은 어떻게 하는가?

A) 함수의 이름을 적어주면 된다.



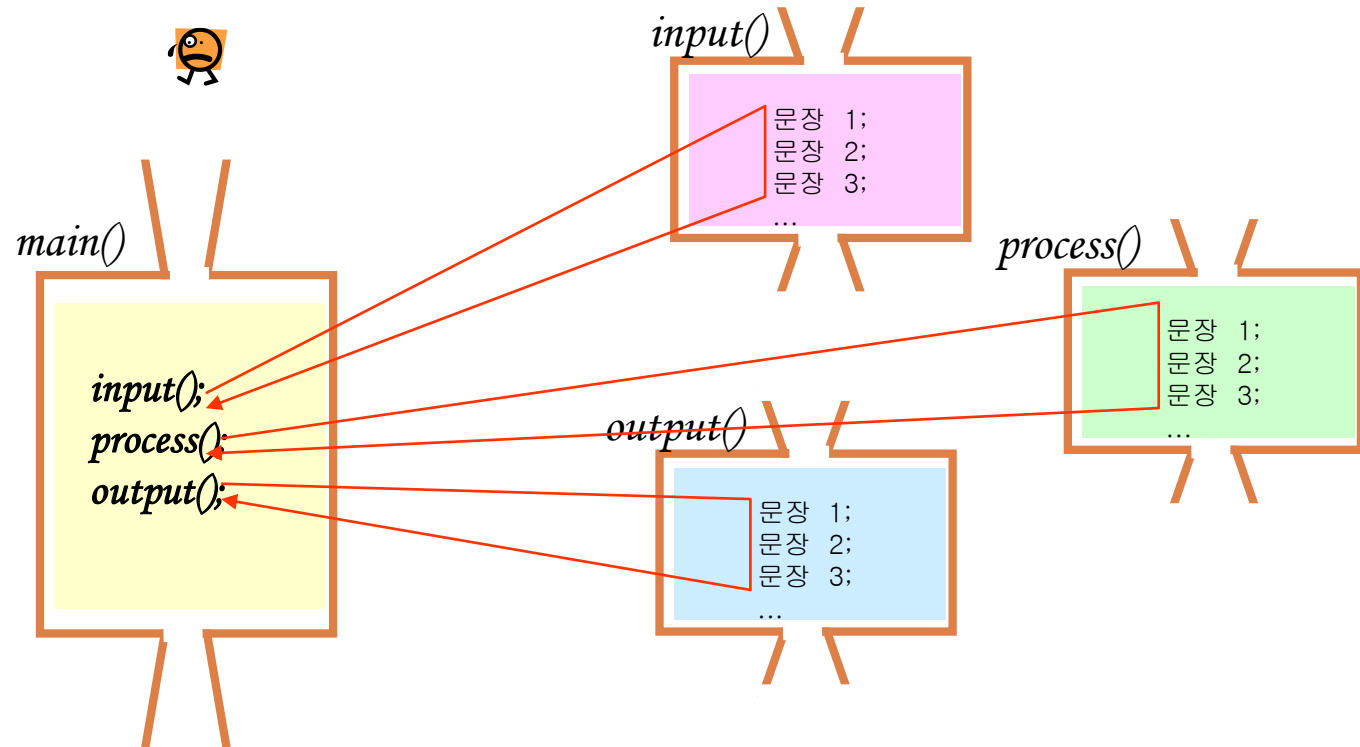


함수 호출

20

Q) 많은 함수 중에서 가장 먼저 실행되는 것은?

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



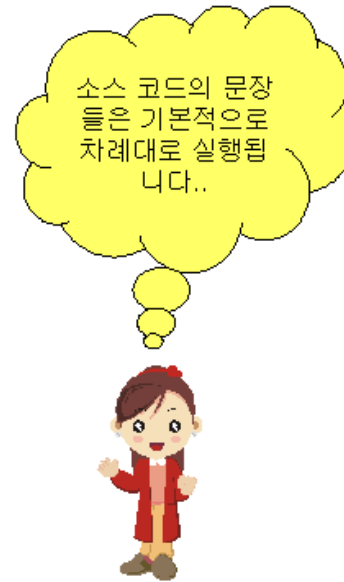
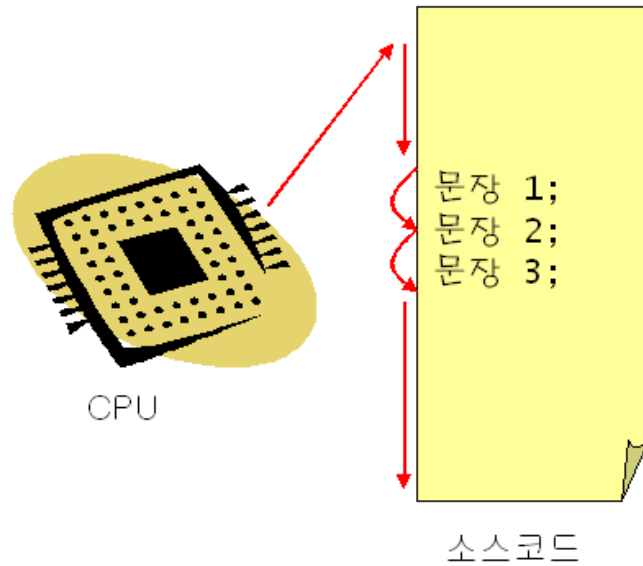


