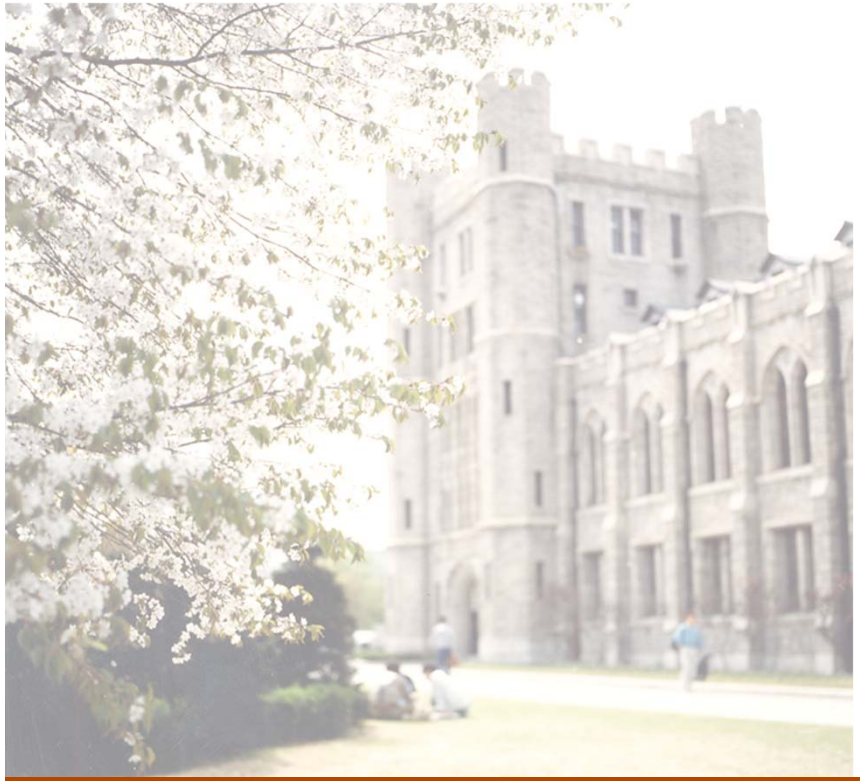
DAEGU UNIVERSITY

42경산 라피신(La piscine) 대비 사전 SW교육과정 - C



Heungwoo Nam

Computer Engineering
Daegu University
2023. 7. 13

Objective & Contents

❑ 수업목표

- 함수의 이해

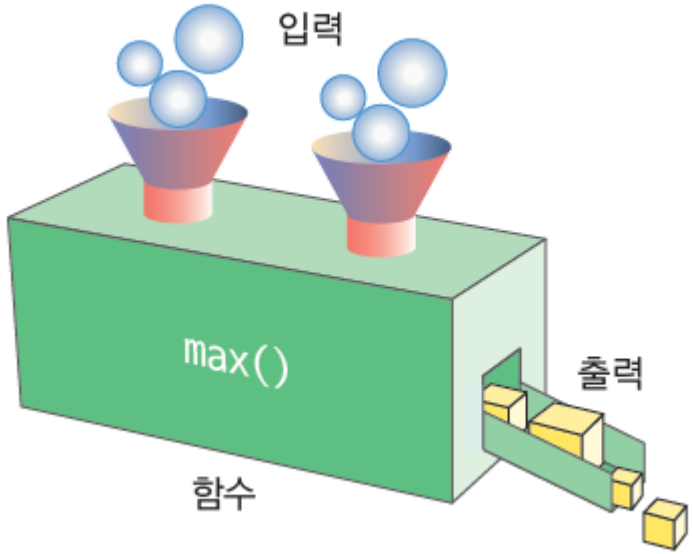
❑ Contents

- 8.1 함수란?
- 8.2 함수 정의
- 8.3 매개 변수와 반환값
- 8.4 함수 원형
- 8.5 라이브러리 함수 (난수)
- 8.6 라이브러리 함수 (수학 함수)

8.1 함수란?

❑ 함수 (function)의 개념

- 함수는 특정 작업을 수행하는 명령어들의 모음에 이름을 붙인 것
- 함수는 입력을 받아서 특정한 작업을 수행하여서 결과를 반환하는 블랙 박스(상자)와 같음
- 함수는 작업에 필요한 데이터를 전달받을 수 있으며, 작업이 완료된 후에는 작업의 결과를 호출자에게 반환함.



