

Tema 2 Elementos básicos en C

Tema 2.- Elementos básicos

- 2.1. Estructura general de un programa
- 2.2. Tipos de datos simples
- 2.3. Funciones de entrada / salida

2.1 Estructura de un programa

Un programa en C es un conjunto de funciones

Una función es un conjunto de instrucciones que realizan una o más acciones (ver tema 5)

2.1 Estructura de un programa

Directivas del preprocesador

Declaraciones globales

Función **main**

Definición de otras funciones

Directivas del preprocesador
Declaraciones globales
Función main
Definición de funciones

Directivas del preprocesador

- El preprocesador es una colección de instrucciones especiales (*directivas*) que se ejecutan al principio de la compilación.

Directivas más habituales

- #include**. Para incluir ficheros de cabecera
- #define**. Para definir constantes

Directivas del preprocesador
Declaraciones globales
Función main
Definición de funciones

Directivas del preprocesador

Estructura (sintaxis)

```
#include <nombreFichero.h>  
#include "nombreFichero.h"
```

Para ficheros de cabecera existentes en el compilador

Para ficheros de cabecera en el directorio actual

Ejemplo:

```
#include <stdio.h>  
#include "misfunciones.h"
```

Directivas del preprocesador
Declaraciones globales
Función main
Definición de funciones

Directivas del preprocesador

Estructura (sintaxis)

```
#define identificador valor
```

Nombre de la constante

Valor para la constante

Secuencia de letras, dígitos y subrayado que comienza con una letra.
C distingue entre mayúsculas y minúsculas

Ejemplo:

```
#define MAYOREDAAD 18
#define PI 3.1416
```

Ejemplo identificadores

- encarna
- miPrimerIdentificador
- mi primer identificador
- mi_primer_identificador
- 2005anno
- anno2005
- mi.primer.identificador

¿Identificadores válidos?

Directivas del preprocesador
Declaraciones globales
Función main
Definición de funciones

Declaraciones globales

- Declaración de variables y prototipos de funciones accesibles desde cualquier punto del programa
- Una variable representa la información que cambia durante la ejecución
- Siempre que necesitemos almacenar información trabajaremos con alguna variable

Directivas del preprocesador
Declaraciones globales
Función main
Definición de funciones

Declaraciones globales

- Declaración de variables

Estructura (sintaxis)

```
tipoDeDato listaDeIdentificadores = valor ;
```

Opcional

Un identificador o más separados por comas

Ejemplo:

```
int a,b,c;
int peso=0;
int k=0,l=0;
```

Directivas del preprocesador
Declaraciones globales
Función main
Definición de funciones

Función main()

- Punto de entrada al programa
- Parte ejecutable de todo programa
- Sólo puede haber una función main

Estructura (sintaxis)

```
main()
{
... <bloque de sentencias>
}
```

Directivas del preprocesador
Declaraciones globales
Función main
Definición de funciones

Definición de funciones

- Zona de declaración de las funciones definidas por el usuario

```
/* comentario en C */
```

2.2 Tipos de datos simples

- Enteros
- Reales
- Caracteres
- Lógicos

2.2 Enteros

Tipo en C	Rango de valores
int	-32.768 .. +32.767
unsigned int	0 .. +65.535
short int	-128 .. +127
long	-2147483648 .. 2147483647
unsigned long	0 .. 4294967295

Ejemplo:

```
int a,b,c;  
short int peso=0;
```

2.2 Constantes enteras

- Secuencia de dígitos. (12458)
- Constantes `long` terminarlas con la letra `L` ó `l` (1245L).
- Constantes `unsigned` terminarlas con la letra `u` (425u)
- Se pueden combinar los sufijos (435uL)

2.2 Reales

Tipo en C	Rango de valores	Precisión
float	3.4×10^{-38} .. 3.4×10^{38}	7 dígitos
double	1.7×10^{-308} .. 1.7×10^{308}	15 dígitos
long double	3.4×10^{-4932} .. 3.4×10^{4932}	19 dígitos

