



C프로그래밍

Lecture 06. printf 함수와 scanf 함수 정리하기

동덕여자대학교
데이터사이언스 전공
권 범

목차

❖ 01. printf 함수 이야기

❖ 02. scanf 함수 이야기

01. printf 함수 이야기

02. scanf 함수 이야기

01. printf 함수 이야기

❖ printf 함수는 문자열을 출력하는 함수이다

```
1  /* printf1.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello Everybody!\n");
7      printf("I Love You!\n");
8      printf("Today is Friday.\n");
9
10     return 0;
11 }
```

Hello Everybody!
I Love You!
Today is Friday.

C언어는 큰 따옴표(" ")를 사용해서
문자열을 표현합니다.

printf 함수 호출 시 문자열을 전달하면,
전달된 문자열이 모니터상에 출력됩니다.

01. printf 함수 이야기

❖ printf 함수는 특수 문자 출력이 가능하다: ① 특수 문자가 필요한 이유 (1/2)

- C언어에는 특수 문자(Escape Sequence)라고 불리는 문자가 있습니다.
- 이러한 문자들을 정의하고 있는 이유는 출력 시 특수한 문자를 출력하기 위함입니다.

특수 문자라는 것이 왜 필요한지
아래 예를 살펴보겠습니다.

```
1  /* printf2.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("앞집 강아지가 말했다. “멍! 멍!” 정말 귀엽다.\n");
7
8      return 0;
9  }
```

우리가 원하는 출력 형태

앞집 강아지가 말했다. “멍! 멍!” 정말 귀엽다.

하지만 **컴파일 오류 발생**

01. printf 함수 이야기

❖ printf 함수는 특수 문자 출력이 가능하다: ① 특수 문자가 필요한 이유 (2/2)

```
printf("앞집 강아지가 말했다. “멍! 멍!” 정말 귀엽다.\n");
```

“앞집 강아지가 말했다. “
멍! 멍!
” 정말 귀엽다.\n”

→ 음 이것은 하나의 문자열이군!
→ 이건 뭐지?
→ 이것도 하나의 문자열이군!

```
printf("앞집 강아지가 말했다. \“멍! 멍!\” 정말 귀엽다.\n");
```

잘못된 printf 함수 호출문

컴파일러의 오해

- ✓ 큰 따옴표는 문자열의 시작과 끝으로 해석이 되니, 큰 따옴표 자체의 출력을 원하는 경우에는 큰 따옴표 앞에 \ 문자를 붙여주기로 하자!

특수 문자의 탄생 배경

제대로 된 printf 함수 호출문

01. printf 함수 이야기

❖ printf 함수는 특수 문자 출력이 가능하다: ② 특수 문자의 종류

특수문자	의미하는 바
\a	경고음
\b	백스페이스(backspace)
\f	폼 피드(form feed)
\n	개 행(new line)
\r	캐리지 리턴(carriage return)
\t	수평 탭
\v	수직 탭
\'	작은 따옴표 출력
\"	큰 따옴표 출력
\?	물음표 출력
\\	역슬래시 출력

외울 필요 없습니다.
대략 종류만 알고 있다가
필요할 때 참고하면 됩니다.

- ✓ wf와 wv는 모니터 출력이 아닌 프린터 출력을 위해 정의된 특수문자이기 때문에 모니터 출력에 사용하면, 이상한 문자가 출력됩니다!

01. printf 함수 이야기

❖ printf 함수의 서식 지정과 서식 문자들 (1/2)

**printf라는 이름의 맨 끝 f는
formatted(서식화된)을 의미합니다.**

- ✓ 따라서 printf 함수를 이용할 경우, 다양한 형태의 숫자를 여러 번 출력하거나, 숫자와 문자를 섞어서 출력하는 등의 출력 양식을 정의할 수 있습니다.

01. printf 함수 이야기

❖ printf 함수의 서식 지정과 서식 문자들 (2/2)

```
1  /* printf3.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int myAge = 12;
7
8      printf("제 나이는 10진수로 %d살, 16진수로 %x살입니다.\n", myAge, myAge);
9
10     return 0;
11 }
```

- %d ← decimal(10진법의)
- %x ← hexadecimal(16진법의)

제 나이는 10진수로 12살, 16진수로 c살입니다.

서식 문자를 이용해서 출력할 문자열의 형태를 조합해 낼 수 있습니다.
즉, 출력의 서식을 지정할 수 있습니다.

