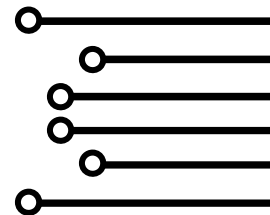




C Embebido





HOLA MUNDO EN C

Especificadores de formato

Los especificadores de formato definen los datos que se imprimirán.

SPECIFIER	USED FOR
%c	a single character
%s	a string
%hi	short (signed)
%hu	short (unsigned)
%Lf	long double
%n	prints nothing
%d	a decimal integer (assumes base 10)
%i	a decimal integer (detects the base automatically)
%o	an octal (base 8) integer
%x	a hexadecimal (base 16) integer

Especificadores de formato

Los especificadores de formato definen los datos que se imprimirán

```
int main(void)
{
    printf("Valor del dato = %d",5+3);
}
```

SPECIFIER	USED FOR
%c	a single character
%s	a string
%hi	short (signed)
%hu	short (unsigned)
%Lf	long double
%n	prints nothing
%d	a decimal integer (assumes base 10)
%i	a decimal integer (detects the base automatically)
%o	an octal (base 8) integer
%x	a hexadecimal (base 16) integer

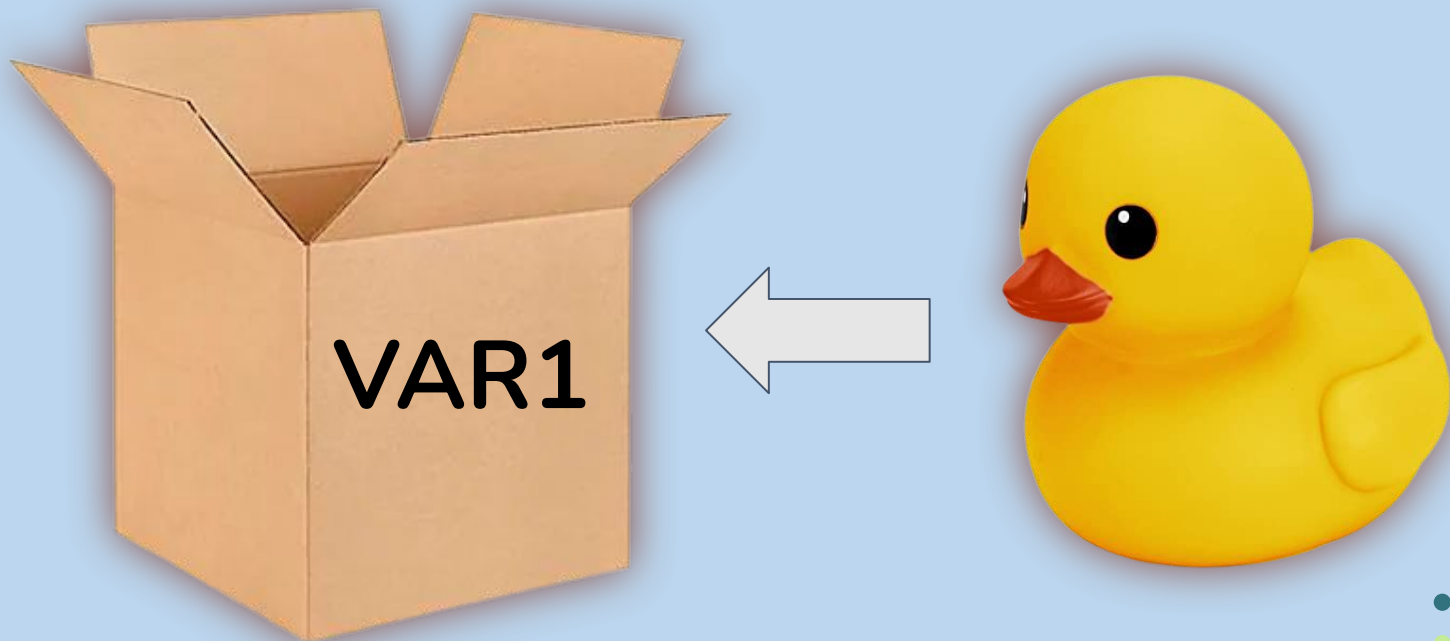
```
int main(void)
{
    printf("Valor del dato = %d",5+3);
}
```

Pero qué pasa si queremos cambiar esa suma o los datos, ¿podríamos hacerlo?

Variables



Variables

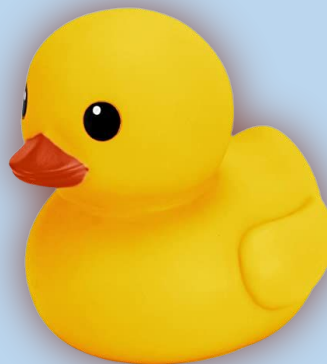


Variables



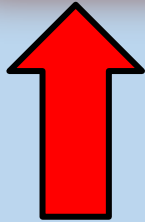
Variables

```
var1 = "patito de hule"
```



Variables

```
var1 = "patito de hule"
```



Debemos declarar que tipo de variable es.