문장

21

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

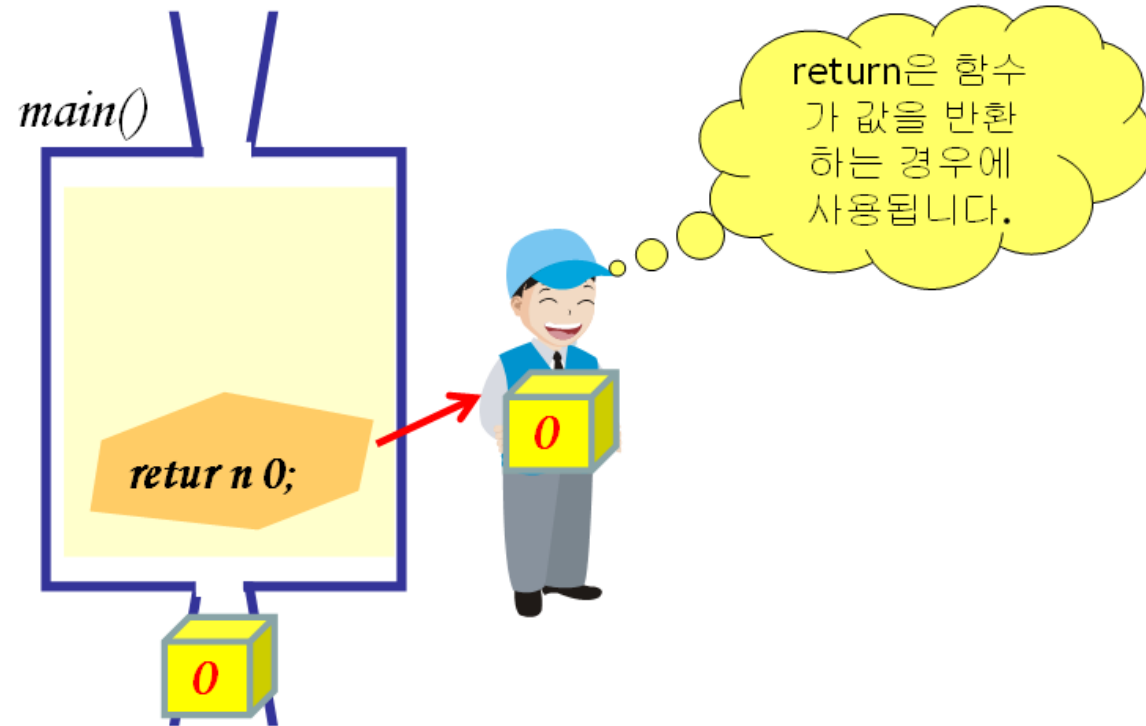
I





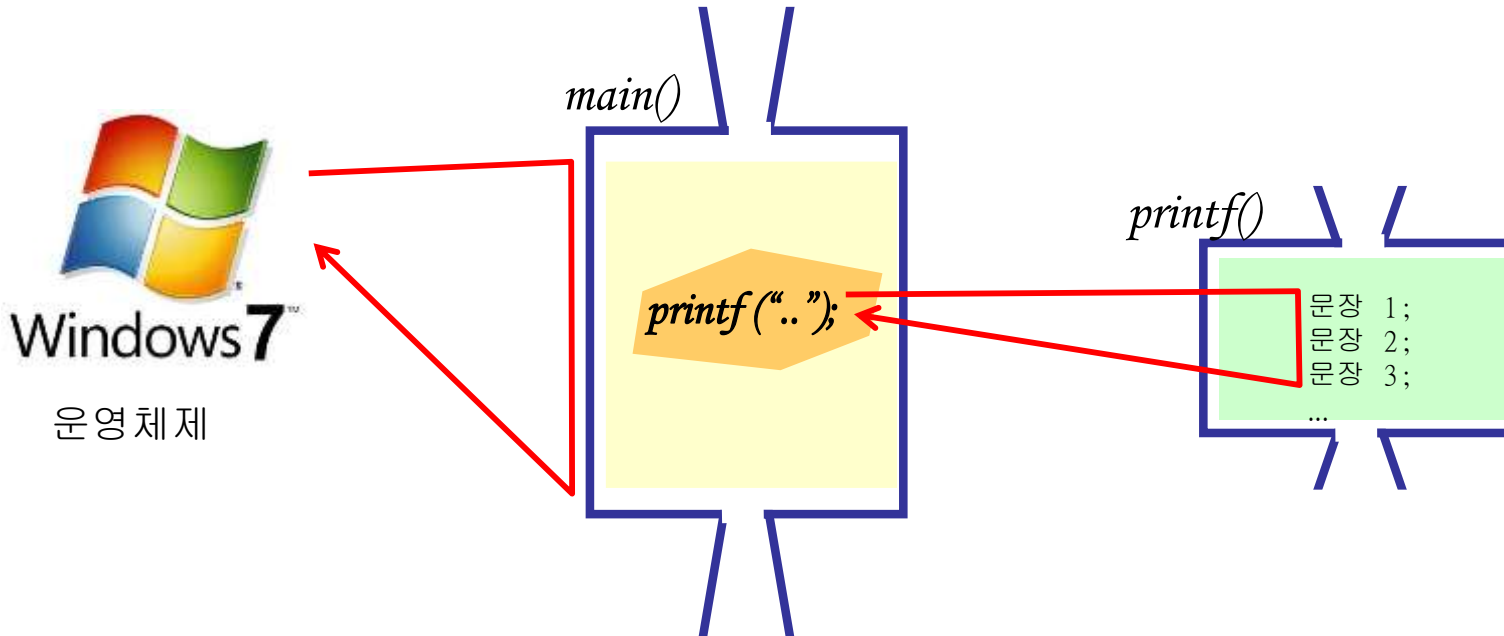
return 문장

22





main()은 누가 호출할까?





중간 점검

24

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



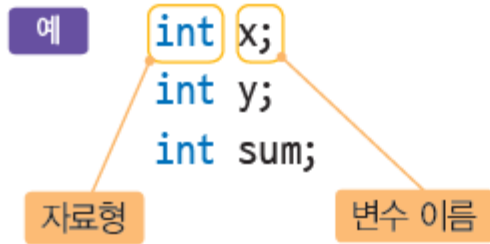


변수

25

- 프로그램이 사용하는 데이터를 일시적으로 저장할 목적으로 사용하는 메모리 공간

Syntax: 변수 선언





변수는 왜 필요한가?

26

- 변수는 데이터 값을 일시적으로 저장하는 역할을 한다.

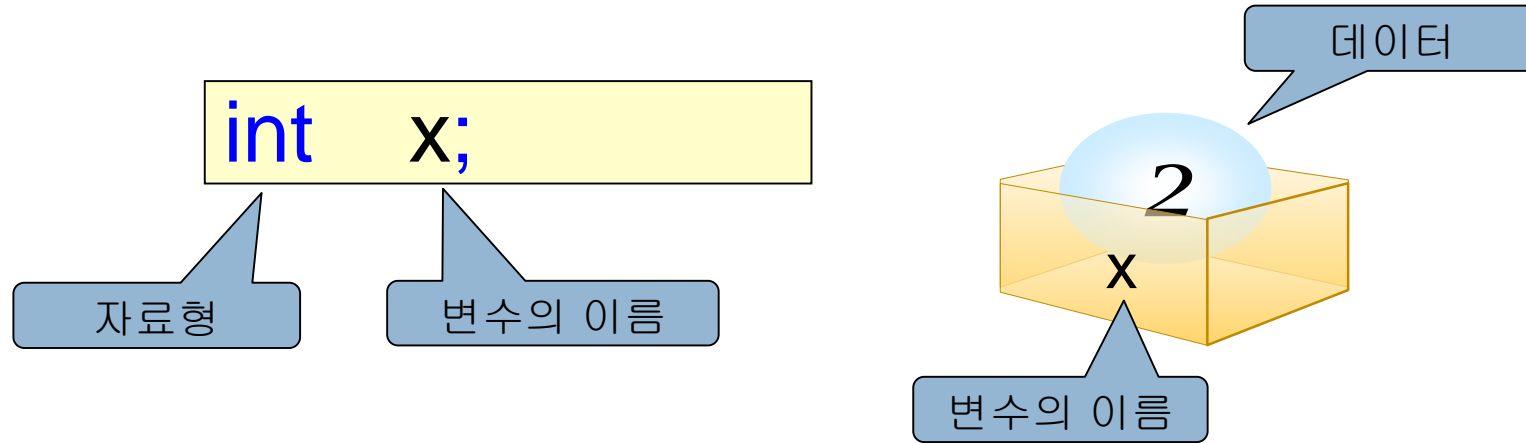




변수의 종류

27

- 변수는 데이터를 담는 상자로 생각할 수 있다.

