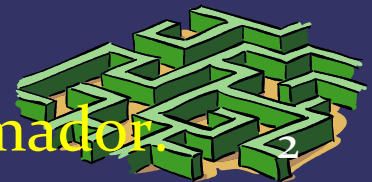


Lenguaje C

Características de C

- ✿ Sensible a mayúsculas y minúsculas: `sum` y `Sum`
- ✿ Identación.
- ✿ Sentencias (terminan con un punto y coma).
- ✿ Bloques (delimitados entre llaves).
- ✿ Elementos de un programa:
 - ✗ Palabras reservadas (muy pocas).
 - ✗ Funciones de librería estándar.
 - ✗ Variables y funciones definidas por el programador.



Palabras Reservadas

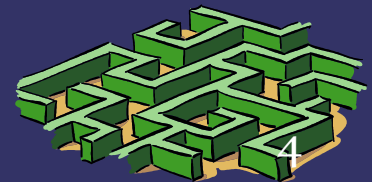
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while



Tipos de errores

Al compilar un programa se producen dos tipos de errores:

- ✿ **ERRORES:** por lo general son fallas de sintaxis, operaciones no permitidas, etc. Un programa no podrá ejecutarse hasta que todos los errores hayan sido corregidos.
- ✿ **PRECAUCIONES (WARNINGS):** son sentencias en las que el compilador sospecha que puede haber problemas. Si bien un programa con warnings puede ejecutarse, los resultados son impredecibles, por lo que es conveniente corregir las sentencias problemáticas.



Estructura de un programa en C

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

← Archivo de cabecera

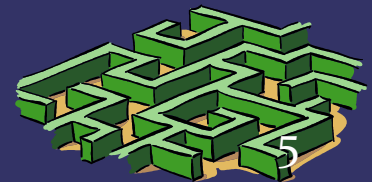
```
int main()
```

← Cabecera de la función

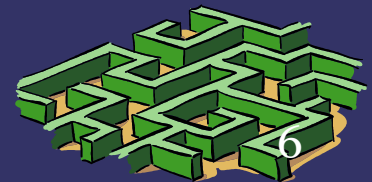
Nombre de la
función

```
{
    printf("Hola mundo!!\n");
    return(0);
}
```

Cuerpo de la
función



- ✱ La directiva **#include** indica al compilador que lea el archivo de cabecera que se nombra a continuación. Estos archivos con extensión **.h** contienen código en C
- ✱ Cada programa C tiene una función **main()** que es el punto de entrada al programa. Un programa debe tener solo una función **main()**.

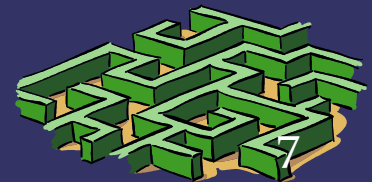


Comentarios

- Los comentarios en C pueden ocupar varias líneas y se encuentran delimitados entre `/*` y `*/`.

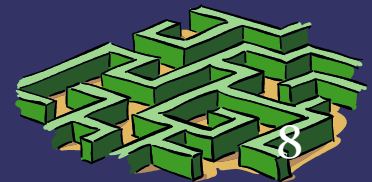
```
int main()
{
    /* Esto es un comentario de varias
       líneas.*/
    // Este también es un comentario

    return(0);
}
```



Variables y Constantes

- ✿ Variable: en C es una posición de memoria con nombre en donde se almacena un valor de un cierto tipo de dato. Los tipos de datos pueden ser: cadenas, números y estructuras.
- ✿ Constante: es una variable cuyo valor no puede ser modificado.



Declaración de Variables

Es una sentencia que proporciona información de la variable al compilador C. Sintaxis:

tipo variable

tipo: es el nombre de un tipo de dato conocido por el C.

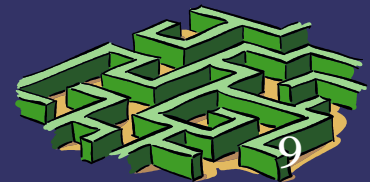
variable: es un identificador (nombre) válido en C.

La declaración de variables debe situarse siempre al principio del bloque.

Inicialización de variables

tipo nombre_variable = expresión

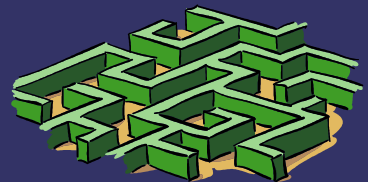
expresión: es cualquier expresión válida cuyo valor es del mismo tipo que *tipo*.



Tipos de datos

En C existen cinco tipos de datos según puede verse en la tabla siguiente:

Tipo de dato	Descripción
char	Carácter
int	Entero
float	Punto flotante
double	Punto flotante de mayor rango
void	Sin tipo (uso especial)



Tipos de datos - Modificadores

Existen cuatro modificadores de tipo, estos permiten cambiar el tamaño de los tipos de datos anteriormente especificados.

