



APRENDERAPROGRAMAR.COM

TIPOS DE DATOS EN C.
DECLARAR E INICIALIZAR
VARIABLES INT, LONG,
FLOAT, DOUBLE, CHAR.
(CU00510F)

Sección: Cursos

Categoría: Curso básico de programación en lenguaje C desde cero

Fecha revisión: 2031

Resumen: Entrega nº10 del curso básico "Programación C desde cero".

Autor: Mario Rodríguez Rancel

VARIABLES EN EL LENGUAJE C

Todo lo expuesto en relación a variables en el curso Bases de la programación nivel I será válido y útil con el lenguaje C, pero con matices, ya que cada lenguaje sigue sus propias normas y pautas. Si no sabes qué es una variable consulta el curso de Bases de la programación de aprenderaprogramar.com.

Vamos a tratar los aspectos más básicos de la declaración y uso de variables con C y a dar algunas orientaciones a través de las que profundizar en el manejo de variables.



NOMBRES Y TIPOS DE VARIABLES. DECLARACIÓN.

Para establecer nombres de variables sigue las pautas indicadas en el curso de Bases de la programación de aprenderaprogramar.com.

El lenguaje C obliga a declarar una variable antes de ser usada. Es decir, no podríamos escribir algo del tipo: *Mostrar velocidad01*, si antes no hemos "dicho" (declarado) que existe una variable con ese nombre. Declarar una variable no significa que se le asigne contenido, sino simplemente se indica que la variable existe. ¿Qué nos mostrará si intentamos mostrar o utilizar la variable si no le hemos asignado contenido? Hay lenguajes que asignan automáticamente contenido cero, vacío o falso a una variable, dependiendo de qué tipo sea. Sin embargo en C no está permitido usar una variable sin antes haberle asignado un contenido, a lo que se denomina "inicializar la variable". Por tanto, al tratar de ejecutar un programa donde una variable está sin inicializar puede producirse un error de compilación. Hay lenguajes donde la inicialización (o incluso la declaración) de las variables no es estrictamente obligatoria como PHP ó Visual Basic.

Nosotros vamos a trabajar con C por lo que habremos de declarar e inicializar las variables que usemos siempre. La declaración e inicialización de variables, aunque pueda resultar a veces "pesada", tienen una serie de ventajas que podemos resumir en servir para:

- Generar buenos programas.
- Evitar errores y confusiones.

En cuanto a los tipos de variables, los más habituales en C los expondremos a continuación.

TIPO DE DATOS	SE ESCRIBE	MEMORIA REQUERIDA*	RANGO ORIENTATIVO*	EQUIVALENCIA EN PSEUDOCÓDIGO	OBSERVACIONES
Entero	int	2 bytes	- 32768 a 32767	Entero	Uso en contadores, control de bucles etc.
Entero largo	long	4 bytes	- 2147483648 a 2147483647	Entero	Igual que int pero admite un rango más amplio

TIPO DE DATOS	SE ESCRIBE	MEMORIA REQUERIDA*	RANGO ORIENTATIVO*	EQUIVALENCIA EN PSEUDOCÓDIGO	OBSERVACIONES
Decimal simple	float	4 bytes	- 3,4 · 10 ³⁸ a 3,4 · 10 ³⁸	Real	Hasta 6 decimales. También admite enteros
Decimal doble	double	8 bytes	- 1,79 · 10 ³⁰⁸ a 1,79 · 10 ³⁰⁸	Real	Hasta 14 decimales. También admite enteros
Carácter	char	1 bytes	0 a 255	Alfanumérica	Carácter, independiente o parte de una cadena

* Podría variar

Existen otros tipos de variables en C como las tipo short (enteros cortos), unsigned int (enteros positivos o naturales) y long double (decimal doble con una mayor precisión).

A diferencia de la mayoría de lenguajes, C no incorpora entre sus tipos predefinidos el tipo booleano (true / false). El motivo para ello es que en realidad no es estrictamente necesario para permitir la programación, ya que puede emularse fácilmente por ejemplo usando un entero al que asignamos valor uno ó valor cero según sea verdadera o falsa una condición. No obstante, dado que la mayoría de los programadores están habituados a usar este tipo, será posible usarlo gracias a que C nos permite definir nuestros propios tipos de datos.

Los tipos enumerados (enum) son variables especiales que tienen un nombre "general" y luego *n* valores posibles, por ejemplo la variable "color" podría tener como valores posibles "rojo", "verde" y "azul". Hay más tipos de datos en C, pero nosotros no vamos a profundizar en su estudio.

