Template de presentación

Introducción a Lenguaje C

- Historia y Estándares
- Estructura de programa en C
- Tipos de Datos
- Variables - Constantes - literales
- Operadores
- Ingreso y Egreso de Datos
- Secuencias de escape

DENNIS RITCHIE

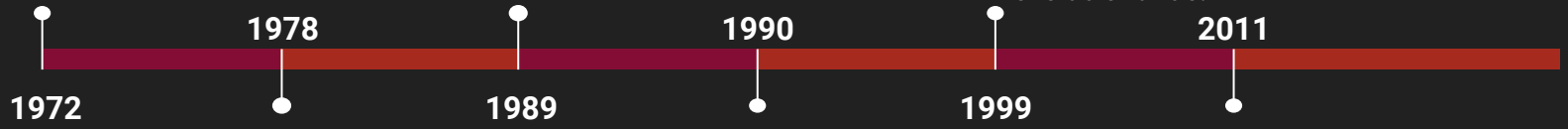
El desarrollo inicial de C se llevó a cabo en los Laboratorios Bell de AT&T entre 1969 y 1973

ANSI C

En 1983, el Instituto Nacional Estadounidense de Estándares (ANSI) organizó un comité, X3j11, para establecer una especificación estándar de C.

C99

Tras el proceso de estandarización de ANSI, la especificación del lenguaje C permaneció relativamente estable durante algún tiempo, mientras que C++ siguió evolucionando.



kERNIGHAN & RITCHIE

Ritchie y Brian Kernighan publicaron la primera edición de El lenguaje de programación C, también conocido como La biblia de C

ANSI/ISO C

En 1990, el estándar ANSI (con algunas modificaciones menores) fue adoptado por la Organización Internacional para la Estandarización (ISO) en el estándar ISO/IEC 9899:1990.

C11

C11 (antes conocido como C1X) es un nombre informal para ISO/IEC 9899:20118. El borrador final, N1570, fue publicado en abril de 2011.

Tipos de Lenguaje y Compiladores

De Máquina

Utilización de lenguaje o código máquina interpretable directamente por el microprocesador.
Lenguaje compuesto por conjunto de instrucciones que indican acciones para la máquina.

- Ensamblador
- Lenguaje Máquina

Compilacion

Convertir de lenguaje fuente a lenguaje de máquina, generando así un programa o archivo ejecutable. Es decir la conversión se realiza toda completa de una vez.

- C
- C++
- Swift
- Rust

Interpretacion

Traduce de a partes, típicamente de a una instrucción del lenguaje fuente. Se traduce, se ejecuta lo traducido y se repite. La velocidad de ejecución es una de las principales desventajas de esta estrategia.

- Python
- Lisp
- PHP
- VBScript

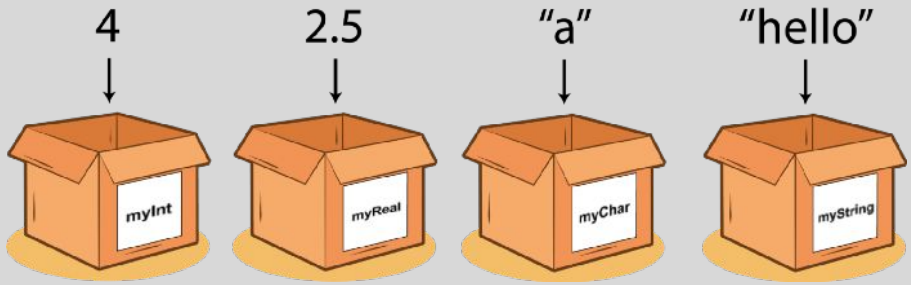
Compilador

Codigo	Pre procesador	Compilador	Linker
Código fuente , escrito por el programador. En el cual se describen las secuencias lógicas del programa. Contiene Comentarios, y directivas del preprocesador siempre precedidas por el simbolo #.	Toma como entrada los archivos fuentes que componen el programa y se encarga de eliminar comentarios, e interpretar y procesar directivas del preprocesamiento: #include: Sustituye la línea por el contenido del fichero especificado. #define: Define una constante (identificador) simbólico.	Analiza la sintaxis y la semántica del código fuente preprocesado y lo traduce, generando un fichero que contiene el código objeto. En el proceso de compilación se interpreta el código fuente salvo las referencias a objetos externos (funciones o variables) en el módulo de código que se está compilando.	Toma los archivos objetos y los combina en una imagen (archivo) ejecutable. Completa las llamadas de un objeto a otro, reemplazando las “anotaciones” dejadas por el compilador con la dirección o desplazamiento real para acceder al código requerido.

Programar en C es como comer una naranja

Variables

En programación tenemos variables, que son espacios de memoria donde se guardan datos, podemos pensarlos como cajas con un nombre donde guardamos números, letras, palabras, direcciones de memoria, etc.



Tipos de datos y tamaño

Existen diferentes tipos de datos en C, las variables, punteros, funciones, etc, pueden tomar esos tipos, para guardar por ejemplo variables de números naturales, variables de letras, variables de números con coma, etc. Cada una de estas variables tiene y ocupa diferentes tamaños en la memoria del procesador.