Tipos de datos

En C, se utiliza para declarar qué tipo de variable es y el tamaño de data.

- Data -> Número entero o decimal.
- Data -> Caracter
- Data -> Cadena (Colección de caracteres)

Wels



STM32 CubeIDE



USAREMOS EL STM32CubeIDE

Tipos de datos

CHAR

Sizeof

Unsigned
Signed

INT

Conozcamos mejor CHAR e INT

CHAR

- A “***char***” consume 1 byte de memoria.
- Se utiliza para almacenar un caracter ASCII.

0 NUL	16 DLE	32	48 0	64 @	80 P	96 `	112 p
1 SOH	17 DC1	33 !	49 1	65 A	81 Q	97 a	113 q
2 STX	18 DC2	34 "	50 2	66 B	82 R	98 b	114 r
3 ETX	19 DC3	35 #	51 3	67 C	83 S	99 c	115 s
4 EOT	20 DC4	36 \$	52 4	68 D	84 T	100 d	116 t
5 ENQ	21 NAK	37 %	53 5	69 E	85 U	101 e	117 u
6 ACK	22 SYN	38 &	54 6	70 F	86 V	102 f	118 v
7 BEL	23 ETB	39 '	55 7	71 G	87 W	103 g	119 w
8 BS	24 CAN	40 (56 8	72 H	88 X	104 h	120 x
9 HT	25 EM	41)	57 9	73 I	89 Y	105 i	121 y
10 LF	26 SUB	42 *	58 :	74 J	90 Z	106 j	122 z
11 VT	27 ESC	43 +	59 ;	75 K	91 [107 k	123 {
12 FF	28 FS	44 ,	60 <	76 L	92 \	108 l	124
13 CR	29 GS	45 -	61 =	77 M	93]	109 m	125 }
14 SO	30 RS	46 .	62 >	78 N	94 ^	110 n	126 ~
15 SI	31 US	47 /	63 ?	79 O	95 _	111 o	127 DEL

CHAR

Rango

- Para enviar caracter se utiliza la comilla simple 'caracter'.
- `char var = '1';`

```
char var_s_char = '1';
```

CHAR

Rango

Un “***char***” consume 1 byte de memoria.

- Rango de char:
 - **Signed:** -128 a 127.
 - **Unsigned:** 0 a 255.

```
unsigned char var_unsig;
```

```
signed char var_sig;
```

```
unsigned char var_car;
```


SIZE OF

- Se utiliza para ver el tamaño de la variable.

```
sizeof(var_sig);
```

¡Veamos un ejemplo!



Signed Unsigned

Unsigned

- Se refiere a una variable que puede almacenar números cero y positivos.

Signed

- Se refiere a una variable que puede almacenar números negativos, cero y positivos.

Signed
Unsigned

Signed char

- Rango de: -128 a 127.

-25



El bit más significativo representa el signo del dato
1: El dato es negativo 0: El dato es positivo

Signed
Unsigned

Signed char

- Rango de: -128 a 127.

-25



El bit más significativo representa el signo del dato
1: El dato es negativo 0: El dato es positivo

Signed
Unsigned

Signed char

- El valor del dato es almacenado en complemento 2.

-25



0xE7

Signed
Unsigned

Signed char

- El valor mínimo negativo es -128

7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0

- El valor máximo negativo es -1

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1

Signed
Unsigned

Signed char

- El valor mínimo positivo es 1

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

- El valor máximo positivo es 127

7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1

Signed
Unsigned

Unsigned char

- El valor mínimo es 0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

- El valor máximo positivo es 255

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1

INT

Int

- Tipo de dato entero, puede consumir 2 bytes o 4 bytes de memoria. Depende del hardware.
- Rango:
 - **Unsigned:** 0 a 4294967295
 - **Signed:** -2147483648 a 2147483647

```
unsigned int var_unsig;
```

```
signed int var_sig;
```

INT

Short Int

- Usada para almacenar datos enteros y de 2 bytes de tamaño. Depende del hardware.
- Rango:
 - **Unsigned:** 0 a 65535
 - **Signed:** -32768 a 32767

```
unsigned short var_unsig;
```

```
signed short var_sig;
```

INT

Long

- Consume 4 bytes o 8 bytes de memoria. Depende del hardware.
- Rango:
 - **Unsigned:** 0 a $(2^{64} - 1)$
 - **Signed:** -2^{63} a $(2^{63} - 1)$

```
unsigned long var_unsig;
```

```
signed long var_sig;
```



Gracias

@welstheory

hola@welstheory.com

+51 918 899 684