Conociendo ya los tipos de variables básicas y cómo nombrarlas, veamos ahora cómo declararlas. La declaración de variables en C debe hacerse al principio del programa. Veremos ahora una forma básica que nos permita empezar a trabajar y, de paso, crear nuestro primer programa.

Usaremos para ello el tipo de variable *int* (entero). La sintaxis que usaremos será:

```
int [Nombre de variable];
```

Para facilitar la corrección y claridad de nuestros programas el lugar donde realizaremos la declaración de variables, al menos de momento, será después de la línea `int main()`, que constituye el inicio del código ejecutivo de nuestro programa. Crea un nuevo proyecto, denomínalo proyectoCurso2 y accede al código. Si no recuerdas cómo hacer esto, lee las explicaciones que hemos dado anteriormente. Una vez en la ventana de código, dentro del `int main() {...}` escribe:

```
int numeroDePlantas;
```

Has declarado la variable *numeroDePlantas* como tipo entero. También podrías asignarle un valor inicial a la variable en la misma línea que la declaras, de esta manera:

```
int numeroDePlantas = 15;
```

Este tipo de escritura nos permite declarar e inicializar la variable con un valor en una sola línea. No siempre lo haremos así, pero en algunas ocasiones nos puede resultar de interés. En otras ocasiones declararemos la variable en una línea y posteriormente le asignaremos contenido en otra.

Supón que declaras: *int edad*; como variable destinada a contener la edad de una persona. Sabemos que la edad de una persona puede oscilar entre cero y 150 años (siendo groseros), y que sus valores no son decimales. Por tanto puede declararse como tipo *int* sin ningún problema. ¿Qué ocurriría si la declaráramos como tipo *double*?

- a) Vamos a ocuparle (estimamos) 8 bytes al sistema cuando podríamos haber ocupado sólo 2. Es una falta de eficiencia y economía.
- b) A la hora de liberar de errores al programa (depurarlo) no sabremos a ciencia cierta qué tipo de datos contiene la variable *edad*, ya que puede contener tanto el valor entero 56 como el valor decimal 56.332. Hacemos al programa más difícil de entender.

A la hora de declarar variables conviene seguir las pautas que ya hemos comentado y, resumidamente:

- Elegir siempre el tipo de variable más sencillo posible. Consideraremos que el grado de sencillez viene definido por la memoria requerida (a menor memoria requerida mayor es la sencillez). Esto redundará en un menor consumo de recursos del sistema y en una mayor rapidez de las operaciones que realiza el ordenador.
- Realizar declaraciones ordenadas y claras.
- Elegir un nombre descriptivo y claro, que comience preferiblemente con minúsculas.

En C podemos declarar variables en múltiples líneas una por línea pero también varias variables en una línea. Existen distintas formas para realizar declaraciones. Veamos ejemplos:

Expresión abreviada	Equivalente
a) <code>int i, j, k;</code>	a') <code>int i; int j; int k;</code>
b) <code>int i, j, k;</code>	b') <code>int i; int j; int k;</code>
c) <code>int i=0, j=4, k=76;</code>	c') <code>int i=0; int j=4; int k=76;</code>
d) <code>int i=0, j=4, k=76;</code>	d') <code>int i=0; int j=4; int k=76;</code>

Las opciones a), b), c), d) dan lugar a un mismo resultado. Se declaran tres variables de tipo entero denominadas *i*, *j*, *k*. Las opciones a' - d'), escritas de forma extendida, también son válidas y con el mismo resultado. Nosotros nos decantamos por no usar declaraciones donde en una misma línea aparezcan variables de un tipo de forma abreviada porque pensamos que resulta menos claro a la hora de leer un programa. Aunque hay ocasiones en las que puede resultar razonable usar declaraciones abreviadas, por ejemplo si se trata de variables que todas ellas son contadores para bucles.

Ten en cuenta que la declaración *int i, j, k;* declara tres variables de tipo entero.

Una expresión como *float i; int j; long k;* es válida, aunque consideramos preferible no declarar distintos tipos de variables en una misma línea sino hacerlo en líneas separadas.

