11장 문자

한빛미디어 Hanbit Media, Inc.

11-1 아스키 코드 값과 문자 입출력 함수

❖ 아스키 코드 (1/2)

- 128개의 문자에 대해 서로 다른 값을 정해놓은 약속

종류	문자 상수	아스키 코드 값	출력할 때
숫자 문자 (10개)	'0' ~ '9'	48 ~ 57	문자 출력
대문자 (26개)	'A' ~ 'Z'	65 ~ 90	문자 출력
소문자 (26개)	'a' ~ 'z'	97 ~ 122	문자 출력
특수 문자 (33개)	' '(공백), '$', '&' …	32, 36, 38 …	문자 출력
제어 문자 (33개)	'\0', '\t', '\n', '\r' ...	0, 9, 10, 13 …	제어 기능 수행

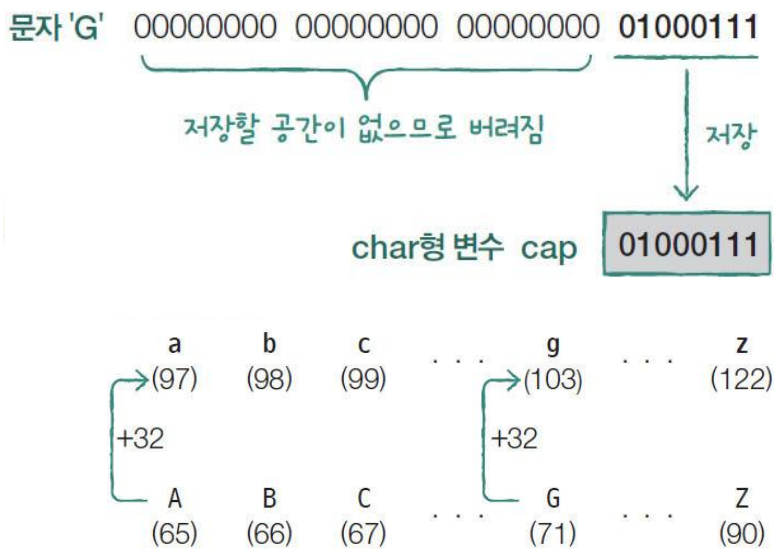
- 특징
  - 알파벳과 숫자는 각각 연속된 아스키 코드 값을 갖는다.
  - 소문자가 대문자보다 아스키 코드 값이 크다.
  - 제어 문자는 백슬래시와 함께 표시하며 출력할 때 그 기능을 수행한다.

11-1 아스키 코드 값과 문자 입출력 함수

❖ 아스키 코드 (2/2)

대문자를 소문자로 변경 소스 코드 예제11-1.c

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      char small, cap = 'G';
06
07      if ((cap >= 'A') && (cap <= 'Z'))
08      {
09          small = cap + ('a' - 'A');
10      }
11      printf("대문자 : %c %c", cap,'\n');
12      printf("소문자 : %c", small);
13
14      return 0;
15  }
```



실행결과

```
대문자 : G
소문자 : g
```

11-1 아스키 코드 값과 문자 입출력 함수

❖ scanf 함수를 사용한 문자 입력 (1/2)

공백이나 제어 문자의 입력 소스 코드 예제11-2.c

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      char ch1, ch2;
06
07      scanf("%c%c", &ch1, &ch2);
        // 2개의 문자를 연속 입력
08      printf("[%c%c]", ch1, ch2);
        // 입력된 문자 출력
09
10      return 0;
11  }
```

a와 b를 연속으로 입력하고 Enter를 치는 경우 ①

실행결과1
ab ↵
[ab]

