

# Unidad 4

Tipos de Datos y Palabras Reservadas

# Tipos de Datos

- ▶ char
- ▶ int
- ▶ float
- ▶ double
- ▶

# Palabras Reservadas

<i>Palabra reservada</i>	<i>Descripción</i>
<code>typedef</code>	Instrucción
<code>union</code>	Especificador de tipo
<code>unsigned</code>	Especificador de tipo
<code>void</code>	Especificador de tipo
<code>volatile</code>	Modificador de clase de almacenamiento
<code>while</code>	Instrucción

**TABLA 4.1** Palabras reservadas de C

<i>Palabra reservada</i>	<i>Descripción</i>
auto	Especificador de clase de almacenamiento
break	Instrucción
case	Instrucción
char	Especificador de tipo
const	Modificador de clase de almacenamiento
continue	Instrucción
default	Etiqueta
do	Instrucción
double	Especificador de tipo
else	Instrucción
enum	Especificador de tipo
extern	Especificador de clase de almacenamiento
float	Especificador de tipo
for	Instrucción
goto	Instrucción
if	Instrucción
int	Especificador de tipo
long	Especificador de tipo
register	Especificador de clase de almacenamiento
return	Instrucción
short	Especificador de tipo
signed	Especificador de tipo
sizeof	Operador
static	Especificador de clase de almacenamiento
struct	Especificador de tipo
switch	Instrucción

# Nota

- ▶ Todas las palabras reservadas del lenguaje C son con minúsculas. Este es sensitivo al uso de mayúsculas y minúsculas. Así que `int` es un tipo de dato y palabra reservada, pero `INT` no!

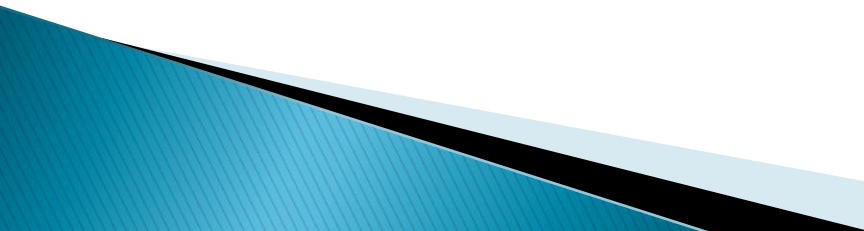
# El tipo de dato char

- ▶ Ejemplo
- ▶ A..Z carácter
- ▶ a...z carácter
- ▶ 0...9 número
- ▶ La computadora solo puede almacenar código numérico. Así que a-z, A-Z tienen un código numérico único. Un carácter ocupa 8 bits es decir un byte para almacenar su código numérico.
- ▶ ASCII (Código Estándar Estadounidense para el Intercambio de la Información)

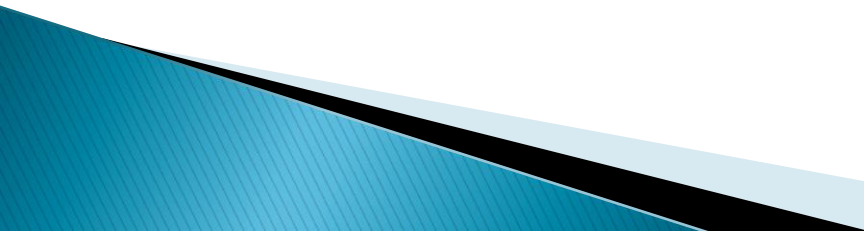
# Variables tipo carácter

- ▶ `char nombre_var`
- ▶ `char nombre_var1, ..., nombre_var9`

## Constantes de carácter

- `char x`
  - `x='A'`
  - `x=65`
- 

# El carácter de escape (\)

- ▶ \n
  - ▶ nueva línea
  - ▶ \b
  - ▶ Carácter de retroceso. Mueve el cursor hacia la izquierda.
  - ▶ \f
  - ▶ Carácter de salto de pagina
  - ▶ \r
  - ▶ Carácter de retorno. Inicio de la línea actual
  - ▶ \t
  - ▶ Carácter tabulador
- 



# Impresión de caracteres

- ▶ `printf()`
- ▶ `%c`
- ▶ `c1 = 'A';`
- ▶ `c2 = 'a';`

# Programa Imprima caracteres en pantalla

- ▶ `#include <stdlib.h>`
- ▶ `#include <stdio.h>`
- ▶ `#include <conio.h>`
  
- ▶ `int main()`
- ▶ `{`
- ▶ `char c1, c2;`
- ▶ `c1 = 'A';`
- ▶ `c2 = 'a';`
  
- ▶ `printf("Convierte a carácter c1 = %c \n", c1);`
- ▶ `printf("Convierte a carácter c2 = %c \n", c2);`
- ▶ `getch();`
- ▶ `return 0;`
- ▶ `}`

# Programa convierta valores numéricos a caracteres

- ▶ `c1 = 65;`
- ▶ `c2 = 97;`

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

- `int main()`
- `{`
- `char c1, c2;`
- `c1 = 65;`
- `c2 = 97;`
- `printf("Convierte a carácter c1 = %c \n", c1);`
- `printf("Convierte a carácter c2 = %c \n", c2);`
- `getch();`
- `return 0;`
- `}`