8.1 함수란?

❑ 함수가 필요한 이유

- 프로그램에서 되풀이되는 작업들이 있음.
  - 예를 들어서 30개의 *를 출력하는 작업이 필요하다고 하자
  - 이러한 코드가 프로그램 안의 여러 곳에서 사용된다고 하자


```
for(int i=0; i<30; i++)
        printf("*");
```

```
for(int i=0; i<30; i++)
        printf("*");
```

  - 함수를 이용하면 우리가 여러 번 반복해야 되는 처리 단계를 하나로 모아서 필요할 때 언제든지 호출하여 사용할 수 있음.

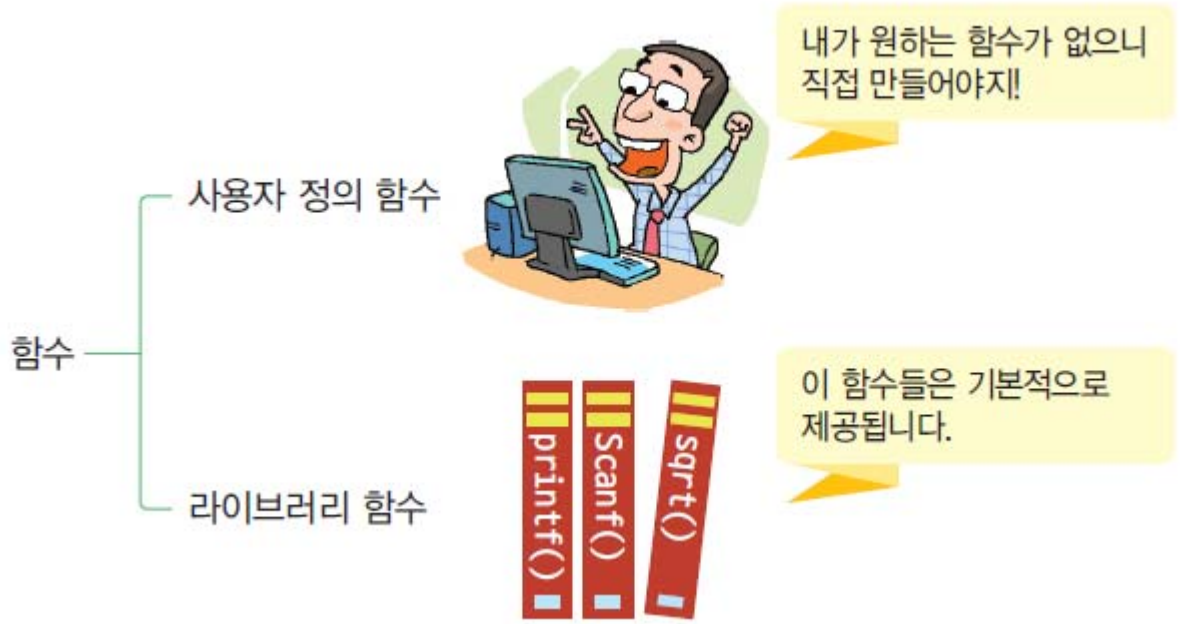

```
print_stars();

print_stars();
```

```
void print_stars()
{
        for(int i=0; i<30; i++)
            printf("*");
}
```

8.1 함수란?

❑ 함수의 종류

- 사용자 정의 함수 (user-defined function)
- 라이브러리 함수 (library function)



8.2 함수 정의

❑ 함수 정의

- 함수 헤더
  - 반환형과 함수 이름, 매개 변수를 합쳐서 지칭함
- 함수 몸체
  - 중괄호로 둘러싸인 부분으로 작업에 필요한 문장들이 들어감

Syntax: 함수 정의

예

```
void print_stars()
{
    for(int i=0; i<30; i++)
        printf("*");
}
```

반환형

함수 이름

매개 변수(현재는 없다)

함수 몸체

8.2 함수 정의

❑ 예제

- print_stars() 함수를 2번 호출하여서 다음과 같이 출력하는 프로그램을 작성해보자.

```
#include <stdio.h>

void print_stars()
{
        for (int i = 0; i < 30; i++)
                printf("*");
}
int main(void)
{
        print_stars();
        printf("\nHello World!\n");
        print_stars();
        printf("\n");
        return 0;
}
```