Tipos de variables

char	1 byte	• Representa caracteres
int	2 byte a 4 byte	• Unsigned quita el signo tomando valores: 0 a 65535 • Valores enteros: -32768 a 32767
float	Valores decimales: 3.4 e^-38 a 3.4 e^38	• Donec risus dolor porta venenatis • Pharetra luctus felis • Proin in tellus felis volutpat
Double	Valores decimales: 1.7 e^-308 a 1.7 e^308	• Donec risus dolor porta venenatis • Pharetra luctus felis • Proin in tellus felis volutpat
bool	Toma valores de Verdadero o falso	• Donec risus dolor porta venenatis • Pharetra luctus felis • Proin in tellus felis volutpat

Declaración de las variables

Ejemplo para la declaración de las 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;
```

Operadores

Aritmeticos

- Asignacion -> =
- Suma y Resta -> + , -
- Multiplicacion y divicion-> * , /
- Modulo -> %

Relacionales

- Menor y Mayor -> < , >
- Menor o igual y mayor o igual-> <= ,>=
- Igualdad y Distinto-> == , !=

Logicos

- NOT (Negacion) -> |
- AND -> &&
- OR-> ||

Bits

- Y -> &
- O (inclusivo) -> |
- O (Exclusivo)-> ^
- Desplazamiento a la derecha-> >>
- Desplazamiento a la izquierda-> <<
- Complemento a uno-> ~

Funciones de entrada y salida

- # 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 %

```
printf("Valor de variable: %d \n", numero_ingresado);

printf("Valor de variable multiplicado: %d \n", numero_ingresado * 2);

printf("Valor de variable multiplicado: %d \t numero literal: %d \n", numero_ingresado * 5, 7);
```

- # Para ingresar un valor en la variable dato

```
int dato ;
scanf("%d", &dato) ;
printf("Valor ingresado en variable dato: %d \n", dato);
```

Caracteres de lectura/escritura en scanf y printf

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)	

Caracteres de escape printf

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			

Estructura básica de un programa en C

Estructura general (simplificada) de un programa C:

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

```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;
}
```

SI NO COLOCAN ; AL FINAL DE CADA SENTENCIA

CONSTANTES

Las constantes nos permite identificar, parámetros numéricos o strings, que no cambian en todo el funcionamiento del programa

Modo “tradicional”

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

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

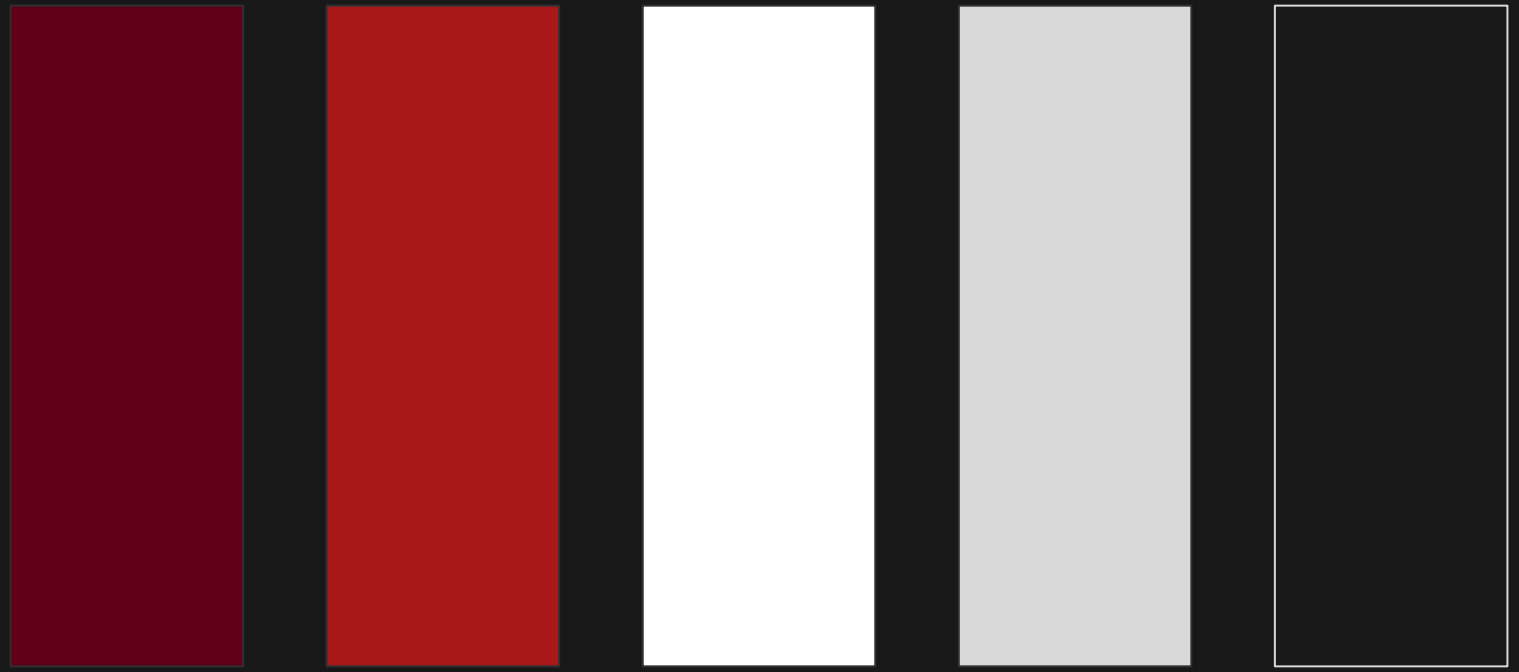
```
const double PI = 3.14159;
perimetro = 2 * PI * radio;
PI = 4; // ERROR al compilar !!
```

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)
```

ATENCION

Paleta de colores



Consejo

ATENCION

Pregunta