

문자와 문자열 Part 2

문자 처리 라이브러리 함수

■ 문자를 검사하거나 문자를 변환한다.

함수	설명
isalpha(c)	c가 영문자인가?(a-z, A-Z)
isupper(c)	c가 대문자인가?(A-Z)
islower(c)	c가 소문자인가?(a-z)
isdigit(c)	c가 숫자인가?(0-9)
isalnum(c)	c가 영문자이나 숫자인가?(a-z, A-Z, 0-9)
isxdigit(c)	c가 16진수의 숫자인가?(0-9, A-F, a-f)
isspace(c)	c가 공백문자인가?(' ', '\n', '\t', '\v', '\r')
ispunct(c)	c가 구두점 문자인가?
isprint(c)	C가 출력가능한 문자인가?
iscntrl(c)	c가 제어 문자인가?
isascii(c)	c가 아스키 코드인가?

함수	설명
toupper(c)	c를 대문자로 바꾼다.
tolower(c)	c를 소문자로 바꾼다.
toascii(c)	c를 아스키 코드로 바꾼다.

예제

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

```
int main( void )
```

```
{
```

```
    int c;
```

```
    while((c = getchar()) != EOF)
```

```
    {
```

```
        if( islower(c) )
```

```
            c = toupper(c);
```

```
        putchar(c);
```

```
    }
```

```
    return 0;
```

```
}
```

소문자인지 검사
대문자로 변환

```
abcdef
ABCDEF
^Z
```

예제

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

int main( void )
{
    int c;

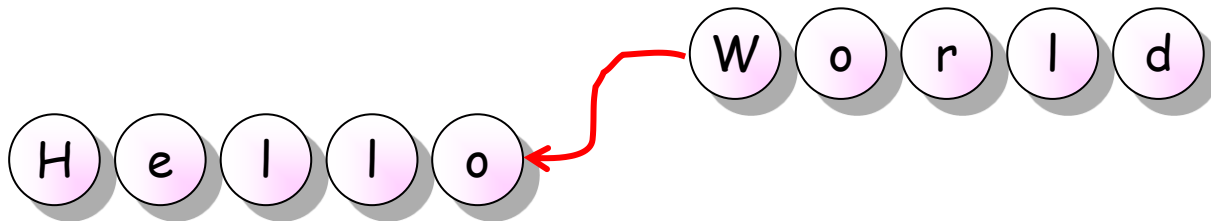
    while((c = getch()) != 'z')
    {
        printf("-----\n");
        printf("isdigit(%c) = %d\n", c, isdigit(c));
        printf("isalpha(%c) = %d\n", c, isalpha(c));
        printf("islower(%c) = %d\n", c, islower(c));
        printf("ispunct(%c) = %d\n", c, ispunct(c));
        printf("isxdigit(%c) = %d\n", c, isxdigit(c));
        printf("isprint(%c) = %d\n", c, isprint(c));
        printf("-----\n\n");
    }
    return 0;
}
```

```
-----
isdigit('') = 0
isalpha('') = 0
islower('') = 0
ispunct('') = 16
isxdigit('') = 0
isprint('') = 16
-----
```

...

문자열 처리 라이브러리

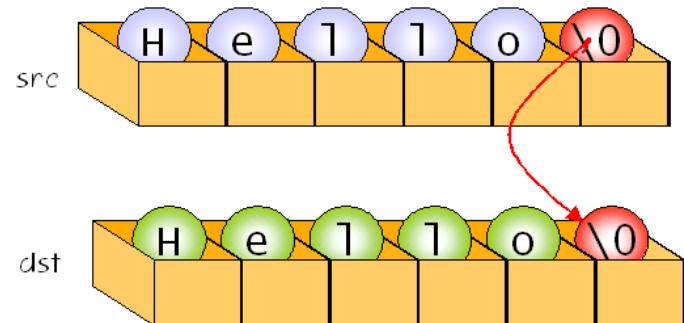
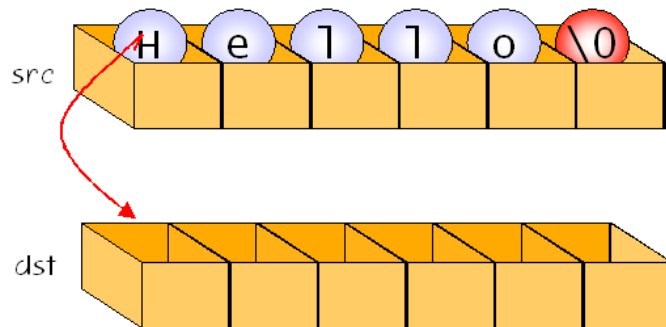
함수	설명
strlen(s)	문자열 s의 길이를 구한다.
strcpy(dst, src)	s2를 s1에 복사한다.
strcat(s1, s2)	s2를 s1의 끝에 붙여넣는다.
strcmp(s1, s2)	s1과 s2를 비교한다.
strncpy(dst, src, n)	s2의 최대 n개의 문자를 s1에 복사한다.
strncat(s1, s2, n)	s2의 최대 n개의 문자를 s1의 끝에 붙여넣는다.
strncmp(s1, s2, n)	최대 n개의 문자까지 s1과 s2를 비교한다.
strchr(s, c)	문자열 s안에서 문자 c를 찾는다.
strstr(s1, s2)	문자열 s1에서 문자열 s2를 찾는다.



문자열 길이 strlen(), 복사 strcpy()

- 문자열의 길이를 구하는 라이브러리 함수
 - strlen("Hello")는 5를 반환
- 문자열을 복사하는 라이브러리 함수

