

# 라이브러리 함수



라이브러리 함수의 종류

수학 라이브러리 함수

문자 및 문자열 처리 라이브러리 함수

난수 생성 라이브러리 함수

## 함수의 종류

```
#include <stdio.h>
int my_strlen(char *s);

int main(void) {
    char str[80];
    int length;

    printf("문자열을 입력하시오(80자 이내) : ");
    scanf_s("%s", str, 80);

    length = my_strlen(str);
    printf("입력 문자열의 길이 = %d\n", length);

    return 0; }

int my_strlen(char *s) {
    int i;

    i = 0;
    while (s[i] != '\0')
        i++;
    return i;
}
```

### 라이브러리 함수(library function)

-C 언어에서 미리 정의해서 제공하는 함수로,  
검증된 코드의 사용 및 개발 시간 절약의 장점 있음  
-필요한 헤더 파일을 명시해야 함  
예)  
printf와 scanf : <stdio.h> 필요

### 사용자 정의 함수(user-defined function)

프로그래머가 직접 작성해서 사용하는 함수  
예)  
문자열의 길이를 구하는 my\_strlen 함수

## 라이브러리 함수의 종류

- 수학 계산
- 문자(열) 입출력 및 처리
- 기타 난수 발생 등

## 수학 라이브러리 함수(1)

- 수학과 관련된 함수
- math.h 필요

예)

- 9 또는 -3.67의 절대값

`int abs(int x), double fabs(double x)`

- x의 절대값을 반환
- `abs(-9)`            `// 9를 반환`
- `fabs(-3.67)`        `// 3.67을 반환`

- $2^3$

`double pow(double x, double y)`

- x의 y 거듭제곱인  $x^y$  을 계산
- `pow(2.0, 3.0);`        `// 8.0을 반환`

- 9의 제곱근

`double sqrt(double x)`

- x의 제곱근을 구한다. (단, 음수 입력 시 오류 발생)
- `sqrt( 9.0 );`            `// 3.0을 반환`

## 수학 라이브러리 함수(2)

```
#include <stdio.h>
#include <math.h>
void main()
{
    //int abs(int x);
    //double fabs(double x);
    printf("%d\n", abs(-9));    // 9를 반환
    printf("%f\n", fabs(-3.67)) ;// 3.67을 반환

    double pow(double x, double y);
    printf("%f\n", pow(2.0, 3.0)); // 8.0을 반환

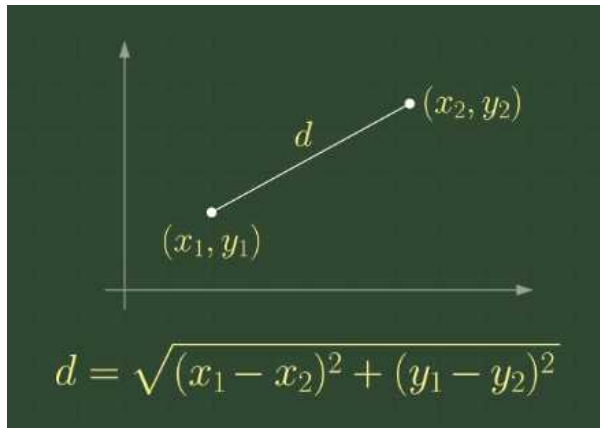
    double sqrt(double x);
    printf("%f\n", sqrt(9.0)); // 3.0을 반환
}
```

## [실습] 다음 조건을 만족하는 코드를 작성하시오

첫 번째 점의 X, Y좌표를 입력하세요: 0 0  
 두 번째 점의 X, Y좌표를 입력하세요: 10 10  
 두 점 사이의 거리 : 14.14

[조건]