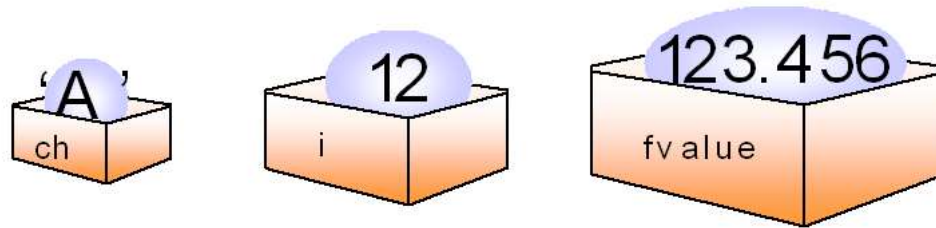




변수의 종류

28

- 변수에는 데이터의 종류에 따라 여러 가지 타입이 존재한다.



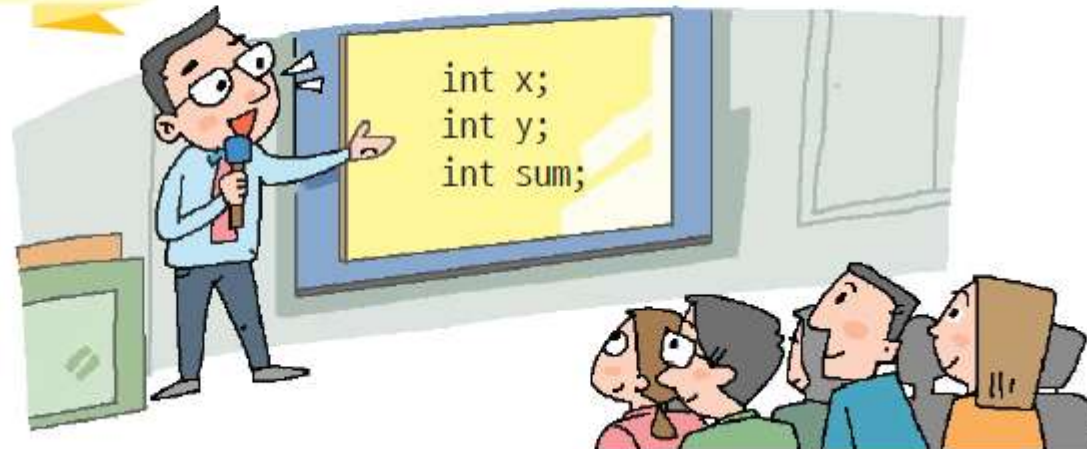


변수 선언

29

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

지금부터 이 프로그램에서 사용될
변수들을 소개하겠습니다.



