

Guía de tipos de datos y uso de printf() y scanf() en C

Especificadores de formato, lectura de variables, impresión de datos, control de decimales, ancho de campo y ejemplos prácticos.

1. Tipos de datos básicos en C

Tipo	Ejemplo	printf()	scanf()
char	'A'	%c	%c
short	100	%hd	%hd
int	1000	%d / %i	%d / %i
long	100000L	%ld	%ld
long long	100000LL	%lld	%lld
unsigned int	100	%u	%u
unsigned long	100	%lu	%lu
unsigned long long	100	%llu	%llu
float	3.14	%f	%f
double	3.141592	%f	%lf
long double	3.141592	%Lf	%Lf
char[]	"Hola"	%s	%s

2. printf(): imprimir variables

printf() se utiliza para mostrar información en pantalla. La estructura básica es:

```
printf("Texto %d", variable);
```

Ejemplo con un entero:

```
int edad = 25;
printf("Edad: %d\n", edad);
```

3. Enteros: %d y %i

Para una variable de tipo int se utiliza normalmente %d. %i también puede utilizarse con printf().

```
int numero = 25;
printf("%d\n", numero);
printf("%i\n", numero);
```

4. Números decimales: %f

Para float, printf() utiliza %f. Por defecto, %f muestra seis posiciones decimales.

```
float temperatura = 27.56;
printf("%f\n", temperatura);
```

5. Control de la cantidad de decimales

La forma %.Nf permite indicar cuántas cifras deben mostrarse después del punto decimal.

Formato	Cantidad de decimales	Ejemplo
%.0f	0	28
%.1f	1	27.6
%.2f	2	27.57
%.3f	3	27.568
%.4f	4	27.5679
%.5f	5	27.56789
%.6f	6	27.567890

```
float temperatura = 27.56789;
printf("%.2f\n", temperatura);
```

Resultado: 27.57

6. double

double permite almacenar números de punto flotante con mayor precisión que float. En printf() se utiliza %f; en scanf() se utiliza %lf.

```
double promedio = 4.56789123;
printf("%f\n", promedio);
printf("%.2f\n", promedio);
printf("%.4f\n", promedio);
```

7. char

char almacena un carácter individual. Se escribe entre comillas simples.

```
char letra = 'A';
printf("%c\n", letra);
```

8. Cadenas de texto: %s

Una cadena puede almacenarse en un arreglo de caracteres.

```
char nombre[] = "Hernan";
printf("%s\n", nombre);
```

9. long, long long y unsigned

```
long numero = 123456789;
printf("%ld\n", numero);

long long grande = 1234567890123;
printf("%lld\n", grande);

unsigned int positivo = 25;
printf("%u\n", positivo);
```

10. long double

```
long double numero = 3.141592653589793;
printf("%Lf\n", numero);
printf("%.10Lf\n", numero);
```

11. Notación científica: %e

```
double numero = 1234567.89;
printf("%e\n", numero);
printf("%.2e\n", numero);
```

12. %g

%g utiliza una representación compacta del número y puede alternar entre notación decimal y científica.

```
double numero = 123.456;
printf("%g\n", numero);
printf("%.4g\n", numero);
```

13. Porcentaje: %%

Para imprimir literalmente el símbolo %, se utiliza %%. Esto es especialmente útil para variables expresadas como porcentajes.

```
float humedad = 75.5;
printf("Humedad: %.1f%%\n", humedad);
```

14. scanf(): ingresar datos por teclado

scanf() permite leer datos introducidos por el usuario. En las variables simples se utiliza normalmente & para indicar la dirección de memoria donde se almacenará el dato.

```
int edad;
printf("Ingrese su edad: ");
scanf("%d", &edad);
printf("Su edad es %d\n", edad);
```

15. scanf() con float

```
float temperatura;
printf("Ingrese la temperatura: ");
scanf("%f", &temperatura);
printf("Temperatura: %.2f\n", temperatura);
```

16. scanf() con double

Para double, scanf() requiere %lf. Esta es una diferencia importante respecto de printf().

```
double temperatura;
scanf("%lf", &temperatura);
printf("Temperatura: %.2f\n", temperatura);
```

17. scanf() con char

Se recomienda colocar un espacio antes de %c para ignorar espacios o saltos de línea pendientes.

```
char letra;
scanf(" %c", &letra);
printf("Letra: %c\n", letra);
```

18. scanf() con cadenas

Con un arreglo de caracteres no se coloca & delante del nombre del arreglo.

```
char nombre[30];
scanf("%s", nombre);
printf("Hola %s\n", nombre);
```

19. Varios datos en un mismo scanf()

```
int edad;
float peso;
printf("Ingrese edad y peso: ");
scanf("%d %f", &edad, &peso);
printf("Edad: %d\n", edad);
printf("Peso: %.2f\n", peso);
```

20. printf() con varias variables

```
int edad = 25;
float peso = 70.5;
char inicial = 'H';
printf("Edad: %d, Peso: %.2f, Inicial: %c\n", edad, peso, inicial);
```

21. Ancho de campo

En un formato como %10.2f, el 10 indica un ancho mínimo de campo y .2 indica dos decimales.

```
printf("%10d\n", 25);
printf("%10.2f\n", 25.567);
printf("%-10d\n", 25);
```

El signo - permite alinear el contenido hacia la izquierda; sin él, los valores numéricos se alinean normalmente hacia la derecha.

22. Tabla rápida de printf() y scanf()

Tipo	printf()	scanf()	Observación
char	%c	%c	Carácter
int	%d	%d	Entero
float	%f	%f	Decimal
double	%f	%lf	Mayor precisión
long	%ld	%ld	Entero largo
long long	%lld	%lld	Entero muy largo
unsigned int	%u	%u	Sin signo
char[]	%s	%s	Cadena; no usar & en scanf
long double	%Lf	%Lf	Alta precisión

23. Ejemplo completo

```
#include <stdio.h>

int main() {

    int edad;
    float temperatura;
    double humedad;
    char inicial;
    char nombre[30];

    printf("Ingrese su edad: ");
    scanf("%d", &edad);

    printf("Ingrese la temperatura: ");
    scanf("%f", &temperatura);

    printf("Ingrese la humedad: ");
    scanf("%lf", &humedad);

    printf("Ingrese su inicial: ");
    scanf(" %c", &inicial);

    printf("Ingrese su nombre: ");
    scanf("%s", nombre);

    printf("\n--- RESULTADOS ---\n");
    printf("Nombre: %s\n", nombre);
    printf("Inicial: %c\n", inicial);
    printf("Edad: %d años\n", edad);
```

```
printf("Temperatura: %.2f °C\n", temperatura);  
printf("Humedad: %.2f %%\n", humedad);  
  
return 0;  
}
```

24. Reglas fundamentales para recordar

- printf() muestra información en pantalla.
- scanf() recibe datos desde el teclado.
- En scanf(), las variables simples normalmente llevan &.
- En scanf(), un arreglo char[] usado como cadena no lleva &.
- float usa %f en printf() y scanf().
- double usa %f en printf(), pero %lf en scanf().
- %.2f significa mostrar dos decimales.
- %% permite imprimir literalmente el símbolo %.
- %s se utiliza para cadenas de caracteres.
- %c se utiliza para un carácter individual.

Nota: la precisión mostrada por printf() no cambia el valor almacenado en la variable; solo controla cómo se presenta el número en pantalla.