5.1.3 기본 입출력

함수 printf

함수 printf는 출력을 담당하는 함수이며 확장 문자를 포함하여 숫자, 문자, 문자열 외에 주소(포인터)를 형식에 맞게 출력합니다. 함수 printf의 원형과 인자는 다음과 같습니다.

printf	함수원형	int printf(const char *format [, argument, ...]	
	함수인자	format	형식 제어 문자열(형식 지정자, 확장 문자)
		argument	변수나 상수 또는 연산식의 리스트
	반환 값	출력한 byte수를 반환하며, 오류 발생시는 EOF를 반환. 반환 값은 거의 사용되지 않음	

출력될 내용의 줄을 바꿀 때 사용하는 \n은 기호 '\'(back slash)와 'n'이 합쳐진 형태인데 '\'와 함께 사용되어 특별한 기능을 하는 문자를 확장 문자(escape sequence)라 합니다. \n은 출력 형식의 앞이나 뒤, 중간 어느 부분에나 사용할 수 있습니다.

함수 printf에서 사용하는 확장 문자 :

확장 문자	사용 예	의미
\a	printf("\a");	beep 소리를 출력(audible bell)
\b	printf("\b");	커서를 한 문자 뒤로 이동(back space)
\f	printf("\f");	한 페이지 앞으로 전진(form feed)
\n	printf("Line1 \n"); printf("Line2");	커서를 다음 줄로 이동시키므로 Line2는 'Line1'의 다음 줄에 출력(new line)
\t	printf("\tNAME");	커서를 일정한 값(tap크기, 즉 7칸)만큼 오른쪽으로 이동시키므로 NAME은 왼쪽에서 8번째 칸부터 출력
\"	printf("\"NAME\"");	NAME을 출력하되 양 끝에 "(double quote)를 출력
\'	printf("\'NAME\'");	NAME을 출력하되 양 끝에 '(single quote)를 출력
\\	printf("\\NAME");	NAME을 출력하되 양 끝에 \(back slash)를 출력
\%	printf("\%"); printf("%%");	기호 %를 출력

입력 또는 출력에 사용될 데이터의 형식을 제어하는 문자들을 형식 지정자(format specifier)라 합니다. 형식 지정자는 '%'를 사용하는데 이후의 문자는 제어할 데이터의 형(type)을 나타냅니다. 데이터의 형에 따라 구분하면 다음과 같습니다. 정수형 형식 지정자에는 8진수나 16진수로 출력하는 %o와 %x가 포함되어 있습니다.

형식 지정자

공통 형식 지정자	인수형	의 미
%d	정수형	정수형 인수를 10진(Decimal) 정수로 입출력
%o	정수형	정수형 인수를 8진(Octal) 정수로 입출력
%x	정수형	정수형 인수를 16진(heXadecimal) 정수로 입출력
%u	정수형	정수형 인수를 부호 없는(Unsigned) 10진수로 입출력
%c	정수형	문자형(정수형) 인수를 단일 문자(Character)로 입출력
%s	문자열	문자열(String)로 입출력 (문자열 포인터)
%f	실수형	소숫점을 포함하는 실수형(Floating point)으로 입출력
%e	실수형	실수형 인수를 지수형(Exponent)으로 입출력
%g	실수형	%e와 %f 중에서 더 짧은 표현을 사용

printf에서 형식 지정자의 사용방법

```
01  #include <stdio.h>
02  int main(void)
03  {
04   printf("10진수 정수형 %%d: %d \n", 65);
05   printf(" 8진수 정수형 %%o: %o \n", 65);
06   printf("16진수 정수형 %%x: %x \n", 65);
07   printf("문자형 %%c: %c \n", 65);
08   printf("실수형 %%f: %f \n", 65.);
09   return 0;
10  }
```

[실행결과]

```
10진수 정수형 %d: 65
 8진수 정수형 %o: 101
16진수 정수형 %x: 41
문자형 %c: A
실수형 %f: 65.000000
```

정수형 데이터의 자릿수를 맞출 경우에는 다음과 같이 %와 d 사이의 ⓝ 위치에 정수 숫자를 사용하는데 ⓝ은 출력할 데이터의 최대 자릿수를 의미하고 ⓝ개의 자릿수에 대해 오른쪽 맞춤으로 출력합니다. 정수형 데이터의 자릿수를 지정하는 방법 외에 기호 '+', '-' 또는 '0'을 함께 사용하여 출력방법을 다르게 조절할 수 있습니다.