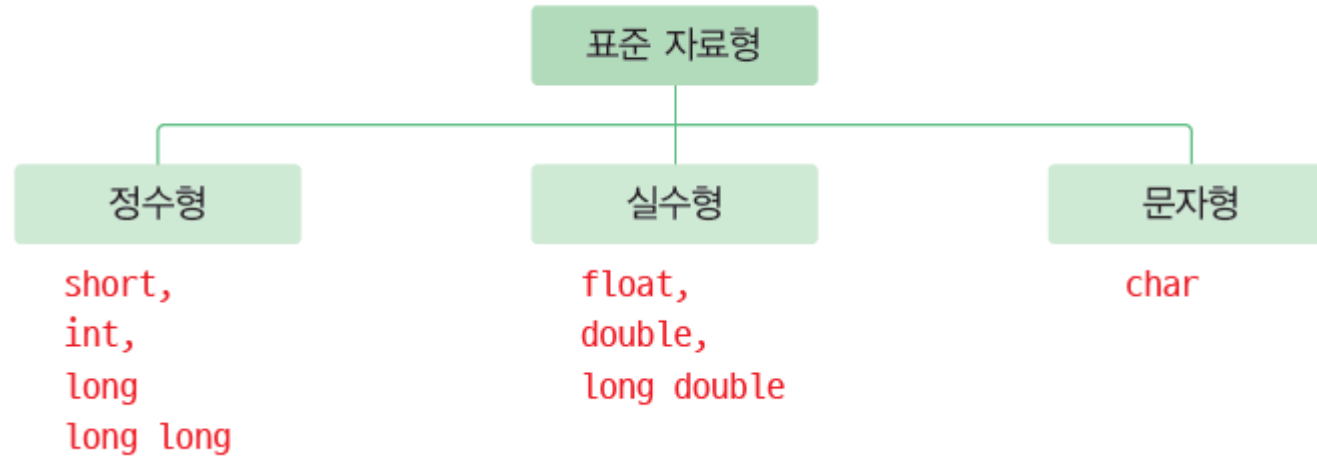
컴파일러



자료형

30

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

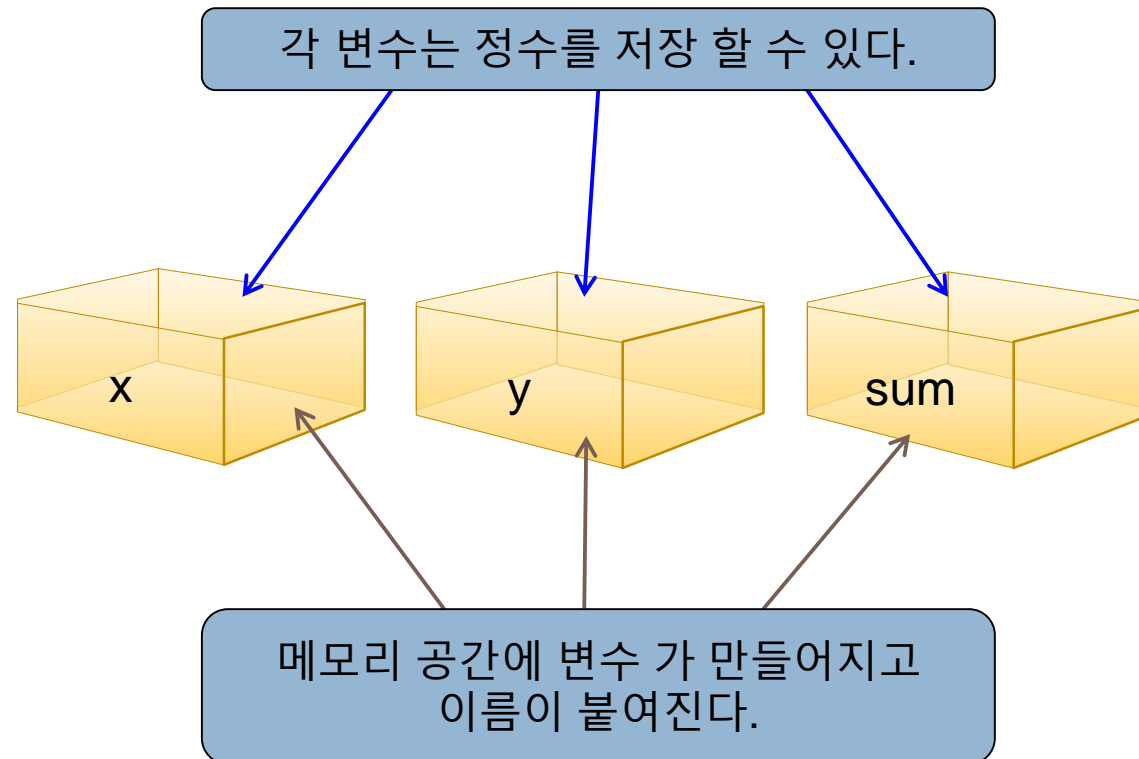




변수 선언

31

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





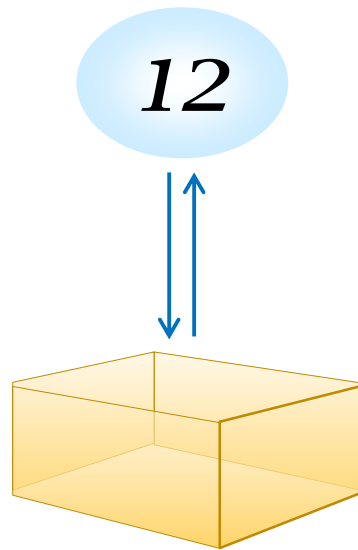
상수

32

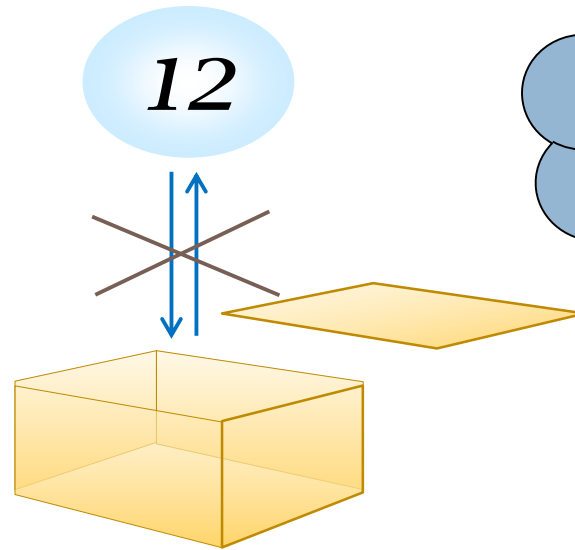
```
x = 100;  
y = 200;
```

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.





변수의 이름

33

- 식별자(identifier): 변수나 함수의 이름





변수의 이름

34

- 식별자 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.



키워드

35

- 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 이름

36



```
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)
```

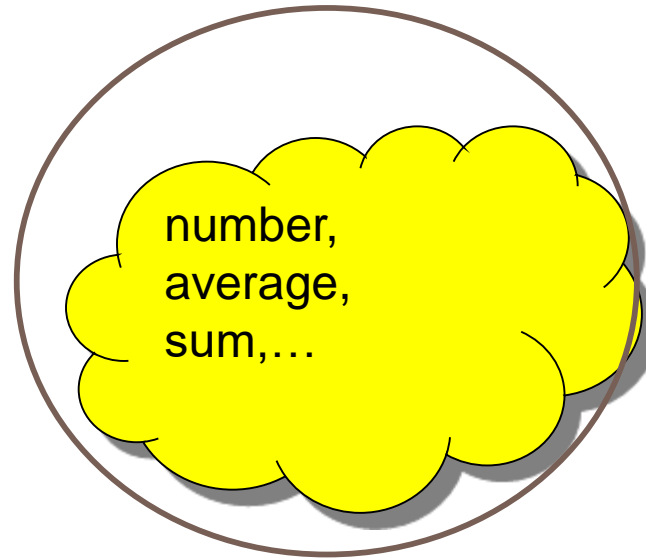
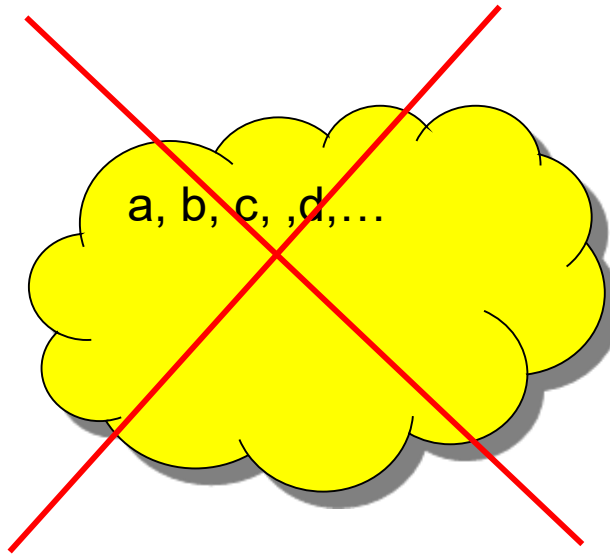
// 숫자로 시작할 수 없다.
// #과 같은 기호는 사용할 수 없다.
// double은 C언어의 키워드이다.



좋은 변수 이름

37

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





중간 점검

38

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





변수의 초기화

39

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

초기값





수식

40

- 수식(expression): 피연산자와 연산자로 구성된 식
- 수식은 **결과값**을 가진다.



