Jin

Printf

Contents

OVERVIEW

Printf 함수에 대한 간단한 정리 및 디자인 노트

SUMMARY

- Basic Format : %[flag][width][.precision][conversion type character]
- Printf 함수는 두개의 매개변수를 취하며, 첫번째 스트링은 formatting string, 두번째는 데이터
- Formatting string은 " "에 씌워져 있어야 하며, 그러지 않을 시 작동 되지 않는다.
- Formatting string안에는 지시자가 들어가 있으며, 변환 지시자가 %사인 뒤에 따라온다.

DETAIL

Conversion Type Character : %[flag][width][.precision][conversion Type]

- Format Specifier라고도 하며, 변수를 사용 하여 같은 내용을 여러개 출력하거나, 출력 형태를
  바꿀 때 사용할 수 있다.

서식	출력타입	설명	예시
%d	int	int형 출력 지시자로 0 - 9의 정수 출력	printf("%d" , 10); = 10
%x	Unsigned int	10진수를 받아 16진수로 변환	printf("%x", 10); = 'a'
%u	Unsigned int	Unsigned int 0 - 9의 정수 출력	printf("%u", 10); = 10
%f	Float	소수점을 다루는 데이터를 10진수로 출력	printf("%f", 10.1); = 10.10000
%c	Char	Char 데이터를 출력한다	printf("%c", 'a'); = 'a'
%%	-	% 사인 출력	printf("%%"); = %
%s	Char *	문자열을 출력한다	printf("%s", "hello"); = hello
%o	Unsigned int	10진수를 받아 8진수로 변환	printf("%o", 10); = 12
%p	Void *	포인터의 주소값을 출력	printf("%p", &ptr); = 00x00xxx

Error Handling

- 데이터 출력타입과의 매치 확인
- 기본 10지수, Boolean, Ascii 캐릭터에는 Precisions를 적용할 수 없음
- [-] flag는 width와 결합 되어야 한다.
- [+]와 [space] flag는 결합 될 수 없다.
- [0] flag는 width와 결합되어야 한다.
- 포맷 순서는 %[Flag][.precision][modifier][conversion type Char]가 되어야 한다.
- [*] flag는 [space]플래그와 겹친다.
- Width는 0으로 정의 될 수 없다.
- [#]을 %d와 결합 하면 안된다.
- [0] flag는 %s와 같이 쓰일 수 없다.
- [0] flag는 '-', %d, %i, %u, %X, %x, %u 를 무시한다.
- [+] flag는 '%o', '%u', '%x', '%X'와 같이 쓰일 수 없다
- [#] flag는 '%u', '%i', '%s', %d'와 같이 쓰일 수 없다.

서식	+	-	#	0
%d			ign	ign
%x	ign			ign
%f				
%c				
%o	ign			
%s			ign	
%u		ign	ign	
%p				
%%				

Precision : %[flag][width][.precision][conversion Type]

- Precision은 정수 출력의 갯수를 정해준다.
- Precision이 없을 때에는, 6개의 정수를 출력한다.
- 만약 마지막 숫자가 5보다 크거나 같으면 반올림 한다.

Function call	출력
printf("%.0f", 12.3456);	12
printf("%.1f", 12.3456);	12.3
printf("%.2f", 12.3456);	12.34
printf("%.3f", 12.3456);	12.345
printf("%.4f", 12.3456);	12.3456
printf("%.5f", 12.3456);	12.34560
printf("%.2s", "hello");	he
printf("%.3b", "true");	tru
printf("%.2c", 'A');	error
printf("%.3d", 123);	error

Width : %[flag][width][.precision][conversion Type]

- Width는 %사인과 conversion Type사이에 위치한다. (precision이 없을 시)
- width는 몇개의 스페이스가 발생하는지를 컨트롤 한다.
  - Ex: printf("%8d %n", 123); = _ _ _ _ _ _ 1 2 3
  - Ex: printf("%4d %5d", 123, 456); = _ 1 2 3 _ _ 4 5 6
  - Ex: printf("%4d", 12345); = 만약 width가 캐릭터 수보다 작으면 ignore됨
  - Ex: printf("%6.2f", 123.45); = 1 2 3 . 4 5 6
- 만약 width가 음수로 설정되었다면, 오른쪽에 스페이스가 출력된다
  - Ex: printf("%-6s%5d%", "num =), 456) = n u m _ = _ _ _ 4 5 6

Flags : %[flag][width][.precision][conversion Type]

- 숫자 데이터 타입에서 사용 {%d, %o, %x, %X, %e, %E, %f, %g, %G, %a, %A}
- 1[-]. 필드에서 값을 왼쪽으로 정렬한다.
- 2[+]. 부호가 있는 값이 사용될 때, 음수가 아닌 값에 대해서는 값 앞에 + 기호를 붙여 나타낸다.

Function call	출력
printf(“%+d”, 123);	+123
printf(“%+d”, -123);	-123
printf(“%+.2f”, 123.12);	+123.12
printf(“%+.2f”, -123.12);	-123.12

- 3[0]. 숫자 값이 오른쪽 정렬일때, 남게 되는 빈 공간을 0으로 채운다..
- 4[space]. 만약 숫자가 양수일 때, 숫자 앞에 스페이스를 출력한다. 모든 숫자 타입 플래그에 사용된다

  - Ex : printf(“% %d”, 123) = _ 123, (“% %d”, - 123); = -123
- 5[#]. 진법에 맞게 숫자 앞에 0, 0x, 0X 를 붙인다.
- [h] Short int 형을 다룬다.
- [hh] signed char와 unsigned char 형을 다룬다.
- [l] long int와 unsigned long int 형을 다룬다.
- [ll] long int와 unsigned int 형을 다룬다.
- [z] size_t와 ssize_t 형을 다룬다.
- [*] width와 precision을 argument로 다룬다.

  - Ex : printf(“%.*s”, 4, “42”); = _ _ 4 2