정수형 : %ⓝd

ⓝ은 출력할 전체 자릿수

정수형 숫자에 대해 자릿수를 맞추어 출력

```
01  #include <stdio.h>
02  int main(void )
03  {
04   int a=-5, b=23, c=5376;
05   long d=325678;
06   printf("정수형 숫자 format\n");
07   printf("\n1. a=%d", a);
08   printf("\n2. a=%6d", a);
09   printf("\n3. a=%06d", a);
10   printf("\n4. b=%6d", b);
11   printf("\n5. b=%+6d", b);
12   printf("\n6. c=%6d", c);
13   printf("\n7. c=%-6d", c);
14   printf("\n8. d=%6ld\n", d);
15   return 0;
16  }
```

[실행결과]

```
정수형 숫자 format

1. a=-5
2. a=    -5
3. a=-00005
4. b=    23
5. b=   +23
6. c=  5376
7. c=5376
8. d=325678
```

실행결과 설명

프로그램 line	프로그램	출력 설명
08	printf("\n2. a=%6d", a);	□ □ □ □ - 5 오른쪽 맞춤
09	printf("\n3. a=%06d", a);	- 0 0 0 0 5 오른쪽 맞춤
11	printf("\n5. b=%+6d", b);	□ □ □ + 2 3 부호표시
13	printf("\n7. c=%-6d", c);	5 3 7 6 □ □ 왼쪽 맞춤
14	printf("\n8. b=%6ld", d);	3 2 5 6 7 8 오른쪽 맞춤

실수형 데이터의 자릿수를 맞출 경우에는 다음과 같이 %와 f 사이에 두 개의 숫자가 필요합니다. 먼저 ⓝ은 소숫점을 포함한 전체 자릿수를 말하며, ⓓ는 소숫점 이하의 자릿수를 지정합니다. ⓝ과 ⓓ가 모두 사용되지 않을 경우에는 왼쪽 맞춤으로 출력합니다.

실수형 : %ⓝ.ⓓf
ⓝ은 소숫점을 포함하여 출력할 전체 자릿수
ⓓ는 소숫점 이하 자릿수

실수형 숫자에 대해 자릿수를 맞추어 출력

```
01  #include <stdio.h>
02  int main(void )
03  {
04    float a=-437.46, b=1278.9;
05    double c=5.567;
06    printf("실수형 숫자 format\n");
07    printf("\n1. a=%8.3f", a);
08    printf("\n2. b=%8.3f", b);
09    printf("\n3. c=%8.3f", c);
10    printf("\n4. c=%8.f", c);
11    printf("\n5. c=%8.1f", c);
12    printf("\n6. c=%.2f", c);
13    printf("\n7. a=%+.2f", a);
14    printf("\n8. b=%+.2f", b);
15   return 0;
16  }
```

[실행결과]

```
실수형 숫자 format

1. a=-437.460
2. b=1278.900
3. c=   5.567
4. c=       6
5. c=     5.6
6. c=5.57
7. a=-437.46
8. b=+1278.90
```

실행결과 설명

프로그램 line	프로그램	출력 설명	
08	printf("\n2. b=%8.3f", b);	1 2 7 8 . 9 0 0	오른쪽 맞춤
10	printf("\n4. c=%8.f", c);	6	오른쪽 맞춤
12	printf("\n6. c=%.2f", c);	5 . 5 7	왼쪽 맞춤
14	printf("\n8. b=%+.2f", b);	+ 1 2 7 8 . 9 0	부호표시

문자열을 출력할 때는 형식 지정자 %s를 다음과 같이 사용합니다. 문자열을 변수에 저장하여 처리할 때는 문자형 배열이나 포인터를 이용할 수 있습니다.