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

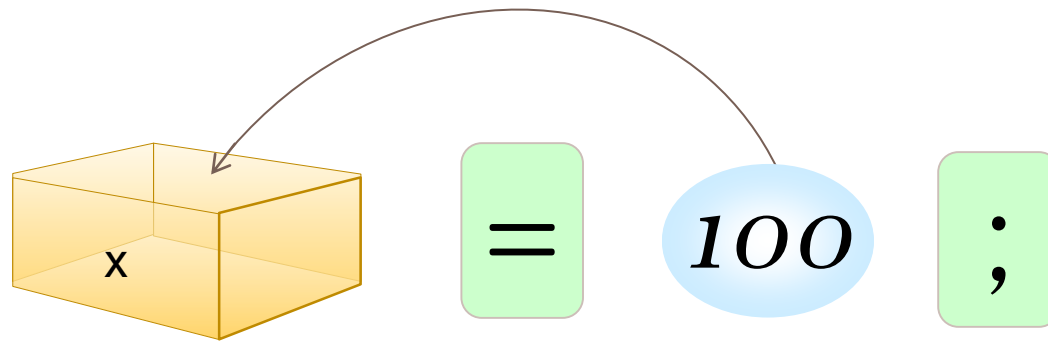



대입 연산

41

```
x = 100;
```

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





산술 연산

42

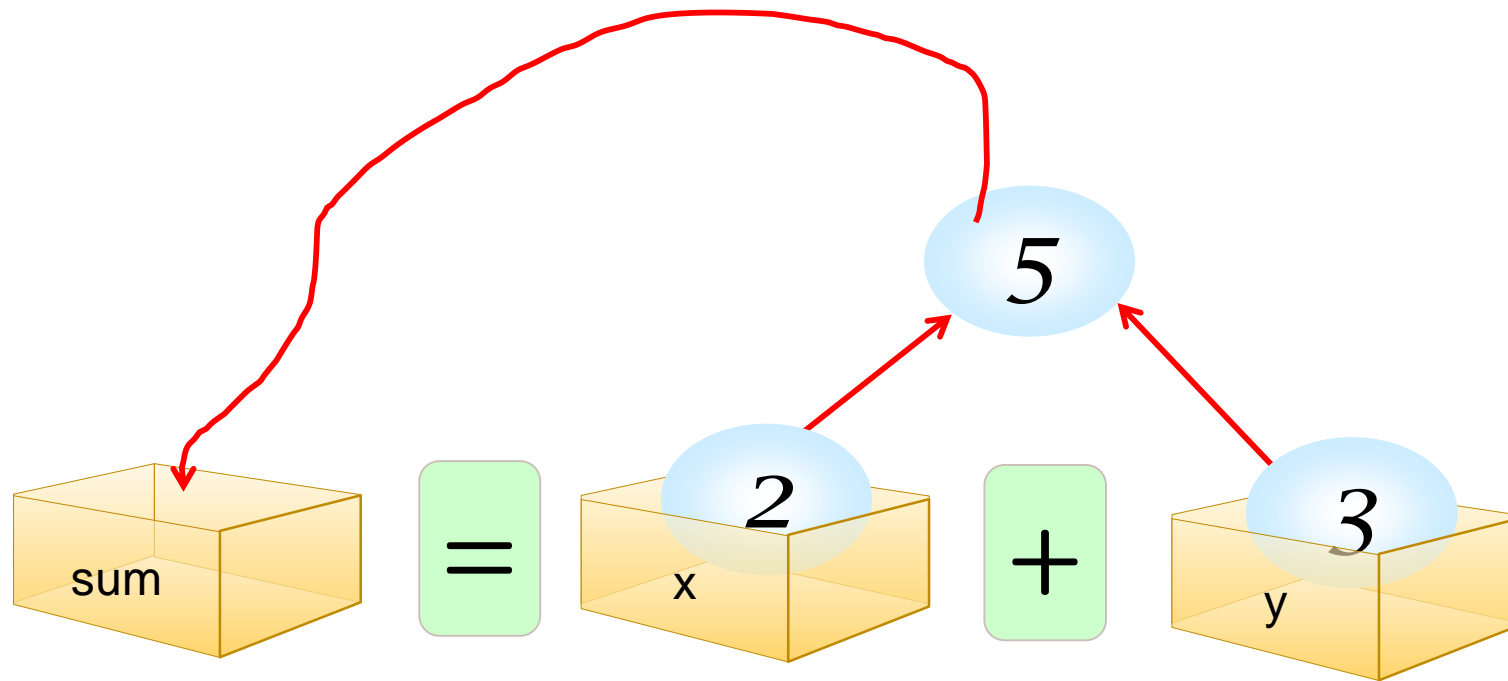
연산	연산자	C 수식	수학에서의 기호
덧셈	+	$x + y$	$x + y$
뺄셈	-	$x - y$	$x - y$
곱셈	*	$x * y$	xy
나눗셈	/	x / y	x/y 또는 $\frac{x}{y}$ 또는 $x \div y$
나머지	%	$x \% y$	$x \bmod y$



산술 연산

43

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





정리

add1.c

44

프로그램

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

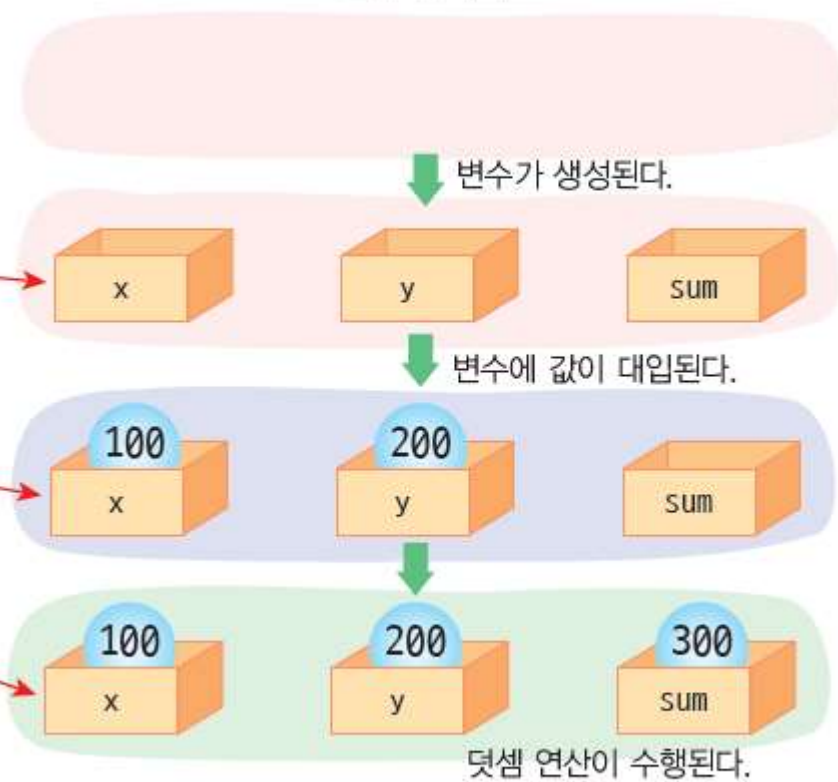
    x = 100;
    y = 200;

    sum = x + y;

    printf("두수의 합: %d", sum);