함수정의

함수호출

8.3 매개 변수와 반환값

❑ 매개 변수 및 반환값의 개념

Syntax: 함수의 구조

예

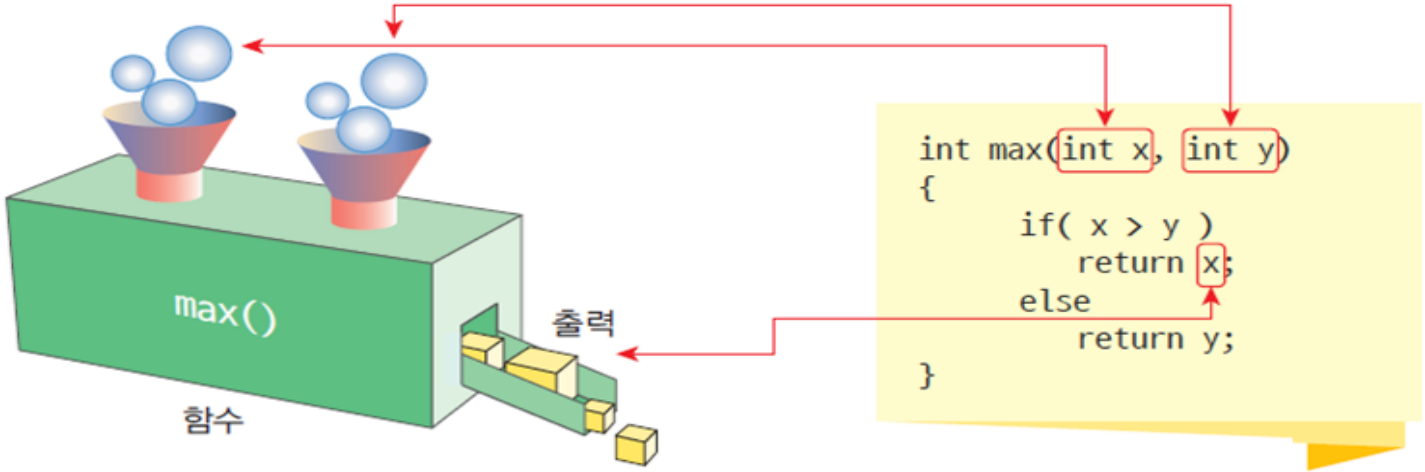
```
int max(int x, int y)
{
    if( x > y )
        return x;
    else
        return y;
}
```

