

Language “C”

Primeros Pasos

Origen y Evolución

- Entre 1969 y 1973, Dennis Ritchie, crea el lenguaje en los Laboratorios Bell.
- Modifica el Lenguaje B, creado por Ken Thompson en 1969.
- Reescribe el Sistema Operativo Unix
- En 1978, Dennis Ritchie y Brian Kernighan, publican la primer edición del libro “El Lenguaje de programación C”
- A mediados de los 80, se crea el Lenguaje C++, que incorpora orientación a objetos.
- En 1988, Ritchie y Kernighan, publican la segunda edición de su libro
- En 1989 se crea el primer estándar, conocido como ANSI C
- Posteriormente aparecen los estándares ISO, C89 y C90
- GNU/Linux

Características

- **Es compilado** (utiliza un compilador que convierte el programa fuente en un programa en lenguaje de máquina, que puede ser ejecutado por esta)
- **Es Imperativo** (pertenece al grupo de los lenguajes cuyo tipo de programación es imperativa, mediante instrucciones se indica la secuencia de pasos a dar; a diferencia de la declarativa en la que uno indica que es lo que desea obtener o que se está buscando)
- **Es de Alto nivel** (Es cercano al lenguaje humano, lejano al lenguaje de máquina. Podría considerarse de nivel intermedio, por la posibilidad que da de manejar ciertos recursos como la memoria, que otros lenguajes de alto nivel, no lo tienen)
- **Fuertemente tipado** (una variable con un valor de un tipo determinado, no se puede usar como si fuera de otro tipo, a menos que se haga una conversión)
- **Es Estructurado** (Permite escribir programas siguiendo las reglas del paradigma de la programación estructurada)

Compilador Vs Intérprete

Compilador		Intérprete	
Es el encargado de traducir nuestro programa fuente, usualmente a lenguaje de máquina		Es el encargado de traducir instrucciones de código a medida que se solicitan, instrucción por instrucción	
Ventajas	Desventajas	Ventajas	Desventajas
Listo para ser Ejecutado	No es multiplataforma	Es multiplataforma	Se requiere intérprete instalado
Usualmente más rápido	Poco flexible	Fácil de testear	Más lento
El código fuente es privado	Es necesario realizar pasos adicionales	Fácil de buscar y corregir errores (Debugging)	El código fuente es público

Algunos elementos del lenguaje

- Primer Programa
- Definición de Variables
- Operadores Aritméticos
- Operadores de Comparación
- Operadores Lógicos
- Estilos de programación, comentarios y otros
- Estructura de Control Condicional - If
- Ayuda y documentación

Definiendo Variables

[Tipo de Dato] [Nombre de la Variable]

ó

[Tipo de Dato] [Nombre de la Variable] = [Valor]

↑
Operador de Asignación

int x

ó

int x = 100

Cómo muestro el valor de una variable?

```
printf( "%i", edad )
```

↑
Carácter de formateo

```
printf( "Edad: %i", edad )
```

Formateador	Salida
%d ó %i	entero en base 10 con signo (int)
%u	entero en base 10 sin signo (int)
%o	entero en base 8 sin signo (int)
%x	entero en base 16, letras en minúscula (int)
%X	entero en base 16, letras en mayúscula (int)
%f	Coma flotante decimal de precisión simple (float)
%lf	Coma flotante decimal de precisión doble (double)
%e	La notación científica (mantisa / exponente), minúsculas (decimal precisión simple ó doble)
%E	La notación científica (mantisa / exponente), mayúsculas (decimal precisión simple ó doble)
%c	caracter (char)
%s	cadena de caracteres (string)

Formateador	Salida
%07i	largo mínimo de 7 dígitos, justificado a la derecha, rellena con ceros
%.7i	justificado a la derecha, 7 dígitos de largo, sin relleno
%8.2f	tamaño total de 8 dígitos, con dos decimales
%.*f",x,d)	tamaño predeterminado,x numeros decimales
%.*.*f",x,y,d)	tamaño igual a x, y numeros decimales
%s	cadena terminada en null
%5s	primeros cinco caracteres o delimitador
%.5s	primeros cinco caracteres, sin tener en cuenta el delimitador
%20.5s	primeros cinco caracteres, justificados a la derecha, con 20 caracteres de largo
%-20.5s	primeros cinco caracteres, justificados a la izquierda, con 20 caracteres de largo

Cómo solicito el ingreso de un valor para una variable?

scanf (tipo, &variable)

scanf("%i", &edad)

Operadores Aritméticos

Operación	Operador
Suma	+
Resta	-
Multiplicación	*
División	/
Resto de la División	%

Operadores de Comparación

Comparaciones	
Mayor	>
Menor	<
Igual	==
Distinto	!=
Mayor Igual	>=
Menor Igual	<=

Operadores Lógicos

Conector Lógico	
Conjunción - Y	&&
Disjunción - O	
Not lógico	!

Delimitando Bloques y respetando un estilo

- Un bloque de código va encerrado entre `{ }`
- Las instrucciones terminan con `;`
- Para **comentarios de varias líneas**, iniciar con `/*` y finalizar con `*/`
- Para **comentarios de una línea** o a continuación de código, anteponer `//`.
- Indentar el código para una mejor legibilidad, utilizando una tabulación de 4 espacios
- Utilizar minúsculas para las instrucciones
- Utilizar nombre descriptivos en minúsculas, para variables y funciones. Cuando estén compuestos por más de una palabra, unirlos mediante `_`
Por ejemplo: `id_cliente`
- Las constantes, escribirlas en mayúsculas