    return 0;
}
```

컴퓨터 내부





중간 점검

45

- 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.

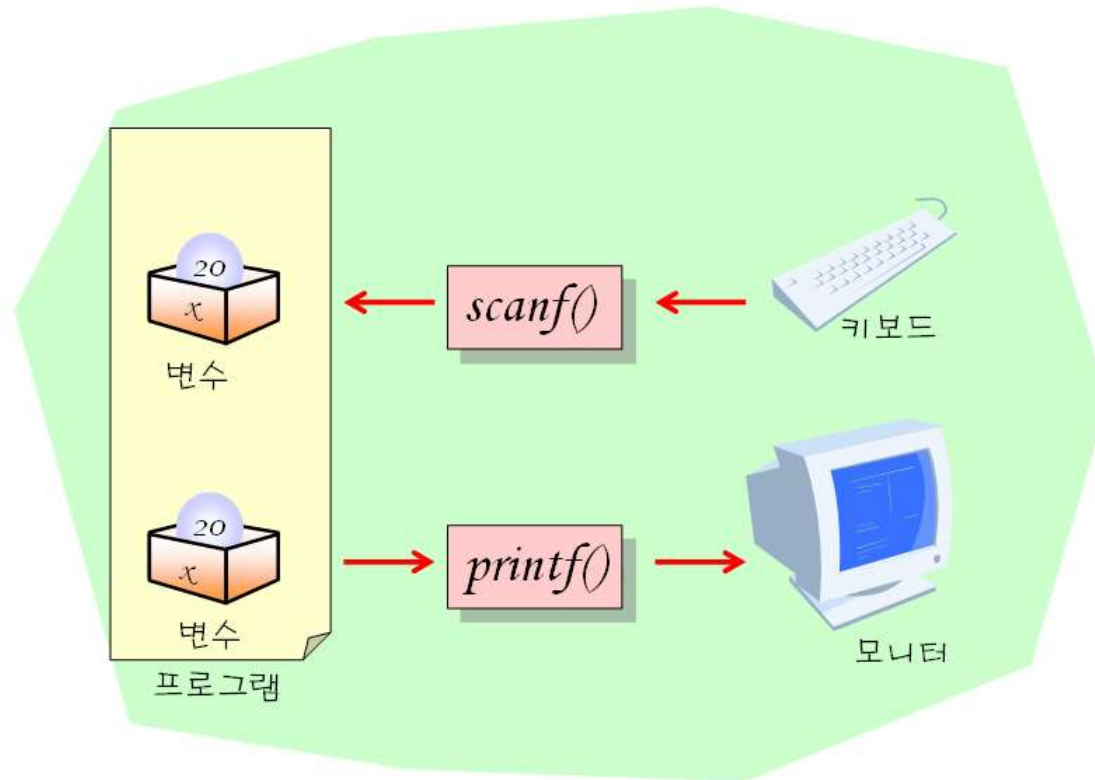




printf()

46

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



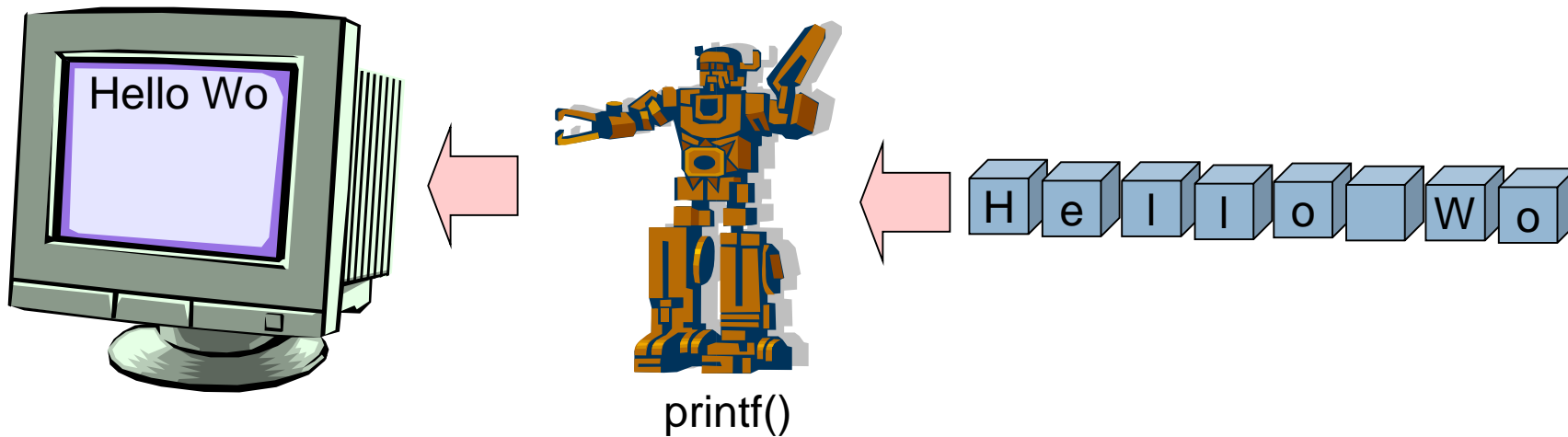


문자열 출력

47

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

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





변수값 출력

48

형식 제어 문자열

