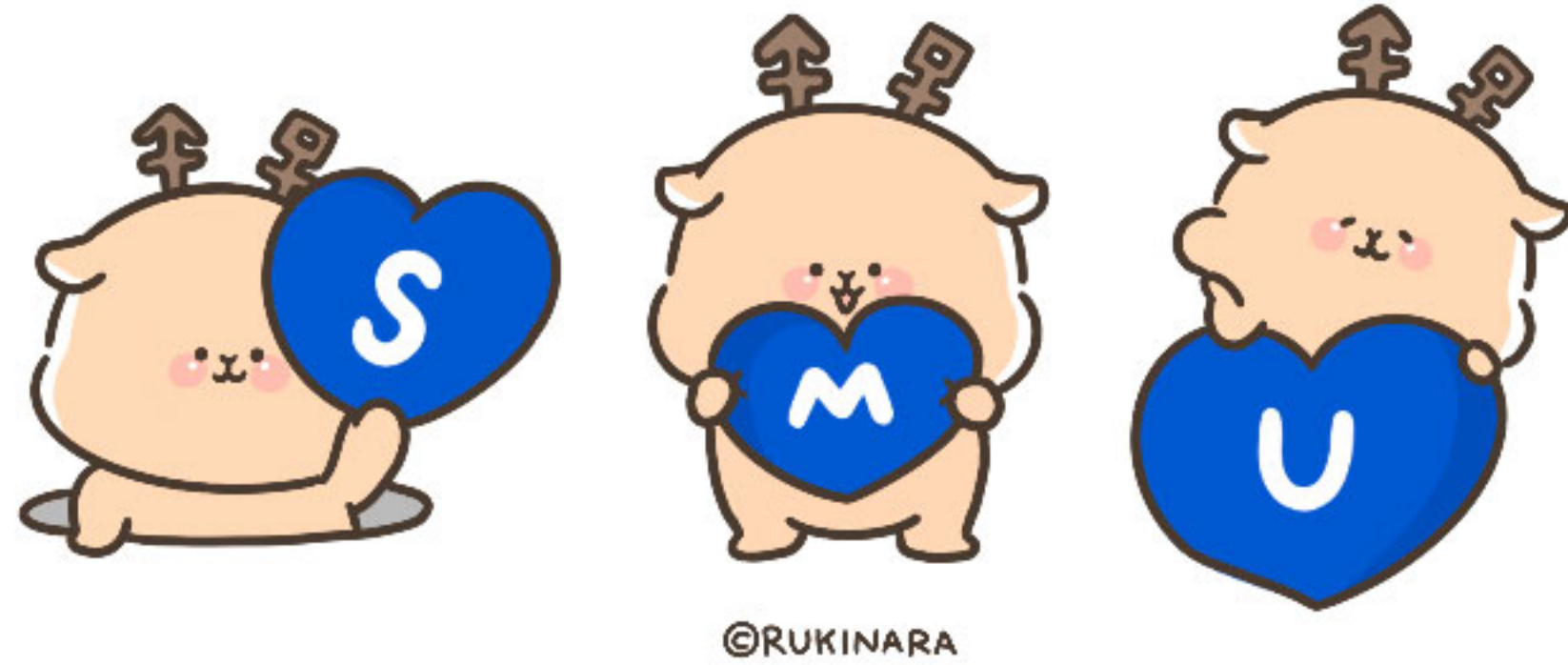


# C언어 멘토링 4주차

## PRINTF & SCANF



멘토 : 이승준

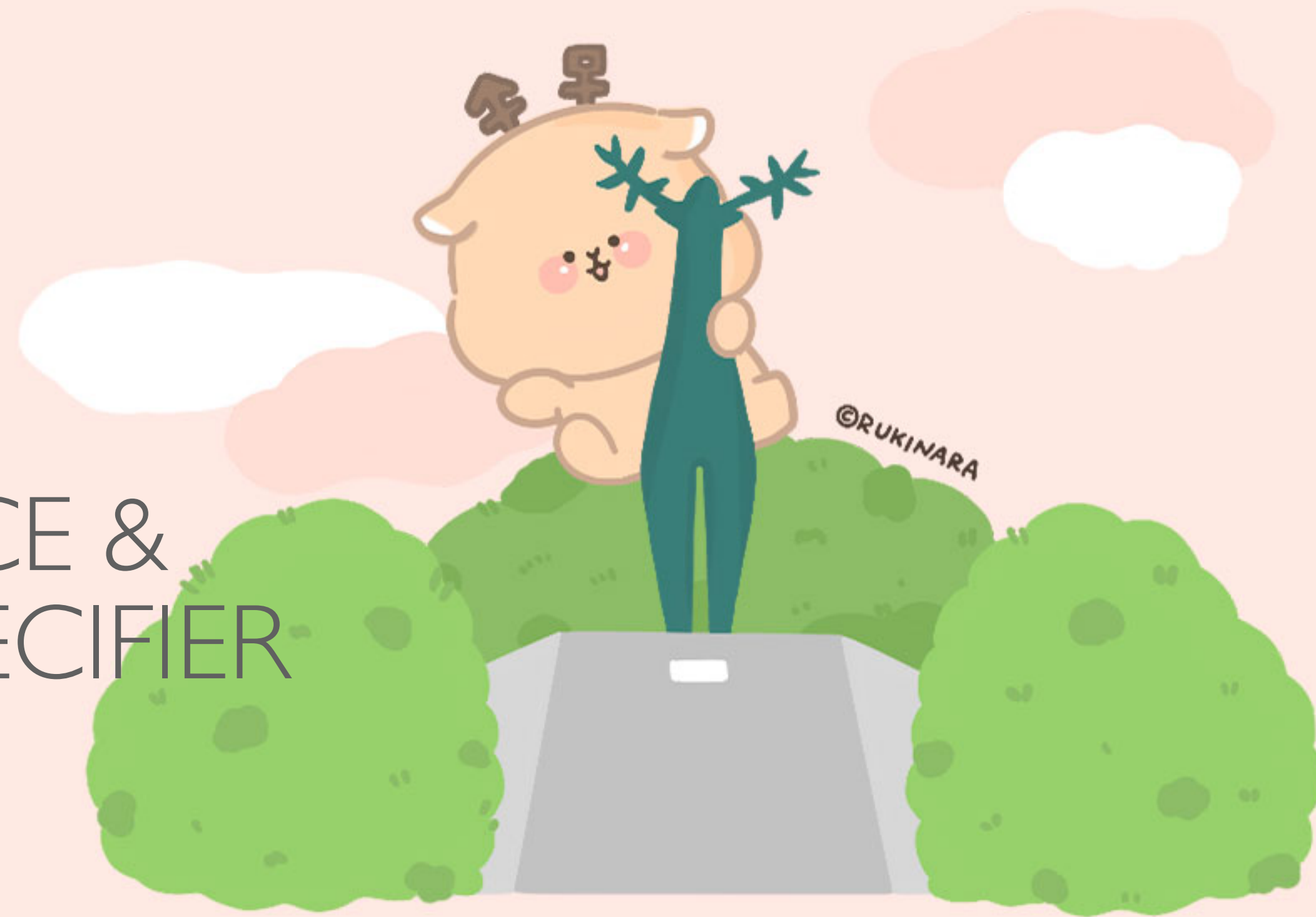
멘티 : 김도영

멘티 : 홍우기

SUMOONG ( designed by @RUKINARA)

# PRINTF

## ESCAPE SEQUENCE & CONVERSION SPECIFIER



HELLO, SUMOONG

# ESCAPE SEQUENCE

기호	내용
\a	경고음
\b	백스페이스
\f	폼 피드
\n	개행(new line)
\r	캐리지 리턴(carriage return)
\t	수평 탭
\v	수직 탭
\' \' \' \' \? \\\	문자열 내에서 \를 제외하고 출력(시험문제 가능)