a와 공백(␣)을 연속으로 입력하고 Enter를 치는 경우 ②

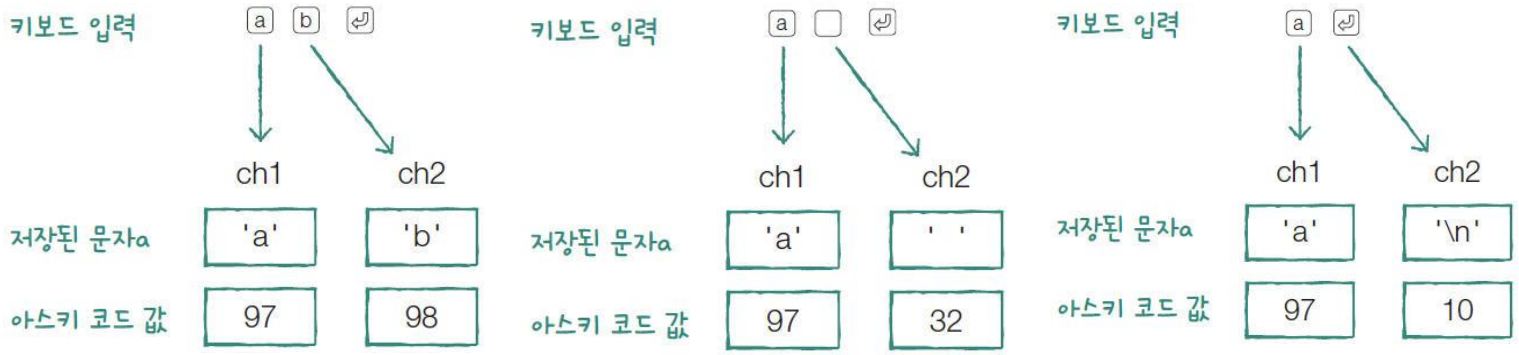
실행결과2
a␣ ↵
[a]

a만 입력하고 Enter를 치는 경우 ③

실행결과3
a ↵
[a
]

❖ scanf 함수를 사용한 문자 입력 (2/2)

- %c 변환문자는 공백, 탭, 개행문자를 입력한다.



- 공백, 탭, 개행문자를 제외하고 입력할 때는 %c 앞에 공백, 탭, 개행문자 중 하나 이상을 추가한다.

```
scanf(" %c %c", &ch1, &ch2);
```

%c 앞에 화이트 스페이스 사용
(탭 문자 \t나 개행 문자 \n도 사용 가능)

11-1 아스키 코드 값과 문자 입출력 함수

❖ getchar 함수와 putchar 함수

getchar 함수와 putchar 함수 사용 소스 코드 예제11-3.c

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      int ch;                 // 입력 문자를 저장할 변수
06
07      ch = getchar();         // 함수가 반환하는 문자를 바로 저장
08      printf("입력한 문자 : ");
09      putchar(ch);            // 입력한 문자 출력
10      putchar('\n');          // 개행 문자 출력
11
12      return 0;
13  }
```

실행결과

```
A
입력한 문자 : A
```

마무리

키워드로 끝내는 핵심 포인트

- 모든 **문자 상수**는 **아스키 코드** 값으로 바뀌어 숫자로 저장되고 연산된다.
- **%c 변환 문자**는 화이트 스페이스(공백 문자, 탭 문자, 개행 문자)도 입력하며, %c 앞에 공백을 사용하면 화이트 스페이스를 입력에서 제외할 수 있다.
- **getchar**, **putchar** 함수는 문자 전용 입출력 함수이다.

마무리

표로 정리하는 핵심 포인트

표 11-1 문자 입출력 함수

구분	사용 예	기능
입력	`char ch;` `scanf("%c", &ch);`	char형 변수 사용 %c 변환 문자로 입력 공백 문자, 탭 문자, 개행 문자도 입력
	`int ch;` `ch = getchar( );`	int형 변수 사용 입력 문자의 아스키 코드 값 반환 공백 문자, 탭 문자, 개행 문자도 입력
출력	`printf("%c", ch);` `putchar(ch);`	%c 변환 문자 사용 문자 출력 전용 함수, 출력할 문자 전달

11-2 버퍼를 사용하는 입력 함수

❖ scanf 함수가 문자를 입력하는 과정

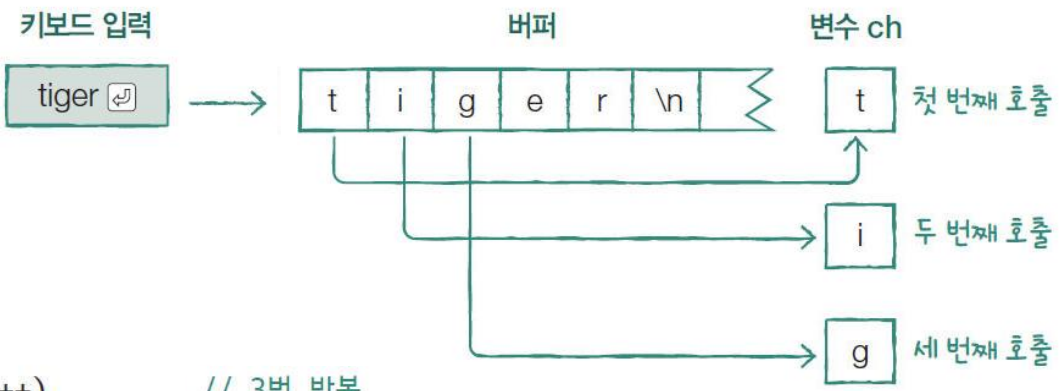
버퍼를 사용하는 문자 입력 소스 코드 예제11-4.c

