Ejemplo simple

```c
#include <stdio.h>
/* Esto es un comentario
   en varias líneas*/

int main()
{
        printf("Hello world!\n");
        // otro comentario: printf es una función (de stdio)
        return 0;
}
```

Estructura general (simplificada) de un programa C:

- Directivas del preprocesador
- Definiciones de tipos de datos
- Implementación de funciones

Tipos de datos

- char : Caracteres (en realidad números que representan caracteres)
- int : Números enteros, de distintos tamaños (short, long, long long)
- float : Punto flotante de 32 bits
- double : Punto flotante de 64 bits
- long double : Punto flotante de mayor precisión 80 a 128 bits
- bool: agregado recién en C99 es en realidad 0 o 1

Diferentes estándares

- C K&R
  - El explicado en el libro “El lenguaje C” de 1978
  - Presenta cambios respecto al C original (1972)
- ANSI C
  - ANSI X3.159-1989 conocido como C89
  - ISO/IEC 9899:1990 conocido como C90 (es prácticamente idéntico a C89)
- C99
  - ISO/IEC 9899:1999
- C11 (por 2011)
  - ISO/IEC 9899:2011

Variables, constantes y literales

Declaración de variables

tipo_dato nombre_variable;

```
char c;
unsigned long cantidad;
int i;
```

tipo_dato nombre_variable = valor_inicial;

```
char letra = 'A';
unsigned long total = 0xA01D7F;  // 10493311UL
int dato = 2;
bool flag = true;
```

tipo_dato nombre_variable1 , nombre_variable2;

```
float fuerza, gramos;
short vueltas, envios;
```

tipo_dato nombre_variable1[= valor_inic_1] [,nombre_variable2[= valor_inic_2]] … ;

```
unsigned char c = 'A' , letra = 133;
int i = 2, j = 3;
double aceleracion = 3.0, masa = -5.44;
```

Variables, constantes y literales

Constantes y literales

Modo “tradicional”

```
#define PI 3.14159
perimetro = 2 * PI * radio;
```

Modo tomado de C++ (introducido en ANSI C)

```
const double PI = 3.14159;
perimetro = 2 * PI * radio;
PI = 4; // ERROR al compilar !!
```

Literales

```
3             //entero
true, false   //booleanos
'a'           //caracter
“Hola”        //String → en realidad “arreglo (vector) de char”
3.5           // punto flotante
'\n'          // line feed (10 o 0xA)
```

Secuencias de escape de caracteres

Código	Significado	Valor ASCII (Decimal)	Valor ASCII (Hexadecimal)
'\n'	Nueva línea (dependiente SO)	10	0x0A
'\r'	Retorno de carro	13	0x0D
'\t'	Tabulador (horizontal)	09	0x09
'\f'	Nueva página	12	0x0C
'\a'	Alerta (campana)	07	0x07
'\b'	Retroceder un caracter	08	0x08
'\v'	Tabulador (vertical)	11	0xB
'\\'	Barra invertida	92	0x5C
'\''	Comilla simple	39	0x27
'\"'	Comilla doble	34	0x22
'\ddd'	El caracter ASCII cuyo código sea ddd en octal		
'\xhh'	El caracter ASCII cuyo código sea nn en hexadecimal		
Nota: Este listado no es completo			

Operadores

- Aritméticos
  - Asignación → =
  - Suma, resta → + , -
  - Multiplicación, división → * , /
  - Módulo → %
- Relacionales
  - Menor, mayor → < , >
  - Menor o igual, mayor o igual → <= , >=
  - Igual, distinto → == , !=

Operadores

- Lógicos
  - Not (Negación) → !
  - y → &&
  - o → ||
- Bits
  - y → &
  - o (inclusivo) → |
  - o (exclusivo) → ^
  - Desplazamiento a derecha → >>
  - Desplazamiento a izquierda → <<
  - Complemento a uno → ~

Operadores

- Incremento y decremento → ++ , --
  - Pre: ++i (incremento, luego uso)
  - Post: i++ (uso, luego incremento)
- Operar y asignar
  - Los operadores aritméticos y de manejo de bits pueden combinarse con la asignación
  - Ejemplo de un sumador
    - `total = total + dato;`
    - `total += dato;`
  - Genéricamente si el operador es X entonces
    - `a = a X b;` ≡ `a X= b;`
    - **Nota**: En realidad `a X= b` ≡ `a = a X (b)`

Ingreso y Egreso de datos

- Funciones printf y scanf
  - Pertenecen a la biblioteca estándar “stdio” (standard input output)
  - Se utilizan secuencias de escape para indicar donde van los datos. Estas secuencias comienzan con %
  - Supongamos que x es una variable int con valor 3

```
printf("Valor inicial: %d \t valor final: %d\n", x * 2, 7);
```

  - Genera como salida

```
Valor inicial: 6    valor final: 7
```

  - Para ingresar un valor en la variable dato

```
scanf("%d", &dato);
```

Secuencias de escape en E/S

Secuencia	Uso
d,i	Números enteros en base 10
o	Enteros en base octal
X,x	Enteros en base Hexadecimal X usa letras mayúsculas, x usa minúsculas
c	Caracter
s	String
f	double en printf (**float en scanf**)
%	Para poder mostrar un % en la salida
Modificadores	
l para long (por ej: ld para long int) L para long double (se usa Lf)	

Precedencia de los operadores

Operators	Associativity
`() [] -> .`	left to right
`! ~ ++ -- + - *` (*type*) `sizeof`	right to left
`* / %`	left to right
`+ -`	left to right
`<< >>`	left to right
`< <= > >=`	left to right
`== !=`	left to right
`&`	left to right
`^`	left to right
`|`	left to right
`&&`	left to right
`||`	left to right
`?:`	right to left
`= += -= *= /= %= &= ^= |= <<= >>=`	right to left
`,`	left to right

Licencia