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



서식문자	입력 자료형	입력형태
%d, %i	int	부호 있는 10진수 정수
%ld	long	부호 있는 10진수 정수
%lld	long long	부호 있는 10진수 정수
%u	unsigned int (10)	부호 없는 10진수 정수
%o	unsigned int (8)	부호 없는 8진수 정수
%x	unsigned int (16)	부호 없는 16진수 정수
%f , %e, %g	float	부호 있는 실수
%lf	double	부호 있는 실수
%Lf	long double	부호 있는 실수
%c	char	값에 대응하는 문자 (ASCII CODE 표)
%s	char*	문자열
%p	void*	포인터의 주소값

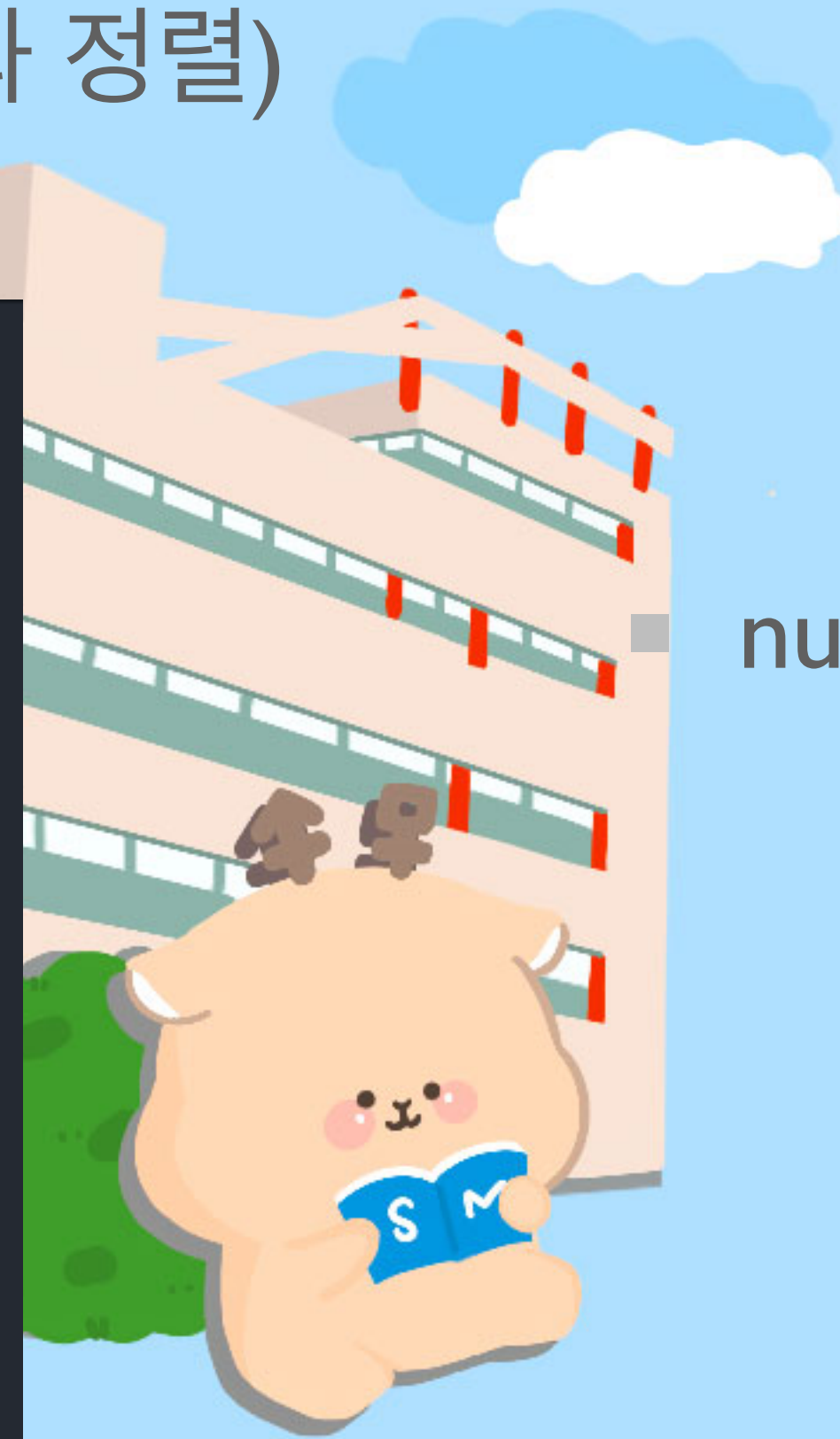
# 필드폭 지정

■ 서식문자에 들어가는 필드폭 지정(+ : 우 정렬, - : 좌 정렬)

```
1  #include <stdio.h>
2
3  int main (void){
4
5      int num =76;
6      int num2 =576;
7
8      printf("num is %-3d %3d\n",num,num);
9      printf("num is %-3d %3d\n",num2,num2);
10     //%-3d 뒤에 3칸 확보후 앞에서부터 넣기 [7,6, ]
11     // %3d 앞에 3칸 확보후 뒤에서부터 넣기 [ ,7,6]
12     return 0;
13 }
```