```c
#include <stdio.h>

int main(void)
{
    char ch;
    int i;

    for (i = 0; i < 3; i++)          // 3번 반복
    {
        scanf("%c", &ch);            // 문자 입력
        printf("%c", ch);            // 입력된 문자 출력
    }

    return 0;
}
```



실행결과

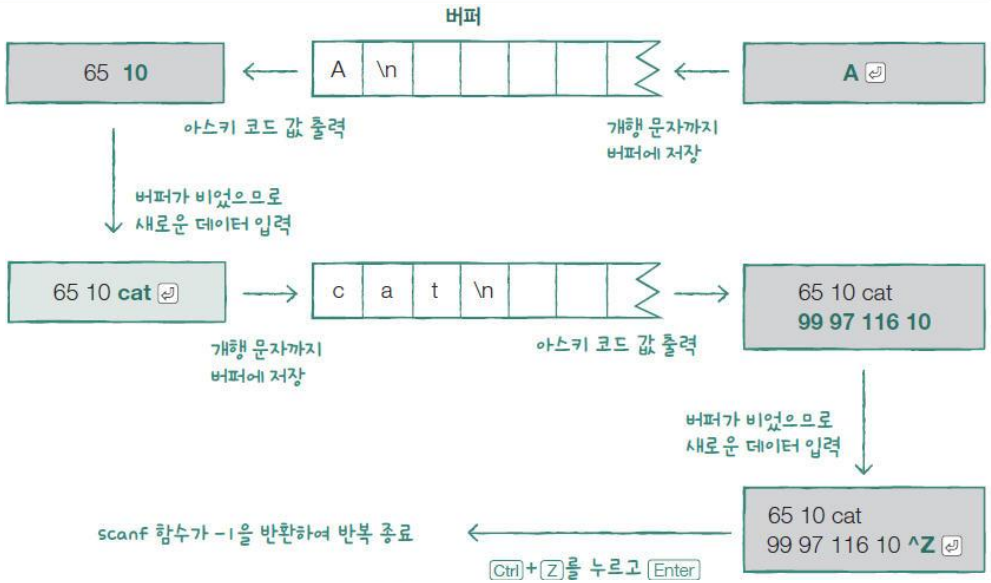
```
tiger
tig
```

11-2 버퍼를 사용하는 입력 함수

❖ scanf 함수의 반환값 활용

입력 문자의 아스키 코드 값을 출력하는 프로그램 소스 코드 예제11-5.c

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      int res;
06      char ch;
07
08      while (1)
09      {
10          res = scanf("%c", &ch);
11          if (res == -1) break;
12          printf("%d ", ch);
13      }
14
15      return 0;
16  }
```



실행결과

```
A
65 10 cat
99 97 116 10 Ctrl+Z
Ctrl+Z
```

11-2 버퍼를 사용하는 입력 함수

❖ getchar 함수를 사용한 문자열 입력 (1/2)

getchar 함수를 사용한 문자열 입력 소스 코드 예제11-6.c

```
01  #include <stdio.h>
02
03  void my_gets(char *str, int size);
04
05  int main(void)
06  {
07      char str[7];                          // 문자열을 저장할 배열
08
09      my_gets(str, sizeof(str));            // 한 줄의 문자열을 입력하는 함수
10      printf("입력한 문자열 : %s\n", str);      // 입력한 문자열 출력
11
12      return 0;
13  }
14
```