```
char dst[6];  
char src[6] = "Hello";  
strcpy(dst, src);
```



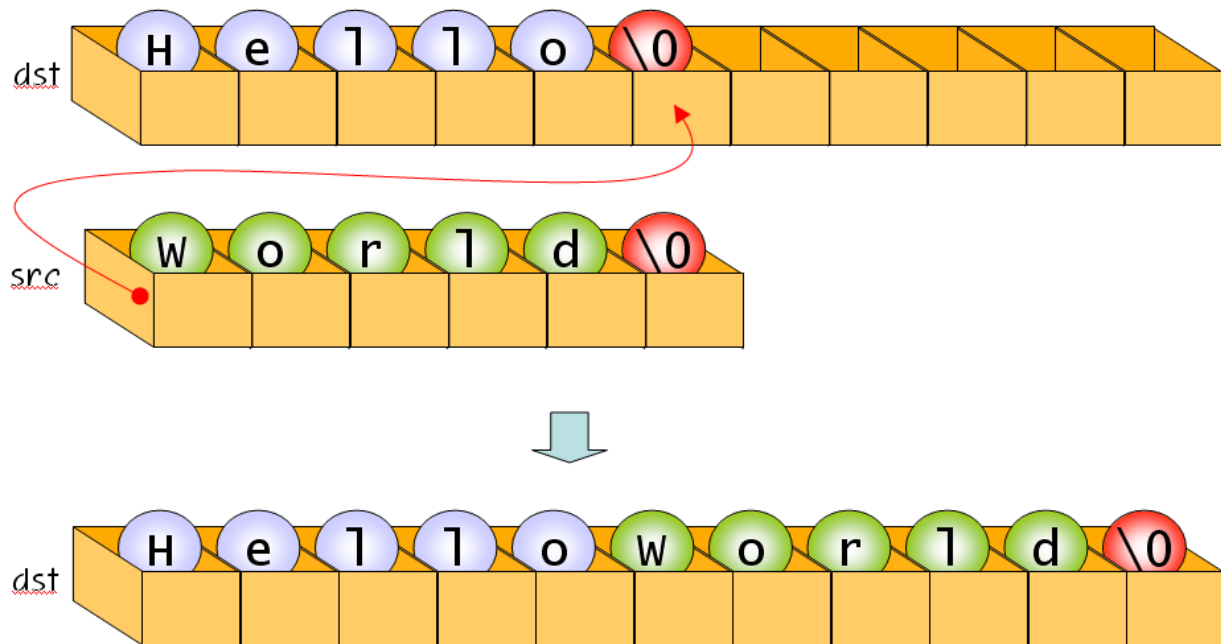
문자열 연결 strcat()

■ 문자열 연결

```
char dst[12] = "Hello";
```

```
char src[6] = "World";
```

```
strcat(dst, src);
```



예제

```
// strcpy와 strcat
#include <string.h>
#include <stdio.h>

int main( void )
{
    char string[80];

