

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
- Números reales
- Lectura de datos desde el teclado

4 Operadores

- Tipos
- Operaciones con variables
- Conversión de tipos de datos

Hello World!

```
#include <stdio.h>

int main()
{
    printf("Hello World!");

    return 0;
}
```

Hello World! (2)

```
#include <stdio.h>

int main()
{
    printf("Hello World!\n");
    printf(";Hola Mundo!\n");
    printf("Bonjour le Monde!\n");
    printf("Hallo Welt!\n");

    return 0;
}
```

Comentarios

```
/** Este simple programa sirve para
    mostrar un mensaje en pantalla */
#include <stdio.h>

// Todo programa necesita una función main.
// Su contenido está delimitado entre llaves
int main()
{
    //La función printf muestra el mensaje en pantalla
    // Atención: el mensaje debe ir entre comillas
    printf("Hello World!\n");
    // La función main devuelve un entero con return.
    return 0;
} // Aquí acaba main y por tanto el programa
```

Palabras clave en C

Modificadores de tipo	Tipos primitivos	Tipos datos avanzados	Sentencias de control
auto	char	enum	break
extern	double	struct	case
register	float	typedef	continue
static	int	union	do
const	long		else
volatile	short		if
signed	void		for
unsigned	sizeof		while
			return
			goto
			default
			switch

- 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
 - Números reales
 - Lectura de datos desde el teclado
- 4 Operadores
 - Tipos
 - Operaciones con variables
 - 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
 - Números reales
 - Lectura de datos desde el teclado
- 4 Operadores
 - Tipos
 - Operaciones con variables
 - Conversión de tipos de datos

- Los ordenadores utilizan el sistema de numeración binario (dos dígitos, 0 y 1) para almacenar información.
- Un **dígito binario** (0 ó 1) se denomina *bit* (*binary digit*).
- Con **N bits** pueden representarse **2^N símbolos o 2^N números**
 - ▶ Ejemplo: con $N = 8$ bits se pueden representar los números positivos del 0 al 255 ($2^8 - 1$).

Representación de la información: binario y decimal

Ejemplo en decimal: 3452

10^3	10^2	10^1	10^0
1000	100	10	1
<hr/>			
3	4	5	2

$$3452 = 3 \cdot 1000 + 4 \cdot 100 + 5 \cdot 10 + 2 \cdot 1$$

Ejemplo en binario: 10001111

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
<hr/>							
1	0	0	0	1	1	1	1

$$128 + 8 + 4 + 2 + 1 = 143$$

Unidades de almacenamiento

- Byte: 8 bits ($2^8 = 256$)
- Kilobyte (KB): 1024 bytes ($2^{10} = 1024$)
- Megabyte (MB): 1024 KB (2^{20} bytes = 2^{10} KB)
- Gigabyte (GB): 1024 MB (2^{30} bytes = 2^{10} MB = 2^{20} KB)
- ...

- 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
 - Números reales
 - Lectura de datos desde el teclado
- 4 Operadores
 - Tipos
 - Operaciones con variables
 - Conversión de tipos de datos

Tipos de datos

`int` números enteros. Pueden ser **signed** o **unsigned**.

100 -41 0 12345

`float y double` números reales

3.0 101.2345 -0.0001 2.25e-3

`char` caracteres

's' '4' ';' '

`_Bool` *booleanos*, 0 y 1

Nombres de constantes y variables

- Primer carácter: letra o carácter de subrayado (_) (**nunca un número**).
- Una o más letras (A-Z, a-z, *ñ excluida*), dígitos (0-9) o caracteres de subrayado.
- Tienen que ser distintos de las palabras clave.
- Las mayúsculas y las minúsculas son diferentes para el compilador.
- Es aconsejable que los nombres sean representativos

Constantes y Variables: declaración

- Son espacios de memoria que contienen datos. Se declaran para reservar la memoria en el ordenador.
- Declarar significa definir el tipo, asignarle un nombre y darle, si es necesario, un valor inicial.

tipo nombre=valor_inicial;

Constantes datos cuyo valor NO SE PUEDE MODIFICAR (**const**).

Variables datos cuyo valor se puede modificar mediante el operador *asignación* (=)

Definición de constantes y variables enteras

```
#include <stdio.h>
int main()
{
    // declara una variable ENTERA con el identificador v1
    int v1;
    // declara una constante ENTERA con el identificador c1
    const int c1 = 4;
    // declara una variable ENTERA v2, y le asigna el valor 2
    int v2 = 2;
    // asigna el valor de la constante c1 a la variable v1 (cambia su valor previo)
    v1 = c1;
    // asigna el valor de la constante c1 a la variable v1
    v2 = c1;
    // error: c1 es una constante
    c1 = 3;

    return 0;
}
```


Definición de constantes y variables decimales

```
#include <stdio.h>
int main()
{
    // declara una variable DECIMAL con el identificador v1
    float v1;
    // declara una constante DECIMAL con el identificador c1
    const float c1 = 4;
    // declara una variable DECIMAL v2, y le asigna el valor 2
    float v2 = 2;
    // asigna el valor de la constante c1 a la variable v1 (cambia su valor previo)
    v1 = c1;
    // asigna el valor de la constante c1 a la variable v1
    v2 = c1;
    // es esto válido?
    c1 = 3;

    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
 - Números reales
 - Lectura de datos desde el teclado
- 4 Operadores
 - Tipos
 - Operaciones con variables
 - Conversión de tipos de datos

No sólo números - caracteres

- Cualquier información puede representarse con un conjunto de bits.
- ASCII (American Standard Code for Information Interchange): Estándar de 7 bits (128 caracteres), 95 caracteres imprimibles (del 32 al 126).

01001000	01101111	01101100	01100001
72	111	108	97
H	o	l	a

Tabla ASCII (no está entera)

32		44	,	56	8	68	D	80	P	92	\	104	h	116	t
33	!	45	-	57	9	69	E	81	Q	93]	105	i	117	u
34	"	46	.	58	:	70	F	82	R	94	^	106	j	118	v
35	#	47	/	59	;	71	G	83	S	95	_	107	k	119	w
36	\$	48	0	60	<	72	H	84	T	96	`	108	l	120	x
37	%	49	1	61	=	73	I	85	U	97	a	109	m	121	y
38	&	50	2	62	>	74	J	86	V	98	b	110	n	122	z
39	'	51	3	63	?	75	K	87	W	99	c	111	o	123	{
40	(52	4	64	@	76	L	88	X	100	d	112	p	124	
41)	53	5	65	A	77	M	89	Y	101	e	113	q	125	}
42	*	54	6	66	B	78	N	90	Z	102	f	114	r	126	~
43	+	55	7	67	C	79	O	91	[103	g	115	s		

Definición de constantes y variables de tipo caracter

```
#include <stdio.h>

int main()
{
    //declara una variable de tipo CHAR con el identificador letra
    char letra = 'z';
    // declara una constante CHAR con el identificador c1
    const char c1 = '8';
    // asigna el valor de la constante c1 a la variable letra
    letra = c1;
    // Y char con un número? SI. Usando la Tabla ASCII. Qué caracter es?
    letra = 122;
    // es esto válido?
    c1 = 122;

    return 0;
}
```