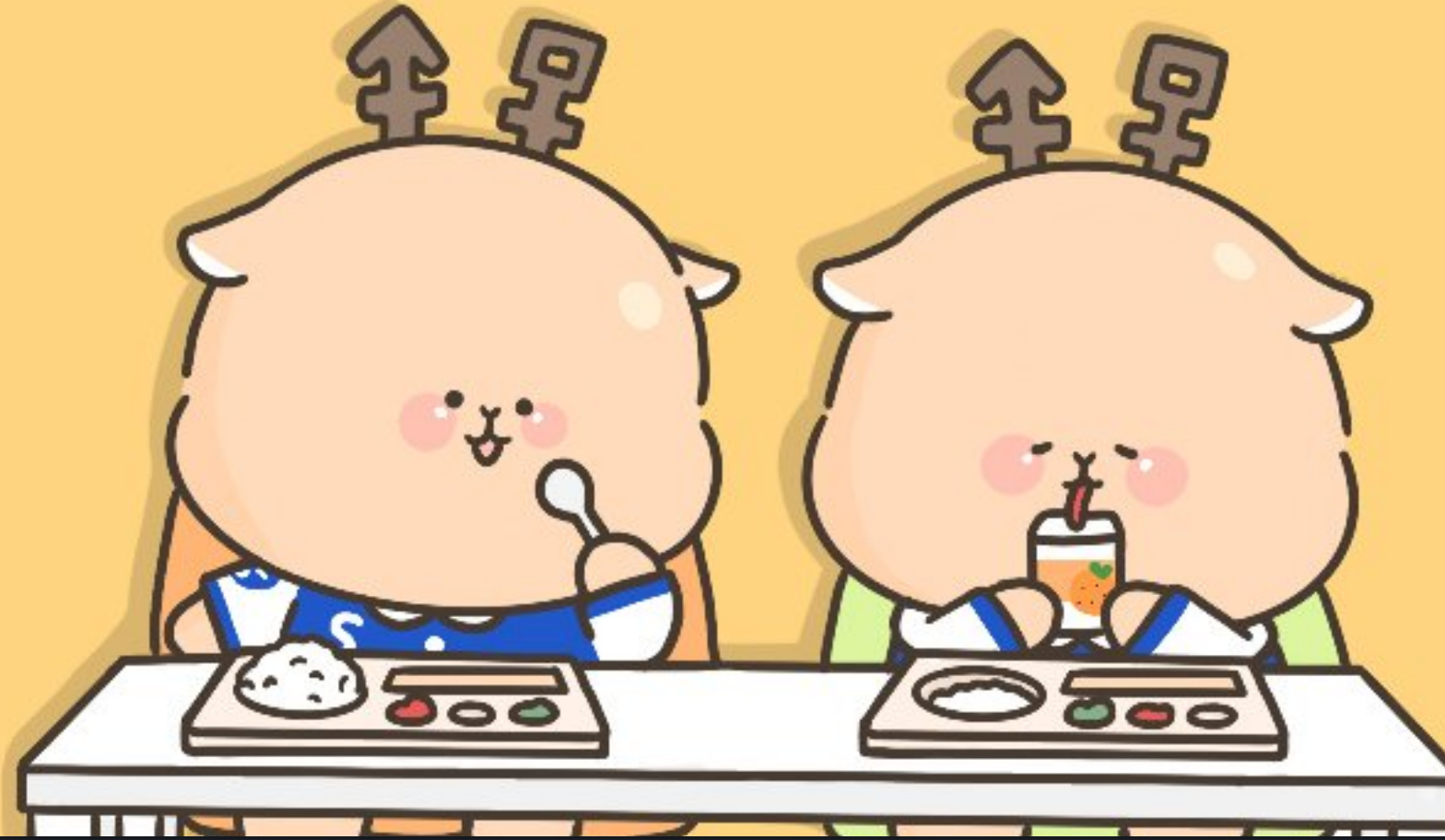
num is 76 76  
num is 576 576

num is [7,6, ∅]∅[∅,7,6]\n





## 10,8,16진수를 출력하는 방법



```
1  #include <stdio.h>
2
3  int main (void){
4
5      //10, 8, 16 진수의 형태로 저장하는 법
6      int num =76;
7      int num2 = 076;
8      int num3 = 0x76;
9
10     //10, 8, 16 진수의 형태로 출력하는 법
11     //8,16진수에서 출력형태 조절
12     printf("num은  10진수로는 %3d 이고 8진수로는 %4o 16진수로는 %3x 이다.\n",num,num,num);
13     printf("num2는 10진수로는 %3d 이고 8진수로는 %4#o 16진수로는 %3#x 이다.\n",num2,num2,num2);
14     printf("num3는 10진수로는 %3d 이고 8진수로는 %4#o 16진수로는 %3#x 이다.\n",num3,num3,num3);
15
16     return 0;
17 }
```

변수	10진수	8진수	16진수
num	76	114	4c
num2	62	076	0x3e
num3	118	0166	0x76

# 실수 형태의 출력

```
2
3  int main (void){
4
5     double num = 0.1234000;
6     printf("%f\n",num);      0.123400
7     printf("%e\n",num);      1.234000e-01
8     printf("%f\n",num);      0.123400
9     printf("%e\n",num);      1.234000e-01
10    printf("%0.3f\n",num);    0.123
11    //e-01 = 10^-1 = 10의 -1승
12    //e 표기법이라고 한다.
13    return 0;
14 }
```



- **%f와 %e** 의 차이는 출력 방식의 차이이다. **%f**는 소수점 방식  
**%e**는 e표기법 방식이다.
- **%0.3f** 로 소수점 자리의 출력 조절 가능



# 실수 형태의 출력

- **%g**  
7자리수 이하는 **%f**,  
초과하면 **%e**를 쓰는 방식이다.