```
printf("두수의 합: %d", sum);
```

두수의 합: 30

형식 지정자의 개수와 변수의 개수와 순서는 같아야 합니다.





형식 지정자

49

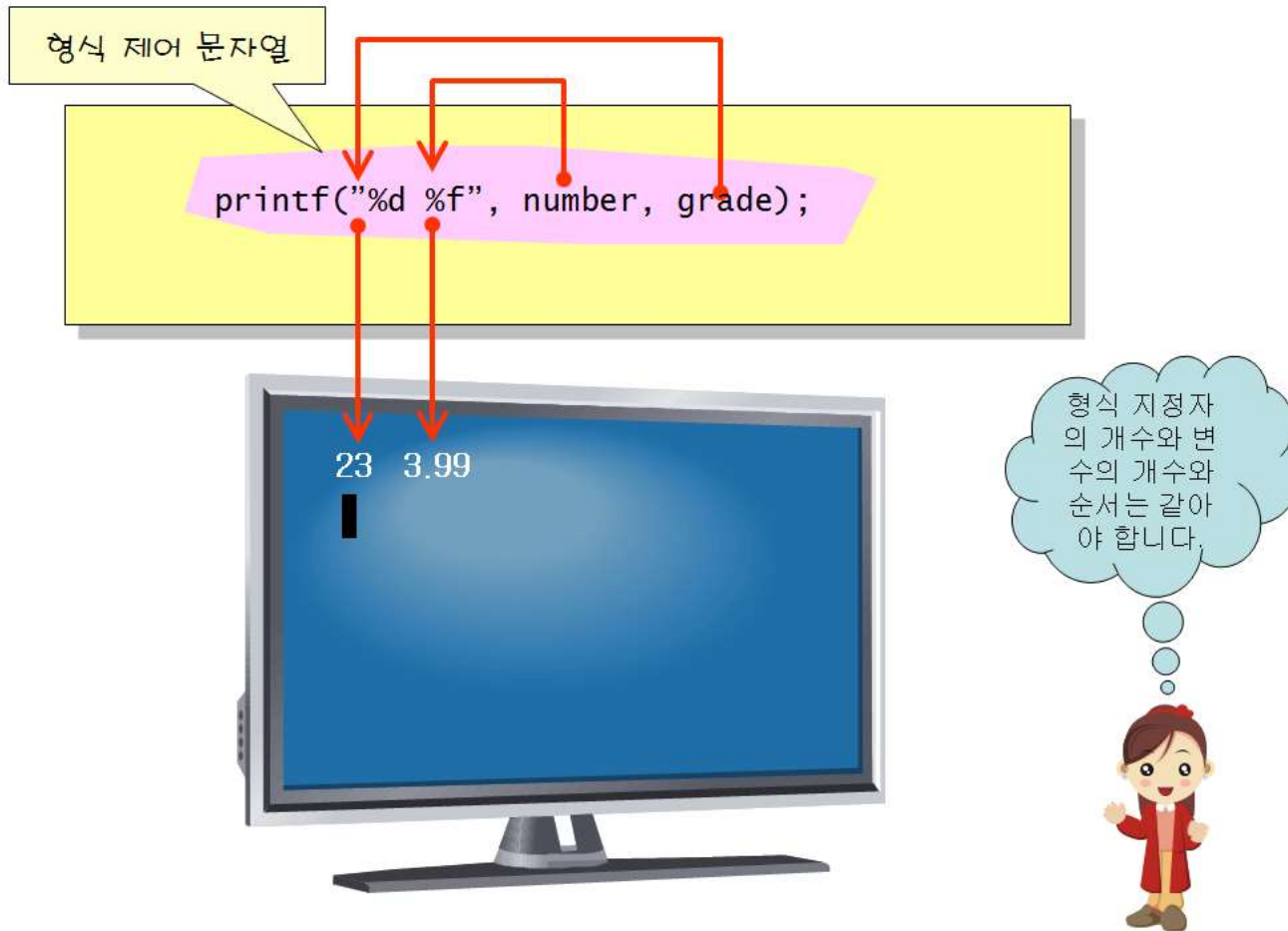
- 형식 지정자: printf()에서 값을 출력하는 형식을 지정한다.

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

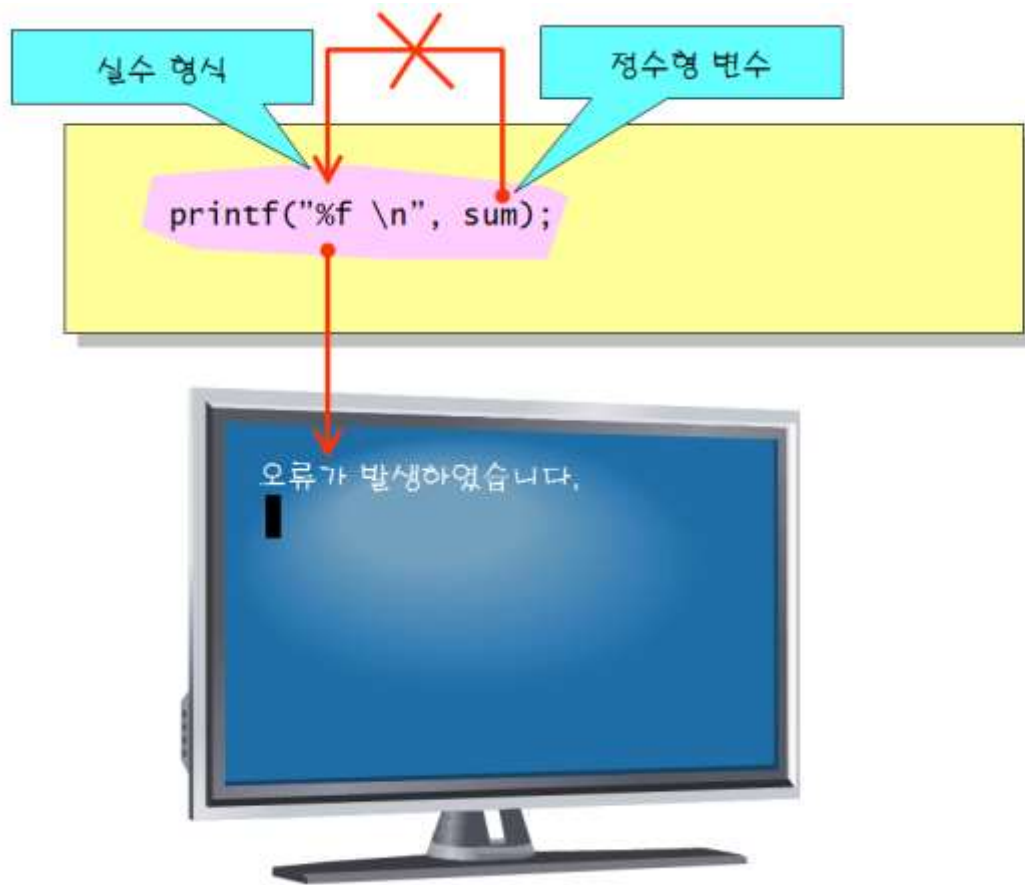
50





주의!

51





중간 점검

52

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





Lab: 사칙 연산

53

- 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.

📌 실행결과

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



Solution

calculation.c

54

