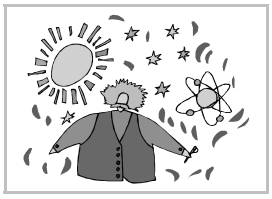

Elementos fundamentales

Palabras reservadas

En un lenguaje de programación se llaman palabras reservadas a aquellas que no puede usar libremente el programador, puesto que tienen significados específicos en el lenguaje. C es uno de los lenguajes que tiene menos palabras reservadas. Son éstas (siempre en minúsculas):

```
auto      const     double    float     int       short     struct    unsigned
break     continue  else      for       long      signed    switch    void
case      default   enum      goto      register  sizeof    typedef   volatile
char      do        extern    if        return    static    union     while
```

Hay que añadir también las palabras reservadas por las librerías, así como las que incorporan algunos compiladores para extender las capacidades del lenguaje.

Comentarios

En C se introducen comentarios en el código comenzando con `/*` y terminando por `*/`. Los comentarios pueden ocupar varias líneas y situarse en cualquier lugar respecto al resto del código. No está permitido *anidar* comentarios, es decir, escribir comentarios dentro de otros comentarios.

Sentencias

Son la materia prima de un programa. En principio, cada instrucción que se escribe es una sentencia. Todas las sentencias simples deben acabar en punto y coma. Se admite la sentencia vacía: sólo el punto y coma. Hay sentencias compuestas, que consisten en varias sentencias simples encerradas entre llaves.

Identificadores

Los identificadores son las palabras que introduce el programador para referirse a los distintos elementos que necesita: variables, funciones, estructuras, etc. Los identificadores pueden contener letras (del inglés), dígitos y caracteres de subrayado, pero el primer carácter debe ser una letra. Serán tenidos en cuenta los 31 primeros caracteres, aunque se pueden usar más. El C distingue las mayúsculas de las minúsculas, de modo que los identificadores **`Fecha`**, **`fecha`** y **`FECHA`** son diferentes. Un identificador nunca puede coincidir con una palabra reservada.

Tipos de datos

Hay cuatro tipos básicos de datos en C, aunque cada uno de ellos admite variantes.

- Tipo **carácter**. Sólo admite la variante **`char`**. Se utiliza para designar caracteres, aunque internamente se consideran números enteros de 8 bits. Los caracteres se deben escribir entre comillas simples. Ejemplos: **`'A'`**, **`'c'`**.
- Tipo **entero**, denominado int, pero que admite cuatro variantes. Es útil para describir números enteros de 16 ó 32 bits. Ejemplos: **`23`**, **`-14`**.
- Tipo **coma flotante**, con las variantes **`float`** y **`double`**. Se usa para representar números con decimales. Es obligatorio escribirlos con un punto decimal. Ejemplos: **`0.12`**, **`3.0`**, **`-0.003`**.
- Tipo **vacío**, llamado **`void`**. es útil tanto para representar la ausencia de tipo como para hacer manipulaciones avanzadas de tipos.

Constantes

Las constantes se representan cada una según su tipo de dato, tal como se ha explicado más arriba; pero, a efectos prácticos, para representar constantes se suele utilizar el precompilador. Por ejemplo, para designar con el identificador **`PULGADA`** el número en coma flotante **`2.54`** se escribe:

```
#define PULGADA 2.54
```

Así, el precompilador sustituirá las referencias a **`PULGADA`** por el número **`2.54`**. Esto recibe el nombre de **macro constante**. Es costumbre escribir los macros constantes en mayúsculas, aunque no es obligatorio.

Variables

Son zonas de memoria RAM reservadas por el compilador para almacenar un valor. Las variables se nombran con un identificador. Siempre se deben declarar antes de usar. Su valor puede cambiar en cualquier momento que decida el programador.

Tipos de variables

Según dónde se declaren, existen dos tipos de variables:

- Variables **locales**. Se declaran dentro de una función, antes de cualquier otro tipo de operación. Las variables locales sólo se pueden utilizar dentro de la función en que se definen, y desaparecen cuando el flujo del programa sale de ella.
- Variables **globales**. Se declaran fuera de todas las funciones del programa, y son accesibles desde todas ellas. En general, se desaconseja su uso.

Declaración de variables

En cada sentencia se escribe el tipo de las variables y a continuación la lista de variables de ese tipo, separadas por comas. Ejemplo:

```
int   i, j, k;       /* Tres índices             */
float Inicio, Fin;   /* Valores inicial y final  */
char  Respuesta;     /* La letra que se contesta */
```

Operadores y expresiones

Los operadores son símbolos que representan operaciones matemáticas. Por ser éste un curso elemental, de simple iniciación, sólo se explicarán los operadores más sencillos, en C existen más. Las expresiones son combinaciones de constantes, variables y operadores.

Operador	Signo
Suma	+
Diferencia	-
Producto	*
Cociente	/
Módulo	%
Incremento	++
Decremento	--

Operadores aritméticos

A la derecha se ve una tabla con siete operadores aritméticos y sus significados. Se usan según las reglas habituales del álgebra, y por supuesto se pueden añadir paréntesis donde sea necesario.

Operador de asignación

Se representa con el signo **=**. Permite asignar a una variable, que se pondrá a la izquierda, el valor de una expresión, que estará a la derecha. Ejemplos:

```
Inicio = 8 * PULGADA + 0.34;  /* PULGADA es un macro                  */
Fin = (i+j) / 2.0;            /* Fin es la media de i y j             */
k = j++;                      /* Se asigna j a k y luego se incrementa */
```

Operadores lógicos

Una expresión con estos operadores podrá ser verdadera (valor 1) o falsa (valor 0). Por tanto, se suelen utilizar para escribir condiciones y a partir de ellas tomar decisiones. Se ven aquí:

Operador	Significado	Operador	Significado	Operador	Significado
<	Menor	>=	Mayor o igual	&&	Y
<=	Menor o igual	==	Igual	||	O
>	Mayor	!=	Distinto	!	No

Ejemplos:

```
i<j || k>0     /* i es menor que j o k es positivo   */
a!=2 && a!=7   /* a es distinto de 2 y distinto de 7 */
```