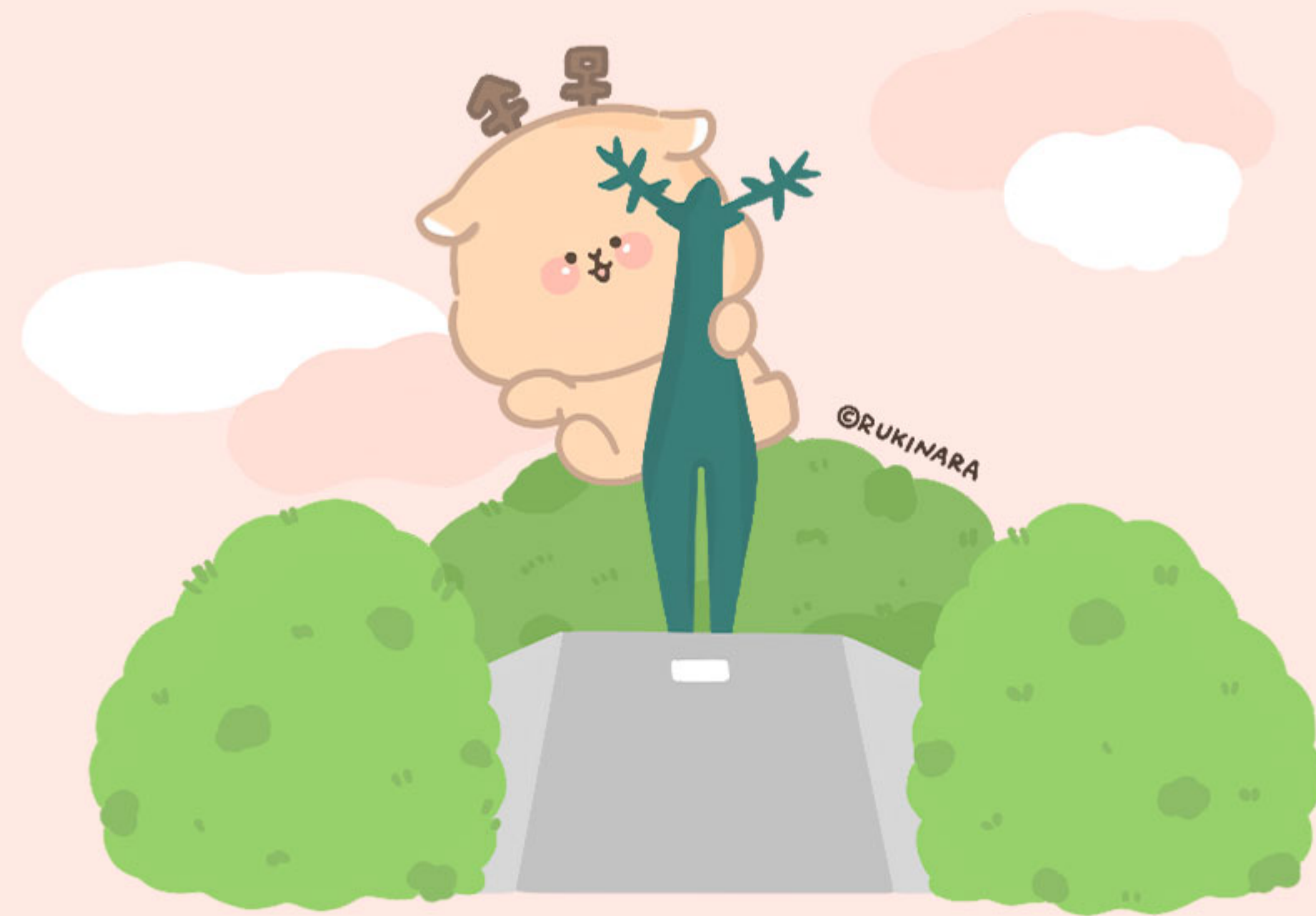
©RUKI



```
2
3  int main (void){
4
5     double d1 = 1.234e-3;
6     double d2 = 1.234e-4;
7     double d3 = 1.234e-5;
8     double d4 = 1.234e-6;
9     printf("%g\n", d1);
10    printf("%g\n", d2);
11    printf("%g\n", d3);
12    printf("%g\n", d4);
13
```

0.001234
0.0001234
1.234e-05
1.234e-06

SCANF



HEIIO, SUMOONG

서식문자	입력 자료형	입력형태
%d, %i	int	부호 있는 10진수 정수
%ld	long	부호 있는 10진수 정수
%lld	long long	부호 있는 10진수 정수
%u	unsigned int (10)	부호 없는 10진수 정수
%o	unsigned int (8)	부호 없는 8진수 정수
%x	unsigned int (16)	부호 없는 16진수 정수
%f , %e, %g	float	부호 있는 실수
%lf	double	부호 있는 실수
%Lf	long double	부호 있는 실수
%c	char	값에 대응하는 문자 (ASCII CODE 표)
%s	char*	문자열
%p	void*	포인터의 주소값



# SCANF

```
1  #include <stdio.h>
2
3  int main (void){
4
5      int num1, num2, num3;
6      printf("세 개의 정수를 입력 : ");
7      scanf("%d %o %x", &num1,&num2,&num3);
8      printf("입력된 정수를 10진수로 출력 : ");
9      printf("%d %d %d\n",num1, num2, num3);
10     return 0;
11 }
```

```
세 개의 정수를 입력 : 12 12 12
입력된 정수를 10진수로 출력 : 12 10 18
```

- scanf( 입력의 형식 , 입력의 장소(주소) )
- 입력 형식(= 서식 문자 ‘앞의 표 참고’)  
입력 장소(변수 이름 앞에 &를 붙인다.)