Modificador	Tipo de modificador		Descripción
signed	char	int	Con signo (por defecto)
unsigned	char	int	Sin signo
long	int	double	Largo
short	int		Corto

Ejemplo de declaración de variables:

```
unsigned char a;  
long double b;  
short int i;
```



Entradas y Salidas

La biblioteca C que proporciona las funciones para la entrada/salida es **stdio.h**

Salida: la función **printf()** visualiza en pantalla datos del programa.

```
printf(cadena_de_control , dato1, dato2,... )
```

cadena_de_control contiene los tipos de datos y forma de mostrarlos

dato1, dato2 variables, constantes, datos de salida.



Entrada: la función más utilizada para entrada formateada es **scanf()**

scanf(cadena_de_control , var1 , var2,....)

cadena_de_control: contiene los tipos de datos y si se desea su ancho.

var1, var2: variables del tipo de los códigos de control

Ejemplo:

```
printf ("Introduzca v1 y v2:");
```

```
scanf("%d %f", &v1, &v2);
```

Las variables que se pasan a scanf() se transmiten por referencia para poder ser modificadas y transmitir los datos de entrada, para ello se hacen preceder de &.



Ejemplo

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

```
int main (void)
```

```
{
```

```
    int A=8,resultado = 1;
```

```
    resultado = resultado * A;
```

```
    A = resultado + A;
```

```
    printf ("el resultado es: %d\n", resultado);
```

```
    printf ("el valor de A es: %d\n",A);
```

```
}
```



La cadena de **control** contiene códigos de formato que se asocian uno a uno con los datos. Cada código comienza con el carácter %, seguido del ancho mínimo del dato y termina con el carácter de conversión.

Códigos de formatos mas utilizados

%d	El dato se convierte a entero decimal
%o	El dato entero se convierte a octal
%x	El dato entero se convierte a hexadecimal
%c	El dato se considera de tipo carácter
%s	El dato ha de ser una cadena de caracteres
%f	El dato se considera de tipo float



Ejemplo

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

```
int main (void)
```

```
{
```

```
    int A,resultado ;
```

```
    printf ("Ingrese el valor de A: ");
```

```
    scanf ("%d",&A);
```

```
    printf ("Ingrese el valor de resultado: ");
```

```
    scanf ("%d",&resultado);
```

```
    resultado = resultado * A;
```

```
    A = resultado + A;
```

```
    printf ("el resultado es: %d\n", resultado);
```

```
    printf ("el valor de A es: %d\n",A);
```

```
}
```



Operadores Binarios: es cuando el operador actúa sobre dos valores (5 + 8)

Operadores Unitarios: es cuando el operador actúa sobre un solo valor (- 5)

Operadores de asignación

Símbolo	Uso	Descripción
=	a = b	Asigna el valor de b a a
*=	a *= b	Multiplica a por b y asigna el resultado a la variable a
/=	a /= b	Divide a entre b y asigna el resultado a la variable a
%=	a %= b	Fija a al resto de a/b
+=	a += b	Suma b y a y lo asigna a la variable a
-=	a -= b	Resta b de a y asigna el resultado a la variable a



OPERADORES

Operadores aritméticos

Operador	Tipos enteros	Tipos reales	Ejemplo
+	Suma	Suma	$x + y$
-	Resta	Resta	$b - c$
*	Producto	Producto	$x * y$
/	División entera: cociente	División en coma flotante	$b / 5$
%	División entera: resto		$b \% 5$



OPERADORES

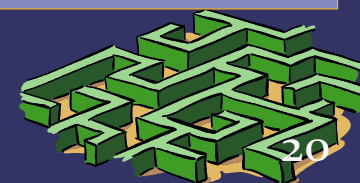
Precedencia de operadores matemáticos básicos

Operador	Operación	Nivel de Precedencia
$+$ $-$	$+25$ -6.745	1
$*$ $/$ $\%$	$5 * 5$ es 25 $25 / 5$ es 5 $25 \% 6$ es 1	2
$+$ $-$	$2 + 3$ es 5 $2 - 3$ es -1	3



Operadores Relacionales

Operador	Significado	Ejemplo
$=$	Igual a	$a = b$
\neq	No igual a	$a \neq b$
$>$	Mayor que	$a > b$
$<$	Menor que	$a < b$
\geq	Mayor o igual que	$a \geq b$
\leq	Menor o igual que	$a \leq b$



OPERADORES

Operadores Lógicos		
Operador	Operación lógica	Ejemplo
Negación (!)	No lógica	!(x >= y)
Y lógica (&&)	Operando_1 && Operando_2	m < n && i > j
O Lógica (! !)	Operando_1 ! ! Operando_2	m = 5 ! ! n != 10

