

Tema 2: Fundamentos de C

Oscar Perpiñán Lamigueiro - David Álvarez

- 1 Primeros pasos
- 2 Datos en C
 - Almacenamiento de la información
 - Datos en un programa en C
 - Caracteres
- 3 Interacción entrada/salida (I/O)
 - Especificaciones de formato
 - Imprimir en el terminal
 - Lectura de datos desde el teclado
- 4 Operadores
 - Aritméticos
 - Comparadores
 - Lógicos
- 5 Conversión de tipos de datos

- 1 Primeros pasos
- 2 Datos en C
 - Almacenamiento de la información
 - Datos en un programa en C
 - Caracteres
- 3 Interacción entrada/salida (I/O)
 - Especificaciones de formato
 - Imprimir en el terminal
 - Lectura de datos desde el teclado
- 4 Operadores
 - Aritméticos
 - Comparadores
 - Lógicos
- 5 Conversión de tipos de datos

Especificadores de formato y secuencias de escape

Formato	Significado
%c	Carácter simple
%d	Entero decimal
%i	Entero decimal, octal o hexadecimal
%e	Coma flotante con exponente
%f	Coma flotante sin exponente
%g	Usa %e o %f, el más corto
%lf	Usar con tipo double
%o	Entero octal, sin el cero inicial
%s	Cadena de caracteres
%u	Entero decimal sin signo
%x	Entero hexadecimal sin 0x.

Secuencia	Significado
\n	Salto de línea
\t	Tabulador horizontal
\"	Comilla doble
\\	Barra invertida

- 1 Primeros pasos
- 2 Datos en C
 - Almacenamiento de la información
 - Datos en un programa en C
 - Caracteres
- 3 Interacción entrada/salida (I/O)
 - Especificaciones de formato
 - **Imprimir en el terminal**
 - Lectura de datos desde el teclado
- 4 Operadores
 - Aritméticos
 - Comparadores
 - Lógicos
- 5 Conversión de tipos de datos

Imprimir frases en el terminal

```
#include <stdio.h>

int main()
{
    // printf para imprimir una frase
    printf("El símbolo del hidrógeno es H, su número atómico es 1, y su masa atómica  
es 1.00794");

    // Y si lo queremos hacer con variables?
    // símbolo
    // número atómico
    // masa atómica

    return 0;
}
```

Tabla periódica

Imprimir datos en el terminal

```
#include <stdio.h>

int main()
{
    // símbolo
    char sim = 'H';
    // número atómico
    int num_at = 1;
    // masa atómica
    float masa_at = 1.00794;

    // printf para imprimir una frase
    printf("El símbolo del hidrógeno es %c, su número atómico es %i, y su masa atómica es %f", sim, num_at, masa_at);

    return 0;
}
```

Imprimir datos en el terminal

```
#include <stdio.h>

int main()
{
    // símbolo
    char sim = 'H';
    // número atómico
    int num_at = 1;
    // masa atómica
    float masa_at = 1.00794;

    // printf para imprimir una frase
    printf("El símbolo del hidrógeno es %c, su número atómico es %i, y su masa atómica es %f", sim, num_at, masa_at);

    return 0;
}
```


Imprimir datos en el terminal, números y caracteres

```
int main()
{

    char sim = 'H'; // símbolo
    int num_at = 1; // número atómico
    float masa_at = 1.00794; // masa atómica

    // printf para imprimir una frase
    printf("El símbolo del hidrógeno es %c, su número atómico es %i, y su masa atómica es %f\n", sim, num_at, masa_at);

    // OJO!! Controlamos cuantos decimales aparecen
    printf("El símbolo del yodo es %c, su número atómico es %i, y su masa atómica es %.3f", sim+1, num_at+52, 126.90447);

    return 0;
}
```

Imprimir en el terminal tamaño de variables

sizeof

```
#include <stdio.h>

int main()
{
    int a = 0;
    printf("Un int ocupa %i bytes", sizeof(a));

    // cual es el tamaño de las otras variables?

    // para que sirven las palabras de C long y short?

    return 0;
}
```