Si en un programa tenemos únicamente una declaración de variables como esta:

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

int main() {
    int i, j, k;
    return 0;
}
```

Si tratamos de ejecutar el programa (en CodeBlocks, en el menú "*Build*" elegimos la opción "*Build and run*") obtenemos un mensaje de error que será similar a este:

```
C:\ProyectosCursoC\proyectoCurso2\main.c | 7 | aviso: variable 'k' sin usar [-Wunused-variable] |
```

Esto es debido a que el compilador comprueba si las variables declaradas son usadas en el programa, y si no lo son, impide la ejecución y muestra un mensaje de error. No te preocupes por ello en este momento, ya que ahora únicamente estamos tratando de ver cómo se realiza la declaración de variables.

Una peculiaridad de C respecto a otros lenguajes es que no tiene un tipo predefinido "String" o cadena de caracteres. Aunque sí dispone de distintas funciones relacionadas con cadenas, cuando queramos declarar una variable como tipo alfanumérico tendremos que hacerlo declarando un array de caracteres. Es decir, en C las cadenas se declaran como vectores de tipo *char*. Por ejemplo la palabra "curso" estaría formada por cinco caracteres que son 'c', 'u', 'r', 's', 'o'. Veremos cómo trabajar con arrays y con variables alfanuméricas más adelante.

EJERCICIO RESUELTO

Realizar una declaración de variables para las siguientes propuestas de programas. No es necesario crear los programas, sino únicamente indicar cómo se haría la declaración de variables.

- a) Un programa que muestra un número decimal con la mayor precisión posible.
- b) Un programa que nos pide nuestra edad y nos muestra el año en que nacimos.
- c) Un programa que nos muestra el valor medio de tres magnitudes.

- d) Un programa que trabaja con: tres contadores para bucles, tres valores de momentos en una viga (de magnitud no conocida), valores de longitud de la viga, canto y ancho y un valor interruptor que permite o impide que se muestren resultados.
- e) Un programa que dado el diámetro de un círculo obtenga el valor del área del mismo.

SOLUCIÓN

- a) *double numero*; Cuando sea necesaria la mayor precisión posible usaremos el tipo *double*.
- b) *int edad*; Se podría definir otra variable llamada *annoNacimiento* según el programador lo estime oportuno o no. Evitaremos el uso de la letra ñ en el nombre de variables.
- c) *float magnitud1, magnitud2, magnitud3*;

float media;

Hemos utilizado el tipo *float* para cubrirnos las espaldas. No sabemos con qué tipo de valores vamos a trabajar y por ello cogemos un tipo de variables bastante amplio como es el *float*. Podríamos haber usado el tipo *double* si lo hubiéramos considerado necesario.

Hemos utilizado dos líneas pero sólo un tipo de variable ¿Por qué? El programador ordena la declaración de variables como estime más conveniente. En este caso ordenamos "valores para el cálculo" en una línea y "resultados" en otra. Resulta más fácil de leer y analizar que estando todo en una sola línea.

- d) *int i, j, k*;

float momento1, momento2, momento3;

float largo, canto, ancho;

int muestraResultados;

Siempre será preferible usar las expresiones *momento1*, *momento2* y *momento3* antes que *m1*, *m2* y *m3* que resultan más confusas. Sin embargo, este tipo de decisiones siempre quedan a criterio del programador. Para el caso del valor interruptor se supone que este tipo de variable sólo tendrá uno entre dos valores posibles: true ó false, 1 ó 0, 't' ó 'f'. Nosotros hemos optado por declarar el valor como tipo *int* de forma que un cero lo consideraremos equivalente a "falso" y un 1 lo consideraremos equivalente a "verdadero".

- e) *double diametro*;

double NumeroPi;

En este caso hemos supuesto que vamos a requerir la mayor precisión posible, de ahí que hayamos cogido tipos double. También hubiera resultado válido usar tipo float. El número pi es un valor constante, por lo que podremos declararlo como una constante en lugar de cómo una variable, como veremos más adelante. Evitaremos el uso de tildes ("letras acentuadas") en el nombre de variables.

EJERCICIO

Realizar una declaración de variables para las siguientes propuestas de programas. No es necesario crear los programas, sino únicamente indicar cómo se haría la declaración de variables.

- a) Un programa que trabaja con tres valores interruptores. La utilidad del primero es indicar si las unidades de trabajo están expresadas en el sistema internacional. La utilidad del segundo es indicar si el cálculo es determinista o no determinista. La utilidad del tercero es indicar si se deben mostrar todos los resultados o solo un resumen de resultados.
- b) Un programa que dado el radio de una esfera obtenga el valor de volumen de la misma.
- c) Un programa que muestra la altura de una puerta con la mayor precisión posible
- d) Un programa que nos pregunta el número de hermanos que tenemos (sin incluirnos a nosotros mismos) y nos muestra el número de hermanos totales que somos (incluyéndonos a nosotros).
- f) Un programa que trabaja con seis contadores para bucles.

Para comprobar si tus respuestas son correctas puedes consultar en los foros [aprenderaprogramar.com](http://www.aprenderaprogramar.com).

Próxima entrega: CU00511F

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=82&Itemid=210