함수 이름

반환형

매개 변수

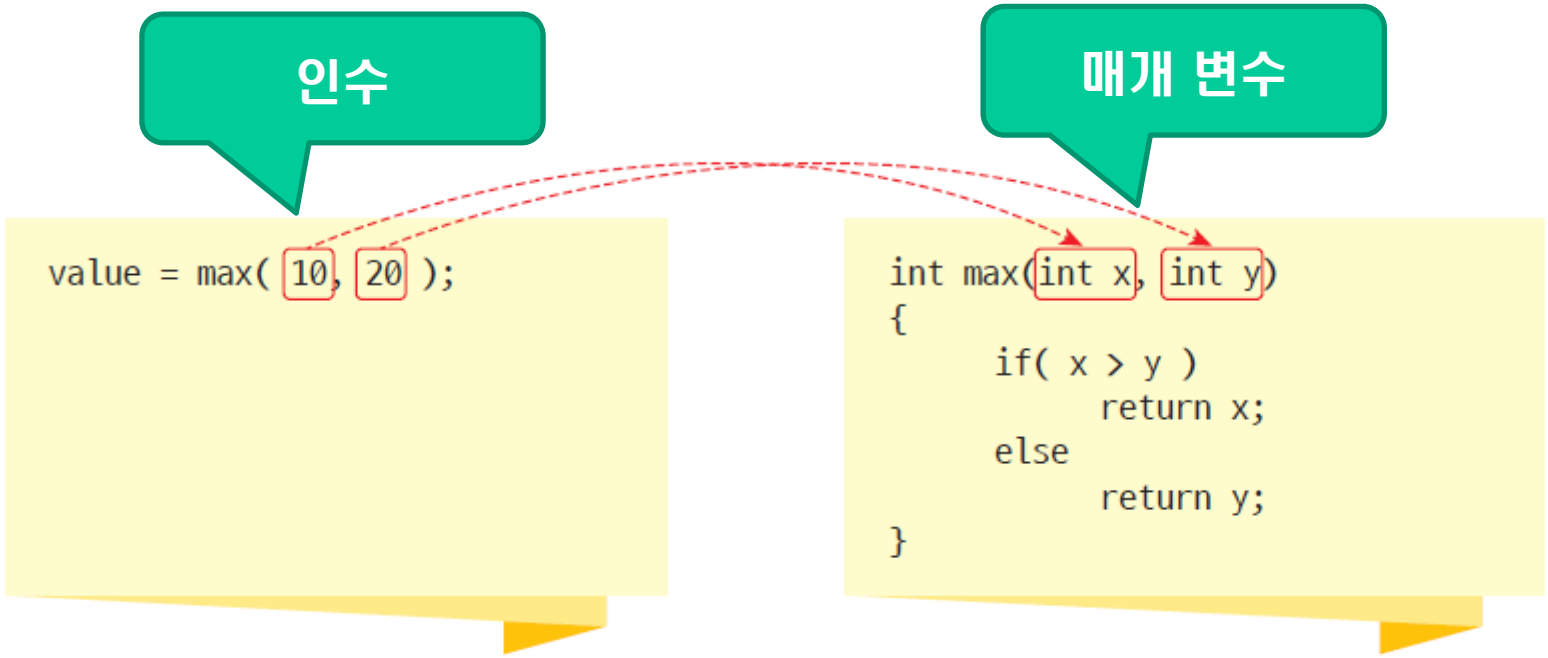
함수 몸체



8.3 매개 변수와 반환값

❑ 인수와 매개 변수

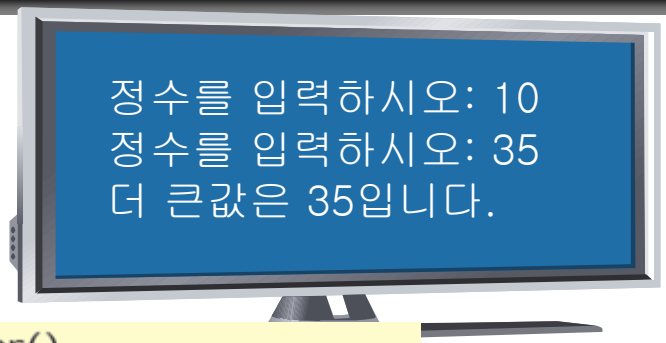
- 인수(argument): 호출 프로그램에 의하여 함수에 전달되는 값
- 매개 변수(parameter): 인수 값을 전달받는 변수
  - 함수 정의 시, 매개변수의 자료형과 이름을 지정하여야 함.
  - 매개변수가 없는 경우 void 또는 ()로 작성함.
  - 매개변수와 인수의 개수는 정확히 일치하여야 함.