# El tipo de datos int

- ▶ Longitud
- ▶ 32 bits  $2^{31} - 1$
- ▶ 16 bit  $2^{15} - 1$

## Declaración de int

- Programa que imprima enteros (+, -)

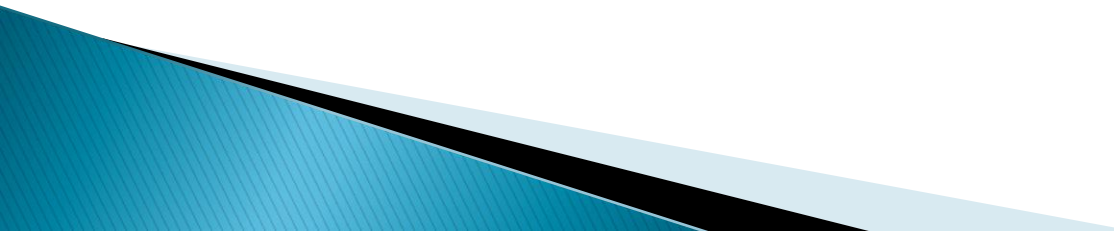
# Cómo mostrar los valores numéricos de los caracteres

- ▶ `printf()`
- ▶ `%c`
- ▶ `%d`

# Programa que muestre los valores numéricos de los caracteres

- ▶ `#include <stdlib.h>`
- ▶ `#include <stdio.h>`
- ▶ `#include <conio.h>`
  
- ▶ `int main()`
- ▶ `{`
- ▶ `char c1, c2;`
- ▶ `c1 = 'A';`
- ▶ `c2 = 'a';`
  
- ▶ `printf("Convierte a carácter c1 = %d \n", c1);`
- ▶ `printf("Convierte a carácter c2 = %d \n", c2);`
- ▶ `getch();`
- ▶ `return 0;`
- ▶ `}`

# El tipo de datos float

- ▶ Rango
  - ▶  $\pm 1.0 \times 10^{37}$
  - ▶ Declaración de variables
  - ▶ `float nombre;`
  - ▶ Especificador float (%f) función printf
- 

# Impresión de float

- ▶ `#include <stdlib.h>`
- ▶ `#include <stdio.h>`
- ▶ `#include <conio.h>`
  
- ▶ `int main()`
- ▶ `{`
- ▶ `int i1,i2;`
- ▶ `float f1,f2,f3;`
- ▶ `i1 = 125/10;`
- ▶ `i2 = 987/87;`
- ▶ `f1 = 147.14/10.0;`
- ▶ `f2 = 36.9/4;`
- ▶ `f3 = 789/12.5;`
  
- ▶ `printf("division entera, i1, 125/10= %d \n", i1);`
- ▶ `printf("division entera, i2, 987/87= %d \n", i2);`
- ▶ `printf("division punto flotante, f1, 147.14/10.0= %f \n", f1);`
- ▶ `printf("division punto flotante, f2, 36.9/4= %f \n", f2);`
- ▶ `printf("division punto flotante, f3, 789/12.5= %f \n", f3);`
  
- ▶ `getch();`
- ▶ `return 0;`
- ▶ `}`



# Notación científica (%e, %E)

- ▶ Ejemplos

- ▶ 5000

- ▶ 5e3

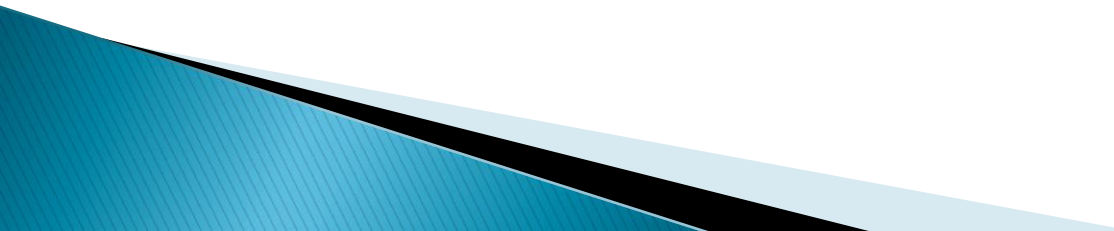
- ▶ -300

- ▶ -3e2

- ▶ 0.0025

- ▶ 2.5e-3

# El tipo de datos double

- ▶ double similar a float
  - ▶ double tiene el doble de bits que float
  - ▶ float (6 posiciones decimales)
  - ▶ double (10 posiciones)
- 

# Preguntas y Ejercicios

3. ¿Cómo puede representar los siguientes números en notación científica?
  - 3500
  - 0.0035
  - -0.0035
4. ¿Son válidos los siguientes nombres de variables?
  - 7o\_Calculo
  - Método\_de\_Tom
  - \_marca
  - Etiqueta\_1

## Ejercicios

1. Escriba un programa que imprima los valores numéricos de los caracteres Z y z.
2. Dados los valores numéricos 72 y 104, escriba un programa que imprima los dos caracteres correspondientes.
3. ¿Se puede asignar el valor de 72368 a una variable entera de 16 bits?
4. Dada la declaración `double dbl_num = 123.456;`, escriba un programa que imprima el valor de `dbl_num` en los formatos tanto de punto flotante como de notación científica.
5. Escriba un programa que pueda imprimir el valor numérico del carácter de nueva línea (`\n`). (Sugerencia: asigne `'\n'` a una variable de tipo carácter.)