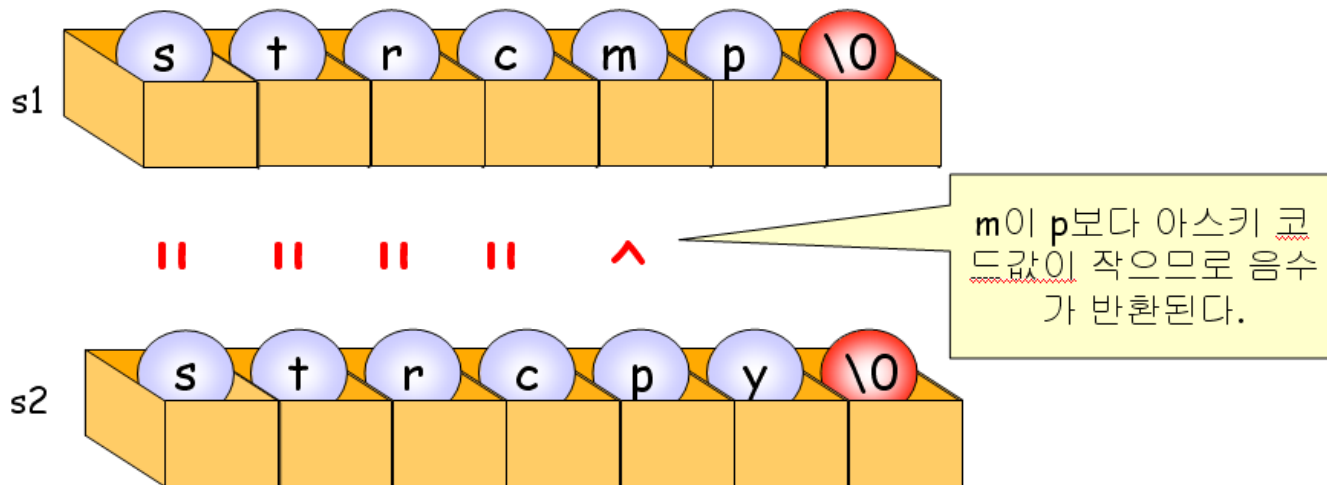
    strcpy( string, "Hello world from " );
    strcat( string, "strcpy " );
    strcat( string, "and " );
    strcat( string, "strcat!" );
    printf( "string = %s\n", string );
    return 0;
}
```

string = Hello world from strcpy and strcat!

문자열 비교 strcmp()

```
int strcmp( const char *s1, const char *s2 );
```

반환값	s1과 s2의 관계
-	s1이 s2보다 작다
0	s1이 s2와 같다.
+	s1이 s2보다 크다.



예제

```
// strcmp() 함수
#include <string.h>
#include <stdio.h>

int main( void )
{
    char s1[80];          // 첫번째 단어를 저장할 문자배열
    char s2[80];          // 두번째 단어를 저장할 문자배열
    int result;

    printf("첫번째 단어를 입력하시오:");
    scanf("%s", s1);
    printf("두번째 단어를 입력하시오:");
    scanf("%s", s2);

    result = strcmp(s1, s2);
    if( result < 0 )
        printf("%s가 %s보다 앞에 있습니다.\n", s1, s2);
    else if( result == 0 )
        printf("%s가 %s와 같습니다.\n", s1, s2);
    else
        printf("%s가 %s보다 뒤에 있습니다.\n", s1, s2);
    return 0;
}
```

첫번째 단어를 입력하시오:Hello
두번째 단어를 입력하시오:World
Hello가 World보다 앞에 있습니다.

문자 검색 strchr(), 문자열 검색 strstr()

■ 문자열에서 문자 검색

```
char s[] = "language"; // 문자열
char c = 'g';           // 찾고자 하는 문자
char *p;                // 문자 포인터

p = strchr(s, c);       // s에서 c를 찾는다.
```

■ 문자열에서 문자열 검색

```
char s[] = "A joy that's shared is a joy made double"; // 입력 문자열
char sub[] = "joy"; // 찾으려고 하는 문자열
char *p; // 문자 검색 위치 저장 포인터

p = strstr(s, sub); // s에서 sub를 찾는다.
```

문자열 토큰 분리 strtok()

■ 문자열을 토큰으로 분리

```
// strtok 함수의 사용예
#include <string.h>
#include <stdio.h>

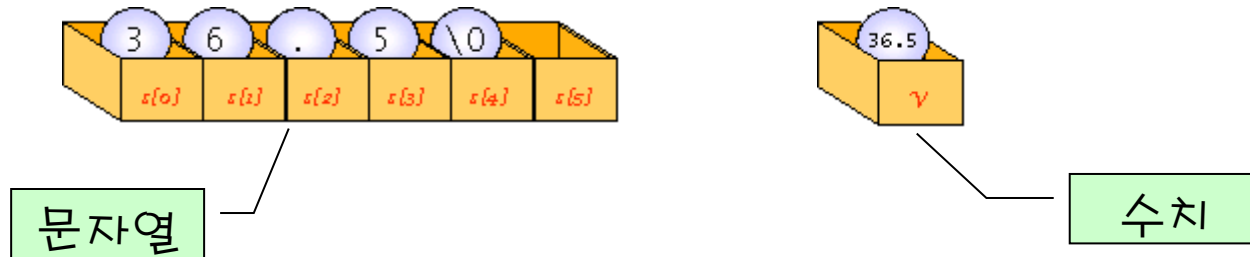
char s[] = "Man is immortal, because he has a soul";
char seps[] = " ,\t\n";
char *token;