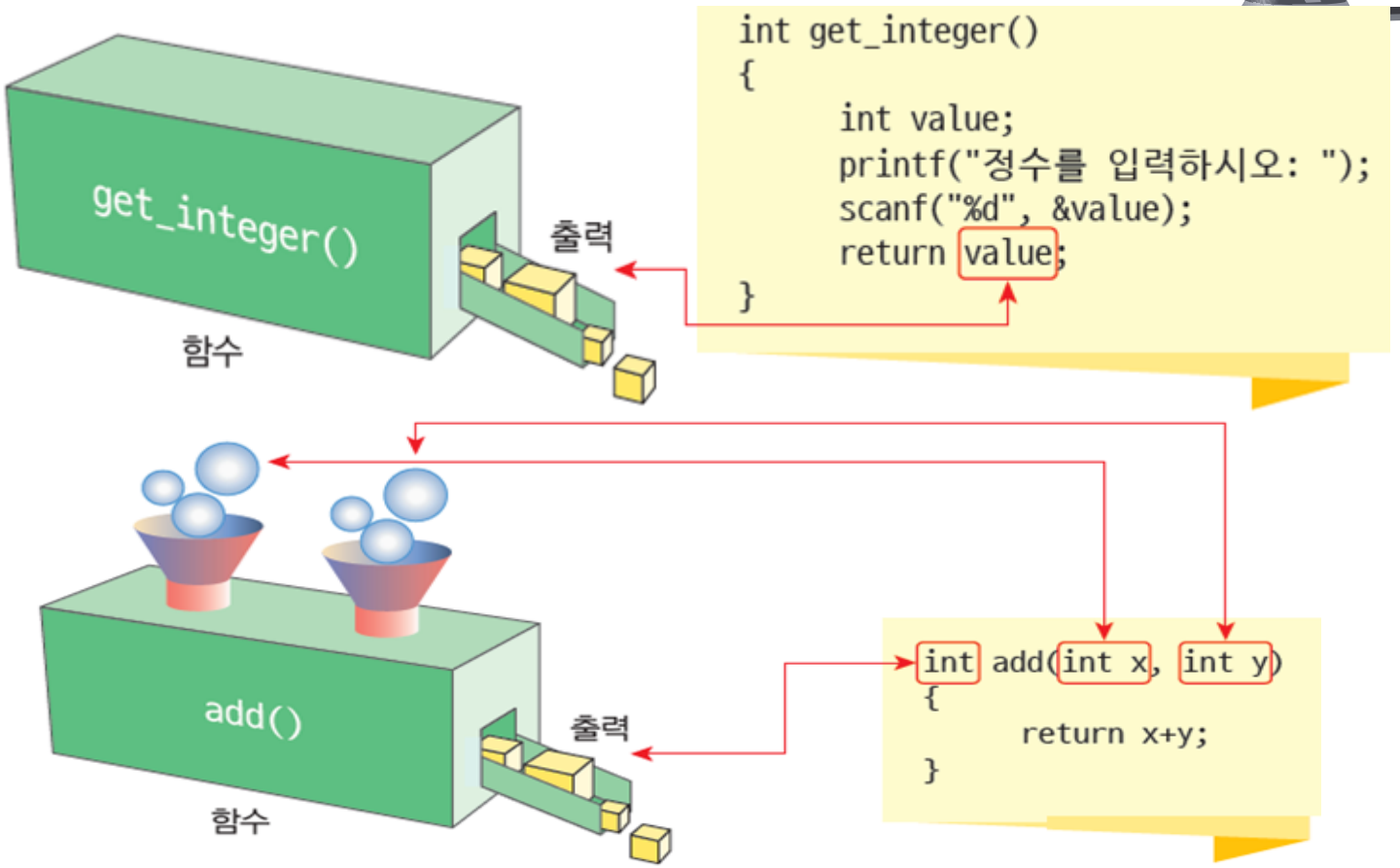


8.3 매개 변수와 반환값

❑ Lab

- 정수를 입력받는 get_integer() 함수
- 정수의 합을 계산하는 add() 함수





8.4 함수 원형

❑ 함수 원형(function prototyping)

- 함수 원형은 함수이름, 매개변수, 반환형을 함수가 정의되기 전에 컴파일러에게 함수에 대하여 미리 알리는 것
- 함수 원형은 함수 헤더에 세미콜론(;)을 추가한 것과 같음.

```
#include <stdio.h>
double c_to_f(double c_temp); // 함수 원형

int main(void)
{
        printf("섭씨 %lf도는 화씨 %lf입니다. \n", 36.0, c_to_f(36.0));
        return 0;
}
double c_to_f(double c_temp)
{
        return 9.0 / 5.0 * c_temp + 32;
}
```

8.5 라이브러리 함수(난수)

❑ 라이브러리 함수(library function):

: 컴파일러에서 제공하는 함수

- 표준 입출력
- 수학 연산
- 문자열 처리
- 시간 처리
- 오류 처리
- 데이터 검색과 정렬

8.5 라이브러리 함수(난수)

❑ 난수 함수

- 난수(random number): 규칙성이 없이 임의로 생성되는 수
  - 난수는 암호학이나 시뮬레이션, 게임 등에서 필수적임
- rand()
  - 의사난수(pseudo random number)를 생성하는 함수
    - 정말 다음에 뭐가 나올지 모르는 진짜 난수가 아니라 초기값에 따라서 나오는 순서가 어느 정도 결정되어 있는 난수
  - 0부터 RAND_MAX (32767)까지의 난수를 생성