%ns : 문자열이 출력될 n개의 자리를 확보하고 오른쪽 맞춤
%-ns : 문자열이 출력될 n개의 자리를 확보하고 왼쪽 맞춤
%.ns : 문자열에 대해 n자리 수만큼 왼쪽 맞춤

문자열에 대해 자릿수를 맞추어 출력

```
01  #include <stdio.h>
02  int main(void)
03  {
04   char *name1="string";
05   char *name2="chapter";
06   printf("문자열 format\n");
07   printf("\n1. name1=%10s", name1);
08   printf("\n2. name2=%10s", name2);
09   printf("\n4. name2=%-10s", name2);
10   printf("\n5. name1=%.3s", name1);
11   printf("\n6. name2=%.4s\n", name2);
12   return 0;
13  }
```

[실행결과]

```
문자열 format

1. name1=    string
2. name2=   chapter
4. name2=chapter
5. name1=str
6. name2=chap
```

프로그램 설명

프로그램 line	프로그램	출력 설명
07	printf("%10s", name1);	[ ][ ][ ][ ][s][t][r][i][n][g] 오른쪽 맞춤
09	printf("%-10s", name2);	[c][h][a][p][t][e][r][ ][ ][ ] 왼쪽 맞춤
10	printf("%.3s", name1);	[s][t][r] 세 자리까지만 왼쪽 맞춤 출력

진법(base)을 출력하는 방법과 주소에 대한 형식은 다음과 같습니다.

구분	형식	출력 방법
8진법	%no	n개의 자릿수를 확보하여 오른쪽 맞춤(octal)
	%-no	n개의 자릿수를 확보하여 왼쪽 맞춤
16진법	%nx	n개의 자릿수를 확보하여 오른쪽 맞춤(hexadecimal)
	%-nX	n개의 자릿수를 확보하여 왼쪽 맞춤, 대문자로 출력
주소	%u	주소(address)를 부호 없는 10진수로 변환하여 출력
	%p	주소를 16진수로 출력(pointer)

8진수, 16진수와 주소를 출력

```
01  #include <stdio.h>
02  int main(void)
03  {
04   int a=32, b=512;
05   long c=4874;
06   printf("진법 및 주소 format\n");
07   printf("\n1. base 16 of a = %5X", a);
08   printf("\n2. base 16 of a = %-5x", a);
09   printf("\n3. base  8 of b = %5o", b);
10   printf("\n4. base  8 of b = %-5o", b);
11   printf("\n7. address of a = %p", &c);
12   printf("\n8. address of a = %u\n", &c);
13   return 0;
14  }
```

[실행결과 : Visual C++]

```
진법 및 주소 format

1. base 16 of a =    20
2. base 16 of a = 20
3. base  8 of b =  1000
4. base  8 of b = 1000
7. address of c = 0012FF74
8. address of c = 1245044
```

[실행결과 : Turbo C++]

```
진법 및 주소 format

1. base 16 of a =    20
2. base 16 of a = 20
3. base  8 of b =  1000
4. base  8 of b = 1000
7. address of c = 1957:227A
8. address of c = 8826
```

주소의 값은 컴파일러와 컴퓨터에 따라 다르게 출력됩니다.

프로그램 설명

프로그램 line	프로그램	출력 설명
07	printf("%5X", a);	␣ ␣ ␣ 2 0 오른쪽 맞춤, 16진수로 출력
10	printf("%-5o", b);	1 0 0 0 ␣ 왼쪽 맞춤, 8진수로 출력
11	printf("%p", &c);	변수 c의 주소를 16진수로 표시, 16진수 227A는 10진수로 8826
12	printf("%u", &c);	변수 a의 주소를 부호 없는 10진수로 표시