Imprimir en el terminal tamaño de variables, sizeof

```
#include <stdio.h>

int main()
{
    // cual es el tamaño de las otras variables?
    // para que sirven las palabras de C long y short?
    // cambia el tamaño si uso unsigned y signed?

    printf("Un int ocupa %i bytes\n", sizeof(int) );
    printf("Un float ocupa %i bytes\n", sizeof (float) );
    printf("Un short int ocupa %i bytes\n", sizeof(short int) );

    return 0;
}
```

Ejercicio: combinar en un programa, todos los tipos (char, int, float, double), con los modificadores (unsigned, signed, long, short)

Tipos de datos especiales

```
#include <stdio.h>
int main() {
    char a; // caracteres, guardados como enteros sin signo
    int b; // números enteros, 4 Bytes (B)
    float c; // números decimales, 4 B, hasta 6 cifras decimales

    double d; // números decimales, 8 B, hasta 14 cifras decimales

    // tipos compuestos, ejemplos
    unsigned int e; // solo numeros naturales y el 0
    short int f; // números enteros, 2 B

    long double g; // números decimales, 8 B, hasta 14 cifras decimales

    printf("int ocupa: %i B \t short int: %i B\n", sizeof(int), sizeof(short int));
    printf("double ocupa: %i B \t long double: %i B\n", sizeof(double), sizeof(long
        double));
    return 0;
}
```

- 1 Primeros pasos
- 2 Datos en C
 - Almacenamiento de la información
 - Datos en un programa en C
 - Caracteres
- 3 Interacción entrada/salida (I/O)
 - Especificaciones de formato
 - Imprimir en el terminal
 - Lectura de datos desde el teclado
- 4 Operadores
 - Aritméticos
 - Comparadores
 - Lógicos
- 5 Conversión de tipos de datos

Lectura de datos con scanf

scanf("formato", &variable);

```
#include <stdio.h>
int main()
{
    // Primero debo reservar la memoria donde guardar la información
    int num;

    // Pedir el dato por escrito es OPCIONAL, pero deseable
    printf("Escribe un número \t");

    //Atención: SCANF, la variable debe ir precedida de &
    scanf("%i", &num);

    printf("Has escrito el número %i\n", num);

    return 0;
}
```

Lectura distintos tipos de datos con scanf

Ejercicio

```
#include <stdio.h>
int main()
{
    int num_ent;
    float num_dec;
    char character;
    double num_double;

    printf("Escribe un numero \t");
    scanf("%i", &num_ent);
    ...
    // Rellenar el resto
    ...
    return 0;
}
```

Errores comunes con scanf

- Escribir dentro de la cadena de control mensajes y secuencias de escape (p.ej. `\n`).
- Olvidar poner el operador `&` delante de los argumentos cuando son variables de los tipos básicos (`int`, `float`, `double`, `char`)
- Poner un especificador de formato no compatible con el tipo del argumento.

Lectura de caracteres, scanf y getchar

```
scanf("%c", &variable);
```

```
variable = getchar();
```

- **scanf** vale para cualquier tipo de datos, mientras que **getchar** solo sirve para leer caracteres.
- ¿Qué pasa cuando queremos leer 2 caracteres de manera consecutiva?

Ejercicio: leer los caracteres de tu nombre para luego imprimirlos por pantalla. También leerlos como **int** e imprimirlos como **char**, y al revés.

Lectura de varios caracteres seguidos, fflush

```
#include <stdio.h>
int main()
{
    // Primero debo reservar la memoria donde guardar la información
    char ejemplo1, ejemplo2;

    // Pedir el dato por escrito es OPCIONAL, pero deseable
    printf("Escribe un caracter \t");
    ejemplo1 = getchar();

    printf("Escribe otro caracter \t");
    fflush(stdin);
    scanf("%c", &ejemplo2);

    printf("Has los caracteres %c y %c\n", ejemplo1, ejemplo2);

    return 0;
}
```