8.5 라이브러리 함수(난수)

❑ 예제

- 로또 번호 생성하기
  - 1부터 45번 사이의 난수 발생

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
        int i;
        for(i = 0; i < 6; i++)
                printf("%d ", rand());
        return 0;
}
```

0에서 32767 사이의 정수로 생성

8.5 라이브러리 함수(난수)

❑ 예제

- 로또 번호 생성하기
  - 1부터 45번 사이의 난수 발생

```
printf("%d ", 1+(rand()%45));
```

➔ 문제점: 하지만 실행할 때마다 항상 똑같은 난수가 발생된다.

8.5 라이브러리 함수(난수)

❑ 예제

- 로또 번호 생성하기
  - 1부터 45번 사이의 난수 발생

➔ 해결책: 시드(seed) 개념 사용

시드: 씨앗이라는 의미로서 난수 생성시에 씨앗값이 되며,
시드값이 달라지면 이후 생성되는 모든 난수 값이 달라짐.

```
#include <stdlib.h> #include <stdio.h> #include <time.h>
#define MAX 45
int main( void )
{
        int i;

        srand( (unsigned)time( NULL ) );
        for( i = 0; i < 6; i++ )
                printf("%d ", 1+rand()%MAX );
        return 0;
}
```

-srand()는 난수 발생기의 시드 설정.
-시드를 설정하는 가장 일반적인 방법은 현재의 시각을 시드로 사용. 현재 시각은 실행할 때마다 달라지기 때문임.

8.6 라이브러리 함수(수학 함수)

❑ 수학 라이브러리 함수 (math.h)

분류	함수	설명
삼각함수	double sin(double x)	사인값 계산
	double cos(double x)	코사인값 계산
	double tan(double x)	탄젠트값 계산
역삼각함수	double acos(double x)	역코사인값 계산 결과값 범위 $[0, \pi]$
	double asin(double x)	역사인값 계산 결과값 범위 $[-\pi/2, \pi]$
	double atan(double x)	역탄젠트값 계산 결과값 범위 $[-\pi/2, \pi]$
쌍곡선함수	double cosh(double x)	쌍곡선 코사인
	double sinh(double x)	쌍곡선 사인
	double tanh(double x)	쌍곡선 탄젠트

8.6 라이브러리 함수(수학 함수)

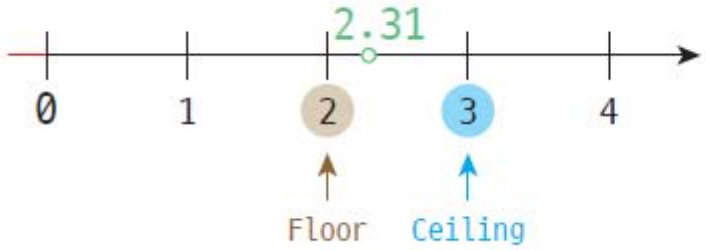
❑ 수학 라이브러리 함수 (math.h)

분류	함수	설명
지수함수	double exp(double x)	e^x
	double log(double x)	$\log_e x$
	double log10(double x)	$\log_{10} x$
기타함수	double ceil(double x)	x보다 작지 않은 가장 작은 정수
	double floor(double x)	x보다 크지 않은 가장 큰 정수
	double fabs(double x)	실수 x의 절대값
	int abs(int x)	정수 x의 절대값
	double pow(double x, double y)	x^y
	double sqrt(double x)	$\sqrt{x}$

8.6 라이브러리 함수(수학 함수)

❑ 수학 라이브러리 함수 (math.h)

- floor()와 ceil() 함수




```
double result, value = 1.6;

result = floor(value);                    // result는 1.00이다.
printf("%lf ", result);

result = ceil(value);                     // result는 2.00이다.
printf("%lf ", result);
```

8.6 라이브러리 함수(수학 함수)

❑ 수학 라이브러리 함수 (math.h)

- fabs() 함수

```
printf("12.0의 절대값은 %f\n", fabs(12.0));
printf("-12.0의 절대값은 %f\n", fabs(-12.0));   // result 12
```

- pow()와 sqrt()

```
printf("10의 3승은 %.0f.\n", pow(10.0, 3.0));   // result 1000
printf("16의 제곱근은 %.0f.\n", sqrt(16));       // result  4
```