```
// 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

55

```
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()

56

- 키보드로부터 값을 받아서 변수에 저장한다.
- 변수의 주소를 필요로 한다.

- 형식 지정자

- 값을 저장할 변수의 주소

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




주소가 필요한 이유

57





형식 지정자

58

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>



실수 입력시 주의할 점

59

- float 형은 %f 사용

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

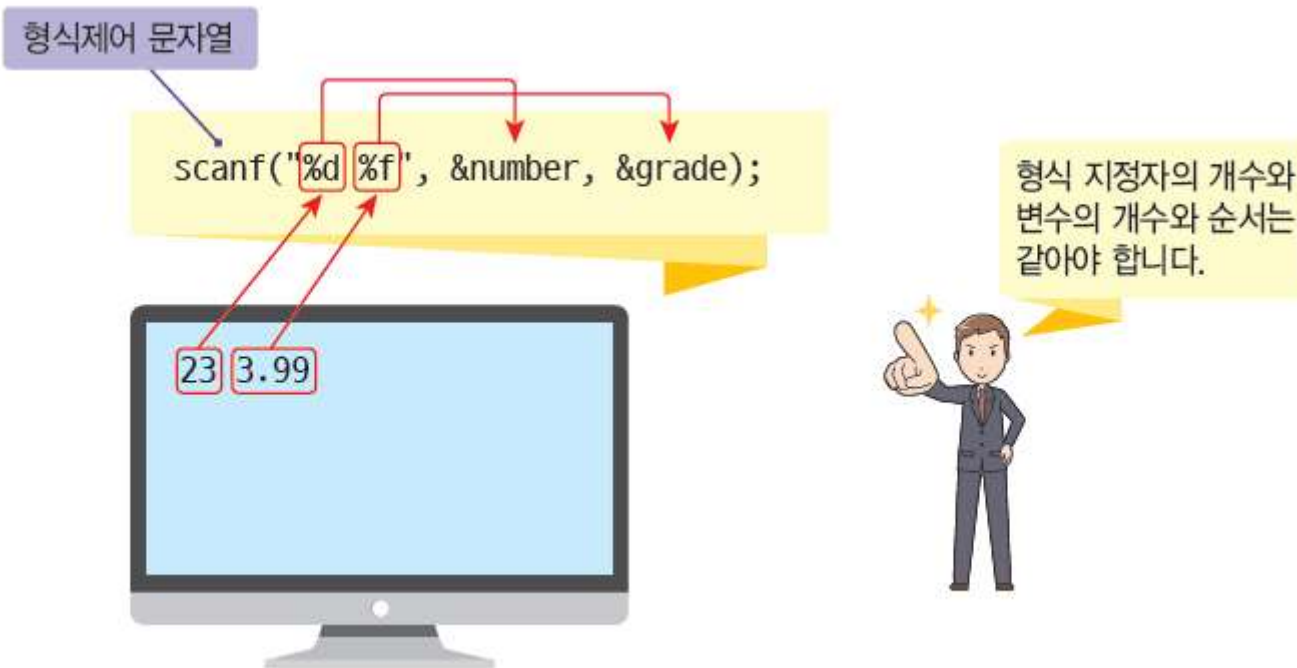
```
double scale = 0.0;  
scanf("%lf", &scale);
```

- double 형은 %lf 사용



scanf()

60





printf 보충설명

61

□ 우측정렬

- ▣ `x = 12;`
- ▣ `printf("x=%4d입니다.", x);`
 - `x= 12`입니다.
- ▣ `printf("x=%04d입니다.", x);`
 - `x=0012`입니다.

□ 좌측정렬

- ▣ `x = 12;`
- ▣ `printf("x=%-4d입니다.", x);`
 - `x=12` 입니다

□ 부동소숫점

- ▣ `y = 12.3456`
- ▣ `printf("%f", y)`
 - `12.345670`
- ▣ `printf("%.0f", y)`
 - `12`
- ▣ `printf("%.1f", y)`
 - `12.3`
- ▣ `printf("%10.3f", y)`
 - `12.346`
- ▣ `printf("%12.6f", y)`
 - `12.345670`



중간 점검

62

- scanf()를 사용하여 사용자로 부터 실수값을 받아서 double형의 변수 value에 저장하는 문장을 작성하여 보자.





덧셈 프로그램 #2

63

- 사용자로부터 입력을 받아보자.

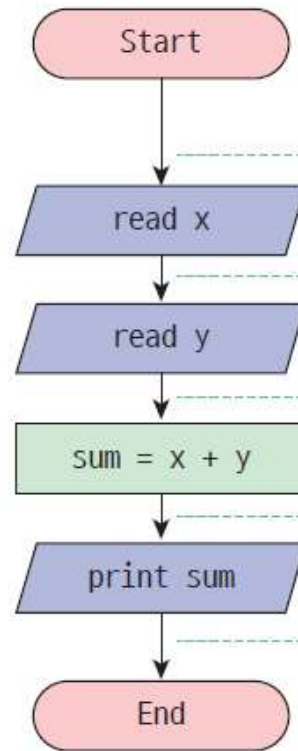
↓ 실행결과

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



알고리즘

64



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

    scanf("%d", &x);

    scanf("%d", &y);

    sum = x + y;

    printf("d", sum);

    return 0;
}
```




두번째 덧셈 프로그램

add2.c

65

```
// 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
}
```



연봉 계산기 프로그램

66

- 사용자가 연봉을 입력하면 월 수령액을 계산하는 프로그램을 작성해보자.

↓ 실행결과

연봉을 입력하시오(단위: 만원): 5000
월수령액(단위: 만원):416



두번째 덧셈 프로그램

salary.c

67

```
/* 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

68

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

📌 실행결과

반지름을 입력하시오: 10.0
원의 면적: 314.000000



원의 면적 계산 프로그램

circle.c

69

```
#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

70

- 사용자가 입력하는 원화를 달러화로 계산하여 출력하는 프로그램은 작성하여 보자.

▶ 실행결과

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



```
/* 환율을 계산하는 프로그램*/
```

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

```
exchange_rete.c
```

```
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
```

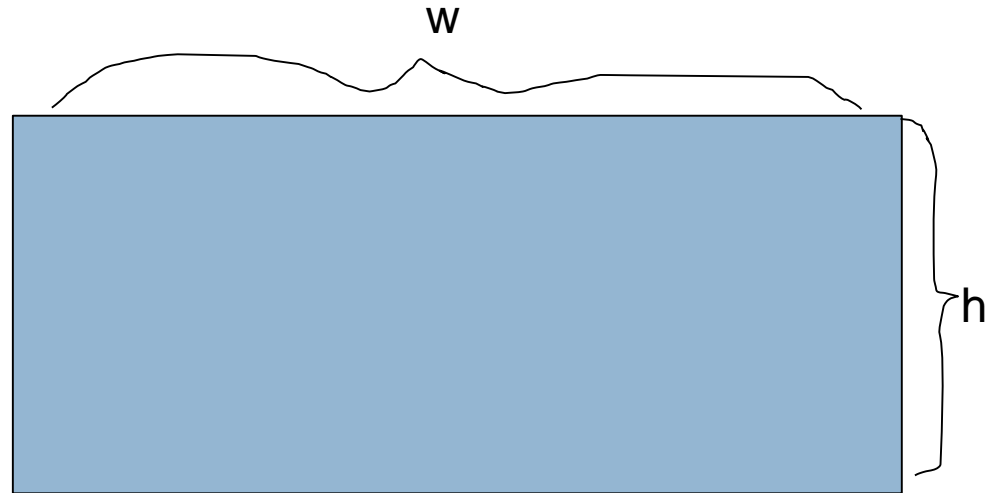
```
    return 0;                       // 함수 결과값 반환
```

```
}
```



mini project: 사각형의 둘레와 면적

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





프로그램의 실행 화면

73





```
#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: %1f\n", area);  
    printf("사각형의 둘레: %1f\n", perimeter);  
    return 0;
```

```
}
```





도전문제

75

1. 한번의 printf() 호출로 변수 perimeter와 area의 값이 동시에 출력되도록 변경하라.
2. 변수들을 한 줄에 모두 선언하여 보자.
3. w와 h의 값을 사용자로부터 받도록 변경하여 보자. %f를 사용한다.





Q & A

76