❖ getchar 함수를 사용한 문자열 입력 (2/2)

getchar 함수를 사용한 문자열 입력 소스 코드 예제11-6.c

```
15  void my_gets(char *str, int size)          // str은 char 배열, size는 배열의 크기
16  {
17      int ch;                                // getchar 함수의 반환값을 저장할 변수
18      int i = 0;                             // str 배열의 첨자
19
20      ch = getchar();                        // 첫 번째 문자 입력
21      while ((ch != '\n') && (i < size -1))  // 배열의 크기만큼 입력
22      {
23          str[i] = ch;     // 입력한 문자를 배열에 저장
24          i++;             // 첨자 증가
25          ch = getchar();  // 새로운 문자 입력
26      }
27      str[i] = '\0';       // 입력된 문자열의 끝에 널 문자를 저장
28  }
```

실행결과1

```
a boy
입력한 문자열 : a boy
```

실행결과2

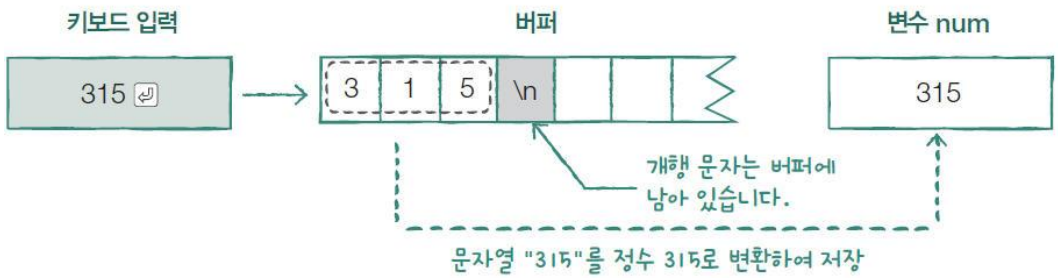
```
Be happy!
입력한 문자열 : Be hap
```

11-2 버퍼를 사용하는 입력 함수

❖ 입력 버퍼 지우기

버퍼의 내용을 지워야 하는 경우 소스 코드 예제11-7.c

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      int num, grade;
06
07      printf("학번 입력 : ");
08      scanf("%d", &num);          // 학번 입력
09      getchar();                  // 버퍼에 남아 있는 개행 문자 제거
10      printf("학점 입력 : ");
11      grade = getchar();          // 학점 입력
12      printf("학번 : %d, 학점 : %c", num, grade);
13
14      return 0;
15  }
```



실행결과

```
학번 입력 : 315
학점 입력 : A
학번 : 315, 학점 : A
```

실행결과 9행이 없을 때 출력 결과

```
학번 입력 : 315
학점 입력 : 학번 : 315, 학점 :
_
```

← 학점을 입력할 수 없고 학점을 출력할 곳에서 줄이 바뀜

← 커서가 다음 줄로 내려옴

마무리

키워드로 끝내는 핵심 포인트

- **scanf 함수**는 입력할 때 가장 먼저 버퍼의 상태를 확인한다.
- **버퍼**에 저장되는 데이터의 끝에는 항상 개행 문자가 있다.
- scanf 함수는 Ctrl + Z 를 누르면 **EOF**(-1)를 반환한다.

마무리

표로 정리하는 핵심 포인트

표 11-2 입출력 버퍼의 이름을 직접 사용하는 함수

구분	함수 사용법	기능
입력	`int ch;` `ch = fgetc(stdin);`	int형 변수에 입력 공백 문자, 탭 문자, 개행 문자도 입력 입력 문자의 아스키 코드 값 반환 입력 버퍼 stdin 사용
출력	`fputc(ch, stdout);`	문자 출력 전용 함수 출력할 문자와 출력 버퍼 stdout 사용