Ejemplo:

```
float f;  
double d=0.0;
```

2.2 Constantes reales

- Existen dos notaciones para los reales
 - Notación punto
digitos.digitos Ejemplo: 34.7
 - Notación científica
enteroOreal eóE enteroConOsinSigno
Ejemplos: 5.4e10 (equivale a 4.4×10^{10})
345E-3 (equivale a 345×10^{-3})

2.2 Caracteres

Tipo en C	Rango de valores
char	0 .. 255

- Cualquier carácter ASCII
- Internamente los caracteres se almacenan como números (A es la 65, B 66, ...)

2.2 Constantes carácter

- Cualquier carácter ASCII entre comillas simples.
Ejemplo: 'A'
- Caracteres de escape más comunes

'\n'	nueva línea
'\t'	tabulación
'\v'	tabulación vertical
'\a'	alerta (pitido)
'\\'	barra inclinada inversa
'\''	comilla simple
'\"'	comilla doble
'\?'	signo de interrogación

2.2 Tipo lógico

- En ANSI C no existe este tipo de dato.
- Simula este tipo usando `int`
- Para C cualquier valor distintos de 0 se considera verdadero y 0 falso

2.2 Constante cadena

- Es una secuencia de caracteres entre comillas dobles
Ejemplo: "pedazo de cadena"

2.3 Funciones de Entrada Salida

- Las funciones de entrada salida más habituales se encuentran en el archivo de cabecera `stdio.h`
- La entrada estándar será el teclado y la salida la pantalla

2.3 Salida

```
printf(cadenaDeControl, listaExpresiones)
```

Contiene códigos de formato que se asocian a cada uno de los datos a visualizar

Conjunto de expresiones separadas por comas

2.3 Salida

Códigos de formato más utilizados

%d	El dato se convierte a entero decimal
%c	El dato se considera de tipo carácter
%e	El dato se considera de tipo <code>float</code> , en notación científica
%f	El dato se considera de tipo <code>float</code> en notación decimal
%g	El dato se considera de tipo <code>float</code> (notación más corta)
%s	El dato es una cadena de caracteres
%lf	El dato se considera de tipo <code>double</code>
%ld	El dato se considera de tipo entero largo
%o	El dato entero se convierte a octal
%x	El dato entero se convierte a hexadecimal
%u	El dato entero se convierte a entero sin signo

2.3 Salida

```
#include<stdio.h>
char c;
int a;
float f;
```

```
main(){
  c='A';
  a=25;
  f=3.45;
```

```
  printf("hola que tal\n");
  printf("El valor es %d",a);
  printf("f es %f\n",f);
  printf("f es %e\n",f);
  printf("%c",c); }
```

Salto de línea después de visualizar la cadena

```
hola que tal
El valor es 25f es 3.450000
f es 3.450000e+000
A
```

Formato para visualizar un entero

2.3 Salida

```
#include<stdio.h>
int a;
float f;
```

```
main(){
  a=25;
  f=3.45;
```

```
  printf("el valor %5d\n",a);
  printf("f es %.2f\n",f);
}
```

Se puede especificar la anchura en la que visualizar en entero

```
El valor es 25
f es 3.45
```

Para especificar el número de decimales

2.3 Entrada

```
scanf(cadenaDeControl, listaVariables)
```

Contiene códigos de formato a leer. Son los mismos que para la entrada

Conjunto de variables a leer. Hay que pasarle la dirección (&)

2.3 Entrada

```
#include<stdio.h>
```

```
int edad;
float peso;
```

```
main(){
  printf("Introduce tu edad y peso ");
  scanf("%d %f",&edad,&peso);

  printf("\n Tienes %d años",edad);
  printf("\n Pesas %.1f kilos",peso);
}
```

Dirección de las variables edad y peso

```
Introduce tu edad y peso 25 50.5
Tienes 25 años
Pesas 50.5 kilos
```