01. printf 함수 이야기

❖ 서식 문자의 종류와 그 의미 (1/2)

서식문자	출력 대상(자료형)	출력 형태
%d	char, short, int	부호 있는 10진수 정수
%ld	long	부호 있는 10진수 정수
%lld	long long	부호 있는 10진수 정수
%u	unsigned int	부호 없는 10진수 정수
%o	unsigned int	부호 없는 8진수 정수
%x, %X	unsigned int	부호 없는 16진수 정수
%f	float, double	10진수 방식의 부동소수점 실수
%Lf	long double	10진수 방식의 부동소수점 실수
%e, %E	float, double	e 또는 E 방식의 부동소수점 실수
%g, %G	float, double	값에 따라 %f와 %e 사이에서 선택
%c	char, short, int	값에 대응하는 문자
%s	char *	문자열
%p	void *	포인터의 주소 값

외울 필요 없습니다.
필요할 때 참고하면 됩니다.

01. printf 함수 이야기

❖ 서식 문자의 종류와 그 의미 (2/2)

```
1  /* printf4.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int num1 = 7, num2 = 13;
7
8      printf("%o, %#o\n", num1, num2);
9      printf("%x, %#x\n", num1, num2);
10     return 0;
11 }
```

7 07
d 0xd

- %d ← decimal(10진법의)
- %o ← octal(8진법의)
- %x ← hexadecimal(16진법의)

#을 삽입하면 8진수 앞에 0을,
16진수 앞에 0x가 삽입됩니다.

01. printf 함수 이야기

❖ 실수의 출력을 위한 서식 문자들: %f, %e

```
1  /* printf5.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("%f\n", 0.1234);
7      printf("%e\n", 0.1234);    // e 표기법 기반의 출력
8      printf("%f\n", 0.12345678);
9      printf("%e\n", 0.12345678); // e 표기법 기반의 출력
10     return 0;
11 }
```

0.123400
1.234000e-001
0.123457
1.234568e-001

컴퓨터는 지수를 표현할 수 없으므로,
e 표기법으로 지수를 대신 표현합니다.

0.0000000001

120000

1.0×10^{-10}

지수 표기법

$1.2 \times 10^{+5}$

1.0e - 10


e 표기법

1.2e + 5

01. printf 함수 이야기

❖ %g의 실수 출력

```
1  /* printf6.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      double d1 = 1.23e-3;
7      double d2 = 1.23e-4;
8      double d3 = 1.23e-5;
9      double d4 = 1.23e-6;
10     printf("%g\n", d1);      // %f 스타일 출력
11     printf("%g\n", d2);      // %f 스타일 출력
12     printf("%g\n", d3);      // %e 스타일 출력
13     printf("%g\n", d4);      // %e 스타일 출력
14     return 0;
15 }
```



```
0.00123
0.000123
1.23e-005
1.23e-006
```

- ✓ %g는 실수의 형태에 따라서 %f와 %e 사이에서 적절한 형태의 출력을 진행합니다.
- ✓ %g와 %G의 차이점은 e 표기법의 e를 소문자로 출력하느냐 대문자로 출력하느냐에 있습니다.

01. printf 함수 이야기

❖ %s의 문자열 출력

```
1  /* printf7.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("%s, %s, %s\n", "AAA", "BBB", "CCC");
7      return 0;
8  }
```

AAA, BBB, CCC

- ✓ %s의 문자열 출력과 관련해서는 배열과 포인터 공부 후에 완벽히 이해하도록 합니다!
- ✓ 일단은 %s의 사용법을 예제 기반으로 이해하도록 합니다.

01. printf 함수 이야기

❖ 필드 폭을 지정하여 정돈된 출력 보이기 (1/2)

```
1  /* printf8.c */
2  #include <stdio.h>
3  int main(void)
4  {
5      printf("/%12d/ /%12d/ /%12d/\n", 100, 200, 300);
6      printf("/%+12d/ /%+12d/ /%+12d/\n", 100, 200, 300);
7      printf("/%-12d/ /%-12d/ /%-12d/\n", 100, 200, 300);
8
9      return 0;
10 }
```

- ✓ 서식 문자 사이에 들어가는 숫자는 필드의 폭을 의미합니다.
- ✓ 기본 오른쪽 정렬입니다.
- ✓ -는 왼쪽 정렬을 의미하는 용도로 사용됩니다.