int main( void )
{
    // 문자열을 전달하고 다음 토큰을 얻는다.
    token = strtok( s, seps );
    while( token != NULL )
    {
        // 문자열 s에 토큰이 있는 동안 반복한다.
        printf( "토큰: %s\n", token );
        // 다음 토큰을 얻는다.
        token = strtok( NULL, seps ); //
    }
}
```

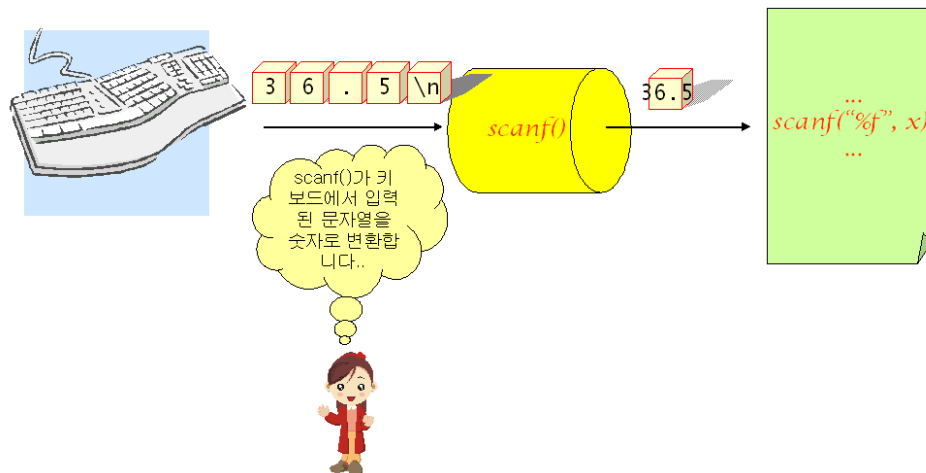
토큰: Man
토큰: is
토큰: immortal
토큰: because
토큰: he
토큰: has
토큰: a
토큰: soul

문자열 수치 변환

■ 문자열과 수치



- `scanf()` 함수는 문자열을 수치로 변환한다.



문자열을 수치로 변환하는 함수

- 전용 함수는 scanf()보다 크기가 작다.
- stdlib.h에 원형 정의- 반드시 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.

예

```
// atoi() 함수
#include <stdio.h>
#include <stdlib.h>

int main( void )
{
    char s[30];
    char t[] = "36.5";
    int i;
    double v;

    printf("정수를 입력하시오:");
    gets(s);
    i = atoi(s);
    printf("입력된 정수: %d \n", i);

    v = atof(t);
    printf("변환된 실수: %f", v);

    return 0;
}
```

정수를 입력하시오:89
입력된 정수: 89
변환된 실수: 36.500000

sscanf(), sprintf()

함수	설명
sscanf(s,...)	문자열 s로부터 지정된 형식으로 수치를 읽어서 변수에 저장한다.
sprintf(s,...)	변수의 값을 형식 지정자에 따라 문자열 형태로 문자 배열 s에 저장한다.

```
int main( void )
{
    char s1[] = "100";
    char s2[] = "12.93";
    char buffer[100];

    int i;
    double d;
    double result;

    sscanf(s1, "%d", &i);
    sscanf(s2, "%lf", &d);

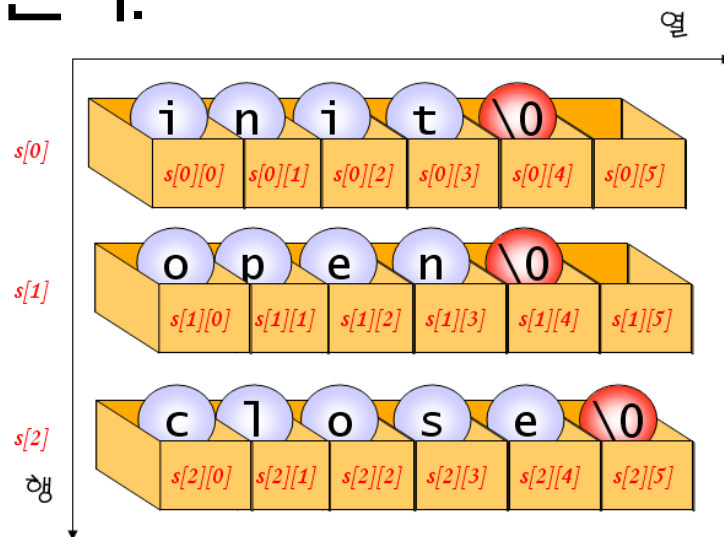
    result = i + d;
    sprintf(buffer, "%f", result);
    printf("연산 결과는 %s입니다.\n", buffer);
    return 0;
}
```

연산 결과는 112.930000입니다.

문자열의 배열

- (Q) 문자열이 여러 개 있는 경우에는 어떤 구조를 사용하여 저장하면 제일 좋을까?
 - (A) 여러 개의 문자 배열을 각각 만들어도 되지만 **문자열 배열**을 만드는 것이 여러모로 간편하다.
- 문자열 자체가 1차원 배열이므로, 문자열 배열은 배열의 배열, 즉 2차원 문자 배열이 된다.