- 수학 라이브러리 함수를 사용
- 두 점 사이의 거리 계산은 사용자 정의 함수를 사용하며,  
거리 계산 공식은 아래와 같다



## 문자 및 문자열 처리 입출력 라이브러리 함수

다음 프로그램의 결과는 무엇인가?  
(단, 입력은 다음과 같이 가정한다.)  
이름을 입력하세요 : **황보 혜정**

```
#include <stdio.h>
int main( )
{
    char name[10];

    printf("이름을 입력하세요 : ");
    scanf_s("%s", &name, sizeof(name));
    //scanf_s("입력받을 형태", &변수이름,
    //        입력받을 크기);
    printf("%s\n", name);

    return 0;
}
```

-scanf는 입력으로 공백 전까지의 자료를 받는다.  
-공백을 포함한 자료를 입력 받으려면,  
gets나 getchar와 같은 라이브러리 함수를 사용한다

이름을 입력하세요 : **황보 혜정**  
황보



## 문자 및 문자열 처리 입출력 라이브러리 함수

`int getchar(void)`

- 키보드로부터 하나의 문자를 입력 받아 반환

`void putchar(int c)`

- 변수 c에 저장된 문자를 화면에 출력

문자만 입출력 하는 경우,  
문자 전용 함수를 쓰는 것이 효율적

`char *gets(char *buf)`

- 키보드로부터 공백 포함 문자열을 입력 받아 buf에 저장
- 자동으로 NULL 문자( ' \0 ' )를 추가

`int puts(const char *str)`

- str이 가리키는 문자열을 받아서 화면에 출력
- 자동으로 줄바꿈 문자( '\n' ) 추가

[실습] 다음 조건을 만족하는 프로그램을 작성하시오

이름을 입력하세요 : 황보 혜정  
황보 혜정

[조건]

- 문자 입출력 라이브러리를 사용

## 문자 및 문자열 처리 라이브러리 함수(1)

- 문자를 검사하거나 문자를 변환한다.
- ctype.h 필요

함수	설명
isupper(c)	c가 대문자(A-Z) 이면 양수 반환, 아니면 0반환
islower(c)	c가 소문자(a-z) 이면 양수 반환, 아니면 0반환
isdigit(c)	c가 숫자(0-9) 이면 양수 반환, 아니면 0반환
isspace(c)	c가 공백문자(' ', '\n', '\t', '\v', '\r')이면 양수반환, 아니면 0반환

## [실습] 다음 조건을 만족하는 프로그램을 작성하시오

```
C:\Windows\system32\cmd.exe
암호를 생성하시오 123456
숫자, 소문자, 대문자를 섞어서 암호를 다시 만드세요!
암호를 생성하시오 abcdef
숫자, 소문자, 대문자를 섞어서 암호를 다시 만드세요!
암호를 생성하시오 ABCDEF
숫자, 소문자, 대문자를 섞어서 암호를 다시 만드세요!
암호를 생성하시오 abCD12
적당한 암호입니다.
계속하려면 아무 키나 누르십시오 . . .
```

### [조건]

- 5자~10자 사이의 암호 안에  
대문자, 소문자, 숫자가 1개 이상 모두 포함될 때까지 암호 생성
- 문자 단위로 입력 받으며, 문자 처리 라이브러리를 사용

## 문자 및 문자열 처리 라이브러리 함수(2)

- 문자열을 복사하거나 비교하는 등 문자열 처리를 한다.
- string.h 필요

함수	설명
strlen(s)	문자열 s의 '\0'을 제외한 길이를 구한다 ※ sizeof : 변수나 자료형에 대한 메모리 크기를 반환
strcpy_s(s1, sizeof(str1), s2)	문자열 s2를 문자열 s1에 복사한다
strcat_s(s1, sizeof(str1), s2)	문자열 s1의 끝에 문자열 s2를 붙여 넣는다
strcmp(s1, s2)	문자열 s1과 문자열 s2를 비교한다.