/	100/	/	200/	/	300/
/	+100/	/	+200/	/	+300/
/100	/	/200	/	/300	/

01. printf 함수 이야기

❖ 필드 폭을 지정하여 정돈된 출력 보이기 (2/2)

```
1  /* printf9.c */
2  #include <stdio.h>
3  int main(void)
4  {
5      printf("%-8s %14s %5s\n", "이 름", "전공학과", "학년");
6      printf("%-8s %14s %5d\n", "김동수", "전자공학", 1);
7      printf("%-8s %14s %5d\n", "이을수", "컴퓨터공학", 2);
8      printf("%-8s %14s %5d\n", "한선영", "미술교육학", 3);
9      return 0;
10 }
```

- ✓ %8d: 필드 폭을 8칸 확보하고, 오른쪽 정렬해서 출력을 진행합니다.
- ✓ %-8d: 필드 폭을 8칸 확보하고, 왼쪽 쪽 정렬해서 출력을 진행합니다.

이 름	전공학과	학년
김동수	전자공학	1
이을수	컴퓨터공학	2
한선영	미술교육학	3

02. scanf 함수 이야기

01. printf 함수 이야기

02. scanf 함수 이야기

❖ 정수 기반의 입력 형태 정의하기 (1/2)

데이터를 입력받는 scanf 함수에게
전달해야 할 두 가지 정보

- ① (입력의 형식) 어떻게 받아들일 거니?
- ② (입력의 장소) 어디에 저장할까?

- ✓ %d: 10진수 정수의 형태로 데이터를 입력받는다.
- ✓ %o: 8진수 양의 정수의 형태로 데이터를 입력받는다.
- ✓ %x: 16진수 양의 정수의 형태로 데이터를 입력받는다.

- %d ← decimal(10진법의)
- %o ← octal(8진법의)
- %x ← hexadecimal(16진법의)

서식 문자의 의미는 출력을 입력으로만 변경하면
printf 함수와 유사합니다.

02. scanf 함수 이야기

❖ 정수 기반의 입력 형태 정의하기 (2/2)

```
1  /* scanf1.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      int num1, num2, num3;
8
9      printf("세 개의 정수 입력: ");
10     scanf("%d %o %x", &num1, &num2, &num3);
11     printf("입력된 정수 10진수 출력: ");
12     printf("%d %d %d\n", num1, num2, num3);
13     return 0;
14 }
```

- %d ← decimal(10진법의)
- %o ← octal(8진법의)
- %x ← hexadecimal(16진법의)

세 개의 정수 입력: 12 12 12

입력된 정수 10진수 출력: 12 10 18

02. scanf 함수 이야기

❖ 실수 기반의 입력 형태 정의하기 (1/2)

float형 데이터의 삽입을 위한 서식 문자

- ✓ printf 함수에서는 서식 문자 %f, %e, %g의 의미가 각각 달랐습니다.
- ✓ 하지만 scanf 함수에서는 'float형 데이터를 입력받겠다'는 동일한 의미를 갖고 있습니다.

double형, long double형 데이터의 삽입을 위한 서식 문자

double	%lf	%f에 l이 추가된 형태
long double	%Lf	%f에 L이 추가된 형태

- ✓ float, double, long double의 데이터 출력: %f, %f, %Lf
- ✓ float, double, long double의 데이터 입력: %f, %lf, %Lf

02. scanf 함수 이야기

❖ 실수 기반의 입력 형태 정의하기 (2/2)

```
1  /* scanf2.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4  int main(void)
5  {
6      float num1;
7      double num2;
8      long double num3;
9
10     printf("실수 입력1(e 표기법으로): ");
11     scanf("%f", &num1);
12     printf("입력된 실수 %f\n", num1);
13
```

실수 입력과정에서 e 표기법을 사용해도 됩니다.

```
14     printf("실수 입력2(e 표기법으로): ");
15     scanf("%lf", &num2);
16     printf("입력된 실수 %f\n", num2);
17
18     printf("실수 입력3(e 표기법으로): ");
19     scanf("%Lf", &num3);
20     printf("입력된 실수 %Lf\n", num3);
21     return 0;
22 }
```

실수 입력1(e 표기법으로): 1.1e-3
입력된 실수 0.001100
실수 입력2(e 표기법으로): 0.1e+2
입력된 실수 10.000000
실수 입력3(e 표기법으로): 0.17e-4
입력된 실수 0.000017

❖ 01. printf 함수 이야기

❖ 02. scanf 함수 이야기

THANK YOU!

Q & A

- Name: 권범
- Office: 동덕여자대학교 인문관 B821호
- Phone: 02-940-4752
- E-mail: bkwon@dongduk.ac.kr