```
char s[3][6] = {  
    "init",  
    "open",  
    "close"  
};
```



메뉴 디스플레이

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

```
int main( void )
```

```
{
```

```
    int i;
```

```
    char menu[5][10] = {
```

```
        "init",
```

```
        "open",
```

```
        "close",
```

```
        "read",
```

```
        "write"
```

```
    };
```

```
    for(i = 0; i < 5; i++)
```

```
        printf("%d 번째 메뉴: %s \n", i, menu[i]);
```

```
    return 0;
```

```
}
```

0 번째 메뉴: init

1 번째 메뉴: open

2 번째 메뉴: close

3 번째 메뉴: read

4 번째 메뉴: write

메뉴 선택

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

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

```
    int i;  
    char buffer[10];  
    char menu[5][10] = {  
        "init",  
        "open",  
        "close",  
        "read",  
        "write"  
    };
```

```
    printf("메뉴를 입력하시오:");  
    scanf("%s", buffer);
```

```
    for(i = 0; i < 5; i++)  
        if( strcmp(buffer, menu[i]) == 0 )  
            printf("%d번째 메뉴를 입력하였습니다.\n", i);
```

```
    return 0;
```

```
}
```

메뉴를 입력하시오:open
1번째 메뉴를 입력하였습니다.

단어 카운팅

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

int count_word(const char *s);

int main( void )
{
    printf("%d\n", count_word("the c book..."));

    return 0;
}

int count_word (const char * s)
{
    int i, wc = 0, waiting = 1;

    for( i = 0; s[i] != NULL; ++i)
        if( isalpha(s[i]) )
        {
            if( waiting )
            {
                wc++;
                waiting = 0;
            }
        }
        else
        {
            waiting = 1;
        }

    return wc;
}
```

문자열 비교

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

int strncmp(const char *s1, const char *s2, int count);

int main( void )
{
    printf("%d\n", strcmp("language C++", "language C", 5));

    return 0;
}

// returns <0 if s1 < s2
// returns 0 if s1 == s2
// returns >0 if s1 > s2
int strcmp (const char * s1, const char * s2, int count)
{
    if (!count)
        return(0);

    while (--count && *s1 && *s1 == *s2)
    {
        s1++;
        s2++;
    }

    return( *s1 - *s2 );
}
```

한영 사전 구현

```
#define ENTRIES 5

int main( void )
{
    int i, index;
    char dic[ENTRIES][2][30] = {
        {"book", "책"},
        {"boy", "소년"},
        {"computer", "컴퓨터"},
        {"lanuguage", "언어"},
        {"rain", "비"},
    };
    char word[30];

    printf("단어를 입력하시오:");
    scanf("%s", word);

    index = 0;
    for(i = 0; i < ENTRIES; i++)
    {
        if( strcmp(dic[index][0], word) == 0 )
        {
            printf("%s: %s\n", word, dic[index][1]);
            return 0;
        }
        index++;
    }
    printf("사전에서 발견되지 않았습니다.\n");
}
```

문자열 → 정수

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

int stoi( const char *s );

int main(void)
{
    printf("%d\n", stoi("-123"));
}

int stoi( const char *s )
{
    int c;           // 현재의 글자
    int total =0;    // 현재의 합계
    int sign;

    c = *s++;
    sign = c;        // 부호를 저장한다.
    if (c == '-' || c == '+')
        c = *s++;    // 부호를 제거한다.

    while (isdigit(c)){
        total = 10 * total + (c - '0'); // 누적시킨다.
        c = *s++;    // 다음 글자를 얻는다.
    }
    if (sign == '-')
        return -total;
    else
        return total; // 필요하면 음수로 만든다.
}
```