## [연습] 다음 코드의 결과를 예측하여 보고, 실행하여 비교하시오

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

int main(void)
{
    char str1[20];

    strcpy_s(str1, 6, "abcde"); // "abcde"를 str1에 복사
    printf("str1: %s   abcde: %s\n", str1, "abcde");

    if (str1 == "abcde")
        printf("두 문자열이 같습니다.\n");
    else
        printf("두 문자열이 다릅니다.\n");

    printf("sizeof: %d   strlen: %d\n", sizeof(str1), strlen("abcde"));

    return 0;
}
```

## [연습] 앞의 코드를 문자열의 내용 비교가 되도록 수정하여 실행하시오

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

int main(void)
{
    char str1[20];

    strcpy_s(str1, 6, "abcde");
    if (str1 == "abcde")
        printf("두 문자열이 같습니다.\n");
    else
        printf("두 문자열이 다릅니다.\n");

    return 0;
}
```

## 수정된 내용

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

int main(void)
{   char str1[20];
    int m;

    strcpy_s(str1, 6, "abcde");
    printf("str1: %s   abcde: %s\n", str1, "abcde");
    m = strcmp(str1, "abcde");
    printf("m: %d \n", m);
    if ( m )
        printf("두 문자열이 다릅니다.\n");
    else
        printf("두 문자열이 같습니다.\n");

    printf("sizeof: %d   strlen: %d\n", sizeof(str1), strlen("abcde"));

    return 0;
}
```



[연습] 다음 코드의 결과를 예측하여 보고, 실행하여 비교하십시오

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

int main(void)
{
    char str1[20]= "abcde";

    printf("str1: %s   strlen: %d\n\n", str1, strlen(str1));

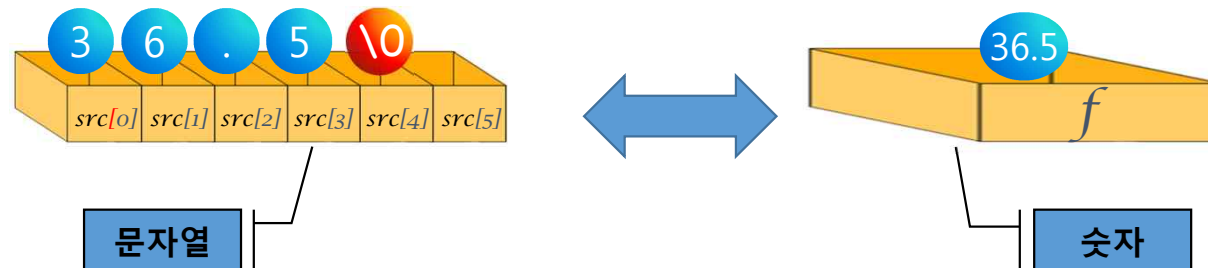
    printf("두 문자열 결합하기\n");
    strcat_s(str1, 11, "fghij");
    printf("str1: %s   strlen: %d\n", str1, strlen(str1));

    return 0;
}
```

## 문자열을 숫자로 변환해 주는 함수(1)

“36.5”+“34” 의 결과는 무엇일까 ?

- 컴파일 에러 발생!
- “36.5”와 같은 문자열은 더하기, 빼기와 같은 계산을 할 수 없다.
  - 문자열 “36.5”와 숫자 36.5는 컴퓨터에서 저장되는 형태가 다르기 때문

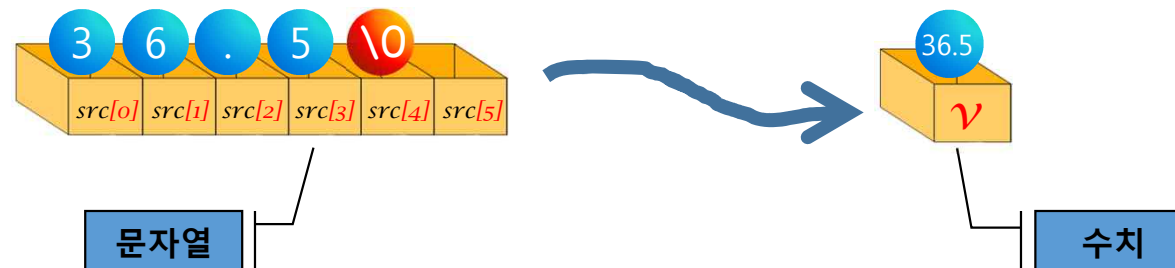


- 문자열을 숫자로 변환해주는 함수를 사용하여 계산할 수 있다

## 문자열을 숫자로 변환해 주는 함수(2)

- 문자열을 숫자로 변환한다
- `stdlib.h` 포함

함수	설명
<code>int atoi( const char *str );</code>	str을 int형으로 변환한다.
<code>long atoi( const char *str );</code>	str을 long형으로 변환한다.
<code>double atof( const char *str );</code>	str을 double형으로 변환한다.



## 생각해보기

- 초등학생 대상의 덧셈과 뺄셈 연습 프로그램을 작성하려고 한다.
  - 문제를 낼 때마다 계산하는 두 수가 달라지게 하려면?
- 두 개의 주사위를 던져 나오는 값의 합을 알아 맞추는 게임 프로그램을 작성하려고 한다.
  - 게임 때마다 두 주사위의 값을 다르게 지정하려면?
- 컴퓨터와 사용자간 가위바위보 게임을 하는 프로그램을 작성하려고 한다.
  - 게임 때마다 컴퓨터가 가위바위보 중 어떤 것을 낼지 사용자가 예측할 수 없도록 하려면?
- 난수(random number)를 활용
  - 특정한 순서나 규칙을 가지지 않는 연속적인 임의의 수  
예) 로또 복권은 1~45 사이의 예측할 수 없는 임의 숫자 6개의 당첨 번호
  - 게임 프로그램, 암호학 등에서 필수적으로 사용

## 난수 생성 라이브러리 함수(1)

- rand 함수
  - 무작위로 선택된 임의의 값을 한 개 반환하는 함수
  - int rand(void)
    - <stdlib.h> 필요
    - 반환 값은 0~32767 범위 안의 임의의 정수

예) 주사위 값 die를 rand 함수를 사용하여 1~6 중 한 개 값이 되게 하려면?

die = rand() % 6 + 1;

0 ~ 5



- 스케일링(scaling)
  - rand 함수의 반환 값을 원하는 값의 범위로 조정하는 작업

## 난수 생성 라이브러리 함수(2)

- 프로그램이 실행될 때 마다 다른 난수가 생성되기 위해서 srand 함수와 time 함수가 필요

srand( time(NULL) )

- 현재 컴퓨터의 시간을 반환,  
- <time.h> 필요
- [srand](#)에 다른 seed 값을 전달하면 [rand](#) 함수로부터 생성되는 난수 값도 매번 전혀 다르게 됨
- 난수 값의 seed를 [현재 컴퓨터의 시간으로 설정](#), 프로그램 실행 때마다 시간이 바뀌므로 난수 값도 바뀜
- <stdlib.h> 필요

[연습] 다음 코드를 실행하여 그 결과를 확인하시오

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main()
{
    int i, random;

    srand(time(NULL));

    for (i = 1; i <= 5; i++)
    {
        random = rand();
        printf("%d번째 난수: %d \n", i, random);
    }

    return 0;
}
```

## [프로젝트] 다음 조건을 만족하는 프로그램을 작성하시오

>> 컴퓨터가 숨긴 수를 맞추는 게임입니다.

```
=====
1 ~ 100 중의 값입니다. 얼마일까요? 50
51 ~ 100 중의 값입니다. 얼마일까요? 90
51 ~ 89 중의 값입니다. 얼마일까요? 83
=====
```

컴퓨터가 숨긴 83를 3번 만에 맞혔습니다.

### [조건]

- 컴퓨터가 숨기고 있는 수는 1~100 중 하나로 프로그램 실행 시 마다 다름
- 사용자 입력에 따라 숨기고 있는 수의 범위를 표시





 T h a n k y o u

## TECHNOLOGY

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Velit ex  
aliqua ipsum, labore sed tempore ratione asperiores des  
consecteture bore sed Tempora rati igert one bore sed teni